UNIVERSITY
OF TECHNOLOGY
SYDNEY

Computational design in digital and bio fabrication

by Hassan Bahrami

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of

Supervisor: A/Prof. Nico Pietroni
Co-Supervisor: Prof. Stuart Perry
Co-Supervisor: Dr. Carmine Gentile

University of Technology Sydney
Faculty of Engineering and Information Technology

May 2024

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, *Hassan Bahrami* declare that this thesis, submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the *School of Computer Science, Faculty of Engineering and IT* at the University of Technology Sydney.

This thesis is wholly my work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Production Note:
Signature removed prior to publication.

SIGNATURE: ______________________

[Hassan Bahrami]

DATE: 13th November, 2024

PLACE: Sydney, Australia

Dedication

Dedicated to the source of beauty, wisdom and goodness and his lovers . . .

Acknowledgments

I express my profound gratitude to Dr. Nico Pietroni for his exceptional guidance and unwavering support throughout my doctoral journey. His mentorship has been invaluable in shaping my research endeavours. Also, I extend my heartfelt appreciation to Prof. Stuart Perry for his enduring patience and steadfast encouragement during the challenging phases of my PhD. I have to thank Dr. Carmine Gentile as well, for teaching and guiding me in his lab, which was a valuable experience.

List of Publications

Related to the Thesis :

1. Physically-based simulation of elastic-plastic fusion of 3D bioprinted spheroids (Journal of Biofabrication)

2. Ruling Patches: Developable Mesh Decompositions (ACM Transactions on Graphics)

Talks and Workshops:

4. 3rd Australian Bioprinting Workshop for Tissue Engineering (Oct. 4-5 2021, Ultimo, NSW, Australia)

5. 4th Australian Bioprinting Workshop for Tissue Engineering (Dec. 1-2 2022, Ultimo, NSW, Australia)

6. 5th Australian Bioprinting Workshop for Tissue Engineering (Nov. 20-21 2023, Ultimo, NSW, Australia)

Abstract

Computational design holds promising applications in digital fabrication and biofabrication, offering innovative solutions to complex challenges in manufacturing and tissue engineering. In this thesis, we explore the application of computational design principles in these domains and present novel approaches to address two specific problems.

In the realm of biofabrication, we introduce a physically-based simulation framework for the elastic-plastic fusion of 3D bioprinted spheroids. Spheroids are microtissues containing cells organized in a spherical shape that are used in biofabrication to create human tissue. Specifically, we employ bioprinting spheroids to fabricate heart tissues in our lab. However, achieving tissue with the desired geometrical shape requires understanding how they fuse after printing.

Therefore, we have developed a physically-based simulation framework based on elastic-plastic solid and fluid continuum mechanics models using the smoothed particle hydrodynamics (SPH) method. This accurately captures the fusion process of spheroids and facilitates reverse engineering to achieve tissue with the desired shape. Our method can save significant time and costs compared to trial-and-error methods. Through extensive sensitivity and morphological analyses, we validate our simulations against in-vitro experiments, demonstrating their capability to predict and control tissue geometries.

Shifting our focus to digital fabrication, we introduce the concept of "Ruling Patches," a method that approximates triangular meshes with developable patches driven by surface line features. Developable shapes find practical application in manufacturing, where surfaces of three-dimensional shapes can be efficiently constructed from flat patches. We demonstrate the effectiveness of our method in achieving aesthetic and manufacturable patch layouts, showcasing its superiority over existing techniques.

Our work advances the understanding of complex phenomena in digital and biofabrication and opens new avenues for research and development in these fields. By providing efficient computational tools and frameworks, our contributions empower researchers and practitioners to tackle emerging challenges and drive innovation in manufacturing and tissue engineering.

Table of Contents

List of Figures

Figure **Page**

List of Tables

Table **Page**

Chapter 1

Introduction

In the ever-evolving landscape of scientific inquiry and technological innovation, the symbiotic relationship between computation and design has emerged as a driving force behind transformative advancements across diverse domains. Rooted in the seamless integration of mathematical algorithms, computational methodologies, and design principles, this interdisciplinary synergy has revolutionised how we conceptualize and create and opened new frontiers of exploration and discovery. At the forefront of this paradigm shift lie two pivotal domains: biofabrication and digital fabrication.

1.1 Computational design in bio fabrication

Within the realm of biofabrication, the convergence of computational design and tissue engineering holds profound implications for regenerative medicine and biomedical research. Tissue engineering endeavours to replicate the complex architecture and functionality of native tissues and organs, offering a promising avenue for addressing critical healthcare challenges such as organ shortage and tissue degeneration. Through the precise manipulation of cellular microenvironments and the orchestration of biological processes, researchers aim to engineer functional tissues capable of integration and regeneration within the human body [31, 66, 148].

In this context, computational design serves as a cornerstone, providing researchers with powerful tools to model, simulate, and optimize the intricate processes underlying tissue morphogenesis. By leveraging mathematical models, predictive algorithms, and simulation frameworks, computational biologists and bioengineers can unravel the complexities of cell behaviour, tissue development, and biomaterial interactions [27, 33, 68, 95, 96, 115, 132]. From the 3D bioprinting of cellular constructs to the simulation of tissue growth and remodelling, computational approaches empower researchers to navigate the multidimensional landscape of tissue engineering with precision and efficacy.

A key challenge in bioprinting is to develop accurate models to predict and guide how spheroids replicate and merge to form complete tissues [34, 134]. Due to the lack of direct control over the fusion process, the final bioprinted tissue can have a significantly different shape than expected. For example, Figure 1.1 shows microscopic images of a spheroid pair's spontaneous fusion. Identifying the parameters that control spheroid fusion is essential to predicting and fabricating a tissue with a desired shape and size. These parameters can include the hydrogel stiffness, spheroids density, and the distance between spheroids. We hypothesize that changing these parameters will affect the shape of the spheroids during the fusion. However, testing all possible parameter values is impractical due to the cost and time constraints. To address this constraint, we propose a simulation framework to model spheroid deformation during fusion. The simulation has its own parameters, allowing us to observe how spheroids behave under different conditions. Optimizing simulation parameters based on the desired shape of spheroids and doing reverse engineering can help scientists create tissue with the desired shape and save time and budget.

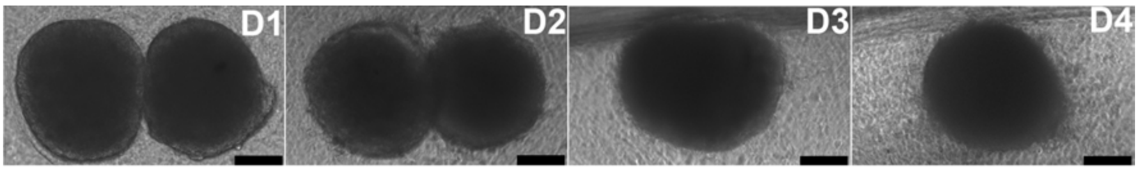


Figure 1.1: Fusion of bioprinted cardiac spheroids in alginate/gelatin hydrogels. Paired cardiac spheroids containing cardiac myocytes, endothelial and fibroblasts were bioprinted in 4% alginate 8% gelatin hydrogels and cultured for four days until complete fusion. Magnification bars equal 50μm

.

Simulation of the bioprinting process has the potential to significantly reduce experimental time and costs [13, 35, 104]. Therefore, this study focuses on developing a new computational and graphical framework for simulating spheroid fusion. The framework focuses on the physical and mechanical properties of fusing spheroids and allows us to replace classical trial-and-error methods. Our framework will enable us to predict the behaviour of spheroids during their fusion by tuning simulation parameters. This approach has not been used previously to model the spontaneous fusion process of spheroids.

The methodology described in this study employs and extends continuum mechanics models used in computer graphics to simulate elastic-plastic fluid, and elastic-plastic solid [72], which is going to be foundational of our models. A common approach to solving the equation and approximating a continuous physical model is to sample the domain into particles. These particles sample different physical quantities over the volume, such as mass, velocity, and viscosity. This method is usually called smoothed particle hydrodynamics (SPH) [141]. Using particle-based systems, we have developed two types of continuum mechanics models:

- The first model considers spheroids as an elastic-plastic solid material, therefore containing elastic, plastic and volume conservation forces.

- The second model considers spheroids as an elastic-plastic fluid material, therefore containing elastic, plastic, viscosity and pressure forces.

We adapted these techniques to simulate the behaviour of spheroids. To model elastic-plastic solid spheroids, we used the physically-based simulation based on smoothed particle hydrodynamics proposed by Mueller and colleagues [72]. Spatial derivatives are interpolated over volume using a Moving Least Squares (MLS) procedure. We use these derivatives to obtain physical quantities such as stresses and elastic forces in the continuous. To model elastic-plastic fluid spheroids and derive numerical solutions, we use the well-established Navier-Stokes equations (Figure 1.2).

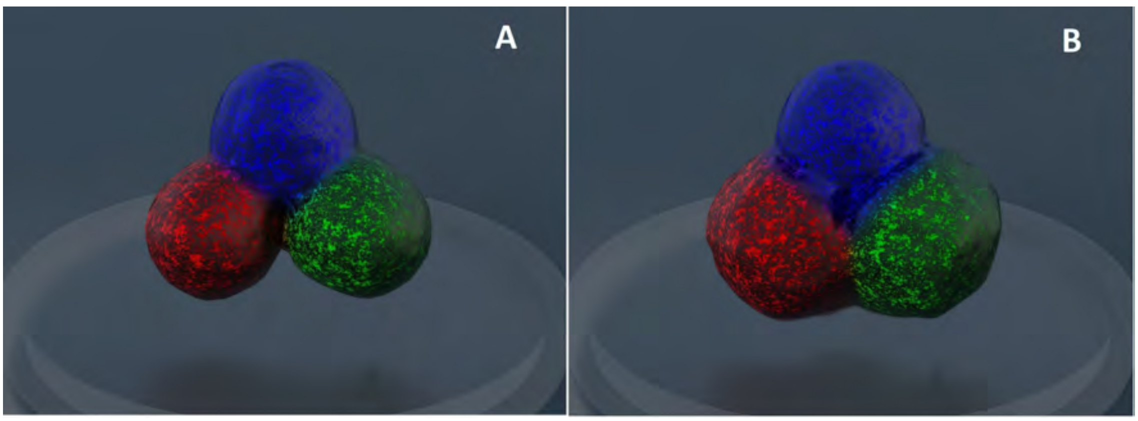


Figure 1.2: Fusion of bioprinted cardiac spheroids using Navier-Stokes equations

.

It is important to note that particles do not represent the cells inside the spheroids; instead, they are physical quantities. Our model's assumptions can be summarized as follows: i) Spheroids are assumed to be symmetrical and spherical continuum media; ii) The particles in our models carry physical attributes such as mass, velocity, and acceleration; iii) Spheroid fusion is assumed to be identical; iv) In the solid-based model, spheroids are considered solid elastic-plastic materials; v) In the fluid-based model, spheroids are modelled as incompressible fluids with elastic-plastic properties; vi) Particles of a spheroid are attracted to the centre of mass of the other spheroid.

Although the specific biological mechanisms underlying spheroid fusion are complex, our observations indicate that the process involves a combination of short-range intermolecular potential forces and surface adhesion forces between the cells of adjacent spheroids upon contact. In our study, we have employed the Lennard-Jones potential to model the short-range attractive forces between spheroids [138]. It has various applications in different fields such as molecular dynamics and chemistry [43, 135].

Regarding the surface adhesion forces, we consider the role of integrins, which are transmembrane receptors involved in cell-cell and cell-matrix interactions. Integrins bind to extracellular matrix components, such as fibronectin, collagen, and laminin, through specific ligand-binding sites. These integrin-mediated adhesion forces can be modelled using various approaches [122, 140]. To simulate this behaviour, we establish connections between particles of different spheroids at the moment of contact. By doing so, we simulate the formation of elastic-plastic links between different particles of adjacent spheroids.

This combined approach allows us to capture the attractive forces between spheroids, incorporating both the short-range intermolecular potential forces described by the Lennard-Jones potential and the surface adhesion forces mediated by integrins. By considering these physical interactions, we aim to provide a comprehensive understanding of the fusion process and its underlying biophysical mechanisms.

Simultaneously, in the realm of digital fabrication, computational design catalyzes innovation and creativity, redefining the boundaries of materiality, form, and fabrication. Additive manufacturing technologies, such as 3D printing, have democratized the production process, enabling designers to materialize intricate geometries and complex structures with unprecedented speed and accuracy. From architectural prototypes to customized medical implants, the marriage of computation and fabrication offers limitless possibilities for realizing creative visions and engineering solutions [10, 41, 85, 144, 146, 147].

1.2 Computatioanal design in digital fabrication

Developability is a well-investigated concept in differential geometry [85], and developable surfaces find broad applications in shape manufacturing, architecture, and garment design [144]. Developable surfaces can be crafted from flat patches that are bent but not stretched. Piecewise, developable shapes usually have a very distinct look because Gaussian curvature is concentrated along patch boundaries. In practice, modelling such shapes requires skills and expertise because of a delicate balance between the number of required patches, aesthetically pleasing placement of boundary seams, approximating the desired surface and adhering to developability constraints. This also renders the design space of piecewise developable surfaces highly complex.

In recent years, we have witnessed the emergence of numerous computational approaches towards developable shape modelling. In general, many of these methods accept as input a triangular mesh and propose a decomposition into a set of flat patches [10, 41, 146, 147]. Nevertheless, automatically finding a piecewise developable shape approximation of a general 3D surface still remains a challenge. Existing methods might struggle with patch seams not being well aligned with features of the target mesh [41, 147], gradient-based methods might not fully converge [121], or methods violate other fabrication constraints. While developability is necessary to enable the fabrication of flat inextensible sheets, it is not a sufficient condition for a design to be manufacturable. A design suitable for fabrication needs to guarantee that the generated patches are free of self-intersections. Moreover, to facilitate assembly, the patches should be smooth and easily bend to the desired shape, and the number of patches should be reasonable.

In the digital fabrication part of the thesis, we present an algorithm for converting an input three-dimensional triangular mesh into a compact set of developable patches. To guide the decomposition we use a smooth 4-rotational-symmetric vector field aligned with the principal curvature direction. This vector field allows us to align patches with interesting features on the mesh, as well as guide the direction of ruling lines that best approximate the input mesh. Ruling lines will help us to make ruled surfaces that are developable.

Ruled surfaces are surfaces where each point belongs to a straight line on the surface, and can be understood as surfaces generated by the motion of a straight line (the ruling line, or rules) [26]. They find many applications in shape manufacturing and architecture due to their favourable characteristics in terms of ease of construction, robustness, and visual appeal. For example, the rules can be implemented by straight beams connected by a textile (serving as a membrane connecting the beams) or by folding paper/cardboard along conveniently straight lines. Consequently, a useful and well-studied task is how to decompose a given shape into a small set of ruled surfaces, striving to maximize the geometric similarity and the length of the rules. This implies the need to orientate the rules of each ruled surface along the local minimal curvature direction, exploiting the ability of each ruled surface to bend freely in the other direction to follow the target shape.

When the design needs to be fabricated with inextensible materials (such as paper, cardboard, plywood, etc), this already difficult task is made more complicated by the need to ensure that each ruling surface is also developable [85, 102, 119]. It is well known that, in order to be developable, a ruled surface must fulfil an additional constraint: all the points along a given rule must share the same tangent plane [121]. In this sense, the task of decomposing a shape into ruled surfaces (with long rules) is one special case of the more general task of decomposing a shape into developable patches, which is a difficult and deeply studied task in geometry processing [121, 131].

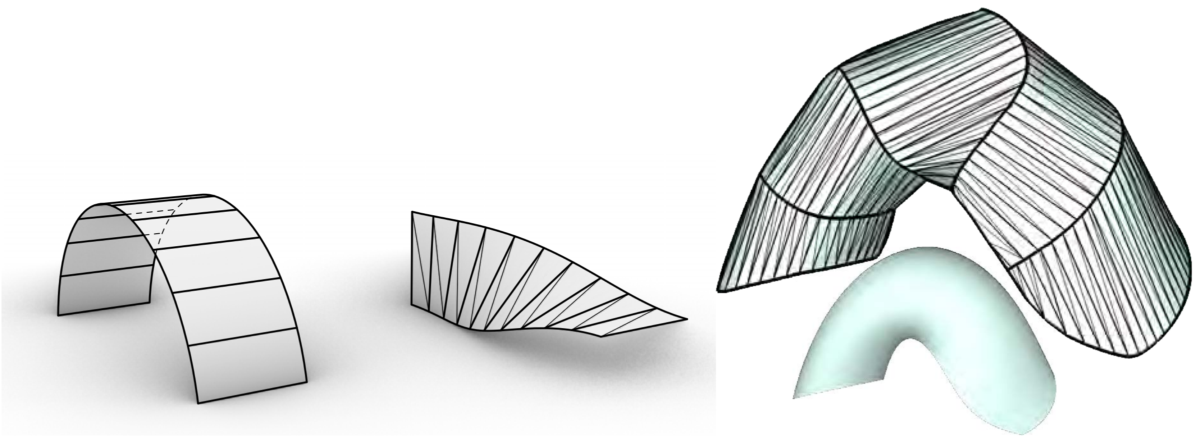


Figure 1.3: A ruled surface (left) cannot feature torsion without compromising its developability. However, even a non-developable ruled surface featuring torsion becomes developable when discretized as a triangle strip (middle), making them better suited to approximate general shapes (right).

In this work, we take a different route. Our key observation is that the discretization of the problem inherently changes the developability constraints. Specifically, any ruled surface, even a non-developable one in the continuous setting, can be trivially discretized into a developable surface by, first, electing a set of rules along the ruled surface, which splits the ruled surface into a set of (non-necessarily flat) rectangular shapes, then splitting each rectangle diagonally, adding new rules. This produces a triangle strip that, as such, is trivially developable by construction because all triangles are flat, and all vertices are on the boundary. This final discretization introduces minimal geometric approximation and, crucially, preserves the key characteristics of the ruled surfaces (of being straight along the rule directions) and, therefore, all their benefits mentioned above (such as realizability using straight beams). By relying on this final discretization step, our construction process can disregard the developability constraints, and this additional freedom potentially results in better decompositions (see Figure 1.3). In other words, the sought objective of using *long* ruling lines doubles as an easy route to ensure the developability of the patches.

Through iterative refinement and optimization, our method promises to redefine the boundaries of digital fabrication, empowering designers to realize their creative visions with precision and efficiency. As we traverse this interdisciplinary terrain, the fusion of computational design with biofabrication and digital fabrication heralds a new era of innovation and discovery. By pushing the boundaries of what is possible at the nexus of biology, engineering, and design, our work seeks to catalyze transformative advancements that will shape the future of medicine, manufacturing, and beyond.

Therefore, the thesis has two application cases. The first case addresses the application of computational design in biofabrication by conducting a physically-based simulation of the spheroids fusion process. The second case addresses the application of computational design in digital fabrication by introducing a new method for decomposing a mesh into developable patches.

The remainder of the thesis is structured as follows: Section 2 reviews the literature for two cases. The section 3 outlines our methodology for addressing the problems, and section 4 provides the detailed implementations of the proposed models. We will present the results of the models in section 5 and their validation in section section 6, the sensitivity and morphological analysis in section **??**, and the discussion in section 7. Finally, we will report the conclusion of the thesis at the end.

CHAPTER 2

LITERATURE REVIEW

As mentioned in the introduction section, the thesis will address two applications of computational design in digital and bio-fabrication. In order to provide a literature review, we have categorised the literature into two subsections. Section 2.1 reviews the recent research done in the area of bio fabrication of heart tissues and specifically spheroids bioprinting. In the section 2.2, the recent advancement of digital fabrication in the area of surface developability and its applications are discussed.

2.1 Related work to the physically-based simulation of cardiac spheroids fusion

In this section, we will present the most recent research that has been done in the area of bioprinting, stem cell modelling, in-silico simulation of spheroids, and spheroids fusion.

2.1.1 3D bioprinting

The 3D bioprinting process is very similar to usual 3D printing, also known as additive manufacturing technology, in which the material is deposited layer by layer to make a final product. However, in bio-printing, instead of metal, polymer or ceramic, a special biomaterial is used that is called bio-ink [78].

Similar to manufacturing 3D printers, there are five significant bioprinting techniques, including extrusion, inkjet, laser-assistant and stereolithography [23].

In extrusion-based bioprinting, biomaterials that should be in a viscous liquid, also known as bioink, are deposited using air pressure, piston force or screw rotation. In inkjet bioprinting, an element heats the biomaterials to make them watery and pass them from the nozzle. Then biomaterials will be dropped from the nozzle to the plate. There are other kinds of inkjet bioprinters in which, instead of using thermal elements, a piezoelectric head is utilized.

In the stereolithography method, controllable light is gained from the light-sensitive polymers to make photo-polymerization precisely using a beam projector.

Laser-assistant bioprinting is another technique in which a laser beam pulses to the biomaterial and deposits it on the substrate. This is a nozzle-free 3D printing technique, as there is no need for the nozzle to guide biomaterials. They will be moved according to the laser direction.

Most researchers have used the extrusion method in the literature because it has advantages, including easy usage and the capability of printing various biomaterials, making it more convenient. While the inkjet technique is unable to print continuous flow, stereolithography and laser-based techniques may damage cells and biomaterials due to using light and heat [23].

Derakhshanfar et al. (2018) [23] have divided the challenges of 3D bio-printing into two general categories: i) the challenges of making bio-materials and ii) the challenges of integrating these bio-materials in vivo conditions. The challenge of making bio-materials relates to the printer's technical issues such as technical defects that may occur while printing, such as blockage of the machine nozzle or other factors such as temperature, nozzle angle, geometric and mechanical properties of printed materials.

To integrate biomaterials in conditions in vivo, it is necessary that all the geometric and mechanical properties of the materials be stable and without changing during and after the transplanting. Geometric properties such as the three-dimensional structure and shape of the desired tissue and mechanical properties such as elasticity and plasticity, the viscosity of biomaterials, and cell support of scaffolds must remain stable during integration and after transplantation to provide the desired results biological performance.

From the beginning of the invention of bio-3D printing technology, tissues such as bone cartilage and skin were the first tissues to use 3D bioprinting. Later, scientists are looking to use this technology to produce tissue from other organs in the body, such as the heart. This project addresses some of the challenges in creating heart tissue through bio-3D printing technology.

Most tissues produced by 3D bioprinting technology have simple geometric structures consisting of one or two cell types that lack a vascular system. One of the main challenges of 3D bioprinting is to create a circulatory and vascular system in the tissue and the use of different types of cells with different biological functions within the tissue that can have far more complex geometric structures.

Another challenge for 3D bioprinting technology is the type of bioink to be used. In general, there are natural and synthetic bioinks: synthetic bioinks have the ability to create better mechanical and geometric properties. In contrast, natural bioinks can perform better in biological functions and cell nutrition. How to combine these two bioinks to create a proper bioink to produce a good product is a significant challenge in the 3D bioprinting area of research [74].

The type of cell used in printing is essential. Since the human body reacts differently to the various materials entered or the organ transplanted to it, and its immune system is susceptible, using cells that do not elicit immune responses is a major challenge in using 3D bioprinters. To solve this problem, scientists have tried different cell types and the most important of which is the use of induced pluripotent stem cells [125].

Another major challenge in tissue engineering with 3D bioprinters is tissue production in “ex vivo” or outside of the body. Because the conditions inside and outside of the body can be very different and the production and engineering of tissue outside the body and transferring it into the body can lead to physical, biological and mechanical changes in the produced tissue. Therefore, simulating a suitable and adaptive environment close to the conditions inside the human body is very important to maintain the tissue’s phenotype and biological and mechanical properties. One of the available solutions is to create a three-dimensional environment based on an extracellular matrix to support cell aggregation and create tissue in conditions that can keep the biological and mechanical properties of tissues [22].

Another fundamental challenge of creating tissue with 3D bioprinters is vascularization. Because the tissues produced in vitro require oxygen and food, and this oxygen and food must be delivered to them by the vascular system, vascularization is vital in tissue engineering by 3D printers. This need is critical in creating tissues with more muscle and thickness as these tissues need more oxygen and food than other parts of the body. On the other hand, the transplantation of these tissues created in vivo into in-vitro conditions requires their delicate fabrication and their connection with the body’s neural networks [74].

In the literature, two main techniques have been proposed for this problem. The first technique is to create tiny channels to improve the passage of nutrients and oxygen into the tissue. These tiny channels will play the same role as vascular in the tissue ([48], [44]). The second is the publishing of cells in the tissue to create the proper vascular to nourish and oxygenate the tissue. This publication requires a specific pattern that must be considered ([94]).

2.1.2 Stem cell modelling

Human pluripotent stem cells have significant advantages that can be used in regenerative medicine, drug design and producing human body tissues. Mathematical modelling of stem cells can help us to understand their behaviour and develop non-invasive treatments. Human pluripotent stem cells (hPSCs) can renew their selves by repeated divisions. Usually, they derive from the early human embryo.

The application of mathematical modelling of stem cells can be categorised into three parts. First, the models address cell kinematics, the second models address colony growth, and the third models address cell pluripotency.

Cell kinematics refers to the movement of a cell concerning another and within the colony. Colony growth refers to cell proliferation, and cell pluripotency refers to a cell's characteristic that helps it to convert to any type of human body specialised cells [134].

Our project mostly relies on cell kinematics, and the two other ones are out of our scope. Cell kinematics also can deal with isolated cells (or pairs) and colony kinematics. The simplest way of modelling isolated cells' motion is a random walk [21], [17]. This random walk can be biased by an external force that can move cells in a particular direction. This type of random walk is called correlated random walk (CRW). When cells move in one direction, these cells tend to keep going in the same direction. This tendency would be presented in the correlation coefficient [38, 99].

Stem cells also can migrate as a larger group called a colony. In order to consist of tissue, cells should move within the colony [70]. Some agent-based models were developed to capture the movement of the cells in colonies [80]. However, there are serious challenges that remain, especially in describing the 3D behaviour of the cells. Some other types of agent-based models consider a continuous movement for cells in any direction [2, 129].

Other models mainly address the morphology of the cells, including cells' random formation, elongation and retraction of pseudopodia that make cells move [40].

Some other researchers used image processing techniques and video tracking to capture the cells' motion [133]. There are some models for 3D simulation of the cells with open-source software such as Physi-cell [34].

Another aspect of kinematic cells addresses the cell colony's shape. Cells colony refers to a larger number of cells when they move with each other. Usually, agent-based models are used to monitor the motion of the cells' colony [80].

According to Wadkin et al. (2017) [133], the single stem cell movement has a random walk pattern. However, stem cells have more systematic movement when there is more than one. As the number of cells increases, they have more cross-talk so that a band of them form the outer edge of the colony.

Potdat et al. (2010) [99] showed that epithelial cell migration paths on 2D plastic substrates could be modelled as a bimodal correlated random walk model (BCRW) when there are no chemo-attractant gradients. BCRW is a kind of correlated random walk that has two types of search. A fast search without considering any targets and a slow search to locate the target. Their research considered "flight" as the portion of cell path and step length as the distance the cell travels over a remarkable period (here it is every 30 seconds). Then, they considered flights to follow an exponential distribution and more step lengths are correlated through turn angles while moving step lengths within the flight have an exponential distribution.

Using biomodal analysis, they segregated the cellular trajectories of individual mammalian cells into directional and re-orientation. So, every 30 seconds, a parameter shows instantaneous direction change compared to a pre-calculated threshold to find whether the cell is going in directional mode or changing direction (re-orientation).

Nguyen et al. (2019) [80] has addressed human-induced pluripotency stem cell (hiPSC) homogeneity using a kinematic model. They used computational modelling to understand the cause of a higher cell movement rate at the peripheral region of the colony. In this paper, they analysed the cells' movement in the colony using a discrete stochastic model. Considering cell migration, division, and quiescence, they described cell connection by using energy between the cells. Cell displacement was used to analyse the movement rates of a cell inside a colony. They have performed a simulation to catch the heterogeneity of cells' velocity in colonies.

Ghaffarizadeh et al. (2018)[34] have developed a physics-based simulator for multicellular systems to study how cells move, grow, interact, divide and die based on 3D simulation. This software tries to provide a virtual laboratory for simulating the tissue microenvironment and dynamic behaviour of the interacting cells. In this paper, an agent-based model has been used, and the agent cell has various properties such as position, volume, motility, radius, cycling and death. They successfully tested the software for simulating tumour spheroids.

Adra et al. (2010) [2] have used agent-based modelling to describe and simulate the biological cells' behaviour. Their model explores the colony formation of normal human keratinocytes (NHK). Their computational modelling consists of a multiscale model of the human epidermis and a 3D dimensional simulation. In multiscale modelling, they have considered sub-cellular, cellular and multicellular rules that affect cells' proliferation, differentiation and migration.

Codling et al. (2008) [17] have reviewed the mathematical application of the random walk in a biology context. According to Codling et al. (2008) [17], the movement of biological organisms can be completely random without any bias to a unique target or direction or can be biased towards a particular direction,Äîthe latter case is called biased random walk (BRW). If there is a tendency or persistence to move forward in the same path the walker had been in, then the random walk is correlated and called (CRW). From an application point of view, random walk is usually applied to the movement and dispersal of biological organisms or chemotaxis models of cell movement. In a simple isotropic random walk, the probability that a walker will be at the particular distance ($m\delta$) from the origin after specific time steps (n) can be calculated by a binomial distribution. For large n, this distribution leads to normal distribution based on the central limit theory.

Advances in microscopy imaging have provided the opportunity for developing more dedicated mathematical and statistical models on biological cells and mainly stem cells, e.g. imaging of bone marrow ([1]) and live video cell microscopy ([118]). Hence, De et al. (2019) [21] have provided a framework for quantitative models for cells' spatial distribution, motility, shape and proliferation.

This framework includes image segmentation as a first step in separating cells from the other structures in the image. The second step is to study the cells' distribution in the image. For this purpose, cells are considered as point particles in the image and described by their coordination. Therefore, it is possible to describe their density by the intensity test cells' homogeneity and quantify spatial correlation.

Cells' motility also can be quantified by calculating cell displacement during the time. So, the cells' trajectory, velocity and other important factors can be studied in this context. In order to model a cell's shape, after segmenting the cell from the image's background, the area occupied by the cell can be approximated by well-known geometrical shapes such as circle, ellipse and convex hull. This can be done by measuring the number of pixels covered by the cell in the image.

Some studies only concern individual cell behaviour, like specific properties. In this point of view, a more delicate mathematical model is required to describe individual behaviour. Since we focus on the specific cell in this attitude, these types of methods are called "cell-based" or "agent-based" methods.

Generally, agent-based mathematical models are either continuous or discrete. Continued models mostly rely on a differential equation. However, discrete models mostly rely on stochastic processes. De et al. (2019) [21] have explained two famous agent-based methods called *centre-based method* and *cellular Potts model*. In the centre-based method, cells are considered a circle in 2D or a sphere in 3D, and the models are developed based on the centre of the cells in a continuous space. All related factors, such as forces and displacement of the cells, are calculated based on the cells' centres. In contrast, the cellular Potts model considers a discrete space and tries to measure the cell's shape's area regarding covered lattice sites. Because the scaffold allows to us to measure the mechanical properties of the cells.

Alt et al. (2017) [3] have reviewed Vertex models that address tissue mechanics. These types of models can explain tissue morphogenesis and cell deformation. The tissue morphogenesis is affected by the mechanical properties of the cells. Behind the scenes is physics that causes a change in forces and cell deformation. Therefore, vertex models try to represent a geometrical tissue model by a set of vertices and lines between them to simulate the deformation of the cells caused by mechanical actions. Alt et al. (2017) [3] classified vertex models into four groups: i) 2D apical vertex models that describe the apical cell surface as polygons with a straight interface between neighbouring cells. ii) 3D apical vertex models that are the extensions of the 2D version. iii) A 2D lateral vertex model describes the lateral part of an epithelium. In these models, cells would be represented by four edges. Two edges represent apical, and the other two edges represent lateral interfaces. iv) 3D lateral vertex models that are the extension of 2D lateral vertex models.

In order to describe cells' motion, the physical forces are calculated. Then, the dynamical and topological aspects of the cells can be modelled by adding a time scale and minimal length of dividing to the cells, respectively.

Osborne et al. (2017) [87] compared five individual-based approaches for modelling multicellular tissues. These five models are based on either on or off-lattice space. The simplest model is cellular automata (CA), in which each cell can be placed in a single lattice site. The second model class is cellular Potts (CP), in which each cell can be placed on several lattice sites. So, it is more appropriate to model the mechanical aspect of the cells.

As it is clear from their approach, CA and CP are on-lattice. Some off-lattice models, such as overlapping sphere (OS) and Voronoi tessellations (VT), also exist. In these approaches, cells are considered a sphere (in OS) or polygon (in VT) and tracked by their centres. Other off-lattice methods that Osborne et al. (2017) [87] have addressed are Vertex models have already been explained.

On the other hand, in contrast to the classical modelling of stem cells based on manual identification and manipulation of cells' morphology, researchers have recently tried to use data-driven models based on observed data to increase the efficiency of their results. Thanks to advanced bio-imaging facilities such as modern microscopes that provide valuable data from bio-processes, these data-driven models are more accessible. The data-driven models are mostly based on statistical (machine) learning methods. We will review the applications of this approach in a separate section.

2.1.3 Spheroid fusion

Increasing advancements in biomedical engineering and additive manufacturing led to the biofabrication of human tissues from cells [31, 66, 148]. In the body, cells grow, replicate and organize in complex tissues and organs to generate the human body [67, 106]. Recent research in biofabrication has enabled us to 3D bioprint cells in hydrogels to replicate human morphogenesis, potentially using them as replacement body parts for patients [27, 33, 68, 95, 96, 115, 132]. The capacity of 3D bioprinters to build tissues by depositing cellular material in specific locations has provided invaluable assistance in producing highly complex and diverse shapes and sizes, including the design of the bioink used in this process [23, 36, 37, 74, 108].

A common approach to promote tissue morphogenesis using cells is based on their ability to organize in spheroid cultures. Spheroids are aggregates of cells that can be used to model tissues and organs. For example, our team has developed cardiac spheroids to mimic the complex human heart microenvironment for in-vitro and in-vivo applications [95, 97, 109]. Spheroids can be used as building blocks to create complex tissues when 3D bioprinted in tissue-tailored hydrogels due to their ability to fuse with each other [14, 27, 28]. They are embedded in biomaterials known as hydrogels [50].

Kim et al. [50] found that the spatial arrangement of the spheroids within the hydrogels played a significant role in regulating spheroid fusion. Specifically, they observed that spheroids closely arranged in a hexagonal pattern showed a higher degree of fusion than those randomly distributed within the hydrogel. They attributed this to the increased contact area between the spheroids in the hexagonal pattern, which promoted fusion. Hence, the hydrogel's properties play a major role in spheroid fusion and can determine the size and shape of the bioprinted construct. Hydrogels can be made from different materials with different physical and chemical properties, affecting the final shape [29, 56].

Central to our inquiry is the simulation of elastic-plastic fusion in 3D bioprinted spheroids, a critical step in tissue engineering processes. By developing a sophisticated simulation framework grounded in continuum mechanics and smoothed particle hydrodynamics (SPH), we aim to decipher the nuanced dynamics of spheroid fusion and glean insights into optimizing tissue engineering strategies [13, 34, 35, 104, 134]. Through rigorous experimentation and computational modelling, we seek to illuminate the underlying biophysical mechanisms governing spheroid behaviour and fusion dynamics, offering novel avenues for enhancing tissue engineering outcomes.

2.1.4 In-silico simulation

Over the last two decades, the games and entertainment industries have extensively used physically-based simulation [77] to simulate rigid [6, 73] or deformable objects [7, 20, 71, 77, 142] and also render realistic fluids [19, 61, 64, 72, 79, 107, 126, 126].

