

# CARNEGIE MELLON UNIVERSITY

## School of Architecture

College of Fine Arts

### Thesis

Submitted in Partial Fulfillment of the requirements for the degree of

### Master of Science

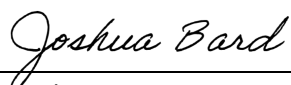
TITLE:

Computational Methods in Knit Design

AUTHOR:

Jamaal Tribune

ACCEPTED BY ADVISORY COMMITTEE:

  
\_\_\_\_\_  
Joshua Bard Principal Advisor

May 10, 2023  
\_\_\_\_\_  
DATE

  
\_\_\_\_\_  
Daniel Cardoso Llach Advisor

May 10, 2023  
\_\_\_\_\_  
DATE

## Acknowledgements

Daniel Cardoso Llach  
Joshua Bard

*I deeply appreciate your support, patience, leadership styles, and overall nurturing of my development in this program and my thesis. You possess some powerful minds that are backed by wisdom in research spaces, your career endeavors, and your time here on this Earth. It was an honor to have you on my advisory team. Your guidance and insights were invaluable throughout the entire process, and I am truly grateful for the time and effort you dedicated to helping me achieve my academic goals. Thank you again for your unwavering support and for being a part of this journey with me. I am excited to take the knowledge and skills that I have gained under your mentorship and apply them to my future endeavors.*

James McCann  
Golan Levin  
Olivia Robinson  
Harrison Apple

*I would like to express my deepest gratitude to the talented artists, technicians, and educators whose studios I had the privilege of being a part of during my studies. Your mentorship and guidance have helped shape me into the creative technologist that I am today. Working alongside you has been an incredible experience that has taught me invaluable skills, both technical and creative. Your willingness to share your expertise and knowledge with me has not only enhanced my technical abilities but has also fueled my passion for interdisciplinary forms of working. Your willingness to challenge me and push me out of my comfort zone has been an invaluable aspect of my growth.*

Precious Parker  
Shiva Peri  
Kedar Benjamin

*I extend my heartfelt gratitude to my dear friends who have been by my side throughout my academic journey. Your friendship has helped me to see my work and myself as valuable, and for that, I will always be grateful. As I move forward from this chapter in my life, I know that your support will continue to be a source of strength for me.*

## Abstract

As a researcher in the field of knit textile production, I conducted an investigation of the interdependent scales involved in this complex process. Through this exploration, I was able to identify opportunities to create a digital fabrication pipeline that addresses the various scales of working with knitted textile materials, using computation. The pipeline itself is an expression of compounding architectures, utilizing a tiered system that encodes various ways of addressing the scales of knitted textile materials using a combination of preexisting software tools and my own scripts. This thesis delves into the challenges and opportunities of integrating computational design methods with knitting processes, with a specific focus on the development of a digital fabrication pipeline for knitted textiles. By examining the properties of yarn, stitch types, design complexities, and garment construction across different scales of the knitting process, from micro to macro and beyond, the research presents a comprehensive workflow for digitally processing knitting machine instructions, offering flexibility and adaptability in the design process. Despite the niche nature of this problem and its inherent challenges, this thesis presents an approach to knit textile production, utilizing the power of computation to address the complexities involved in the process. Through this research, I have identified the limitations and potential of computational knitting systems and emphasize the importance of democratizing access to these workflows. By making these tools more accessible, we can empower designers and researchers to create innovative and high-quality textiles, ultimately contributing to the evolution of the field as a whole. By presenting a digital fabrication pipeline for knitted textiles that addresses various scales of the knitting process, this research opens up new opportunities for innovation in the field.

## Table of Contents

|                                                        |    |           |
|--------------------------------------------------------|----|-----------|
| <b>Introduction</b>                                    |    | <b>1</b>  |
| Extingency                                             | 1  |           |
| Research Questions                                     | 2  |           |
| Aim/Deliverable                                        | 3  |           |
| Literature Review                                      | 4  |           |
| <i>KNITIT</i>                                          | 5  |           |
| <i>Knitted Textiles Exposed to Wind Influence</i>      | 9  |           |
| <i>Multi-Objective Optimization of Knitted Fabrics</i> | 12 |           |
| <i>The Generative Multi-Performance Design System</i>  | 16 |           |
| Background                                             | 22 |           |
| <i>Computational Design</i>                            | 22 |           |
| <i>Knitting As Computation</i>                         | 23 |           |
| Hypothesis                                             | 24 |           |
| <b>Methods</b>                                         |    | <b>25</b> |
| Micro                                                  | 25 |           |
| <i>Contextualization</i>                               | 25 |           |
| <i>Tactile Investigation</i>                           | 27 |           |
| Meso                                                   | 29 |           |
| <i>Contextualization</i>                               | 29 |           |
| <i>Piecing A Pipeline</i>                              | 32 |           |
| Macro                                                  | 37 |           |
| <i>Textile Contextualization</i>                       | 37 |           |
| <i>Design Tool Integration</i>                         | 39 |           |
| <i>Body As A Site</i>                                  | 43 |           |
| <i>Texture Map Experimentation</i>                     | 45 |           |
| <i>Photogrammetry</i>                                  | 49 |           |

Table of Contents

|                                        |           |           |
|----------------------------------------|-----------|-----------|
| Mega                                   | 51        |           |
| <i>Apparel Contextualization</i>       | <i>51</i> |           |
| <i>3D Knitting Tools</i>               | <i>53</i> |           |
| <i>Garment Testing</i>                 | <i>58</i> |           |
| Meta                                   | 60        |           |
| <i>Uncovering The Textile Template</i> | <i>60</i> |           |
| <i>Custom Stitch Meshing</i>           | <i>62</i> |           |
| <b>Research Review</b>                 |           | <b>68</b> |
| Results                                | 68        |           |
| Contributions                          | 70        |           |
| Next Steps                             | 72        |           |
| <b>Bibliography</b>                    |           | <b>74</b> |

## Introduction

# EXIGENCY



Fig 1. Ancient Knitting.<sup>[1]</sup>

Textile producing processes are of the oldest forms of human computation, and textiles themselves are one of the oldest forms of technology. Developed throughout multiple cultures and civilizations, their purposes are multifaceted. Whereas in their behaviors and their uses, textiles are flexible:

We, as humans, adorn and clothe ourselves with textiles. We lay and rest on textiles. We cleanse ourselves and our spaces with textiles. We camouflage ourselves with textiles. We shelter ourselves with textiles. We protect and aid our bodies with textiles. We entomb the deceased in textiles. We create art and narrative through textiles. We gather and collect food with textiles.

Given the ubiquity of textiles and their potential as a valuable technology, asset, and source of opportunity, it's surprising that the commercial textile and apparel industry does not facilitate more avenues for designers and researchers to develop this technology even further. The lack of support and investment in textile innovation outside of institutional research facilities has led to a homogenization of textile products, often rendering them as a mere aesthetic commodity. However, there are emerging research movements that recognize the need for change and are actively working towards more sustainable, ethical, and innovative approaches to textile production within industry. These movements are exploring new materials, techniques, and technologies that have the potential to transform the way we think about textiles and their role in society. With continued research and investment, the future of textiles and their integration with digital technologies looks promising.

---

1. *Ancient Knitting*. February 14, 2014. *The History of Knitting Pt 1: Mysterious Origins*. <https://sheepandstitch.com/the-history-of-knitting-part-1-mysterious-origins/>.

## Introduction

# RESEARCH QUESTIONS

1. How may we recognize and develop apparel design and fabrication technologies in order to gain greater performative control of functional garments?
2. How may we create pathways for apparel and apparel fabrication to communicate with other digital technologies?
3. What data can we use to influence knit structure and thus the performative qualities of knitted textiles?

## Introduction

### AIM

The aim of this exploration is to use computation as a means to create a design-to-fabrication workflow that allows apparel and textile designers to decide knit topologies that satisfy multiple performance criteria, granting designers systemic control of textile properties through a data-inspired process. By decomposing textile materials into various scales, I aim to address how to manipulate these various scales with computationally intervention, with the ultimate objective of codifying information centered around knitted textile materials so that it may communicate with and be informed by data that is collected from the body.

### DELIVERABLE

This thesis delivers a novel digital fabrication pipeline for knitted textiles, addressing the complexities and interdisciplinary nature of digitizing the textile-making process. By overcoming industry barriers and developing a comprehensive workflow that accommodates various stitch types and design complexities, the research contributes to democratizing access to digital fabrication techniques in textile design. The unique pipeline, informed by geometric data and biometric information, showcases the potential for innovation and future research in the computational knitting field.

## Introduction

# LITERATURE REVIEW

This review investigates a variety of projects that probe the intricacies and possibilities inherent in computational methods for textile design, fabrication, and collaboration. Through the investigation of valuable resources for designers, such as pipelines employing familiar tools and languages, we glean insights into the roles of encoding and modular units in the digital representation and fabrication of textiles. The review also highlights the importance of comprehending knitted textiles as physical materials and addresses the challenges and limitations of design technologies when predicting their behavior. It further discusses the potential of collaborative computation techniques, multi-objective optimization, and textile standards to revolutionize knit programming, emphasizing the demand for innovative and high-quality textiles. Ultimately, the review examines the pertinence of generative multi-performance design systems and their wider use cases, offering a structured approach for computational designers navigating the continuously evolving landscape of their field of study.

Additionally, it is important to acknowledge the contributions of scholars and researchers whose work has influenced and informed my own research. While I have not directly cited the works of Delia Dumitrescu's "Relational Textiles: surface expressions in space design", Felecia Davis, Farzaneh Oghazian, and Sam Moradzadeh's "Form-finding of Architectural Knitted Tensioned Structures", and Martin Tamke, Paul Nicholas, and Mette Ramsgard Thomsen's "A new material practice", their seminal work in computational textile fabrication has served as a significant foundation for the context of my research.<sup>[2][3][4]</sup> Their work has effectively demonstrated the vast potential of digital fabrication techniques in the creation of intricate and complex knit textiles. Their innovative contributions have inspired a wave of research in this field.

With their extensive work in this area, they have highlighted the unique properties and capabilities of digital fabrication techniques in knit textile production. By exploring the vast array of materials and design possibilities, their work has shown that digital fabrication techniques offer a tremendous potential for the production of functional and new aesthetics with knitted structures. As my research has progressed, I have had the opportunity to build upon their foundational work, and hope to situate my research alongside theirs, with the ultimate goal of advancing the field of computational textile fabrication.

- 
2. Dumitrescu, Delia. "Relational Textiles: Surface Expressions in Space Design," 2013.
  3. Davis, Felecia, Farzaneh Oghazian, and Sam Moradzadeh. "Form-Finding of Architectural Knitted Tensioned Structures," 2022.
  4. Tamke, Martin, Paul Nicholas, and Mette Ramsgard Thomsen. "A New Material Practice," 2012.

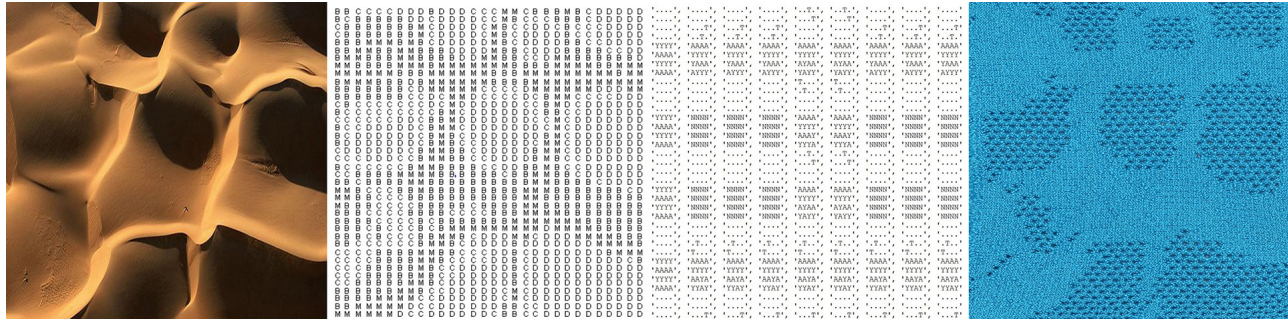


Fig 2. The KNITIT Pipeline: Input Data Is Translated into Strings, Which Are Transformed into an Array of Machine Commands for Fabrication<sup>[5]</sup>

## KNITIT

### ► Project Synopsis

KNITIT is a computational tool—a pipeline—developed collectively in The David & Barbara Blumenthal Israel Center for Innovation and Research in Textiles (CIRTex).<sup>[6]</sup> The center is hosted in the Shenkar College of Engineering, Design, and Art, which is located in Ramat Gan, Israel. The pipeline uses image data as a means to encrypt the stitch and color composition of flatbed machine, weft knitted textiles, creating a means for users to produce complex gradations of stitch structures and yarn color through a non-manual design process. However, while the tool has the ability to produce some compelling textile outputs, the research documentation seems to focus more heavily on the discrepancy between the computational design of textiles and the resulting fabricated output of those physical, textile artifacts. The tool and the corresponding research document addresses this large gap in readily available solutions that allow designers to gauge the physical deformations of knitted textiles on account of their stitch structures. Being able to predict the behavior of these complex material assemblies permits more computational control and iteration in the digital stage of the design process, which ultimately would grant designers the liberty to architect more complex knitted assemblies with more accuracy and precision. Therefore, KNITIT additionally provides a physical simulation component that allows the user to gauge the amount of approximate material deformation that the textile will exhibit as result of its fabrication. Naturally, KNITIT's simulation component does not address the entire opportunity space regarding the physical discrepancy of computationally designed soft goods and the corresponding fabricated artifact. Yet, the pipeline acts as a single solution that contributes to bridging such a seemingly nebulous gap. The resulting pipeline constructs a means to: “Create a fabric with multiple knitting structures [...]”, “Interchange these structures parametrically, in a way that does not alter the overall geometry and dimension [...]”, and “Knit directly from the tool”.<sup>[7]</sup>

5. Karmon, Ayelet, Yoav Sterman, Tom Shaked, and Eyal Sheffer. “KNITIT: A Computational Tool for Design, Simulation, and Fabrication of Multiple Structured Knits,” 2018, 1.

6. Karmon et al. “KNITIT” 2018, 1.

7. Karmon et al. “KNITIT” 2018, 3.

The KNITIT pipeline is arranged in the following stages:

1. A parametric interchangeable input device, whereas image, numerical, vector, raster and potentially other data formats can serve as input
2. A visual flexible interface, that analyzes the input data and relates it to a corresponding knit structure from a pre-composed library of supported structures
3. A visualization and compensation tool, that evaluates the predicted deformations in the knitted textile with a spring-based physical engine
4. A compiler, that converts the visually simulated textile into machine code files that can be interpreted by the appropriate knitting machine

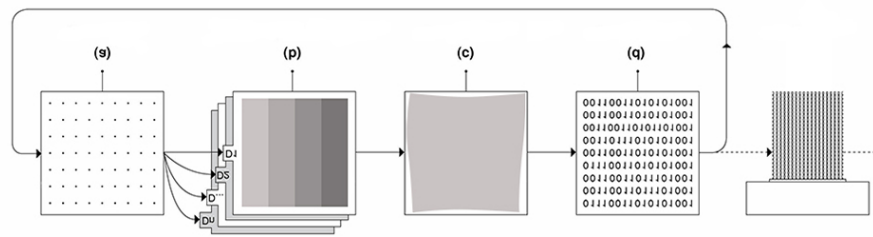


Fig 3. Flow Diagram: (a) grid set-up (b) sampling and allocation of knitting structures (c) compensation evaluation and simulation (d) knitting machine code output.<sup>[8]</sup>

#### ► Insights

The KNITIT pipeline proved to be a very valuable source to reference because the workflow is inline with my non-purist approach to computation.

• From my own anecdotal experience, I've come to understand that architects and designers working through computational means often find themselves as hackers. Whereas, a hacker creates pathways for differentiated tools to aptly communicate with each other in order to address a challenge.

With a growing proficiency in computational systems and nascent experience in creating computational workflows, I sought the guidance of other projects to understand what was possible with my current skill set. KNITIT demonstrated to me that it is possible to create a pipeline using many of the tools and languages in which I am currently choosing to build my primary expertise. According to the KNITIT research documentation, its platform is based on Rhinoceros 3D (Rhino)—a commercial 3D computer graphics and computer-aided design (CAD) application—and Grasshopper 3D—a visual programming language and environment that runs within Rhino. These devices are two of the primary “go-to” tools for computational designers that are designing algorithmically, especially with the intention of working both digitally and physically.

• Contrarily, I would consider a computational purist as one that builds her own tool that is more or less self-contained, rather than one who seeks to build pathways for various systems to communicate.

8. Karmon et al. “KNITIT” 2018, 3.

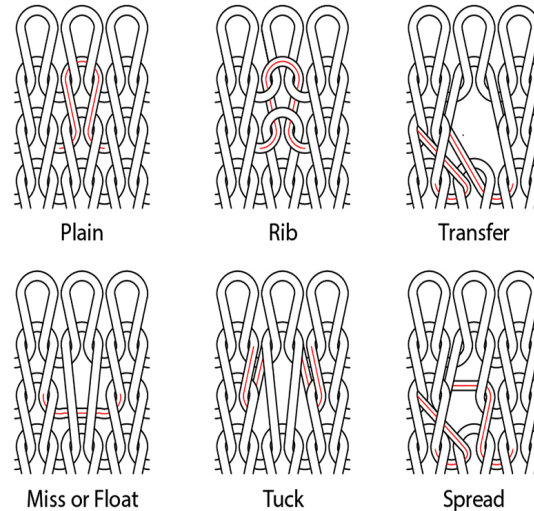


Fig 4. Technical face notation of commonly used weft knit constructions. This project employs a variety structures based on a repeat sequence of Transfer and Tuck stitches.<sup>[9]</sup>

Likewise, there are also tools in this pipeline in which I have little or no experience but currently have enough computational literacy in similar tools to understand their use. KNITIT utilizes the Kangaroo physical engine plugin for Grasshopper, which is a module that allows users to simulate the predicted behavior of a digitally represented mesh soft body in Rhino. The physical engine creates a dynamic iterative simulation until the designated geometry best satisfies a prescribed set of input parameters, otherwise coming to a state of equilibrium. The creators of the pipeline also used Python as a programming language, granted that Python is a flexible, high-level language and can be used to easily manage and process data points. Moreover, in order to fabricate a digitally represented textile, the CIRTex team used two notable types of file formats—the first being a Jacquard file, a file containing a 2D matrix array of characters representing all knitting commands of a designed textile. The second file format is a Sintral file, which is a STOLL provided machine-level control language that is able to continuously pull from the Jacquard file to give the knitting machine clear, machine interpretable instructions.

With there being a set amount of operations that a needle is able to perform, the KNITIT pipeline defines its fundamental needle operations to be:

1. **Knit** ( operation also creates a Purl ) - instructs the needle to draw a new yarn through a previous loop held by the needle, forming a new loop.

