

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

**FLUID FLOW-INSPIRED CURVATURE-AWARE PRINT-PATHS
FROM HEXAHEDRAL MESHES FOR ADDITIVE MANUFACTURING**



M.Sc. THESIS

Serhat ÇAM

Department of Mechanical Engineering

Mechanical Design Programme

DECEMBER 2022

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

**FLUID FLOW-INSPIRED CURVATURE-AWARE PRINT-PATHS
FROM HEXAHEDRAL MESHES FOR ADDITIVE MANUFACTURING**

M.Sc. THESIS

**Serhat ÇAM
(503191217)**

Department of Mechanical Engineering

Mechanical Design Programme

Thesis Advisor: Assoc. Prof. Dr. Erkan GÜNPINAR

DECEMBER 2022

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

**EKLEMELİ İMALAT İÇİN ALTIYÜZLÜ-ELEMAN AĞLARI İÇİN
AKIŞKAN AKIŞINDAN ESİNLENİLEN EĞRİLİĞE DUYARLI
BASKI YOLLARI**

YÜKSEK LİSANS TEZİ

**Serhat ÇAM
(503191217)**

Makina Mühendisliği Anabilim Dalı

Makina Mühendisliği Konstrüksiyon Programı

Tez Danışmanı: Doç. Dr. Erkan GÜNPINAR

ARALIK 2022

Serhat ÇAM, a M.Sc. student of ITU Graduate School student ID 503191217, successfully defended the thesis entitled “FLUID FLOW-INSPIRED CURVATURE-AWARE PRINT-PATHS FROM HEXAHEDRAL MESHES FOR ADDITIVE MANUFACTURING”, which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Assoc. Prof. Dr. Erkan GÜNPINAR**
Istanbul Technical University

Jury Members : **Prof. Dr. Cevat Erdem İmrak**
Istanbul Technical University

Assist. Prof. Dr. Kemal Mert Doğan
Yıldız Technical University

Date of Submission : 10 December 2022

Date of Defense : 22 December 2022



FOREWORD

This dissertation is presented for completion of a master's degree of Mechanical Design Programme at the Istanbul Technical University (ITU). The study was conducted under the supervision of Dr. Erkan Günpınar, who is the mentor of Smart CAD Laboratory in Mechanical Engineering Department of ITU.

The basic motivation of the thesis is to develop a new print-path generator for any hexahedral mesh, which involves a path direction selector without generating cross-wise contact, a new curvature-aware print-path generator, and lastly print-path ordering approach for multi-axis manufacturing.

I would like to pay my sincere gratitude to my thesis supervisor, Dr. Erkan Günpınar, who assisted me all the time by guiding me and sharing his knowledge and experience with me generously during my thesis study. I would like to thank our lab members of Smart Cad Laboratory, Alper Tasmektepligil, Arash Armanfar, and Emre Gunaydin for their valuable friendship and support. Additionally, I would like to thank Umut Can Coskun to provide images of streamlines for fluid flow, The Scientific Research Center of Istanbul Technical University for supporting this thesis study (Project Number: 44308).

December 2022

Serhat ÇAM
(Mechanical Engineer)



TABLE OF CONTENTS

	<u>Page</u>
FOREWORD.....	vii
TABLE OF CONTENTS.....	ix
ABBREVIATIONS	xi
SYMBOLS.....	xiii
LIST OF TABLES	xv
LIST OF FIGURES	xvii
SUMMARY	xix
ÖZET	xxi
1. INTRODUCTION	1
1.1 Thesis Motivation	1
1.2 Proposed Approach	2
1.3 Scope	5
1.4 Outline	6
2. RELATED WORKS	7
2.1 Hexahedral Mesh Generation	7
2.2 Generation of Non-planar AM Print-paths	8
3. METHODS.....	11
3.1 Terminology	11
3.2 Generating Fluid Flow-inspired Print-paths from a Hexahedral Mesh	12
3.2.1 Fluid flow-inspired print-paths	13
3.2.2 Assigning contact types for a block.....	19
3.2.3 A part representing a block-mesh under a loading condition	25
3.2.4 Generating layers and print-paths in a block	26
3.3 Print-path Planning for Multi-axis AM	30
4. RESULTS AND DISCUSSION	33
4.1 Computational Time	40
4.2 Virtual and Physical Validation of Print-paths	41
5. CONCLUSIONS AND FUTURE WORKS.....	47
REFERENCES.....	49
CURRICULUM VITAE	53

ABBREVIATIONS

AM	: Additive Manufacturing
STL	: Standard Triangle Language
FDM	: Fused Deposition Modeling
PC	: Personal Computer
CAD	: Computer Aided Design
B-rep	: Boundary Representation
FEA	: Finite Element Analysis
BC	: Bounding Cone
CC	: Circular Contact
RC	: Curve Contact



SYMBOLS

M	: Hexahedral Mesh
$H, \mathcal{H}$	: Hexahedron, a hexahedral mesh element
V	: Vertices
E	: Edges
Q	: Quadrilateral Face
b	: Block, obtained from mesh decomposition
n_M, n_P, n_R	: Number of vertices in three direction
$G1, \dots, G6$	: Path Assignment Operators
$B, \mathcal{B}$	: Block-mesh
$\mathcal{V}$	: Block-vertices
$\mathcal{E}$	: Block-edges
$\mathcal{Q}$	: Block-faces
e	: Block-edge
f	: Block-face
$f.b$	: The block-faces in block
$f.c$	: The contact type for block-face
$\mathcal{C}$	: The data structure stores contact types
$c_0, \dots, c_5$	: The contact types for each block-face
S_j	: Print-paths
E_1, E_2	: Energy funtions
F_1, F_2	: Energy funtions
ϑ_q	: The principal stress vector
τ_1	: The user-defined threshold for principal stress vector length
τ_2	: The user-defined threshold for angle between to vectors
α_q	: The angle between the vectors
α	: The weighted average of the angles
$b.c$	: The number of contact types assigned for the block-face
$b.e$	: The block-edge
$e.v$	: The valance of a block-edge
L	: The layer of a block
$q[i, j]$	: The two-dimensional array of quads/sub-layers
$p[i, j]$	: The two-dimensional array of quad-points
r	: The print-path radius
$c_{i,j}^{i+1,j}$	: The contact type for quad edges
T	: The print-path
t_k	: The print-path parameter
n_k	: The normal at given print-path parameter
p_k	: The point at given print-path parameter
$\mathcal{V}(T)$	: The sweep volume



LIST OF TABLES

Page

Table 4.1 : Computational times of the algorithm steps for different models (n_{hex} : number of hexahedra of a hexahedral mesh, n_b : number of blocks after the block decomposition (Section 3.1), r : print-path radius, t_a : computational time (in milliseconds) for assigning contact types for blocks (Algorithm 1), t_p : computational time (in seconds) for generating layers and print paths (Section 3.2.4)).....	40
--	-----------



LIST OF FIGURES

	<u>Page</u>
Figure 1.1 : Given a hexahedral mesh model (a), curvature-aware print-paths (in green) [3] for a bone model are generated (b). Observe cross-wise contacts (enclosed by green rectangles) between print-paths. Pink print-paths highlight this problem, which exists in neighboring regions. Fluid flow-inspired print-paths (d) can prevent such cross-wise contacts, which are likened to streamlines of (laminar) fluid flow modeled inside a bone model (c).	3
Figure 1.2 : A column hexahedral mesh model (a) is decomposed into blocks (b, cuboid-like sub-volumes). Contact types (circular or curve) between block-faces and print-paths are then determined. Dark blue and light green represent circular or curve contact types, respectively(c). Layers (in red) are then obtained (d). Print-paths are then generated inside blocks while respecting pre-determined contact types (e). Finally, collision-free print-path planning is carried out (yellow cones representing extruder in (f)).....	4
Figure 1.3 : Work flow for the proposed method in this dissertation.	5
Figure 3.1 : A block, b , (in green) with twelve edges (in yellow), eight corner points (in red) and six faces.	11
Figure 3.2 : A block (on the right) having circular (0, blue circles) and line (1, red lines) contact types between print-paths and block-faces can be covered using print-paths (on the left).....	13
Figure 3.3 : Two-dimensional laminar streamlines (top row) and their corresponding print-paths (bottom row). $G_1, G_2, G_3, G_4, G_5, G_6$ partial print paths for contact type assigned sub-layers, respectively (a,b,c,d,e,f).Green dots show to start or end points for sub-layer print paths, the dark blue line shows the trimmed print path.....	14
Figure 3.4 : Print-path generators, $G_1 - G_6$, can be utilized to generate print-paths for blocks (a-f) with block-faces having CCs (dark blue) and RCs (light green). Bottom images include unavailable cases to generate print-paths.	17
Figure 3.5 : Four blocks, labeled by 1-4, are shown (a) for a partial bone model. If contact types are not carefully assigned, cross-wise contacts may occur between print-paths labeled by 1 and 2 (a). In case of proper assignment for contact types, cross-wise contacts are eliminated (See pink print-paths near the singular-point in red). .	18
Figure 3.6 : (a, b) Two adjacent blocks with CCs (dark blue) and RCs (light blue). (c, d) Energy E_2 in Equation 3.1 is computed based on compatibility of contact types between a quad-face, f , (in yellow) and adjacent neighboring faces ($q_0, \dots, q_3$) of different blocks. Values for $W(f, q_i)$ are listed below.....	20

Figure 3.7	: (a) A block for a bone model having contact types (dark and light green block-faces have CC and RC, resp.). (b) Generation of sub-layers (in different colors) having distances, $2r$, between them for a hexahedral. (c) A layer (in red) obtained after merging sub-layers, having distances of $2r$ between layers. If the layer is next to a hex-face, this distance is r . (d) Finally obtained layers (in different colors).	27
Figure 3.8	: Transferring block contact types (a) to the block-edges (b). From outer (orange) to inner (pink) sub-layers, all contact-types are assigned (b-d).....	28
Figure 3.9	: Contact types from block-faces are copied to all quads (a). Print-paths are generated using print-path generators ($G_1 - G_6$) based on these contact types (b-c). Step-by-step print-path generation for the regions enclosed by yellow and pink rectangles are shown in (d) and (e), resp.....	29
Figure 3.10	: BCs (yellow cones) directed along normals, $\{n_k\}$, at points, $\{p_k\}$, along print-paths in green (a). Extruder orientations are found by smoothly interpolating cone orientations along the print-path (b).	30
Figure 3.11	: Preceding print-paths (in green and blue) at two consecutive printing time-steps (a and b) for a chamfer model. Yellow cones represent BCs and have different orientations along print-paths.	31
Figure 4.1	: Generated print-paths for various hexahedral mesh models: A double-torus (a), a joint (b), a gear (c) and a chamfer (d). Small images depict block-faces having circular-contacts (dark blue) and curve-contacts (light green).	34
Figure 4.2	: Print paths obtained for a fan disk model (a) after computing (b) CCs (dark blue) and RCs (light green). Some print-paths are colored and shown to better analyze path continuities of print-paths (c). Print-paths labeled by 1 and 3 elongate all along the model, which, in fact, goes through multiple blocks. Print-path labeled by 2 is around a singular-point.	35
Figure 4.3	: Different print-path sets (a-c) obtained using the proposed method for a hanger model. CCs (dark blue) and RCs (light green) are shown in below images.	36
Figure 4.4	: Layers (a-b) obtained for a fan disk and a chamfer model. A defect (depicted with the yellow frame) on the chamfer model (c) affects the layers (b) and print-paths (d).....	38
Figure 4.5	: Given a part under a specified loading condition having forces applied on block-faces (a), maximum principal stress directions are computed (b). Contact types for some block-faces are assigned based on these directions (c), and print-paths are then generated, which are compatible with the directions (d).	39
Figure 4.6	: BC (yellow cone) orientations along the print-paths for the fan disk (a-d) and chamfer (e) models.	41
Figure 4.7	: Manufactured models for a column (a, b), and a chamfer models (c).	43
Figure 4.8	: Manufactured model for a fan disk for the print-paths of Gunpinar [3] (a) and the proposed method (b).	44
Figure 4.9	: A chamfer model (a) is manufactured, and a nozzle orientation (b) is shown during a manufacturing time-step. Part with (c) and without (d) a support structure is depicted.	45

FLUID FLOW-INSPIRED CURVATURE-AWARE PRINT-PATHS FROM HEXAHEDRAL MESHES FOR ADDITIVE MANUFACTURING

SUMMARY

Conventional manufacturing methods create objects by removing material via subtractive processes such as milling, drilling, and carving. In contrast, additive manufacturing (AM) deposits material to fabricate parts, which enables the manufacturing of more sophisticated geometries. Planar slicing methods, where layers are always parallel to the XY plane and on the Z axis, are traditionally used in AM to build 3D models layer by layer. However, quite a few hardware platforms have been developed for enabling the use of multi-axis AM with inputs of three-dimensional curves. These curves can be considered as print-paths and represent parts. This situation can be likened to using curved tool-paths in subtractive manufacturing. Models represented using curves conforms better to the part's geometry than those obtained by planar slicing methods. This also enables improved mechanical strengths in the printed parts.

Thanks to recent improvements, computational methods are able to produce high-quality hexahedral meshes, whose hexahedrons are arranged along the principal curvature directions of the input surfaces. Curvature-aware print-paths, proposed by Gunpinar, follow these curvature directions and have enabled a reduction in stair-stepping effect in the printed parts. However, crosswise-contacts can exist between the print-paths (i.e., print-paths are quasi-perpendicular to each other), which is undesirable as failures may occur particularly at those contact regions. Therefore, the present work aims at generation of curvature-aware print-paths (similar to the print-path generation technique of Gunpinar) without crosswise-contacts between them. To solve this problem, we inspire from fluid flow and imitate (laminar) streamlines for an inlet and an outlet of a duct for designing curvature-aware print-paths. A hexahedral mesh is decomposed into blocks (cuboid-like sub-volumes), each of which is covered with fluid flow-inspired print-paths. A multi-axis (collision-free) additive manufacturing (AM) planning technique is also proposed. As a proof of concept, fluid flow-inspired curvature-aware print-paths are validated using a multi-axis AM simulator and machine.



EKLEMELİ İMALAT İÇİN ALTIYÜZLÜ-ELEMAN AĞLARI İÇİN AKIŞKAN AKIŞINDAN ESİNLENİLEN EĞRİLİĞE DUYARLI BASKI YOLLARI

ÖZET

Eklemeli imalat bilgisayar ortamında tasarlanan geometrik (CAD) modellerin hızlı bir şekilde fiziksel olarak üretilmesini sağlar. Eklemeli imalatta ekstrüderden sıcak malzeme çıkarılıp tezgâha koyularak eklenmesi ile elde edilir. Bu yöntemlerden biri olan kaynaşmış birikme modellemesi (FDM) yöntemi, son yıllarda endüstriyel fabrikalardan evlerimize kadar birçok alanda kullanım alanına sahiptir. Parça dilimlenir ve her bir dilimde baskı yolları oluşturularak katman katman imal edilir. Üç eksenli eklemeli imalat makinalarında dilimleme işlemi katmanların XY düzlemine paralel olacak şekilde yapılır. Z-ekseni boyunca bir düzlem geometrik model ile kesiştirilir ve baskı yolları elde edilir.

Dilimleme işlemi hem model için baskı yolu elde edilmesinde hem de elde edilen baskı yolunun üretim planlaması bakımından zahmetsiz olmasından dolayı günümüzde en yaygın kullanılan yöntemdir. Fakat basit olmasının yanında bazı olumsuzluklara sahiptir. Bunlardan biri geometrik modelin katman katman üretilmesinden kaynaklı modelin yüzeyindeki eğri kısımlarda merdiven etkisi (basamak etkisi) oluşmasıdır. Bu durum modeldeki yüzey kalitesini bozduğu için istenmemektedir. Son yıllarda bu sorunlar üzerine birçok araştırma yapılmış olup, birçok farklı yöntem önerilmiştir. Merdiven etkisi için eğri baskı yolları kullanılıp, katmanlardaki baskı yolları ise mekanik stres yönlerine ve model geometrisine göre ayarlanması sağlanmıştır.

Eğri baskı yollarının oluşturulması literatürde birçok farklı yöntem ile yapılmaktadır. Bu yöntemler geometrik modelin depolandığı dosya biçimine göre farklılık göstermektedir. Bu tezde altı yüzlü ağ yapıları tercih edilmiştir. Son yıllarda üretilen bu ağlara ait elemanlar yüzeylerde tanımlı minimum ve maksimum eğrilik yönlerini takip etmektedir. Bundan dolayı da eklemeli imalatı gerçekleştirilen parçada daha az basamak etkisi görülmektedir. Ayrıca altı yüzlü ağ yapıları sonlu eleman simülasyonlarına uyumlu oldukları için gelecekte de tercih edilebilecekleri umulmaktadır.

Altı yüzlü ağ yapılarını ilk defa Günpınar (2021) kullanmıştır. Fakat imal edilen parçalardaki baskı yolları incelendiğinde baskı yolları birbirine dik olarak değmektedir. Bu da parçanın mukavemetini azaltabileceği düşünülmektedir. Bu tez bu problemi çözebilmek için akışkanlar mekaniğindeki laminar akış çizgilerinden esinlenmiştir. Baskı yollarını akış çizgilerine benzeterek yeni bir yaklaşım ortaya koyulmuştur. Bunun sayesinde baskı yolları arasında dik değme problemi ortadan kaldırılmıştır. Geliştirilen baskı yolları kullanılarak parça çok-eksen eklemeli imalat yöntemi ile imal edilmiştir.

Geliştirilen metot teknik olarak dört aşamayı kapsamaktadır. Öncelikle eğrilik yönlerine uyumlu şekilde üretilmiş bir altı-yüzlü ağ girdi olarak alınır. Bu altı-yüzlü ağın elemanları yeniden gruplandırılarak daha kaba bir ağ yapısı olan blok-ağ elde edilir. Elde edilen bu yeni yapı bir veya birden fazla altı-yüzlü ağ elemanından oluşan blok olup, oluşturulmuş blok-ağın elemanı olarak tanımlıyoruz. Blok, altı yüzü bulunan ve her yüzünde en fazla bir tane (komşu) blok bulundurabilme imkânı olan elemanlardır.