As we mentioned, most of these applications have been used in the game and entertainment industries. Therefore, the idea of using physically-based simulation in bioengineering is almost neglected. Some research is addressing in-silico modelling of the spheroids in tumours [4, 5, 52, 93]. Nevertheless, they have focused on mathematical models and numerical analysis of the important parameters. Obviously, numerical analysis is not the best solution for those who are interested in shape analysis or the geometry of the spheroids.

Therefore, the aim of this study is to bring the idea of physically-based simulation into the spheroids fusion process in the bio-fabrication field. The advantage of our proposed method, compared to other related research in the literature, is that it provides a generic framework for not only numerical but also graphical simulation of spheroids' fusion. This will help scientists and biologists who are interested in studying geometry or the shape of the spheroids during their fusion. As we mentioned in the introduction, the geometry and the final shape of the spheroids play a crucial role in fabricating the heart tissues. The final tissue must be in the desired shape and geometry as it is supposed to be implanted in the patient's body to replace the damaged tissues.

2.2 Related work to surface developability

In this section, we start by previewing developability in manufacturing. Next, we summarize recent advancements in the automated generation of developable meshes. Finally, we provide an overview of patch decomposition.

2.2.1 Developability in Manufacturing

Developable surfaces have a long manufacturing tradition. Their applications range from individual components [54], through garments [110, 136], to architecture [46, 62, 137]. One of the main benefits of developable surfaces is that they allow for fast and cost-effective fabrication. Through the use of numerically controlled cutters or prefabricated forms a developable surface can be quickly manufactured from a wide range of materials [45, 51, 112, 127, 145].

Besides admitting a flat configuration, developable surfaces are also locally ruled with constant normals along individual ruling lines [85]. This property can be leveraged in other subtractive manufacturing techniques, such as flank milling or hot-wire cutting, that rely on aligning the fabrication tool with the surface [39].

Once the individual flat pieces are manufactured, they need to be assembled to form the final three-dimensional shape. There are two standard techniques: folding and joining. Folding relies on plastic deformation of bulk material [49, 82, 124]. It is a technique often used in origami [25, 65, 69, 117] and recently even to manufacture functional three-dimensional objects [30]. As an assembly process, folding has two key disadvantages. Since each element needs to be bent separately, there is only a limited number of segments that can be practically assembled. Moreover, as the segment size grows, folding becomes significantly more challenging. In contrast, joining two shapes, even at the architectural scale, is a comparatively simpler process [32, 46, 137]. However, for practical purposes, the patch size plays a critical role in joined assemblies. A larger patch size is tempting as it reduces the overall number of joints. However, from a certain size and scale, large patches become unwieldy. In our method, we take both of these considerations into account. We optimize for a small number of patches to facilitate the assembly. However, we also enforce that each patch bends only along its ruling lines which facilitates fabrication. For more information about developability and its applications towards manufacturing, we suggest the survey of [144].

2.2.2 Developing Discrete Meshes

The study of developability is one of the traditional topics with many generative algorithms proposed [11, 100–102, 128]. However, for fabrication, a more interesting direction is the search for the closest developable approximation of an input discrete mesh. Methods in this category range from convex optimizations on constrained inputs [113], through wrapping the object with developable sheets [41], segmenting the input into developable regions [146], to genetic algorithms optimizing patch decompositions [147]. The main drawback of these methods is that they do not consider the fabricability of the final patch decomposition. More specifically, these methods only enforce flatenability of the generated patches. However, take as an example a crumpled piece of paper. While trivially flatenable, it would be very challenging to exactly reproduce [121]. This observation led to the development of methods that enhance the flatenability criterion with a constraint on the ruling lines of the surface [10, 121]. Our method also belongs to this category and aims to produce patches with non-intersecting, uniformly spaced ruling lines. The key difference is that the prior works do not provide patch decompositions. Their output is triangular meshes composed of developable regions. Generating a fabricable patch decomposition is highly non-trivial [41]. In contrast, our method is built on generating patches from the ground up. As a result, our outputs are ready to fabricate two-dimensional blueprints.

The study of developability is one of the traditional topics in mathematics with many generative algorithms proposed [11, 100–102, 128]. However, for fabrication, a more interesting direction is the search for the closest developable approximation of an input discrete mesh. [121] proposed one of the first automated methods. In their work, the authors argue that to generate practical decompositions, the number of folds needs to be minimized. This is achieved by optimizing the smoothness of ruling lines on the surface. In our work, we adopt a similar approach. To generate a fabricable decomposition, we seek patches with uniform ruling lines. The main difference is that [121] does not include patch decomposition in the optimization, which then requires manual tuning to avoid overlaps. In contrast, our method jointly optimizes the number of patches and their ruling lines, generating intersection-free decompositions automatically.

Generating developable approximations for arbitrary input meshes is a challenging task [47, 54]. To relax the problem [113] proposed constraining the inputs to height-fields. This allows us to formulate the search for developable approximations as an efficient convex program. Unfortunately, there is no direct path to extend the method for arbitrary three-dimensional objects. In contrast, our method works on a full range of inputs from heightfields, through manifold meshes, to shells.

Another option to approximate a surface with developable patches is to iteratively wrap it with non-stretching sheets [41]. To join the individual sheets, the initial mesh is non-linearly deformed to conform to the wrapping material. Unfortunately, the process can still leave uncovered vertices, leading to gaps in the final developable approximation. In contrast, our method explicitly constrains the boundaries to guarantee a watertight solution.

[10] propose another direction for generating developable approximations. Their key observation is that a developable mesh has a thin Gaussian image. They propose a global-local solution that iteratively thins the Gaussian images and provides fully automated approximations. Our work also relies on a global-local step to ensure developability and generate patch decomposition. The main difference is that [10] does not provide the patch decomposition. In contrast, our algorithm generates a manufacturable patch decomposition.

An alternative to decomposition is the option to segment the input mesh into developable regions [146]. The key to finding the decomposition is to start with a nearly developable approximation. The approximation is then further refined with developability-inspired energy, which allows splitting the mesh into developable segments. The cost of the complex optimization is an increased runtime of over one hour per model. In contrast, our method produces developable patch decompositions in order of seconds.

[147] adapt a genetic algorithm to iteratively evolve an input mesh into its developable version. This is achieved by redefining the mutations into mesh space to manipulate the boundaries, splitting, and merging patches. Unfortunately, the generated patches are not guaranteed to be free of self-intersections. While the patches could be further cut to avoid these intersections the cuts might introduce undesired stretch, which lowers the developability of the patches. In contrast, our method is guaranteed to produce patches free of self-intersections.

2.2.3 Patch Decomposition

Flattening a three-dimensional model into a plane is a non-trivial task. Typically, the flattening operation produces large distortions [143] and self-intersections [103] in the 2D domain. To reduce the distortion and eliminate the intersections, it is possible to cut the input mesh into smaller pieces [57, 98, 116]. State-of-the-art techniques rely on vector fields defined on the surface to align the cuts with mesh features [63, 83, 89–91, 105, 130]. In this work we build on [90]. We adapt the method from quadrangles to arbitrary developable patches. For more details about patch decompositions, we refer the reader to the survey of [15].

Chapter 3

Methodology

In this section, we aim to explain the details of the methodologies we have developed to resolve the two cases we discussed in the introduction section. For the bio fabrication case, we came up with the idea to develop two mechanical models that can describe the fusion process of the spheroids and then numerically solve them using the SPH method.

For the digital fabrication case, we developed a geometry processing pipeline based on ruling lines in the surfaces to make developable patches on a mesh. The detail of the bio fabrication case methodology is explained in section 3.1, and our methodology for making developable patches based on ruling lines is explained in section 3.2.

3.1 Physically-based simulation of fusion of cardiac spheroids

To achieve our goals of simulating the cardiac spheroids fusion, we designed a method to simulate the fusion of spheroids after printing physically. Spheroid fusion is a biologically complex process involving many factors, some of which are not directly observable. First, we must know the appropriate mathematical model to describe their physical behaviours. As we specified in the introduction, we first implemented an elastic-plastic solid model based on Muller et al. [72], and then we also developed an elastic-plastic fluid-based model with different fluid parameters. In the following sections,

we will provide the mathematical description of the solid (Section 3.1.1) and fluid

(Section 3.1.2) models and their implementation (chapter 4).

3.1.1 Solid-based model

In this section, we assume that the spheroids are elastic-plastic solid materials. To describe their fusion process, we need to develop an elastic-plastic model that can explain the elasticity and plasticity of the spheroids during the fusion process. To better understand the dynamic behaviour of this system, we will use a continuum mechanics model based on Muller et al. [72] in which an object's elastic and plastic behaviour can be simulated using the SPH method. The main idea of the model is to use a Green-Saint-Venant strain tensor for each particle:

$$\varepsilon = \mathbf{J}^T\mathbf{J} - \mathbf{I} = \nabla\mathbf{u} + \nabla\mathbf{u}^T + \nabla\mathbf{u}\nabla\mathbf{u}^T \tag{3.1}$$

The equation 3.1 uses various matrices, such as $\mathbf{u}$ which represents a continuous displacement vector field, $\mathbf{I}$ is the identity matrix, and $\mathbf{J}$ is the Jacobian matrix of the deformed model. The term ε encompasses both elastic and plastic strain ($\varepsilon = \varepsilon^e + \varepsilon^p$). To extract the plastic strain ε^p from the total strain ε, we need to separate it from the elastic deformation strain ε^e. Initially, we assume plastic strain is zero, and the algorithm proposed by O'Brien et al. [84] is employed to update the plastic strain value (See Algorithm 2). Consequently, stress can be computed using Hooke's law:

$$\sigma = \mathbf{C}\varepsilon \tag{3.2}$$

where $\mathbf{C}$ is a rank four tensor, approximating the constitutive law of the material. After some mathematical manipulation, particle i's elastic force is equal to:

$$\mathbf{f_{ie}} = -\sigma\nabla\mathbf{u}_i\varepsilon \tag{3.3}$$

Green's strain tensor in equation 3.1 will be zero for the volume inverting displacement field. Therefore, there will be no restoring volume inversion force. For this reason, a volume conservation force is needed to penalize deviations of the determinant of the Jacobian from the positive side. Hence, the volume energy term will be:

$$U_v = \frac{1}{2}k_v(|\mathbf{J}| - 1)^2 \tag{3.4}$$

where k_v is the volume constant. Consequently, the volume conservation force is equal to:

$$\mathbf{f_{iv}} = -\sigma\nabla\mathbf{u}_i U_v = -k_v(|\mathbf{J}|-1)\sigma\nabla\mathbf{u}_i\,|\mathbf{J}| \tag{3.5}$$

Therefore, the total internal force applied to the ith particle is equal to the sum of elastic and conservation forces:

$$F_i = f_{ie} + f_{iv} \tag{3.6}$$

The moving least squares method has been used to solve differential displacement equations to obtain the Jacobian matrix. In the next section, we will describe the details of our developed elastic-plastic fluid model.

3.1.2 Fluid-based model

The Navier-Stokes equation generally describes fluid dynamics. Navier and Stokes formulated this fluid mechanics equation, the basis for essentially all fluid mechanical works today. This paper will use the incompressible Navier-Stokes equation that governs fluids like water at normal velocities and temperatures. Other forms of these equations are compressible, generally used for supersonic and hypersonic phenomena, which can have shockwaves and discontinuities in the solution. However, such phenomena are not allowed in incompressible equations. It is worth mentioning that all the everyday intuitions we have about how fluids work correspond to the behaviour of incompressible equations. Hence, the incompressible equations are the ones we consider for physically-based simulation. The general form of the Navier-Stocks equation is as follows:

$$\begin{aligned} \rho[\frac{\partial v}{\partial t} + v.\nabla v] &= \rho g - \nabla p + \mu\nabla^2 v + F_{elastic-plastic} \\ \rho(\nabla \cdot v) &= 0 \end{aligned} \tag{3.7}$$

where ρ is density, p is pressure, g is gravity and v is velocity. $F_{elastic-plastic}$ is computed the same as in the solid-based model. Also, $v.\nabla v$ is convective acceleration. We can use the simplified equation of state to calculate the pressure, denoted by p [60]. Thus we have:

$$p = k(\rho - \rho_0) \tag{3.8}$$

Where ρ_0 is the resting density, pressure and density are scalar variables, while gravity and velocity are vectors. k is the gas constant that depends on the temperature. From the right-hand side of the equation 3.7, we can find that the fluid moves under the force of gravity, pressure, and viscosity. We have convective acceleration from the left-hand side of the motion equation. We know that the gradient is the higher-dimension analog of slope in one dimension. Hence, the gradient of p is given by the partial derivatives of p with respect to x, y and z. It is important to note that the equation of motion is governed by a negative pressure gradient, indicating that the system moves in a downward direction along the pressure gradient. This implies that the system transitions from regions of higher pressure to regions of lower pressure according to its natural equations of motion. The pressure, denoted as p, is defined as the difference between the fluid density at a specific point and a reference value known as the "resting density."

The operator ∇^2 facilitates the diffusion of velocity, resulting in the dispersion of momentum across the system. In essence, this process leads the system to a state of uniform velocity throughout, driven by its inherent forces.

Also, there is a mass continuity equation in equation 3.7. In the case of the incompressible equations, it says that mass will be neither created nor destroyed.

In this section, we provided the methodology of our proposed simulation framework to simulate the cardiac spheroids fusion. The details of the implementation of our approaches have been explained in the section 4.1. In the next section, we will explain our methodology for making a developable mesh using patch decomposition and ruling lines.

3.2 Developable mesh decompositions based on ruling lines

In this section, we will explain our methodology for making developable mesh. We seek to decompose the input target surface into a small set of *patches* that can be well approximated by ruled surfaces. Patches are sections of surfaces delimited by a series of nearly orthogonal corners and mostly straight sides, resembling curved polygons. While they commonly connect along shared seams, their density across the surface varies, incorporating T-junctions. To govern the simplicity of the pattern layout, users have the capability to fine-tune the complexity of the generated patches by adjusting the maximum number of corners for each patch and enabling the use of T-junctions in the final patch decomposition (see Figure 3.2). The system is governed by a user-defined upper bound on the approximation error e_{max}, which serves as a parameter to balance between the simplicity of the design (in terms of the number of the ruled surfaces) and the geometric adherence with the target shape. Figure 3.1 illustrates the effects of this parameter. This feature is deemed crucial, as it ensures the adaptability of our method across a wide range of applications, extending from the manufacturing of small objects to architectural-scale projects.

Then, we produce a developable ruled surface for each patch; finally, each patch is flattened isometrically into a 2D region, with ruled lines mapped over straight segments traversing the region from side to side. The set of the 2D regions serves as a blueprint that can be used to physically fabricate the input shape (possibly assisted by automatic cutting and guided by printed or pre-folded ruling lines). sections of surfaces delimited by nearly orthogonal corners and mostly straight sides, resembling curved polygons.

Variant: T-junctions

Our patch layout can be opted to feature occasional T-junctions, which generally reduces the number of patches; otherwise, a strictly conforming layout can be requested. Figure 3.2 shows a comparison of this choice.

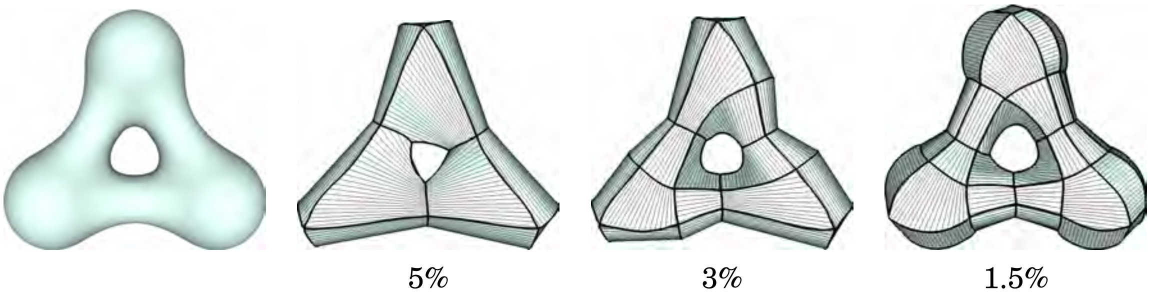


Figure 3.1: Results obtained with different error thresholds e_{max} (expressed as a fraction of the bounding-box diagonal of the original mesh).

Variant: symmetry preservation

If the input surface exhibits a reflection symmetry, we can enforce its preservation by splitting the surface along the symmetry plane, executing our method on one side, and mirroring the results. Figure 3.3 shows a comparison of this choice.

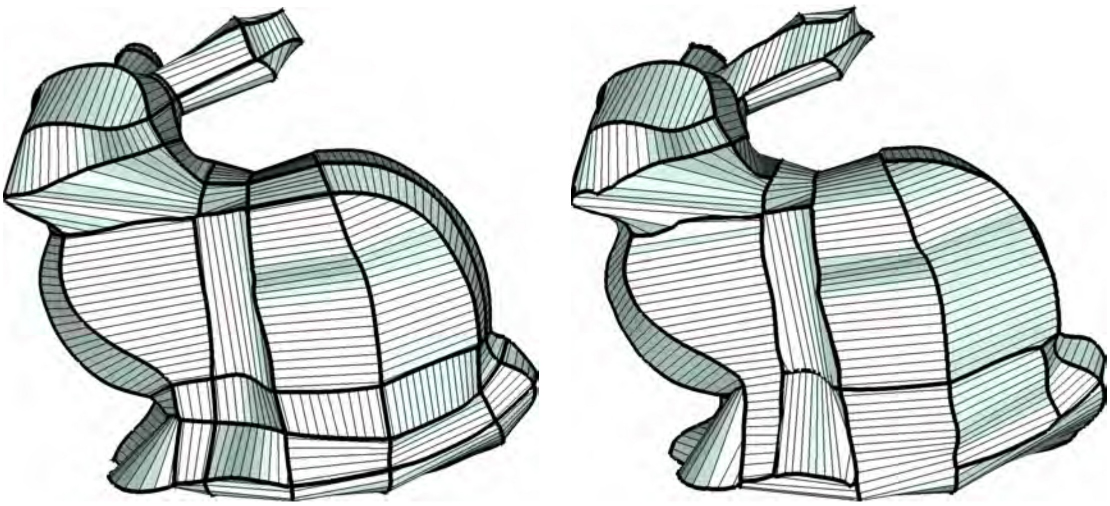


Figure 3.2: An example patch decomposition without (left) and with (right) T-junctions.

3.2.1 Objectives

For a solution to serve as a fabricable design, it must meet several *strict* requirements:

Bijective developability In order to be realizable by cutting and bending a flat sheet of material, patches must admit a fully *isometric* and *bijective* mapping into a 2D region. Bijectivity implies that the 2D image of the patch must be free from overlaps.

Straight Ruling lines The ruling lines of each patch must be perfectly straight (both in 2D and 3D).

Water-tightness Neighboring patches must meet exactly at their boundary, without gaps.

In addition, a good solution must strike a good trade-off between two conflicting objectives (controlled by the parameter e_{max}):

Design simplicity To ease fabricability, we want the number of patches to be limited and the shapes to be simple.

Geometric fidelity The patch layout must approximate the target shape, within the predefined error tolerance.

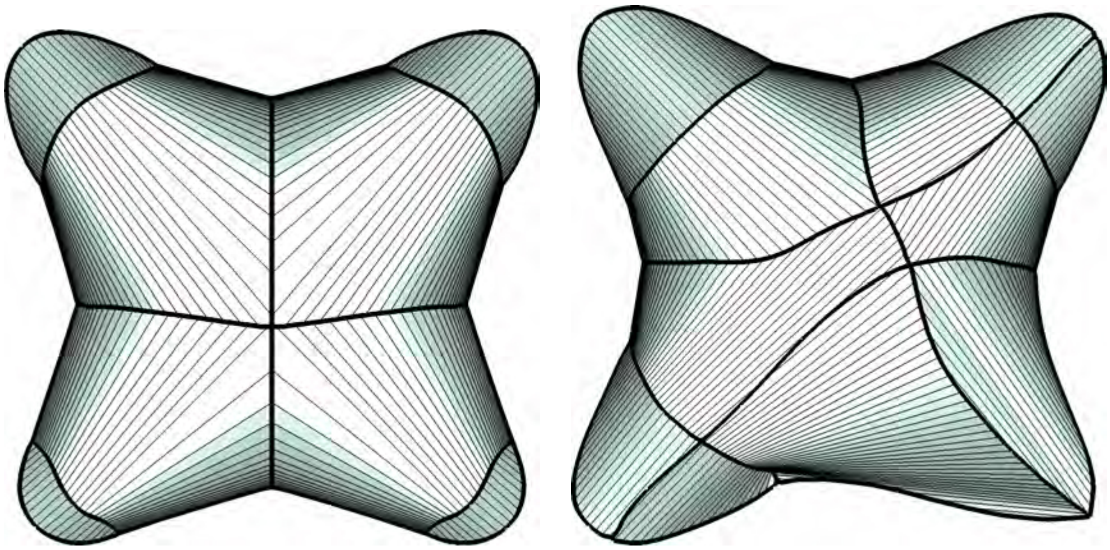


Figure 3.3: An example of results with (left) and without (right) symmetry preservation.

About design simplicity

To maintain practicality in the cut-and-assembly process, seams should be as smooth as possible. We favour a uniform distribution of seams over the 3D mesh surface to prevent clustering or the presence of small patches where not explicitly needed for the sake of geometric approximation. Additionally, we aim for patches with a limited number of corners (typically between 5 and 7), and their angles should be approximately orthogonal.

As a rule of thumb, the complexity of the realization grows with the number and length of the boundaries, which will require cutting (in 2D) and gluing (in 3D). Therefore, we want to avoid unnecessary patches or small patches, whenever not necessary to approximate the shape.

For the same reason, the boundaries of the patches should be preferably smooth and short. In our decomposition, the 2D patches are polygon-like shapes delimited by a small set of smooth curved lines. Additionally, ideal boundary lines are either approximately aligned to the rules of the patch or approximately orthogonal to them. The boundary lines tend to be smooth and to meet at approximately squared angles.

Finally, we need patches to have a **disk-topology**, that is, to feature a single boundary; this disallows, for example, circular patches, or holed patches, which would hinder or considerably complicate most construction processes.

About geometric fidelity

The objective of geometric fidelity implies some degree of **curvature alignment**. By their nature, ruled surfaces can freely curve only in the direction orthogonal to the ruled lines. Therefore, it is beneficial for rules to be aligned to minimal curvature directions, so as to be free to bend along the maximal curvature direction. To this end, we employ a curvature-oriented cross-field to guide our construction process. Additionally, our framework facilitates symmetric patch decomposition, enhancing the overall regularity and soundness of the produced layout.

3.2.2 Overview

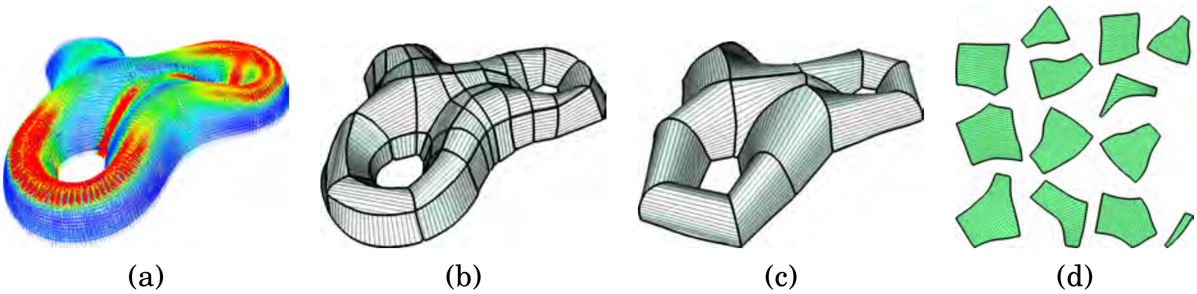


Figure 3.4: Our processing pipeline unfolds as follows: (a) The input mesh, enriched with a curvature-aligned cross-field (red colour indicates areas where the field guides the smoothing process); (b) The result of the patch densification; (c) The outcome of the patch simplification step; (d) Culminating in the final patch layout.

Building on the intuition of Verhoven et al.[131], our pipeline relies heavily on an underlying curvature-oriented cross-field guide. Since patches can only bend orthogonally to the prescribed ruling lines, aligning these lines with the direction of minimal curvature is essential. As illustrated in Figure 3.4, our framework includes several steps:

Input preparation

In addition to input mesh clean-ups, we generate a guiding cross-field on the surface [130], aligned with boundaries and with principal curvature directions where (red parts in Figure 3.4.a).

If a symmetric decomposition is sought, we generate the patch layout conducted on one side of the mesh using a symmetry plane. Then, we replicate this patch layout on the opposite side and retrieve the optimized ruling lines and corresponding 3D embedding.

Throughout this procedure, we introduce one (or more) additional patterns along the symmetry plane. Ultimately, if feasible, we merge patches along the symmetry plane.

The effectiveness of cross fields in guiding high-quality parametrization, achieving superior patch decomposition, and quadrilateral re-meshing has been demonstrated in prior works [53, 58, 139].

Patch layout construction

We populate the mesh with a network of intersecting *paths*, which split the input surface into patches. Paths are field-aligned lines either closed in a loop or terminating at the mesh boundaries. In the first cycle, add paths over the mesh, one by one, until the resulting patch layout is determined to be *viable*. A layout is deemed "viable" if it satisfies all necessary conditions or criteria, such as connectivity and patch segmentation, allowing it to be used as the final layout or as a foundation for further refinement. In other words, the layout should have enough intersecting paths to form meaningful patches on the mesh but not have unnecessary or redundant paths that do not contribute to its effectiveness. At every step, we trace a new path striving to address one remaining problem as detected in the current layout. As this procedure is greedy, a path can be made redundant by subsequently added paths. So, in a second cycle, we test all paths for removal, in inverse order of creation: the tested path is removed if the layout is determined to still be viable without it. Figures 3.4.b and c show an example of the layouts after either cycle.

Patch-layout evaluation

At the core of the above process, we must assess whether a given layout is viable. In addition to topological checks, this is done by attempting to morph the mesh into a spatial configuration that *approximately* meets all the geometric requirements described in Section 3.2.1). If no such morphing is found that adheres to the maximally tolerated displacement e_{max}, then the configuration is determined not to be viable. Importantly, the requirements need only to be approximated in this phase, allowing us to impose them as soft constraints that can be easily optimized globally.

After each step, the viability of the resulting patch layout is evaluated by estimating how easily the surface could be morphed into one satisfying all specifications (Section 4.2.2). The process is terminated when the layout is deemed to be sufficiently viable until all design specifications are met by the resulting layout. This includes geometric accuracy, which is evaluated by optimizing the resulting patches and estimating how well they can be approximated with a ruled surface.

To generate pattern pieces that meet the specified requirements, we establish a network of *paths* across the mesh to delineate distinct panels on the surface. Following the approach of [90], these paths are strategically oriented to align with the underlying cross-field and may form closed loops or terminate at the mesh boundaries. To control patch complexity, users can define the maximum allowable number of corners in a single patch. The patch formation process is governed by a global optimization mechanism. Given a layout, this global optimization first identifies the optimal ruling lines for each patch and then determines the best resulting 3D embedding where the ruling lines are straight. This is achieved by solving a linear system that balances the straightness of the ruling lines, accuracy in the approximation, and overall smoothness. Paths are incrementally introduced until the maximum distance between the resulting 3D surface and the original mesh is within a specified threshold (see Figure 3.4.b). This approach ensures the production of pattern pieces that meet design specifications and adhere to the desired level of accuracy in replicating the input mesh.

Final discretization

Finally, we triangulate each patch into one stripe of long and thin flat triangles, with ruling lines represented as edges. This final re-meshing meets most of the requirements (such as rule straightness and water-tightness) in full and by construction, that is, in virtue of its polygonal connectivity alone; notably, this also applies to developability as well, as observed in Section 1. At the same time, because we already determined that the 3D mesh can be morphed in a configuration close to fulfilling these requirements, we know that this final remeshing introduces a geometric error that is, at most, very close to the prescribed tolerance. The patches are then trivially flattened (without any distortion), obtaining a final 2D layout ready to be cut, bent, and assembled.

CHAPTER 4

ALGORITHMIC DETAILS AND IMPLEMENTATION

In this section, the implementations of the developed models are provided. In section 4.1, the implementation details of the spheroids fusion are described. Firstly, the concept of the SPH method is explained, and then the designed algorithm of the simulation will be reported, and at the end, the surface reconstruction method will be described.

In section 4.2, we will discuss the detailed implementation of our proposed algorithm to make a developable mesh using the idea of ruling lines. This includes how we initial the cross-field on the mesh and decompose the mesh into patches, then how we sample and drive the ruling lines in the patches and how we apply the developability constraints on ruling lines.

4.1 SPH implementation of the spheroids fusion process

The details of the implementation of the models described in the previous section are provided here. The basic idea of the SPH method is to discretize a continuous medium (in our case, a spheroid) to a finite number of particles and distribute the different particle attributes in its volume by using a kernel function (see Figure 4.1). We remark here that particles are *not* meant to represent the cells that constitute the spheroid, as they merely provide a discrete version of the spheroid seen as a continuum. Thus, their number is by no means proportional to the number of cells in a spheroid; it rather controls the trade-off between the computational complexity and the accuracy of the simulation. For our simulations, we considered about 100 particles for each spheroid.

As mentioned in the methodology section, in the SPH method, the displacement of each particle is computed regarding its neighbouring particles. If we want to have only elastic objects, we need to compute each particle's neighbours only at the beginning of the simulation. However, in the case of plastic deformation, we need to update neighbours in each simulation iteration as the reference shape will differ in each time step. In the solid model, a volume conservation force does not allow spheroids to change their volume significantly. Therefore, we do not need to update each particle's neighbours as the reference shape is always the same.

For modelling the attraction force between spheroids, we utilized the Lennard-Jones potential force [42]. The Lennard-Jones potential is commonly employed to approximate the Van der Waals force between neutral atoms or molecules [9]. The following equation expresses the Lennard-Jones potential:

$$V(r) = 4\epsilon[(\frac{\sigma}{r})^{12} - (\frac{\sigma}{r})^{6}] \tag{4.1}$$

Where $V(r)$ represents the potential energy between two particles at a distance r, ϵ denotes the depth of the potential well determining the strength of the interaction between particles, and σ is the distance at which the potential reaches its minimum value. Our observations indicate that spheroids only approach each other if sufficiently close (please refer to Section 6). We can achieve such behaviour with the Lennard-Jones potential by tuning the parameters σ and ϵ. The potential becomes zero beyond a certain distance and increases as the distance between spheroids decreases. To convert this potential energy into an attraction force, we need to compute the derivative of the potential energy with respect to the particles' distance r.

Additionally, as explained in Section 1, we establish connections between particles belonging to different spheroids when they get in contact. We used this approach to model surface adhesion force.

For implementing the SPH solid and fluid models, we have used a polynomial kernel to distribute mass around each particle:

$$W^{poly}(r,h) = \frac{315}{64\pi h^9} \begin{cases} (h^2 - r^2)^3 & if\, 0 \le r \le h \\ 0 & otherwise \end{cases} \tag{4.2}$$

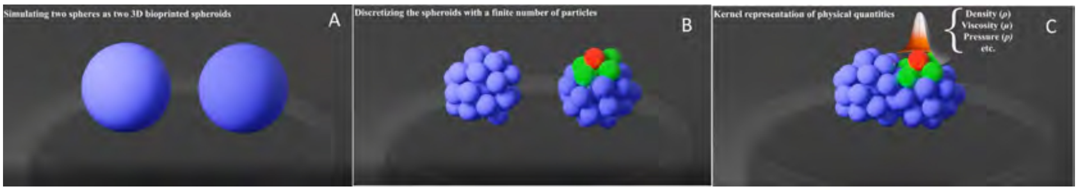


Figure 4.1: A: Solid spheroids simulated. B: Discretized spheroids using a finite number of particles (SPH uses neighbour particles - here the green spheres - to compute a physical quantity of the centre particle - here the red particle. C: Kernel distribution (the shape distribution) over a particle's neighbours using the SPH method. The full video is available here.

where r is the distance to the particle, and h is the kernel's support. There are a couple of properties that the smoothing kernel has to obey. First, the influence of particles at a distance of H from the selected particle must drop away to zero as H increases. This means that particles that are far away do not interact with selected particles. The second is that W has to sum to 1 over a sphere of radius H.

Therefore, the density at particle i can be obtained as:

$$\rho_i = \sum_j m_j w_{ij}, \tag{4.3}$$

where $w_{ij} = W(\left|x_j - x_i\right|, h_i)$ and the volume of the particle i is given by $v_i = m_i/\rho_i$. We have used a moving least squares approximation of the displacement field to solve the differential equations related to deformation. Therefore, we can make an elastic-plastic system that is able to keep its shape by adding some internal force to each particle in case we have external compressive and tensile forces.

Hence, in each frame of the simulation, we need to solve a set of differential equations and update the internal force of each particle. We set the simulation time step and velocity damping to 0.01. The pseudo-code of the simulation loop is shown in algorithm 1.

In some simulations, we assumed elastic collisions between particles. In an SPH simulation, an elastic collision between two particles occurs when they collide and bounce off each other without any loss of kinetic energy [64]. In this case, particles are considered to have collided when the distance between them is less than the radius of one of the particles. Suppose particles i and j have collided. To update their velocities, we can use the following formula:

$$
\begin{aligned}
v_i &= v_i - \frac{2m_j}{m_i + m_j}\frac{< v_i - v_j, x_i - x_j >}{\|x_i - x_j\|^2} \\
v_j &= v_j - \frac{2m_i}{m_i + m_j}\frac{< v_j - v_i, x_j - x_i >}{\|x_j - x_i\|^2}
\end{aligned}
\tag{4.4}
$$

where v_i and v_j are the velocities of particles i and j, respectively, and m_i and m_j are the masses of particles i and j, respectively. $<,>$ denotes the inner product operator [114].

Algorithm 1 Simulation loop of solid-based model

for all particle i **do**
    $Internal\,force \leftarrow 0$
    $Acceleration \leftarrow 0$
    $Displacement \leftarrow New\,position\ -\ Rest\,position$
end for
for all particle i **do**
    $Compute\,\nabla u$
end for
for all particle i **do**
    $Compute\,strain\ \rightarrow\ \varepsilon\quad (Eq.3.1)$
    $Compute\,stress\ \rightarrow\ \sigma\quad (Eq.3.2)$
    $Compute\,internal\,force\ \rightarrow\ \mathbf{F_{internal}}\quad (Eq.3.6)$
end for
for particle i **do**
    for particle j **do**
        $Compute\,elastic\,collision\quad (Eq.4.4)$
        $update\,particle\,i\,velocity$
        $update\,particle\,j\,velocity$
    end for
end for
for all particle i **do**
    $Update\,acceleration$
    $Update\,velocity$
    $Update\,position$
end for

The simulation frame in the fluid-based model is similar to the solid-based model. However, we do not have a volume conservation force as we have plastic deformation, and hence can not guarantee that spheroids will keep their volume during the merging process. Instead, we have added pressure and viscosity as new forces in the model. Also, we need to update each particle's neighbourhood in each simulation iteration as we may have plastic deformation and, consequently, a change in the reference shape. One of the simulation's major computational components is updating each particle's neighbours in each iteration. For determining each particle's search radius, H, we first obtained the 15 nearest neighbours for each particle. Then we set the neighbour's search radius as the maximum distance of each particle to its initial neighbours. Then, we keep this radius for the rest of the simulation and find each particle's neighbours based on the KDTree data structure.

In the fluid-based model, we aim to represent the dynamics of the spheroids by a set of particles. We will do that by taking the material derivative of the equation of motion. The material derivative is the derivative along a path with a given velocity. Hence, particles will be travelling along this path in space. The equation that we actually will solve for each particle i is:

$$\frac{dv_i}{dt} = g - \frac{1}{\rho_i}\nabla p + \frac{\mu}{\rho_i}\nabla^2 v \tag{4.5}$$

where μ is the viscosity, ρ is density and ∇p is the pressure gradient. In particular, we have simplified the incompressible Navier-Stokes equation by taking the material derivative so that we have a simple equation for the motion of one particle.