---

9. Karmon et al. "KNITIT" 2018, 2.

2. **Tuck** - an operation which adds a new yarn to a needle, either previously holding a loop or empty.
3. **Miss/Float** - instructs a needle not to operate, allowing the new yarn to pass without getting hooked.
4. **Transfer** – instructs a needle to pass the held loop(s) to an adjacent needle, which is either empty or already holding a loop (or loops). On some machines, passing a loop to a needle on the same needle bed is not possible and requires a two-step operation.
5. **Split** - an operation which combines knit and transfer into one operation. It instructs the needle to knit through a loop onto the opposite needle bed, without losing the loop in the original knitting needle. <sup>[10]</sup>

These instructions cover all of the basic single-needle operations, and the authors also included split, which can be considered as a combination between knit and transfer. Likewise, these instructions can be architected into a set of instructions that create a more complex procedure, which can be referred to as a stitch pattern or knit structure. Within this project domain, the research group created a library of seven different knit structures. Four of the knit structures are “variations of a sequenced transfer knitting stitch, and differ in density: from a dense, double-face tuck stitch structure, to a loose, net-like structure”. While the remaining three stitch patterns are “based on variations of a three color knitting technique called ‘Bird’s eye Stitch’”.<sup>[11]</sup>

Furthermore, the KNITIT project introduces a crucial methodology that is foundational for the development of my own research: encoding. KNITIT introduces the importance of developing a language—or more appropriately, a language system—that is adequately flexible and detailed to traverse translation from digital representation, to machine code, and then fabricated artifact. With each needle operation being encoded as a data type, the research team composed a library of digitally represented stitch patterns. Each stitch pattern can then be represented as a single procedure, which heightens the potential for design complexity. Employing the commonly used 2D matrix as a general constraint, the pipeline relies on modular, discretized units to represent knitting instructions and color through pixels and type characters. Using such simple and unique means of encoding creates clear opportunities for a designer such as myself to translate such data to another system for processing.

---

10. Karmon et al. “KNITIT” 2018, 5.

11. Karmon et al. “KNITIT” 2018, 6.

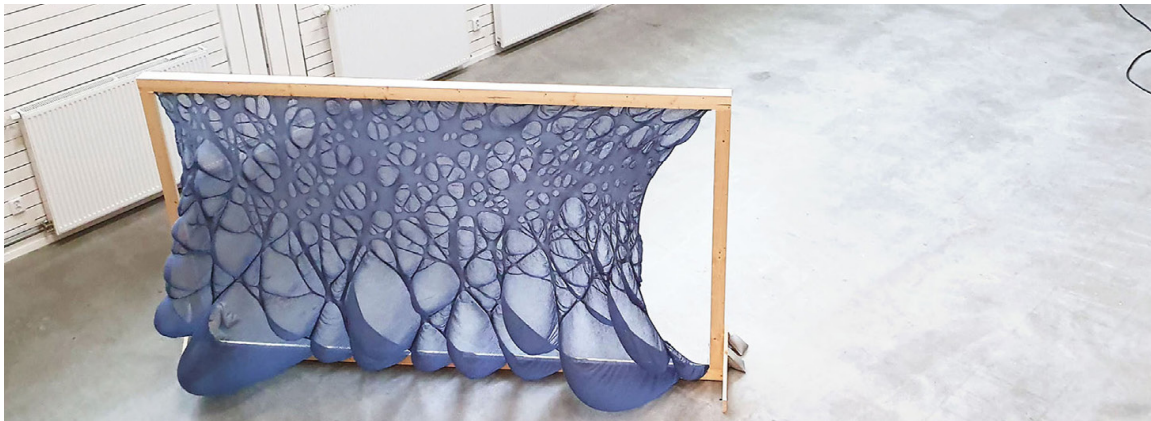


Fig 5. Wind-machine set-up for the larger drop-stitch sample<sup>[12]</sup>

## Knitted Textiles Exposed to Wind Influence

### ► Project Synopsis

Using wind as the primary design consideration, this project is a design study that explores how weft knitted textiles can be used as dynamic, exterior, architectural materials that may serve to be an alternative option to their more static counterparts. The project was conducted by Erica Hörteborn and Malgorzata Zboinska of the Department of Architecture and Civil Engineering at Chalmers University of Technology, located in Göteborg, Sweden.<sup>[13]</sup> Moreover, the department is multidisciplinary, whereas it seems to be broad in its exploration of both humanistic and scientific endeavors. Such endeavors are perhaps indicative of what is commonly thought to be the relationship between architecture and civil engineering: seemingly opposing pursuits that complement each other. As mentioned in the research documentation, the purpose of this particular study is to explore how knitted textiles may be used as a responsive material, responding to wind. Additionally, this effort also investigates interesting aesthetic and geometry possibilities that emergers when designing in such a manner.

When imagining a design-to-fabrication workflow, especially when designing through digital means, it is commonly assumed that the process will be linear, starting with a sketch that exists on paper or in digital space and ends as a machined product. However with textile experiments, the process is often bottom up, where it's necessary to access the textile's physical properties prior to accurately representing and fabricating a textile with complex structure and behavior.

Divided into two main types of experimentation, the methods section documents the experiments as either being a physical or digital experiment. In their physical experiments, the researchers used both a manual knitting machine and a CNC (computer numerically controlled) flat bed knitting machine in order to fabricate knitted textiles.

While describing their knitting technique investigations, the authors argue that there are three levels of hierarchy when considering knitted textiles:

1. Fibre Level (material composition)
2. Yarn Level (twist and tread in the yarn)
3. Knitted Structure/The Textile Level (stitch-composition)<sup>[14]</sup>

The researchers continue assert that varying the elements that compose each level of hierarchy has the potential to conjure virtually an endless amount of design possibilities. In their set of physical experiments, the researchers vary the stitch pattern—stitch pattern being their proposed highest level of hierarchy—with aims

I find it interesting that the authors of this work structured their research document to order the physical experiments first. This is in likeness to the order in which I've conducted my physical and digital experiments.

12. Hörteborn, Erica, and Malgorzata A Zboinska. "Exploring Expressive and Functional Capacities of Knitted Textiles Exposed to Wind Influence," 2021, 6.

13. Hörteborn and Zboinska "Exploring" 2021, 1.

14. Hörteborn and Zboinska "Exploring" 2021, 6.



Fig 6. The drop-stitch pattern combined with three-colour jacquard. Left: the right side of the piece. Right: the purl side of the piece in detail.<sup>[15]</sup>

of adding three-dimensionality in what would normally be a flat knitted textile. With this added dimensionality, the research duo sought to explore the relationship between very pronounced knit structures and their response to wind. The researchers decided to explore four knit structures, which they describe as tuck stitch, hanging stitches, false lace, and drop stitch, each displaying distinct behaviors in terms of material deformation and response to wind.

To more accurately digitally represent the behavior of textiles in motion, the researchers used a myriad of tools and theoretical approaches. The research duo defined the nature of this problem as being a fluid-structure interaction (FSI) problem, a complex branch of computational fluid dynamics (CFD), due to the dynamic nature of the mutual, corresponding

forces enacting between wind and textile. This demands considerable computing power. To simulate loosely moving textiles the team decided to explore smoothed particle hydrodynamics (SPH), a mesh-free method used in computer animation that has the potential to simulate close to real-time. The researchers used Rhinoceros 3D, Grasshopper, Kangaroo 2, and Flexhopper—the last two tools being Grasshopper plugins that permit users to simulate textiles and windforce with real-time visual feedback. However with their decided approach, the research team found that it was quite difficult to accurately simulate the behavior of each stitch pattern and their response to windforce. Requiring a such high level of resolution, the team concluded that their approach is too computationally expensive to create real-time simulations that yield physically accurate results. The calculations needed to bridge digital and physical environments in such a manner are too heavy for even a robust personal computer.

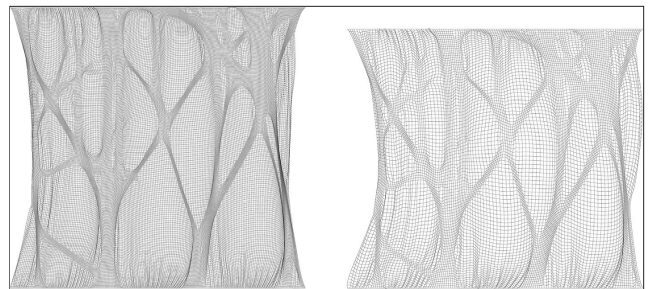


Fig 7. Simulation of the drop-stitch technique, with quad-meshes of varying resolutions, produced with Kangaroo 2. The mesh to the left is also stretched slightly in the simulation to closer represent how the physical piece was mounted on the frame.<sup>[16]</sup>

#### ► Insights

In this project, Hörteborn and Malgorzata demonstrated the complexities that easily arise when attempting to accurately understand knitted textiles as physical materials. First and foremost, scale is important. In any additive procedure, the fabrication process itself increases the level of material organization, thus increasing the scale of complexity of the fashioned material assembly. In weft knitting, one or more yarns are continuously assembled to themselves using various knit operations to form a textile, and we can assume that the basic unit of a knitted textile is a simple knit stitch. Moreover, to traverse the scales of magnitude from a yarn to even a palm sized textile knitted at full gauge (knitted on every needle),

15. Hörteborn and Zboinska "Exploring" 2021, 1.

16. Hörteborn and Zboinska "Exploring" 2021, 6.

the knitting machine must produce many knit units. One of the woes that Hörteborn and Malgorzata experienced in their experiments is the sheer magnitude of knit units they were attempting to digitally calculate in real-time, causing a computational bottleneck. In order to relieve this bottleneck, the duo would have to sacrifice resolution and the accuracy of their digital model—bringing me to the second key takeaway of this project: behavior. Not only was this team attempting to predict the behavior of textiles at such a granular scale, they were also nobly attempting to predict how these textiles responded to windforce.

Using digital means to predict the physical properties of complex, knitted textiles is still a relevant challenge in research. Furthermore, the complexity of the research problem is compounded in the attempt to digitally predict the behavior of a functionally graded material, rendering it as a fluid, while also predicting its relationship to wind, a turbulent fluid. The amount of physical testing and computation power to successfully achieve this near real-time was well beyond the scope of this team's experimental boundaries. Lastly, I think it's also important to note that the tools they used also potentially obstructed the duo from their initial goals. Whereas, the tools they used—Rhino3D, Grasshopper, Flexhopper, and Kangaroo—are high level design tools that make it easier for architects and designers to easily engage computation to aid them in addressing design problems. However, the ease of use and thus, distance from lower level programming languages could have contributed to some additional computational bottlenecks. While this particular project is interesting and noble in effort, this duo's endeavors demonstrated that current the design technologies that designers readily have access to do not support such an ambitious project with textile mediums.

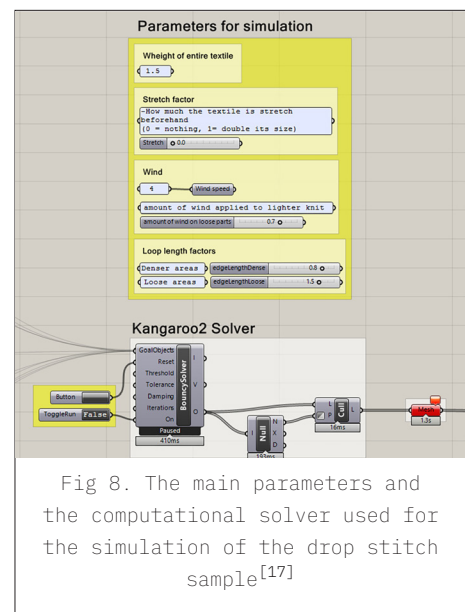


Fig 8. The main parameters and the computational solver used for the simulation of the drop stitch sample<sup>[17]</sup>

17. Hörteborn and Zboinska "Exploring" 2021, 17.

---

Multi-Objective Optimization of Knitted Fabrics► **Project Synopsis**

Submitted as an article to *The Journal of The Textile Institute*, this research project addresses how one may consider conflicting design objectives within a single textile solution.<sup>[18]</sup> The authors are a group of researchers from academic institutions such as the Indian Institute of Technology, the Government College of Engineering & Textile Technology, and so forth. South Asian academic institutions, including Bangladesh, Pakistan, and India, have continued to sustain the region's long and rich history of textile production, engineering, and computation. This research further authenticates this region's ancient history with these disciplines and as well as exhibits how researchers are advancing these disciplines. Engineers and technicians from this region of the planet seem to be more commonly examining textiles as a technology in research. In this particular article, the research group evaluates how a textile may best interact with the body's thermal radiation while providing ample ultraviolet (UV) protection.

Choosing to express UV resistance through ultraviolet protection factor (UPF), the team is able to quantitatively determine how well a textile satisfies that goal. The UPF is a number in which its inverse describes the amount of UV rays that can pass through a textile. For example, a textile with a UPF of 40 suggests that 1/40th or 2.5% of UV rays can pass through it. According to the 1996 New Zealand Standard for Sun Protective Clothing (AS/NZS 4399) and the 2007 International Standard for UV-Protective Textiles (ASTM D 6603-07), textiles with a UPF score of 15-24 can be classified as 'good protection'. Textiles with a UPF score of 25-39 are considered 'very good protection', and a score of 40 and above is 'excellent protection'.<sup>[19]</sup> However, the team asserts that a textile with a high UPF may not be comfortable on human skin under the assumption that a densely knitted textile will not aptly transmit air and moisture-vapor from the body. They continue this assertion by stating that in hot and humid conditions, a textile is considered to be comfortable if it's able to allow heat and moisture to dissipate from the micro-climate of the body to the external, surrounding environment.

In addition to providing ample UV protection, this research group also seeks to engineer a knitted textile that satisfies two objectives: ample air permeability and thermal conductivity. While both constraints attempt to enhance heat transmittance, air permeability is related to textile porosity whereas thermal

---

18. Majumdar, Abhijit, Anindya Ghosh, Prithwiraj Mal, and Debamalya Banerjee. "Multi-Objective Optimization of Air Permeability and Thermal Conductivity of Knitted Fabrics with Desired Ultraviolet Protection." *Journal of the Textile Institute*, 2017, 1.

19. Majumdar et al. "Multi-Objective" 2017, 1.

conductivity is related to textile density, causing a conflict. The research group defines convection and conduction as two related methods of heat transmittance to further explain the difference between air

| Experimental<br>Nos. | Level of factors |       |       |       | Single jersey                                               |                                                       |       | 1 × 1 rib                                                   |                                                       |        |
|----------------------|------------------|-------|-------|-------|-------------------------------------------------------------|-------------------------------------------------------|-------|-------------------------------------------------------------|-------------------------------------------------------|--------|
|                      | $X_1$            | $X_2$ | $X_3$ | $X_4$ | Air permeability<br>(cm <sup>3</sup> /cm <sup>2</sup> /sec) | Thermal<br>conductivity<br>(W/mK × 10 <sup>-3</sup> ) | UPF   | Air permeability<br>(cm <sup>3</sup> /cm <sup>2</sup> /sec) | Thermal<br>conductivity<br>(W/mK × 10 <sup>-3</sup> ) | UPF    |
| 1                    | -1               | -1    | -1    | 0     | 116.6                                                       | 39.1                                                  | 11.65 | 36.7                                                        | 57.6                                                  | 75.1   |
| 2                    | -1               | -1    | +1    | 0     | 116                                                         | 39.1                                                  | 11.25 | 38.68                                                       | 56.8                                                  | 74.26  |
| 3                    | -1               | +1    | -1    | 0     | 110.07                                                      | 39                                                    | 11.47 | 38.05                                                       | 57.1                                                  | 76.02  |
| 4                    | -1               | +1    | +1    | 0     | 113                                                         | 39.1                                                  | 11.69 | 35.1                                                        | 56.7                                                  | 79.53  |
| 5                    | +1               | -1    | -1    | 0     | 162                                                         | 37.4                                                  | 8.42  | 61.58                                                       | 46.3                                                  | 52.22  |
| 6                    | +1               | -1    | +1    | 0     | 160                                                         | 38.5                                                  | 8.66  | 63.83                                                       | 47.9                                                  | 48.07  |
| 7                    | +1               | +1    | -1    | 0     | 161.6                                                       | 37.4                                                  | 8.73  | 67.65                                                       | 46.3                                                  | 45.38  |
| 8                    | +1               | +1    | +1    | 0     | 174.6                                                       | 36.9                                                  | 8.22  | 61.14                                                       | 46.1                                                  | 59.19  |
| 9                    | 0                | 0     | 0     | 0     | 138.1                                                       | 37.8                                                  | 9.33  | 53.16                                                       | 52.7                                                  | 59.99  |
| 10                   | -1               | -1    | 0     | -1    | 42.16                                                       | 44.8                                                  | 22    | 10.3                                                        | 67                                                    | 205.96 |
| 11                   | -1               | -1    | 0     | +1    | 226.4                                                       | 35.4                                                  | 5.37  | 85.17                                                       | 46.1                                                  | 34.1   |
| 12                   | -1               | +1    | 0     | -1    | 41                                                          | 44.7                                                  | 25.1  | 14.61                                                       | 69.1                                                  | 170.74 |
| 13                   | -1               | +1    | 0     | +1    | 235.7                                                       | 38                                                    | 5.21  | 88.47                                                       | 45.4                                                  | 30.31  |
| 14                   | +1               | -1    | 0     | -1    | 62.43                                                       | 45.1                                                  | 17.5  | 29.87                                                       | 62.8                                                  | 88.86  |
| 15                   | +1               | -1    | 0     | +1    | 278                                                         | 33.6                                                  | 4.2   | 124.9                                                       | 40.8                                                  | 22.92  |
| 16                   | +1               | +1    | 0     | -1    | 62.59                                                       | 41.9                                                  | 16.98 | 26.93                                                       | 60.6                                                  | 97.66  |
| 17                   | +1               | +1    | 0     | +1    | 290.2                                                       | 34                                                    | 4.05  | 116.29                                                      | 42.2                                                  | 23.69  |
| 18                   | 0                | 0     | 0     | 0     | 121.9                                                       | 41                                                    | 10.49 | 43.5                                                        | 52.5                                                  | 63.71  |
| 19                   | -1               | 0     | -1    | -1    | 44.5                                                        | 46.1                                                  | 19.58 | 15.07                                                       | 68.4                                                  | 165.23 |
| 20                   | -1               | 0     | -1    | +1    | 240.7                                                       | 34.7                                                  | 5.06  | 83.29                                                       | 46                                                    | 35.47  |
| 21                   | -1               | 0     | +1    | -1    | 42.37                                                       | 44.3                                                  | 23.41 | 14.52                                                       | 70.1                                                  | 185.59 |
| 22                   | -1               | 0     | +1    | +1    | 259.2                                                       | 34.5                                                  | 4.18  | 82.28                                                       | 46.7                                                  | 34.84  |
| 23                   | +1               | 0     | -1    | -1    | 65.67                                                       | 43.7                                                  | 16.32 | 24.98                                                       | 59.1                                                  | 116.51 |
| 24                   | +1               | 0     | -1    | +1    | 316.9                                                       | 34.8                                                  | 3.95  | 128.8                                                       | 42.1                                                  | 22.24  |
| 25                   | +1               | 0     | +1    | -1    | 68.67                                                       | 43.3                                                  | 14.53 | 26.4                                                        | 62.2                                                  | 101.71 |
| 26                   | +1               | 0     | +1    | +1    | 319.9                                                       | 33.4                                                  | 3.93  | 123.1                                                       | 41.9                                                  | 21.88  |
| 27                   | 0                | 0     | 0     | 0     | 144                                                         | 37.7                                                  | 10.16 | 49.16                                                       | 52.1                                                  | 58.64  |
| 28                   | 0                | -1    | -1    | -1    | 53.5                                                        | 44.2                                                  | 17.48 | 20.74                                                       | 64.7                                                  | 120.72 |
| 29                   | 0                | -1    | -1    | +1    | 289.8                                                       | 35.1                                                  | 4.06  | 105.21                                                      | 43.8                                                  | 28.64  |
| 30                   | 0                | -1    | +1    | -1    | 46.66                                                       | 44                                                    | 22.26 | 19.35                                                       | 65.2                                                  | 125.99 |
| 31                   | 0                | -1    | +1    | +1    | 260                                                         | 35.5                                                  | 4.3   | 105.62                                                      | 43.5                                                  | 28.28  |
| 32                   | 0                | +1    | -1    | -1    | 49.24                                                       | 44.9                                                  | 21.84 | 19.48                                                       | 63.3                                                  | 124.65 |
| 33                   | 0                | +1    | -1    | +1    | 297.2                                                       | 36.2                                                  | 4.24  | 107.28                                                      | 46.9                                                  | 29.39  |
| 34                   | 0                | +1    | +1    | -1    | 45.45                                                       | 44                                                    | 20.12 | 15.52                                                       | 62.5                                                  | 133.23 |
| 35                   | 0                | +1    | +1    | +1    | 286                                                         | 35.4                                                  | 4.09  | 103.3                                                       | 45.2                                                  | 29.87  |
| 36                   | 0                | 0     | 0     | 0     | 133                                                         | 38.9                                                  | 10.23 | 49.88                                                       | 54                                                    | 57.53  |

Table 1. Properties of single jersey and 1 × 1 rib knitted fabrics<sup>[20]</sup>

permeability and thermal conductivity. The process of convection allows heat to transmit through the pores of a textile, as it's carried through the convection of air. Convection is therefore a process that researchers deem to be closely associated with air permeability. Whereas, a compact and dense textile is the medium in which heat can more easily dissipate, relating conduction to thermal conductivity. The researchers seek to optimize both objectives in a set of solutions and thus, allow the user's needs to decide which textile solution possesses properly weighted objectives.