Bloklarımız elde edildikten sonra bir sonraki aşama olan blok içindeki baskı yolu yönlerini belirleyen koşullar belirlendi. Baskı yönü koşulları blokların yüzlerine atanan sınır koşullarıdır. Bu sınır koşulları blok içindeki baskı yollarının belirtilen yüz ile ilişkisi olup, eğer baskı yollarının başlangıç veya bitiş noktaları mevcut yüz üzerinde ise dairesel temas, değil ise çizgisel temas olarak adlandırılmışlardır. Bu temas koşulları blokların komşuları ile olan ilişkisini tanımlamaktadır. Bir blok için olabilecek tüm koşullar belirlenip bir veri kütüphanesi oluşturulmuştur. Bu veriler blok için üretilebilirliklerine göre veya kişisel tercihlere göre filtrelenip, blok operatörleri olarak tanımladığımız baskı yönü verilerini oluşturur. Bu operatörler bloklara atanacak sınır koşulları olup, blok içindeki baskı yollarının blok boyunca yönlerini belirtmektedir. Daha sonra blok-ağ içindeki blokların her birine dik temas oluşturmayacak şekilde operatör ataması için bir algoritma geliştirilmiştir. Bu algoritma içinde kullanılan fonksiyonlar tanımlanmış ve her birinin önemi irdelenmiştir. Geliştirilen algoritma sayesinde tamamen rasgele, asal gerilme yönlerine göre veya kullanıcının belirleyeceği parametrelere göre dik temas oluşmadan operatör ataması yapılabilir.

Üçüncü aşama blokların içinde eğri baskı yollarının oluşturulmasıdır. Eğri baskı yollarının özelliği modelin yüzey eğriliklerini takip edip, yüksek bir yüzey kalitesi elde edilmesini sağlamasıdır. Bu baskı yollarını bloklarımızda, bölgesel dilimleme ile elde ettik. Bloklar içindeki her bir altı-yüzlü elemanı kendi içinde dilimleyerek, bu dilimlerin üzerine baskı yolları oluşturuldu. Oluşturulan bu baskı yolları daha sonra komşu altı-yüzlü elemanda bulunan ortak baskı yolu ile birleştirilerek, blok içindeki asıl baskı yolu elde edilmiş olur. Bu strateji ile blok içindeki baskı yollarını en alt hacimlerden başlayarak, tüm blok hacmi için elde edilmiş olur. Son olarak da blok baskı yolları komşu bloklarda bulunan ortak baskı yolları ile birleşerek tüm ağ yapısı hacmine ait olan baskı yollarını oluşturmuş olur. Buradaki ortak baskı yolu tanımını, blokların ortak yüzlerinde aynı başlangıç veya bitiş noktalarına sahip olan baskı yollarıdır.

Dördüncü aşama elde edilen baskı yollarının çok-eksenli eklemeli imalata hazırlanmasıdır. Eklemeli imalatla baskı yolları üretim yüzeyine dik olarak üretilmektedir, bu durum geleneksel dilimlemeyle üretilen modellerde Z-ekseni olduğundan sabittir, fakat çok eksenli üretimde her bir baskı yolu için tanımlanması gereken bir parametredir. Bu çalışmada bu parametre başlangıç olarak, üçüncü aşamada baskı yollarını elde ettiğimiz dilimlerin yüzey normali olarak atanmıştır. Fakat bu yüzey normalleri her zaman baskı yolunun üretimi için uygun olmayıp makinanın kendisiyle veya daha önceden üretilen baskı yollarıyla çarpışmamasını garanti edecek şekilde düzenlenmesi gerekmektedir. Bu aşamada makina tarafından kullanılacak baskı yolu doğrultuları başlangıç doğrultularını referans alarak belirlenmiş ve bu doğrultulara uygun baskı yolu üretim sırası belirlenmiştir.

Son olarak çalışma boyunca elde edilen yöntemleri doğrulamak için modellerin üretimi çok eksenli bir makinada yapıp, sonuçlar önceki çalışmalar ile karşılaştırılmıştır. Elde edilen çıktılar, önerilen yöntemin dik temas olmadan istenilen sonuçları üretebildiğini ve ilaveten üretilmiş modellerin önceki yöntemlere göre daha iyi yüzey özelliklerini temsil edebildiğini gösterilmiştir.





1. INTRODUCTION

Additive manufacturing (AM), which is a rapid prototyping method, has gained importance in recent years. While it was first used to manufacture the spare part needs of manufacturers (to avoid high shipping fees and extended shipping times), it has evolved to be used even for personal use. People prefer custom products instead of mass production products as a result of increasing globalization. This situation has caused AM to attract more attention and has led to much research on it. AM is the construction of a three-dimensional part from a computer-aided design (CAD) model. The process can be accomplished in a variety of processes. Material is deposited under computer control,, typically layer-by-layer. Fused deposition modeling (FDM) using a continuous filament of thermoplastic material is the most common AM process.

AM allows users to manufacture parts with complicated geometries, and therefore, it is recent. Mechanical properties of printed parts rely on the AM process parameters (such as melting temperature), (input) material used, and slicing techniques. The work in the current dissertation aims at the proposition of a new slicing technique based on hexahedral mesh input. Hexahedral mesh is currently not a common input for AM, but it can have appealing properties in AM. In this section, after motivating the thesis approach, overview and the thesis scope will be outlined.

1.1 Thesis Motivation

In AM, different inputs representing the part to be printed are utilized. STL is still the most commonly used file format in AM, which describes only the part's surface geometry. Planar slices are obtained using STL files to create the layers of a model on horizontal planes. The use of different inputs for AM has been recently demonstrated by researchers. Ezair et al. [1] utilized trivariate volumes to generate (non-planar) print-paths. Additionally, Gunpinar and Armanfar [2] employed boundary representation (B-rep) as input to extract print-paths with helical geometries. Recently, Gunpinar et al. [3] investigated the use of hexahedral meshes in AM and

generated curvature-aware (non-planar) print-paths that conformed better to the part geometry, thereby reducing stair-stepping effect in the printed part.

The work presented in this paper is an extension of curvature-aware print-paths [3]. One can observe *crosswise-contacts* between the curvature-aware print-paths of adjacent *blocks* (i.e., cuboid-like sub-volumes of the hex-mesh). In other words, two print-paths are quasi-perpendicular to each other and have circular contact. This may be undesirable in terms of the mechanical performance of the part under a loading condition. Failures potentially occur in such regions, where print-paths are discontinuous. Fig. 1.1 (b) shows several cross-wise contacts between the print-paths (in green) in a bone hexahedral mesh model (a). To highlight this problem, several print-paths are shown in pink color.

To tackle the cross-wise contact problem described above, *fluid flow-inspired print-paths* (Fig. 1.1 (c)) are proposed, which are inspired from streamlines of (laminar) fluid flow modeled inside a model (c). Print-paths are elongated based on given entries and exits for print-paths (like inlet and outlet in fluid flow). However, manufacturing fluid flow-inspired print-paths using 3-axis AM is nontrivial due to the accessibility of a print-head [4]. Therefore, we additionally propose a multi-axis (collision-free) print-path planning technique devised for fluid flow-inspired print-paths.

1.2 Proposed Approach

Given a hexahedral mesh of a column model (Figure 1.2 (a)), a (unique) base complex [5] of the mesh is obtained (b) having blocks, cuboid-like sub-volumes. Blocks are filled using print-paths, which can have circular or curve contacts with block faces. Therefore, these contacts are carefully assigned without allowing cross-wise contacts. Dark blue and light green block-faces in Figure 1.2 (c) depict, respectively, circular or curve contacts between print-paths and block faces. Layers (in red) are next obtained (d). Print-paths (e, in different colors) are finally generated while respecting these contact types for each block faces. Collision-free print-path planning is then carried out to manufacture these print-paths. Extruder (yellow cone) orientations at several print-path locations are shown (f).

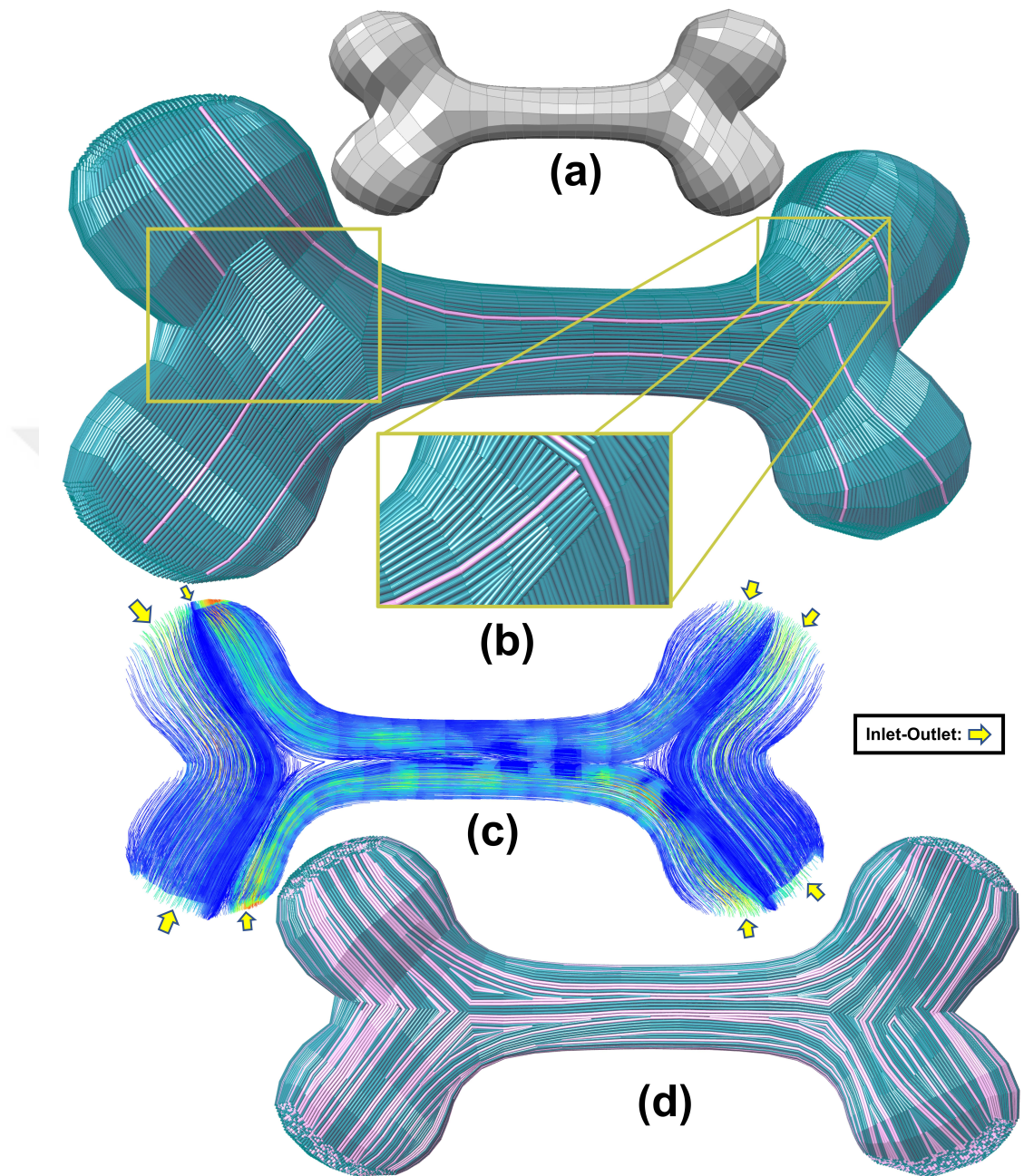


Figure 1.1 : Given a hexahedral mesh model (a), curvature-aware print-paths (in green) [3] for a bone model are generated (b). Observe cross-wise contacts (enclosed by green rectangles) between print-paths. Pink print-paths highlight this problem, which exists in neighboring regions. Fluid flow-inspired print-paths (d) can prevent such cross-wise contacts, which are likened to streamlines of (laminar) fluid flow modeled inside a bone model (c).

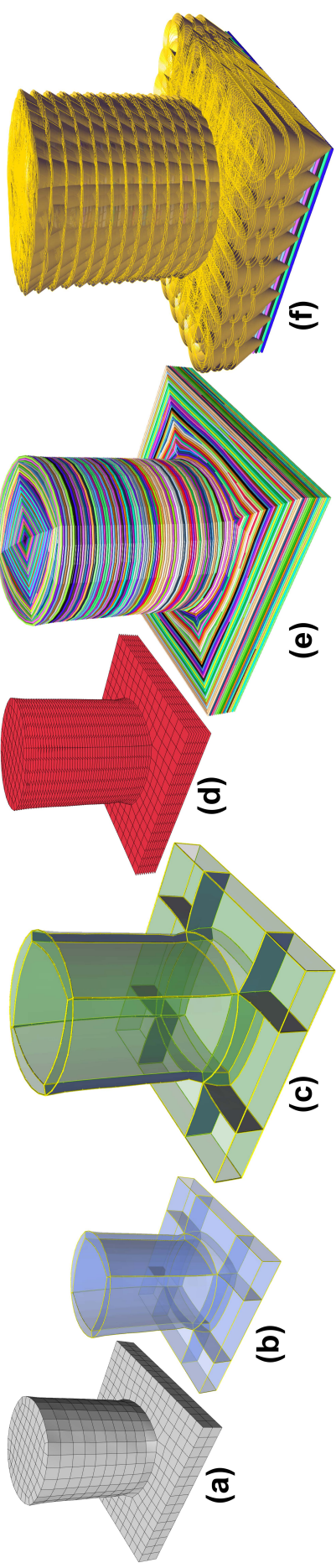


Figure 1.2 : A column hexahedral mesh model (a) is decomposed into blocks (b, cuboid-like sub-volumes). Contact types (circular or curve) between block-faces and print-paths are then determined. Dark blue and light green represent circular or curve contact types, respectively(c). Layers (in red) are then obtained (d). Print-paths are then generated inside blocks while respecting pre-determined contact types (e). Finally, collision-free print-path planning is carried out (yellow cones representing extruder in (f)).

1.3 Scope

This thesis involves print-path generation methods for hexahedral meshes and print-path planning for AM. The simplified workflow is shown in Figure 1.3. Using the proposed approach, the user can use any hexahedral mesh. Print-paths are then generated based on the user desires such as aligning print-paths along principal stress vectors. Alternatively, the proposed method is able to randomly generate print-paths. In all cases, the print-paths are aligned with the minimum and maximum curvature lines in the model. Finally, print-paths are sorted and collision-free multi-axis AM planning is achieved.

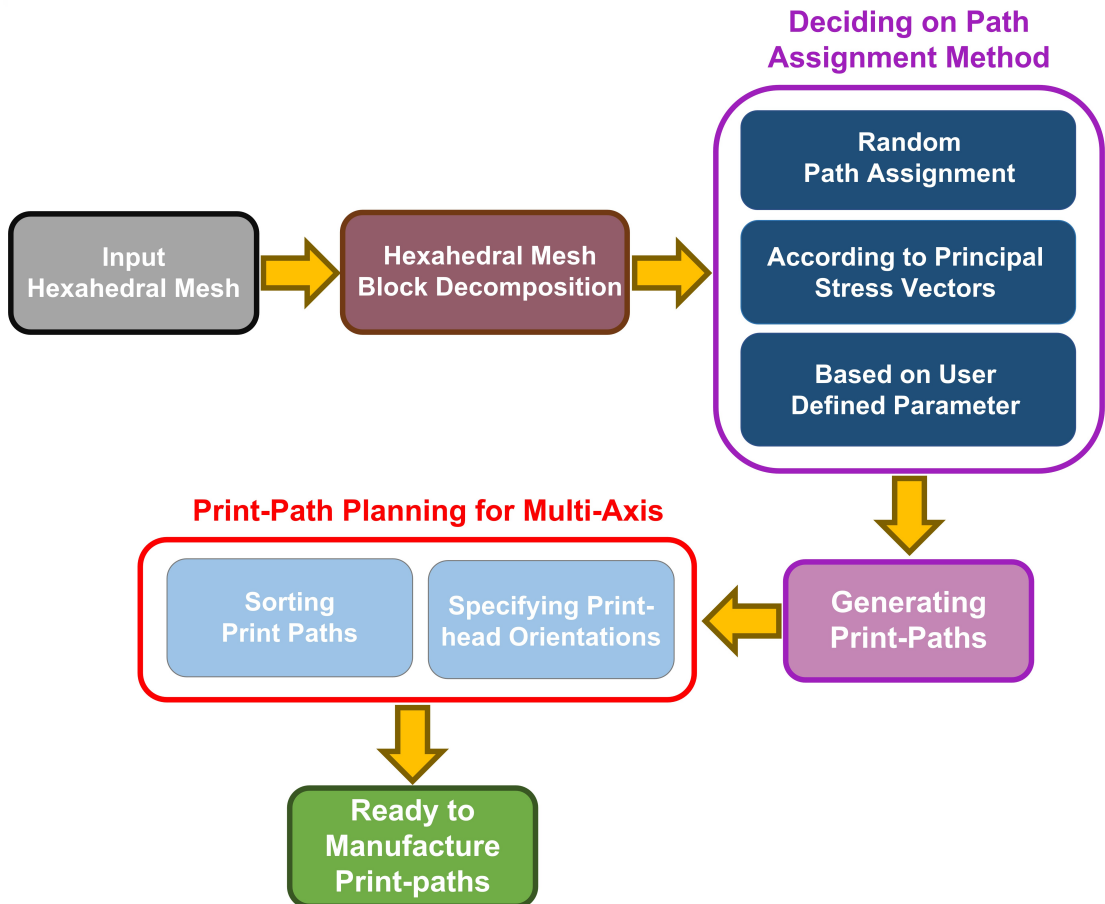


Figure 1.3 : Work flow for the proposed method in this dissertation.

1.4 Outline

The thesis is organized as follows. Section 2 reviews the relevant prior art. In Section 3, details of the proposed method will be explained. Section 3.2 describes a methodology to generate fluid flow-inspired print-paths from a hexahedral. Additionally, Section 3.3 describes a collision-free print-path planning method for multi-axis AM. Section 4 discusses this study's experiments. Finally, in Section 5, we conclude.



2. RELATED WORKS

The work reported here is related to hexahedral mesh and AM print-path generation. Sub-sections below cover some selected previous works about these topics.