According to the Navier-Stokes equation, the density term of each particle can be approximated by its neighbours' masses using the SPH kernel function (see Eq. 4.3). For the fluid-based model, we have used a polynomial kernel (Eq. 4.2) for approximating density, just like the solid-based model. The above equation says that ρ, the density, is approximately equal to the sum of the masses of nearby points weighted by the smooth kernel W. Also, we approximate the pressure gradient as follows:

$$\frac{\nabla p_i}{\rho_i} \approx \sum_j m_j\left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2}\right)\nabla W(r_i - r_j, h) \tag{4.6}$$

Here, W is, in fact, a vector expression, and because a gradient is an effector, as a result, this whole approximation has an effective value, which is what we would expect for the gradient of p. W is the Spiky kernel function, and it is defined as:

$$W^{spiky}(r,h) = \frac{15}{\pi h^6}\begin{cases}(h-r)^3 & if\, 0 \le r \le h \\ 0 & otherwise\end{cases} \tag{4.7}$$

where r is the distance to the particle, and h is the kernel's support. The reason for using this kernel instead of the polynomial kernel provided in Eq. (4.2) is that the spiky kernel generates repulsion forces near the centre of the kernel and avoids particles clustering around the centre of the kernel.

Finally, the last term that should be approximated is viscosity which can be computed as:

$$\frac{\mu}{\rho_i}\nabla^2 v_i \approx \frac{\mu}{\rho_i}\sum_j m_j(\frac{v_j - v_i}{\rho_j})\nabla^2 W(r_i - r_j, h) \tag{4.8}$$

where μ is a scalar coefficient that defines the viscosity of the fluid. If μ has a small value, it is a relatively non-viscous fluid like water; if μ has a large value, it can be a relatively viscous fluid. v_i is the velocity of particle i and ρ_i is density of particle i. Also, for the viscosity force, we need to use a kernel whose Laplacian is positive everywhere. Thus we define W in Eq. 4.8 as:

$$W^{viscosity}(r,h) = \frac{15}{2\pi h^3}\begin{cases} -\frac{r^3}{2h^3} + \frac{r^2}{h^2} + \frac{h}{2r} - 1 & if\, 0 \le r \le h \\ 0 & otherwise \end{cases} \tag{4.9}$$

Note that there is a vector subtraction in the Eq. 4.8; if two particles have the same velocity, the viscosity force will be zero. It means that there will be no viscous interaction between them. On the other hand, if the vectors are slightly different, they will have a small viscous interaction. They will have a relatively large viscous interaction if they differ significantly. Hence, we can see how this term encourages the particles to travel together or move in the same direction over time. In fact, particles near each other and travelling in the same direction will not influence each other. However, particles that are near each other and travelling at different velocities will influence each other and try to pull each other together into a common velocity. The simulation framework of the fluid-based model is explained in algorithm 2:

In the next section, we will show samples of our simulation results with their corresponding parameters and compare them with actual experiments to see how much our models can describe the spheroids' mechanical behaviour during the merging process.

4.1.1 Surface reconstruction

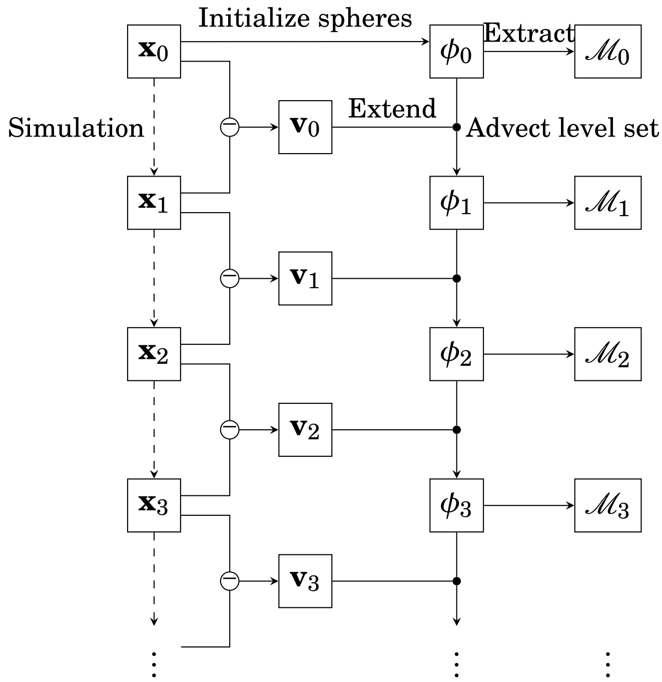
The physical simulation updates the position of the particles at every loop. We use the list of particle positions for each frame to produce a temporal sequence of 3D triangle

Algorithm 2 Simulation loop of fluid-based model

for all particle i **do**
    $Pressure\,force \leftarrow 0$
    $Viscosity\,force \leftarrow 0$
    $Elastic-plastic\,force \leftarrow 0$
    $Acceleration \leftarrow 0$
    $Displacement \leftarrow New\,position - Rest\,position$
end for
for all particle i **do**
    Update neighbourhoods
end for
for all particle i **do**
    $Update\,density \rightarrow \rho$
    $Update\,pressure \rightarrow p$
    $Update\,volume \rightarrow v$
end for
for all particle i **do**
    $Compute\,pressure\,force \rightarrow f_{press}$
    $Compute\,viscosity\,force \rightarrow f_{viscos}$
end for
for all particle i **do**
    $Compute\,\nabla u$
end for
for all particle i **do**
    $Compute\,strain \rightarrow \varepsilon = \varepsilon^p + \varepsilon^e$
    $\varepsilon' = \varepsilon^e - \frac{Tr(\varepsilon^e)}{3}\mathbf{I}$
    $Compute\,stress\,\sigma$
    if $\gamma 1 \leq \|\varepsilon'\|$ **then**
        $\Delta\varepsilon^p = \frac{\|\varepsilon'\|-\gamma 1}{\|\varepsilon'\|}\varepsilon'$
        $\varepsilon^p = (\varepsilon^p + \Delta\varepsilon^p)\min(1, \frac{\gamma 2}{\|\varepsilon^p + \Delta\varepsilon^p\|})$
    end if
    $Compute\,elastic-plastic\,force \rightarrow \mathbf{f_{ep}}$
end for
for all particle i **do**
    $F_{Total} = f_{press} + f_{viscos} + f_{ep}$
    $Update\,acceleration$
    $Update\,velocity$
    $Update\,position$
end for

meshes representing the spheroid fusion process. The surface reconstruction program is written in C++ and based on the OpenVDB library [75] for sparse volumetric processing.

The created meshes can then be exported as OBJ files or visualized with the interactive viewer included in our program.



	Legend
$\mathbf{x}_i$	Particle positions
$\mathbf{v}_i$	Particle velocities
ϕ_i	Level sets
$\mathscr{M}_i$	Surface meshes

Figure 4.2: Diagram of the surface reconstruction algorithm.

We operate on implicit surfaces in the form of narrow-band level sets, in order to avoid dealing with topological changes of the surface, such as merging events. We use the implementation of narrow-band level sets provided by OpenVDB. Initially, the first frame of the simulation sequence is used to generate an implicit representation of the spheroids. In order to do so, we first determine the spheroid to which each particle belongs, with a simple clustering step. Then, assuming that the spheroids are perfect spheres in their initial state, we create the starting implicit surface. Finally, we create the initial mesh with OpenVDB's isosurface extraction utility, which is conceptually similar to the dual marching cubes algorithm [81].

For every frame, except the last, we determine the velocity of each particle by the difference with the next frame in the sequence (Figure 4.2). We treat this information as a sparse sampling of a globally-defined velocity field, which we reconstruct with kernel interpolation. The global velocity field is then used to deform and displace the implicit surface via level set advection to produce the next implicit surface. By construction, the advected surface follows the motion of the particle system. We make use of OpenVDB's advection solver for this purpose. The generated implicit surface serves as the starting point for the next advection step, and it is also used to produce the corresponding frame's mesh, again via isosurface extraction.

The reconstruction process has several parameters that can be tuned to manage the trade-off between surface quality and execution times. These include grid coarseness for the implicit surfaces, the width of the narrow band, and meshing adaptivity.

4.2 Developable patch decompostion

This section will address the implementation of our design algorithm for decomposing a mesh into a finite number of developable patches. As described in the methodology section, the implementation process includes cross-field computation, decomposing the mesh into a few patches based on cross-field, sampling and deriving ruling lines and applying developability constraints with regards to the ruling lines to make each patch developable. The above-mentioned steps will be conducted in an iterative process to converge to a developable mesh consisting of as little as a number of patches while it tries to satisfy aesthetical constraints.

4.2.1 Input preparation

We assume the input mesh to be a manifold with well-shaped triangles; if these conditions are not met, we uniformly re-mesh the input using standard tools (we use the Open Source suite [16]). In the additional materials, we include all the used input meshes.

Our initial step involves the construction of a 4-rotational-symmetric (4-RoSy) tangent-vector field [130] on the input surface. This cross-field plays a pivotal role in our framework, steering the entire tracing process; the cuts conform to the directions dictated by this cross-field. Aligning seams with the principal curvature directions intuitively emerges as an effective strategy to enhance the developability of the resulting patches, a concept supported by numerous quadrangulation methods showcasing a robust correlation between developability and curvature alignment.

To initialize the field, we extract curvature directions at a lower scale using the method introduced by Panozzo and colleagues [88]. We leverage the anisotropy of curvature directions (i.e., the ratio between the norms of the two principal curvature directions) to identify regions where these curvatures play a crucial role [12].

Subsequently, we employ a globally smooth method proposed in [24], introducing soft constraints to align the field with the principal curvature directions. We modulate each soft constraint value by using the ratio between the norm of the two principal curvature directions, ensuring adherence to the main curvature directions in relevant regions while maintaining smoothness elsewhere. In addition to these soft constraints, we enforce alignment with the boundaries, ensuring that the traced paths intersect the boundary orthogonality. The result of the initialization and smoothing of the guiding cross field is shown in Figure 4.3.

Cross-Field Computation

To construct the cross-field (a smooth 4-rotational-symmetric tangent vector field) required by our method, first, we extract the two principal curvature directions, at a low scale, using the method by Panozzo and colleagues [88] (Figure4.3, left); then (Figure4.3, right), we employ the method proposed in [24] to produce a globally smooth cross-field, enforcing alignment to the principal curvature directions with soft constraints. We modulate the constraints with the anisotropy factor, i.e. the ratio between the norms of the two principal curvature directions. This results in an adherence to curvature directions only in anisotropic regions, while maintaining smoothness elsewhere. An additional set of hard constraints enforces alignment with the mesh boundaries (if any).

4.2.2 Patch-layout evaluation

To steer the patch layout construction, we need to estimate if the given patch layout is viable: to be viable, a patch layout must satisfy topological and geometric conditions. This assessment must be efficient, because it is repeated after each path insertion or attempted removal.

Topological conditions

Given a set of intersecting paths, each path is split into *arches* by intersection with other paths. The arches divide the mesh into patches; each patch is delimited by a number of arches, touching the corners of the patch.

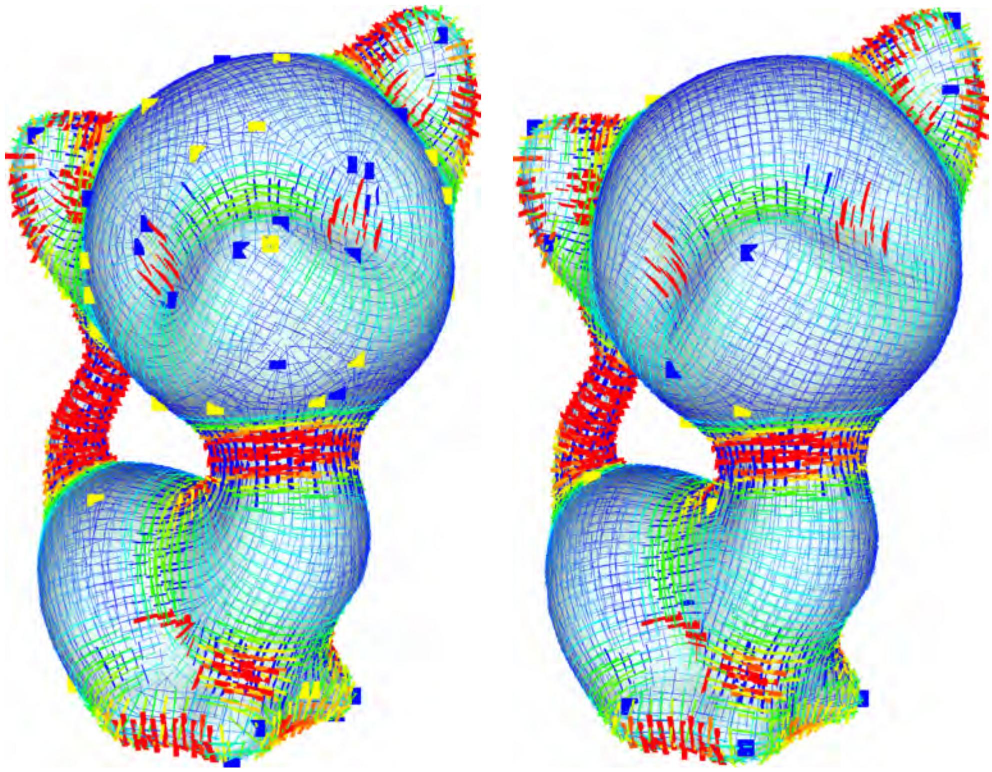


Figure 4.3: The initial curvature field with singularities (left) and the final result after smoothing. The red areas indicate anisotropic regions, where curvature directions are more significant and thus better preserved in the smoothing process.

We check that each patch is topologically equivalent to a disk; then, we check that no patch exceeds n distinct arches on its boundary, because that would contradict the objective of simplicity. Guided by empirical evaluation, we used $n = 6$. If either condition is not met, the layout is deemed not to be viable.

Geometric condition

To be viable, a patch layout must also admit a sufficiently good approximation as a set of valid ruled surfaces. To estimate this, we attempt to globally morph the surface in a way that (approximately) deforms each patch into a valid ruled surface. This is cast as a global optimization problem, minimizing the energy that penalizes the introduced displacements and promotes the geometric characteristics of ruled surfaces. The minimizer of this energy is then tested: if any introduced displacement exceeds e_{max}, it is an indication that the set of patches cannot be forced into a set of ruled surfaces within the prescribed tolerance, and the layout is deemed non-viable.

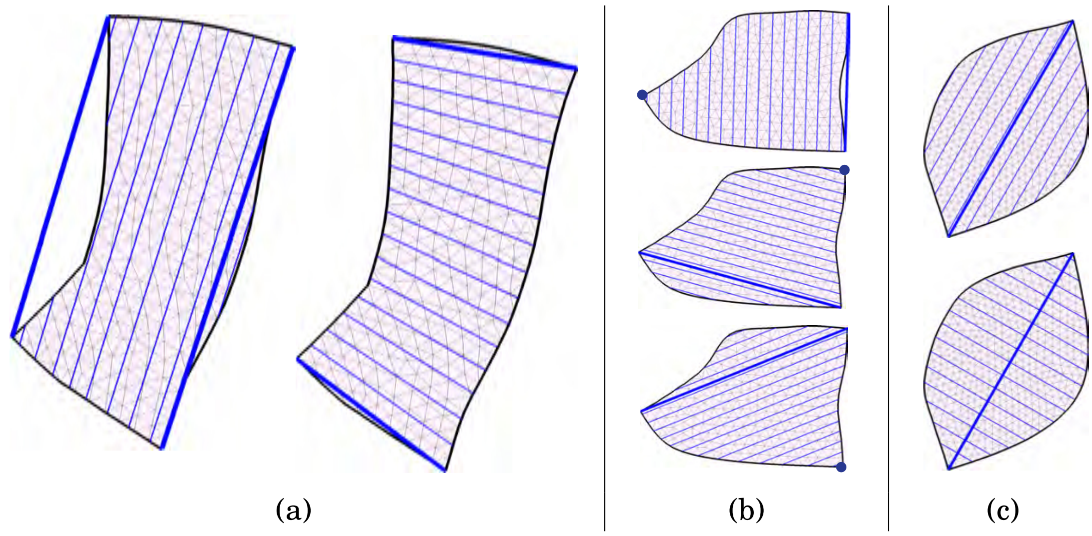


Figure 4.4: Sampling ruling lines: (a) In the case of a patch with an even number of sides, we create sets of sweeping segments between pairs of non-adjacent sides; (b) For a patch with an odd number of sides, we additionally sweep segments between a side and the opposite vertex; (c) In the scenario where a patch has only two sides, we generate a set of samples parallel and orthogonal to the two corners;

To define this morphing, we first need to determine the ideal orientations for the rules in each patch, as follows.

4.2.3 Derivation of Ruling lines

The task of determining an optimal set of ruling lines has been tackled in prior work. The method proposed in Dev2PQ [131] involves extracting straight lines from a level set function defined by a curvature-aligned line field. While this approach is elegant and optimal, its applicability is significantly influenced by the distribution of field singularities. Our optimization strategy necessitates frequent modifications to the patch layout since the insertion or removal of paths entails the recomputation of ruling lines, and we cannot assure a consistent singularity distribution at every step of this process. Consequently, we have chosen a different approach, prioritizing robustness and efficiency over optimality.

4.2.3.1 Sampling ruling lines

Our devised ruling line sampling procedure capitalizes on the inherent alignment of our decomposition boundaries with the directions of the cross-field. We introduce a novel, efficient, and robust combinatorial method to sample ruling lines. Suppose we have a

2D mapping of a patch (we used least-squares conformal mapping [55]). For each pair of non-adjacent sides, we consider the two 2D segments that are created by connecting the corners of those sides in 2D, and we generate an internal sequence of segments that sweeps between those two, with the purpose of sample uniformly the 2D space and follow the directions of the boundaries. When segments are outside the interval limited by the sides, we simply replicate them in parallel (see Figure 4.5.a shows the effect of the sampling procedure). We test every possible pair of non-adjacent edges and generate a candidate set of ruling lines. Then, we rank the optimality of each set by considering the straightness of its segments when mapped back to the 3D space (we simply verify this condition by sampling regular points in 2D and reporting to 3D to esteem the length of the corresponding curve). When a patch has an odd number of corners, we also test the edge with the opposite vertex (see Figure 4.5.b). When a patch has just two corners, we generate a set of segments parallel to the side and in an orthogonal direction (see 4.5.c). If a patch has no sides, we subdivide a 180-degree angle interval regularly to select the best one.

4.2.3.2 Choosing the Ruling lines

The task of determining an optimal set of ruling lines over a given patch has been tackled in prior work. The method proposed in Dev2PQ [131] involves extracting straight lines from a level-set function defined by a curvature-aligned line field. This approach is elegant and optimal, but its applicability depends strongly on the presence and distribution of field singularities. In addition, this solution is not efficient enough for our purposes. Instead, we have devised a different approach, that prioritizes robustness and efficiency.

We parameterize the patch in 2D, using As-Rigid-As-Possible parametrization [59], which is efficient and finds the most isometric mapping. If this mapping (which is unique up to 2D rotation and translations) presents self-overlaps, the patch layout is deemed not viable, as it infringes the condition of bijectivity (self-overlaps are detected by looking for self-intersections of the patch boundary).

Otherwise, we consider a number of candidate sets of ruling lines, each defined by a family of straight lines in parametric space. The parametrization maps each corner of the patch into a 2D position. We consider the 2D polygon defined by these n positions: for each pair of opposite sides, we generate a sequence of 2D segments sweeping the space between those two sides at regular intervals (see Figure 4.5.b). If the number of corners is odd, we also generate a sweeping between each edge and its opposite corner (see Figure 4.5.b). When the polygon has only two corners, we generate two candidate sets: one in the direction connecting the two corners and the other in the orthogonal direction (see 4.5.c). Each line is intersected with a 2D image of the patch, giving a 2D segment traversing the patch side to side.

We pick the candidate family of lines that best preserves the straightness of all lines when mapped back to 3D space. We measure this by splitting each 2D line regularly into a number of small sub-segments and averaging the norms of the cross products between consecutive seg-segments.

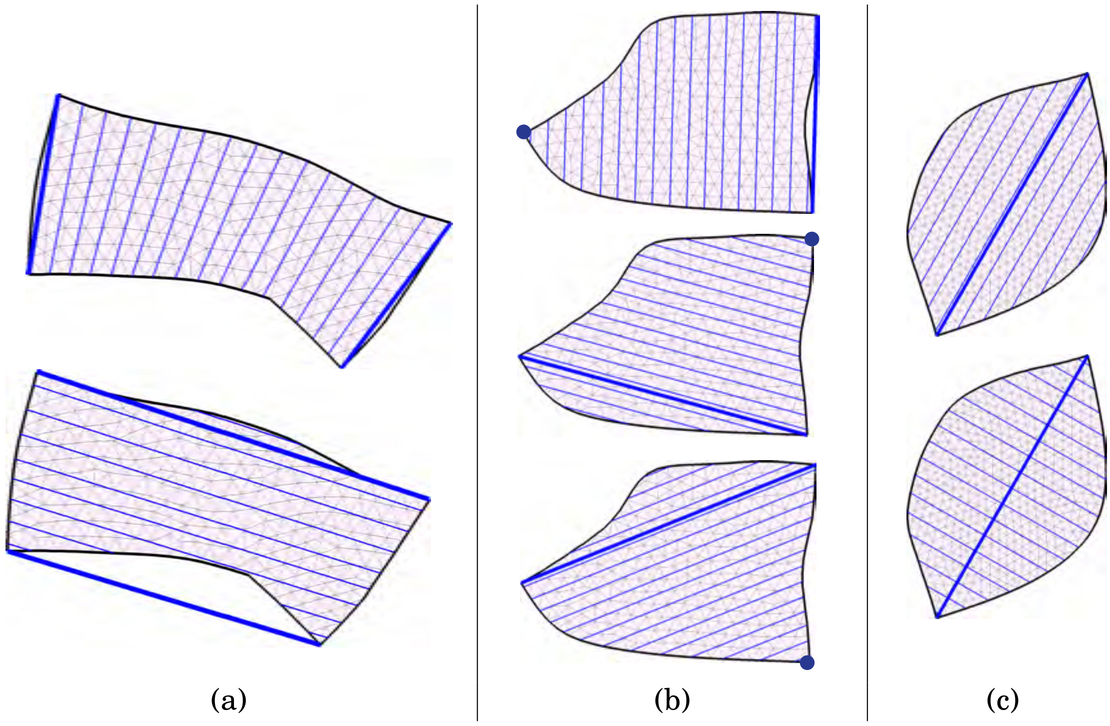
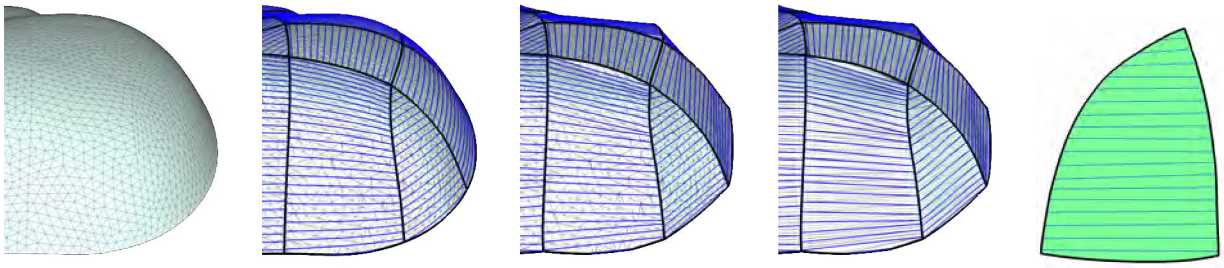


Figure 4.5: Examples of candidates for ruling lines considered for (a) a four-sided patch, (b) a three-sided patch, and (c) a two-sided patch. The patches are shown in 2D parametric space.

4.2.3.3 Embedding Selected Ruling Lines

After carefully selecting ruling lines, we seamlessly incorporate them into the triangulation process by subdividing triangles within the 2D spaces defined by the chosen set of segments. This approach enables the imposition of constraints to ensure the straightness of the ruling lines while optimizing the embedding of the patch layout in 3D. Refer to Figure 4.7 for a visual representation of the results achieved through this procedure.



(a) (b) (c) (d) (e)

Figure 4.6: A summary of meshing used during the entire process. Edges representing ruling lines are in blue, and thick black lines are edges forming the patch boundary: (a) Original meshing, after path smoothing (Section 4.2.4); temporarily re-meshed surface before (b), and after (c), the Morphing into approximate ruled surfaces (Section 4.2.3.4); (d) final remeshing of the patches (Section 4.2.5); (e) the final 2D layout of one patch.

4.2.3.4 Morphing into approximate ruled surfaces

Next, we want to find a morphing of the surface into a set of ruled surfaces that satisfies the necessary conditions. Strict adherence to these conditions is not necessary, as this is done solely for the purpose of evaluating the potential layout.

As a preliminary, we temporarily re-mesh the input surface, so that rules are now represented as collections of mesh edges (without affecting the shape of the surface). This is done by leveraging the parametrization found in the previous phase. Each edge intersecting a ruling line in 2D is split, inserting a new vertex at the intersection point; vertices at the boundary of all patches are all duplicated so that each patch becomes a disconnected component of the mesh. Figure 4.6-b shows an example of this remeshing.

In the following, we group the vertices into three sets: vertices on the ruling lines RL, vertices on the border of a patch B, and internal vertices that do not belong to any ruling line I.

We define an energy E_{tot}, as a squared function of the positions of the vertices, designed to penalize a number of undesirable configurations; E_{tot} is the sum of a number of terms, as follows.

Approximation faithfulness term A term E_{approx} measures the introduced deformation (thus the approximation errors), penalizing the distance of each vertex v_i from its original position v_t:

$$E_{approx} = \sum_{v_i \in RL} ||v_i - v_t||_2^2. \tag{4.10}$$

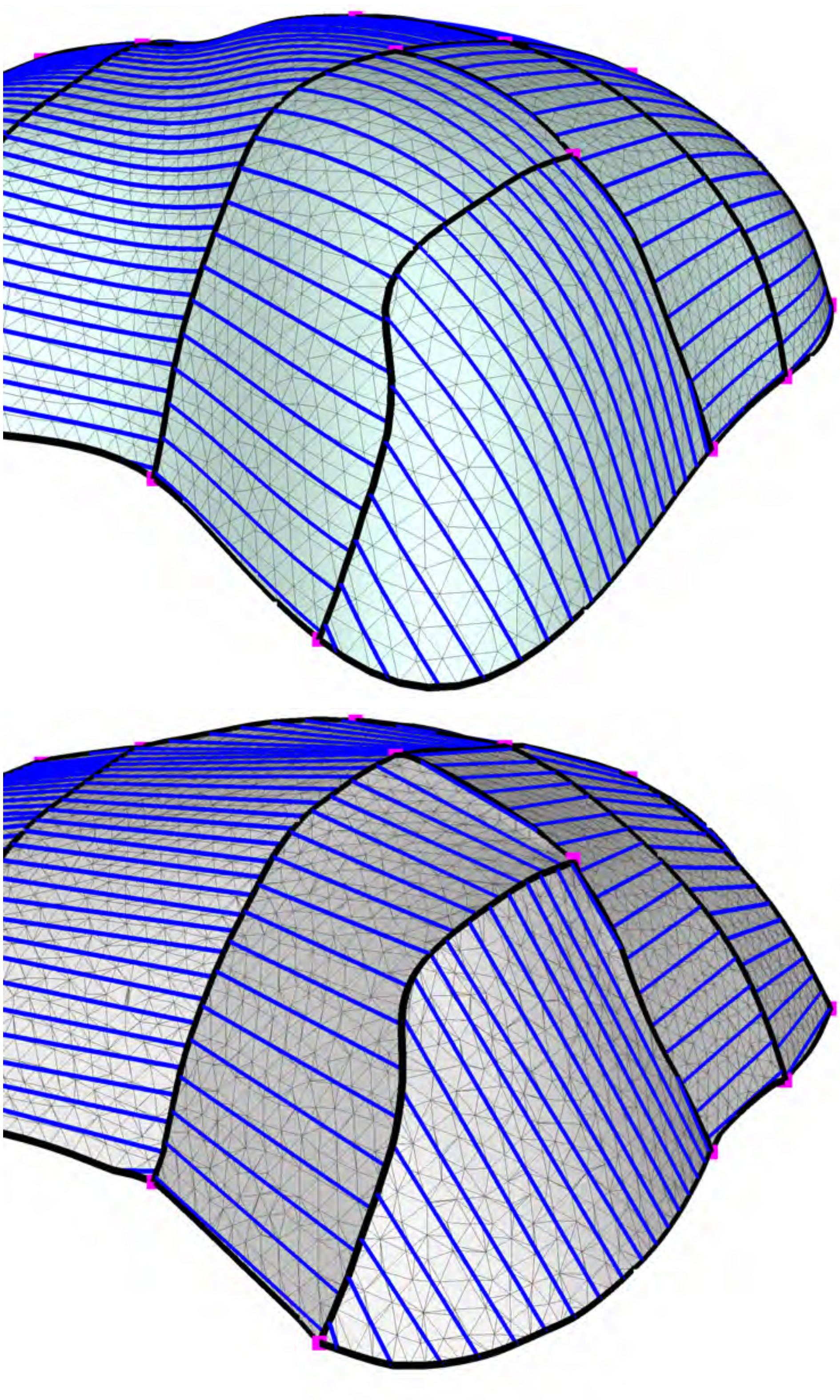


Figure 4.7: The effect of the straightness of the ruling lines on a 3D patch. This constraint forces each patch to be ruled in a plane

Rule straightness term To encourage straightness of ruling lines, we add a term that dictates that, for every triplet of consecutive vertices on a ruling line v_i, v_j, v_k in RL, the position of the middle one v_j must be a linear interpolation of the other two. The interpolation weight α is a constant determined by the parametric positions in 2D of the three vertices:

$$E_{straight} = \sum_{\text{consecutive } v_{i,j,k} \in RL} ||\alpha v_i + (1-\alpha)v_k - v_j||_2^2. \tag{4.11}$$

Local planarity term The remaining vertices should lie in the plane defined by their neighbours. An additional energy term $E_{laplace}$ strives to position each vertex v_i in $\mathbb{B}$ or $\mathbb{I}$ in the weighted average of its 1-ring neighbour $N(v_i)$, defined by the cotangent weights [92] as computed in the 2D layout.

$$E_{laplace} = \sum_{v_i \in B} ||v_i - N(v_i)||_2^2. \tag{4.12}$$

C^0 continuity term In order to preserve C^0 continuity across patches, it is necessary that each boundary vertex v_i in B is in the same position as its matching vertex v_i' on the boundary of a neighbouring patch. This not being the case is penalized by the energy term:

$$E_{seams} = \sum_{v_i \in B \cup I} ||v_i - v_i'||_2^2. \tag{4.13}$$

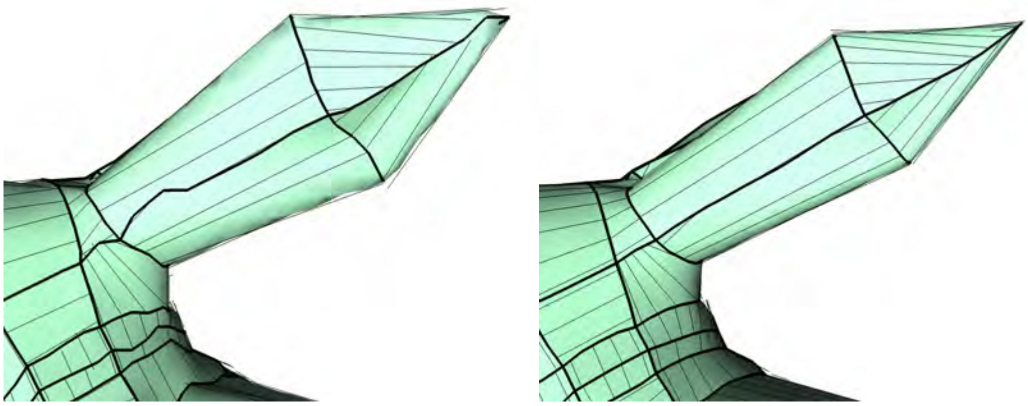


Figure 4.8: The effect of the energy term E_{smooth} on the boundary shapes (left: without; right: with).

Boundaries smoothness term To encourage smoothness of the patch boundaries, we add additional energy terms, E_{smooth}, as the bi-Laplacian smoothing term computed on sequences of boundary vertices. Without this term, the patch boundary can morph into unnecessarily wiggly lines, as shown in Figure 4.8.

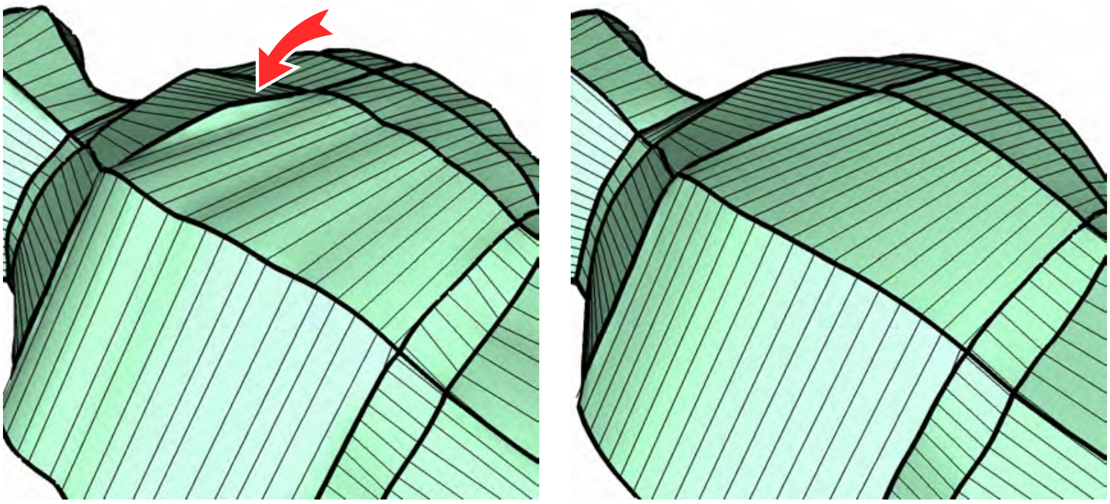


Figure 4.9: Patches can bend abruptly over ruling lines are orthogonal, even when the target surface is flat (left). This undesired occurrence is prevented by introducing the energy term E_{flat} (right).

Bending avoidance term Another potential problem is that a patch can bend abruptly, folding along a ruling line. This configuration does not infringe the definition of ruled surfaces, nor its developability, but can still be undesirable for a number of reasons (including ease of construction or aesthetics), when the sudden bent is not found in the target surface. Figure 4.9, left, shows one example. To address this, we add an energy term, E_{flat}, that promotes the flatness of the patch for such reason. E_{flat} is defined in the same way as equation (4.11), but acts on triplets of vertices $v_{i,j,k}$ on three *different* consecutive rules. This term should not trigger in areas where the original surface actually bends abruptly. Therefore, we weigh the contribution for each triplet according to the local isotropy factor, the middle vertex v_j, defined as one minus the ratio of the magnitudes of the associated minimal and maximal curvature directions.

The total energy is the sum of all the above terms, each weighted by constant factors reflecting their relative importance:

$$E_{tot} = E_{approx} + E_{laplace} + 10^5 E_{straight} + 100 E_{seams} + E_{smooth} + 2 E_{flat}. \tag{4.14}$$

The weights are chosen based on experience to get the best results. It is obvious that a higher weight on each term will force the program to comply with those constraints more than others. Figure 4.6-c shows an example of the resulting morphing.

4.2.4 Patch Layout creation

Our patch creation strategy requires tracing a sequence of paths oriented with the cross-field.