To computationally address this dilemma, the team decided to utilize multi-objective optimization, as it's an approach that is able to quickly produce many potential solutions that prioritize each objective differently. Distinguishing between two approaches, the classical method "simplifies the multiple

objectives into a single composite objective by considering the weighted sum”.<sup>[21]</sup> Each objective is ranked and subsequently weighted in proportion to their rank/importance. The researchers note that this ranking system proceeds from high-level information, which they define as the conscious bias that the engineer/designer/technician embeds into the investigation. It’s often “non-technical, qualitative, experience-driven and thus subjective”.<sup>[22]</sup> Additionally, because the classical method condenses each objective into a single, composite objective function, it produces only one optimum solution. This grants majority of control to the biases of the technician. On the contrary, the ideal method—as coined by the research group—produces multiple optimum solutions, otherwise known as Pareto optimal solutions or non-dominated solutions. These are solutions that cannot be quantitatively “beat” by another solution. The ideal method can produce many non-dominated solutions for a single optimization problem. This set of solutions is known as the Pareto front of the solution space, where each solution is considered equally plausible as the best solution. It would then be the task of whomever is appointed to choose the solution that best suits her. The control of the chosen solution is thus more democratized. Using the Non-dominated Sorting Genetic Algorithm (NSGA-II) in their investigation, the team ultimately decided to use the ideal method.

In order to produce knitted samples that prioritized each objective differently, the research group chose four parameters of the textile production process to vary: loop length, carriage speed, yarn input tension, and yarn count. These parameters are assigned with numerical values and therefore can be quantitatively measured and managed. The researchers then decided to knit 36 single jersey samples and 36 1x1 rib samples, using a 12-gauge CNC flatbed knitting machine. With each parameter of each stitch type having three numerical variations, the team created a table to categorize each value. Thus, the team was then able to systematically choose from that table the values that each parameter is to be assigned in a given sample. Moreover, based on a value’s position in that table, the team was then able to record the values of each parameter in each knitted sample using a signed binary system. These signed binary values are stored in matrices. The team used TEXTTEST AG’s FX3300 to measure air permeability and an Alambeta instrument, a thermal tester to measure thermal conductivity. Labsphere’s 2000F, a UV transmittance analyser was used to measure the UPF of the knitted samples.<sup>[23]</sup> The use of these testing machines provides a systematic and objective approach to evaluate the performance of fabrics and can serve as an essential tool for textile engineers and researchers to improve the properties and characteristics of fabrics for specific end-use applications.

Ultimately the team found that for single jersey and rib fabrics, loop length, yarn count, the square ( $x^2$ ) of the yarn count are the main parameters that affect air permeability, thermal conductivity, and UPF. Carriage speed had no noticeable effect and yarn input tension only had effect on single jersey fabrics so these parameters were discarded as insignificant for the experiment’s overall goals.

#### ► Insights

While my understanding of multi-objective optimization was very limited by the time I began conducting my own experiments, the idea of granting the computer more liberty to manage data and generate a myriad of effective solutions provoked a desire to investigate it. Computational approaches that enable collaborative decision making between human and computer have the potential to revolutionize the

---

21. Majumdar et al. “Multi-Objective” 2017, 1.

22. Majumdar et al. “Multi-Objective” 2017, 1.

23. Majumdar et al. “Multi-Objective” 2017, 2.

field of knit programming by allowing designers and engineers to have alternate technical and creative pathways to create knitted textiles. Rather than relying solely on preconceived notions of traditional material practices, knit programmers can engineer innovative designs that were previously thought to be too complex or labor intensive to plan and schedule. In this case, NSGA-II aided textile engineers to understand which physical parameters have the most significant effects for their objectives and were able to appropriately modify said parameters. The integration of collaborative computation techniques into textile production enables designers and engineers to consider a diverse array of design objectives and generate innovative and optimized solutions, which would otherwise be prohibitively expensive or time-consuming to explore through manual methods.

Additionally, the use of textile standards, as demonstrated in the research, provides a framework for viewing textiles as a technology, thereby emphasizing the importance of textiles beyond just their aesthetic qualities. Committing a project to satisfying predetermined criteria presents additional guidelines that also act as a common language for textile engineers to assess a fabric's performance in relation to established technological benchmarks. These standards, developed both by industrial organizations or governmental agencies, give a quantitative basis to ensure that textiles meet certain requirements and perform as intended. Standards serve an essential purpose in guaranteeing that textiles operate as planned, helping to ensure that textiles are safe, reliable, and of high quality. The general populace often remains unaware of the existence and significance of textile standards. However, this research has broadened my awareness and knowledge of the extensive domain, and consequently I'm now able to use them as a point of reference when evaluating the performance of textiles.

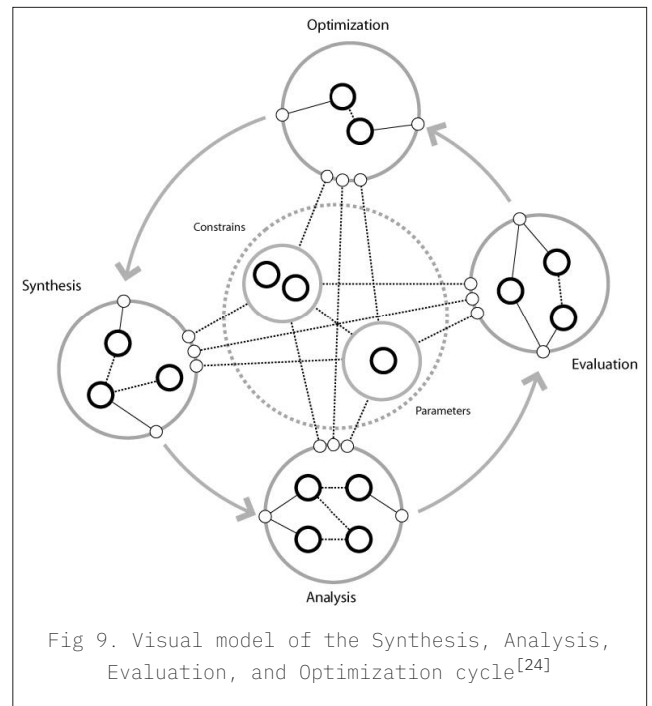
---

The Generative Multi-Performance Design System

## ► Project Synopsis

This research paper is a proposal for a design framework—a methodology that aids the designer to devise a workflow that generates solutions through a computational design system. This work is authored by Anas Alfaris and Riccardo Merello, both of which possess training from the Massachusetts Institute of Technology (MIT). Alfaris is an alumnus of the Design Computation program and Merello—Civil and Environmental Engineering. Unlike the previous projects, the deliverable of this research is not the culmination of digital and physical experiments. Instead, this work aims to evaluate the design process itself, especially under the pretenses of computational collaboration. Thus, the deliverable takes the form of a more abstract tool that one is able to use when planning and executing steps of the design process.

Because the deliverable is more abstract, the authors establish legitimacy by drawing more heavily from the investigations of other research authors—the first of these authors presenting the preciptice that this research paper stands on. Herbert A. Simon, now deceased, was a political scientist and coined the term ‘ill-structured problem’. The term refers to a problem “that cannot be solved by a linear chain of reasoning derived from the problem statement”.<sup>[25]</sup> Alfaris and Merello assert that this is the basis of the design process itself. Design in itself is an act of defining the problem space in which one investigates. Therefore, no single, most optimal solution may exist but yet a multitude of possible solutions. Moreover, due to the nature of the design problem, the research duo assert that in order to




---

24. Alfaris, Anas, and Riccardo Merello. “The Generative Multi-Performance Design System.” Thesis, Acadia, 2008, 449.

25. Alfaris and Merello “Generative” 2008, 449.

commence the design process one must inherently operate from assumption. These assumptions can only be truly verified when the designer arrives at a solution. Therefore, the lack of predictive insight decreases the level of certainty when devising a design process, especially when the process involves computational collaboration. However, this conundrum makes iteration a necessary feat.

•  
*I can attest that in multidisciplinary, academic settings where there are people who are respectively from both design and engineering backgrounds, one can easily distinguish the two by their immediate response to project prompts. Engineers are more accustomed to compounding smaller experiments to form a larger project. Designers are generally more panicked by this approach due to the desire to establish the concept first.*

In attempt to devise a general design framework, the two authors give theoretical structure to the ill-structured problem. Highlighting two popular approaches in systemics, the research duo state that in order to design such a framework one must begin with the top-down approach, which can be characterized by the decomposition of an abstract design concept into specificities or more actionable steps. The concept/idea is prioritized, and the particulars that will contribute to the concept must be determined. The duo are applying this approach to a meta-process—the design framework as previously mentioned. In this case, the meta-process is a process that yields other processes, referring to the design framework that aids to systemically guide design exploration. Alfaris and Merello divide their proposed general framework into what they refer to as ‘mechanisms’. While they do not explicitly define the term, one can infer that a mechanism is essentially a stage of the framework in which the current knowns and created artifacts are further processed and prepared for the next stage of refinement.

•  
Decomposing the design framework into smaller components, the researchers give name to each mechanism: synthesis, analysis, optimization, and evaluation. They mention that problem solving in design is an iterative process, typically cycling through a synthesis mechanism and analysis mechanism. This is what Rowe refers to as the generate and test cycle. The late Marvin Lee Minsky, co-founder of MIT’s AI lab, also coined the term progress principle in his book, *Society of the Mind*, which alludes to the need for an optimization mechanism.<sup>[26]</sup> This additional mechanism guides the search towards a solution rather than aimlessly generating all possible results, providing a metric for progress. Judging each generated design iteration is relatively simple when there is only one design objective to satisfy. However, when there are multiple—possibly conflicting—objectives, recognizing when a potential solution has best satisfied each objective proves to be far more difficult. Alfaris and Merello state that this problem introduces the need for the evaluation mechanism, which considers the high-level decision making needed to address tradeoffs of multiple design objectives. These four mechanisms give definition to the more explicit synthesis, analysis, evaluation, and optimization cycle, rather than just a synthesis and analysis cycle.

•  
*Without explicitly referencing machine learning this research employs the language used in devising machine learning architecture.*

With each constituent of the general design framework loosely defined, the research authors switch their investigation to a bottom-up approach, allowing them to build complexity by compounding simple systems into more elaborate ones. They achieve this by correlating each mechanism to a model. A model can be defined as an abstract description of a real-world phenomenon that serves to represent the functions of a complex system. In layman terms, a model unambiguously describes how a goal will be reached with methods or approaches. Ultimately, these models aid to explicitly define the function that each mechanism ought to possess.

---

26. Alfaris and Merello “Generative” 2008, 449.

Discussing the synthesis mechanism, a (computational) designer is able to decompose a design concept into a set of synthesis models by correlating objectives to a set of parameters or algorithms. A synthesis mechanism should generate solutions. Generative design grammars such as shape grammars, Lindenmayer systems (L-systems), and cellular automata are proper examples of generative approaches. Yet, it's important to note that these algorithmic approaches are not solutions in and of themselves but encode the solution in the resulting topology that they produce.

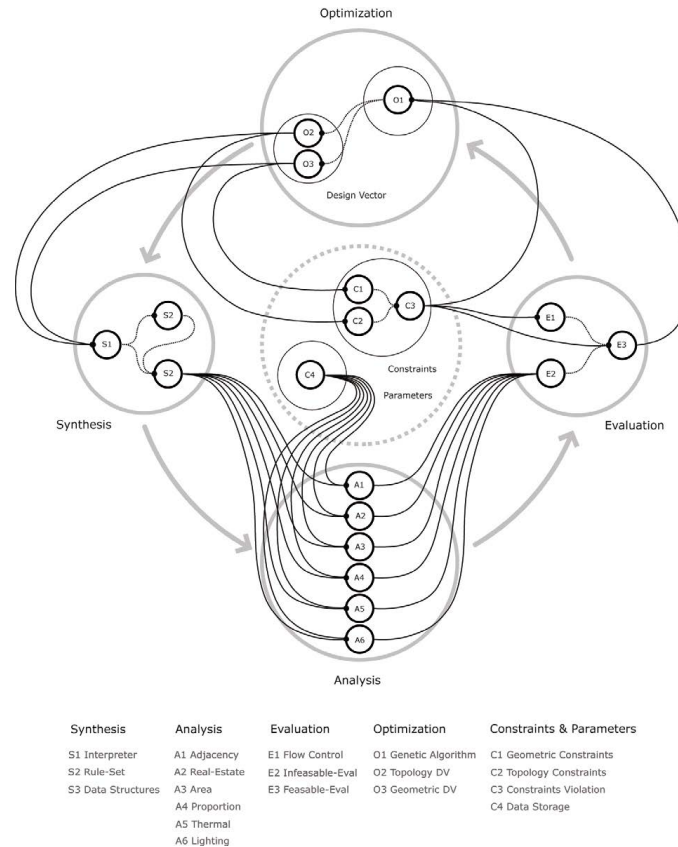


Fig 10. Detailed model scheme for fictional use case<sup>[27]</sup>

The recently passed architect and design theorist, Christopher Alexander, defines analysis in his book, *Notes on the Synthesis of Form*. According to Alfaris and Merello, Alexander defines analysis to be the measure that assesses how well a design solution suits the predetermined set of goals.<sup>[28]</sup> Therefore, the role of the analysis mechanism is to utilize a suitable analysis model that will reveal appropriate performance metrics of a solution. Moreover, Alfaris and Merello distinguish between two classes of analysis models. Low order (low fidelity) models are observational and take a more heuristic approach. The designer

27. Alfaris and Merello "Generative" 2008, 455.

28. Alfaris and Merello "Generative" 2008, 450.

herself uses her previous knowledge and experience to assess the performance of the solution. Secondly, high order (high fidelity) models are based on first principles equations and theoretical models that may supersede human intuition. Examples of such are finite element analysis (FEA) and computational fluid dynamics (CFD), which inherently provide a multitude of data points that are far less biased. Nevertheless while a high order model may appear to provide the most optimal solution for analysis, one must be aware of the tradeoff between fidelity and time—complexity and speed. The designer has to often negotiate the expenses of a process to yield the best deal.

Serving as the decision making tool, the evaluation mechanism needs a model that evaluates quality. Quality is often revealed by an objective function, which produces a numerical value that quantitatively describes the grade of a solution. The research duo discuss two types of problems when attempting to evaluate a solution: single objective optimization and multiple objective optimization. With single objective optimization, a single, most optimal solution can be found if it exists. The search direction to produce a more optimal iteration is relatively simple. However, it's rare that complex, real world problems are presented in this manner, which gives light to multiple objection optimization. A multiple objective problem handles many design parameters and attempts to find an optimal solution by balancing the importance of each parameter. Finding equilibrium between many, potentially competing, objectives doesn't produce a single, most optimal solution however. Naturally, a multiple objective problem can yield many optimal solutions that respectively value one objective over another. Scalarization is a common approach that accommodates the problem of choice in this circumstance. It's a method that combines objectives into a singular scalar cost function.<sup>[29]</sup> The research authors address weighted-sum and utility function as two of the several viable approaches of scalarization one could use. A Pareto set emerges when the best of each tradeoff is found, giving the designer the choice to pick the solution that best addresses the prompt and her values.

While the evaluation mechanism determines the means to critique a design solution, the optimization mechanism determines how to improve a design solution. More generally, the optimization mechanism can be thought of as the compass of the workflow, as it provides a model to guide the search for progress by minimizing or maximizing an objective function. According to the authors, optimization techniques can generally be divided into numerical optimization techniques or heuristics algorithms. Numerical optimization techniques can be further classified into either being constrained or unconstrained optimization. Whereas in constrained optimization, the optimizing variable can only adopt values within a predetermined range. For numerical optimization techniques, this includes the simplex method, sequential quadratic techniques, and the exterior and interior penalty methods. Unconstrained optimization does away with that predetermined range, granting the optimizing variable to adopt a greater range of values. Gradient based algorithms like Newton's method, steepest descent, and conjugate gradient are proper examples. On the other hand, heuristic algorithms gear towards using non-gradient methods to steer towards a solution, which is their appeal. These include methods like evolutionary algorithms, simulated annealing, and tabu search algorithms. Regardless of method, it's important to note that no single approach guarantees the optimal solution—the global optimum—within a search domain. Gradient based methods are more consistent due to their locality, finding similar solutions with their previous runs but may fail to find the global optimal solution. On the other hand, heuristic algorithms are not as local and can find good and diverse solutions in the search domain.<sup>[30]</sup>

---

29. Alfari and Merello "Generative" 2008, 451.

30. Alfari and Merello "Generative" 2008, 451.

However, they lack consistency, presenting a new solution with each run, which lowers confidence of these solutions' optimality.

► **Insights**

Alfaris and Merello offer a generative multi-performance design system to better guide a computational designer along the design process. I find it interesting that without explicitly addressing machine learning these researchers use the language of machine learning architecture to review approaches that designers can undertake in order to better collaborate with computation. Since this paper is dated at 2008, the burst of machine learning innovation around the mid 2010s hasn't quite penetrated the design zeitgeist. From my understanding, this research served to introduce designers to a language and methodology that will be useful as this field of study becomes more advanced. However, because the research duo sought to offer a general design methodology, I find that this approach has broader use cases. Granted that one conceptually understands the language used in this paper, I find that this methodology can extend beyond human-computer collaboration. This approach, I believe, gives structure to the often messy design process. Understanding the process in which humans collaborate with computers to produce and refine design solutions in turn sheds more light on purely human design processes. As I intend on collaborating with computers in this research, I will use this methodology to better understand the nature of the duties of the various aspects of my own workflow.