2.1 Hexahedral Mesh Generation

There are thorough overviews of hexahedral meshing techniques in [6–9]. Nieser et al. [10] created a hexahedral mesh from a parametrization based on a frame-field. Li et al. [11] created a boundary-aligned frame field to build a hexahedral mesh. Jiang et al. Kowalski et al. [12] produced block structures that were combined using a mapping technique. The orthogonal direction field estimated by Solomon et al. [13] can be used to influence the placement of hexahedral edges and mesh singularities. A method for creating octahedral fields with specified hex-meshable singularity graphs that produced a valid and less distorted hexahedral mesh was developed by Liu et al. [14]. Takayama [15] presented an interactive method, in which the user created a collection of surfaces intersecting one another. This collection was then employed to create a hexahedral mesh. An approach for producing feature-aware hexahedral meshes with necessary interior singularities was proposed by Guo et al. [16]. Furthermore, Livesu et al. [17] provided an automated hexahedral meshing technique for producing high-quality meshes, whose elements were aligned with a cross-field specified on the surface as well as feature curves on the surface. Palmer et al. [18] presented an algebraic representation for octahedral fields and optimized these fields while preserving the singular structure of a hexahedral mesh. Livesu et al. [19] obtained a grid-based hexahedral mesh by using topological schemes to regularize the valence of inner vertices and edges using a mesh dualization tool. Gao et al. [5, 20] extracted a base-complex of a hexahedral mesh while repairing misalignments of singularities. Cherchi et al. [21] improved the polycube-based meshing pipeline by remapping the positions of the polycube corners to produce both surface and volumetric coarse block-structured meshes of general shapes. Edges of most of the above hexahedral meshes are aligned with surface curvature and feature-lines, and therefore, the present

work can take them as input for the generation of fluid flow-inspired curvature-aware print paths.

2.2 Generation of Non-planar AM Print-paths

Non-planar slicing methods in AM have been popular in recent years. Curved layers for FDM were initially developed by Chakraborty et al. [22], where the filaments were deposited along the curved routes. Alkadi et al. [23] proposed a conformal layering method for material extrusion printing procedure constructing a model on a freeform surface. To lessen the requirement for support structures and the number of layers, several works [24–26] introduced non-planar slicing approaches. To enhance the surface finish quality of the printed parts, Isa and Lazoglu [27] developed a five-axis FDM technique based on a nonplanar slicing using transition layers for a model. Llewellyn-Jones [28] generated print-paths for concave/convex geometries utilizing curved layer fused filament fabrication and demonstrated that the surface quality was improved compared to planar printing. Chen et al. [29] used nonplanar sliced layers in FDM with varied thicknesses for thin-shell components.

To minimize support structures and staircase faults, Etienne et al. [30] proposed a slightly curved slicing method for a 3-axis FDM. Additionally, Xu et al. [31] generated print-paths from a (closed) triangular mesh by breaking it down into curved layers. To overcome the anisotropic mechanical properties of parts manufactured using AM, Fang et al [32] used principal stress lines for generating curved slicing layers, on which print paths are then generated. Chen et al. [33] generated print paths for principal stress lines and manufactured them using carbon fiber material. Bi et al. [34] presented a design technique for print-paths that are manufactured using multi-axis AM. Xie et al. [35] proposed an ellipsoidal slicing while considering support-free and gouging-free for multi-axis AM. Li et al. [36] additionally considered mechanical properties of a part to be printed while generating print-paths using a vector field-based curved layer.

Contrary to the above works, Ezair et al. [1] proposed a method for generating volumetric covering print-paths for AM from trivariate volumes. Despite the fact that using hexahedral meshes for AM is not a common practice, Gunpinar [3] has been utilized them as AM input to generate curvature-aware print-paths, which resulted in reduction of stair-stepping effect in the AM parts. The present work in this paper

similarly utilizes also a hexahedral mesh representing a part to be printed and extends the work by Gunpinar [3] to eliminate cross-wise contacts between these print-paths.





3. METHODS

3.1 Terminology

A hexahedral mesh, $M = (V, E, Q, H)$ decomposes a volumetric region into hexahedrons, H , and is constructed by a set of vertices, V , edges, E and quadrilateral faces (quad-faces), Q . The valence of an edge is the number of its incident quad-faces, and an edge is *regular* if its valence is two (on the boundary) or four (in the interior). Otherwise, it is irregular. A *singular-edge* is a set of connected, same-valence irregular edges.

Definition 3.1.1 A block, b , is a structured sub-volume of $\mathcal{H}$, in which no singular-edges are contained. More precisely, b consists of a sorted set of vertices: $b = \{v[i, j, k], i = 0, 1, \dots, n_M - 1, j = 0, 1, \dots, n_P - 1, k = 0, 1, \dots, n_R - 1\}$, where $v[i, j, k] \in V$, and n_M, n_P and n_R are the number of vertices in three (block) dimensions.

Fig. 3.1 shows a block containing twelve edges (in yellow), eight corner points (in red), six faces and 490 hexahedrons. $v[i, j, k]$ are colored in yellow.

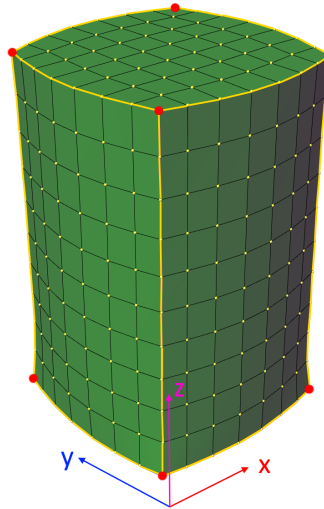


Figure 3.1 : A block, b , (in green) with twelve edges (in yellow), eight corner points (in red) and six faces.

Gao et al.'s technique [5] decomposes M into a new coarse mesh, which we call a *block-mesh*, $\mathcal{B} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$. $\mathcal{B}$ consists of a set of block-vertices, $\mathcal{V}$, block-edges, $\mathcal{E}$, block-faces, $\mathcal{F}$, and block, $\mathcal{H}$, as described below:

- A block-edge, e , has two block-vertices (i.e., $e \in \{\mathcal{V} \times \mathcal{V}\}$).
- A block-face, f , is formed by four block-edges (i.e., $f \in \{\mathcal{E} \times \dots \times \mathcal{E}\}$). f contains regular (two-dimensional) arrangement of quadrilateral faces in M .
- A block, b , consists of six block-faces (i.e., $b \in \{\mathcal{F} \times \dots \times \mathcal{F}\}$).

We support the following block-operators in $\mathcal{B}$:

- $f.b$ returns block(s) incident to a block-face, f . The number of blocks is one if f is a boundary block-face. Otherwise, it is two.
- $f.opp(b)$ returns opposite block-face of f in a block, b .
- $b.f$ returns block-faces in b (i.e., $b.f = [f_0, \dots, f_5]$). Here, $f_{i+3} = f_i.opp(b)$, where $i = 0, 1, 2$.
- $f.adj(b)$ returns four neighboring block-faces of f in b .
- $f.adj2(b)$ returns neighboring block-faces of f , which are not block-faces of a block, $b' \in f.b$ (i.e., $f' \in f.adj2(b) \rightarrow f' \notin b'.f$).
- $b.b$ returns neighboring blocks to b .

3.2 Generating Fluid Flow-inspired Print-paths from a Hexahedral Mesh

This section first introduces the concept of *fluid flow-inspired print-paths* (Section 3.2.1). A methodology for the generation of these print-paths in a block is then explained (Sections 3.2.2 and 3.2.4). A print-path is geometrically defined as follows:

Definition 3.2.1 A *print-path*, S , is a solid model that is obtained by sweeping a circle with a radius, r , along a curve, C .

3.2.1 Fluid flow-inspired print-paths

We inspire from flow behaviour of fluids while generating print-paths in a block, and therefore, introduce a novel terminology to relate print-paths with streamlines in fluid mechanics. A laminar flow is characterized by fluid particles following flow trajectories without mixing up. Streamlines of laminar flow in a simple duct/block can be considered as the print-paths covering geometry of a duct. For volumetric coverage of the blocks, we first divide blocks into conformal layers. Print-paths are then generated on these layers by imitating these two-dimensional streamlines. Fig. 3.3 shows laminar flow streamlines (top, in different colors) and inspired print-paths (bottom, blue lines). In order to guarantee complete volumetric coverage by print-paths in the hexahedral mesh model, the print paths are not smooth like streamlines.

Definition 3.2.2 A block-face, f , in B and a print-path, S , have a circular-contact (CC) if their intersection is circle. Additionally, curve-contact (RC) can occur if the intersection is line.

In Figure 3.3, 0 (Inlet - Outlet) corresponds to CC type and 1 (Wall, black lines) corresponds to RC type. Fig. 3.2 shows the print-paths and the contact types in a block. Blue circles depict CCs, and red lines show CCs between block-faces and print paths.

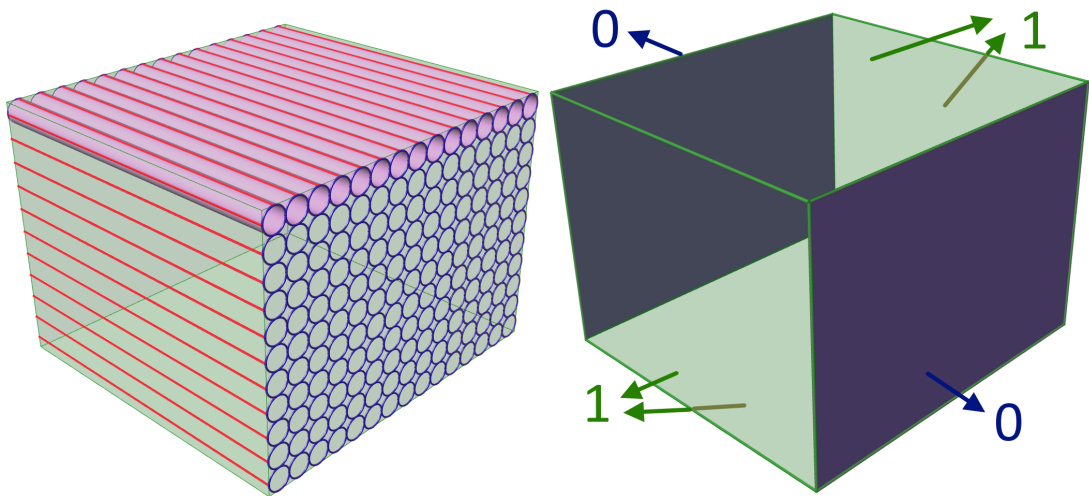


Figure 3.2 : A block (on the right) having circular (0, blue circles) and line (1, red lines) contact types between print-paths and block-faces can be covered using print-paths (on the left).

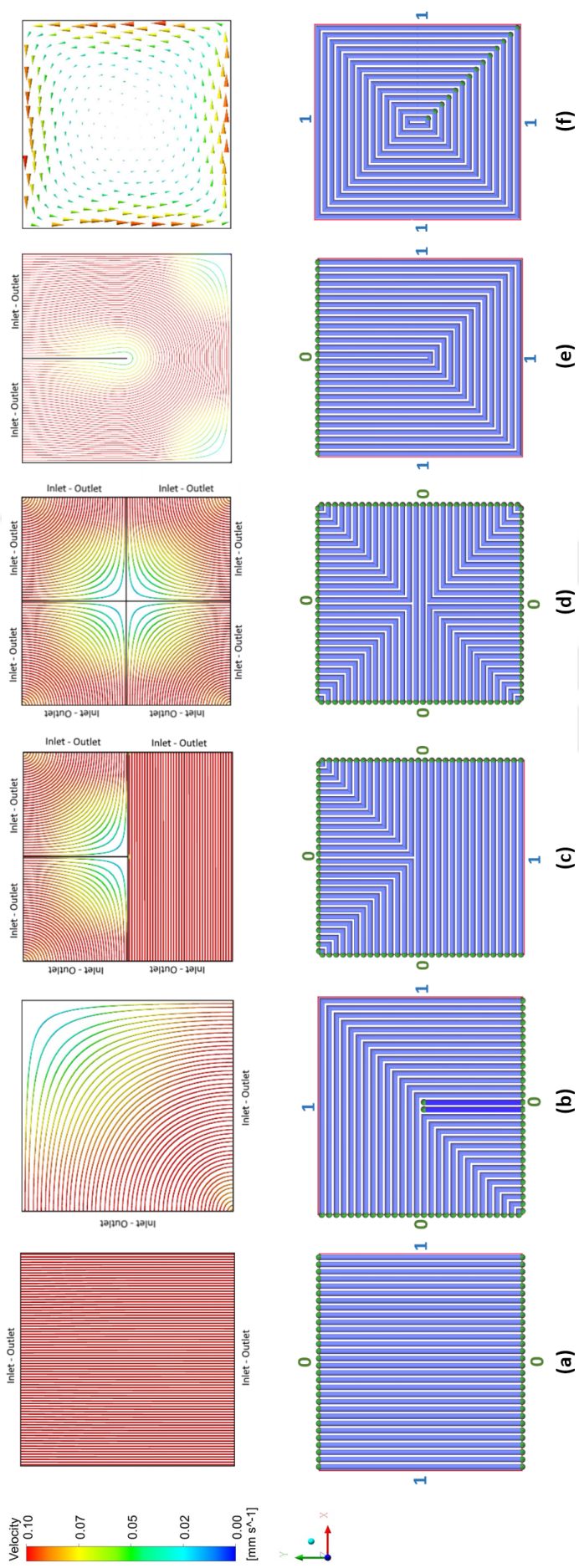


Figure 3.3 : Two-dimensional laminar streamlines (top row) and their corresponding print-paths (bottom row). G1,G2,G3,G4,G5,G6 partial print paths for contact type assigned sub-layers, respectively (a,b,c,d,e,f). Green dots show to start or end points for sub-layer print paths, the dark blue line shows the trimmed print path.

Here, we introduce a data structure, $\mathcal{C} \in \{c_0 \times \dots \times c_5\}$, which contains contact types between block-faces, $b.f = [f_0, \dots, f_5]$, and a set of print-paths, $\{S_j\}$. Recall that $c_0 - c_5$ can either be 0 (for CCs) or 1 (for RCs), and $\mathcal{C}$ involves 2^6 different value-sets. These sets are classified according to contacts between block-faces and print-paths, which are summarized below:

1. **CCs between one block-face and print-paths:** Only one of block-faces in b has circular-contacts with print-paths. 6 different value-sets exist: $\mathcal{C} = [0, 1, 1, 1, 1, 1] \vee [1, 0, 1, 1, 1, 1] \vee [1, 1, 0, 1, 1, 1] \vee [1, 1, 1, 0, 1, 1] \vee [1, 1, 1, 1, 0, 1] \vee [1, 1, 1, 1, 1, 0]$ ¹.
2. **CCs between two block-faces and print-paths - 1:** Only a block-face, f , and $f.\text{opp}(b)$ have CCs with print-paths. Here, 3 distinct value-sets exist: $\mathcal{C} = [0, 1, 1, 0, 1, 1] \vee [1, 0, 1, 1, 0, 1] \vee [1, 1, 0, 1, 1, 0]$.
3. **CCs between two block-faces and print-paths - 2:** f and a block-face, $g \in f.\text{adj}(b)$, have CCs with print-paths. 12 different value-sets exist: $\mathcal{C} = [0, 0, 1, 1, 1, 1] \vee [0, 1, 0, 1, 1, 1] \vee [0, 1, 1, 1, 0, 1] \vee [0, 1, 1, 1, 1, 0] \vee [1, 0, 0, 1, 1, 1] \vee \dots \vee [1, 1, 1, 0, 1, 0]$.
4. **CCs between three block-faces and print-paths - 1:** f , $f.\text{opp}(b)$ and g have CCs with print-paths. 12 different value-sets exist: $\mathcal{C} = [0, 0, 1, 0, 1, 1] \vee [0, 1, 0, 0, 1, 1] \vee [0, 1, 1, 0, 0, 1] \vee [0, 1, 1, 0, 1, 0] \vee [0, 0, 1, 1, 0, 1] \vee \dots \vee [0, 1, 0, 1, 1, 0]$.
5. **CCs between three block-faces and print-paths - 2:** f , g and a block-face, h , where $h \in f.\text{adj}(b)$ and $h \in g.\text{adj}(b)$, have CCs with print-paths, where g and h are different block-faces in b ($g \neq h$). 8 different value-sets exist: $\mathcal{C} = [0, 0, 0, 1, 1, 1] \vee [1, 0, 0, 0, 1, 1] \vee [1, 1, 0, 0, 0, 1] \vee [1, 1, 1, 0, 0, 0] \vee [0, 0, 1, 1, 1, 0] \vee \dots \vee [0, 1, 0, 1, 0, 1]$.
6. **CCs between four block-faces and print-paths - 1:** f , $f.\text{opp}(b)$, g and $g.\text{opp}(b)$ have CCs with print-paths. 3 different value-sets exist: $\mathcal{C} = [0, 0, 1, 0, 0, 1] \vee [1, 0, 0, 1, 0, 0] \vee [0, 1, 0, 0, 1, 0]$.
7. **CCs between four block-faces and print-paths - 2:** f , g , h and a block-face, $k \in f.\text{adj}(b)$, have CCs with print-paths, where g , h and k are different block-faces

¹ $\vee$ denotes logical or.

in b ($g \neq h \neq k$). 12 different value-sets exist: $\mathcal{C} = [0, 0, 0, 1, 0, 1] \vee [1, 0, 0, 0, 0, 1] \vee [1, 0, 1, 0, 0, 0] \vee [0, 0, 1, 1, 0, 0] \vee [0, 0, 0, 0, 1, 1] \vee \dots \vee [0, 0, 0, 1, 1, 0]$.

8. **CCs between five block-faces and print-paths:** f and block-faces, $f.\text{adj}(b)$, have CCs with print-paths. 6 different value-sets exist: $\mathcal{C} = [0, 0, 0, 0, 0, 1] \vee [0, 0, 0, 0, 1, 0] \vee [0, 0, 0, 1, 0, 0] \vee [0, 0, 1, 0, 0, 0] \vee [0, 1, 0, 0, 0, 0] \vee [1, 0, 0, 0, 0, 0]$.
9. **CCs between six block-faces and print-paths:** All block-faces have CCs with print-paths. A single value-set exists: $\mathcal{C} = [0, 0, 0, 0, 0, 0]$.
10. **RCs between six block-faces and print-paths:** All block-faces have RCs with print-paths. A single value-set exists: $\mathcal{C} = [1, 1, 1, 1, 1, 1]$.

The contact types labeled by 2, 3, 4, 6, 1 and 10 can be utilized in AM, others are infeasible to obtain fluid flow-inspired print-paths (Refer Section 3.2.4 for the generation of block layers from contact types). For the feasible contact types, we, respectively, define print-path generators, $G_1, G_2, G_3, G_4, G_5, G_6$, which generates a set of print-paths, $\{S_j\}$, in b , from $\mathcal{C}$. Top images in Figure 3.4 (a-f) provides an illustration for these cases. Here, dark blue and light green faces indicate CCs and RCs, resp. Obtained print-paths using G_3 (See also Figure 3.3 (c)) are shown in (c). Bottom images in Figure 3.4 involve some infeasible contact types in a block.