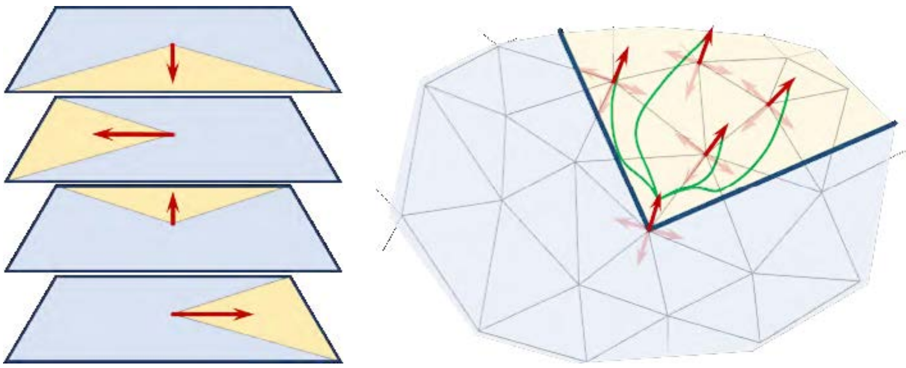


Figure 4.10: We frame the problem of tracing field-aligned paths as the minimal path over a graph with four nodes for each vertex, one for each tangent direction of the cross-field (left). The graph arches are defined by connecting nearby nodes with matching cross-field directions (right).

Tracing Field-Oriented paths To trace paths aligned with the field, we adopt the efficient graph-based methodology adopted in Nuvoli et al. [83] and Pietroni et al. [90], which we summarize for completeness. A graph is constructed with four nodes for each vertex, each aligned to one cross-field direction. Then, the problem of path tracing as a shortest-path search, where a designated *source node* is connected to one of potential *destination nodes* within the graph. We trace two types of paths: those connecting border-to-border and those forming loops. When tracing a *loop* from a specific vertex, an internal node is selected as the source, with the objective of returning to the same node. For border-to-border paths, a source node is chosen on a boundary vertex whose direction *enters* the mesh, and the destination nodes encompass all border nodes whose direction *exits* the mesh.

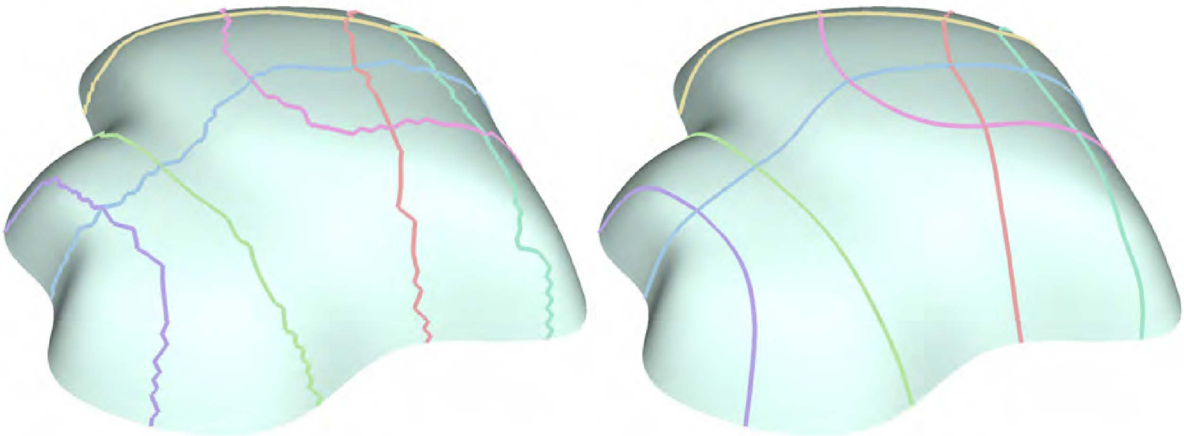


Figure 4.11: Traced path as identified as minimal paths over the mesh graph (left), and after the smoothing phase.

Path smoothing Given that paths consist of sequences of edges of the triangle mesh, they often exhibit a zig-zagging shape, contrary to our objective of layout simplicity. To counter this, we employ extrinsic smoothing to all the vertices on the path, followed by their re-projection onto the original surface (see Figure 4.11).

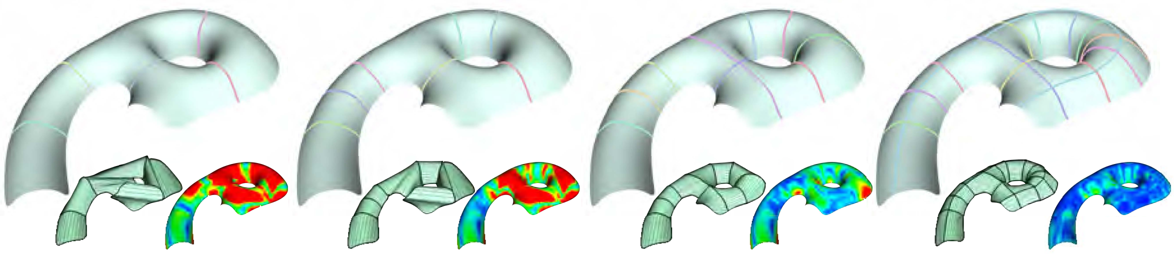


Figure 4.12: A few steps of path insertion with developable 3D embedding and relative approximation error.

Path insertion strategy We begin by sampling a set of candidate paths that trace loops or extend from border to border. For efficiency, we uniformly subsample the source nodes both on the border and within the mesh interior. After [63], we insert paths using a greedy approach that prioritizes paths furthest from those previously inserted. Distances for each node are computed using the $M4$ stratification of the graph and are subsequently averaged for each path.

For each insertion step, we update the entire patch layout to evaluate its viability. Then, we mark each patch failing to meet the topology (as outlined in section 7.3.1) or geometry criteria. A candidate path is considered for insertion only if it splits a patch that does not match the specified goals and does not intersect tangentially with any previously inserted path. Notice that, in line with the approach presented in [63, 91], determining the tangential intersection between two candidate paths is straightforward: it involves a simple check to verify whether two paths pass through the same vertex in non-orthogonal directions.

An example of the step taken by this sampling procedure is depicted in Figure 4.12.

Path removal strategy As mentioned, paths are tested for removal in inverse order of creation, until only paths deemed necessary for the viability of the patch layout remain. Figure 4.13 shows an example. As a variant, we can test the removal of individual arches rather than entire paths. This results in a patch layout with T-junctions (see Figure 3.2).

4.2.5 Final Remeshing

Finally, the last produced morphed mesh is remeshed to obtain the final discretized ruled surfaces (which are developable by construction). The final meshing is obtained by sampling over each arch at the boundary between (up to) two patches, all the intersections with every ruled line on either side. Intersection points on opposite sides of a patch are connected, creating mesh edges that correspond to the selected ruling lines. Additional edges are inserted to triangulate the entire mesh, prioritizing edges with a direction similar to the ruling lines. See Figures 4.14, 4.6-d,e for examples of this remeshing, and of the corresponding isometric flattening.

Final patch-shape optimization Over highly non-developable parts of the mesh, the orientation of the ruling lines determined by our parametrization-based approach (Figure 4.15, a) will inevitably undergo high distortion during the morphing (Figure 4.15, b). While this does not compromise the ability of the morphing to reliably assess the viability of a patch layout, the shape of patch boundaries can be adversely affected. As a simple way to counter this, we apply an extra optimization, only in the last iteration, consisting of repeating the entire morphing procedure a second time, including the determination of the ruling lines, this time targeting the already morphed mesh. (as shown in Figure 4.15.c).

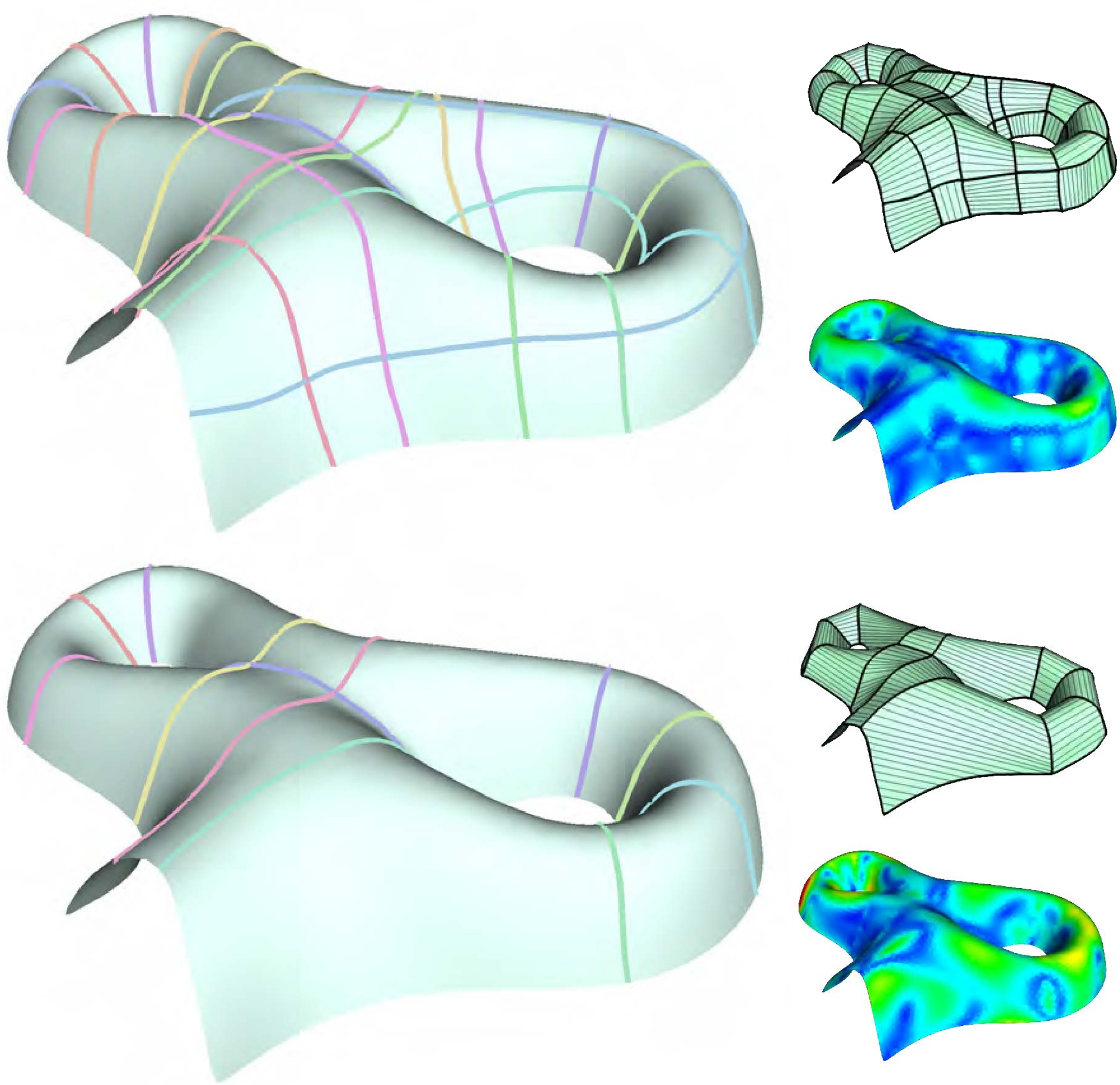


Figure 4.13: The few final steps of path removal cycle, with the relative approximation error.

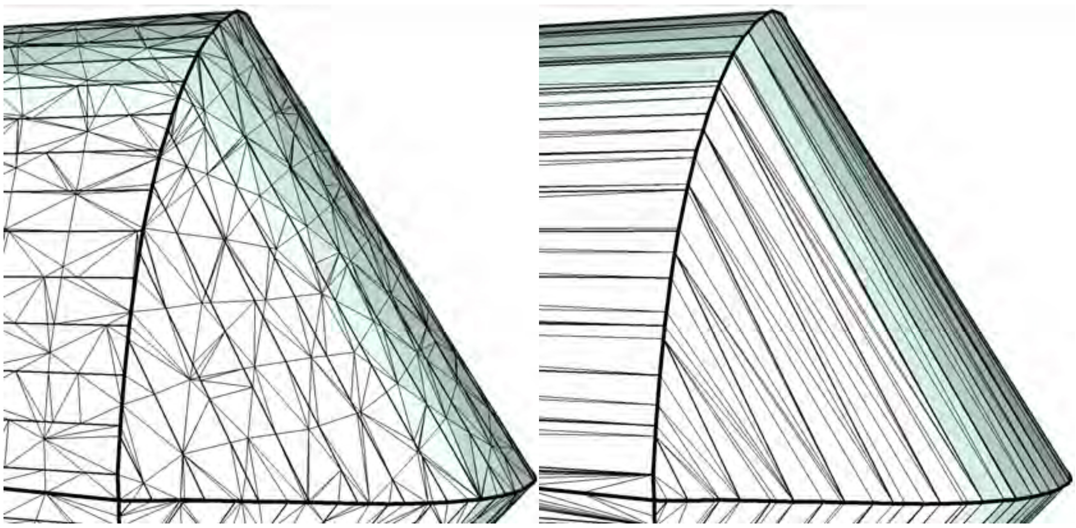
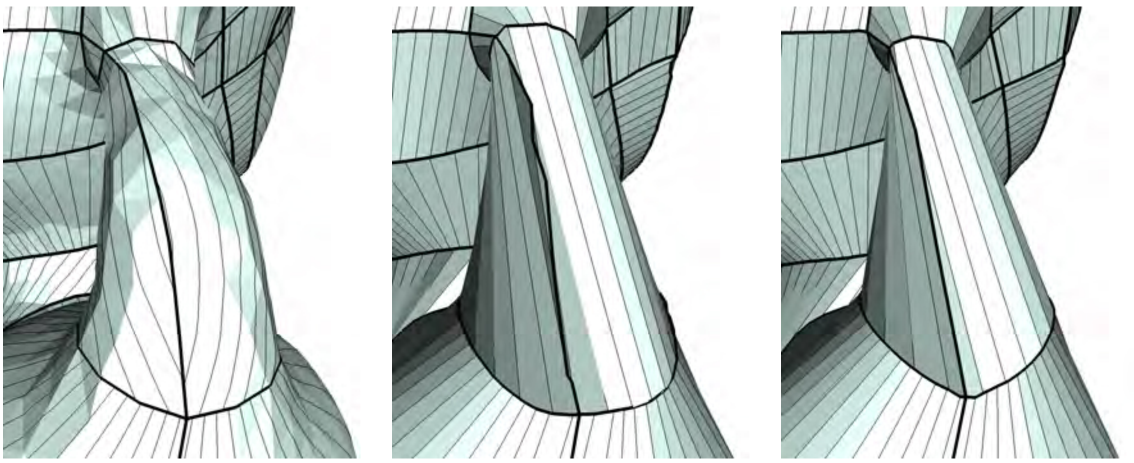


Figure 4.14: The morphed original mesh (left) and the final discretization (right).



(a) (b) (c)

Figure 4.15: Final shape optimization: before the morphing, ruling lines traced over a highly non-developable surface (a) can adversely affect the regularity of the patch boundary after the morphing (b); this is countered by final patch-shape optimization (c).

CHAPTER 5

RESULTS

Since our applications are in bio and digital fabrication, we will present their results separately in section 5.1 and section 5.2, respectively. In section 5.1, firstly, we will show the results of the SPH simulation without extracting the surface of the spheroids and then the results of the fusion after extracting the surface. These results are obtained from a simulation of two and three spheroids during their fusion. For the digital fabrication part, in section 5.2, we will present the results of our algorithm for a wide range of input meshes, including geometrical, architectural and manufactures.

5.1 In-silico simulation of cardiac spheroids fusion results

We provide the results of our simulations with different parameters. All code is implemented in Python and C++ programming languages. For computation, we have used a notebook with AMD Ryzen PRO 4750U and Radeon Graphics 1.70 GHz. OpenGL platform and OpenVDB [76] have been utilized for visualization and surface extraction. Also, the results were rendered with the Blender software.

We defined a range of changes for each parameter to see the effect of that parameter on the simulation result. For this purpose, we have summarized the range of each parameter we have used in our simulation in Table 5.1.

In our experiments, stiffness, rest density and viscosity coefficient are only used in

Table 5.1: Simulation parameters and their change steps

Parameter (abbreviation)	Range	Change step
Young modulus (YM)	0.5-3.5	0.5
Poisson ratio (PR)	0.35-0.49	0.04
Volume constant (VolC)	0.1-2	0.5
Plastic limit (PL)	0-0.9	0.1
Stiffness (S)	0.1-3	0.5
Rest density (RD)	0.1-2.5	0.4
Viscosity coefficient (VC)	0.1-3	0.5

the fluid model, and the rest of the parameters are common between the fluid and the solid-based models.

As mentioned in the implementation section, we performed 46 simulations with different combinations of the parameters in Table 5.1 to simulate different behaviour of spheroids during the merging process. The primary purpose of this series of simulations is to show the flexibility of the proposed physically-based framework in matching the different parameters involved in the spheroid merging process. In the following subsections, we show just some selected results. Also, the developed framework is generic so that it is able to be used for simulating different applications with the same material properties or with more than two spheroids (Figure 5.1).

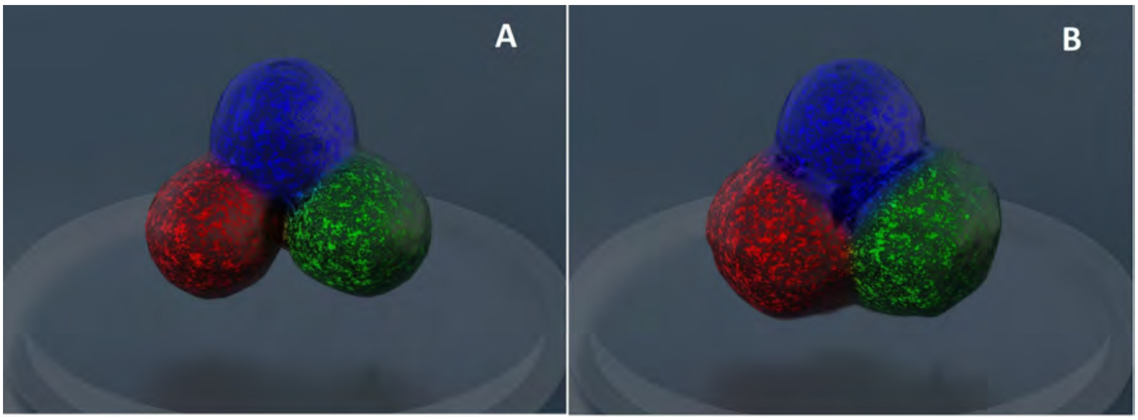


Figure 5.1: Simulation of fusion of three spheroids: a) frame 1 of the simulation (beginning of the fusion), b) frame 100 of the simulation (middle of the fusion)

.

5.1.1 Simulations without surface

The first set of results only contains particles (without surface reconstruction) to show how the simulation is working. The simulation parameters and the full video link of each simulation are provided in the caption of the figures. Based on the video, spheroids first go toward each other and then start bouncing back and expanding a little bit due to the elastic force. (Figure 5.2 and 5.3).

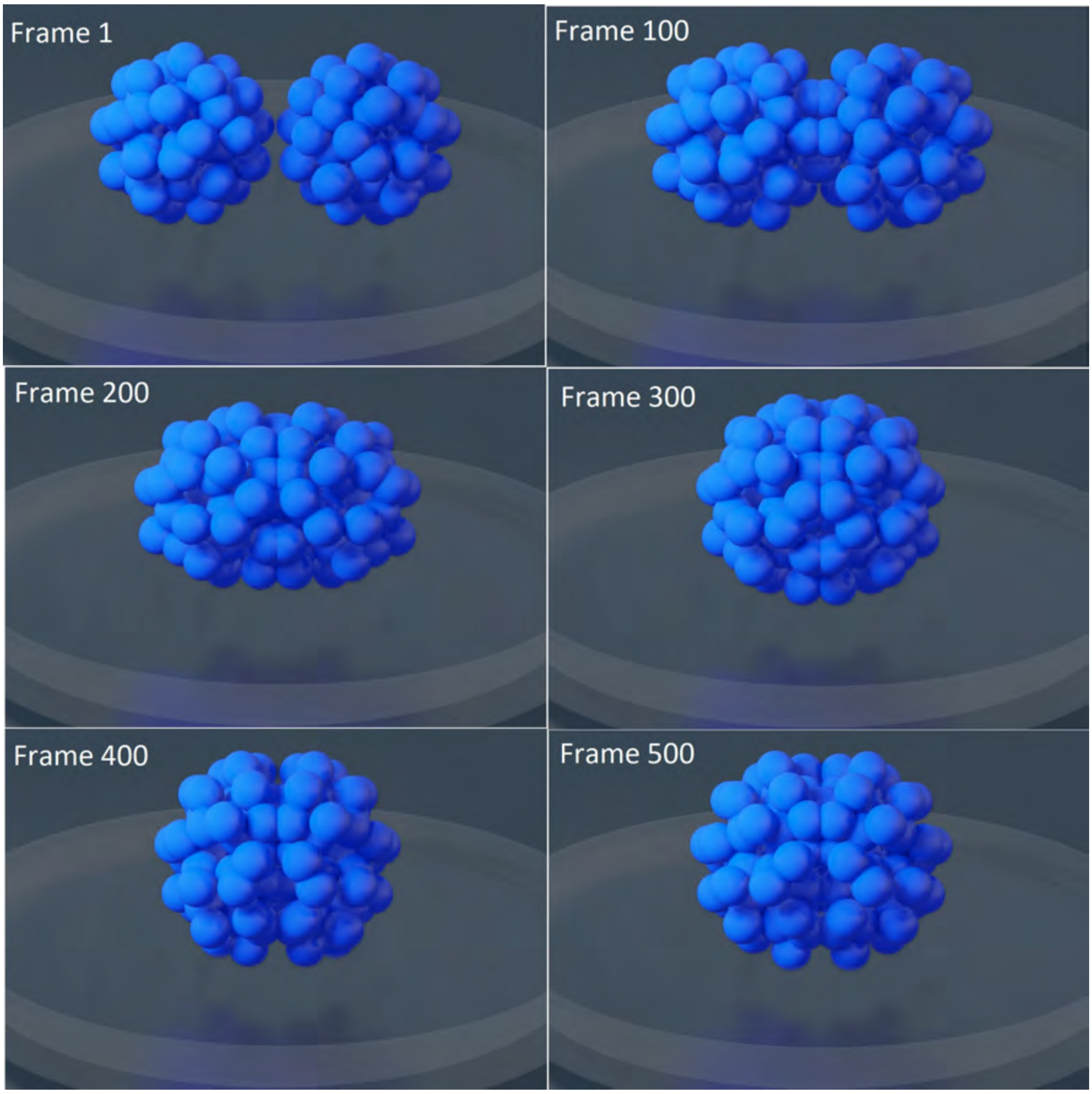


Figure 5.2: Fusion process of spheroids in 500 simulated frames using the fluid-based SPH model and without surface reconstruction (parameters: YM=3, PR=0.4, VolC=1.4, PL=0.1, S=1, RD=1, VC=1). The full video is available here

.

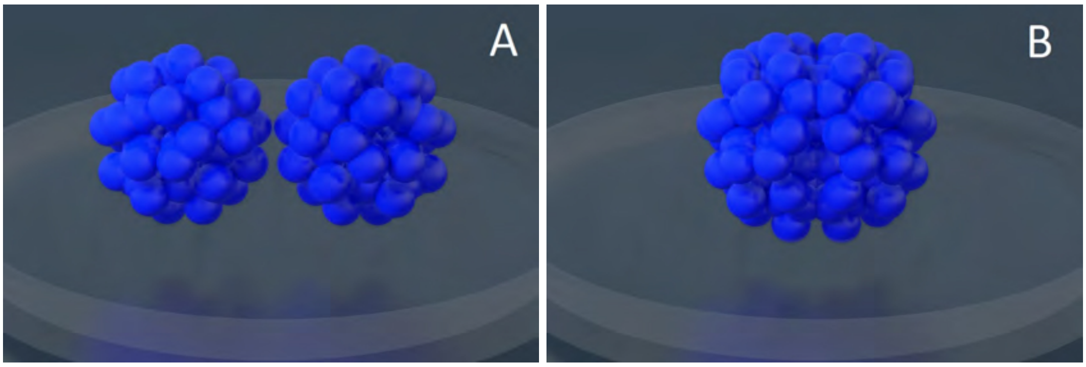


Figure 5.3: A: The beginning of the simulation when spheroids move towards each other. B: The end of the simulation when particles are stabilized (parameters: YM=3, PR=0.45, VolC=1.4, Pl=0.3, S=1, RD=1.5, VC=1). The full video is available here

.

5.1.2 Simulations with surface

In the next set of results, we will show the fusion process of spheroids with their surfaces. This set of results can mimic the actual process using appropriate parameters. Therefore, we will present at least one example of each model (solid or fluid) with different simulation parameters.

In Figure 5.4, selected frames of a solid-based simulation are shown. In this set of results, the spheroid surfaces have been extracted and shown with different colours (red and green). As expected, the surface's behaviour depends on moving particles connected to the surface.

Based on our surface extraction algorithm, surfaces start joining at the beginning of the simulation to form a final shape of the fused spheroids. However, we can see different kinds of deformation on the surface during the simulation caused by the interior interaction of the spheroid particles.

Figure 5.5 shows two selected simulations of solid-based models. The main differences between these two simulations are plastic limit and Poisson ratio parameters which are 0.1 and 0.45 for Figure 5.5.a, and 0.9 and 0.35 for Figure 5.5.b, respectively. The effects of these parameters are clearly seen, as the volume of spheroids in Figure 5.5.a has decreased a lot due to having more plastic properties. Hence, we can see in Figure 5.5.a, there is more plastic deformation in the spheroids.

Based on the theory of continuum mechanics, decreasing Young's modulus or increasing the Poisson ratio will make a more elastic object. To prove this fact, in Figure 5.6, we compared four simulations in which we decreased Young's modulus and increased Pois-

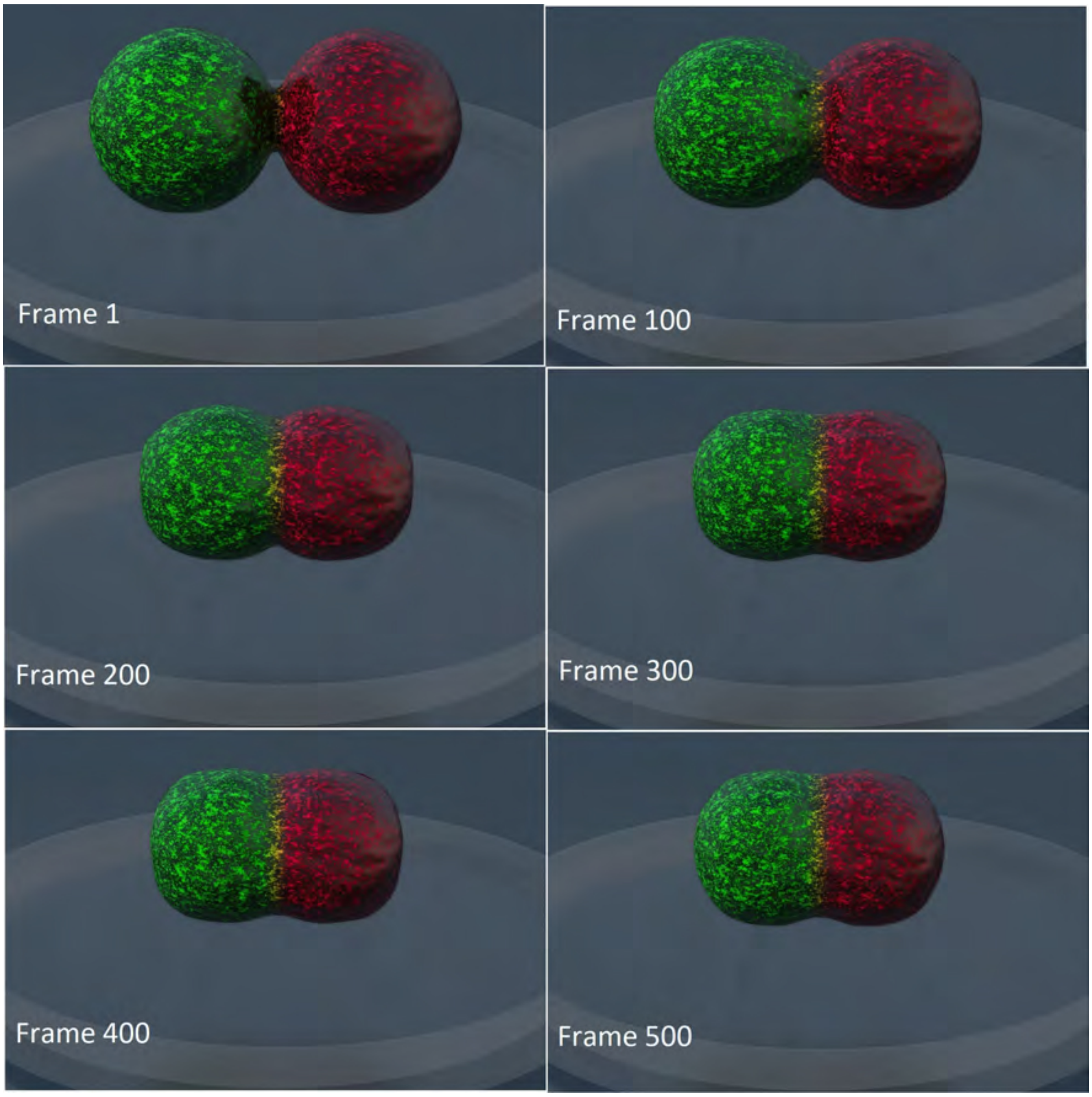


Figure 5.4: Fusion process of spheroids in 500 simulated frames using the solid-based SPH model with surface extraction (parameters: YM=1, PR=0.45, VolC=1.4, PL=0.1). The full video is available here.

son ratio from top left to bottom right, respectively. It can be seen that with decreasing Young modulus and increasing Poisson ratio, the volume and area of spheroids change significantly.

It is worth mentioning that gravitational force is applied to each spheroid particle toward the other spheroid's centre. As a result, due to the spheroids' visco-elastic properties, this gravitational force causes the spheres to crumple and lose volume during the fusion process. However, surprisingly, we see an increase in volume and area in the last

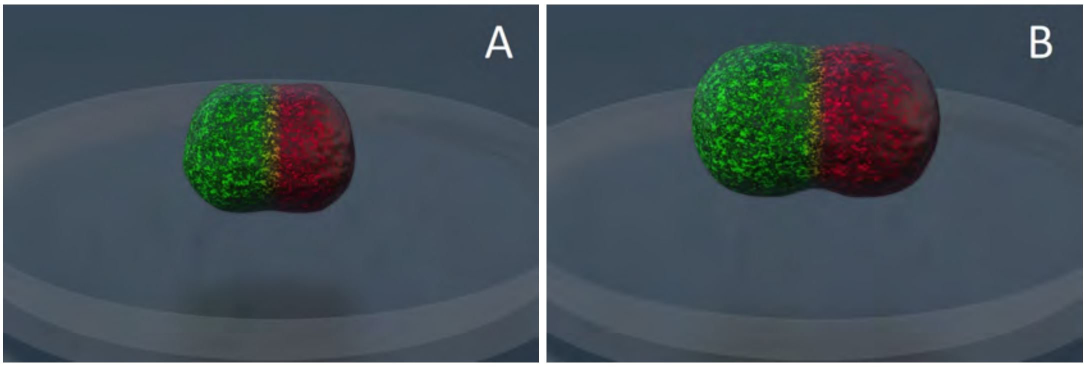


Figure 5.5: A: Complete fusion of spheroids made by the visco-elastic solid model (parameters: YM=3, PR=0.45, VolC=1.4, PL=0.9). The full video is available here. B: Half fusion of spheroids made by the visco-elastic solid model (parameters: YM=3, PR=0.35, VolC=1.4, PL=0.1). The full video is available here.

simulation. This can be due to the dominance of the elastic force, which tends to return the spheroid to the initial shape, over the gravity force, which tends to crumple the spheroids. Because in the last simulation, Poisson's coefficient and Young's modulus have their maximum value, and the spheres have assumed the most elastic state (Figure 5.6).

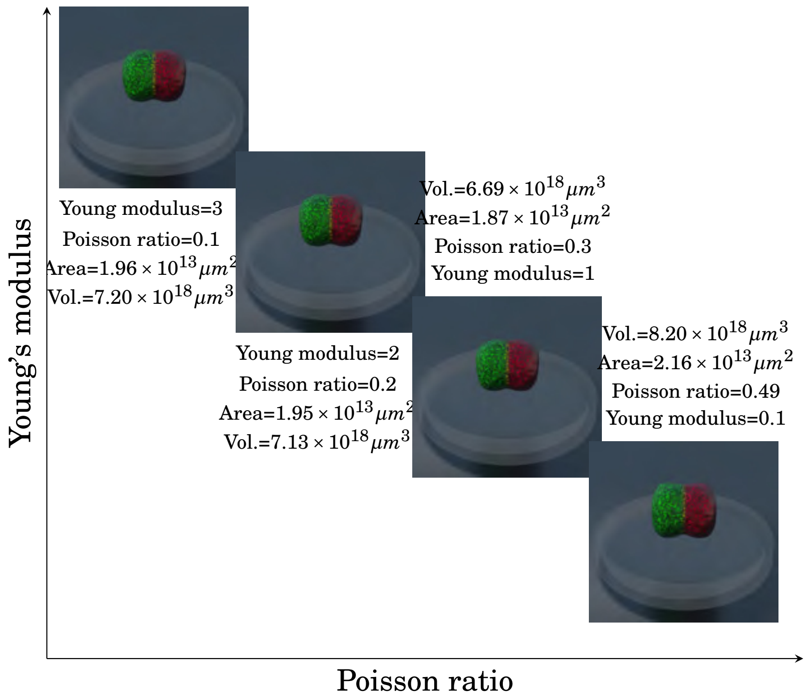


Figure 5.6: Effect of increasing Poisson ratio and Young's modulus on total area and volume (total deformation) of spheroids.

The results of simulating as visco-elastic fluids for full (Figure 5.7.a) and partial (Figure 5.7.b) fusion are shown in Figure 5.7. The meaning of partial or full fusion is the degree of interpenetration of the spheroids, which can be adjusted by varying the attraction force between them and an opposing external force caused by the hydrogel. The parameter values of each simulation can be seen in the caption of the figures. In this result, spheroids have an egg-shaped deformation in complete fusion. This is because of the continual gravitational force between spheroids particles, which forces the particles to spread along the edge of spheroids collision. The other important difference between Figures 5.7.a and 5.7.b is the stiffness parameter which is 0.5 for Figure 5.7.a and 1 for 5.7.b. That is why we can see a little shrinkage in spheroid volume in Figure 5.7.a.

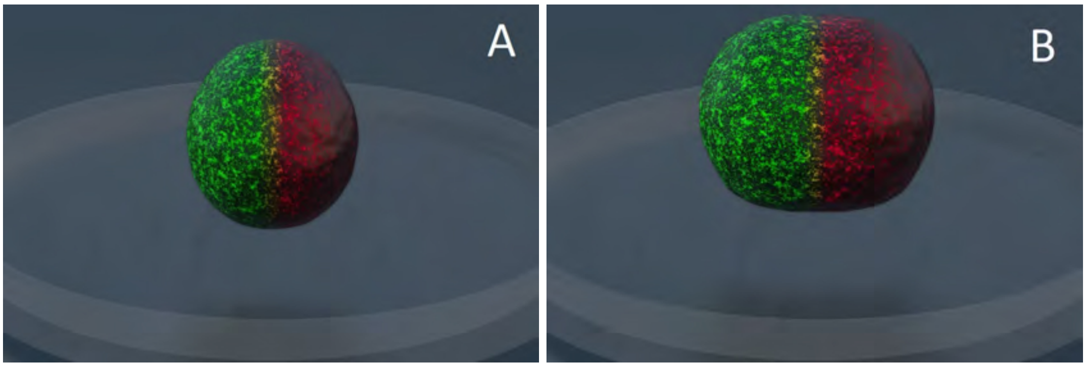


Figure 5.7: A: Complete fusion of spheroids made by the visco-elastic fluid model (parameters: YM=3, PR=0.45, VolC=1.4, PL=0.1, S=0.5, RD=1, VC=1). The full video is available here. B: Half fusion of spheroids made by the visco-elastic fluid model. (parameters: YM=3, PR=0.45, VolC=1.4, PL=0.1, S=1, RD=1, VC=0.5). The full video is available here.