---

## Literature Review Reflection

Upon reflection of the previously mentioned projects, I have developed a framework of “what, how, and why” to categorize and analyze their contributions to my research. The first two projects, “KNITIT” and “Knitted Textiles Exposed to Wind Influence,” demonstrate the “what” of my research by showcasing the potential of computationally driven fabrication pipelines to output interesting and structurally complex knitted textiles. The KNITIT pipeline provides a comprehensive understanding of how to encode stitches information with computational data, while “Knitted Textiles Exposed to Wind Influence” exemplifies the potential to exploit high-level design tools to manipulate and the structure of a textile at a larger scale. These projects have directly inspired the desired output of my work.

The subsequent two projects, “Multi-Objective Optimization of Knitted Fabrics” and “The Generative Multi-Performance Design System,” inspire the “how” of my work. “Multi-Objective Optimization of Knitted Fabrics” provides a technical account of how to use analytical approaches to achieve a knitted textile with optimal performances by utilizing textile standards, textile structures, and algorithmic approaches. “The Generative Multi-Performance Design System” offers a valuable context for developing a design system using computational thinking. These projects have guided the approach and angle of my research.

The “why” of my research stems from my own intention to address the exigency: developing a pipeline that enables researchers and developers to intuitively drive textile performance with data. By drawing upon the insights and contributions of the aforementioned projects, I aim to develop a pipeline that facilitates the exploration and production of functional and visually appealing knitted textiles.

## Introduction

# BACKGROUND

---

## Computational Design

Computational design is an interdisciplinary process that integrates computational methods to aid various forms of design. Unlike other disciplines such as fashion, automotive, or product design, computational design does not describe the medium being designed but rather the process of manipulating a medium. It is a practice that can enhance any other form of design.

I found that there are three main pillars of computational design: design, simulation, and fabrication.<sup>[31]</sup> Through computational methods, these aspects can be managed efficiently by a single person using a computer. This allows for supply chains and command structures to be streamlined and localized between one person and a machine, offering significant advantages in planning and execution. Architects have embraced computational methods due to the benefits they offer in managing and integrating complex systems and large amounts of data. The use of computers enables architects to achieve precision in integrating various systems and components. However, computational design is not limited to the field of architecture. Many other design disciplines have found value in the principles of computational design, leading to innovative shapes, forms, and performances.

The accessibility of digital tools, rendering designs in digital environments, and the ability to communicate with machinery for precise fabrication has revolutionized architecture. In the same vein, it is interesting to explore how these principles can be transferred to other disciplines, where the term “architect” is transformed from a noun to a verb. This raises the question of how similar principles used by architects in computational design practices can be applied to other mediums. In essence, computational design is a versatile and interdisciplinary process that can enhance various design practices, allowing for improved efficiency, precision, and innovation. By exploring the potential of computational methods, designers from all disciplines can push the boundaries of their respective fields and redefine the possibilities within their creative endeavors.

---

31. Knight, Terry, and George Stiny. “Classical and Non-Classical Computation.” *Architectural Research Quarterly* 5, no. 4, 2001.

---

## Knitting As Computation

Knitting is a unique textile production process that belongs to one of the three major textile categories, including woven and nonwoven textiles. Unlike woven textiles, which are produced through weaving processes, or nonwoven textiles that encompass any textile not created through the algorithmic interlocking of distinguishable yarns, knitting requires only a single, continuous yarn interlocked onto itself through loops. Knitting is an algorithmic process in which a single yarn is used to create loops and various structures through operations such as knit and purl

Knitting's algorithmic nature has led to its application in encoding information, such as encrypting messages within textiles or creating intricate patterns. It is this inherent algorithmic nature of knitting that makes it an ideal candidate for exploration with computational methods, machinery, and digital technologies. Computational approaches can facilitate software interaction with the knitting process, expanding the possibilities inherent in knitting's algorithmic properties. The combination of knitting with digital technologies presents a unique and exciting field ripe for exploration. Digital technologies enable designers to create intricate designs and simulations, explore new materials and shapes, and create innovative structures that would be difficult or impossible to achieve using traditional knitting techniques. Additionally, computational methods can be used to simulate complex textile behaviors and patterns, providing designers with greater control and access to the medium.

The integration of computational methods with knitting offers numerous possibilities yet to be discovered, although it also presents its own set of challenges. Textiles, even with the aid of computation, can be difficult to predict and control, as their behavior stems from the relationship of a large network of individual single stitch units that can vary the textile's shape and performance. Nonetheless, with the appropriate tools and resources, computational design workflows offer the advantages of rapid prototyping, scalability, and digital understanding of the material, yielding opportunities for material exploration. The integration of computational methods in knitting has the potential to revolutionize the textile industry by offering greater control, flexibility, and agility. With the ability to produce complex textiles and forms, computational knitting has applications in various fields such as fashion, architecture, and soft goods design.

---

**Introduction**

## HYPOTHESIS

My hypothesis is that a computationally informed methodology can exploit, codify, and manage the creation of complex knitted textiles at various scales of granularity: micro, meso, macro, meta, and mega. I hope to explore software and develop my own tools to develop these materials in a vertically integrated process that demonstrate the capacity for human-computer collaborations to spawn more elaborate material architectures. In digitally managing these materials, I aspire to create pathways for these materials to work in tandem with data pertaining to the body.

## Methods

# MICRO

---

### Contextualization

In the context of my research, I consider the micro scale to be the yarn itself. Yarns are essentially long, winding, flexible materials that can be composed of many fibers, stretched or extruded from pliable materials, or any combination of both. Considering that my research mainly focuses on stitch patterns, experimentation at this scale did not serve as a primary focus of mine. Regardless, my goal with yarn was to use the same yarn for all of my primary experimentation, even with color. Therefore, I chose to do all my experiments with a white acrylic yarn, defining yarn choice as a controlled variable. This would allow me to understand more deeply how resulting textiles or forms created from the yarn would behave in relation to the knitting process without having to greatly consider how the yarn choice itself affects the results. While the choice of yarn is, of course, an important consideration to understand the greater overarching behavior of the knitted artifact, I found that with the limited time and resources available, it was more important to compare each knitted artifact in relation to each other. Using yarn choice as a controlled variable allowed me to limit the complexity of the measures I was handling.

In the earlier stages of my experimentations, I did opt to explore other more specialized yarn choices. As a means to create more practical design constraints, I sought to define a specific focus group that could potentially benefit from an updated apparel solution through the computationally driven pipeline I'm proposing. Through my inspection, I found that a possible focus group could be structural firefighters, which are firefighters that respond to fires in or around structural and urban environments.

These first responders use turnout gear, or bunker gear, which is the heavy, outer layer of the personal protective equipment (PPE) that firefighters wear to protect them from excessive heat and noxious gasses. The textile components of turnout gear is primarily composed of sewn woven textiles. Because the weaving process can allot for a denser textile than the knitting process, I thought it more worthwhile to inquire into other opportunities where a knitted apparel solution could be more beneficial. In speculation, I presumed that turnout gear may be more effective in protecting the wearer through a tighter textile structure. So as I continued into this inquiry, I found that a potential design opportunity could be to explore how to improve station wear—the uniforms that structural firefighters wear at fire stations and under their turnout gear.

---

Although standard 1975 of the National Fire Protection Association (NFPA) details the highest standards for fire protection gear, firefighters may not always wear the most optimal station wear that could serve as a secondary protection from external stressors and primary aid for the body.<sup>[32]</sup> For example, NFPA 1975 prohibits wearing materials that can melt or drip under heat. However, spandex and polyester garments and undergarments fall into this category. Additionally, the underwire of bras could be made of plastic or metal wire, which may conduct a vast amount of heat close to the body. Moreover, cotton is an approved material to wear as station wear, and through my survey of online, personal anecdotes, many firefighters find cotton station wear to be incredibly comfortable. However cotton, though comfortable, absorbs sweat, which can steam off of the body. Untreated cotton also ignites under fire and isn't durable through abrasion. Therefore in the earlier stages in my experimentation, I sought to acquire high performance yarns, like Nomex yarn, which is Dupont's high-grade solution for a fire-resistant textile material. Because it serves as a secondary layer of protection and there is a leniency for more diverse design solutions, station wear was far more approachable to design for than turnout gear. Additionally, station wear seemed to be a good anchor to constrain the types of design parameters I sought to design for, as I found it quite difficult to define which design parameters I sought to consider.



Fig 11. Flame resistant meta-aramidic yarn, 100% Nomex<sup>[33]</sup>

Yet, I found difficulty acquiring such a flame-resistant, high performance yarn. Even with the professional help of one of my research advisors, getting in contact with any of the manufacturers that produced Nomex yarn, in particular, proved to be very difficult. Through this experience, I've come to understand that the volume in which manufacturers sell yarn far exceeds the quantity of yarn I needed for research purposes. In the case where I were to get any supply of Nomex yarn spools, these spools would most likely be spare spools donated from the textile mill due to how negligible the financial impact would be to the manufacturer whether or not the CMU Textiles Lab purchased a manageable number of yarn spools. Moreover, because we are not a potential large buyer, we, as a small research lab, may not have been the most desirable contact.

The failed attempt to acquire a high performance yarn from manufacturers exhibited how difficult it can be to gain access to cutting edge yarn technology, even through the lab of a global research university. This may be a large roadblock from industry that prevents designers, engineers, and small research entities from having access to fiber technology and therefore hinders the development of potential textile solutions. Additionally, because I wasn't able to acquire a fire resistant yarn, I forfeited using structural firefighters as a focus group and decided to constrain myself to more general performative constraints.

32. Patrick J. Gaffney, "The NFPA 1975 Standard on Station/Work Uniforms: A Necessity or a Luxury?," The NFPA 1975 standard on station/work uniforms: A necessity or a luxury? § (1990), 1.

33. Flame Resistant Meta-Aramid Yarn. n.d. NOMEX Yarns. <https://www.micspa.com/en/yarns/260/nomex->

---

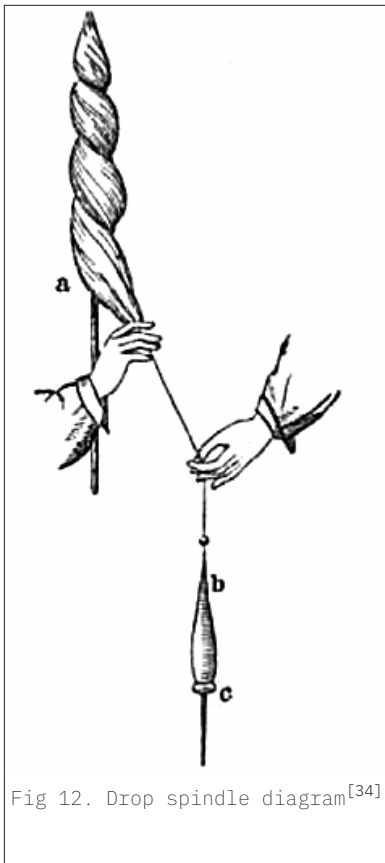
Tactile Investigation


Fig 12. Drop spindle diagram<sup>[34]</sup>

In exploring avenues to advance the industrial knitting process, I've come to understand that when working to establish a digital-to-physical workflow it's essential to develop a relationship with the physical material itself. This is especially true with textiles, which is a class of material that is derived from diverse material compositions, formed by several fabrication processes, and possesses varied morphological assemblies. Understanding textiles at their most basic stages provides a valuable foundation for understanding how their behaviors compound as they assemble into more complex configurations. With this knowledge, there is potential to develop digital design systems that more accurately simulate and predict their behaviors. Digital simulation that accurately reflects the physical properties of these materials enables designers to improve the effectiveness of the design system and allows for interesting types of exploration.

Gaining a deeper understanding of these properties at the micro scale of this research, I spun my own yarn using a drop spindle with the guidance of the Associate Director of CMU's STUDIO for Creative Inquiry, Harrison Apple. A drop spindle is a traditional, handheld tool, usually made of metal or wood, that's used to spin fiber into yarn. Its anatomy is fairly simple, as it consists of two main intrinsic parts: a shaft and a whorl. The shaft is a long stick that the newly formed yarn wraps around and stores itself on. The whorl is the weighted disc that helps tension the yarn.

To use the drop spindle, I, the spinner, first attached a leader to the

---

34. *Spinning with a Suspended Spindle, Spindle (Textiles)* (Wikipedia, n.d.), [https://en.wikipedia.org/wiki/Spindle\\_%28textiles%29](https://en.wikipedia.org/wiki/Spindle_%28textiles%29).

spindle, which is a piece of pre-twisted yarn that is used to start the spinning process. I then attached a length of fiber to the leader, dexterously holding it in place with my fingers. I then twisted the spindle, causing the fiber to wrap around itself, creating a strand of yarn. As the spindle spins, I use the other hand to draft the fiber, pulling it apart slightly to create tension and thinning it out to create a smooth, even strand of yarn. The weight of the whorl helps to maintain the momentum of the spindle as it spins, further allowing me to create a consistent and even yarn. Once enough yarn was on the spindle, I wound the yarn onto the shaft, creating a cop, or ball of yarn.

The properties of handspun yarn are influenced by a multitude of factors: the weight and size of the spindle, the force of gravity, the tension held by the human hand, and the makeup of the fiber material. These parameters work in tandem to ultimately determine the physical properties of the yarn—its thickness, twist, strength, etc. Despite not using the yarn for further experimentation, the process of exploring ways in which its physical and mechanical properties can be manipulated has provided valuable insights into the ways in which yarn crafters can effectively create a customized material even at this scale. The work of my colleagues at the CMU Textiles lab, “SoftNote” for example, proves that researchers can work at scales even smaller than that of the yarn.<sup>[35]</sup> Select projects in this lab introduce conductive materials into more traditional yarns, making yarns that are effectively functionally composite materials. Even at the smallest of scales, textile designers and engineers possess the ability to create customized materials, and understanding this level of control is crucial when working at larger scales and simulating these materials in digital space.



Fig 13. The drop spindle used

35. Yu, Catherine, Willa Yunqi Wang, and Jaiqi Liu. SoftNote, n.d. <https://catherineyu.com/SoftNote-2021-1>.

## Methods

# MESO

---

### Contextualization

In the context of my research, I consider the meso scale to be the stitch. With the use of digital workflows, computationally enabled designers have the ability to communicate with a myriad of digital fabrication tools. Literacy of digital fabrication tools allows for designers to explore the design and construction of physical artifacts. With the use of digital fabrication tools like, cnc mills, 3D printers, laser cutters, etc., designers are able to physically render complex forms and geometries that are more difficult and in some cases, virtually impossible for human hand or any other industrial process to fabricate. Moreover, in the pursuit of creating textile artifacts, the use of CNC tufting machines, knitting machines, weaving machines, CNC embroidery machines, robotic weaving and so forth have served computationally literate designers to gain more precision and efficiency in the production of soft goods. When yarns are structured into more complex material assemblies, the assembling process affects the overall, networked behavior of the architected assembly. In comparison to more rigid materials, like woods and metals, textile behavior seems to be more difficult to predict and control, which also suggests that it's difficult to accurately simulate in a digital environment. Consequently, designing performance driven textiles through digital means may require quite a bit of testing and iteration. For my research, I chose to experiment with knitting as a means for fabrication because of its unique properties as a textile production process. Knitting is algorithmic, and thus, with either the aid of a knitting machine or the patience of the hand, one is able to recreate virtually identical textiles using the same arrangement of instructions in a knitting procedure. Additionally, knitting is an additive process where only a single, continuous yarn is required to create a textile, which attributes to its flexibility to create highly variable textile structures. Moreover, through knitting, one has the capacity to create complex 3-dimensional forms. Knitting is frequently compared to 3D printing, whereas 3D printers commonly have a single material extruder, are an additive manufacturing tool, and are able to quickly create geometrically complex 3D forms.

In this research, I am in the pursuit of testing the performative qualities of textile structure through an algorithmic fabrication process, which is to say that weaving would be the next most promising fabrication process to explore. However, knitting has presented itself to be a more suitable process for my research purposes, because through knitting, one is able to create forms and thus whole, seamless garments.

---

Weaving doesn't possess the same form making capabilities as knitting. So, working only with flat, woven textiles would then lead me to explore how digital design methods could work within a cut-and-sew process. A cut-and-sew process is a garment fabrication process where a seamstress or seamster cuts out pattern pieces from a textile and sews the pieces together to create a garment. Cut-and-sew is the most common fabrication process in production. However, it's hard to automate because it requires awareness of a changing, flexible material and the dexterity to successfully direct the material. Reflecting on choosing to explore the knitting process, it's more apparent that I sought to explore what a nearly automated garment production process looks like, where knitting could produce a whole garment from an algorithm.



Fig 14. Shima Seiki SWG-X 2-bed flatbed industrial knitting machine

With access to the CMU Textiles Lab, I had access to the Shima Seiki SWG-X—a flatbed knitting machine with two needle beds. The machine is 15th gauge, which means that every 15th needle is the distance of an inch. There are two knitting beds—a front knitting bed and a back knitting bed—which aids in the machine's ability to create 3-dimensional form. The machine also has several yarn carriers that one is able to use if there is a desire to knit with multiple yarns. Shima Seiki's design software is KnitPaint. To design a garment or textile, the user of KnitPaint uses the color swatches to then color in a grid pattern in the main window. The colored cells in the pattern are encoded knit instructions, as each color represents a different knit operation.

At the meso scale of my research, I've come to understand that much of this work focuses on the digital

encoding of information. I'm tasked with digitally processing different types of data by integrating various programming tools, establishing communication and interoperability among them. By leveraging the capabilities of multiple types of programming languages, I aim to facilitate the exchange and transformation of data across different domains, with the ultimate goal of achieving a more comprehensive workflow. This process involves identifying key data structures and language protocols, using available pre-established means of data transfer, and developing of custom algorithms and scripts to enable data transfer. Through this approach, I'm able to leverage the strengths of different programming languages and overcome their limitations, thereby achieving a more robust and flexible digital processing pipeline.

To make further sense of this process, I've developed an organizational scheme for classifying different programming languages based on their level of encoding, with higher tiers indicating an added level of encoding. KnitPaint, which occupies tier 1 of the encoding hierarchy, is a tool for schematically planning and scheduling the fabrication of knit textiles on Shima Seiki knitting machines. It's an essential tool that's packaged with Shima Seiki machines, and is considered an industry standard. Knitout, a language developed in the CMU Textiles Lab, occupies tier 2 of the funnel. Knitout is a low-level programming language used to script instructions for the knitting machine in text format. The platform in which it's hosted also provides a basic visual representation of the coded knit fabric, which is useful to debug designs. Tier 3 consists of a custom Python script that utilizes the character data type to encode and store the most basic grammar of a given stitch type so that the pattern may be later articulated into a larger, more elaborate textile. Each character or set of characters represent a needle operation, which is a higher degree of abstraction. Lastly, tier 4 is simply a naming convention to store each stitch type in a stitch library file, which attempts to store stitch types by their common name, such as 'jersey'.