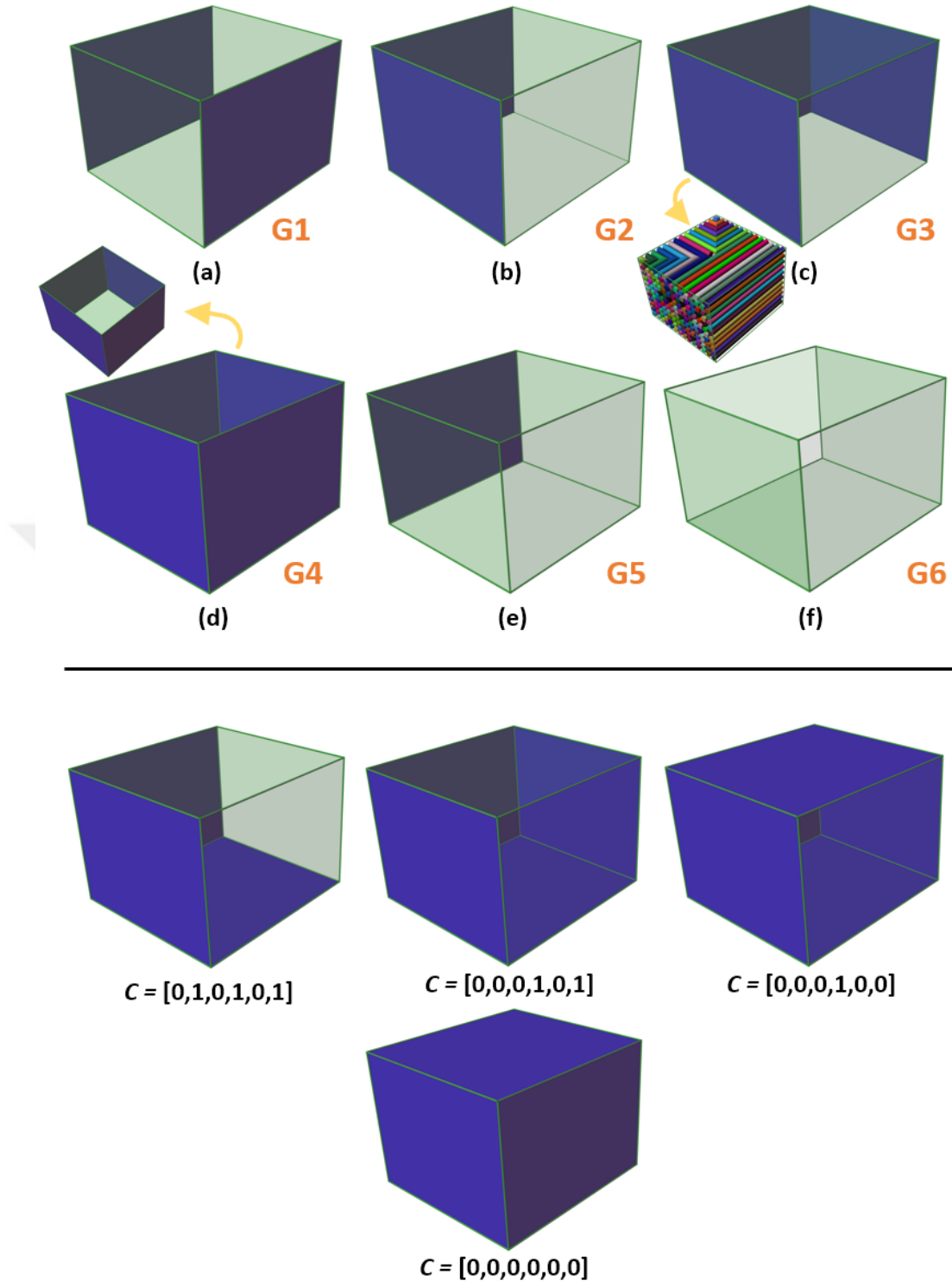


Figure 3.4 : Print-path generators, $G_1 - G_6$, can be utilized to generate print-paths for blocks (a-f) with block-faces having CCs (dark blue) and RCs (light green). Bottom images include unavailable cases to generate print-paths.

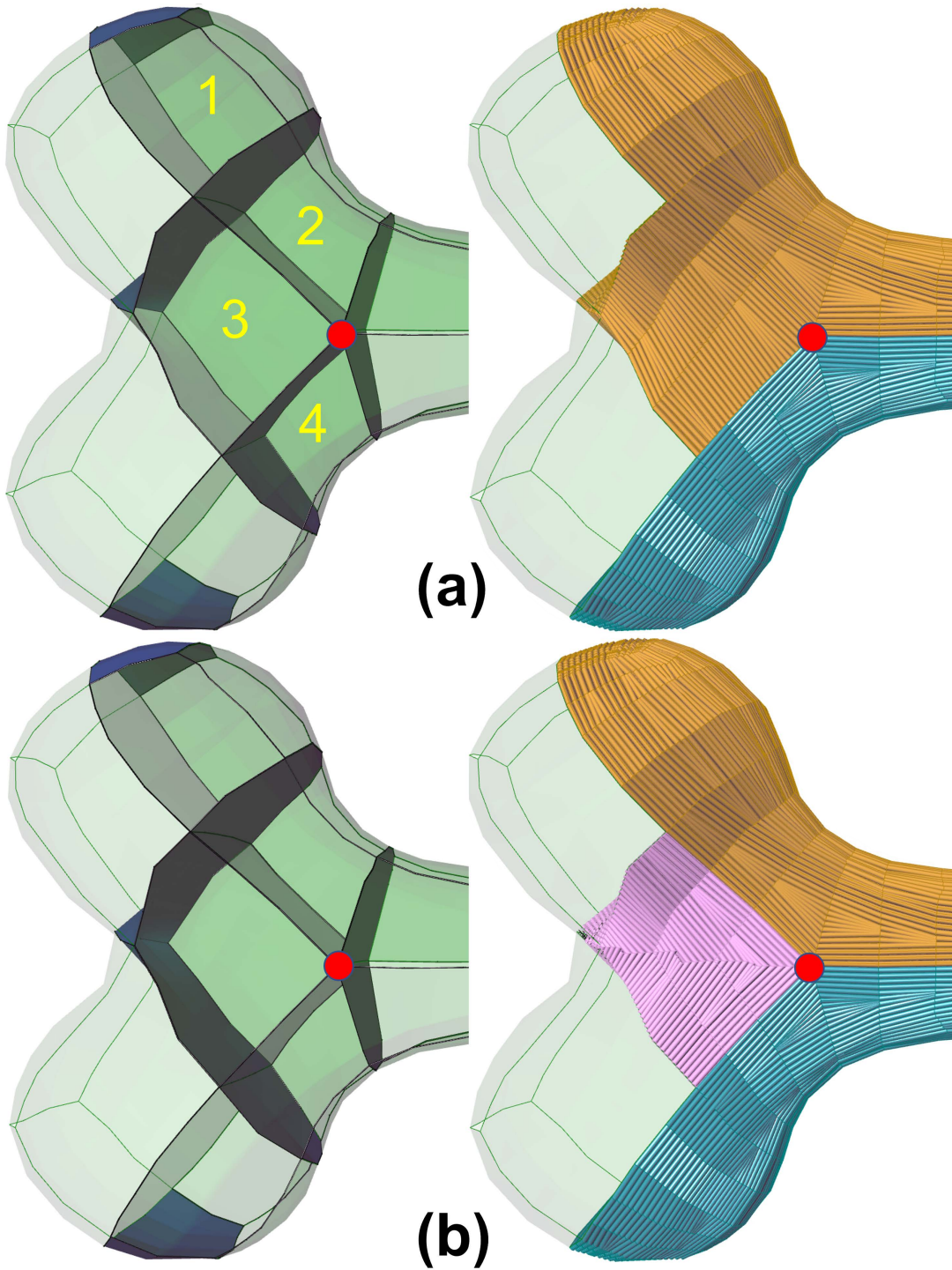


Figure 3.5 : Four blocks, labeled by 1-4, are shown (a) for a partial bone model. If contact types are not carefully assigned, cross-wise contacts may occur between print-paths labeled by 1 and 2 (a). In case of proper assignment for contact types, cross-wise contacts are eliminated (See pink print-paths near the singular-point in red).

3.2.2 Assigning contact types for a block

Contact types in a block, b , are assigned according to generators, $\{G_i\}$, and compliance of contact types of b with those of blocks adjacent to b . Therefore, we introduce two energy functions, E_1 and E_2 , for b , which are described as follows:

- E_1 is a heuristic-based function defined on generators, $\{G_i\}$, as follows: $E_1(G_1) < E_1(G_2) < E_1(G_3) < E_1(G_4) < E_1(G_5) < E_1(G_6)$.

(1) G_1 is the simplest operator among others. Print-paths obtained using G_1 simply elongates from one block-face to its opposite block-face. Such print-paths stop at a block-face and can continue their elongation from its neighboring block. Fig. 3.5 depicts a partial bone model. Notice how print-paths in block labeled by 1 continue in another block labeled by 2 (a). Here, dark block-faces (on the left images) have CCs, while others have RCs with print-paths.

(2) However, utilizing only G_1 may result in cross-wise contacts (See orange and cyan print-paths of blocks labeled, respectively, by 3 and 4). This may happen between print-paths of neighboring blocks incident to singular-edges (red dot). In such situations, the simplest generator is G_2 , which generates pink print-paths in a block labeled by 3 (b). Therefore, $E_1(G_2) < E_1(G_i)$, where $i = 3, \dots, 6$.

(3) E_1 for $G_3 - G_6$ can be defined based on a user according to application. We arrange energy values according to their simplicity: $E_1(G_3) < E_1(G_4) < E_1(G_5) < E_1(G_6)$. While one can also equalize these energy values: $E_1(G_3) = E_1(G_4) = E_1(G_5) = E_1(G_6)$.

- Print-paths of neighboring blocks elongate in the same direction are desirable to have print-path continuity, and thereby reinforcing each other. To favor such print-paths, E_2 for a block, b , is designed as follows:

$$E_2 = \sum_{f \in b.f} \sum_{f_2 \in f.adj2(b)} W(f.c, f_2.c), \quad (3.1)$$

where

$$W(q, r) = \begin{cases} 0, & \text{if } q = -1 \vee r = -1 \\ |q - r|, & \text{otherwise.} \end{cases} \quad (3.2)$$

Here, $f.c$ is an operator, which returns an integer value, 0 or 1, representing the contact types, CCs and RCs, resp. Additionally, $|x|$ symbolizes absolute value of an integer number, x .

Fig. 3.6 (a and b) illustrates two blocks with CCs (dark blue) and RCs (light green). Let f (in yellow, (c-d)) be a quad-face of the left block with RCs (i.e., $c = 0$). And $\{q_i, i = 0, \dots, 3\}$ represent adjacent blocks to f . Assume contact types of block-faces in (a) and (b) are $[-1, -1, 1, -1]$ and $[-1, -1, 0, -1]$, resp. $W(f, q_i)$ is 0 in the former case, while it is 1 for q_2 . This means that former case is preferable as E_2 in Equation 3.1 can potentially be minimized for that case. The latter case is not desirable since print-paths in adjacent blocks do not elongate in the same direction (i.e., quasi-perpendicular).

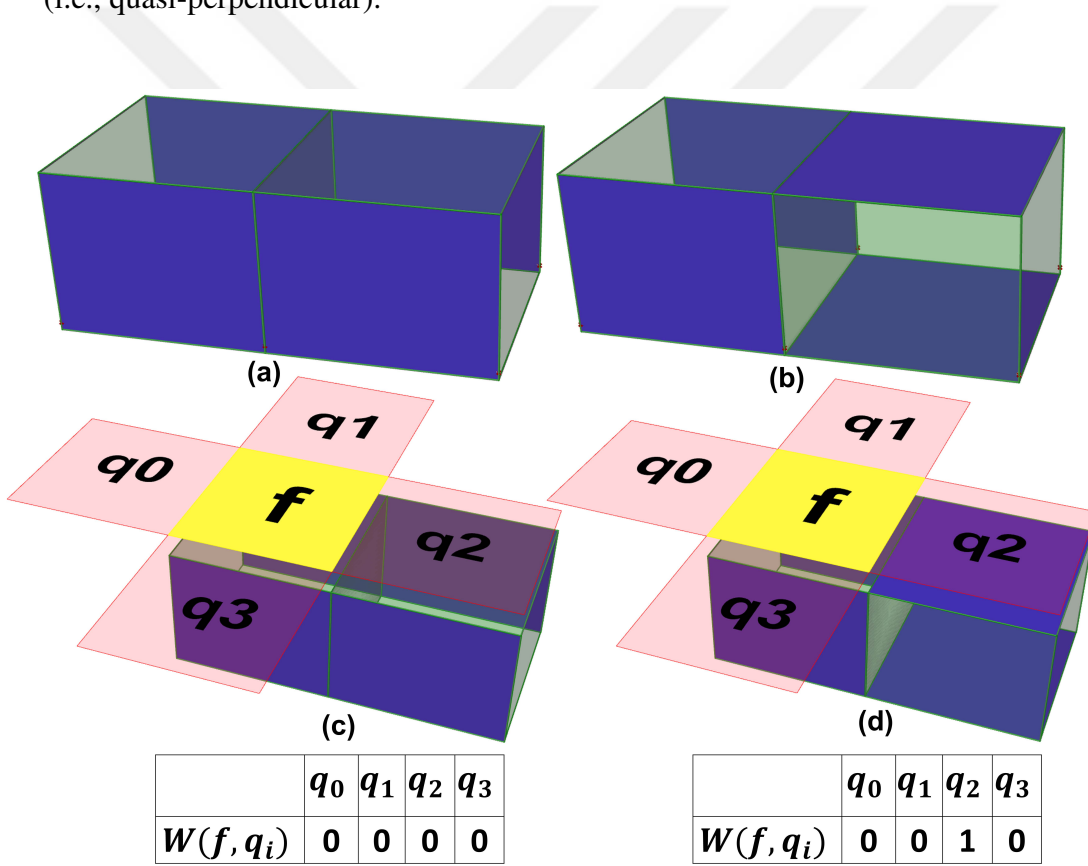


Figure 3.6 : (a, b) Two adjacent blocks with CCs (dark blue) and RCs (light blue). (c, d) Energy E_2 in Equation 3.1 is computed based on compatibility of contact types between a quad-face, f , (in yellow) and adjacent neighboring faces ($q_0, \dots, q_3$) of different blocks. Values for $W(f, q_i)$ are listed below.

Here, we define a heuristic for block selection before utilizing the energy functions E_1 and E_2 . Utilizing solely these energy functions may result in multiple options for block

selections. Therefore, we introduce two new energy functions, F_1 and F_2 . Additional block operators are supported to describe these energy functions:

- $b.c$ returns the number of contact types (already) assigned for the block-faces, $b.f$, of a block, b . $b.c$ takes a value between 0 and 6.
- $b.e$ returns the block-edges of b .
- $e.v$ returns the valence of a block-edge, e .

F_1 and F_2 are assigned for a block, b . $F_1 = b.c$ and F_2 is computed as follows:

$$F_2 = \begin{cases} \operatorname{argmax}_{e \in b.e} e.v, & \text{if } e \text{ is a singular-edge} \\ 0, & \text{otherwise.} \end{cases} \quad (3.3)$$

Blocks having larger values of F_1 should be selected first to avoid the usage of undesirable print-path generators, such as G_4 . Recall that G_1 is preferable compared to others (i.e., $E_1(G_1) < E_1(G_4)$). Therefore, contact types for such blocks (having large F_1 values) are assigned first.

F_2 favors the selection of blocks incident to singular-edges. Hexahedral mesh elements incident to singular-edges exhibit irregular arrangements, which can disallow usage of simpler (i.e., lower cost) print-path generators (such as G_1). Thus, blocks incident to high-valence irregular edges are selected first to overcome such difficulties. The number of already assigned contact types for a block is critical, and therefore, higher priority can be given for F_1 compared to F_2 . In case, values of F_1 and F_2 for multiple blocks are same, one of them can be randomly chosen.

Pseudo-codes for the proposed contact type assignment for a set of given blocks is summarized in Algorithm 1 calling Algorithm 2.

- Given a set of blocks, the function SelectBlocks in Lines 2 and 9 of Algorithm 1 returns a subset of these blocks having maximum F_1 values (Line 4). If there are multiple blocks having maximum and same F_1 values, blocks having maximum F_2 values are chosen. In the algorithm, $b.F_1$ and $b.F_2$ return, respectively, values of F_1 and F_2 , of a block, b .

- Given a block, b , and contact types for the block-faces, contact types for b and energies, E_1 and E_2 , are determined using Algorithm 2 (Lines 8 of Algorithm 1). As $E_1(G_1) < E_1(G_2) < E_1(G_3) < E_1(G_4) < E_1(G_5) < E_1(G_6)$, a generator, G_i , having lower value for i is chosen. The function, GetContactTypes, lists up a set of possible contact types, Δ , for G_i , b and a set of already assigned contact types, $\{c_j\}$. After that E_2 is computed for each element of Δ .

Algorithm 1 first chooses a random starting block in the block-mesh, $\mathcal{B}$, and insert in Σ_2 . As long as Σ_2 has the element (unassigned block-face), the SelectBlocks function chooses a set of blocks, Σ , inside the Σ_2 (Line 6). One of the blocks (b) among Σ is then randomly chosen(Line 7). Possible contact types for b are then selected (Line 8) and these contact types using the conditions in Lines 9 to 17, are inserted Ψ . In case multiple contact type solution (same E_1 and E_2) one of them randomly selected (Line 18), and assigned for block b . Unassigned neighboring blocks for the block, b , are again inserted in Σ_2 (Line 20), and b is removed from Σ_2 (Line 21). Algorithm 1 terminates when all contact types for the block-faces are assigned and returns $\{c_j\}$.