To better understand the effect of stiffness and constant volume parameters, Figure 5.8 shows the last frame four simulations in which we have an increase in volume constant and stiffness parameters from bottom left to top right. As it might be hard to see the amount of deformation in the total configuration due to the size of the images, we have measured the area and volume of the total configuration at the beginning and end of each simulation. Of course, at the beginning of all simulations, the area and volume of the configuration are fixed and equal to $3.88 \times 10^{13} \mu m^2$ and $1.66 \times 10^{19} \mu m^3$, respectively. Increasing either volume constant or stiffness will increase the volume and area of the total spheroid configuration except for the last picture (top right of Figure 5.8). This exception is due to the balancing of the forces and stability in the position of the particles. The volume constant, stiffness, area and volume of each simulation are written at the bottom or above each picture.

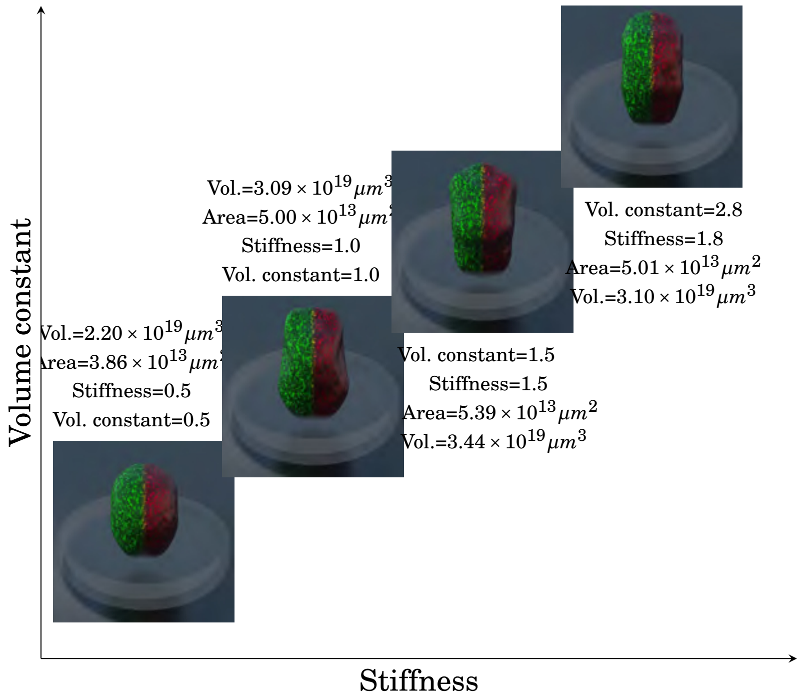


Figure 5.8: Effect of increasing stiffness and volume constant on spheroids total area and volume (total deformation).

Another kind of simulation is when we applied elastic collision between particles of two spheroids. In this case, when two spheroids touch each other, their particles collide and bounce back due to the elastic collision force. Then, the attraction force brings them back together, and they collide again. This process will repeat until the particles get balanced in a final position. As shown in Figure 5.9, when we have elastic collision in our simulation, spheroids do not get entirely merged as particles of each spheroid do not go through each other. Indeed, particles get spread along the boundary of collision.

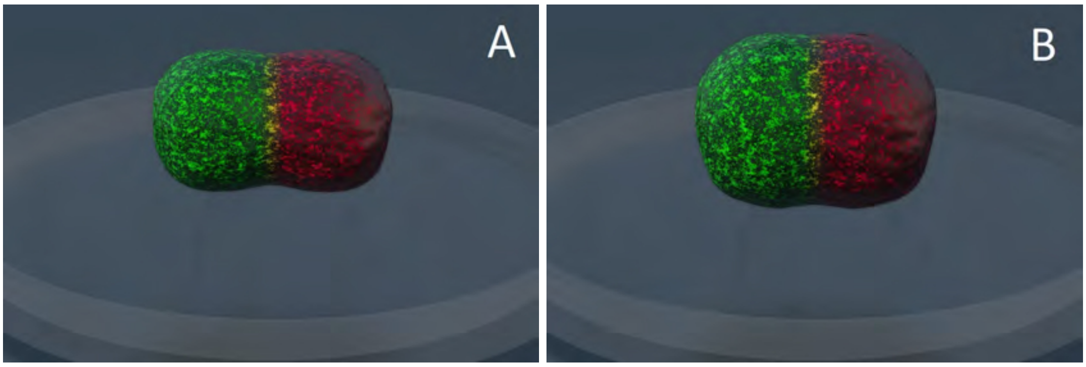


Figure 5.9: A: Merging of spheroids made by the visco-elastic solid model with less plastic deformation (YM=0.5, PR=0.45, VolC=1.4, PL=0.1). The full video is available here. B: Merging of spheroids made by the visco-elastic solid model with more plastic deformation (YM=3, PR=0.45, VolC=2, PL=0.5). The full video is available here.

In the simulations of Figure 5.9, we have used two different plastic limit values. It is evident that the simulation with a larger plastic limit shows more plastic deformation (Figure 5.9.b).

5.2 Developable patch decomposition results

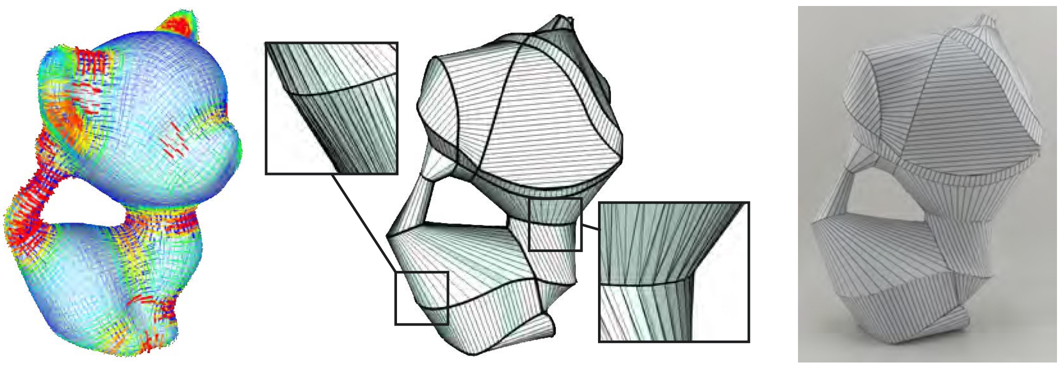


Figure 5.10: Starting with a triangular mesh with a curvature-aligned cross-field (left), our system produces a decomposition into a seamless set of ruled surfaces (middle) that serves as a blueprint to easily fabricate the surface by means of inextensible materials, such as paper (right).

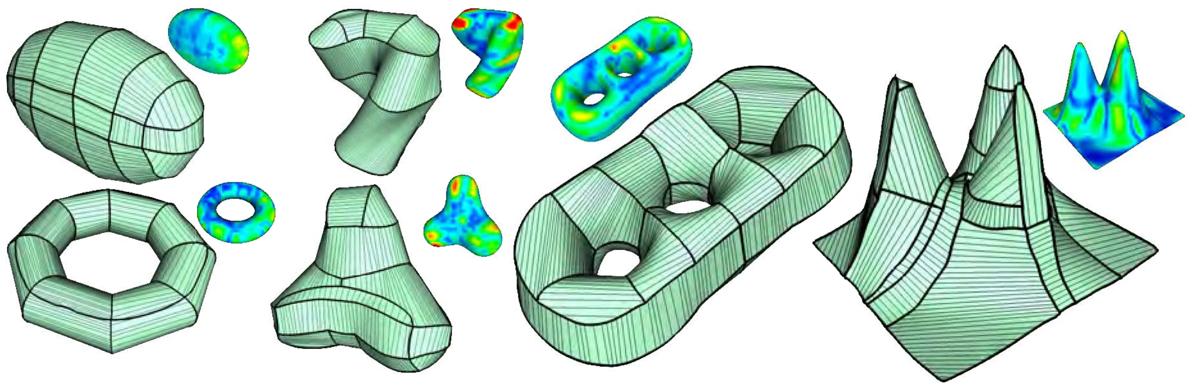


Figure 5.11: The layout of ruling patches for a selection of regular geometry. The figure also incorporates the distance error to the original mesh. The maximum error is highlighted in red, corresponding to 2.5% of the input meshes bounding box diagonal.

We successfully tested our method on a wide range of different input geometry. The results in Figure 5.10 show how our method performs on fairly regular input shapes. Notably, we observe satisfactory behaviour, with outcomes closely resembling those achievable through manual modelling. It's worth emphasizing that our method operates entirely automatically, making it particularly noteworthy that it can yield globally consistent and regular results. A pivotal factor contributing to the success of our approach is the incorporation of the guiding cross-field, which serves as a valuable abstraction for capturing the overall structure of the input shape.

We implemented our method in an Apple M1 Max notebook with 32 GB of RAM. The method performs at the order of seconds to minutes. The examples shown in the paper took from 6 seconds (the models in figure 5.11) up to 80 seconds (for the complex models shown in Figure 5.17). These times allow users to adjust the parameters e_{max}, searching for a preferred trade-off (Figure 3.1). In all the examples, we used a maximum error ranging from 1 to 5 % of the diagonal of the input bounding box.

Other output results shown in Figure 5.11, 5.15, 5.16, and elsewhere in the paper indicate that our method performs well on fairly regular input shapes, yielding consistently good results. In the attached materials, we make all input meshes and the final design (with the 2D layouts available both as UV maps of the meshes and as separated SVG files).

A pivotal factor contributing to the success of our approach is the incorporation of the guiding cross-field, which serves as a valuable abstraction for capturing the overall structure of the input shape.

We test our method on shapes inspired form various specific contexts where developable ruled surfaces can find application, such as architectural design, with both symmetrical (Figure 5.14), and non-symmetrical (Figure 5.12) examples, and fashion design (5.13). In all cases, our algorithm generates high-quality consistent rulings that smoothly flow in the principal curvature direction. Although our method is not specialized for deriving garment pattern layouts [89], it produces convincing paper patterns.

We can observe that our algorithm generates high-quality consistent rulings that smoothly flow away from the symmetric axis along the principal curvature. The final result is a visually pleasing approximation.

Finally, Figure 5.17 showcases a particularly intricate model designed to highlight the robustness of our proposed method.

5.2.1 Physical realizations

We fabricated several physical exemplars to demonstrate the applicability of the generated design.

First, we fabricated a paper covering of a vase-lion, Figure 5.18. The statue highlights the advantage of our discrete patches in allowing both bending and twisting to approximate the target mesh. The advantage of twisting can be appreciated in the region under the lion's mouth, where a single patch is split into two ends.

Our second example (Figure 5.19) showcases a potential application of our method towards construction. To physically realize the piece we 3D printed a rod-like support following the boundary of our patch layout. The ruled patches are then cut from cardboard and glued onto this supporting, resulting is a robust model with a very limited amount of supporting material.

Our last example demonstrates an application of our method towards the stylized fabrication of a lamp, Figure 5.20. In this case, the produced design naturally preserved the 3-way symmetry of the input, without explicitly enforcing it.

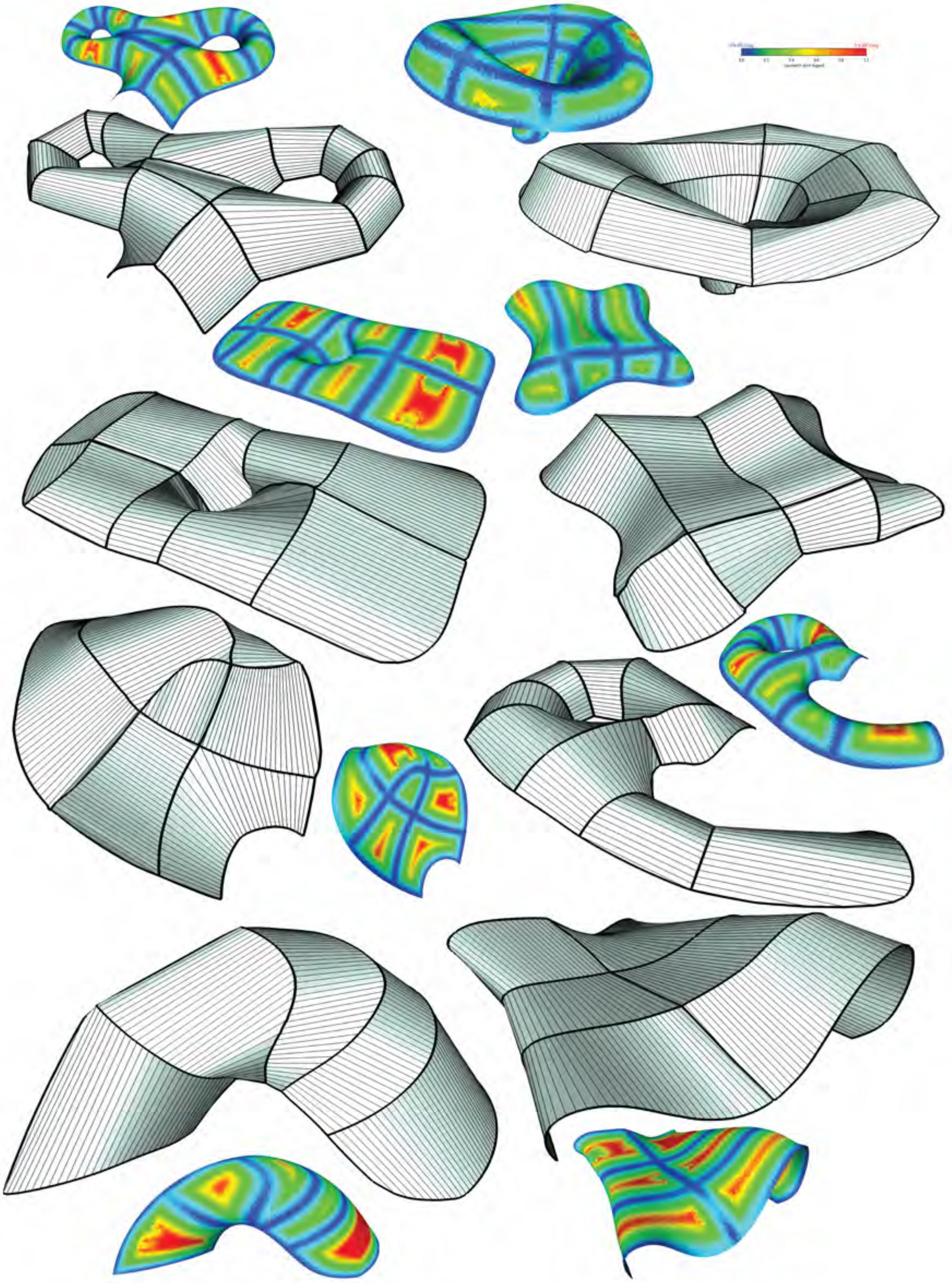


Figure 5.12: The layout of ruling patches for a selection of architectural models.

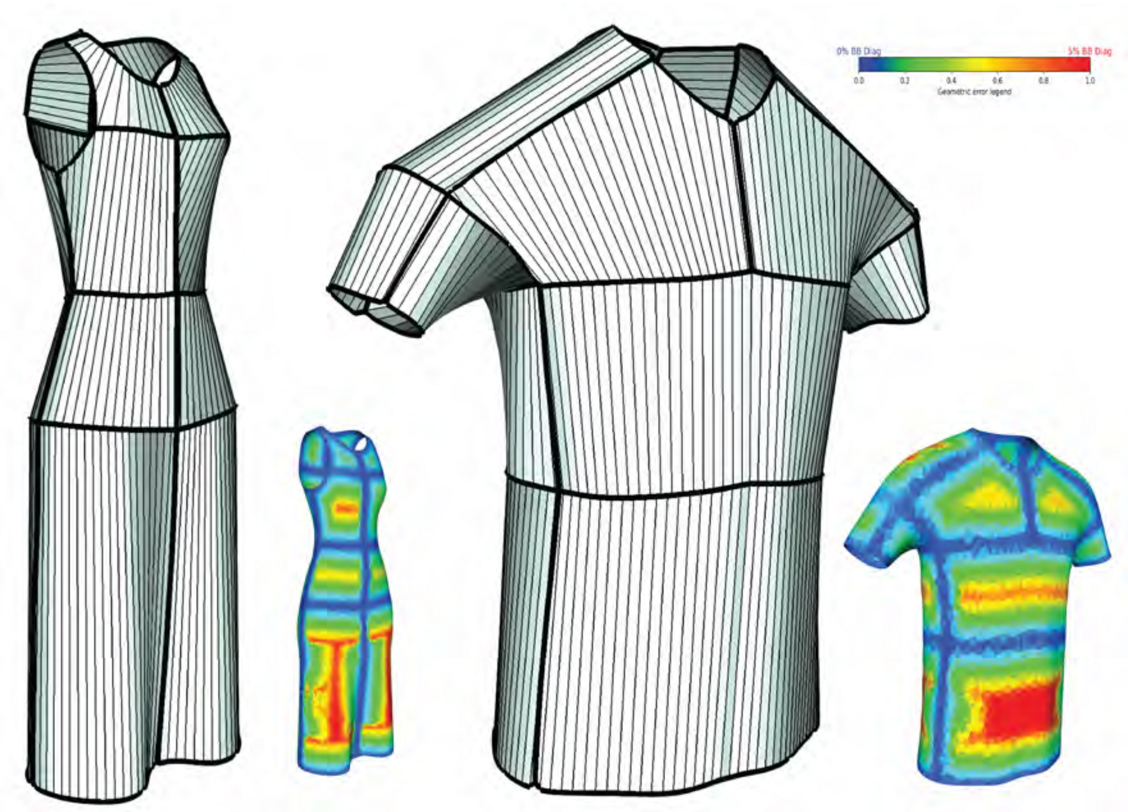


Figure 5.13: The layout generated for Fashion Design.

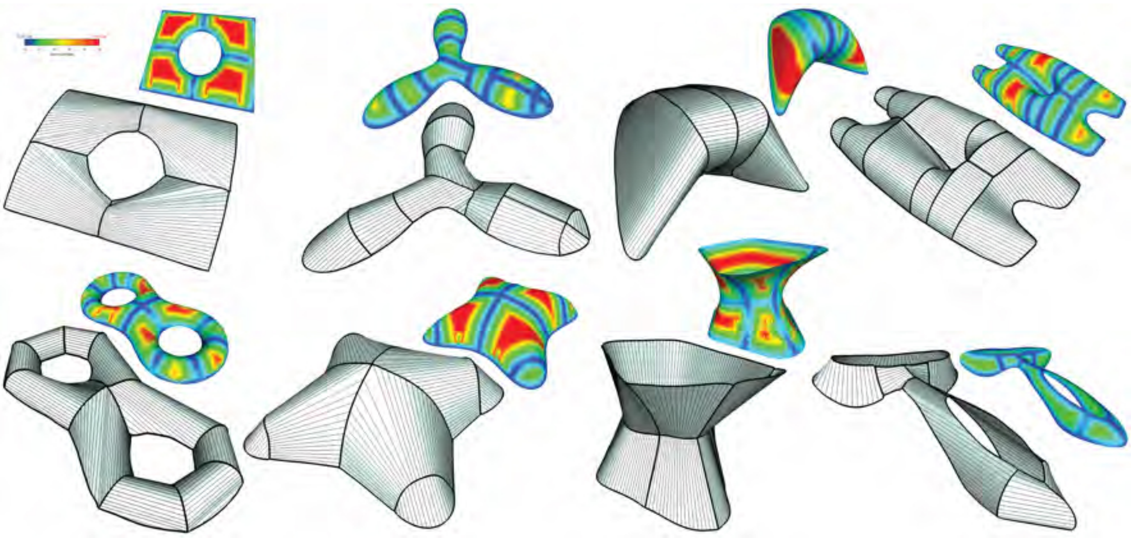


Figure 5.14: The layout of ruling patches for a selection of architectural models with symmetric decomposition.

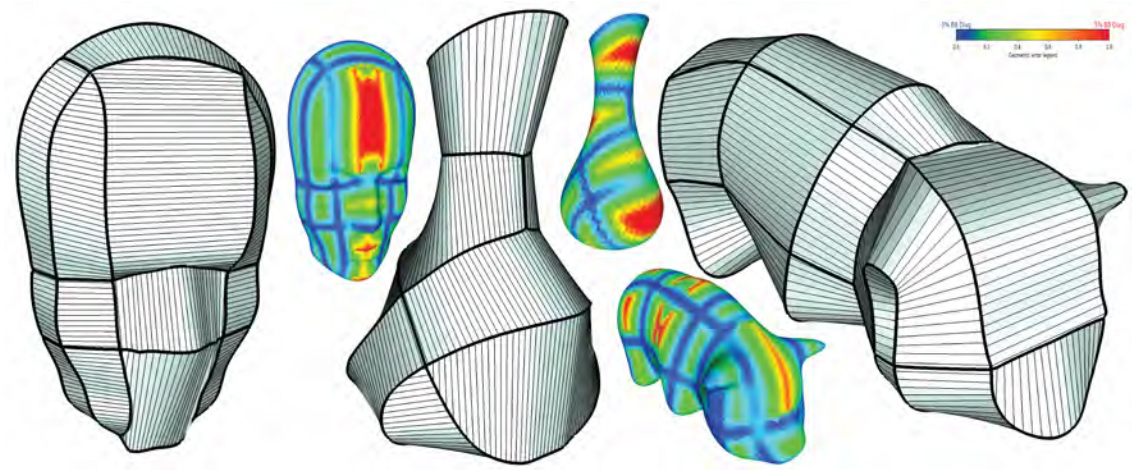


Figure 5.15: The layout of ruling patches for a selection of general models.

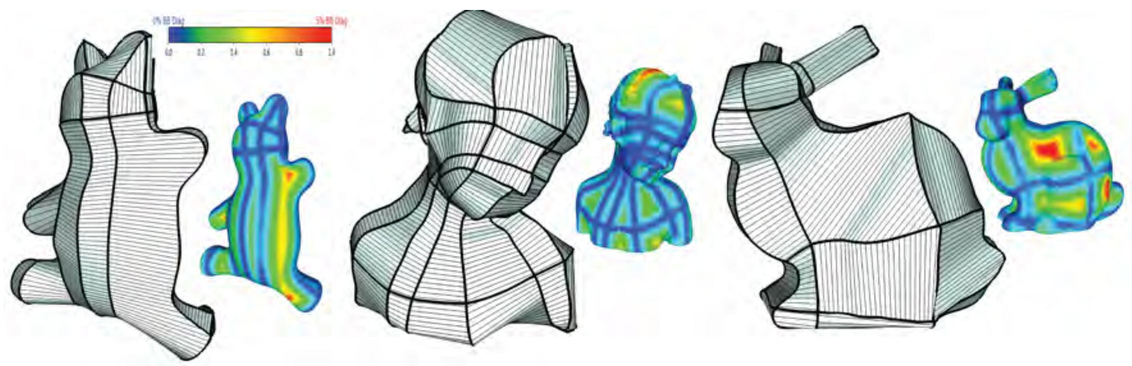


Figure 5.16: The layout of ruling patches for a selection of general meshes.

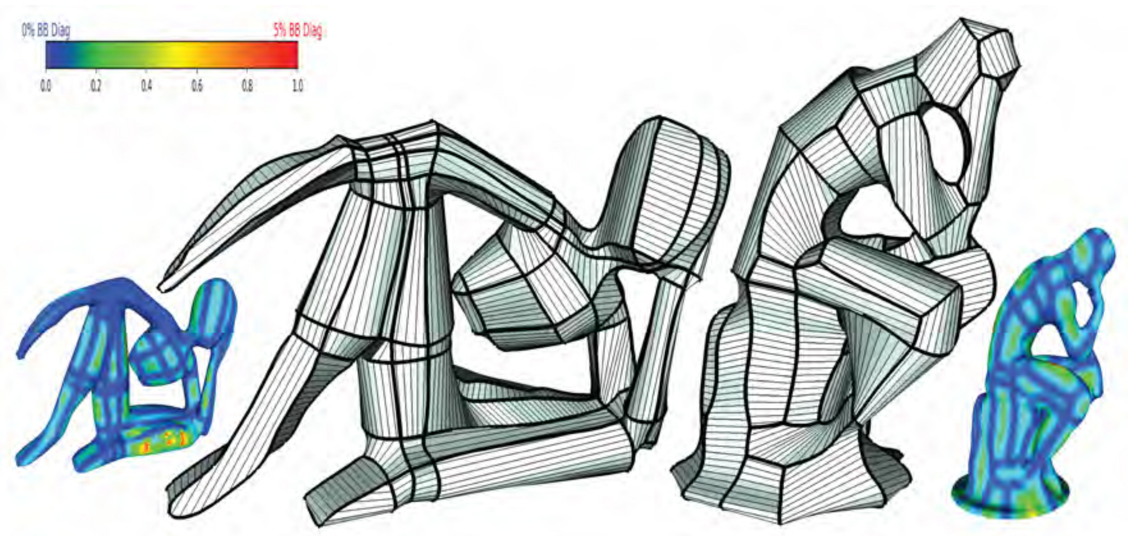


Figure 5.17: Discretized versions of two complex meshes showcasing the robustness of our patch generation.

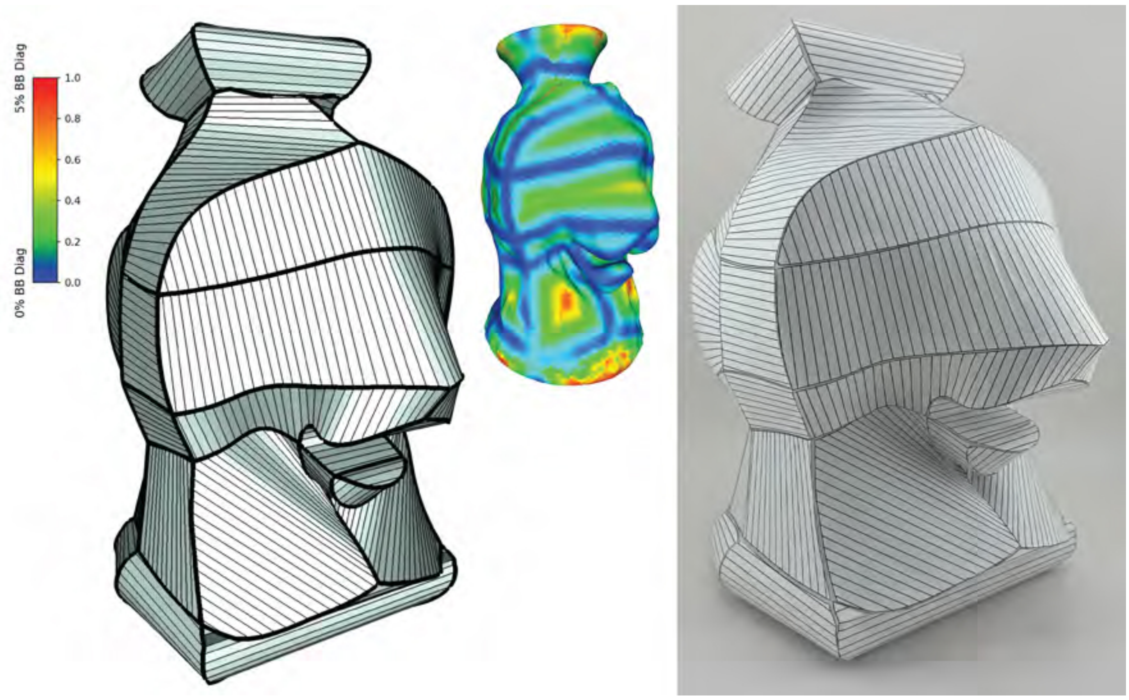


Figure 5.18: Fabricated lion statue composed of several complex patches exhibiting various degrees of bending and twisting.

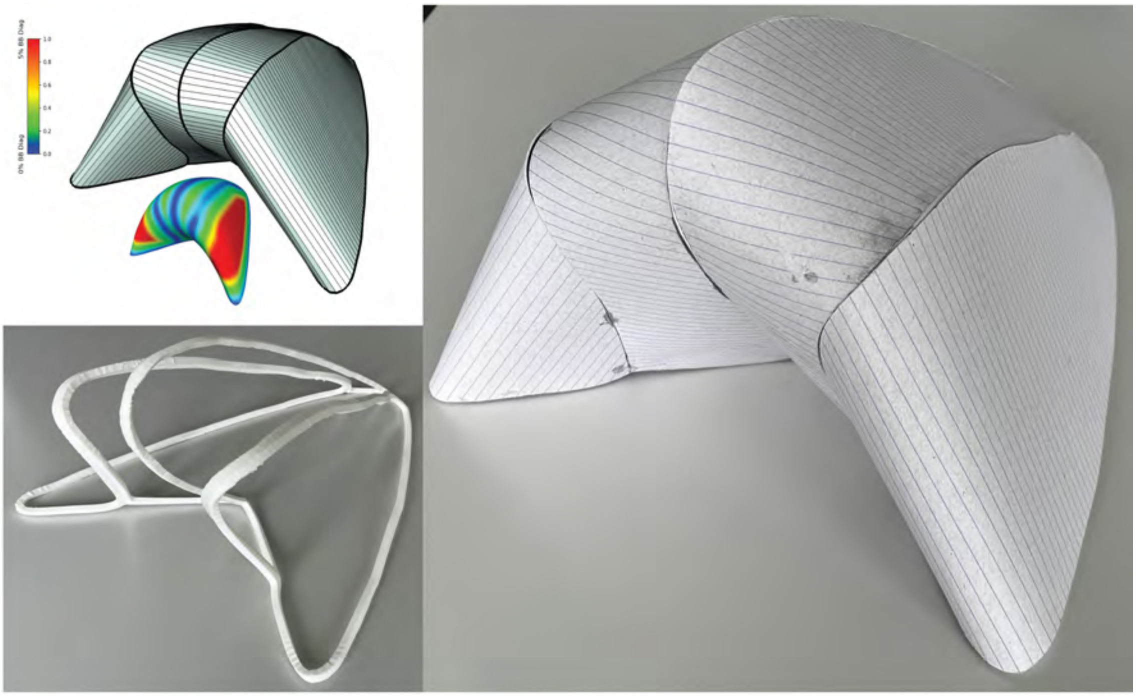


Figure 5.19: Thin rod-like scaffolds are used to reproduce an architectural piece by attaching large pieces of developable material to cover the roof.

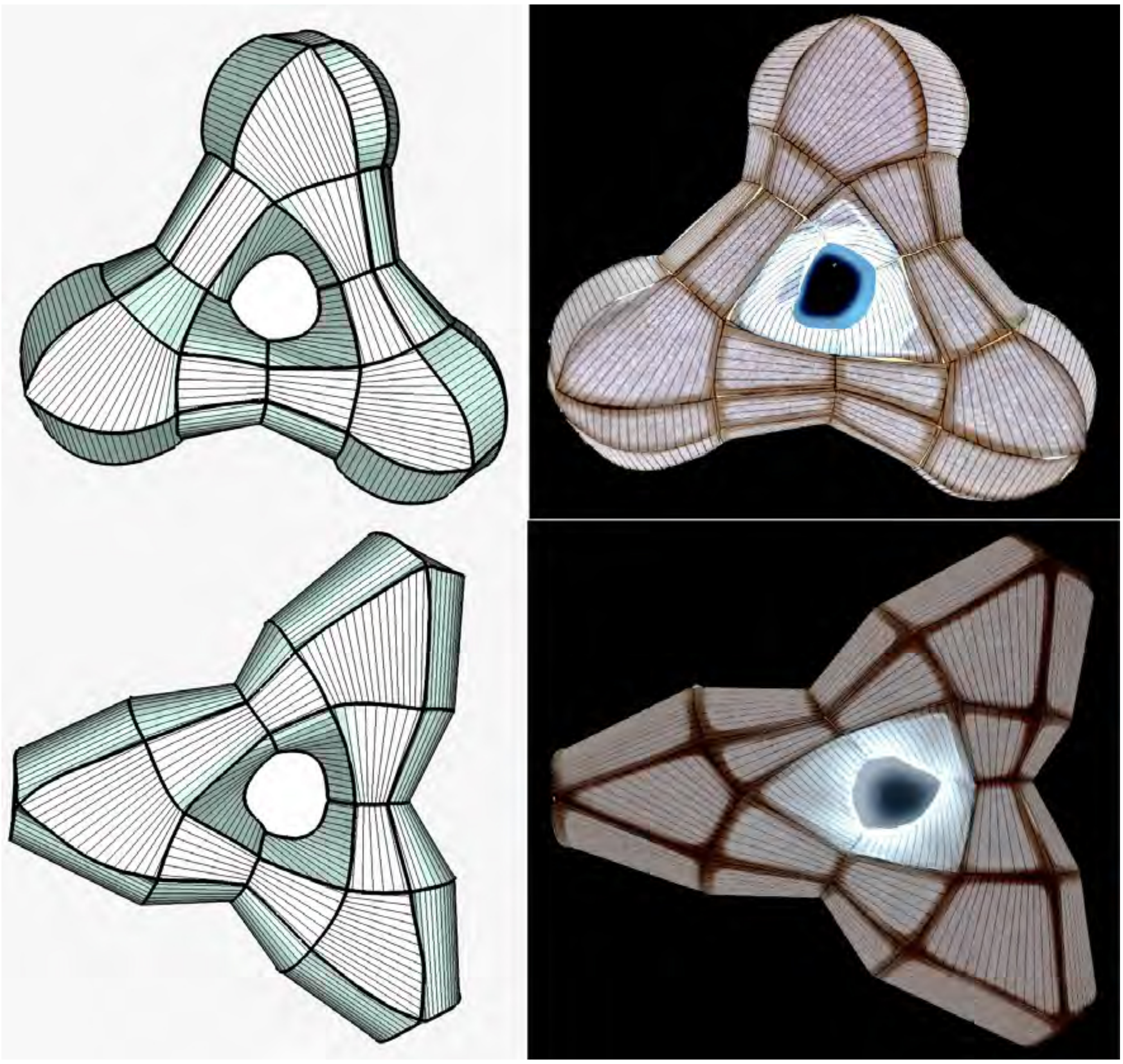


Figure 5.20: Fabricated paper lamp.

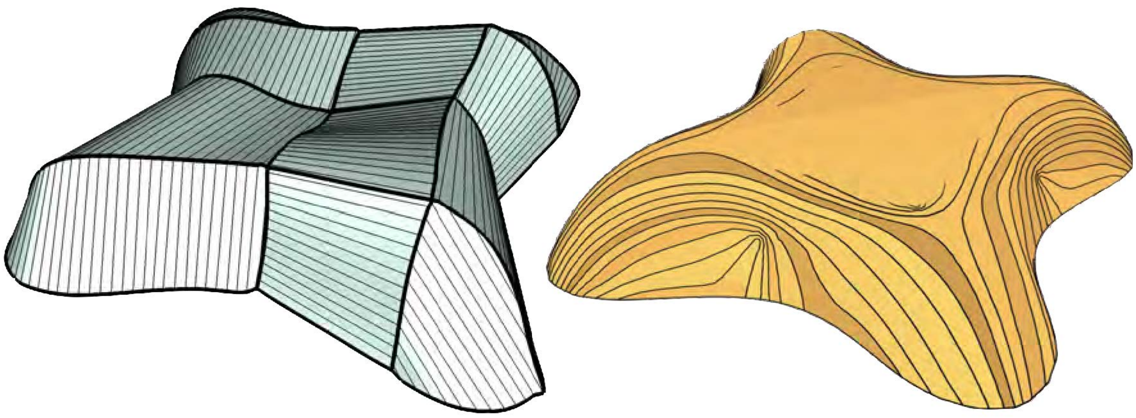


Figure 5.21: Comparison of our method with Dev2Qp [131]. As highlighted in the original paper, Dev2Qp is unable to provide a solution for the field shown on the right. In contrast, our method successfully derives a valid ruling patch layout (image courtesy of [131]).