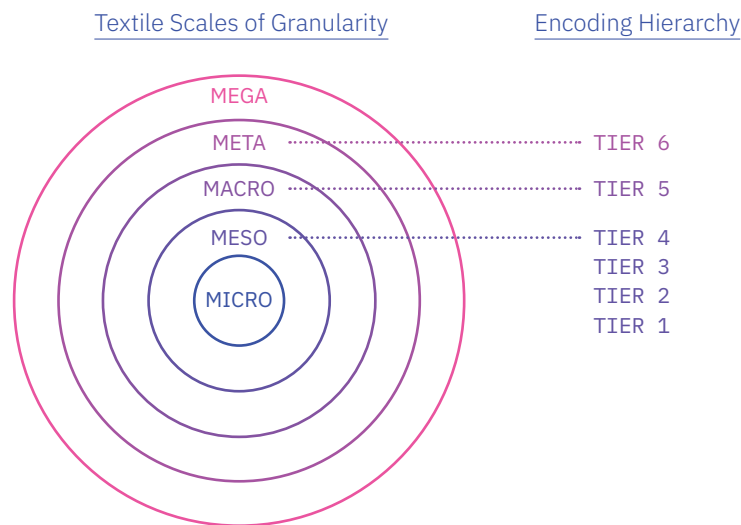


Fig 15. Diagram of textiles scale vs. the developing pipeline's encoding hierarchy

---

## Piecing A Pipeline

### ► Tier 1.

Developing literacy in KnitPaint requires a significant investment of time and effort. Its approach to encoding knit instructions through cell blocks presents challenges in designing garments, particularly because it's far more difficult to understand 3D forms on a flat grid. Strictly operating at the cell-level presents challenges in realizing the design at a global perspective, impeding on the ability to efficiently iterate the direction of a design when creating complex textiles. Moreover with the number of assigned color swatches reaching the hundreds, using color to denote different knit operations creates a potential accessibility issue for users with even the most slight impaired color vision disability. Recalling which color corresponds to which knitting instruction can be challenging, especially when dealing with color swatches that appear identical in hue. Despite the fact that KnitPaint does offer the capacity to create procedures, enabling the user to assign compound knit instructions to color swatches, it does not negate the reality that KnitPaint still lacks the sophistication of more advanced design software. Overall, KnitPaint, as a design software, seems rudimentary as a design software, lacking intuitive representation of design capabilities. To design with greater understanding and sophistication, especially as a beginner with the software, it was necessary to realize the design by other means and to then translate into instructions that KnitPaint could interpret.

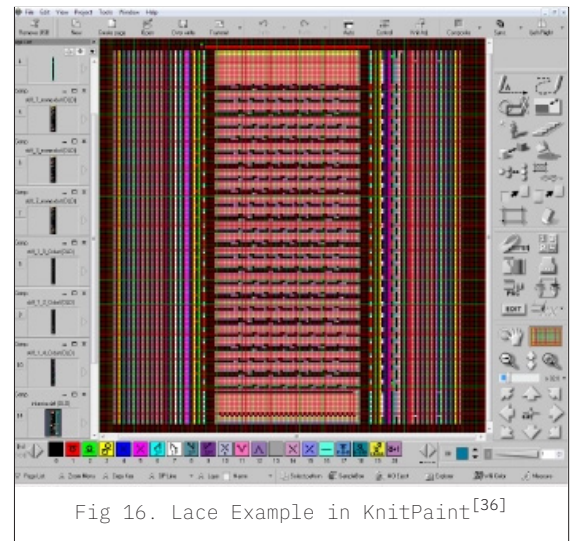


Fig 16. Lace Example in KnitPaint<sup>[36]</sup>

### ► Tier 2.

In order to import a set of knit instructions into KnitPaint, it must be in the form of a “DAT file”. Knitout, a text-based programming solution, provides a means to convert its file type “k” using a javascript

---

36. Yu, Tianhong Catherine, and James McCann. “Coupling Programs and Visualization for Machine Knitting,” 2020, 2.

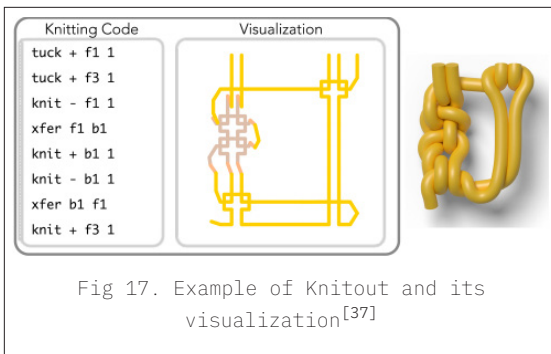


Fig 17. Example of Knitout and its visualization<sup>[37]</sup>

library. Knitout resembles G-code but for the knitting machine, whereas one can assign a needle, a needlebed, or a yarn carriage an instruction at any point in time amongst other things. This enables easier instruction of the knitting machine through written language. Additionally, Knitout has compatibility with Javascript, a feature that I chose not to utilize. Using Knitout, a programmer is able to read the operations that the knitting machine will do in a continuous sequence. In addition to the text based scripting language, the Knitout Live Visualizer is a helpful web-based tool that allows users to visualize the knitting process. Using Knitout code as input, it displays a representation of the knitted fabric that is being scheduled to the knitting machine, which is a helpful

debugging tool.

```

21
22 def appStarted(app, encodedBoard):
23     app.machine = "SWG091N2"
24     app.maxNeedles = 520
25     app.gauge = 1
26     app.position = "Left"
27
28     app.fileName = "DwM_curveAttractorKnitDrawing_1Scaled.k"
29
30     castOn.tuckRows = 2
31     castOn.knitRows = 2
32     castOn.rowNum = castOn.tuckRows + castOn.knitRows
33
34     app.encodedBoard = encodedBoard
35     app.row = len(app.encodedBoard)
36     app.col = len(app.encodedBoard[0])
37
38     char.Instr = ['k', 'p', 'd', 'tf', 'tb', '-']
39     char.knit = char.Instr[0]
40     char.purl = char.Instr[1]

```

```

;!knitout-2
;;Machine: SWG091N2
;;Gauge: 1
;;Carriers: 1 2 3 4 5 6 7 8 9 10
;;Position: Left
inhook 1
tuck - f100 1
tuck - f98 1
tuck - f96 1
tuck - f94 1
tuck - f92 1
tuck - f90 1
tuck - f88 1
tuck - f86 1
tuck - f84 1
tuck - f82 1
tuck - f80 1

```

Fig 18. Snippet of "encodedToKnitout.py" and its output

### ► Tier 3.

To work with more abstract representations of stitch patterns, I developed a custom Python script called "encodedToKnitout.py", which occupies tier 3 of the hierarchy. Tier 3 is where I began intervening between preexisting systems with my own tools. By leveraging the flexibility and versatility of Python, this script acts as an interpreter, translating character strings and integers into machine instructions written in Knitout. This script provides a streamlined method to create Knitout files by automating the process of typing essential header code, generating instructions for stitch patterns at a desired size, and adding the proper cast-on and bind-off procedures, although the bind-off procedure was not implemented prior to fabricating material tests. To use this script effectively, one must represent and store a stitch pattern

37. Yu and McCann "Coupling" 2020, 4.

in its most rudimentary form, which allows this program the ability to work with stitch patterns of any size and calculate which needles ought to be assigned an instruction at a given point in time. Ultimately, “encodedToKnitout.py” allows users to bypass the tedious and error-prone process of manually assigning explicit instructions to each needle. Javascript can be a wrapper for Knitout, allowing users to create code sequences to expedite the manual nature of Knitout. However, scripting in that manner is more manual than using this custom Python script, which mainly requires the basic representation of the stitch pattern and the desired fabric size.

To digitally represent machine instructions in a way that is easily visually recognizable, I encoded a set of knit operations using a distinct terminology of strings and integers. In devising this cipher, I found it important to differentiate operations executed on the front bed versus operations on the back bed. The character ‘k’ represents a knitting action. It’s the most basic and common of machine knitting operations and happens on the front bed. The character ‘p’ represents purl, which isn’t a recognized operation in Knitout. The purl stitch is a combination of two operations: transfer and knit, whereas the yarn is knitted to the back bed of the machine. Tucks, the act of adding yarn to a needle, are represented by ‘tf’ and ‘tb’ and correspond to “tuck front” and “tuck back”, whereas “tuck back” is a combination of a transfer and a tuck. Miss, an (non)operation in which a needle refrains from creating a knit loop, is noted by ‘-’. Lastly, transfers and racking, operations that transfer or shift knit loops to different locations, are represented by an integer. The integer 0 represents a direct transfer of the yarn from one needle bed to the other. Any integer less than or greater than zero within the racking range of the knitting machine denotes the direction and the distance that the front needle bed shifts. This is racking. It’s important to note that while I found it important to codify misses, transfers (as a lone operation), and racking in this cipher, I did not explore stitch patterns that required these three operations. The use of knits, purls, and tucks facilitated a wide enough range of stitch patterns for a preliminary round of experimentation.

|    |               |
|----|---------------|
| K  | Knit          |
| P  | Purl          |
| TF | Tuck Front    |
| TB | Tuck Back     |
| —  | Miss          |
| 0  | Rack/Transfer |

Fig 19. Cipher of machine operations

With these operations encoded in character strings and integers, I stored the values in 2D lists using the Python list data structure. The stitch patterns are stored in their most simplest form in 2D lists. The script then creates an empty 2D list, which I refer to as a board, that then populates in accordance to the desired size, pulling information from the stitch pattern. The board is then iterated through and translated from the cipher into Knitout code where the more explicit instructions are produced. Key information such as which needle bed is performing the operation, the movement direction of the yarn carrier, and the number of the needle performing the operation is calculated based on some changeable parameters and the structure of the board.

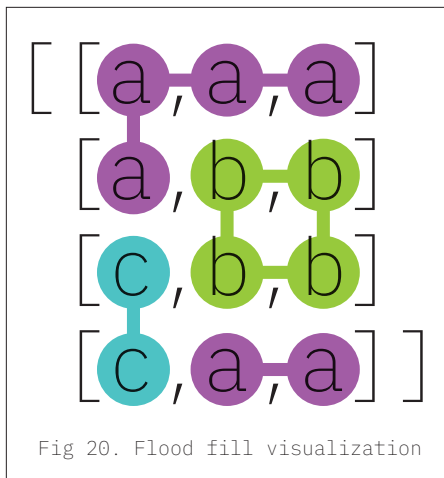
#### ► Tier 4.

Due to the interdependence of the tiers, it’s difficult to explain the function of one tier without alluding to the architectures of analogous tiers. With that being noted, tier 3 is primarily concerned with encoding

single operations—writing commands to control the yarn carriers, needles, and needle bed. Whereas, tier 4 is predominantly focused on the stitch patterns on a larger scale. Moreover, I developed another custom Python script called “typeFilledToEncoded.py” that takes a 2D list—a board—of various stitch types. It translates the board of stitch types into the previously mentioned cipher codex. Therefore, the output of this script is the input of the previously discussed script, “encodedToKnitout.py”. This tier 4 script, “typeFilledToEncoded.py”, possesses the ability to integrate a variety of different stitch types to occupy the same board in a complex configuration. As mentioned previously, a stitch pattern will be stored in a 2D list in its simplest form. In this script, each stitch pattern is further abstracted by representing them with a single character of the alphabet. For example, I can arbitrarily label jersey stitch as “a”, label garter stitch as “b”, and lace as “c”. A board populated with any configuration of these characters is the input for the script.

The logic of the script is underpinned by an iterative procedure that evaluates each position on the board. When a dissimilar stitch type is detected, the algorithm recognizes this position on the board as a primary base point. The algorithm will then search the neighbors of the primary base point to find a neighbor

with a similar stitch type. When the algorithm successfully finds a similar neighbor it becomes a secondary base point. The algorithm then attempts to find similar neighbors around the secondary base point, checking for potential tertiary base points, but the algorithm will eventually return to the position of the primary base point to search another similar neighbor of the primary base point. All the neighboring coordinates on the board with the same stitch type to the primary base point are recorded, sorted, and grouped into an island. Each island is then filled with the stitch pattern that corresponds with its type. Because textiles are typically rectangular, stitch patterns are devised to be repeated through rectangular tessellation. While each island’s shape may be irregular, the algorithm approximates the best-fitting rectangle to the island, enabling it to be populated with instructions from the cipher codex to execute the designated stitch type. Ultimately, each island is placed into a new board at the same coordinates of the original board, and this newly formed board is the input for the tier 3 script. With the guidance of James McCann of the CMU Textile Lab, I was able to implement this algorithm. This algorithm belongs to the class of flood fill algorithms.



This script creates desired complexity in the design process by permitting multiple stitch types, of varying dimensions and shapes, to coexist within a single design plan. In this research, I used four distinct stitch types to conduct this coding experiment and each subsequent inquiry pertaining to stitch types, namely: jersey, garter, 2x2 rib, and seed stitch. Despite the fact that a simple knit operation generates both jersey and garter stitch, the all stitch types possess dissimilarities that warrant experimentation, as their respective aesthetics and performance attributes diverge significantly. Combining these stitch types also presents an opportunity to uncover emergent properties that may arise from their proximity.

The methodological limitations of this workflow also warrant consideration. The present procedure doesn’t account for stitch types that require racking and more intricate transfers. Presently, this script

assumes that each coordinate on the board corresponds to a single needle operation. It's highly plausible that avenues exist where stitch types of more complex architecture can be digitally represented and accommodated. However at present, those research vectors remain unexplored.

## Methods

# MACRO

---

### Textile Contextualization

The macro level of this research focuses on the textile level, where the goal is to output a data type that can serve as input for the meso scale. This stage and beyond introduces potential methodologies that have the possibility to integrate within the previously established workflow but these exercises currently may not fit seamlessly into the pipeline. They are potential avenues that one could potentially integrate into the previously established workflow. These methodologies require more refinement, which is a critical aspect of this research, as its purpose strives to develop a more comprehensive and cohesive approach to computational design in textile fabrication.

As this research progresses, it becomes evident that the macro level onward starts to present more diverse possibilities to integrate into the currently developing pipeline. One of the first experiments conducted at that level introduces Rhino3D and Grasshopper into the workflow. This experiment allows the potential user to interactively manipulate a curve and a graph, amongst other numerical parameters, to dynamically change the arrangement of stitch types on a digitally-represented textile design plan. This dynamic and interactive interface not only provides designers with greater control over the design process but also allows them to explore and evaluate multiple design options rapidly. Notably, this represents the final experiment in the developing pipeline that can be seamlessly integrated with the prior research tests without further adjustments and refinement.

Granted computational design seeks to utilize and develop computer competencies, methodologies associated with this field can aid humans in designing and decision making. Rhino3D and Grasshopper are popular tools commonly associated with the field. Grasshopper, a visual scripting language embedded in the Rhino3D application, allows users to encode information through geometry, offering both parametric and interactive ways to manipulate geometric input data. Rhino3D and Grasshopper create the means to manage and manipulate explicit data types, encouraging designers to think in terms of data and processes. This combination of tools allows those literate in computational design principles the ability to design artifacts that would be difficult to devise through human cognition alone. The present development platform represents an essential stakeholder in the process of bridging the gap between

---

design and engineering. This platform achieves this objective by skillfully combining a range of tools and data that enable designers to create and refine complex geometries with a high degree of precision and efficiency. By leveraging the power of computational design techniques, this platform offers designers a means to explore and evaluate a broad range of design options, while also ensuring that their designs are optimized for key performance criteria.

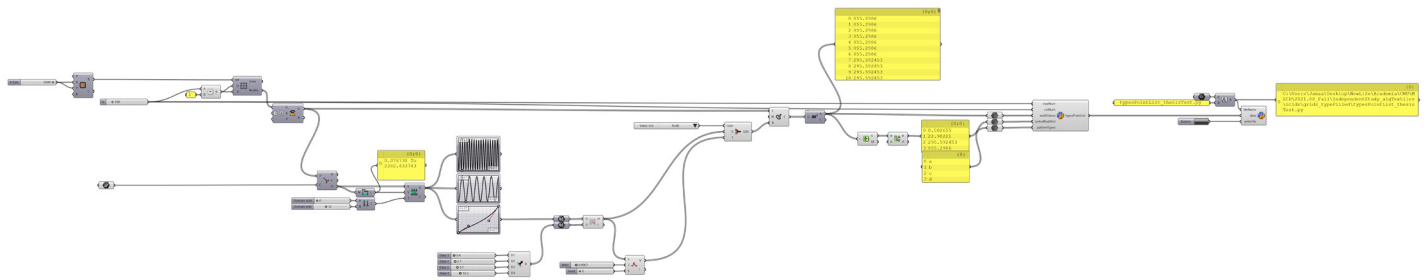


Fig 21. Overview of the textile generating Grasshopper script

---

## Design Tool Integration

### ► Tier 5.

In Tier 5 of my research, I employed a Grasshopper script to create complexity at the textile scale. Although Tier 4 also pertains to textiles to some degree, Tier 5 is more characteristic of exemplifying the textile scale, because I'm modifying the textile at a more global scale.

The primary function of this script is to utilize geometric input to affect the stitch type around it. The size of the grid corresponds to the size of the textile and the number of stitches in that textile. In this script, the grid is populated with circles, an arbitrary method of representing a stitch. By employing circles, I can visually represent various stitch types with the radii of the circles, allowing me to assign a specific stitch type to a radius. Other geometry could have been used to represent stitch types. I could have also used consistently sized circle radii and denoted different stitch types by filling these circles with different colors. However, to avoid recreating the same issues I found with KnitPaint, I opted for a method that visually represents stitches without being overly reliant on color. In future development, the color of the circles could represent the yarn color used at those stitches, further increasing the complexity of this script.

I used a curve attractor to affect the radii sizes, and thus the stitch type, of the surrounding stitches according to their distance from the curve. Theoretically, the curve attractor would represent a particular design constraint, such as the location on the textile that requires more heat ventilation for example. The closer, smaller circles would represent a stitch type that best addresses heat ventilation, while the larger, more distant circles would represent a stitch type that does not need to acutely address ventilation.

In this Grasshopper script, I can also adjust the distance range that assigns the radii of the circles. Consequently, there can be one-to-one correlation between the distance range and the radii of the circles. Grid 1 is most representative of the 1:1 correlation between distance and radii, where the bezier curve that remaps the distance values is very slight. However, I can also remap the range using graphs to achieve interesting and unexpected results. Using more extreme sine graphs, Grid 2 and Grid 3 exhibit more extreme and very different behaviors from each other. Grid 2 is more regular, and the grid appears to

---

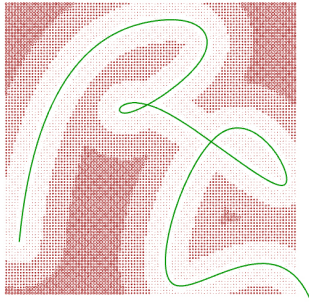


Fig 22. Grid1

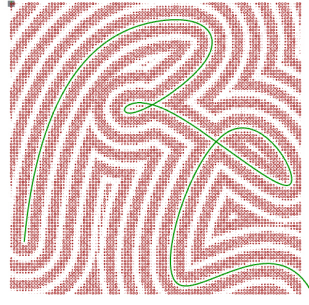


Fig 23. Grid2

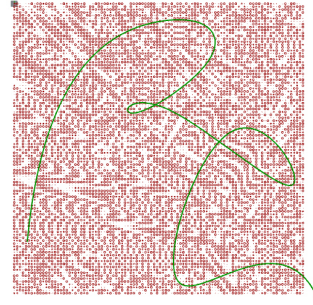


Fig 24. Grid3



Fig 25. Sample1

Fig 26. Sample2

Fig 27. Sample3

visually oscillate, whereas Grid 3 is far less predictable and much more visually complex. Altering the grid to produce a significantly different textile takes only a matter of seconds. Modifying the input curve, changing the size of the grid, reassigning stitch types, altering the graph, and adjusting the range values are all that is required. Intertwining stitch types with such complexity as this script does would take a skilled knit programmer much longer in a manual process.