Algorithm 1 Assignment of contact types, $\{c_j\}$, for blocks, $\{B_i\}$

Input:

1. A set of blocks, $\{B_i\}$, in the block-mesh, $\mathcal{B}$;

Output:

1. Contact types, $\{c_j\}$, for the block-faces of $\mathcal{B}$;

Algorithm:

```
1: Initialize  $\{c_j\} := [-1, \dots, -1]$ ;
2: A block,  $b := \text{RandomizeBlock}(\{B_i\})$ ;
3:  $\Sigma_2 \leftarrow \emptyset$  and  $\Sigma_2 := \text{Insert}(b)$ ;
4: while  $\Sigma_2 \neq \emptyset$  do
5:    $E_2^{\min} \leftarrow 9E^9$ ;
6:    $\Sigma := \text{SelectBlocks}(\Sigma_2)$ ;
7:   A block,  $b := \text{RandomizeBlock}(\Sigma)$ ;
8:    $\Delta, E_1, E_2 := \text{SelectCTypes}(b, \{c_j\})$  /*Refer Alg.2*/;
9:   for all  $c \in \Delta$  do
10:    if  $(E_2 < E_2^{\min})$  then
11:       $E_2^{\min} \leftarrow E_2$ ;
12:       $\Psi \leftarrow \emptyset$  and  $\Psi := \text{Insert}(c)$ ;
13:    end if
14:    if  $E_2 = E_2^{\min}$  then
15:       $\Psi := \text{Insert}(c)$ ;
16:    end if
17:  end for
18:   $c := \text{RandomizeCTypes}(\Psi)$ ;
19:   $\{c_j\} := \text{UpdateCTypes}(c)$ ;
20:   $\Sigma_2 := \text{Insert}(b.b)$ ;
21:   $\Sigma_2 := \Sigma_2 \setminus b$ 
22: end while
23: Return  $\{c_j\}$ ;
```

Algorithm 2 SelectCTypes function in Algorithm 1

Input:

1. A block, b ;
2. Contact types, $\{c_j\}$, for the block-faces;

Output:

1. Contact types sets, Δ , for b ;
2. Energies, E_1 and E_2 , for c of b ;

Algorithm:

- 1: **for** $i = 1$ to 6 /* For the contact type cases, whose print-paths are generated via $G_1 - G_6$. */ **do**
 - 2: $\Delta, E_1 := \text{GetContactTypes}(i, b, \{c_j\})$;
 - 3: **if** $\Delta \neq \emptyset$ **then**
 - 4: **break**;
 - 5: **end if**
 - 6: **end for**
 - 7: $E_2 := \text{ComputeEnergy}(b, \Delta, \{c_j\})$;
 - 8: **Return** Δ, E_2 ;
-

3.2.3 A part representing a block-mesh under a loading condition

The proposed contact type assignment method can simply be customized for a block-mesh, $\mathcal{B}$, where orthogonal forces are applied to its block-faces, $\{f_k\}$. We initially set $f_k.c = 0$ for $\forall f_k$ and simply execute the proposed algorithms. Other possible and (potentially) better approach here is to elongate print-paths in the maximum principal stress directions. Authors in [32,37] have proven that parts having such print-paths were stronger. Here, we explain an approach to align print-paths in this way. Let $f.q$ return quadrilateral faces belonging to a block-face, f , of $\mathcal{B}$. A stress vector, ϑ_q , is assigned to a quadrilateral face, $q \in f.q$ if $\|\vartheta_q\| > \tau_1$, where τ_1 is a user-defined threshold to discard small lengths, $\|\vartheta_q\|$, for ϑ_q . Otherwise, no stress vector is assigned to q (i.e., $\vartheta_q = \emptyset$). α_q denotes an angle between the (average) normal vector for q and ϑ_q if ϑ_q is defined. And α is the weighted average of the angles, $\{\alpha_q\}$, for the quadrilateral faces of $f.q$, whose $\vartheta_q \neq \emptyset$.

$$\alpha = \frac{\sum_{q \in f.q, \vartheta_q \neq \emptyset} \alpha_q \|\vartheta_q\|}{\sum_{q \in f.q, \vartheta_q \neq \emptyset} \|\vartheta_q\|} \quad (3.4)$$

where,

$$f_c = \begin{cases} 0, & \text{if } 0 < \alpha < 90 - \tau_2 \\ 1, & \text{if } \tau_2 < \alpha < 90 \\ -1, & \text{otherwise.} \end{cases} \quad (3.5)$$

Contact type assignment is performed for all block-faces. In case of small vectors whose lengths are smaller than τ_2 , no contact type is assigned for a block-face. We then execute the proposed contact type assignment algorithms given an initial set of contact types for the block-faces.

3.2.4 Generating layers and print-paths in a block

A layer is defined as follows:

Definition 3.2.3 A layer, L , of a block, b , is a two-dimensional array of quads/sub-layers, $\{q[i, j]\}$, formed by quad-points, $\{p[i, j]\}$.

More precisely, a quad $q[i, j]$ is formed by a set of quad-points, $p[i, j], p[i + 1, j], p[i + 1, j + 1], p[i, j + 1]$, arranged in a counter-clockwise direction.

The material extruded in AM is assumed to have a circular cross-section so that target layer height and width are equal to each other ($2r$). A set of quasi-parallel (non-intersecting) layers are obtained from b , which are basis for the print-paths generated in the subsequent stage. Furthermore, target distances between two neighboring layers are $2r$.

Given block-faces and their contact types with the print-paths.

Rule 1. A layer is never quasi-parallel with a block-face that has circular-contacts (CCs) with the print-paths.

Sub-layers are first generated in each hex-cell of a block without violating Rule 1. A sub-layer, l , is formed by sequentially moving opposite quads by r along (quasi-perpendicular) hex-edges (if l is next to a hex-face) or $2r$ (otherwise) in a hex-cell. This operation stops when whole hex-cell is covered by the sub-layers. These sub-layers are then merged with sub-layers of neighboring hex-cells to obtain layers in b . Fig. 3.7 illustrates layer generation strategy for a block (a). Sub-layers (b, in different colors) for each hexahedral are first generated based on a distance ($2r$) between them. Layers are then obtained by combining sub-layers (c, See the layer in red). The layers generated are depicted with different colors in (d).

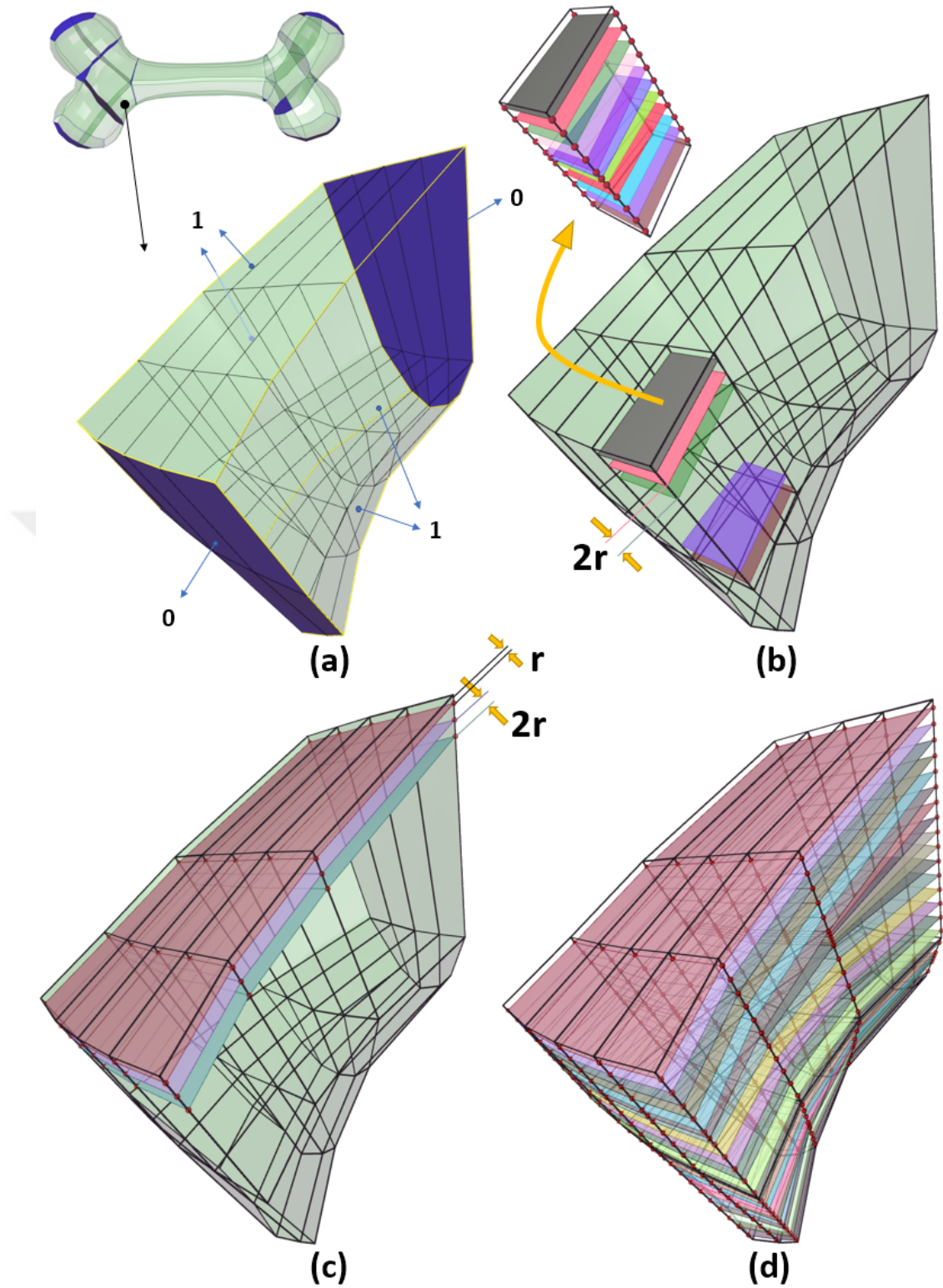


Figure 3.7 : (a) A block for a bone model having contact types (dark and light green block-faces have CC and RC, resp.). (b) Generation of sub-layers (in different colors) having distances, $2r$, between them for a hexahedral. (c) A layer (in red) obtained after merging sub-layers, having distances of $2r$ between layers. If the layer is next to a hex-face, this distance is r . (d) Finally obtained layers (in different colors).

To generate print-paths on a layer, contact types of a block are copied to edges connecting quad-points representing the layer boundaries. Let $c_{i,j}^{i+1,j}$ denote a contact type for edges connecting two quad points, $p[i,j]$ and $p[i+1,j]$. We then generate partial print-paths in each quad separately by selecting a print-path generator (G_1, G_2, G_3, G_4) having the lowest cost. This means that contact types, $c_{i,j}^{i+1,j}$, $c_{i+1,j}^{i+1,j+1}$, $c_{i+1,j+1}^{i,j+1}$ and $c_{i,j+1}^{i,j}$, for a quad, $q[i,j]$, are set in this step. Applying this print-path generation strategy from quads (nearer to layer boundaries) to those (far from the boundaries) enables us to generate a set of print-paths on a layer. Fig. 3.8 illustrates a block with contact types (top-left) and copying these types to block-edges of a block-face (in orange). Contact types for quads incident to block-edges are first assigned (a). Contact types for sub-layers/quads (in orange) incident to block-edges are then assigned (b). From outer to inner sub-layers, all contact-types are assigned (c, pink sub-layers).

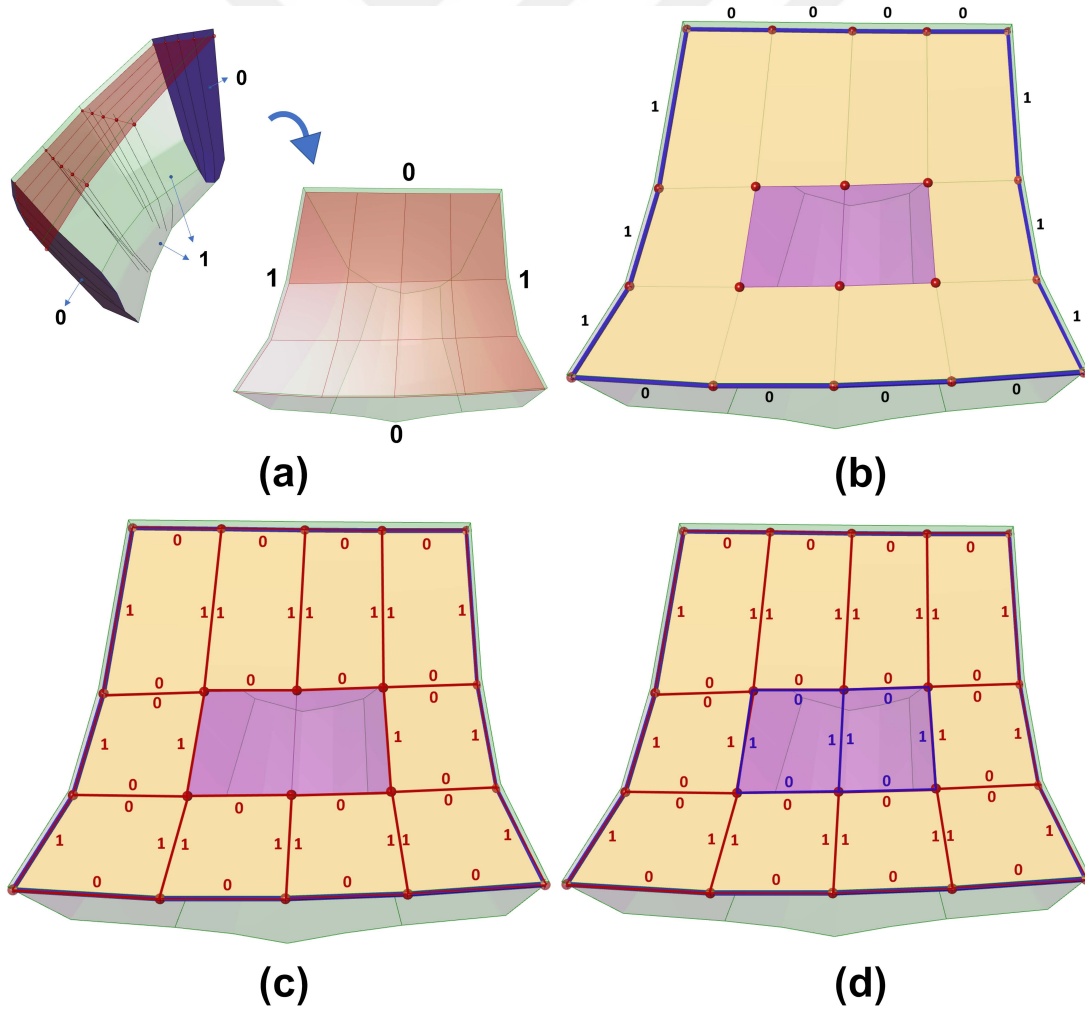


Figure 3.8 : Transferring block contact types (a) to the block-edges (b). From outer (orange) to inner (pink) sub-layers, all contact-types are assigned (b-d).

Edges connecting quad-points are offset to form (partial) print-paths in each quad, $q[i, j]$. Fig. 3.9 shows such offsetting process in a layer to generate print-paths. After assigning contact types for each quad (a), we perform offsetting (b-c) in each quad separately using print-path generators ($G_1, \dots, G_6$). Print-path generation process in quads highlighted using yellow and pink rectangles are shown in (d-e). The (partial) print-paths in adjacent sub-layers of a same layer are then joined to obtain a single continuous print-path. Print-paths in neighboring layers may overlap as the distance between a layer and a layer on its top-bottom may be smaller than $2r$, and therefore, a trimming process is ultimately performed as in [3].

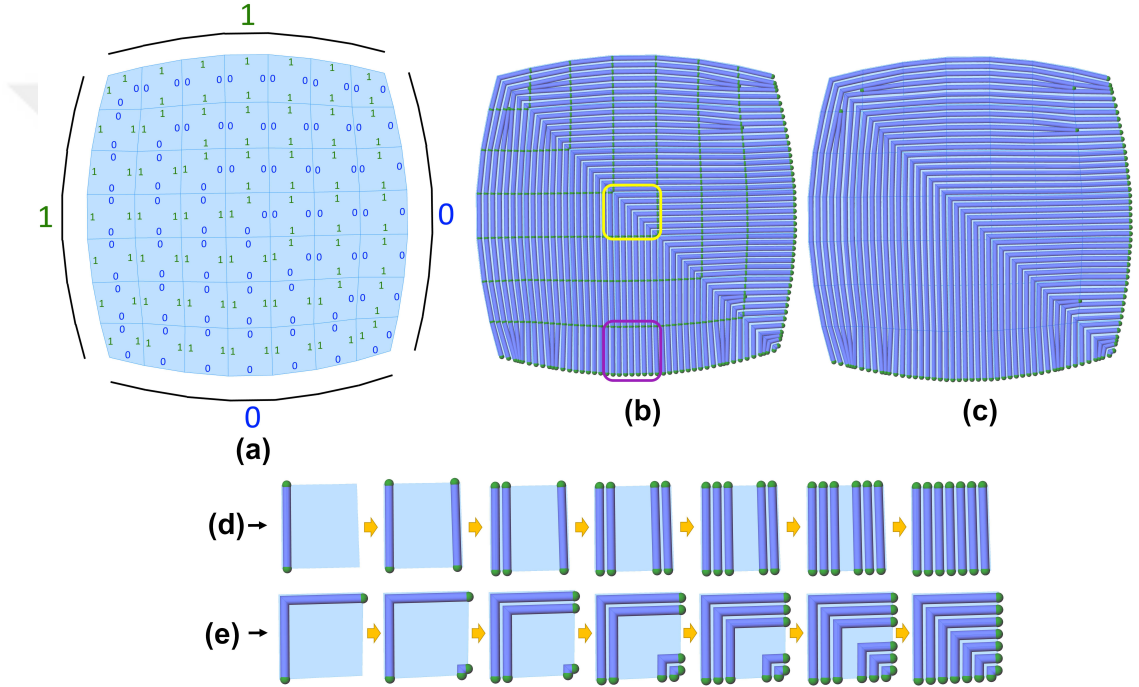


Figure 3.9 : Contact types from block-faces are copied to all quads (a). Print-paths are generated using print-path generators ($G_1 - G_6$) based on these contact types (b-c). Step-by-step print-path generation for the regions enclosed by yellow and pink rectangles are shown in (d) and (e), resp.

Unlike curvature-aware print-paths of Gunpinar et al. [3], layering strategy here guarantees continuity of print-paths between layers of neighboring blocks. This is crucial to obtain a printed part having good mechanical properties. Additionally, in case of a hexahedral mesh with hex-edge sizes smaller than $2r$, a coarser hexahedral mesh can be obtained by merging neighboring hex-cells. The proposed layering approach can then be utilized.