CHAPTER 6

MODEL VALIDATION

6.1 In-vitro validation

In-vitro validation refers to testing our proposed method in real experiments and comparing its results in order to validate its accuracy and efficiency. Therefore, this is only applicable to the bio-fabrication case.

In order to validate the effectiveness of the presented mathematical models, we have conducted a series of practical experiments to compare the results obtained from the simulation with images obtained from the actual fusion of spheroids. We designed our experiments based on potential factors that might affect the fusion process namely: the Hydrogel type, the density of spheroids, and the distance between spheroids. In this section, we present our experiments and discuss the validation results.

6.1.1 Design of experiment

We have considered 36 different scenarios corresponding to four types of hydrogel, three different densities of spheroids and three different distances between spheroids. We conducted a full factorial experiment with three replications for each combination of parameters leading to a total of 108 experiments *in vitro*.

First, three types of cells were cultured:

- Human Cardiac Fibroblasts Cells (HCFCs)

- Human Coronary Artery Endothelial Cells (HCAECs)
- Human Cardiac Myocytes Cells (HCMCs)

Each type of cell was grown using its specific media and growth protocol. After growing a sufficient number of cells, we made spheroids by combining them. A 384-hanging drop culture plate was used to culture cells in it. Based on previous studies, the three types of cells, HCFCs, HEAECs and HCMSs, are combined with a ratio of 1:1:2 to form spheroids. The culture plate is incubated at 37 degrees (average human body temperature) for 4 to 5 days to generate human spheroids. Cardiac spheroids usually have a diameter of around $100-200\mu m$ and volume of around $4.2\times10^6-3.4\times10^7\mu m^3$. For more details on this process, please refer to Polonchuk et al. [97].

Hydrogel

The stiffness of the hydrogel in which the spheroids get printed is usually the most critical factor, because its stiffness may prevent the spheroids from approaching each other. In order to produce different types of fusion, we have prepared four types of hydrogel based on their stiffness:

- Hydrogel 1: alginate/gelatin
- Hydrogel 2: alginate/gelatin+1% silk
- Hydrogel 3: alginate/gelatin+2% silk
- Hydrogel 4: alginate/gelatin+3% silk

Adding more silk will make a stiffer hydrogel and make spheroid movement harder.

Density of the spheroids

We measured the number of combined cells per spheroid while forming the spheroids. In this way, we can obtain different spheroids densities. For this study, we considered three different spheroids densities:

- spheroids with 10000 cells
- spheroids with 20000 cells
- spheroids with 30000 cells

Distance of the spheroids

We considered three different distances between pairs of spheroids:

- very close distance ($dist. \leq 400\,\mu m$)
- close distance ($400\,\mu m \leq dist. \leq 700\,\mu m$)
- far distance ($dist. \geq 700\,\mu m$)

6.1.2 Results of validation

After placing spheroids in a well-plate, we monitor their fusion within six days by taking an image at different times. Our observations show that spheroids behave differently depending on the distance and the environment they are in. A detailed investigation of this hypothesis requires statistical tests that it is beyond the scope of this article. However, examination of the images shows that some spheroids do not fuse at all, some have little fusion, some have moderate fusion, and some are entirely fused. Most complete fusions occur in hydrogel 1 (alginate/gelatin), which is less stiff than other hydrogels. In contrast, spheroids fusion can be seen among all densities of spheroids. In terms of the distance between spheroids, all fusion occurred among the spheroids with very close distances. Hence, it is obvious that the distance is a very significant factor in our experiments and all fusion results presented in Sec. 6.1.4-6.1.6 are belonged to the “very close distance" category.

6.1.3 No fusion

Some of the spheroids in the experiments did not fuse at all. The reason can be the time, hydrogel and distance. Most fusions happen within the first few minutes of the spheroids being close to each other. Since placing the spheroids in a well plate is a relatively time-consuming process, it may prevent their fusion during the experiment. Because according to our observation, fused spheroids start their fusion at the very beginning of being beside each other, and if they are not close enough, the chance of fusion decreases as time elapses. Stiffer hydrogel makes the fusion harder - as the second factor that affects the fusion process. The last factor is distance. Our observations show that all fusions have happened with spheroids that are very close to each other. Apparently, the gravitational force between them is weaker than the viscosity force of the hydrogel. Below it can be seen two examples of non-fusion of spheres over six days (Figure 6.1).

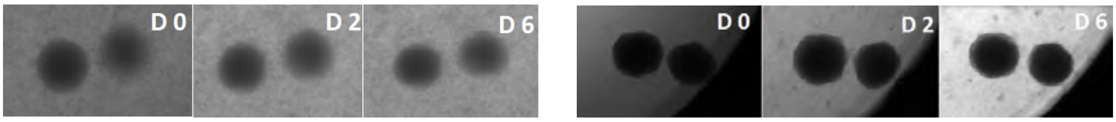


Figure 6.1: An example of two spheroids that are relatively close but are not fused within six days: a) Environment: hydrogel 1, density: 20000 cells, distance: close, b) Environment: hydrogel 2, density: 30000 cells, distance: close.

6.1.4 Low Fusion

Another case of fusion is when there is little fusion between the spheroids. Based on our observations, this mode is the most common among the fusion types, especially when spheroids are embedded in a hydrogel (Figure 6.2 and Figure 6.3). The proposed simulation framework can successfully simulate this type of fusion in different cases. Table 6.1 compares experiment and simulation parameters of Figure 6.2 and 6.3.

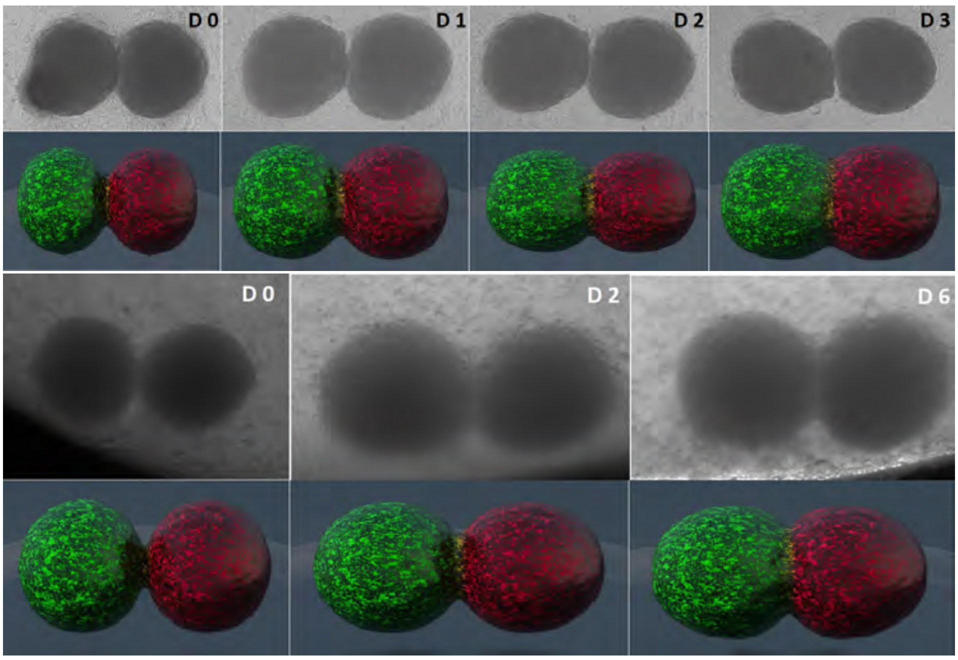


Figure 6.2: Two examples of a pair of spheroids that have low fusion: a) environment: hydrogel 1, density: 10000 cells, b) environment: hydrogel 1, density: 10000 cells, c) environment: hydrogel 2, density: 20000 cells.

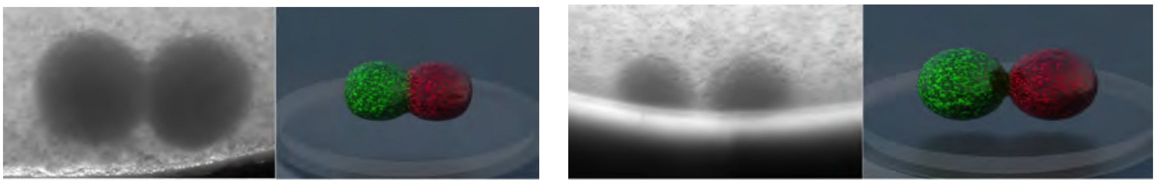


Figure 6.3: Two examples of two spheroids that have low fusion at the sixth day: a) Environment: hydrogel 2, density: 20000 cells, b) Environment: hydrogel 3, density: 20000 cells.

Table 6.1: Comparing experiment parameters and simulation parameters for low fusion

Figure	Experiment parameters	Simulation parameters
Figure 6.2.a	Environment: hydrogel 1 Density: 10000 cells	Young modulus: 3 Poisson ratio: 0.35 Volume constant: 1.4 plastic limit: 0.1
Figure 6.2.b	Environment: hydrogel 1 Density: 30000 cells	Young modulus: 2.6 Poisson ratio: 0.38 Volume constant: 1.5 plastic limit: 0.2
Figure 6.3.a	Environment: hydrogel 2 Density: 20000 cells	Young modulus: 3 Poisson ratio: 0.3 Volume constant: 1.5 plastic limit: 0.2
Figure 6.3.b	Environment: hydrogel 3 Density: 20000 cells	Young modulus: 2.5 Poisson ratio: 0.3 Volume constant: 1.5 plastic limit: 0.2

6.1.5 Medium fusion

In some cases, the spheroids fuse to a greater extent, which we call moderate fusion. Examples of this type of fusion are presented below (Figure 6.4 and 6.5). This type of fusion can be simulated via the proposed models appropriately. Table 6.2 compares experiment and simulation parameters of Figure 6.4 and 6.5.

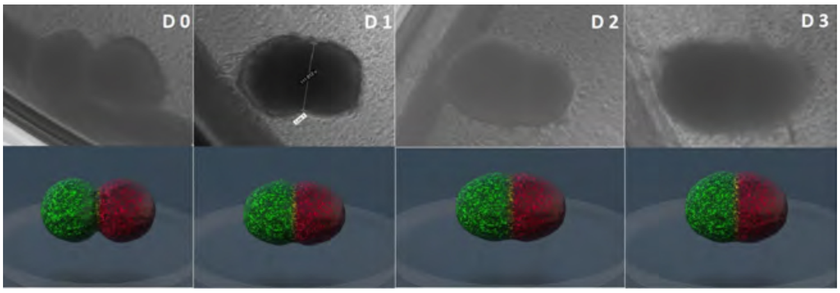


Figure 6.4: An example of two spheroids that have medium fusion within three days: Environment: hydrogel 1, density: 10000 cells.

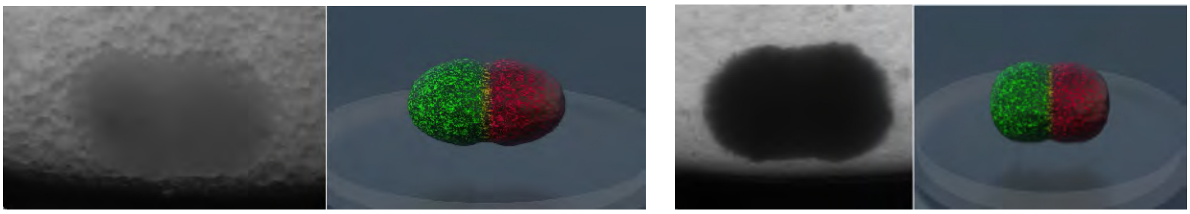


Figure 6.5: Two examples of two spheroids that have medium fusion at the sixth day: Environment: a) hydrogel 4, density: 20000 cells, b) Environment: hydrogel 1, density: 30000 cells.

Table 6.2: Comparing experiment parameters and simulation parameters for medium fusion

Figure	Experiment parameters	Simulation parameters
Figure 6.4	Environment: hydrogel 1 Density: 10000 cells	Young modulus: 3 Poisson ratio: 0.35 Volume constant: 1.4 plastic limit: 0.1
Figure 6.5a	Environment: hydrogel 4 Density: 20000 cells	Young modulus: 2 Poisson ratio: 0.45 Volume constant: 1.5 plastic limit: 0.2 stiffness: 1 rest density: 1 viscosity coefficient: 0.5
Figure 6.5b	Environment: hydrogel 1 Density: 30000 cells	Young modulus: 2.8 Poisson ratio: 0.35 Volume constant: 1.5 plastic limit: 0.6

6.1.6 Complete fusion

The last case is the complete fusion of the spheroids, in which they merge almost completely and usually form a larger spheroid. Below are examples of the complete fusion of spheroids and their corresponding simulations. This type of fusion usually has the largest deformation in the final shape of the fused spheroids. As it is shown in Figure 6.6 and 6.7, our simulator has the ability to simulate the corresponding deformation in the spheroids. Table 6.2 compares experiment and simulation parameters of Figure 6.6 and 6.7.

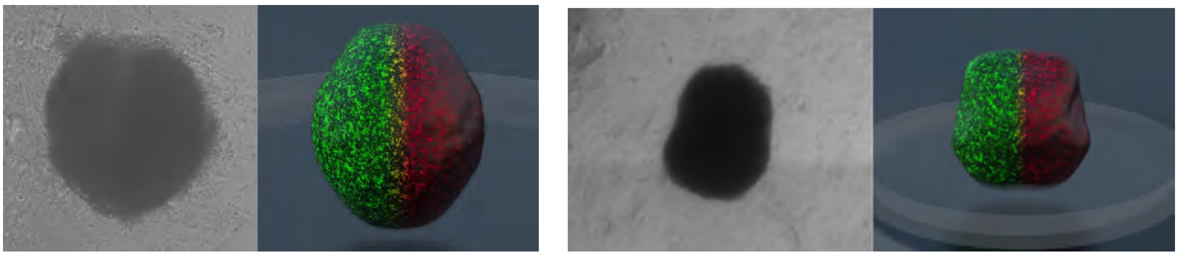


Figure 6.6: Two examples of two spheroids that have complete fusion at the sixth day: a) Environment: hydrogel 1, density: 10000 cells, b) Environment: hydrogel 1, density: 20000 cells.

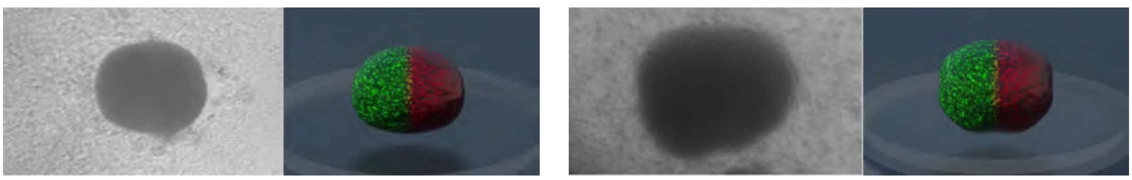


Figure 6.7: two example of two spheroids that have complete fusion at the sixth day: a) Environment: hydrogel 1, density: 10000 cells, b) Environment: hydrogel 2, density: 30000.

Table 6.3: Comparing experiment parameters and simulation parameters for complete fusion

Figure	Experiment parameters	Simulation parameters
Figure 6.6a	Environment: hydrogel 1 Density: 10000 cells	Young modulus: 0.5 Poisson ratio: 0.45 Volume constant: 1.5 plastic limit: 0.1 stiffness: 1 rest density: 1 viscosity coefficient: 1
Figure 6.6b	Environment: hydrogel 1 Density: 20000 cells	Young modulus: 0.8 Poisson ratio: 0.4 Volume constant: 1 plastic limit: 0.2 stiffness: 1 rest density: 0.8 viscosity coefficient: 0.5
Figure 6.7a	Environment: hydrogel 1 Density: 10000 cells	Young modulus: 1 Poisson ratio: 0.4 Volume constant: 1.5 plastic limit: 0.2
Figure 6.7b	Environment: hydrogel 2 Density: 30000 cells	Young modulus: 1.5 Poisson ratio: 0.4 Volume constant: 1.5 plastic limit: 0.3

This section's pictures show that the proposed models can simulate all fusion states between spheroids. Tables 6.1, 6.2, and 6.3 compare biological and simulation parameters for the spheroids fusion. As we expected, for lower fusion, there is not much deformation on spheroids. Because of that, the physical parameters of the corresponding simulation are tuned to make a less deformable object (e.g. relatively larger Young modulus and less Poisson ratio). However, the more fusion the spheroids have, the more deformation they will have. Hence, the physical parameters of the corresponding simulation for higher fusion are tuned to make a softer object.

Since all fusions occur only when the distance between spheroids is very close ($\leq 400\mu m$), therefore, the distance in all the simulations is supposed to be very close (in the scale of the simulation).

6.2 Morphologicl analysis

Morphology plays a crucial role in spheroid fusion, as it characterizes the geometric properties of the spheroids throughout the fusion process. To validate our model and simulations, it is essential to compare the morphological aspects between in-silico simulations and in-vitro experimental data.

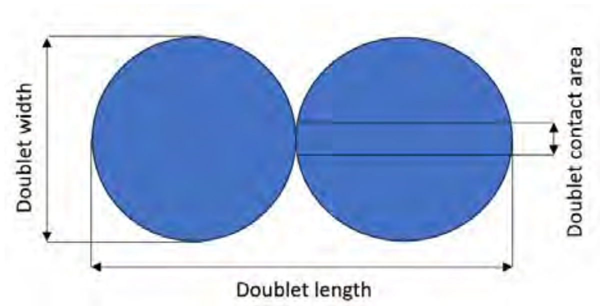


Figure 6.8: Morphological parameters measured during fusion

Following the methodology introduced by Susienka et al. [123], we defined three variables length, width, and the length of the contact area of spheroid doublets to represent the spheroid morphology during fusion (refer to Figure 6.8). These variables were measured using Fiji software in the experimental data [111]. Similarly, the same variables were measured in the simulation data corresponding to the experimental conditions. The data were categorized into three fusion types: low, medium, and complete fusion. Subsequently, statistical tests were conducted to analyze the results, employing the t-test to compare the means of each variable between the in-vitro and in-silico datasets. The summary of the test results is presented in Table 6.4.

Table 6.4: The results of t-test for comparing the mean of morphological variables for in-vitro and in-silico data

Type of fusion	Morphological variables	Hypothesis	p-value
Low fusion	Doublet length	$H_0 : \mu_{in-vitro} = \mu_{in-silico}$	0.8333 (ns)*
	Doublet width		0.5277 (ns)
	Doublet contact area		0.9797 (ns)
Medium fusion	Doublet length	$H_0 : \mu_{in-vitro} = \mu_{in-silico}$	0.7997 (ns)
	Doublet width		0.3978 (ns)
	Doublet contact area		0.4062 (ns)
Complete fusion	Doublet length	$H_0 : \mu_{in-vitro} = \mu_{in-silico}$	0.8510 (ns)
	Doublet width		0.7885 (ns)
	Doublet contact area		0.1072 (ns)

* not significant

The obtained p-values from the performed tests, as shown in Table 6.4, indicate their relatively high values, rendering them statistically insignificant. This suggests that there is no significant difference between the mean values of each morphological variable in the experimental data and the simulations. Thus, the simulations successfully capture the observed morphological behaviour in the experiments.

For a more comprehensive analysis, Figure 6.9 presents box plots illustrating the distribution of each variable for different fusion types in both in-vitro and in-silico settings. These plots reveal a remarkable similarity in the distribution of variables between the in-vitro and in-silico data. Although the variance of the in-vitro observations is generally higher, as expected, these graphs effectively demonstrate the close resemblance between the experimental and simulated data in terms of mean and variance.

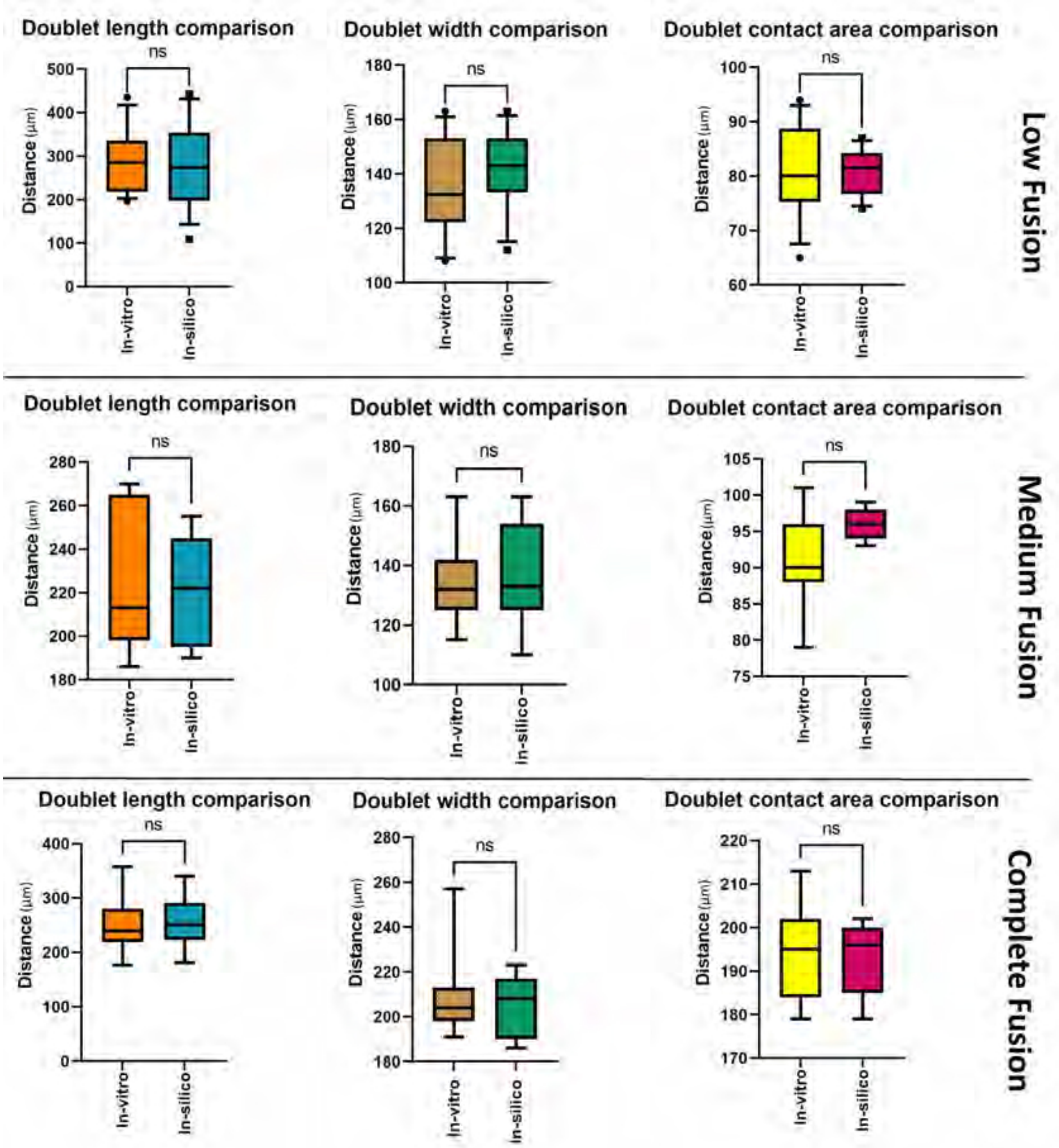


Figure 6.9: Box plots of morphological variables: the first row corresponds to the doublets length, width and contact area of low fusion, the second row corresponds to the doublets length, width and contact area of medium fusion, and the third row corresponds to the doublets length, width and contact area of complete fusion

This morphological analysis provides compelling evidence that our simulations accurately reproduce the observed behaviour of spheroid fusion. By quantifying parameters such as contact area and doublet dimensions, we aimed to demonstrate the model's ability to describe the temporal evolution of tissue shape changes.

CHAPTER 7

DISCUSSION

In this section, we will discuss the pros and cons of the proposed methods in this thesis, as well as the recently published articles in related areas. Like the whole thesis, the discussion is explained in two separate subsections for bio and digital fabrication cases.

7.1 Discussion on bio fabrication case

Physically-based simulations, especially the SPH method, are highly sensitive to model parameters such as kernel radius and the number of neighbours for each particle. Therefore, to avoid any simulation breakdown, we fixed the simulation time step at 0.01 and set the search radius for each particle so that at the beginning of the simulation, each particle has 15 neighbours. Of course, after running the simulation, the number of neighbours for each particle may change.

In this section, we will discuss the sensitivity of the simulations to the various parameters of the developed models. Observing the performed simulations, some of which were displayed in the previous section, shows that Young's modulus, Poisson ratio, volume constant, and stiffness are the most influential parameters in controlling deformation in spheroids. After that, the plastic limit has the most significant effect in controlling the amount of deformation.

In the video of Figure 5.2, when two spheroids move towards each other, they also expand their volume due to internal pressure. The model's volume constant and stiffness parameters can tune this internal pressure. Increasing the stiffness and volume constant parameters leads to an increase in volume.

The effect of internal pressure is also visible in Figure 5.3, where after spheroids reach each other, each particle goes toward the other spheroids' particles and continues its movement until the particles get balanced in a final position. It means that particles move until the total force applied to them becomes zero. In the case of Figure 5.2, the particles spread apart slightly due to the elastic force between particles. However, in the simulation of Figure 5.3, we do not see any increase in volume in spheroids due to not increasing the stiffness parameter. Therefore, spheroids approach each other and get compressed slightly at the beginning, increasing density; consequently, internal pressure force spreads them out again.

It is worth mentioning that elastic-plastic parameters are considered in all simulations. However, we have only mentioned the most important parameters that cause the specific behaviour of the spheroids during the simulation. For example, in the case of Figure 5.3, the most important parameters were stiffness and density.

While the volume constant and stiffness are the most influential parameters in the fluid-based model, our observations show that Young's modulus and Poisson's ratio are the most critical parameters in the solid-based model. In Figure 5.8, we compared four simulations while increasing their stiffness and volume constant to demonstrate the importance of these parameters. The results show that an increase in stiffness and volume constant will cause significant deformation in spheroids.

On the other hand, the effect of Young's modulus and Poisson's ratio is shown in Figure 5.6, where we compared four simulations with different values of these parameters in terms of their total volumes and areas. The results indicate that increasing the Poisson's ratio and decreasing Young's modulus will cause more deformation in spheroids.

According to Figure 5.6, from the top left to the bottom right of the graph, we have decreased the Poisson ratio and increased Young's modulus while we measured the final configuration's area and volume. All information about these simulations is mentioned at the bottom and above each picture. The interesting point of these simulation results is that by increasing the Poisson ratio and decreasing Young's modulus, we can see a decrease in volume and area except in the last simulation (bottom right of Figure 5.6). We can conclude that increasing the Poisson ratio and decreasing Young's modulus will give the spheroids more elastic properties. Therefore, we can be assured that our mathematical model and its implementation are valid. The exception that happened in the last simulation (bottom right of Figure 5.6) is due to having the maximum Poisson ratio (0.49) and minimum Young modulus (0.1), which caused the spheroid to relax and increase the volume a little bit.

A final observation we would like to make about Figure 5.6 and 5.8 is that simulated spheroids have a larger scale due to the ease of implementation compared to natural spheres. However, this difference does not affect the obtained results because the amount of deformation is scaled to the proportion of the initial and final shape of spheroids.

Another behaviour we aim to simulate is the external force caused by the interaction of the hydrogel and the spheroids. Although modelling the interaction between these two forces is not one of the goals of this study, it can be considered a topic for future research. However, we considered this interaction the total external force applied to the spheroid particles. In that case, we should be able to simulate a half or complete fusion. This can be seen in the simulations of Figure5.5. a and Figure5.5.b where we consider the spheroids an elastic-plastic solid material.

As shown in Figs. 5.5 and 5.7, both solid and fluid simulation behaviours are very similar as elastic-plastic material is a phase between pure solid and pure fluids. The only difference between the two models is that spheroids change their volume permanently during the fusion process in fluid-based simulations. This is not the case in the solid model because the volume conservation force will suppress volume change in the spheroids. Therefore, the fluid model does a better job from this point of view, as in actual images, final merged spheroids are increased in volume.

To summarize the effect of the parameters on the model, we observed that a lower Young's modulus and larger Poisson ratio produce larger elastic properties in spheroids. Increasing the volume constant will increase the volume of spheroids (in case we do not have a volume conservation force). The elastic limit will tell us when the spheroids will switch to plastic deformation. Having a larger plastic limit allows spheroids to have more plastic deformation. Regarding fluid-specific parameters, stiffness affects the internal pressure of spheroids. The larger the stiffness, the larger the internal pressure and the increase in volume during the fusion. The simulation does not show significant sensitivity to the rest of the density parameter. Last but not least, increasing the viscosity coefficient will decrease the velocity of spheroids during the fusion.

7.2 Comparison with previous studies

Our study introduces a novel approach for simulating bioprinted spheroid fusion, making it the first framework in the literature to combine graphical simulation and Smoothed Particle Hydrodynamics (SPH) for bioprinting simulation. While Göhl and colleagues

[35] also present a computer-based simulation of hydrogel in 3D bioprinting, there are fundamental differences between their work and our study. They used IPS IBOFlow software without proposing a numerical scheme for solving their model's equations. In contrast, we developed and implemented our own SPH method, providing full control over simulation parameters and enabling better tuning of the physical simulation to match real-world behaviour.

Unlike Göhl et al., our Lagrangian-based SPH method is mesh-free, offering greater flexibility in handling complex geometry and deformations. Grid-based methods, employed by Göhl et al., are limited by the initial mesh and are less adaptable for simulating spheroid fusion. Furthermore, they used an elastic-plastic model for the bioink, while our study utilizes an elastic-plastic model to simulate spheroid diffusion, capturing non-reversible deformations.

Compared to similar studies, the most important advantage of the present study is the use of a general framework that has high flexibility in simulating the behaviour of spheroids during fusion. Although visco-elastic differential equations have been used in the study of Beaune et al. (2022) [8] and Ongenae et al. (2021) [86]; however, the fundamental difference in their solution method is an advantage for this research. First, the solution method presented in this paper is not dependent on the mesh. In other words, the Lagrangian approach is used to solve the model's differential equations, in which there is no need to discretize the spherical mesh. Therefore, our approach offers several advantages over the studies mentioned above for solving continuum mechanics differential equations: i) Flexibility in domain representation: Since we are using meshless methods, we do not require a predefined mesh structure, allowing for a more flexible and adaptive representation of complex domains with irregular geometries. This makes our methods particularly suitable for problems involving free surfaces like deformations with changing topologies. ii) Easy handling of moving boundaries: our method excels in handling problems with moving boundaries or interfaces. Since the computational particles used in our methods can easily be repositioned, tracking and handling moving boundaries becomes more straightforward than mesh-based methods requiring mesh regeneration or remeshing. iii) Reduced computational cost: our method often exhibits computational efficiency advantages for problems with large deformations or when high accuracy is required. It does not suffer from the mesh distortion issues typically encountered in mesh-based methods, reducing computational effort associated with mesh updates or element distortion. iv) Adaptive refinement: our method can achieve adaptive refinement by locally increasing the density of computational particles in regions where high accuracy is needed. This adaptivity allows for efficient concentration of computational effort in areas of interest without global mesh refinement. v) Natural treatment of discontinuities: our method naturally handles problems involving discontinuities, such as the fusion of spheroids. By placing computational particles near the discontinuities, accurate representation and modelling of these features can be achieved without requiring unique treatments or complex mesh configurations. vi) Last but not least is generality: our proposed framework is general. It means that it can simulate more than just two spheroids fusion, as shown in Figure 5.1.

Ongenae et al. (2021) [86] research has explored the concept of “arrested coalescence," where certain factors hinder the complete fusion of spheroids or droplets. Unlike Ongenae et al. (2021), who model arrested coalescence with a jamming effect in viscous models, our general framework accommodates both solid and fluid states. The degree of fusion in our model is determined by specific parameters, such as stiffness, volume constant, internal pressure, Poisson ratio, and Young’s modulus of particles. In summary, arrested coalescence in our paper is modelled as balanced between internal and external forces. Internal forces are elastic-plastic, volume conservation, pressure and viscosity and external force is the gravitational force between spheroids.

A significant difference in our approach lies in the method of force generation. Ongenae et al. (2021) primarily rely on geometric forces arising from changes in distances and contact angles between cells and spheroids. In contrast, our approach is rooted in inherent material properties, such as elasticity and plasticity, and derives forces from physical equations representing chemical interactions resulting from these properties. Material properties, which are the root cause of fusion and deformation in spheroids, can better link the chemistry and biology of such a complex phenomenon to physics.

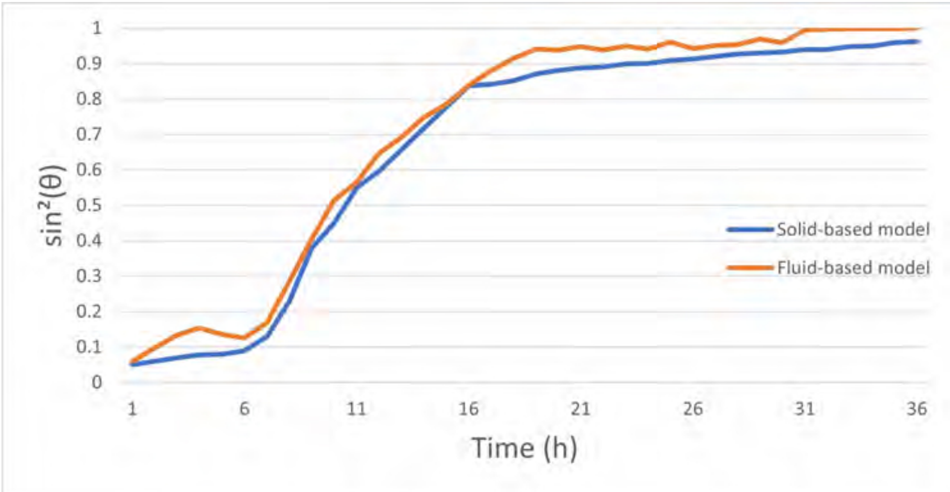


Figure 7.1: the values of $sin^2(\theta)$ for solid-based and fluid-based models.

More in detail, while Ongenae et al. (2021) focus on geometric forces and consider spheroids as perfect spheres, we simulate various deformations during fusion through our algorithm, considering particle positions on the spheroid surface. Our approach ensures that geometry and final shape are dictated by physics, taking into account elastic-plastic forces and volume conservation in solid-based models and pressure and viscosity forces in fluid-based models. Considering the material properties gives us the ability to control the geometry of the spheroids by tuning the current properties, such as stiffness or using different materials. While in Ongenae et al. (2021) paper, we do not have the ability to control the fusion process.

However, to quantitatively compare our presented model with the research of Ongenae et al. (2021), we conducted an analysis of two variables from their work. The first variable, denoted as $sin^2(\theta) = (\frac{x}{a})^2$, where x is the length of contact area, and a is the spheroid's radius. θ is the contact angle, and $sin^2(\theta)$ captures the dynamics of fusion according to Ongenae et al. (2021). Figure 7.1 illustrates the fusion dynamics for solid and fluid-based models of our approach. In both cases, fusion initiates gradually with a small increase in the theta angle, indicating a low attraction force between the spheroids. However, as a bond is established between the spheroid particles, the fusion process accelerates until the gravitational force achieves equilibrium with internal forces, resulting in stable fusion. This observation contrasts with Ongenae et al. (2021), where fusion starts rapidly and continues at high speed until reaching equilibrium in Figure 3. B and Figure 4.A. This discrepancy arises from the different definitions of gravity and internal forces in their model and ours. We believe that a gradual fusion rate at the start and end of the process aligns more naturally with our observations from in-vitro data. Unlike the model proposed by Ongenae et al. (2021), in our model, fast fusion is hindered by elastic collision and viscosity at the beginning. However, as we update the neighbours of each particle, bonds between particles of two spheroids form, accelerating fusion. This exemplifies the advantage of employing forces derived from material properties.