Fabricating these grids also yielded intriguing results. Using the GHPython component, I assigned each stitch position on the grid according to the size of the circle's radius at that location. I then created a 2D board of stitch types that served as the input for "typeFilledToEncoded.py"—tier 4 of the hierarchy—allowing me to fabricate them. Again, I used four stitch types: jersey, garter, 2x2 rib, and seed stitch. Additionally, it's worth noting that the dimensions of the grids are not directly reflected in the dimensions of the fabricated textile samples. I found that the ratio of the dimensions of the fabricated textile tends to be 4:3 (width:height), whereas in the grid's visualization, I maintained a 1:1 ratio, a square.

The fabrication time for Sample1, 2, and 3 was just under 5 minutes. In these samples, it is easy to see the stark differences between jersey and garter stitch, as they are the densest and directly opposite of each other. The edges where jersey stitch and garter stitch meet tend to buckle and protrude. Distinguishing between 2x2 rib and seed stitch is somewhat more difficult when they do not have enough surface real estate and can appear as visual noise.

To achieve Grid3, I modified the sine graph to have a high amount of frequency, producing a result that could only be quickly realized through computation. When fabricating Grid3 to produce Sample3, the complexity of the textile was lost in its small resolution. To investigate what a higher resolution version of this complex textile would look like, I scaled it up by a factor of 4, resulting in scaledGrid3. scaledGrid3 is far denser than Grid3 while maintaining the overall arrangement scheme of the stitch types. I then repeated the same process for Grid1. Both scaled textiles took approximately 45 minutes to fabricate..

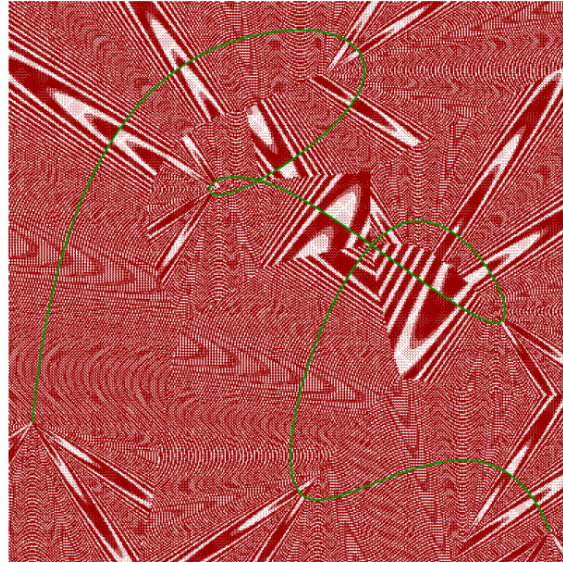


Fig 28. scaledGrid3

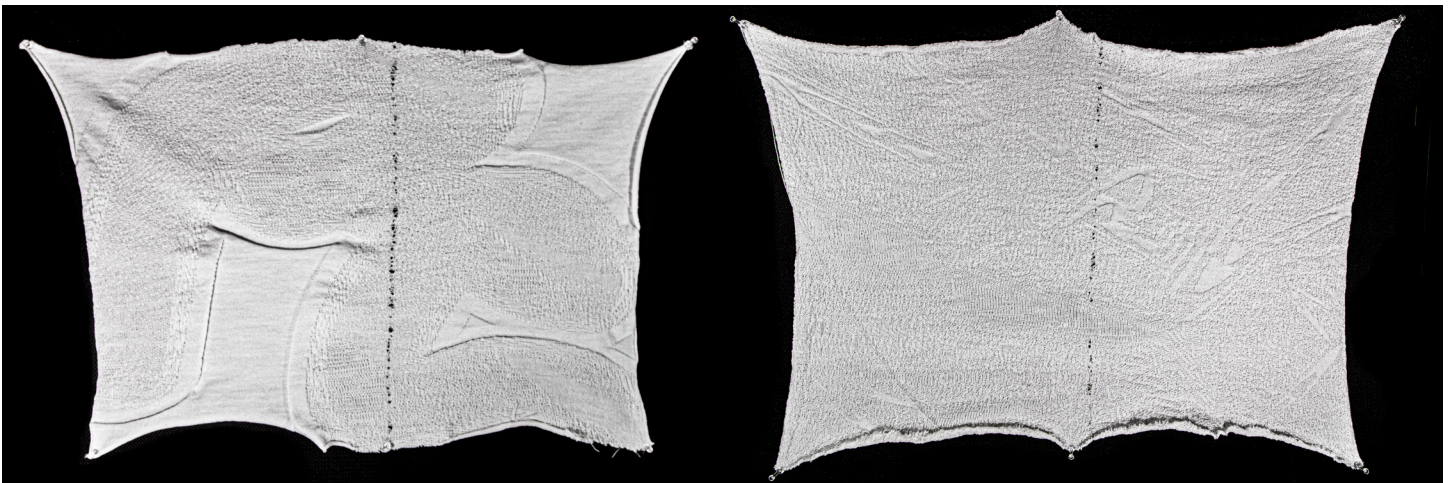


Fig 29. scaledSample1

Fig 30. scaledSample3



Fig 31. Detail shot of scaledSample3

scaledSample3 is undoubtedly a complex textile that would be very difficult to conceive using more traditional ways of designing textiles. It's always noteworthy to highlight that the faulty needle in the knitting machine revealed itself through the missed stitches in scaledSample1 and scaledSample3. This serves as a reminder of how computer, machine, and human all contribute to the production process of textile production in this workflow.

The Grasshopper script used in Tier 5 of my research allows me to create complex and intricate textiles at a rapid pace. The flexibility and adjustability of the script enabled me to explore various stitch types, arrangements, and resolutions that would be challenging to achieve through traditional methods. By utilizing this tool, I am able to iterate through design variations at a much faster pace and explore a wide array of creative avenues.



Fig 32. Size reference of scaledSample3

---

## Body As A Site

The macro level of this research also involves working with a digital model of the human form. A digital approximation of my own body was created by inputting measurements into the CLO3D interface, which parametrically alters the dimensions of a stock avatar. This stage introduces an experiment that seeks to understand the body as a site, akin to how an architect analyzes a parcel of land as an architectural site. Sites can be viewed adaptively, whereas the viewer can examine it through various modalities to suit specific design objectives. Architects often have to consider a variety of factors, including but not limited to the historical significance of the site, social and environmental needs, and topological constraints. Similarly, the human body can be approached through a similar multifaceted lens. By treating the body as a site, the design process can take into account a wide range of factors and modalities that account for its unique characteristics.

In this research, the “skin”, or mesh, of the digital avatar serves as a canvas for storing information relating to the body. However, the scope of the research is limited to representing speculated data, rather than actually collecting and recording data about the body. Some speculated data points entail, but are not confined to heat radiation, sweat gland activity, range of motion (flexion, extension, rotation), abrasion probability, and impact probability. Through the use of texture maps, each data point can be represented on the surface of the mesh, effectively capturing the visual representation of its concentration and distribution at a given locality on the body. The term “heat mapping” is commonly used to describe this process. However, because a heat map can both refer to a texture map that specifically represents temperature—heat—or a texture map that more broadly represents another data point, I will continue to refer to this process as texture mapping in this paper to mitigate confusion.

In this phase of the research, I began developing an algorithm that grants designers the ability to assign a weight to each data point, thus ranking the significance of each design objective in relation to the others. To clarify, an objective refers to an abstract goal to be achieved while a data point is a quantifiable and measurable piece of information that is recorded to help determine the most effective means to accomplish that goal. For example, if the objective is thermal regulation of the body, then heat radiation would be an

---

appropriate data point that corresponds with that objective, as heat energy and location are measurable pieces of information. Ultimately, the goal of this tool was to grant the computer competency in decision-making. While inspired by multi-objective optimization (MOO), the understanding of that technique was limited during the development of the algorithm. I used elements inspired by MOO without having an informed scheme or a verified approach of MOO. Nonetheless, in using texture maps and objective weighting as input, I sought to enable this tool the ability to determine which data point takes precedence at a given locality on the body, creating a new flattened texture map. With the information from the flattened map, I would then be able to create a blueprint of which stitch type to assign to specific localities on the body, assuming a stitch type can appropriately address a design objective. Overall, this phase of the research aims to create avenues to engineer textiles through a workflow that can dynamically employ stitch types in appropriate applications. Through data-informed methodologies, designers and engineers possess the ability to more effectively create textiles and garments that address various performative needs.

---

## Texture Map Experimentation

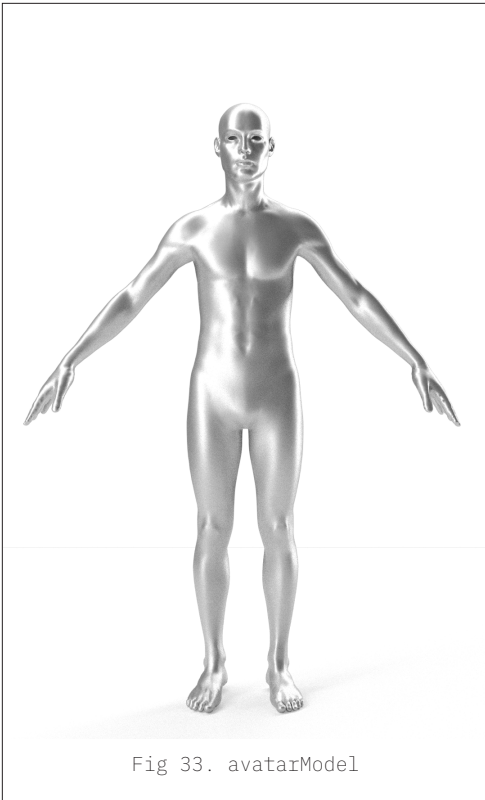


Fig 33. avatarModel

With my body's measurements documented, I entered a diverse range of measurements into a CLO3D interface, allowing me to alter the dimensions of a stock avatar model. This digital model closely approximates my actual body. I then exported the model into Blender, enabling me to work with texture maps. My goal is to create multi-modal maps of my body, generating multiple texture maps using the digital model as the base geometry. Each texture map represents a distinct data point, and I intended to represent various types of information about the body, assigning weights to their importance for a specific design purpose. For instance, in designing a lightweight summer motorcycle jacket, I might prioritize ventilation over abrasion resistance, even though both factors will be considered for the success of the garment.

My next step was to use an image processing algorithm to make sense of these texture maps, deciphering how to best address areas on the body with overlapping texture information. This part of the research asks the question: Which data point is more important at overlapping localities? My first course of action was to unwrap my digital avatar. I performed this task manually, a tedious and challenging process, particularly around the phalanges. This involved placing seam lines on the body to create a flattened UV map, which is then represented as an image. Areas with high curvature required more seam lines; however, due to the manual nature of the process, many high curvature sections of the mesh were not adequately flattened.

The unwrapped mesh is represented in the unwrappedAvatarMesh. As I intended to use an image processing algorithm, I would be working with pixels. An improperly flattened

---

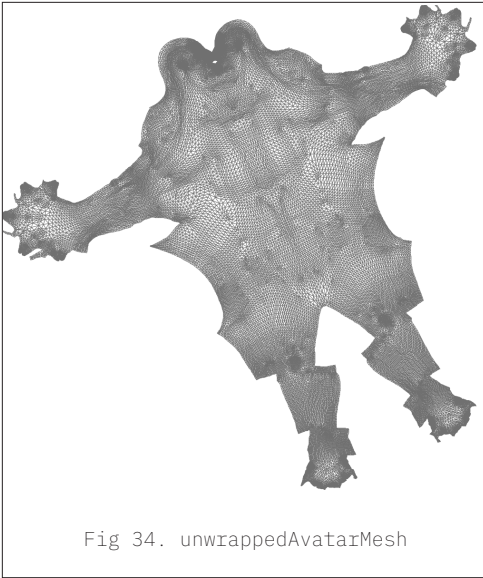


Fig 34. unwrappedAvatarMesh

3D model presents issues since each pixel is a constant size, but each part of the mesh will not be represented equally. Flatter areas, like the bust of this avatar, will have more pixel information than more curved areas, such as the face. Flattening algorithms exist to address this issue, but I didn't use them due to time and scope constraints. This research is primarily focused on digital textile fabrication, and it's worth noting that mapping information on the body is dense enough to warrant its own complete thesis.

The second task was to fill the texture map with information. I created mock data by manually painting each new texture. Various tools and methods could be employed to measure and gather different types of data related to the body. In the Blender Texture Paint workspace, I painted the avatar, which was synced to a specific texture map. This can be observed in the redTexturePainting (Figure 34) and blueTexturePainting (Figure 35). By painting the texture map, I embedded two valuable pieces of information: hue and alpha. Hue (color) indicates the type of data I aim to represent, while alpha (transparency) signifies the quantity of that data type recorded, with more opacity corresponding to a higher

magnitude of that data type.

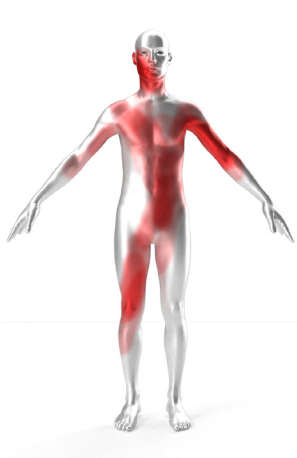


Fig 35. redTexturePainting



Fig 36. blueTexturePainting

Drawing inspiration from Multi-Objective Optimization (MOO), I sought to determine which data type is most important to consider at any given location on the body. To accomplish this third step, I created a Python script that takes painted texture maps of different hues and generates a new, flattened texture map with the "winning" data. I first created two new texture maps and painted them arbitrarily. For testing purposes, they are small, square images, represented by blueSquare (Figure 36) and redSquare (Figure 37). The script, "textureMapProcessor.py", allows the user to assign a weight to each hue, with the

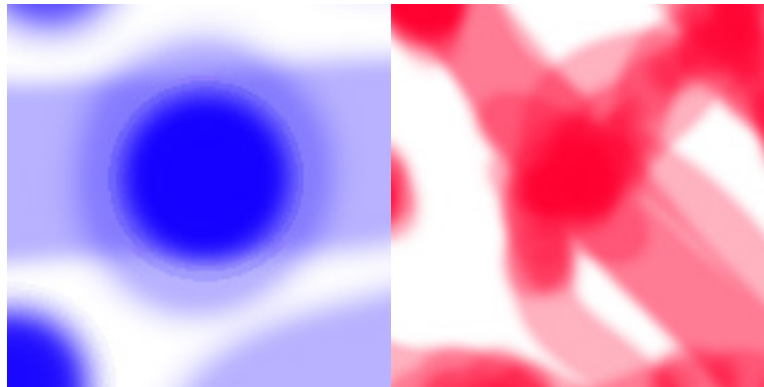


Fig 37. blueSquare

Fig 38. redSquare



Fig 39. combinedColorChannels

Fig 40. outputBlueChannel

Fig 41. outputRedChannel

total weights equaling 1.00. The script then converts the images to grayscale and iterates through each pixel of the texture maps. The algorithm evaluates the score of each texture map at each pixel location, multiplying the weight assigned to that map by the grayscale value at the pixel. The texture map with the highest scoring pixel places its pixel in the new texture map—the output of this script. The objective of this exercise is to encode information about the body itself to inform how a textile or garment may be better engineered. Although I did not integrate this exercise into the pipeline, I believe it has the potential to provide new visual understanding of design constraints.

Additionally, in the pursuit of using texture maps to more specifically influence knit structures, each hue could represent a different stitch type. I could also integrate this workflow into the pipeline by creating a script that approximates curve attractor from the image data—the inverse of creating a curve attractor. I would then be able to combine this workflow with the previous Grasshopper script.

However, there are some notable issues with “textureMapProcessor.py” and other research vectors to explore. The script is based on a zero-sum game, where each data type is competing for real estate on the

final texture map, rather than interacting with each other in alternative ways. The scoring system is also faulty. I have to weigh each input texture map with numerical values that are close to each other, or else the final output texture map will directly reflect the input map with the highest weight. There is much more testing and research to be done in this part of my research.

If Multi-Objective Optimization (MOO) does prove to be an effective approach for processing these texture maps, I would consider using the Non-dominated Sorting Genetic Algorithm II (NSGA-II) for a more sophisticated method. This would allow for a more nuanced understanding of the relationships between different data types and their importance in designing garments or textiles that respond to the unique needs and characteristics of individual bodies.



Fig 42. Input image frames that created my crouching digital avatar

---

## Photogrammetry

As part of my research, I conducted an experiment to further explore the use of digital representation of the body and its potential application in the design process. Thus, I used Agisoft Metashape software, which would potentially allow me to create a more accurate representation of the human body through photogrammetry.

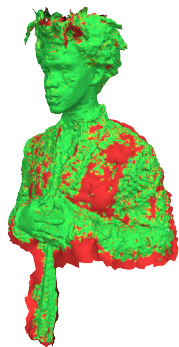


Fig 43. bustConfidenceMap

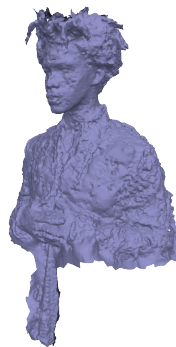


Fig 44. bustSolidModel



Fig 45. bustStereoPointCloud

To capture the physicality of the body, I instructed a colleague to take videos of me from various angles while I stood still. I then created a script to extract the frames of the videos, which I used as input for the Agisoft Metashape software. The photogrammetry process allowed me to create a physically accurate representation of my body which is translated into digital space. In this process, my avatar could be rendered as a solid model, a wireframe, and a point cloud. However, it is essential to note that this may necessitate a thorough data cleansing procedure; however, the resulting model may still not include all the desired information necessary to generate a reliable model. The parametric digital model from CLO3D,

---

while a simple and clean process, is only an approximation of the body and is limited in its representation.

Using photogrammetry, I am able to capture the nuances and details of the physical body in a digital format, a beneficial feature for studying its unique features and providing a more accurate representation of the human form. Nevertheless, as a nascent user of photogrammetry, the digital models I created were not very precise. Further investment and development in this technology could prove useful in better representing the human body, its proportions, and its unique features. By experimenting with this process, I hope to include photogrammetry and similar processes to better understand the body as a sight and as a base for apparel design.

Additionally, while the resulting digital models are of a clothed body, the use of this technology in apparel design would benefit from capturing a more baseline human figure. This would provide a better understanding of the body so that one can capture and overlay additional information, such as texture maps, allowing designers to accurately and precisely investigate human subjects for tailored design solutions in a way that more traditional methods cannot.