3.3 Print-path Planning for Multi-axis AM

We assume a 5-axis AM having an extruder rotating along two-axes for print-path planning. A print-path, $T = \{t_k, k = 0, \dots, R-1\}$, is a set of successive points lying on a set of consecutive, adjacent sub-layers with normals, $\{n_k\}$. Material deposition direction for the extruder (i.e., extruder orientation) at t_k is taken as n_k . A cone with a cone angle, α , bounding the extruder is called *BC*. $\mathcal{V}(T)$ denotes a volume of a sweep for BC along T , where BC is orientated along n_k at t_k . BC's orientation between the (successive) print-path points, p_k and p_{k+1} , is obtained by smoothly rotating BC from a cone axis, n_k , to n_{k+1} . Fig. 3.10(a) shows a print-path (in green) and its BCs (yellow cones) directed along $\{n_k\}$ at print-path points, $\{p_k\}$. Volume of a sweep for BC, $\mathcal{V}(T)$, is obtained by smoothly rotating BCs along two consecutive print-path points (b).

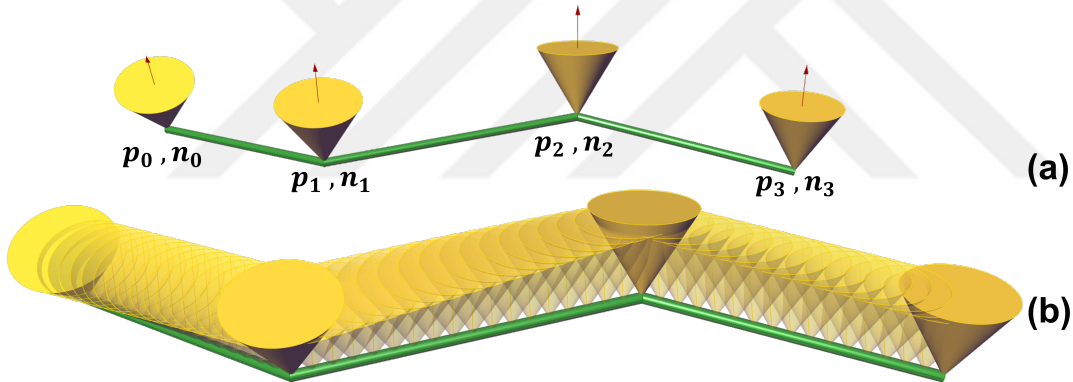


Figure 3.10 : BCs (yellow cones) directed along normals, $\{n_k\}$, at points, $\{p_k\}$, along print-paths in green (a). Extruder orientations are found by smoothly interpolating cone orientations along the print-path (b).

Definition 3.3.1 A print-path, T_m , is below a print-path, T_n , denoted $T_m < T_n$, if there is an intersection between T_n and $\mathcal{V}(T_m)$.

Definition 3.3.2 A cycle-of-invisibility is an arrangement of $T_m < T_n$ and $T_n < T_m$, where T_m and T_n hide each other.

Note that a cycle-of-invisibility may also occur in a print-path such that $T_m < T_m$. An accessibility graph, G , is formed between print-paths, as follows:

1. A node is assigned for each print-path, $t_k \in T$.

2. A directed link is added from a node corresponding to t_m to that corresponding to t_n if $T_m < T_n$.

There may exist cycle-of-invisibilities in G , which should be resolved to guarantee manufacturing (deposition) of all print-paths. For such print-paths, multiple extruder orientations are added at all print-path points, p_k , as carried out by Gunpinar and Cam [4]. Dijkstra's algorithm [38] is then applied to find the collision-free extruder orientations that are similar to $\{n_k\}$.

Definition 3.3.3 *A print-path is a preceding print-path if there is no graph link directed to it.*

Fig. 3.11 (a, b) shows preceding print-paths (in green and blue) at two consecutive printing time-steps. Extruder (BC) with collision-free orientations is shown with yellow cone. Once preceding print-paths are ordered, nodes corresponding to them and links from/to them are removed from G . The algorithm again finds next preceding print-paths for setting manufacturing orders. Print-path sorting process is terminated once no nodes are left in G .

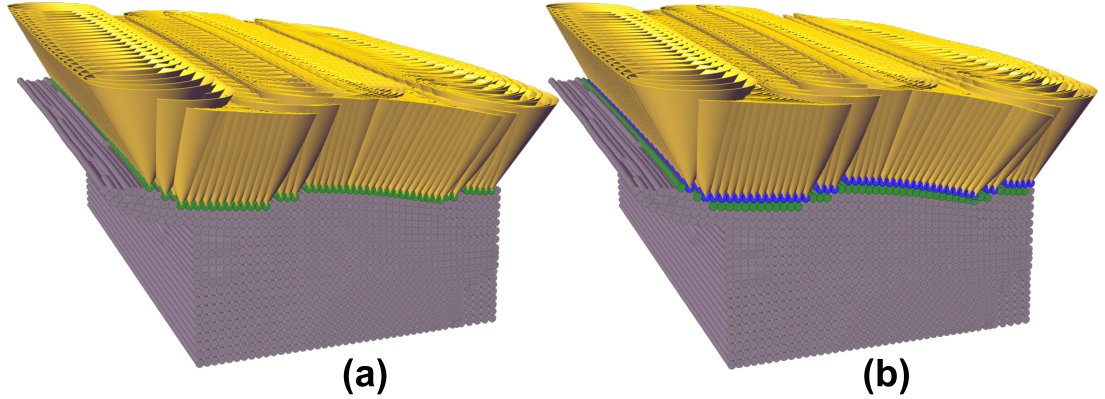


Figure 3.11 : Preceding print-paths (in green and blue) at two consecutive printing time-steps (a and b) for a chamfer model. Yellow cones represent BCs and have different orientations along print-paths.



4. RESULTS AND DISCUSSION

Various hexahedral mesh models (taken from Hexalab [39]) was employed to validate the algorithm provided in this paper. Fig. 4.1 shows generated print-paths for a double-torus (a), a joint (b), a gear (c) and a chamfer (d) models. Small images near the models depict block-faces having CCs (dark blue) and RCs (light green). It can be observed that the print-paths continuously elongate through neighbor blocks without having crosswise-contacts (between them) even though print-paths are separately generated in each block. Furthermore, print-paths are aligned in one of the principal curvature directions thanks to the hexahedral meshes generated using this criterion. Fig. 4.2 shows a fan disk model with block-faces (a) having CCs (in dark blue) and RCs (in light green) and print-paths (b). Print-paths in the widest part of the model are highlighted and colored for better observation (c). Notice print-paths labeled by 1 and 3. They pass through several blocks and continue from one side to another boundary. They both turn smoothly at their middle. Observe print-path labeled by 2, which pass around a singular-point. No cross-wise contact occur there.

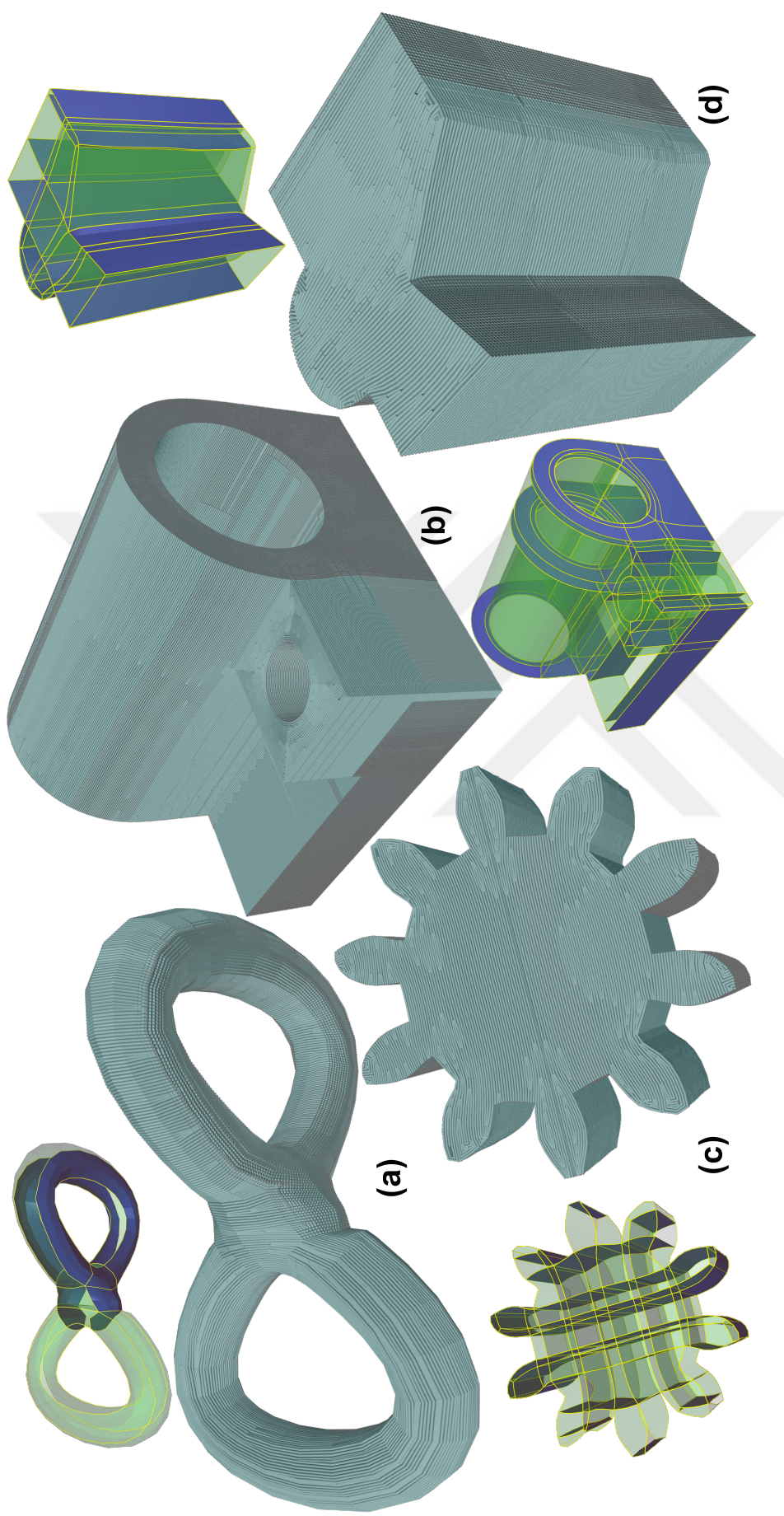


Figure 4.1 : Generated print-paths for various hexahedral mesh models: A double-torus (a), a joint (b), a gear (c) and a chamfer (d). Small images depict block-faces having circular-contacts (dark blue) and curve-contacts (light green).

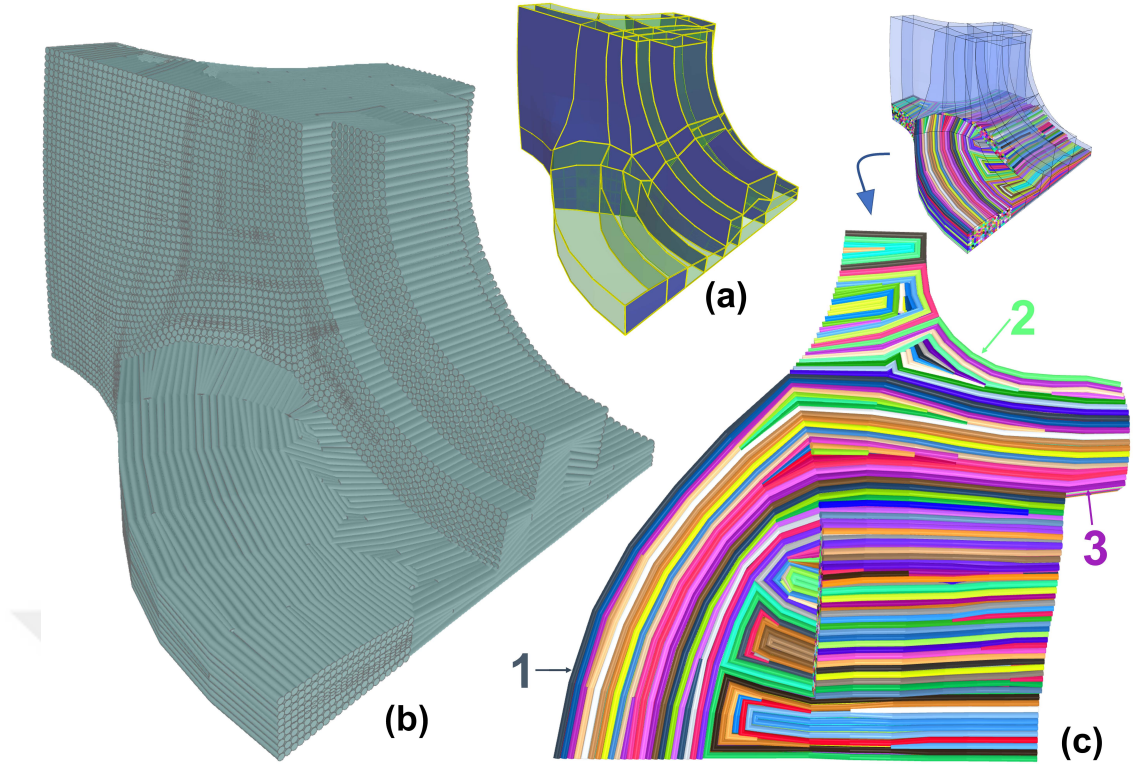


Figure 4.2 : Print paths obtained for a fan disk model (a) after computing (b) CCs (dark blue) and RCs (light green). Some print-paths are colored and shown to better analyze path continuities of print-paths (c). Print-paths labeled by 1 and 3 elongate all along the model, which, in fact, goes through multiple blocks. Print-path labeled by 2 is around a singular-point.

Given a hexahedral mesh with a (unique) block-mesh, print-paths obtained are not canonical (unique). In other words, different sets of print-paths are obtained using the proposed contact type assignment method (Algorithm 1 in Appendix) due to its randomized/generative nature. Recall that contact types are assigned based on energy functions (E_1 , E_2 , F_1 and F_2). In case there exists multiple selections (of block-faces and their contact types) available (due to same energy values for these selections), a random block-face (with contact types) is chosen. Such randomization can increase number of different print-path sets to be obtained, which can be preferable depending on the application. Fig. 4.3 illustrates three different print-path sets results for a hanger model, all of which do not contain any cross-wise contacts between the obtained print-paths. It can be observed that how elongations of print-paths change according to different print-path sets obtained. We think that this variation can be one of the main achievements of this study. A designer can select suitable print-paths depending on his/her intended usage/application.

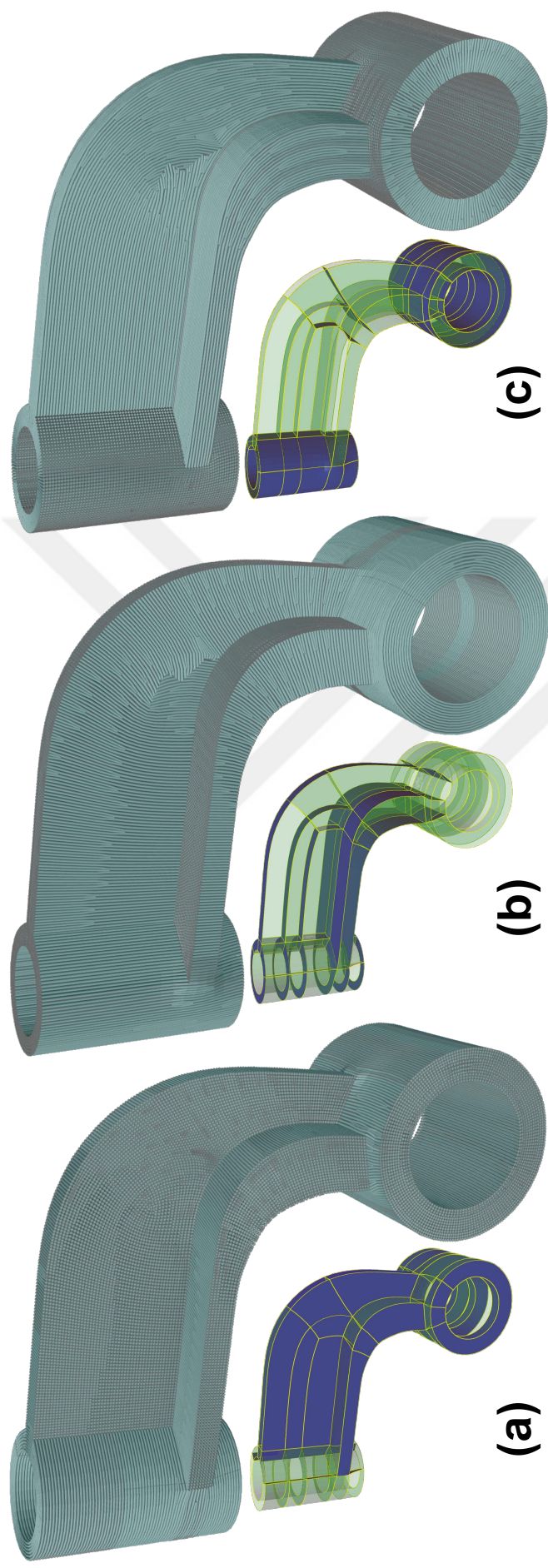


Figure 4.3 : Different print-path sets (a-c) obtained using the proposed method for a hanger model. CCs (dark blue) and RCs (light green) are shown in below images.

Fig. 4.4 (a-b) shows obtained layers of a fan disk and a chamfer models. Print-paths generated from these layers are shown in Figures 4.2 and 4.6. Layers smoothly cover the models, while a defect (depicted with the yellow frame in (c)) in the hexahedral mesh affects the layers (b), which is directly transferred to the print-paths (d).

Print-paths of a part under a loading condition are also generated (Refer Section 3.2.3). Due to the anisotropic material properties, print-paths are elongated in the maximum(absolute) principal stress directions. Fig. 4.5 shows a bone model under a loading situation having forces applied on two different block-faces (a), which produces a maximum principal stress directions (b). Contact types are first assigned using $\tau_2 = 70^\circ$ and $\tau_1 = 500$ introduced in Section 3.2.3. Algorithm 1 is then executed to set remaining (unassigned) contact types (c). Observe how print-paths (d) are compatible with the maximum stress directions, while no cross-wise contacts occur. It is worth mentioning here that depending on the parameter settings of τ_1 and τ_2 different print-paths sets can be obtained (by means of Algorithm 1's generative nature), which are all compatible with the given maximum stress directions.