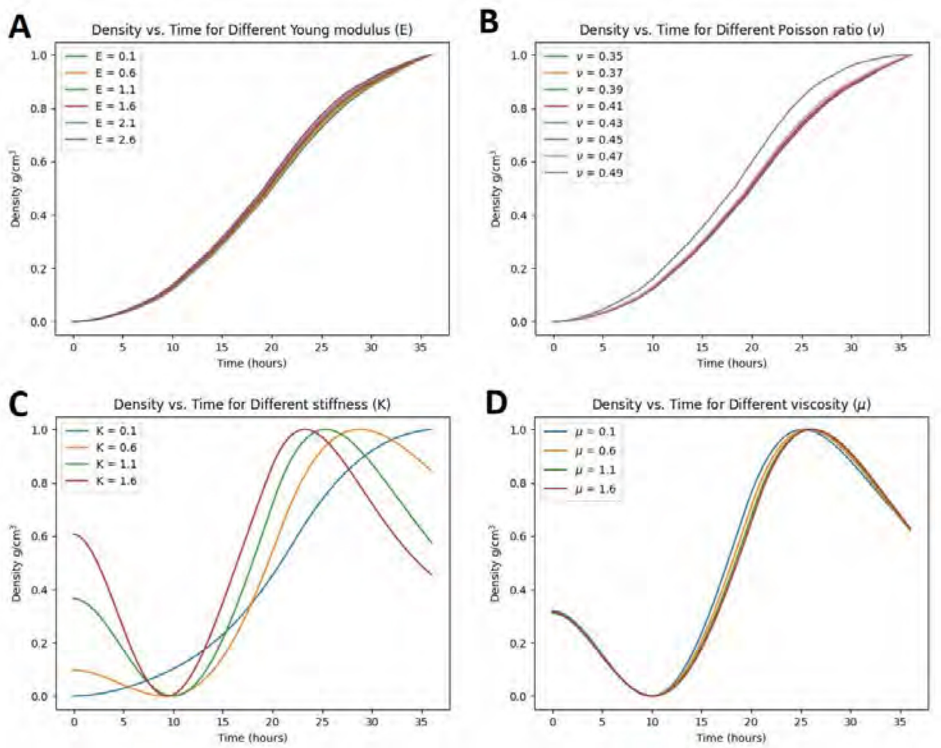


Figure 7.2: A) Average density vs time for different Young's modulus. B) Average density vs time for different Poisson ratios. C) Average density vs time for different fluid stiffness. D) Average density vs time for different fluid viscosity

The second variable is the density of the entire system, which represents the average density of all particles contained with the two spheroids, plotted against a 36-hour fusion time. To compare the average density values for each model, we focused on the most important parameters of our model, namely Young's modulus, Poisson's ratio, fluid stiffness, and viscosity, as they significantly influence this variable.

Figure 7.2 displays the average density of spheroids over time for various values of Young's modulus, Poisson's ratio, stiffness, and viscosity. All four graphs exhibit similar trends, with increasing density as the fusion progresses. Specifically, in the fluid model, increasing stiffness and viscosity lead to an S-shaped behaviour in the density graph, where density initially increases and then decreases towards the end. This behaviour is attributed to the rise in fluid stiffness, which elevates internal pressure, resulting in increased volume and, consequently, decreased density. Comparing our results with Figure 3.A from Ongenae et al. (2021) article reveals similar density trends during the fusion process, which is reasonable as we expect an increase in density when two spheroids are fused. Our approach adeptly captures arrested coalescence without relying on the “jamming effect". Instead, we leverage the intrinsic material properties of the spheroids to create internal forces that balance the gravitational force between them, resulting in the termination of fusion. This method is not only more realistic and versatile but also highly effective in modelling various types of behaviour beyond arrested coalescence. We regenerated the parameters that Ongenae et al. (2021) utilized in their study, which explicitly outline fusion dynamics as detailed in their paper and compared them with the result of Ongenae et al. (2021). The primary advantage of our approach lies in its ability to not only capture arrested coalescence with similar properties (as demonstrated in the density graph) but also to model various other fusion phenomena, including different types of deformation. This introduces a novel perspective to modelling spheroid fusion. In our case, we are capable of modelling fusion based on material properties rather than solely the geometry of the spheroids. This ability implies the potential for controlling the fusion process using diverse biological materials.

For future studies, to enhance the accuracy and relevance of our simulation and better align it with in-vivo experiments, it is important to consider additional factors, such as the inclusion of the hydrogel medium and the composition of the extracellular matrix (ECM). The ECM composition in fibroblasts, myocytes, and coronary cells plays a vital role in tissue structure and function, with fibroblasts producing collagen, fibronectin, and proteoglycans, myocytes interacting with collagen, laminins, and elastin, and coronary cells interacting with collagens, proteoglycans, and glycoproteins. The ECM provides structural support, regulates cell adhesion and migration, and contributes to tissue integrity. However, further investigation is needed to understand the specific ECM composition and its impact on spheroid fusion. Developing a fluid model that can accurately simulate the properties of the hydrogel and incorporate boundary conditions influenced by the ECM can be the subject of our future research.

Although the model used in Muller et al.’s (2004) [72] article is old, it has been considered as a basis for developing the models used in this research. In this regard, the use of different basis functions, the online updating of the nearest neighbours search in the fluid-base model, and the addition of elastic collisions between particles in the solid-base model are among the changes we made in the Mueller et al.‚Äôs (2004) model to adapt it better to our case study.

The innovation of this research is interesting from several aspects. First, the use of a physically-based simulation method in the advanced Biofabrication of human tissue can potentially open a new chapter for solving complex biological problems using physical equations. This method can be very effective, especially in the case of creating a 3D human texture, because the goal of physically-based simulation is the 3D world that is usually used in computer games or animations. Second, the presented models have been updated according to our case study to justify the real conditions of the experiments as much as possible. Finally, the design of the experiments and the number of experiments (96 experiments) to validate the models have also been unique in their kind.

7.3 Discussion on developable mesh decomposition

Developability, as a fundamental concept in differential geometry [85], has garnered significant attention in computational design, particularly in decomposing triangular meshes into flat patches for manufacturing purposes [10, 41, 146, 147]. The practical utility of these methods lies in their ability to simplify manufacturing processes by generating flat pieces that are easier to produce, pack, and assemble into complex 3D models.

However, despite the emphasis on fabrication, many existing methods fail to explicitly consider crucial constraints related to manufacturability. While they ensure developability, they often overlook the need to ensure self-intersection-free, smooth, easily bendable patches of reasonable size, particularly crucial for large-scale designs in architecture. This oversight often results in suboptimal decompositions that hinder the manufacturability of the final product.

In our work, we present an algorithm that explicitly integrates fabrication constraints into the decomposition process, with a specific focus on ensuring ruling line consistency for easy assembly. Unlike prior methods, our algorithm takes into account various constraints simultaneously, optimizing the number of patches and their ruling lines to produce intersection-free decompositions automatically.

The exploration of developability is deeply rooted in mathematical algorithms [11, 100–102, 128]. However, for practical applications in fabrication, the focus shifts towards finding the closest developable approximation of a given mesh [121]. Unlike previous approaches that address specific input constraints or rely on global-local optimization strategies [10, 41, 113], our method offers a more comprehensive solution that accommodates a wide range of inputs, making it applicable across various scenarios.

To ensure the manufacturability of the final result, we impose several constraints, including the absence of self-intersections, adherence to developability, smooth variation of ruling lines, and a limited number of patches. Leveraging curvature-aligned cross-fields, we decompose meshes into patches aligned with curvature flow, a feature unique to our approach.

Our method addresses the challenge of efficiently decomposing meshes into developable patches while ensuring manufacturability without compromising quality. Through rigorous optimization and curvature-guided decomposition, we provide a robust solution that can be applied across various industries and applications, making it a valuable contribution to computational design for manufacturing.

Moreover, we demonstrate the efficacy of our method through comparative analysis with existing approaches. Our algorithm not only outperforms previous methods in terms of producing manufacturable decompositions but also achieves superior results in terms of computational efficiency.

In conclusion, our work represents a significant advancement in the field of computational design for manufacturing. By integrating fabrication constraints into the decomposition process and leveraging curvature-guided techniques, we provide a practical solution for generating developable patches from triangular meshes. Our method has the potential to revolutionize the way complex 3D models are fabricated, opening up new possibilities for efficient and cost-effective manufacturing processes.

7.3.1 Comparison with previous studies

We compared our method with the one proposed by Stein et al. [121] using the reference implementation provided in Meshlab [16] (see Figure 7.6). Since [121] does not produce a patch layout automatically, we split the resulting mesh along edges whose angle exceeded a certain threshold (12 degrees). As shown in Figure 7.4, our method produces a better-structured patch layout that uses fewer patches but comes with a slightly higher approximation error. For the two examples shown in the figure, the Hausdorff mean distance is below 0.45% for Stein et al.'s method and below 0.8% for our proposed method. Despite this, we believe that the range of introduced approximation error is still acceptable for practical applications.

Figure 7.4 shows a comparison with the recent approach of Zhao et al. [147]. Both meshes have similar boundary lengths; however, our method produces a more regular and easier-to-assemble pattern layout. Similar to the previous comparison with the approach

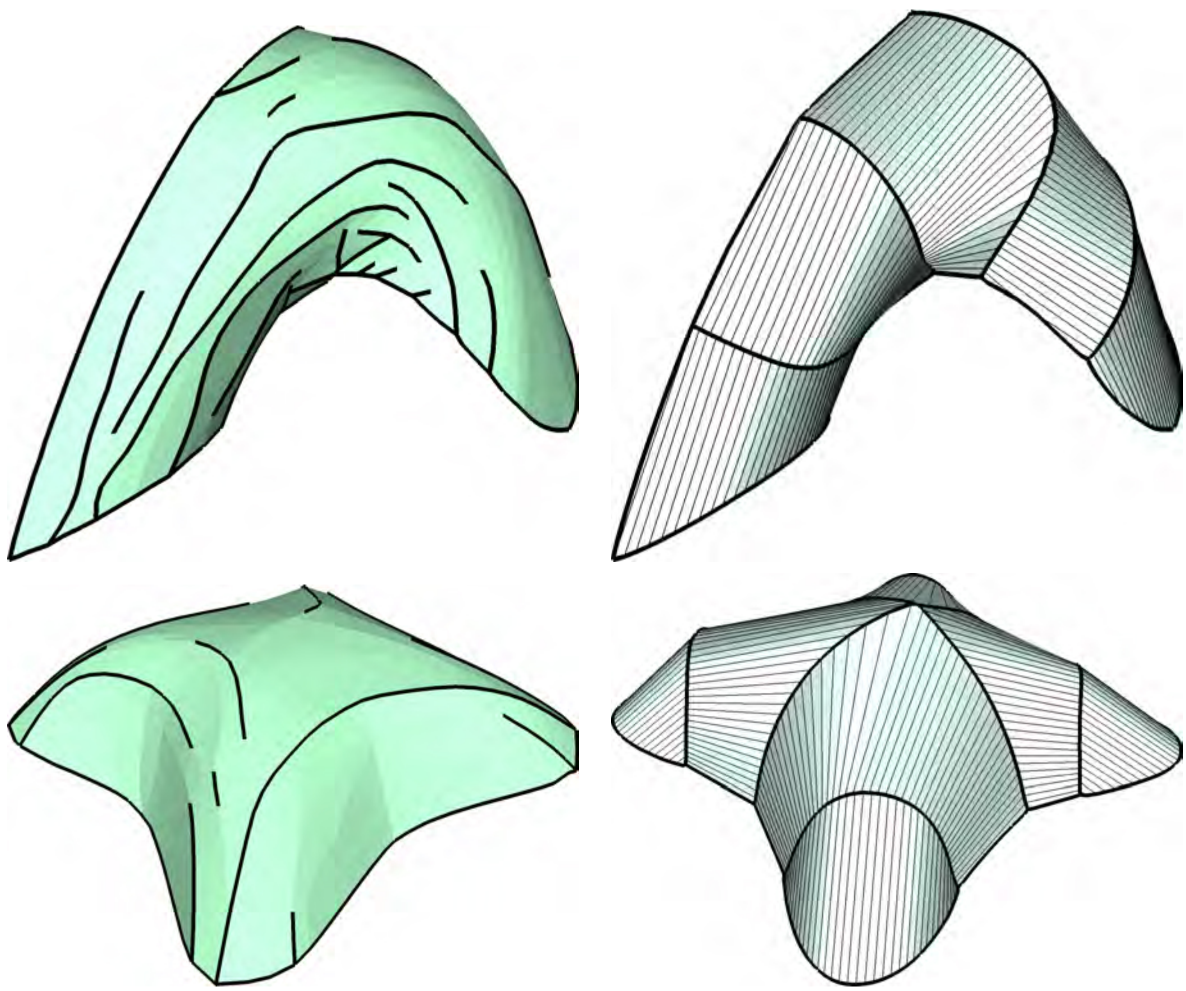


Figure 7.3: Comparison of the method proposed by Stein and colleagues [121] (left) versus ours (right).

of Stein and colleagues [121], our method introduces a slightly higher approximation error (an average of 0.8% of the bounding box diagonal). Nonetheless, this error is still acceptable for fabrication purposes. Notably, Zhao et al.'s approach does not guarantee that the patches are fully developable, as highlighted in our experiment.

Finally, in Figure 7.10, we present an example where our method successfully derives a proper decomposition, whereas the approach of Dev2PQ [131] fails. As highlighted in the original paper, Dev2PQ strictly depends on the singularity layout of the guiding line field (shown on the right side of the figure).

Field Alignment The key design principle of *aligning the ruling lines along the directions of minimal curvature* is crucial for reducing the approximation error in the

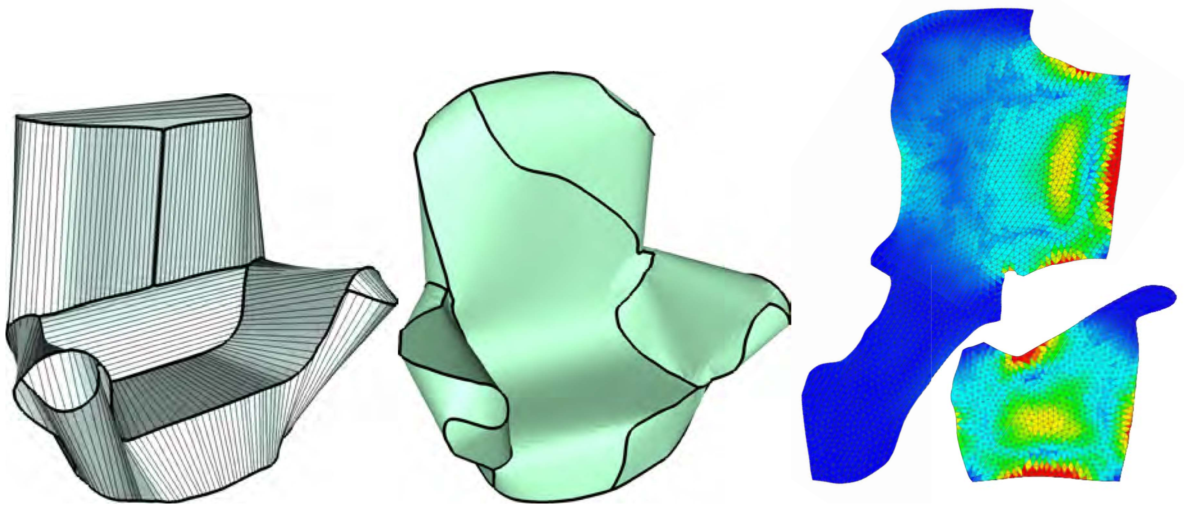


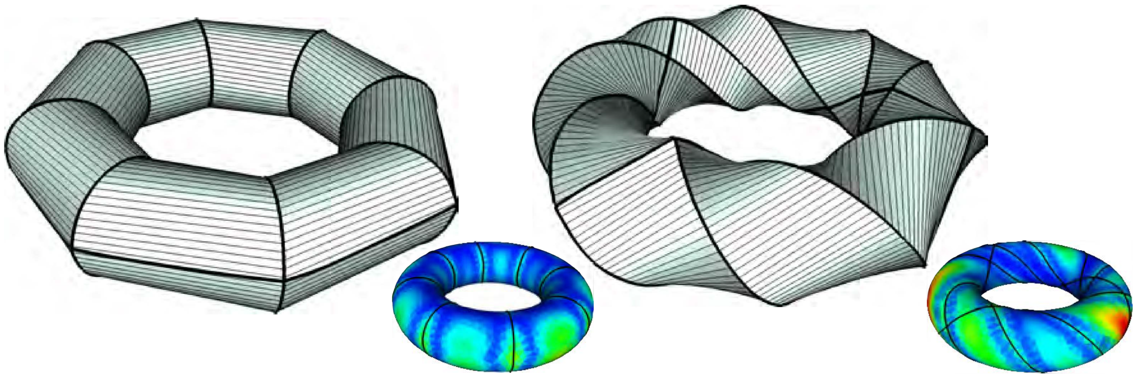
Figure 7.4: Comparison of our method (left) with the one recently proposed by Zhao et al. [147] (right). The two layouts have similar boundary lengths. The last column shows the distortion in the 2D flattening of two patches from the dataset of Zhao et al. We used the As-Rigid-As-Possible deformation technique [120], where the red colour corresponds to 2.5% edge elongation.

discretized ruled surfaces.

In the example depicted in Figure 7.5, two patch layouts with comparable border lengths are contrasted. One layout follows the minimal curvature direction, while the other does not. The former exhibits a significantly lower approximation error, underscoring the importance of proper ruling alignment. This highlights how optimizing the geometric configuration according to curvature can greatly enhance the fidelity of the surface approximation.

Full developability While aligning ruling lines to the principal curvature direction is crucial, it is not the only factor in achieving an optimal patch decomposition. Another key aspect is aligning the boundaries of the decomposition to the primary curvature direction and placing the seams in optimal locations. Since seams must align correctly when assembled, their placement significantly impacts the overall shape. Figure 7.6 demonstrates how proper boundary alignment can further reduce the approximation error and result in a more compact patch layout. This experiment highlights the superiority of field-aligned patch decomposition compared to other methods with the same boundary length, such as Simple Voronoi with Delaunay relaxation and Variational Shape Approximation [18].

Geometric approximation alone is not sufficient to achieve a patch decomposition



$\varepsilon = 0.41\%$, $\mathscr{E} = 2.19\%$ $\varepsilon = 0.71\%$ $\mathscr{E} = 4.7\%$

Figure 7.5: Proper alignment of the ruling lines and patch layout significantly reduces the average approximation error ε from 0.71% (right) to 0.41% (left) and the maximum error $\mathscr{E}$ from 4.7% (right) to 2.19% (left). The error is measured as the Hausdorff distance to the target surface, normalized as a percentage of the bounding box diagonal.

that is both realistic and fabricable. Since some common fabrication materials, like paper, are inextensible [**?**], it is crucial to ensure the patch layout is ideally 100% developable. As shown in Figure 7.7, simply using a patch decomposition method, such as those proposed by Pietroni and colleagues [83, 90], and parametrizing each patch with as-rigid-as-possible parametrization [59] is often insufficient. The resulting patches can still exhibit significant stretch (up to 50% in some areas, as seen in the dolphin decomposition). Remarkably, our discretized ruled surface tessellation, we can guarantee *fully developable patches*.

This advantage is also demonstrated in Figure 7.8, where we compare our method to that of Stein and colleagues [121]. Their decomposition is not fit for fabrication as it is not bijective, and the flattening distortion exceeds 5% in several regions. Further comparison in Figure 7.9 highlights the superiority of our method over Zhao et al. [147] in terms of developability, compactness, and the quality of the resulting patch layout.

Contribution We summarized in table 7.1 an overview of the proposed method and its main characteristics in relation to key aspects such as developability, patch layout, and curvature alignment, as discussed earlier. Among the compared methods, Dev2PQ [131] is the only one that explicitly uses a field as input. While Dev2PQ ensures full developability, it imposes strict constraints on the derivation of the final decomposition. As noted in the original paper, Dev2PQ heavily relies on the singularity layout of the guiding line field (illustrated on the right side of the figure). In Figure 7.10, we

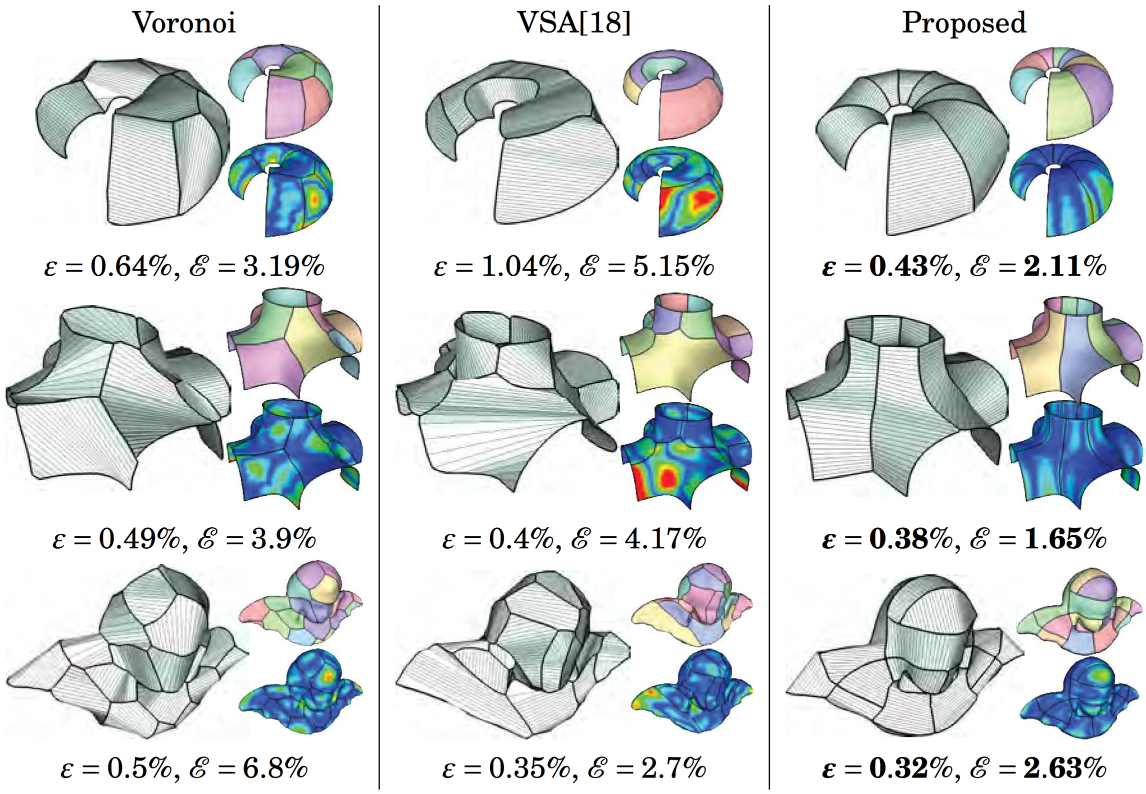


Figure 7.6: This comparison of our field-aligned patch layout with the Voronoi partition with Delaunay relaxation and Variational Shape Approximation [18] shows the superiority of the proposed method in approximating the geometry and generating a compact patch layout, even in regions where precise curvature directions are not well-defined.

Method	**Curv Align**	**Full Develop**	**Automatic Layout**
Tang et al. (2016) [128]	(✓)	✓	✗
Schuller st al. (2018) [112]	✗	✓	✓
Stein et al. (2018) [121]	(✓)	✗	✓
Zhao et al. (2023) [147]	✗	✗	✓
Verhoeven et al. (2022) [131]	✓	✓	(✓)
Proposed	✓	✓	✓

Table 7.1: A table providing an overview of state-of-the-art methods and their respective capabilities.

demonstrate an example where our method successfully produces a valid decomposition, whereas the Dev2PQ approach [131] fails to do so.

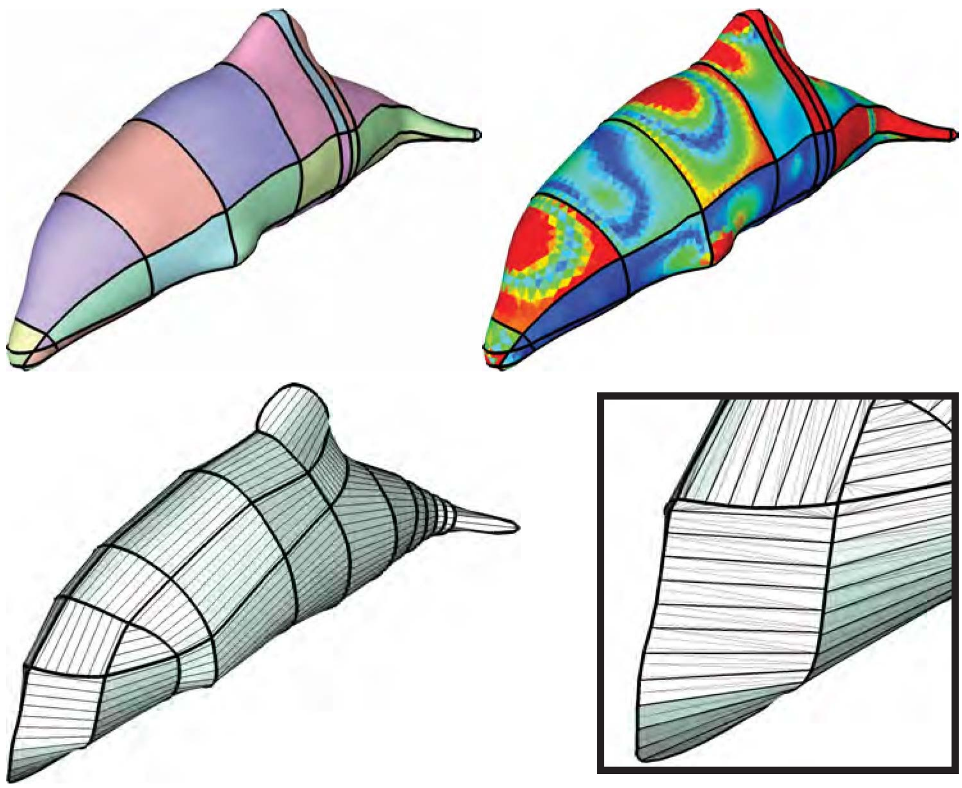


Figure 7.7: A simple curvature-aligned patch decomposition, such as the one proposed by Pietroni and colleagues [90], combined with surface flattening techniques like as-rigid-as-possible parametrization [59], is generally not suitable for fabrication, as it does not guarantee full developability (see left). The red regions indicate areas where stretching or compression exceeds 5% of the original length. In contrast, our approach produces a tessellation that is 100% developable by construction.

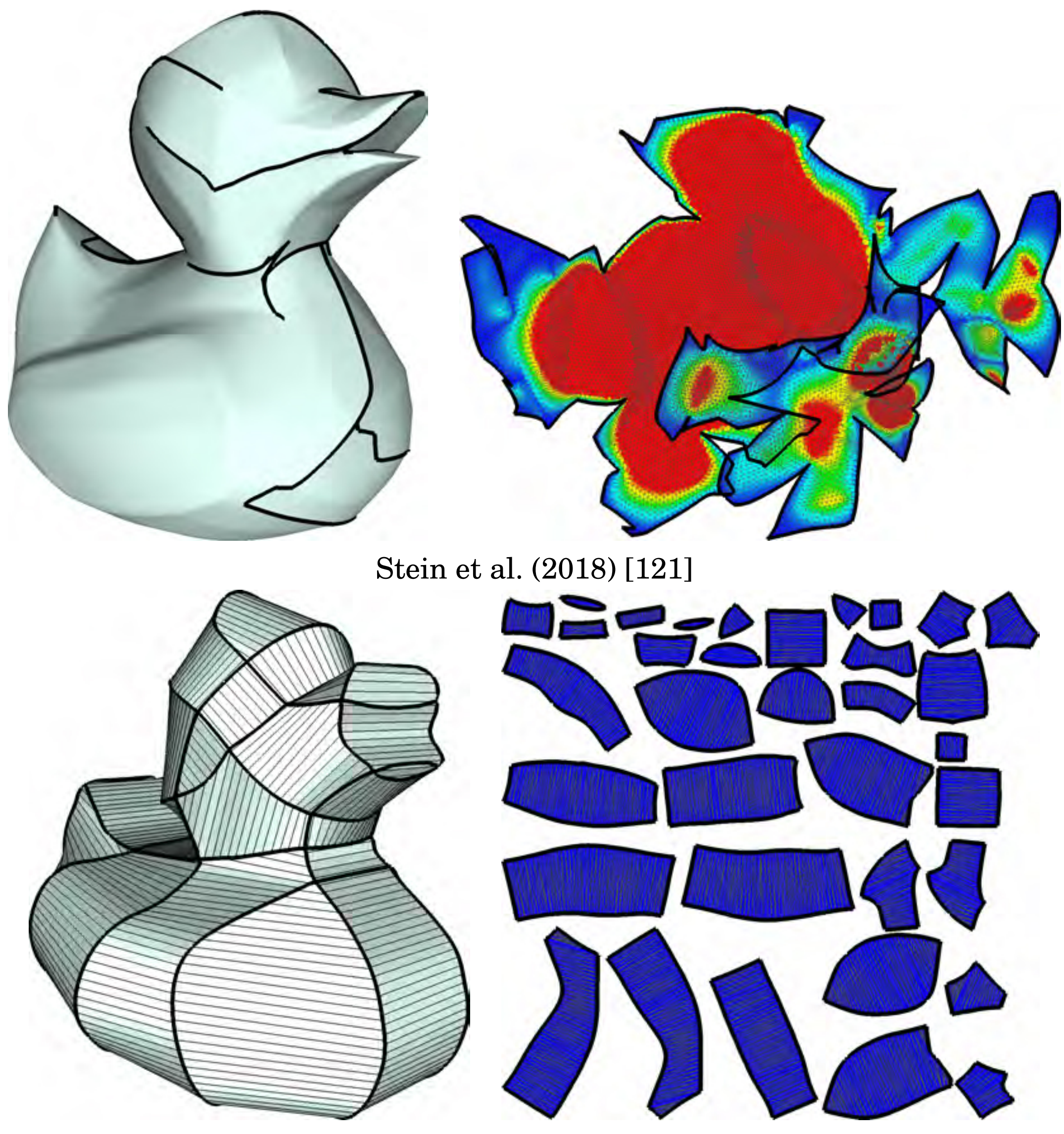


Figure 7.8: Comparison of the proposed method (bottom) with that of Stein et al. [121] (top). Unlike the concurrent method, our approach ensures both bijectivity and full developability (red areas indicating where stretching exceeds 5%.

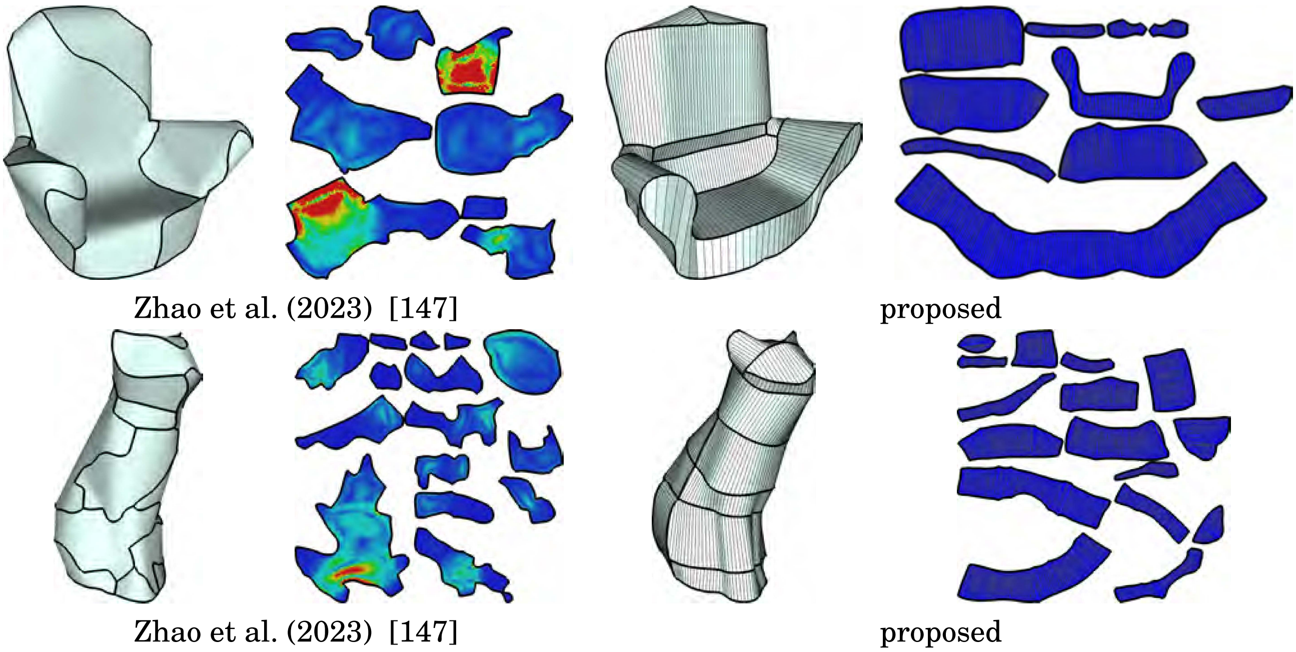


Figure 7.9: Comparison of our method (right columns) with Zhao et al. [147] (left columns). The layouts have similar boundary lengths. The last column shows the distortion in the 2D flattening of two patches. We used the As-Rigid-As-Possible deformation technique [120], where red corresponds to 1% edge elongation.

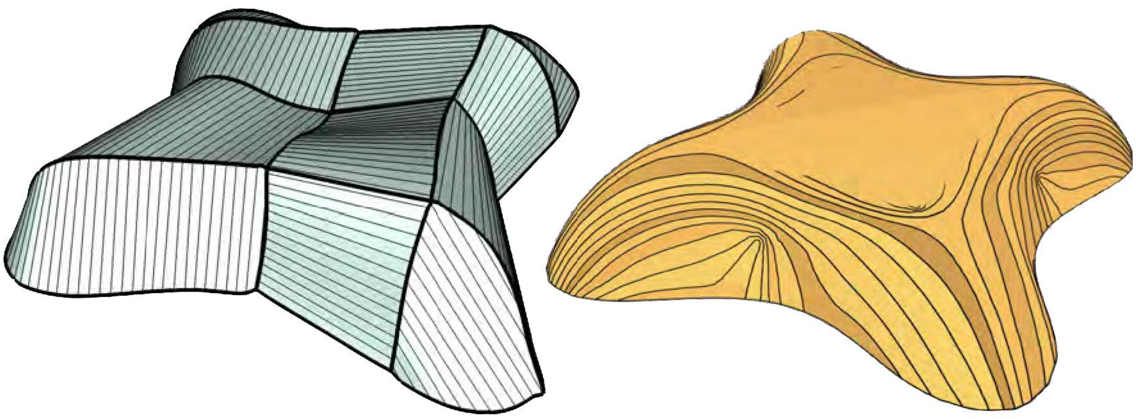


Figure 7.10: Comparison of our method with Dev2Qp [131]. As highlighted in the original paper, Dev2Qp is unable to provide a solution for the field shown on the right. In contrast, our method successfully derives a valid ruling patch layout (image courtesy of [131]).

Chapter 8

Conclusion

In the realm of computational design, where innovation intersects with bio fabrication and digital fabrication, our research represents a concerted effort to advance the frontiers of both disciplines. Through the synthesis of physical modelling and algorithmic approaches, we have made significant strides in shaping the future of tissue engineering and mesh decomposition techniques.