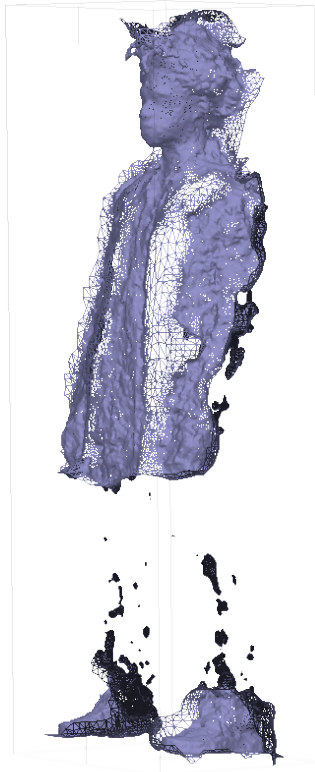


Fig 46. standingPoseWireframe1

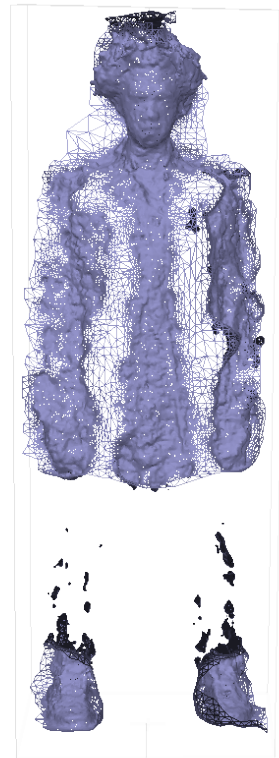


Fig 47. standingPoseWireframe2

## Methods

# MEGA

---

### Apparel Contextualization

The Mega level addresses objects—things with form. In this research, this level is the highest level of operating and the hardest to predict. As with any of the previous scales of magnitude, this level adopts the complexity from former levels, adding additional degrees of intricacy. Designing and planning at the Mega level demands some sense of control of the material at its more granular scales. As the fashioning of the material becomes more complex, its behavior becomes more challenging to predict, especially as a deformable material. Additionally, the form itself alters the way the material behaves, underscoring the importance of understanding the interdependent relationship between form and behavior. Digital tools and simulations can aid in the design process at this level, but physical testing and refinement are still necessary. Therefore, prototyping and testing are crucial at this level to ensure that the form of the object aligns with its intended purpose.

Numerous studies in computational knitting research are being conducted at the Mega scale. There is considerable interest amongst scholars in this domain to devise methods that enable users to fabricate a knitted object from a 3D digital model. I find that interest comes from computationally-driven architects and computer graphics programmers. Both demographics' interests converge in their desire to bridge digital space to physical space, where both architects and computer graphics programmers are working to develop a more advanced understanding of the design and fabrication of knitted materials. However, architecturally-focused papers concentrate more on material and form, aiming to develop complex knit structures and complex formal geometries. On the other hand, computer graphics papers seem to be more focused on technical comprehension of digital geometry, manipulating geometry with the understanding of mathematical theorems and algorithmic processing. Computer graphics driven projects often aim to develop bespoke software, whereas architects place a greater emphasis on constructing workflows that can seamlessly integrate with their existing suite of design tools. Given the range of developing research within this domain, it's worth mentioning that there isn't a single best solution to this problem, rendering it to be a challenging task nonetheless. However, the growing trend towards interdisciplinary research is driving the creation of innovative design tools.

Working at the Mega level presents the challenge of creating form from a digital mesh. To address this challenge, I sought to integrate a pre-existing tool into my developing pipeline. The use of this tool would enable me to effectively create complex forms more precisely and efficiently. While my work aligns more closely with the architecturally leaning research, the tools that these researchers developed were often less accessible. Architects were often more protective over the distribution of their tools, limiting their availability to a wider audience. As a result, I relied on tools that were developed on the more computer graphics side of the research space, as these tools were either publicly available or attainable by directly contacting the researchers. This is likely due to the fact that these tools were developed with the intention of being used and tested by a wide range of users. While these tools were developed with slightly different goals in mind, they still proved to have potential to be integrated into my work. A successful build of any of these programs would enable me to create more complex forms from digital meshes.

I explored five tools that would potentially enable me to create a machine-interpretable stitch mesh from a quadrilateral or triangular mesh. Testing these tools, I failed to integrate any one of them into the developing pipelines. Additionally, of the four tools that are designed to generate a stitch mesh, I only managed to get one tool to consistently output a stitch mesh. Hence, the measure of success shifted from evaluating each program's functionality to verifying if I could build each program to run on my machine. The programs were primarily developed using C++, which required me to download dependencies, potentially employ legacy versions of dependencies, and build files with limited knowledge of C++. Through the process of testing these tools, I encountered a number of challenges, yet the experience provided valuable insights into the research space and allowed me to think critically about my own work. Notably, I observed a cultural difference in sharing tools between the computer graphics and architectural subfields. Computer graphics practitioners appeared to be more open to sharing tools, which may be attributed to a greater culture of sharing work within computer science fields. However, utilizing these tools requires that one is able to build experimental software, a task that proved to be an additional challenge. Nevertheless, due to the unique approaches of each tool, I was required to consider how each one could be integrated into my pipeline, prompting me to critically evaluate my research and identify additional opportunities for improvement.

Out of the five tools, Autoknit, developed by the CMU Textiles Lab, was the only tool that provided me success in using the software and fabricating samples using the tool. As a result, Autoknit became the primary tool that I relied on the most at the Mega scale. The successful build of Autoknit could be further attributed to my close relationship to its creators and their assistance. While the other tools I tested had potential, the challenges associated with building the programs made them difficult to incorporate into my workflow. However, Autoknit is a vertically integrated software that is fully automated. Despite its potential value, I encountered difficulty in integrating it into my pipeline. While investigating new tools presented challenges, exploring these tools were beneficial to understand potential design possibilities.

## 3D Knitting Tools

### ► Knitting 4D Garments

In this paper, a new computational pipeline for designing and fabricating 4D garments as knitwear is introduced, aiming to provide comfort during body movement. This is achieved by carefully controlling

the elasticity distribution to reduce uncomfortable pressure and unwanted sliding caused by body motion. The team used a single-jersey jacquard (SJJ) knitting technique, which allowed them to knit patterns with different elastic levels using two yarns. The elasticity distribution of a garment is designed using physics-based computation, and the optimized elasticity is converted into instructions for a digital knitting machine using two algorithms proposed in the paper. One is a graph-based algorithm, which generates knittable stitch meshes that accurately capture the 3D shape of a garment. The other is a tiling algorithm that assigns SJJ patterns on the stitch mesh to realize the designed elasticity distribution. The

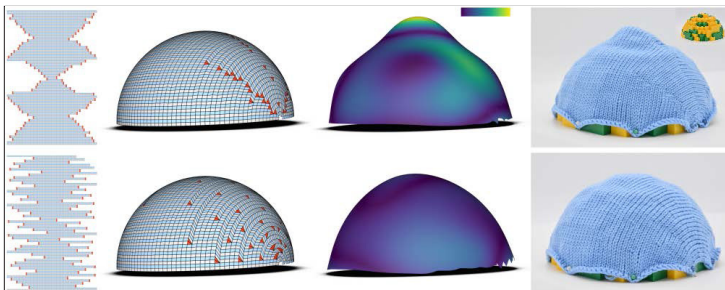


Fig 48. Knitted hemisphere. (Left-to-right) Knitting map, stitch mesh, 3D shape of knitting result simulated by FEA, and the physical result of knitting<sup>[38]</sup>

effectiveness of this approach is verified through simulation results and physical specimens fabricated by knitting machines. <sup>[39]</sup>

I reached out to the team behind this project through remote communication and was granted access to the “Knitting Short Rows” module, which creates the stitch mesh. Although I was able to build the project and get stitch meshes from example files, I encountered difficulties moving beyond this point due to the lack of available documentation. Furthermore, I wasn’t able to test the module that creates stitches, which would have been more relevant to my project’s goal of using knitted textiles as a performative material.

38. Liu, Zishun, Xingjian Han, Yuchen Zhang, Xiangjia Chen, Yu-Kun Lai, Eugeni L Doubrovski, Emily Whiting, and Charlie C. L. Wang. “Knitting 4D Garments with Elasticity Controlled for Body Motion.” *ACM Transactions on Graphics* 40, no. 4. 62, 2021, 628.

39. Liu et al. “4D” 2021, 628.

► **Stitch Meshing:**

In this paper, the team created a fully automatic pipeline that converts arbitrary 3D shapes into knit models. Based on a global parametrization remeshing pipeline, their method produces an isotropic quad-dominant mesh aligned with a 2-RoS field (two radial orthogonal systems). The knitting directions are determined using custom topological operations and a two-step global optimization that minimizes irregularities. The resulting mesh is then converted into a valid stitch mesh representing the knit model. To generate yarn curves from the stitch mesh, this team employed a yarn-level relaxation process, creating topologically valid models suitable for yarn-level simulation. The algorithm ultimately generates knit models from complex 3D shapes and processes numerous models without any user



Fig 49. Labeling input meshes into knitting rows<sup>[40]</sup>

input or parameter tuning. Unfortunately, I could not build the software, and after rereading the paper, I realized that the fabricated knitted structures are actually 3D printed and not machine knitted. Due to the paper's technical complexity, the fact that the stitch meshes haven't been machine knitted, and my own lack of knowledge, I found this project to be a costly investment of time. However, the work's quality is impressive, and I would have been eager to experiment with the generated stitch meshes.<sup>[41]</sup>

► **Computational Design of Knit Templates**

This paper presents an interactive design system for knitting, which enables users to create template patterns that can be fabricated using an industrial knitting machine. The design tool provides direct control

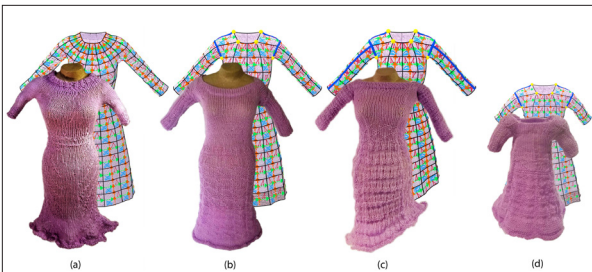


Fig 50. A design system exploring design axes for knitting to generate highly customized patterns from input shape templates<sup>[43]</sup>

of key knitting design axes consistently across variations of an input parametric template geometry. Users build a coarse quadrilateral mesh that adheres to knit design guidelines using an interactive meshing tool. Low-level machine knitting constraints are formalized at the level of the coarse quad mesh, ensuring machine-knitability, also providing automatic design assistance through auto-completion and suggestions. The system's capabilities are demonstrated through a set of fabricated examples, illustrating its effectiveness in creating various objects and interactively exploring the space of design variations.<sup>[42]</sup>

Again, I was unable to build this project successfully. However, the project's point of contact was eager for me to test the interface. I found this project particularly interesting, as

it addressed not only whole garment knitting but also embedding different stitch types in the textile at a granular level. This would help resolve the global issue of creating a plan for fabricating a 3D model

40. Yuksel, Cem, Jonathan M Kaldor, Doug L James, and Steve Marschner. "Stitch Meshes for Modeling Knitted Clothing with Yarn-Level Detail." *ACM Transactions on Graphics* 31, no. 3, 2012, 5.

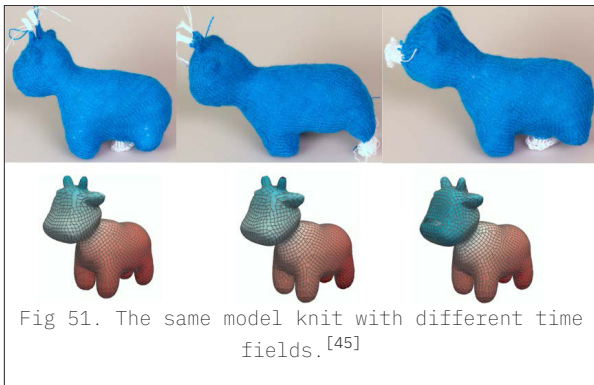
41. Yuksel et al. "Stitch" 2021, 5.

42. Jones, Benjamin, Yuxuan Mei, Haisen Zhao, Taylor Gotfrid, Jennifer Mankoff, and Adriana Schulz. "Computational Design of Knit Templates." *ACM Transactions on Graphics* 41, no. 2. 16, 2021, 1.

while allowing me to experiment with a system that addresses the complexity of introducing various stitch types. Even if I couldn't integrate this tool into my developing workflow, I haven't yet explored how different stitch types affect the form itself. It would have been fascinating to use a tool that could grant me that understanding.

#### ► Autoknit

This paper introduces the first computational approach capable of transforming 3D meshes, created by traditional modeling programs, directly into instructions for computer-controlled knitting machines.



Given user-defined starting and ending points on an input mesh, the system incrementally builds a helix-free, quad-dominant mesh with uniform edge lengths. A tracing procedure is then run over this mesh to generate a knitting path, and the knitting instructions for this path are scheduled in a way that is compatible with machine constraints. The approach is demonstrated on a wide range of 3D meshes.<sup>[44]</sup>

Autoknit is a prominent tool in this field of research and was developed by the talented team at the CMU Textile Lab, making it easier for me to build and understand the system. The tool involves four main steps in creating knit instructions: Constraints, Peeling/Linking, Scheduling, and Knitout. It is a fully automatic system that is well-referenced

in research, creating a network of knowledgeable people who have examples using the tool. With the right dependencies, one can create a "DAT" file compatible with KnitPaint (tier 1) in the terminal. I also



Fig 52. Digital toroid

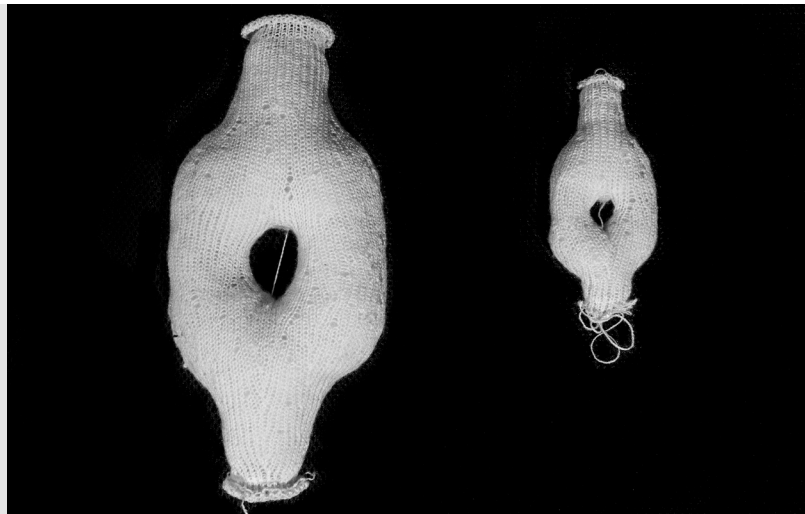


Fig 53. Knitted tests of toroid with different scale values

43. Jones et al. "Templates" 2021, 1.

44. Narayanan, Vidya, Lea Albaugh, Jessica Hodgins, and James McCann. "Automatic Machine Knitting of 3D Meshes." *ACM Transactions on Graphics*, 1, no. 1. 1, 2018, 12.

45. Narayanan et al. "Automatic" 2018, 1.

used Max Eschenbach's Grasshopper plugin, GHAutoknit, which provides an intuitive visual interface to save information in a Grasshopper file and easily modify the input parameters needed for the Autoknit system.<sup>[46]</sup> GHAutoknit is an interface that communicates with Autoknit, making it more user-friendly. I tested the example geometry of GHAutoknit, which is a mesh of a toroid, and successfully knitted two 3D geometries—a small and a large toroid—in a fully automated process.

While Autoknit is the most successful and accessible tool I tested, I still encountered some issues integrating it into my developing pipeline. The main problem is the lack of control when using the system, as it is a separate ecosystem with its own self-contained hierarchy. Autoknit automatically knits everything using jersey knitting at  $\frac{1}{2}$  gauge, where every other needle is knitted on. Additionally, one input value needed for the system is scale, but I wasn't able to predict the size of the knitted artifact. It can take time to determine if the peeling and linking stage is successful, which doesn't guarantee that the job scheduling will be successful. Autoknit requires users to relinquish a significant amount of control, but it remains a popular research tool that yields results.

#### ► Visual Knitting Machine Programming

This paper presents the first general visual programming interface for creating 3D objects with complex surface finishes on industrial knitting machines. Designing patterns for these machines usually requires

extensive training, which this interface seeks to address. At its core, the interface utilizes an augmented version of the stitch mesh data structure, which stores low-level knitting operations per face and encodes dependencies between faces using directed edge labels. The system can generate knittable augmented stitch meshes from 3D models and allows users to edit these meshes in a way that preserves their knittability. The execution order and location of each face can be scheduled for production on a knitting machine.<sup>[47]</sup>

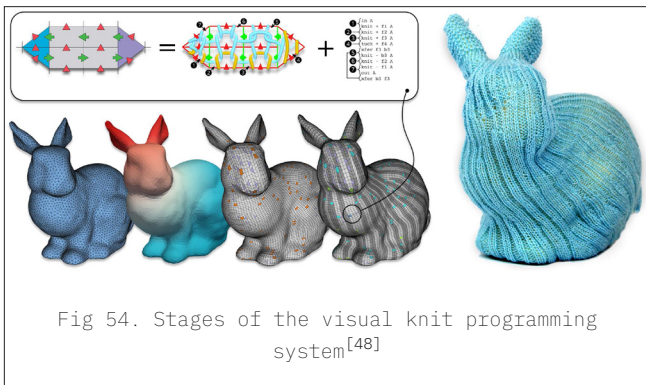


Fig 54. Stages of the visual knit programming system<sup>[48]</sup>

Visual Knitting Machine Programming (Visual Knit) was also developed by members of the CMU Textile Lab. Visual Knit is a system that considers how to embed different stitch types when machine knitting a form. However, due to some proprietary issues, I was only able to gain access to its scheduling module. Because Visual Knit's underlying architecture is compatible with Autoknit, I could combine Visual Knit's scheduler with the rest of the Autoknit ecosystem. This would allow me to plan and schedule a form with various stitch types. Ultimately, I chose not to continue investing time in this route because the Autoknit and Visual Knit ecosystems proved to be very dense. It would have required more time for me to comprehend how to properly integrate with these systems and develop a deeper understanding of C++ . Instead, I focused on other tools and approaches that could more easily fit within my developing pipeline and provide the level of control and customization I desired.

While I encountered challenges in building and integrating some of the projects and tools I tested, my

46. Eschenbach, Max. "GHAutoknit." Computer software. A Prototypical Bridge between Grasshopper and Autoknit. , October 2021. <https://github.com/fstwn/ghautoknit>.

47. Narayanan, Vidya, Kui Wu, Cem Yuksel, and James McCann. "Visual Knitting Machine Programming." *ACM Transactions on Graphics* 38, no. 4. 63, 2019, 1.

48. Narayanan et al. "Visual" 2019, 1.

exploration of the various computational knitting systems provided valuable insights into the possibilities and limitations of this field. The perfect tool or system may not exist, but by understanding the strengths and weaknesses of each, I can make more informed decisions about which tools to use and how to integrate them into my workflow. Ultimately, the key to success in computational knitting lies in finding the right balance between automation, control, and customization while staying open to experimentation.



Fig 55. Views of the avatar's compression body suit

---

## Garment Testing

The Mega scale of this research project also addresses the design of garments. However, because the primary focus of this work is centered on the knitting process, I placed less emphasis on apparel design. Using the existing digital model of my body, I created simple garments that closely contour the digital model's form. My intention was to employ each of the previously mentioned research tools to generate stitch meshes corresponding to the same full-scale garment. These stitch meshes would be translated into machine-readable instructions, fabricated, and worn to evaluate the differences in each tool's approach and the fabricated output. Despite my inability to fully conduct this experiment, the challenges encountered provided valuable insights into the existing gaps within this research field, as well as my own comprehension of the subject matter. This experience underscores the importance of addressing these gaps and refining the research methodologies to further advance the integration of computational design techniques in garment creation. As the field continues to evolve, a deeper understanding of these complexities will facilitate the development of innovative solutions that bridge the gap between digital design and physical fabrication, ultimately resulting in more sophisticated and functional garments.