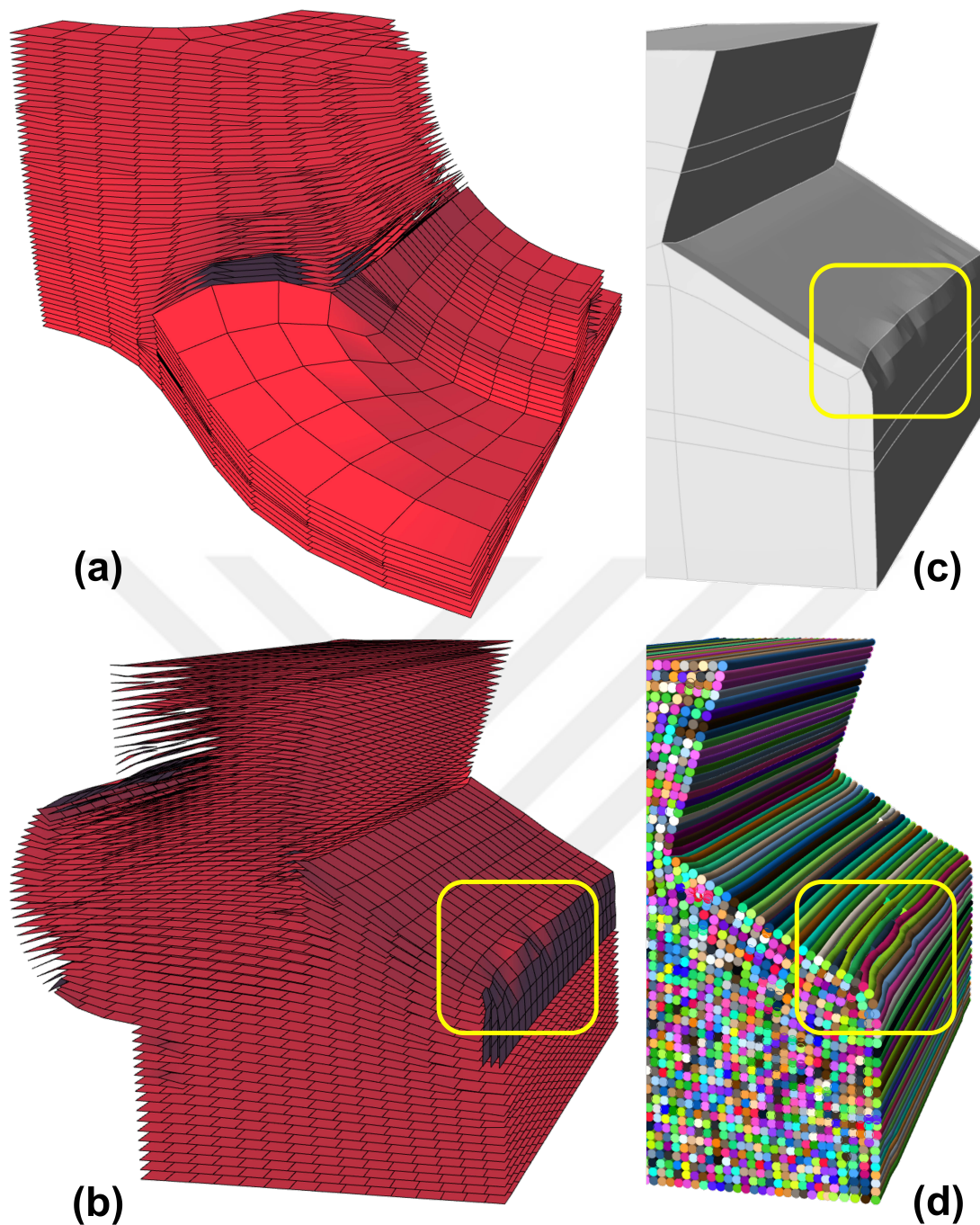


Figure 4.4 : Layers (a-b) obtained for a fan disk and a chamfer model. A defect (depicted with the yellow frame) on the chamfer model (c) affects the layers (b) and print-paths (d).

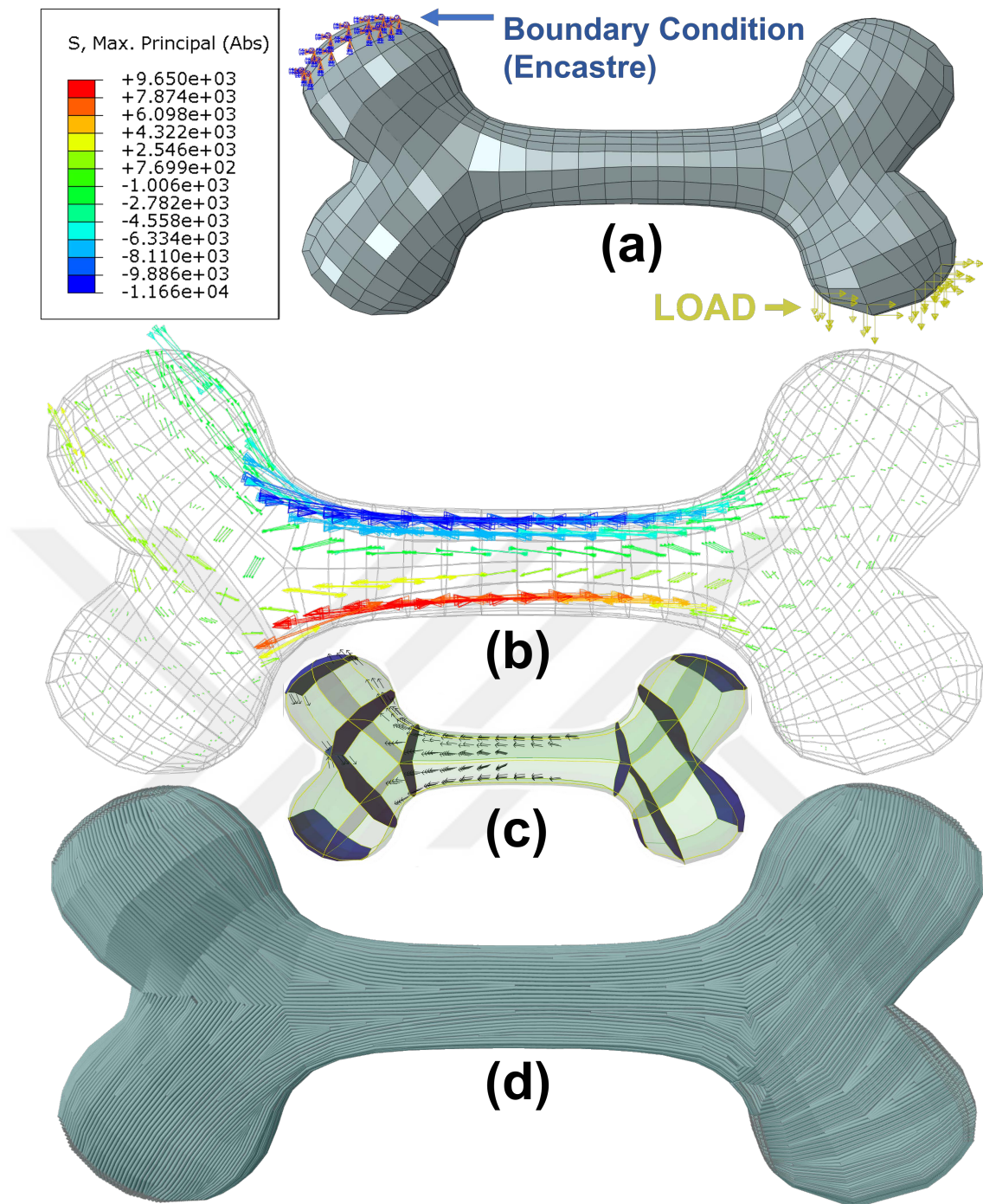


Figure 4.5 : Given a part under a specified loading condition having forces applied on block-faces (a), maximum principal stress directions are computed (b). Contact types for some block-faces are assigned based on these directions (c), and print-paths are then generated, which are compatible with the directions (d).

4.1 Computational Time

A PC with an Intel Core i5-6300HQ CPU 2.30GHz processor and 16 GB memory was used in the experiments of this study. The implementation was single-threaded. The proposed algorithms (in this paper) were implemented in C# using Rhino - Grasshopper [40] library. Table 4.1 shows computational times for different hexahedral mesh models in Figures 4.1, 4.2, 4.3 and 4.5. Higher the number of hexahedra, higher total computational time is required in general. Processing times for assigning contact types depends also on number of blocks (See fan disk and bone models). Furthermore, computational times for generating layers and print-paths time mainly depends on the print-path radius used in addition to the number of hexahedra in a mesh.

Table 4.1 : Computational times of the algorithm steps for different models (n_{hex} : number of hexahedra of a hexahedral mesh, n_b : number of blocks after the block decomposition (Section 3.1), r : print-path radius, t_a : computational time (in milliseconds) for assigning contact types for blocks (Algorithm 1), t_p : computational time (in seconds) for generating layers and print paths (Section 3.2.4)).

<i>Models</i>	n_{hex}	n_b	r	$t_a(ms)$	$t_p(s)$
<i>Bone</i>	844	16	0.3	85	48
<i>Column</i>	1276	18	0.4	90	29
<i>Chamfer</i>	5773	22	0.4	280	194
<i>Chamfer-R</i>	5773	22	0.3	276	227
<i>Fandisk</i>	357	49	0.3	172	16
<i>Gear</i>	8640	85	0.2	695	260
<i>Wrench</i>	1796	61	0.3	255	33
<i>Double-torus</i>	632	5	0.25	31	88
<i>Hanger-Fig.4.3(a)</i>	4539	50	0.25	310	221
<i>Hanger-Fig.4.3(b)</i>	4539	50	0.25	290	201
<i>Hanger-Fig.4.3(c)</i>	4539	50	0.25	301	193
<i>Joint</i>	17784	83	0.35	767	1345

4.2 Virtual and Physical Validation of Print-paths

Print-paths obtained in this work have been validated via virtual manufacturing. An AM simulator has been used to simulate the AM processes for a column, a chamfer, a fan disk, a hanger, a joint, a bone, a wrench and a gear models (Watch video [41]). Fig. 4.6 illustrates manufacturing processes for the fan disk and chamfer models, where yellow cones represent the extruder and its orientations at print-path points. Recall that extruder orientations are obtained from the layer normals. Therefore, extruder is generally perpendicular to the print-paths, and its orientations smoothly changes along the print-paths.

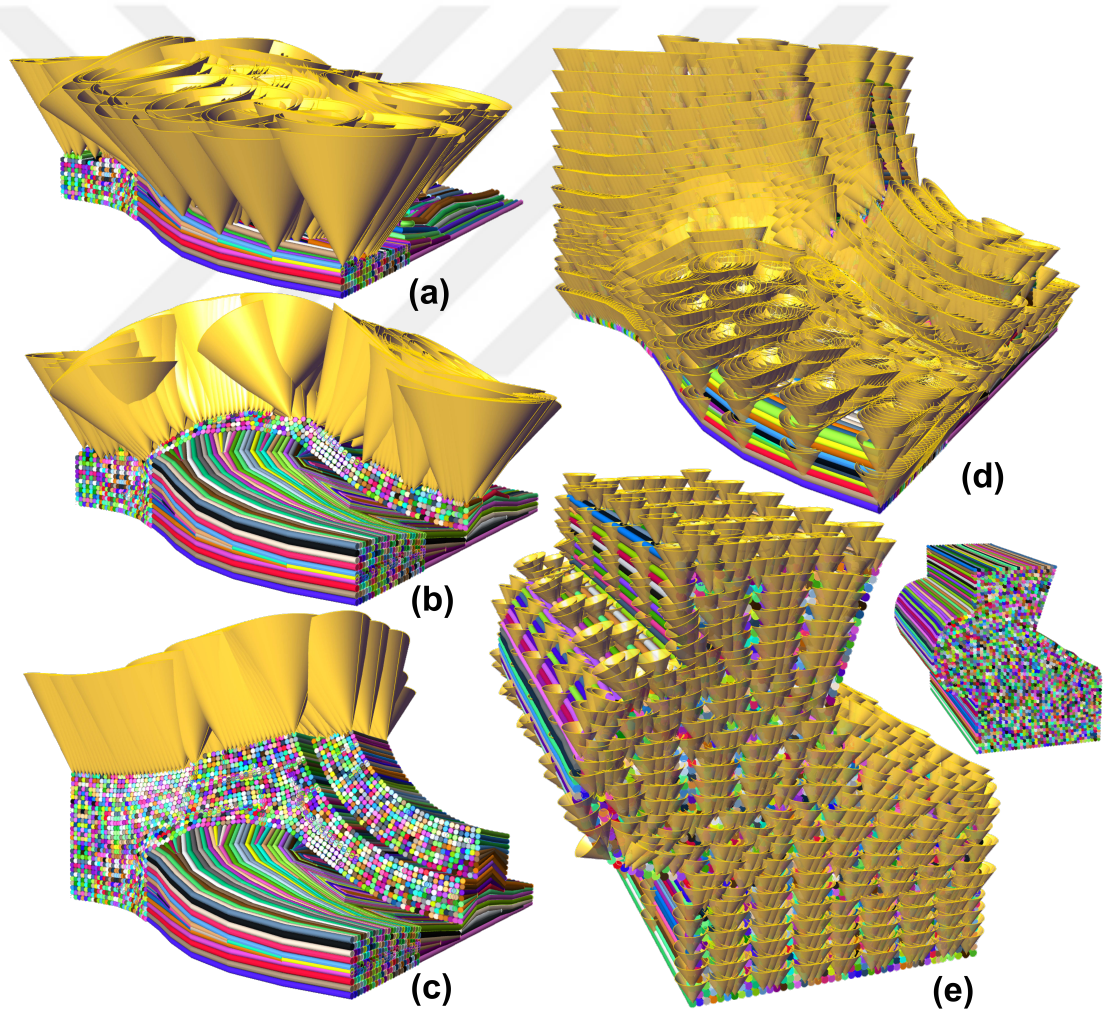


Figure 4.6 : BC (yellow cone) orientations along the print-paths for the fan disk (a-d) and chamfer (e) models.

Additionally, a column, a chamfer, and a fan disk models whose print-paths shown in Figures 1.2,4.1,4.2 were manufactured using 5AxisMaker [42], an AM machine.

Fig. 4.7 (a) and (b) depicts two different print-paths for a column model due to generative (randomized) nature of the proposed algorithm. No cross-wise contacts have been observed. In (b), notice closed (circular or rectangular) print-paths. Here, no fluid outlet (or inlet) was allowed for the block-faces so that the model boundaries were acted as walls. Moreover, in (c), manufacturing result for a chamfer model is shown. The print-paths followed the model's features (sharp corners). Because of the mesh defect (highlighted with yellow frame) discussed before (Refer Figure 4.4), an undesirable surface quality can be seen. Therefore, the print-paths were highly affected by the mesh quality, and therefore, either smoothing meshes or utilizing meshes from additive manufacturing-aware hex-meshing algorithms can avoid such problems.



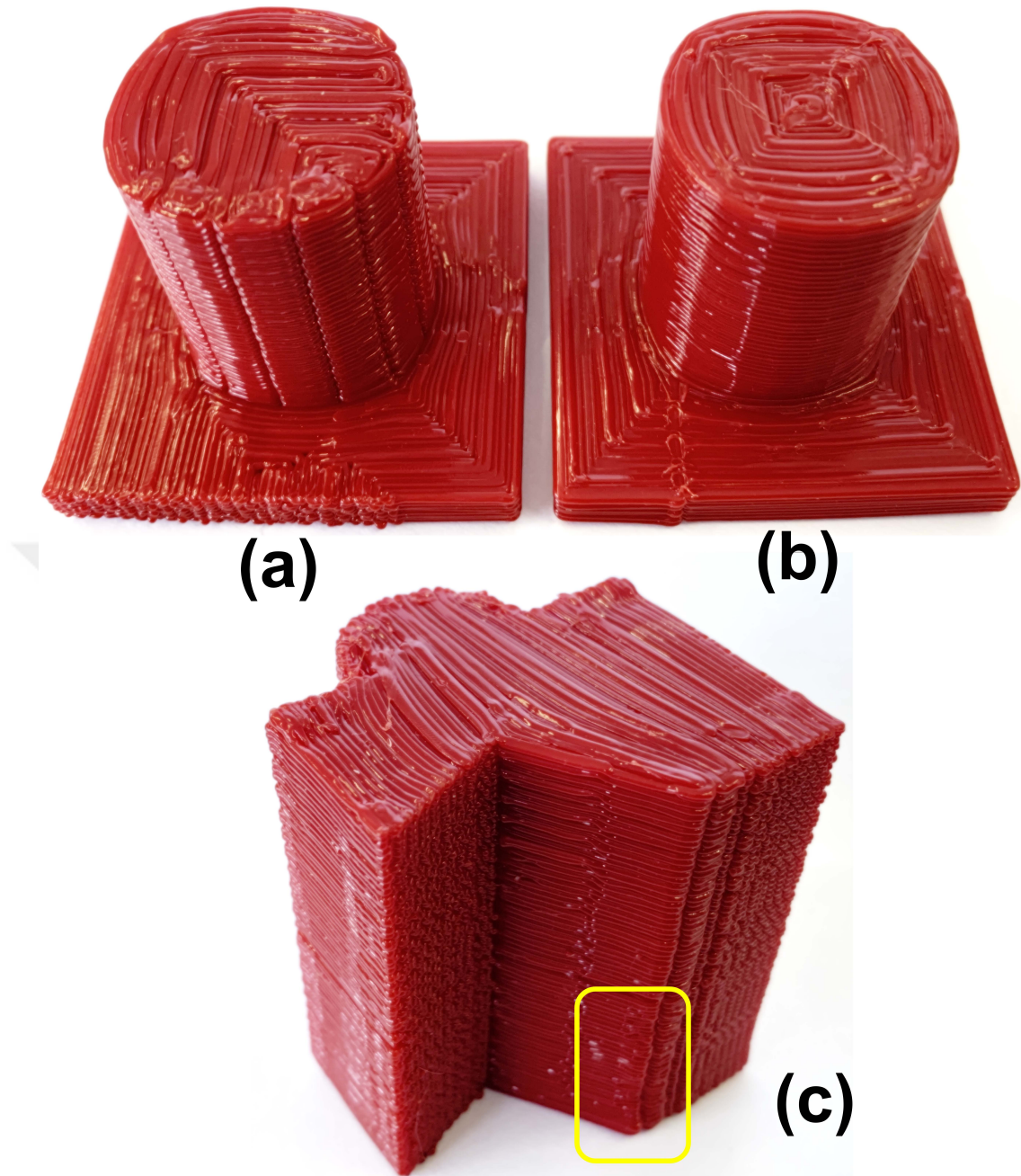


Figure 4.7 : Manufactured models for a column (a, b), and a chamfer models (c).