Our work on the physically-based simulation of visco-elastic fusion in 3D bioprinted spheroids stands as a testament to the potential of computational modelling in bio fabrication. By developing continuum models and employing smoothed particle hydrodynamics (SPH), we have simulated the intricate fusion process of spheroids with remarkable fidelity. This research not only deepens our understanding of tissue formation dynamics but also offers practical insights into optimizing bioprinting processes for tailored tissue engineering applications.

Simultaneously, our contributions to mesh decomposition through the introduction of ruling patches signify a paradigm shift in digital fabrication methodologies. By devising a novel method to decompose meshes into compact, aesthetically pleasing developable patches with embedded ruling lines, we have unlocked new possibilities for efficient fabrication of complex structures. Our approach, though non-interactive at present, demonstrates efficiency and versatility, showcasing the potential for transformative applications in various industries.

Looking ahead, our research trajectory remains anchored in the pursuit of excellence at the intersection of computational design, bio fabrication, and digital fabrication. We envision further refinement of our methodologies, leveraging machine learning techniques for parameter optimization and incorporating physical simulations to enhance precision and fidelity. By bridging the gap between theoretical insights and practical applications, we aim to catalyze innovation in both biological and digital realms, shaping a future where computational design serves as a cornerstone for transformative advancements in fabrication technologies.

In essence, our work embodies the ethos of interdisciplinary collaboration and innovation, driving forward the frontiers of computational design for bio fabrication and digital fabrication. As we continue to push the boundaries of what is possible, we remain steadfast in our commitment to harnessing the power of computational tools to unlock new opportunities and address pressing challenges in the realms of tissue engineering and digital fabrication.

BIBLIOGRAPHY

[1] M. ACAR, K. S. KOCHERLAKOTA, M. M. MURPHY, J. G. PEYER, H. OGURO, C. N. INRA, C. JAIYEOLA, Z. ZHAO, K. LUBY-PHELPS, AND S. J. MORRISON, *Deep imaging of bone marrow shows non-dividing stem cells are mainly perisinusoidal*, Nature, 526 (2015), pp. 126–130.

[2] S. ADRA, T. SUN, S. MACNEIL, M. HOLCOMBE, AND R. SMALLWOOD, *Development of a three dimensional multiscale computational model of the human epidermis*, PloS one, 5 (2010), p. e8511.

[3] S. ALT, P. GANGULY, AND G. SALBREUX, *Vertex models: from cell mechanics to tissue morphogenesis*, Philosophical Transactions of the Royal Society B: Biological Sciences, 372 (2017), p. 20150520.

[4] P. M. ALTROCK, L. L. LIU, AND F. MICHOR, *The mathematics of cancer: integrating quantitative models*, Nature Reviews Cancer, 15 (2015), pp. 730–745.

[5] R. P. ARAUJO AND D. S. MCELWAIN, *A history of the study of solid tumour growth: the contribution of mathematical modelling*, Bulletin of mathematical biology, 66 (2004), pp. 1039–1091.

[6] D. BARAFF, *Physically based modeling: Rigid body simulation*, SIGGRAPH Course Notes, ACM SIGGRAPH, 2 (2001), pp. 2–1.

[7] J. BARBIČ AND J. POPOVIĆ, *Real-time control of physically based simulations using gentle forces*, ACM transactions on graphics (TOG), 27 (2008), pp. 1–10.

[8] G. BEAUNE, L. SINKKONEN, D. GONZALEZ-RODRIGUEZ, J. V. TIMONEN, AND F. BROCHARD-WYART, *Fusion dynamics of hybrid cell–microparticle aggregates: A jelly pearl model*, Langmuir, 38 (2022), pp. 5296–5306.

[9] K. BERLAND, V. R. COOPER, K. LEE, E. SCHRÖDER, T. THONHAUSER, P. HYLDGAARD, AND B. I. LUNDQVIST, *van der waals forces in density func-*

tional theory: a review of the vdw-df method, Reports on Progress in Physics, 78 (2015), p. 066501.

[10] A. BINNINGER, F. VERHOEVEN, P. HERHOLZ, AND O. SORKINE-HORNUNG, *Developable approximation via gauss image thinning*, in Computer Graphics Forum, vol. 40, Wiley Online Library, 2021, pp. 289–300.

[11] P. BO AND W. WANG, *Geodesic-controlled developable surfaces for modeling paper bending*, in Computer Graphics Forum, vol. 26, Wiley Online Library, 2007, pp. 365–374.

[12] D. BOMMES, H. ZIMMER, AND L. KOBBELT, *Mixed-integer quadrangulation*, ACM Trans. Graph., 28 (2009).

[13] A. F. BONATTI, I. CHIESA, G. VOZZI, AND C. DE MARIA, *Open-source cad-cam simulator of the extrusion-based bioprinting process*, Bioprinting, 24 (2021), p. e00172.

[14] M. CAMPBELL, M. CHABRIA, G. A. FIGTREE, L. POLONCHUK, AND C. GENTILE, *Stem cell-derived cardiac spheroids as 3d in vitro models of the human heart microenvironment*, in Stem Cell Niche, Springer, 2018, pp. 51–59.

[15] M. CAMPEN, *Partitioning surfaces into quadrilateral patches: A survey*, in Computer graphics forum, vol. 36, Wiley Online Library, 2017, pp. 567–588.

[16] P. CIGNONI, G. RANZUGLIA, M. CALLIERI, M. CORSINI, F. GANOVELLI, N. PIETRONI, M. TARINI, ET AL., *Meshlab*, (2011).

[17] E. A. CODLING, M. J. PLANK, AND S. BENHAMOU, *Random walk models in biology*, Journal of the Royal Society Interface, 5 (2008), pp. 813–834.

[18] D. COHEN-STEINER, P. ALLIEZ, AND M. DESBRUN, *Variational shape approximation*, ACM Trans. Graph., 23 (2004), p. 905‚Äì914.

[19] J. CORNELIS, J. BENDER, C. GISSLER, M. IHMSEN, AND M. TESCHNER, *An optimized source term formulation for incompressible sph*, The Visual Computer, 35 (2019), pp. 579–590.

[20] I. F. COSTA AND R. BALANIUK, *Lem-an approach for real time physically based soft tissue simulation*, in Proceedings 2001 ICRA. IEEE International Confer-

ence on Robotics and Automation (Cat. No. 01CH37164), vol. 3, IEEE, 2001, pp. 2337–2343.

[21] W. de Back, T. Zerjatke, and I. Roeder, *Statistical and mathematical modeling of spatiotemporal dynamics of stem cells*, in Stem Cell Mobilization, Springer, 2019, pp. 219–243.

[22] D. B. Deegan, C. Zimmerman, A. Skardal, A. Atala, and T. D. Shupe, *Stiffness of hyaluronic acid gels containing liver extracellular matrix supports human hepatocyte function and alters cell morphology*, Journal of the mechanical behavior of biomedical materials, 55 (2016), pp. 87–103.

[23] S. Derakhshanfar, R. Mbeleck, K. Xu, X. Zhang, W. Zhong, and M. Xing, *3d bioprinting for biomedical devices and tissue engineering: A review of recent trends and advances*, Bioactive materials, 3 (2018), pp. 144–156.

[24] O. Diamanti, A. Vaxman, D. Panozzo, and O. Sorkine-Hornung, *Designing* N-*polyvector fields with complex polynomials*, Comput. Graph. Forum, 33 (2014), pp. 1–11.

[25] L. H. Dudte, E. Vouga, T. Tachi, and L. Mahadevan, *Programming curvature using origami tessellations*, Nature materials, 15 (2016), pp. 583–588.

[26] G. E. Farin, *Curves and surfaces for computer-aided geometric design: A practical guide*, 1997.

[27] P. A. Fleming, W. S. Argraves, C. Gentile, A. Neagu, G. Forgacs, and C. J. Drake, *Fusion of uniluminal vascular spheroids: a model for assembly of blood vessels*, Developmental Dynamics, 239 (2010), pp. 398–406.

[28] G. Forgacs, R. A. Foty, Y. Shafrir, and M. S. Steinberg, *Viscoelastic properties of living embryonic tissues: a quantitative study*, Biophysical journal, 74 (1998), pp. 2227–2234.

[29] R. A. Foty and M. S. Steinberg, *The differential adhesion hypothesis: a direct evaluation*, Developmental biology, 278 (2005), pp. 255–263.

[30] M. Freire, M. Bhargava, C. Schreck, P.-A. Hugron, B. Bickel, and S. Lefebvre, *Pcbend: Light up your 3d shapes with foldable circuit boards*, ACM Transactions on Graphics (TOG), 42 (2023), pp. 1–16.

[31] L. GAO, Z. R. GREGORICH, W. ZHU, S. MATTAPALLY, Y. ODUK, X. LOU, R. KANNAPPAN, A. V. BOROVJAGIN, G. P. WALCOTT, A. E. POLLARD, ET AL., *Large cardiac muscle patches engineered from human induced-pluripotent stem cell–derived cardiac cells improve recovery from myocardial infarction in swine*, Circulation, 137 (2018), pp. 1712–1730.

[32] K. GAVRIIL, R. GUSEINOV, J. PÉREZ, D. PELLIS, P. HENDERSON, F. RIST, H. POTTMANN, AND B. BICKEL, *Computational design of cold bent glass façades*, ACM Transactions on Graphics (TOG), 39 (2020), pp. 1–16.

[33] C. GENTILE, *Filling the gaps between the in vivo and in vitro microenvironment: engineering of spheroids for stem cell technology*, Current stem cell research & therapy, 11 (2016), pp. 652–665.

[34] A. GHAFFARIZADEH, R. HEILAND, S. H. FRIEDMAN, S. M. MUMENTHALER, AND P. MACKLIN, *Physicell: an open source physics-based cell simulator for 3-d multicellular systems*, PLoS computational biology, 14 (2018), p. e1005991.

[35] J. GÖHL, K. MARKSTEDT, A. MARK, K. HÅKANSSON, P. GATENHOLM, AND F. EDELVIK, *Simulations of 3d bioprinting: predicting bioprintability of nanofibrillar inks*, Biofabrication, 10 (2018), p. 034105.

[36] J. GROLL, T. BOLAND, T. BLUNK, J. A. BURDICK, D.-W. CHO, P. D. DALTON, B. DERBY, G. FORGACS, Q. LI, V. A. MIRONOV, ET AL., *Biofabrication: reappraising the definition of an evolving field*, Biofabrication, 8 (2016), p. 013001.

[37] J. GROLL, J. A. BURDICK, D.-W. CHO, B. DERBY, M. GELINSKY, S. C. HEILSHORN, T. JUENGST, J. MALDA, V. A. MIRONOV, K. NAKAYAMA, ET AL., *A definition of bioinks and their distinction from biomaterial inks*, Biofabrication, 11 (2018), p. 013001.

[38] R. HALL, *Amoeboid movement as a correlated walk*, Journal of mathematical biology, 4 (1977), pp. 327–335.

[39] R. F. HARIK, H. GONG, AND A. BERNARD, *5-axis flank milling: a state-of-the-art review*, Computer-Aided Design, 45 (2013), pp. 796–808.

[40] M. HOFFMANN, J.-P. KUSKA, M. ZSCHARNACK, M. LOEFFLER, AND J. GALLE, *Spatial organization of mesenchymal stem cells in vitro,Äîresults from a new individual cell-based model with podia*, PLoS One, 6 (2011), p. e21960.

[41] A. Ion, M. Rabinovich, P. Herholz, and O. Sorkine-Hornung, *Shape approximation by developable wrapping*, ACM Transactions on Graphics (TOG), 39 (2020), pp. 1–12.

[42] J. N. Israelachvili, *Intermolecular and surface forces*, Academic press, 2011.

[43] S. S. Iyer, K. P. Mazarakos, and H.-X. Zhou, *Molecular dynamics simulations of droplet fusion reveals shear thinning*, Biophysical Journal, 121 (2022), p. 472a.

[44] R. K. Jain, P. Au, J. Tam, D. G. Duda, and D. Fukumura, *Engineering vascularized tissue*, Nature biotechnology, 23 (2005), pp. 821–823.

[45] C. Jiang, F. Rist, H. Pottmann, and J. Wallner, *Freeform quad-based kirigami*, ACM Transactions on Graphics (TOG), 39 (2020), pp. 1–11.

[46] C. Jiang, C. Wang, F. Rist, J. Wallner, and H. Pottmann, *Quad-mesh based isometric mappings and developable surfaces*, ACM Transactions on Graphics (TOG), 39 (2020), pp. 128–1.

[47] D. Julius, V. Kraevoy, and A. Sheffer, *D-charts: Quasi-developable mesh segmentation*, in Computer Graphics Forum, vol. 24, Amsterdam: North Holland, 1982-, 2005, pp. 581–590.

[48] H.-W. Kang, S. J. Lee, I. K. Ko, C. Kengla, J. J. Yoo, and A. Atala, *A 3d bioprinting system to produce human-scale tissue constructs with structural integrity*, Nature biotechnology, 34 (2016), pp. 312–319.

[49] M. Kilian, S. Flöry, Z. Chen, N. J. Mitra, A. Sheffer, and H. Pottmann, *Curved folding*, ACM transactions on graphics (TOG), 27 (2008), pp. 1–9.

[50] S.-j. Kim, H. Byun, S. Lee, E. Kim, G. M. Lee, S. J. Huh, J. Joo, and H. Shin, *Spatially arranged encapsulation of stem cell spheroids within hydrogels for the regulation of spheroid fusion and cell migration*, Acta Biomaterialia, 142 (2022), pp. 60–72.

[51] M. Konaković, K. Crane, B. Deng, S. Bouaziz, D. Piker, and M. Pauly, *Beyond developable: computational design and fabrication with auxetic materials*, ACM Transactions on Graphics (TOG), 35 (2016), pp. 1–11.

[52] K. Landman and C. Please, *Tumour dynamics and necrosis: surface tension and stability*, Mathematical Medicine and Biology: A Journal of the IMA, 18 (2001), pp. 131–158.

[53] C. H. Lee, A. Varshney, and D. W. Jacobs, *Mesh saliency*, in ACM SIGGRAPH 2005 Papers, 2005, pp. 659–666.

[54] K. W. Lee and P. Bo, *Feature curve extraction from point clouds via developable strip intersection*, Journal of Computational Design and Engineering, 3 (2016), pp. 102–111.

[55] B. Lévy, S. Petitjean, N. Ray, and J. Maillot, *Least squares conformal maps for automatic texture atlas generation*, ACM Trans. Graph., 21 (2002), p. 362‚Äì371.

[56] J. Li, C. Wu, P. K. Chu, and M. Gelinsky, *3d printing of hydrogels: Rational design strategies and emerging biomedical applications*, Materials Science and Engineering: R: Reports, 140 (2020), p. 100543.

[57] M. Li, D. M. Kaufman, V. G. Kim, J. Solomon, and A. Sheffer, *Optcuts: Joint optimization of surface cuts and parameterization*, ACM Transactions on Graphics (TOG), 37 (2018), pp. 1–13.

[58] M. Limper, A. Kuijper, and D. W. Fellner, *Mesh saliency analysis via local curvature entropy.*, in Eurographics (Short Papers), 2016, pp. 13–16.

[59] L. Liu, L. Zhang, Y. Xu, C. Gotsman, and S. J. Gortler, *A local/global approach to mesh parameterization*, SGP '08, Goslar, DEU, 2008, Eurographics Association, p. 1495‚Äì1504.

[60] M. Liu and G. Liu, *Smoothed particle hydrodynamics (sph): an overview and recent developments*, Archives of computational methods in engineering, 17 (2010), pp. 25–76.

[61] S. Liu, Q. Liu, and Q. Peng, *Realistic simulation of mixing fluids*, The Visual Computer, 27 (2011), pp. 241–248.

[62] Y. Liu, H. Pottmann, J. Wallner, Y.-L. Yang, and W. Wang, *Geometric modeling with conical meshes and developable surfaces*, in ACM SIGGRAPH 2006 Papers, 2006, pp. 681–689.

[63] M. LIVESU, N. PIETRONI, E. PUPPO, A. SHEFFER, AND P. CIGNONI, *Loopycuts: Practical feature-preserving block decomposition for strongly hex-dominant meshing*, ACM Transactions on Graphics (TOG), 39 (2020), pp. 121–1.

[64] M. MACKLIN, M. MÜLLER, N. CHENTANEZ, AND T.-Y. KIM, *Unified particle physics for real-time applications*, ACM Transactions on Graphics (TOG), 33 (2014), pp. 1–12.

[65] F. MASSARWI, C. GOTSMAN, AND G. ELBER, *Papercraft models using generalized cylinders*, in 15th Pacific Conference on Computer Graphics and Applications (PG'07), IEEE, 2007, pp. 148–157.

[66] S. MATTAPALLY, W. ZHU, V. G. FAST, L. GAO, C. WORLEY, R. KANNAPPAN, A. V. BOROVJAGIN, AND J. ZHANG, *Spheroids of cardiomyocytes derived from human-induced pluripotent stem cells improve recovery from myocardial injury in mice*, American Journal of Physiology-Heart and Circulatory Physiology, 315 (2018), pp. H327–H339.

[67] D. MAWAD, G. FIGTREE, AND C. GENTILE, *Current technologies based on the knowledge of the stem cells microenvironments*, Stem Cell Microenvironments and Beyond, (2017), pp. 245–262.

[68] V. MIRONOV, R. P. VISCONTI, V. KASYANOV, G. FORGACS, C. J. DRAKE, AND R. R. MARKWALD, *Organ printing: tissue spheroids as building blocks*, Biomaterials, 30 (2009), pp. 2164–2174.

[69] J. MITANI AND H. SUZUKI, *Making papercraft toys from meshes using strip-based approximate unfolding*, ACM transactions on graphics (TOG), 23 (2004), pp. 259–263.

[70] K. MUGURUMA, A. NISHIYAMA, H. KAWAKAMI, K. HASHIMOTO, AND Y. SASAI, *Self-organization of polarized cerebellar tissue in 3d culture of human pluripotent stem cells*, Cell reports, 10 (2015), pp. 537–550.

[71] M. MÜLLER, J. BENDER, N. CHENTANEZ, AND M. MACKLIN, *A robust method to extract the rotational part of deformations*, in Proceedings of the 9th International Conference on Motion in Games, 2016, pp. 55–60.

[72] M. MÜLLER, R. KEISER, A. NEALEN, M. PAULY, M. GROSS, AND M. ALEXA, *Point based animation of elastic, plastic and melting objects*, in Proceedings of

the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation, 2004, pp. 141–151.

[73] M. MULLER, M. TESCHNER, AND M. GROSS, *Physically-based simulation of objects represented by surface meshes*, in Proceedings Computer Graphics International, 2004., IEEE, 2004, pp. 26–33.

[74] S. V. MURPHY, P. DE COPPI, AND A. ATALA, *Opportunities and challenges of translational 3d bioprinting*, Nature biomedical engineering, 4 (2020), pp. 370–380.

[75] K. MUSETH, *Vdb: High-resolution sparse volumes with dynamic topology*, ACM Trans. Graph., 32 (2013).

[76] K. MUSETH, J. BUDSBERG, R. HANKINS, T. KEELER, D. BAILEY, AND R. HOETZLEIN, *Openvdb*, in ACM SIGGRAPH 2017 Courses, SIGGRAPH '17, New York, NY, USA, 2017, Association for Computing Machinery.

[77] A. NEALEN, M. MÜLLER, R. KEISER, E. BOXERMAN, AND M. CARLSON, *Physically based deformable models in computer graphics*, in Computer graphics forum, vol. 25, Wiley Online Library, 2006, pp. 809–836.

[78] W. L. NG, C. K. CHUA, AND Y.-F. SHEN, *Print me an organ! why we are not there yet*, Progress in Polymer Science, 97 (2019), p. 101145.

[79] D. Q. NGUYEN, R. FEDKIW, AND H. W. JENSEN, *Physically based modeling and animation of fire*, in Proceedings of the 29th annual conference on Computer graphics and interactive techniques, 2002, pp. 721–728.

[80] T. N. T. NGUYEN, K. SASAKI, AND M. KINO-OKA, *Elucidation of human induced pluripotent stem cell behaviors in colonies based on a kinetic model*, Journal of bioscience and bioengineering, 127 (2019), pp. 625–632.

[81] G. NIELSON, *Dual marching cubes*, in IEEE Visualization 2004 - Proceedings, VIS 2004, 11 2004, pp. 489– 496.

[82] Y. NOMA, K. NARUMI, F. OKUYA, AND Y. KAWAHARA, *Pop-up print: Rapidly 3d printing mechanically reversible objects in the folded state*, in Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology, 2020, pp. 58–70.

[83] S. NUVOLI, A. HERNANDEZ, C. ESPERANA, R. SCATENI, P. CIGNONI, AND N. PIETRONI, *Quadmixer: layout preserving blending of quadrilateral meshes*, ACM Transactions on Graphics (TOG), 38 (2019), pp. 1–13.

[84] J. F. O'BRIEN, A. W. BARGTEIL, AND J. K. HODGINS, *Graphical modeling and animation of ductile fracture*, in Proceedings of the 29th annual conference on Computer graphics and interactive techniques, 2002, pp. 291–294.

[85] B. O'NEILL, *Elementary differential geometry*, Elsevier, 2006.

[86] S. ONGENAE, M. CUVELIER, J. VANGHEEL, H. RAMON, AND B. SMEETS, *Activity-induced fluidization and arrested coalescence in fusion of cellular aggregates*, Frontiers in Physics, 9 (2021), p. 649821.

[87] J. M. OSBORNE, A. G. FLETCHER, J. M. PITT-FRANCIS, P. K. MAINI, AND D. J. GAVAGHAN, *Comparing individual-based approaches to modelling the self-organization of multicellular tissues*, PLoS computational biology, 13 (2017), p. e1005387.

[88] D. PANOZZO, E. PUPPO, AND L. ROCCA, *Efficient multi-scale curvature and crease estimation*, in 2nd International Workshop on Computer Graphics, Computer Vision and Mathematics, GraVisMa 2010 - Workshop Proceedings, 2nd International Workshop on Computer Graphics, Computer Vision and Mathematics, GraVisMa 2010 - Workshop Proceedings, 2010, pp. 9–16.
2nd International Workshop on Computer Graphics, Computer Vision and Mathematics, GraVisMa 2010 ; Conference date: 07-09-2010 Through 10-09-2010.

[89] N. PIETRONI, C. DUMERY, R. FALQUE, M. LIU, T. VIDAL-CALLEJA, AND O. SORKINE-HORNUNG, *Computational pattern making from 3d garment models*, 41 (2022).

[90] N. PIETRONI, S. NUVOLI, T. ALDERIGHI, P. CIGNONI, M. TARINI, ET AL., *Reliable feature-line driven quad-remeshing*, ACM Transactions on Graphics, 40 (2021), pp. 1–17.

[91] N. PIETRONI, E. PUPPO, G. MARCIAS, R. SCOPIGNO, AND P. CIGNONI, *Tracing field-coherent quad layouts*, in Computer graphics forum, vol. 35, Wiley Online Library, 2016, pp. 485–496.

[92] U. PINKALL AND K. POLTHIER, *Computing discrete minimal surfaces and their conjugates*, Experimental Mathematics, 2 (1993), pp. 15 – 36.

[93] C. PLEASE, G. PETTET, AND D. MCELWAIN, *A new approach to modelling the formation of necrotic regions in tumours*, Applied mathematics letters, 11 (1998), pp. 89–94.

[94] M. T. POLDERVAART, H. GREMMELS, K. VAN DEVENTER, J. O. FLEDDERUS, F. C. ÖNER, M. C. VERHAAR, W. J. DHERT, AND J. ALBLAS, *Prolonged presence of vegf promotes vascularization in 3d bioprinted scaffolds with defined architecture*, Journal of controlled release, 184 (2014), pp. 58–66.

[95] L. POLONCHUK, M. CHABRIA, L. BADI, J.-C. HOFLACK, G. FIGTREE, M. J. DAVIES, AND C. GENTILE, *Cardiac spheroids as promising in vitro models to study the human heart microenvironment*, Scientific reports, 7 (2017), pp. 1–12.

[96] L. POLONCHUK AND C. GENTILE, *Current state and future of 3d bioprinted models for cardio-vascular research and drug development*, ADMET and DMPK, 9 (2021), pp. 231–242.

[97] L. POLONCHUK, L. SURIJA, M. H. LEE, P. SHARMA, C. L. C. MING, F. RICHTER, E. BEN-SEFER, M. A. RAD, H. M. S. SARMAST, W. AL SHAMERY, ET AL., *Towards engineering heart tissues from bioprinted cardiac spheroids*, Biofabrication, 13 (2021), p. 045009.

[98] R. PORANNE, M. TARINI, S. HUBER, D. PANOZZO, AND O. SORKINE-HORNUNG, *Autocuts: simultaneous distortion and cut optimization for uv mapping*, ACM Transactions on Graphics (TOG), 36 (2017), pp. 1–11.

[99] A. A. POTDAR, J. JEON, A. M. WEAVER, V. QUARANTA, AND P. T. CUMMINGS, *Human mammary epithelial cells exhibit a bimodal correlated random walk pattern*, PloS one, 5 (2010), p. e9636.

[100] M. RABINOVICH, T. HOFFMANN, AND O. SORKINE-HORNUNG, *Discrete geodesic nets for modeling developable surfaces*, ACM Transactions on Graphics (ToG), 37 (2018), pp. 1–17.

[101] ——, *The shape space of discrete orthogonal geodesic nets*, ACM Transactions on Graphics (TOG), 37 (2018), pp. 1–17.

[102] ——, *Modeling curved folding with freeform deformations*, ACM Transactions on Graphics (TOG), 38 (2019), pp. 1–12.

[103] M. Rabinovich, R. Poranne, D. Panozzo, and O. Sorkine-Hornung, *Scalable locally injective mappings*, ACM Transactions on Graphics (TOG), 36 (2017), p. 1.

[104] H. Ramezani, S. Mohammad Mirjamali, and Y. He, *Simulations of extrusion 3d printing of chitosan hydrogels*, Applied Sciences, 12 (2022), p. 7530.

[105] F. H. Razafindrazaka, U. Reitebuch, and K. Polthier, *Perfect matching quad layouts for manifold meshes*, in Computer Graphics Forum, vol. 34, Wiley Online Library, 2015, pp. 219–228.

[106] R. Reis, *Encyclopedia of tissue engineering and regenerative medicine*, Academic Press, 2019.

[107] B. Ren, X.-Y. Yang, M. C. Lin, N. Thuerey, M. Teschner, and C. Li, *Visual simulation of multiple fluids in computer graphics: A state-of-the-art report*, Journal of Computer Science and Technology, 33 (2018), pp. 431–451.

[108] C. D. Roche, R. J. Brereton, A. W. Ashton, C. Jackson, and C. Gentile, *Current challenges in three-dimensional bioprinting heart tissues for cardiac surgery*, European Journal of Cardio-Thoracic Surgery, 58 (2020), pp. 500–510.

[109] C. D. Roche, P. Sharma, A. W. Ashton, C. Jackson, M. Xue, and C. Gentile, *Printability, durability, contractility and vascular network formation in 3d bioprinted cardiac endothelial cells using alginate–gelatin hydrogels*, Frontiers in bioengineering and biotechnology, 9 (2021), p. 636257.

[110] K. Rose, A. Sheffer, J. Wither, M.-P. Cani, and B. Thibert, *Developable surfaces from arbitrary sketched boundaries*, in SGP'07-5th Eurographics Symposium on Geometry Processing, Eurographics Association, 2007, pp. 163–172.

[111] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, et al., *Fiji: an open-source platform for biological-image analysis*, Nature methods, 9 (2012), pp. 676–682.

[112] C. Schüller, R. Poranne, and O. Sorkine-Hornung, *Shape representation by zippables*, ACM Transactions on Graphics (TOG), 37 (2018), pp. 1–13.

[113] S. Sellán, N. Aigerman, and A. Jacobson, *Developability of heightfields via rank minimization.*, ACM Trans. Graph., 39 (2020), p. 109.

[114] R. A. Serway and J. W. Jewett, *Physics for scientists and engineers*, Cengage learning, 2018.

[115] P. Sharma, C. L. C. Ming, X. Wang, L. A. Bienvenu, D. Beck, G. Figtree, A. Boyle, and C. Gentile, *Biofabrication of advanced in vitro 3d models to study ischaemic and doxorubicin-induced myocardial damage*, Biofabrication, 14 (2022), p. 025003.

[116] N. Sharp and K. Crane, *Variational surface cutting*, ACM Transactions on Graphics (TOG), 37 (2018), pp. 1–13.

[117] I. Shatz, A. Tal, and G. Leifman, *Paper craft models from meshes*, The Visual Computer, 22 (2006), pp. 825–834.

[118] S. Skylaki, O. Hilsenbeck, and T. Schroeder, *Challenges in long-term imaging and quantification of single-cell dynamics*, Nature Biotechnology, 34 (2016), pp. 1137–1144.

[119] J. Solomon, E. Vouga, M. Wardetzky, and E. Grinspun, *Flexible developable surfaces*, Comput. Graph. Forum, 31 (2012), p. 1567,Äì1576.

[120] O. Sorkine and M. Alexa, *As-rigid-as-possible surface modeling*, in Proceedings of EUROGRAPHICS/ACM SIGGRAPH Symposium on Geometry Processing, 2007, pp. 109–116.

[121] O. Stein, E. Grinspun, and K. Crane, *Developability of triangle meshes*, ACM Trans. Graph., 37 (2018).

[122] Z. Sun, M. Costell, and R. Fässler, *Integrin activation by talin, kindlin and mechanical forces*, Nature cell biology, 21 (2019), pp. 25–31.

[123] M. J. Susienka, B. T. Wilks, and J. R. Morgan, *Quantifying the kinetics and morphological changes of the fusion of spheroid building blocks*, Biofabrication, 8 (2016), p. 045003.

[124] Y. Tahouni, T. Cheng, D. Wood, R. Sachse, R. Thierer, M. Bischoff, and A. Menges, *Self-shaping curved folding: A 4d-printing method for fabrication*

of self-folding curved crease structures, in Proceedings of the 5th Annual ACM Symposium on Computational Fabrication, 2020, pp. 1–11.

[125] K. TAKAHASHI, K. TANABE, M. OHNUKI, M. NARITA, T. ICHISAKA, K. TOMODA, AND S. YAMANAKA, *Induction of pluripotent stem cells from adult human fibroblasts by defined factors*, cell, 131 (2007), pp. 861–872.

[126] T. TAKAHASHI, Y. DOBASHI, I. FUJISHIRO, T. NISHITA, AND M. C. LIN, *Implicit formulation for sph-based viscous fluids*, in Computer Graphics Forum, vol. 34, Wiley Online Library, 2015, pp. 493–502.

[127] M. TAKEZAWA, T. IMAI, K. SHIDA, AND T. MAEKAWA, *Fabrication of freeform objects by principal strips*, ACM Transactions on Graphics (TOG), 35 (2016), pp. 1–12.

[128] C. TANG, P. BO, J. WALLNER, AND H. POTTMANN, *Interactive design of developable surfaces*, ACM Transactions on Graphics (TOG), 35 (2016), pp. 1–12.

[129] P. VAN LIEDEKERKE, A. BUTTENSCHÖN, AND D. DRASDO, *Off-lattice agent-based models for cell and tumor growth: numerical methods, implementation, and applications*, in Numerical methods and advanced simulation in biomechanics and biological processes, Elsevier, 2018, pp. 245–267.

[130] A. VAXMAN, M. CAMPEN, O. DIAMANTI, D. PANOZZO, D. BOMMES, K. HILDEBRANDT, AND M. BEN-CHEN, *Directional field synthesis, design, and processing*, in Computer graphics forum, vol. 35, Wiley Online Library, 2016, pp. 545–572.

[131] F. VERHOEVEN, A. VAXMAN, T. HOFFMANN, AND O. SORKINE-HORNUNG, *Dev2pq: Planar quadrilateral strip remeshing of developable surfaces*, ACM Transactions on Graphics (TOG), 41 (2022), pp. 1–18.

[132] R. P. VISCONTI, V. KASYANOV, C. GENTILE, J. ZHANG, R. R. MARKWALD, AND V. MIRONOV, *Towards organ printing: engineering an intra-organ branched vascular tree*, Expert opinion on biological therapy, 10 (2010), pp. 409–420.

[133] L. WADKIN, L. ELLIOT, I. NEGANOVA, N. PARKER, V. CHICHAGOVA, G. SWAN, A. LAUDE, M. LAKO, AND A. SHUKUROV, *Dynamics of single human embryonic stem cells and their pairs: a quantitative analysis*, Scientific reports, 7 (2017), pp. 1–12.

[134] L. WADKIN, S. OROZCO-FUENTES, I. NEGANOVA, M. LAKO, A. SHUKUROV, AND N. PARKER, *The recent advances in the mathematical modelling of human pluripotent stem cells*, SN Applied Sciences, 2 (2020), p. 276.

[135] C. C. WALKER, J. GENZER, AND E. E. SANTISO, *Development of a fused-sphere saft-γ mie force field for poly (vinyl alcohol) and poly (ethylene)*, The Journal of Chemical Physics, 150 (2019), p. 034901.

[136] C. C. WANG AND K. TANG, *Achieving developability of a polygonal surface by minimum deformation: a study of global and local optimization approaches*, The Visual Computer, 20 (2004), pp. 521–539.

[137] H. WANG, D. PELLIS, F. RIST, H. POTTMANN, AND C. MÜLLER, *Discrete geodesic parallel coordinates*, ACM Transactions on Graphics (TOG), 38 (2019), pp. 1–13.

[138] X. WANG, S. RAMÍREZ-HINESTROSA, J. DOBNIKAR, AND D. FRENKEL, *The lennard-jones potential: when (not) to use it*, Physical Chemistry Chemical Physics, 22 (2020), pp. 10624–10633.

[139] N. WEI, K. GAO, R. JI, AND P. CHEN, *Surface saliency detection based on curvature co-occurrence histograms*, IEEE Access, 6 (2018), pp. 54536–54541.

[140] S. E. WINOGRAD-KATZ, R. FÄSSLER, B. GEIGER, AND K. R. LEGATE, *The integrin adhesome: from genes and proteins to human disease*, Nature reviews Molecular cell biology, 15 (2014), pp. 273–288.

[141] A. WITKIN, *Physically based modeling particle system dynamics*, ACM SIGGRAPH Course Notes, (2001).

[142] J. WU, R. WESTERMANN, AND C. DICK, *A survey of physically based simulation of cuts in deformable bodies*, in Computer Graphics Forum, vol. 34, Wiley Online Library, 2015, pp. 161–187.

[143] S. YOSHIZAWA, A. BELYAEV, AND H.-P. SEIDEL, *A fast and simple stretch-minimizing mesh parameterization*, in Proceedings Shape Modeling Applications, 2004., IEEE, 2004, pp. 200–208.

[144] C. YUAN, N. CAO, AND Y. SHI, *A survey of developable surfaces: From shape modeling to manufacturing*, arXiv preprint arXiv:2304.09587, (2023).

[145] X. ZHANG, G. FANG, M. SKOURAS, G. GIESELER, C. C. WANG, AND E. WHITING, *Computational design of fabric formwork*, ACM Transactions on Graphics, 38 (2019), pp. 1–13.

[146] Z.-Y. ZHAO, Q. FANG, W. OUYANG, Z. ZHANG, L. LIU, AND X.-M. FU, *Developability-driven piecewise approximations for triangular meshes*, ACM Transactions on Graphics (TOG), 41 (2022), pp. 1–13.

[147] Z.-Y. ZHAO, M. LI, Z. ZHANG, Q. FANG, L. LIU, AND X.-M. FU, *Evolutionary piecewise developable approximations*, ACM Transactions on Graphics (TOG), 42 (2023), pp. 1–14.

[148] C. ZUPPINGER, *Measurement of contractility and calcium release in cardiac spheroids*, in Calcium-Binding Proteins of the EF-Hand Superfamily, Springer, 2019, pp. 41–52.