Fig. 4.8 shows print-paths and manufacturing results (via 3-axis) for a fan disk model obtained from Gunpinar [3] (a) and the present work (b, See also Figure 4.2). Inclined part (depicted with yellow curve) of the fan disk was represented using print-paths till the face with cyan boundaries in (b). While, in (a), the print-paths were shorter and did not reach this face. This negatively affected the surface quality of the manufactured model. In addition to this, stair-step effect has been reduced in (b).

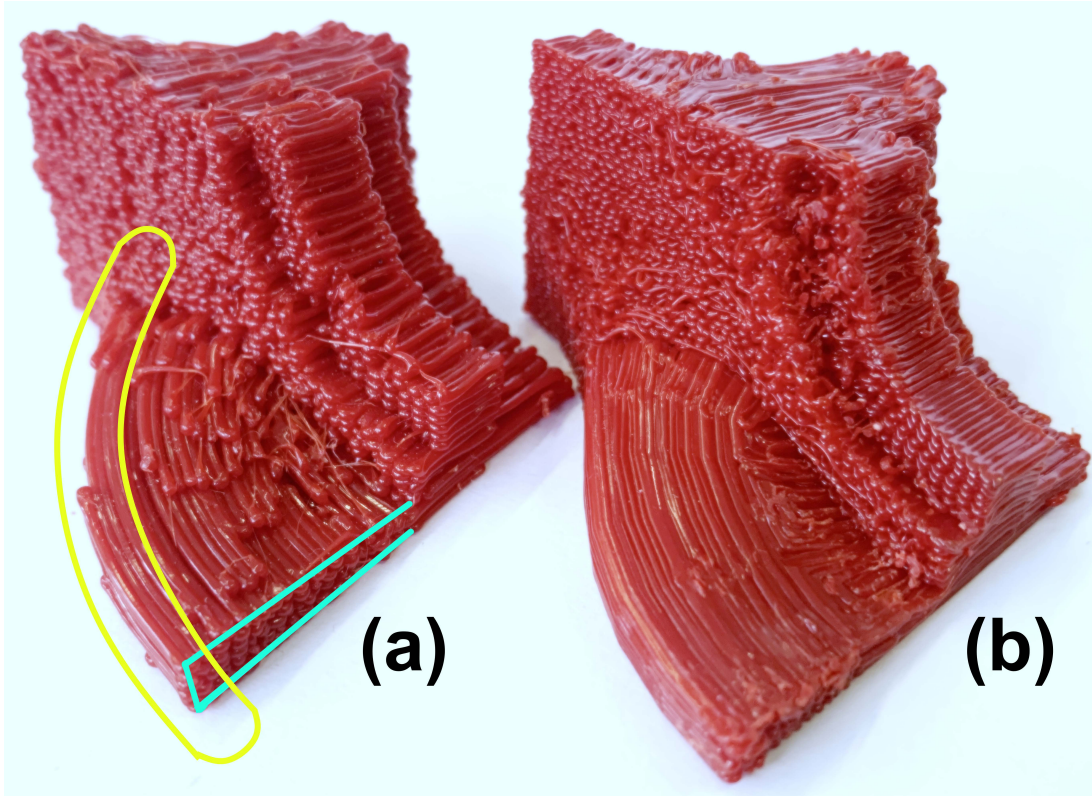


Figure 4.8 : Manufactured model for a fan disk for the print-paths of Gunpinar [3] (a) and the proposed method (b).

A chamfer model shown in Figure 4.9 (a, Chamfer-R in Table 4.1) has been rotated, and a support structure was added as discussed in [3]. Manufacturing process for this model can be seen in video [43]. The manufactured part before (c) and after (d) dismantling the support structure is also shown. A nozzle orientation is shown in (b), which is obtained from a sub-layer normal (Section 3.3). If normals between the neighboring sub-layers are variant, a post-processing is added to smooth the nozzle orientations.

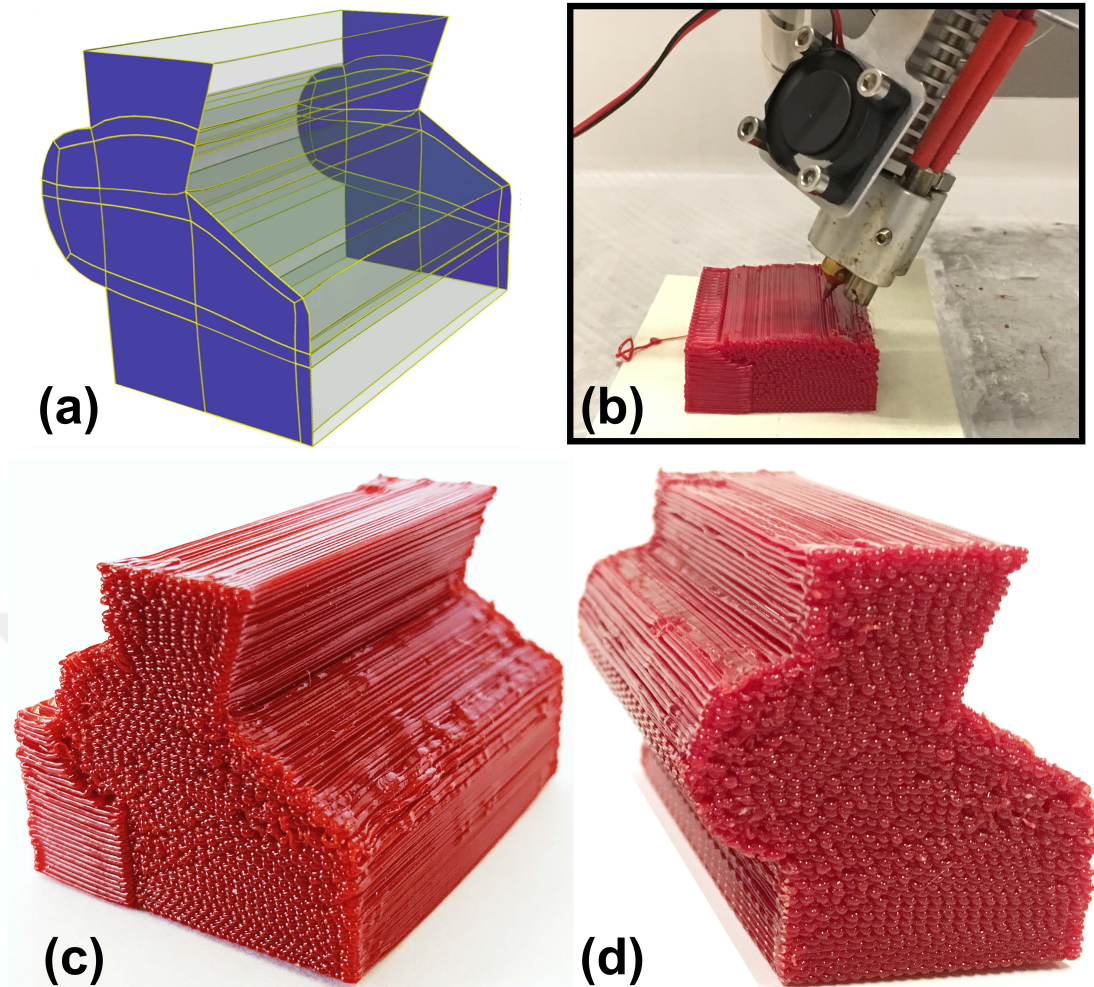


Figure 4.9 : A chamfer model (a) is manufactured, and a nozzle orientation (b) is shown during a manufacturing time-step. Part with (c) and without (d) a support structure is depicted.



5. CONCLUSIONS AND FUTURE WORKS

This paper introduces an approach for the generation of curvature-aware print-paths (without crosswise-contacts between them) from a hexahedral mesh model. We inspire from streamlines in fluid flow to design such print-paths. Blocks are first obtained from a hexahedral mesh, in which layers are created based on fluid flow-inspired print-path generation operators. Print-paths are then generated on these layers. Finally, a collision-free production sequence is set for these print-paths.

Given a block-mesh with contact types between block-faces and print-paths, canonical (unique) print-paths can be obtained. Utilizing only contact types (assigned for each block-face) allows compact storage of print-paths.

As a future work, fluid flow inspired strategy can be extended to G-Lattices [37], a novel lattice structure family. To decompose a hexahedral mesh into blocks, 3D motorcycle complex [44] will be utilized, which can reduce the number of blocks. Finally, we hope that the present work motivates more additive manufacturing-aware hex-meshing algorithms to obtain desirable AM print-paths.



REFERENCES

- [1] **Ezair, B., Fuhrmann, S. and Elber, G.** (2018). Volumetric covering print-paths for additive manufacturing of 3D models, *Computer-Aided Design*, 100, 1–13.
- [2] **Gunpinar, E. and Armanfar, A.** (2022). Helical5AM: Five-axis parametrized helical additive manufacturing, *Journal of Materials Processing Technology*, 304, 117565.
- [3] **Gunpinar, E.** (2021). Curvature and Feature-Aware Print-Paths from Hexahedral Meshes for Additive Manufacturing, *Computer-Aided Design*, 141, 103101.
- [4] **Gunpinar, E. and Cam, S.** (2022). 4 and 5-Axis additive manufacturing of parts represented using free-form 3D curves, *Graphical Models*, 120, 101137.
- [5] **Gao, X., Deng, Z. and Chen, G.** (2015). Hexahedral Mesh Re-Parameterization from Aligned Base-Complex, *ACM Trans. Graph.*, 34(4).
- [6] **Armstrong, C.G., Fogg, H.J., Tierney, C.M. and Robinson, T.T.** (2015). Common Themes in Multi-block Structured Quad/Hex Mesh Generation, *Procedia Engineering*, 124, 70–82, 24th International Meshing Roundtable.
- [7] **Yu, W., Zhang, K. and Li, X.** (2015). Recent algorithms on automatic hexahedral mesh generation, *2015 10th International Conference on Computer Science & Education (ICCSE)*, pp.697–702.
- [8] **Shepherd, J. and Johnson, C.** (2008). Hexahedral mesh generation constraints, *Engineering with Computers*, 24, 195–213.
- [9] **Pietroni, N., Campen, M., Sheffer, A., Cherchi, G., Bommes, D., Gao, X., Scateni, R., Ledoux, F., Remacle, J. and Livesu, M.** (2022). Hex-Mesh Generation and Processing: A Survey, *ACM Trans. Graph.*, 42(2).
- [10] **Nieser, M., Reitebuch, U. and Polthier, K.** (2011). CubeCover– Parameterization of 3D Volumes, *Computer Graphics Forum*, 30(5), 1397–1406.
- [11] **Li, Y., Liu, Y., Xu, W., Wang, W. and Guo, B.** (2012). All-Hex Meshing Using Singularity-Restricted Field, *ACM Trans. Graph.*, 31(6).
- [12] **Kowalski, N., Ledoux, F. and Frey, P.** (2016). Smoothness driven frame field generation for hexahedral meshing, *Computer-Aided Design*, 72, 65–77, 23rd International Meshing Roundtable Special Issue: Advances in Mesh Generation.

- [13] **Solomon, J., Vaxman, A. and Bommes, D.** (2017). Boundary Element Octahedral Fields in Volumes, *ACM Trans. Graph.*, 36(4).
- [14] **Liu, H., Zhang, P., Chien, E., Solomon, J. and Bommes, D.** (2018). Singularity-Constrained Octahedral Fields for Hexahedral Meshing, *ACM Trans. Graph.*, 37(4).
- [15] **Takayama, K.** (2019). Dual Sheet Meshing: An Interactive Approach to Robust Hexahedralization, *Computer Graphics Forum*, 38(2), 37–48.
- [16] **Guo, H.X., Liu, X., Yan, D.M. and Liu, Y.** (2020). Cut-Enhanced PolyCube-Maps for Feature-Aware All-Hex Meshing, *ACM Trans. Graph.*, 39(4).
- [17] **Livesu, M., Pietroni, N., Puppo, E., Sheffer, A. and Cignoni, P.** (2020). LoopyCuts: Practical Feature-Preserving Block Decomposition for Strongly Hex-Dominant Meshing, *ACM Trans. Graph.*, 39(4).
- [18] **Palmer, D., Bommes, D. and Solomon, J.** (2020). Algebraic Representations for Volumetric Frame Fields, *ACM Trans. Graph.*, 39(2).
- [19] **Livesu, M., Pitzalis, L. and Cherchi, G.** (2022). Optimal Dual Schemes for Adaptive Grid Based Hexmeshing, *ACM Transactions on Graphics*, 41(2), 1–14.
- [20] **Gao, X., Panozzo, D., Wang, W., Deng, Z. and Chen, G.** (2017). Robust Structure Simplification for Hex Re-Meshing, *ACM Trans. Graph.*, 36(6).
- [21] **Cherchi, G., Livesu, M. and Scateni, R.** (2016). Polycube Simplification for Coarse Layouts of Surfaces and Volumes, *Computer Graphics Forum*, 35(5), 11–20.
- [22] **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.
- [23] **Alkadi, F., Lee, K.C., Bashiri, A.H. and Choi, J.W.** (2020). Conformal additive manufacturing using a direct-print process, *Additive Manufacturing*, 32, 100975.
- [24] **Zhao, G., Ma, G., Feng, J. and Xiao, W.** (2018). Nonplanar slicing and path generation methods for robotic additive manufacturing, *The International Journal of Advanced Manufacturing Technology*, 96, 3149–3159.
- [25] **Wu, C., Dai, C., Fang, G., Liu, Y.J. and Wang, C.C.** (2017). RoboFDM: A robotic system for support-free fabrication using FDM, *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp.1175–1180.
- [26] **Dai, C., Wang, C.C.L., Wu, C., Lefebvre, S., Fang, G. and Liu, Y.J.** (2018). Support-Free Volume Printing by Multi-Axis Motion, *ACM Trans. Graph.*, 37(4).

- [27] **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.
- [28] **Llewellyn-Jones, T., Allen, R. and Trask, R.** (2016). Curved Layer Fused Filament Fabrication Using Automated Toolpath Generation, *3D Printing and Additive Manufacturing*, 3(4), 236–243.
- [29] **Chen, L., Chung, M.F., Tian, Y., Joneja, A. and Tang, K.** (2019). Variable-Depth Curved Layer Fused Deposition Modeling of Thin-Shells, *Robot. Comput.-Integr. Manuf.*, 57(C), 422–434.
- [30] **Etienne, J., Ray, N., Panozzo, D., Hornus, S., Wang, C.C.L., Martínez, J., McMains, S., Alexa, M., Wyvill, B. and Lefebvre, S.** (2019). CurviSlicer: Slightly Curved Slicing for 3-Axis Printers, *ACM Trans. Graph.*, 38(4).
- [31] **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.
- [32] **Fang, G., Zhang, T., Zhong, S., Chen, X., Zhong, Z. and Wang, C.C.L.** (2020). Reinforced FDM: Multi-Axis Filament Alignment with Controlled Anisotropic Strength, *ACM Trans. Graph.*, 39(6).
- [33] **Chen, X., Fang, G., Liao, W.H. and Wang, C.C.** (2022). Field-Based Toolpath Generation for 3D Printing Continuous Fibre Reinforced Thermoplastic Composites, *Additive Manufacturing*, 49, 102470.
- [34] **Bi, D., Xie, F. and Tang, K.** (2021). Generation of Efficient Iso-Planar Printing Path for Multi-Axis FDM Printing, *Journal of Manufacturing and Materials Processing*, 5, 59.
- [35] **Xie, F., Jing, X., Zhang, C., Chen, S., Bi, D., Li, Z., He, D. and Tang, K.** (2022). Volume decomposition for multi-axis support-free and gouging-free printing based on ellipsoidal slicing, *Computer-Aided Design*, 143, 103135.
- [36] **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.
- [37] **Armanfar, A. and Gunpinar, E.** (2022). G-Lattices: A Novel Lattice Structure and Its Generative Synthesis Under Additive Manufacturing Constraints, *Journal of Mechanical Design*, 145(1), 011702.
- [38] **Bondy, J.A. and Murty, U.S.R.** (1976). *Graph Theory with Applications*, Elsevier North-Holland.
- [39] **Bracci, M., Tarini, M., Pietroni, N., Livesu, M. and Cignoni, P.** (2019). HexaLab.net: An online viewer for hexahedral meshes, *Computer-Aided Design*, 110, 24 – 36, <https://www.hexalab.net/>.

- [40] **URL-1**, <<https://www.rhino3d.com/6/new/grasshopper/>>, date retrieved 02.01.2023.
- [41] **URL-2**, <<https://youtu.be/pZYTDIq8yUg>>, date retrieved 02.01.2023.
- [42] **URL-3**, <<http://www.5axismaker.com/>>, date retrieved 02.01.2023.
- [43] **URL-4**, <<https://youtu.be/Qp3wngD11uM>>, date retrieved 02.01.2023.
- [44] **Brückler, H., Gupta, O., Mandad, M. and Campen, M.** (2022). The 3D Motorcycle Complex for Structured Volume Decomposition, *Computer Graphics Forum*, 41(2), 221–235.



CURRICULUM VITAE

Name Surname : Serhat ÇAM

EDUCATION:

- **B.Sc.** : 2018, Karadeniz Technical University, Faculty of Mechanical Engineering, Department of Mechanical Engineering
- **M.Sc.** : 2022, Istanbul Technical University, Faculty of Mechanical Engineering, Department of Mechanical Engineering

PUBLICATIONS

- 2022** Gunpınar, E., and Cam, S., 2022. “4 and 5-Axis additive manufacturing of parts represented using free-form 3D curves”, *Graphical Models*, vol. 120, pp. 101137., DOI : <https://doi.org/10.1016/j.gmod.2022.101137>.
- 2022** Özdemir, Yusuf Burak and Karpuzcu, Yalçın and Çam, Serhat and Günpınar, Erkan, 2022. “Wheel Rim Design Parametrization and Its Effect on Optimization”, *Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi*, vol. 24, no. 72, pp. 913 - 926., DOI : <https://doi.org/10.21205/deufmd.2022247220>.