In my pursuit, I decided to create some very simple garments that were simply subsections of my digital avatar. These apparel were akin to compression garments, designed to conform to the shape of the body. I labeled these garments as *avatarBodySuit* and *avatarShorts*, respectively. The intention behind these experiments was to gauge the efficacy of Autoknit when confronted with human-scale objects. My objective was to determine if Autoknit would be capable of generating a human-sized garment, and if so, to what degree of precision would the shaping of the garment be at this scale.

As previously discussed, I encountered difficulty in comprehending the relationship between the scale input parameter and the final knitted form created through Autoknit. To address this issue, I attempted to use a larger input mesh. It's important to acknowledge that most examples of artifacts created through Autoknit are typically small in scale, such as socks or sweaters for teddy bears. Consequently, testing at this scale meant that I would be pushing the boundaries of this system. Moreover, in experimenting with Autoknit, I did discover some computational bottlenecks.

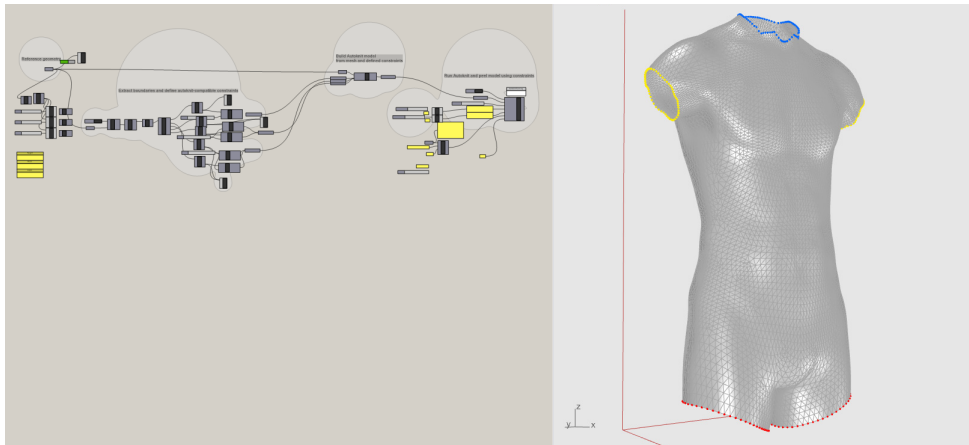
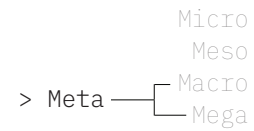


Fig 56. An attempt at knitting the digital body suit using GhAutoknit

I encountered a significant time hurdle when waiting for Autoknit to complete the peeling and linking stage. Even with a fairly robust gaming laptop, I had to wait 12 hours for Autoknit to finish this stage. This long wait also didn't guarantee a successful stitch mesh. Consequently, I faced an additional challenge of being unaware of the job status until the software processed the entire geometry. This remains true if the system fails to create the proper linkages in a previous part of the geometry. As a result, most of the time, this stage of the process was unsuccessful, significantly impeding progress in my experimentation. Nevertheless, I managed to obtain a successful enough stitch mesh to put it through the scheduling stage. This process, unfortunately, had the same issues of time expense and lack of transparency.

Though it was exciting to test Autoknit's capacity to process human-scaled geometry, the system proved to be too slow due to its computational bottlenecks. In light of this experience, it became clear that finding a balance between the capabilities of the software and the desired level of customization and control is essential for success in computational knitting. While Autoknit might not be the best solution for every situation, it still offers valuable insights and serves as a stepping stone for further exploration and development in the field.



## Methods

# META

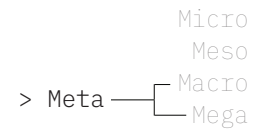
---

## Uncovering The Textile Template

After encountering roadblocks in my workflow, I took a step back to reassess the objectives I aimed to achieve in developing this pipeline. My initial goal was to create, connect, and augment tools that would enable designers to intuitively design and engineer entire garments with complex knitted structures. However, I realized that knitting intricate structures in conjunction with whole garment knitting is an immensely complicated task on a knitting machine with only two needle beds. A four-bed knitting machine—a machine I did not have access to—would be better suited to handle such complexity with greater ease. Moreover, controlling and integrating the only external tool I was able to successfully use proved to also be challenging. Integrating my pipeline with the Autoknit ecosystem required a significant investment of time and knowledge, making it an expensive system to modify. After reconsidering which of my goals warranted revision, it became apparent that the objective of whole garment knitting deserved reevaluation. It created a computational bottleneck. Whole garment knitting is not completely necessary to minimize material waste, reduce labor in the garment-making process, and create complex materials. By revising this goal, I aimed to strike a balance between computational efficiency and the pursuit of innovation in garment design.

The challenges I faced during my research led me to discover an additional “in-between” scale that I named the Meta scale. In examining the PhD work of Rickard Lindqvist, I came to understand that it’s possible to create a garment from a single, strategically cut piece of fabric. Although I did not develop proficient competency in applying Lindqvist’s theory, its principles served as the foundation that aided me to pivot the development of the pipeline. By reducing the complexity of scheduling a whole garment with intricate knitted structures, I would be able to rely on Lindqvist’s work and strategically plan the placement of seams of the garment. This would allow me to knit geometry like a flat textile. However, because the textile would be fabricated through industrial knitting, I would still retain the ability to shape it. Instead of crafting a 3D object, I would knit a 3D surface—a surface with dimension. This knitted 3D surface would then be sewn together to complete the garment’s fabrication. Irrespective of whether this surface is 2D or 3D, I defined a knitted surface of this kind to be a “textile template”. Whereas, although it has not yet been fashioned into a 3D object, the textile template is intended to be fabricated into a specific object. This

---



differs from a typical textile, which is generally intended to be cut, sewn, and used for a broader range of purposes. While I did not fully apply Lindqvist's principles, his work served as a catalyst for me to pivot my approach, enabling me to explore new possibilities in the realm of garment construction.

In order to knit a 3D surface, however, it was still essential for me to find an adequate method to create a stitch mesh and integrate it within the existing pipeline. Further research led me to discover a paper published by Keenan Krane, a computer graphics professor at CMU, which outlined a novel approach to creating equidistant contours on a surface using the heat equation—a partial differential equation that simulates the physical dispersal of heat on a surface. Generating equidistant contours on a complex 3D surface, while taking its topology into account, appeared to be the primary challenge of generating a stitch mesh. These contours could serve as knit course rows—the weft lines on a knitted textile. Utilizing these weft lines, I would then be able to calculate the corresponding warp lines, or wales. Daniel Piker, the creator of the Kangaroo plugin for Grasshopper, implemented this principle using C# within the Grasshopper environment and published the Grasshopper script. Building upon Piker's script, I began adapting and expanding its functionality so that I would be able to create a stitch mesh. While this aspect of my research remains a work in progress, I am working to integrate these methods and tools into my developing pipeline.

## Custom Stitch Meshing

### ► Tier 6.

In my pivot, I came across two pieces of literature that informed my new direction. In addition to Daniel Piker's technical implementation of Keenan Crane's "The Heat Method for Distance Computation", I found inspiration in Rickard Lindqvist's "Kinetic Garment Construction" and Mariana Popescu's computational knitting chapter in her PhD thesis, "KnitCrete".<sup>[49][50][51]</sup>

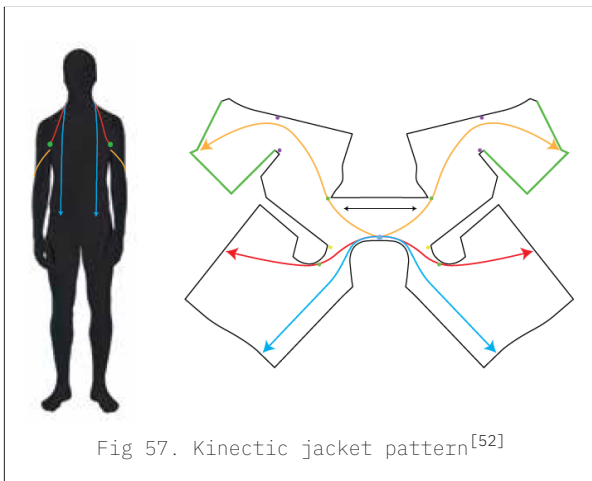


Fig 57. Kinectic jacket pattern<sup>[52]</sup>

Lindqvist's work didn't directly influence the technical aspects of my pivot. However, it helped me understand that there are alternative ways of constructing and tailoring garments, beyond the prescribed methods of the 18th century. With this knowledge, I felt confident that I didn't have to use sewing techniques that adhere to traditional tactics. Lindqvist's work offered a new model for understanding the body as an organic form and provided a fresh approach to creating garment patterns. This realization led me to believe that my workflow could incorporate sewing and construction processes while still breaking away from traditional garment-making tactics. In the continuation of this research, I aim to further explore how industrial knitting and 3D surface generation can work with this method of garment construction. For now, I understand that I can hypothetically fashion 3D surfaces into garments using Lindqvist's methods.

Popescu's "KnitCrete" is an essential reference for anyone attempting to think like an architect and use computation to create complex knitted structures. Her work provided me with a solid technical foundation for altering and augmenting Daniel Piker's Grasshopper script to serve computational knitting purposes. In chapter 6 of her thesis, Popescu gives an overview of how she turned a 3D surface into knitting instructions. While her tools and explicit methods remain unpublished, her thesis details her methods

49. Crane, Keenan, Clarisse Weischedel, and Max Wardetzky. "The Heat Method for Distance Computation." *ACM Transactions on Graphics*, 2017.

50. Lindqvist, Rickard. "Kinectic Garment Construction," 2015. 221.

51. Popescu, Mariana. "KnitCrete," 2012.

52. Rickard "Kinectic" 2015, 221.

to a degree that allows others to follow and reimplement her workflow with their interpretation. This proved invaluable to my work.

Although I tried to recreate Popescu's workflow, even attempting to recreate her example base geometry (with some changes), I didn't follow her thesis too closely due to our different precedents, tools, and knowledge bases. However, I was greatly inspired by it. This experience demonstrated how even when attempting to recreate another researcher's experiment, the slight degree of separation between us can yield different results.

When working with digital data of any form, including textiles, it's crucial that the data is clean and reliable. Using the correct geometry is the foundation of digitally representing a knitted textile. This issue lies at the heart of the problem. At its core, this problem can be plainly explained as a remeshing or tiling issue. The process of creating a stitch mesh involves reprocessing a base mesh to eventually yield a tiling pattern that makes sense for the knitting process. However, this doesn't make the process an easy problem to address. In its most basic form, creating a stitch mesh is akin to attempting to plan a scheme for laying bricks for a surface of any shape. Of course, the additional physical constraints of gravity and compression aren't relevant in this analogy. All in all, attempting to create a complex shape from modules of a fixed size can be a challenging problem to address.

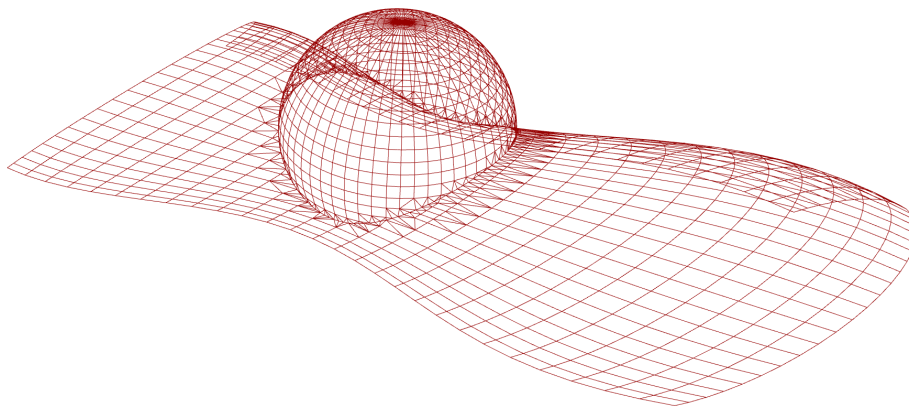
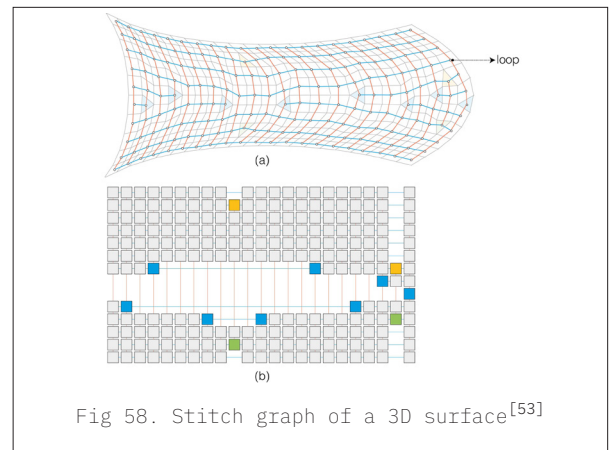


Fig 59. Step 1: uncleanMesh

The first step in creating a stitch mesh using the heat method is to ensure that the input mesh is a triangular mesh, derived from triangular faces. To accomplish this, I first created a surface that combined a hemi-sphere (positive Gaussian curvature) and a curved surface (zero Gaussian curvature), resulting in

53. Popescu "KnitCrete" 2012, 104.

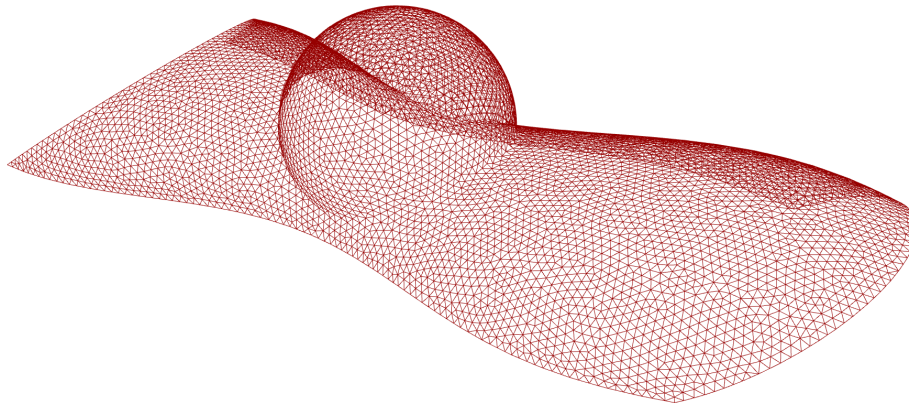


Fig 60. Step 2: triangulatedMesh

an irregular, messy mesh. Using Grasshopper, I triangulated the `uncleanMesh` to create `triangulatedMesh`. With Grasshopper, I found that I could triangulate virtually any 3D surface and use it as an input mesh. Daniel Piker's implementation of "The Heat Method for Distance Computation", spearheaded by Keenan Crane, was created in Grasshopper. This C# driven Grasshopper script simulates how heat may disperse on a surface. The simulation is visually represented through a texture map, with each color separation representing an increment of time. Within seconds, this simulation visually resolves how heat can transmit through a surface. The result is a textured mesh, holding information on how to evenly contour the mesh. However, since texture information is not geometric, additional work is required to translate this information.

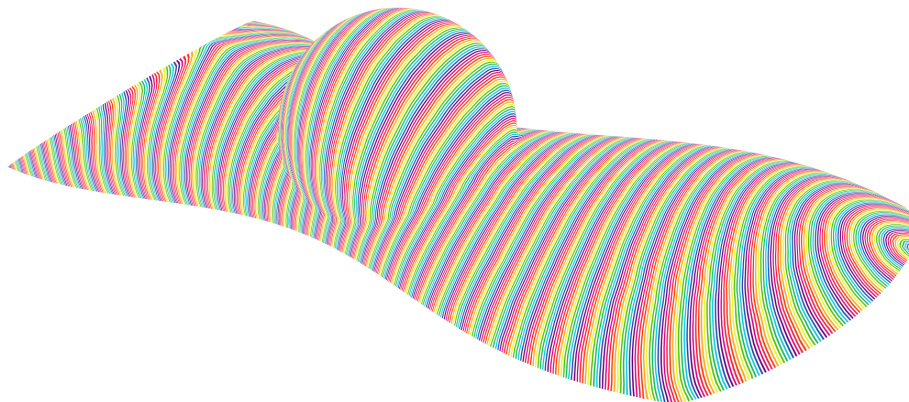


Fig 61. Step 3: Simulation

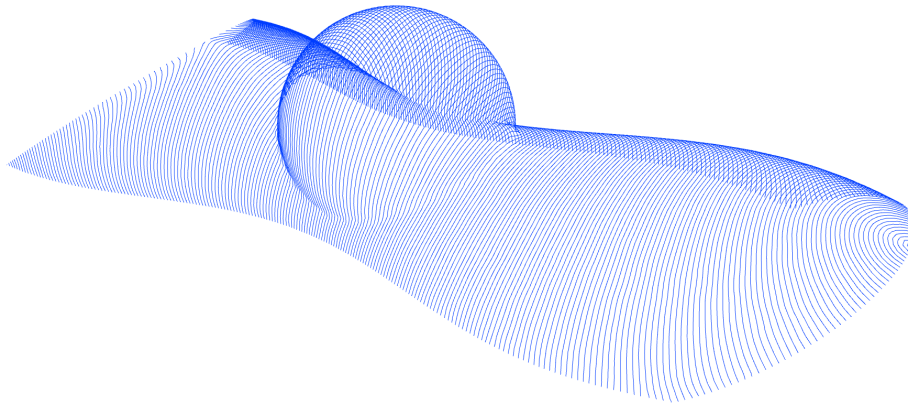


Fig 62. Step 4: contouredMesh

Daniel Piker, in this script, has an additional C# component that converts this graphical information to curve geometry. With this contour component, the user can change the number of curves that contour the surface. These contours are guided by the shape of the texture information and represent the course rows of a stitch mesh. However, for my purposes, I needed the contours to be based on their distance apart, not their quantity. I altered this part of the script so that the contours are derived from distance, not count. This change allowed me to control the height of each stitch. Nevertheless, there was still error, particularly in areas of high curvature. Whereas the simulation struggled to resolve the geometry to have equally spaced intervals. Despite these issues, this simulation proved to be the most usable and quickest method I've found so far.

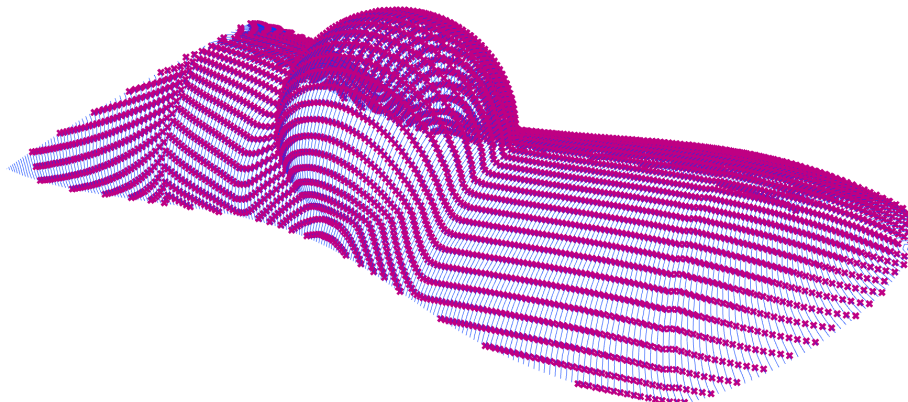


Fig 63. Step 5: dividedMesh

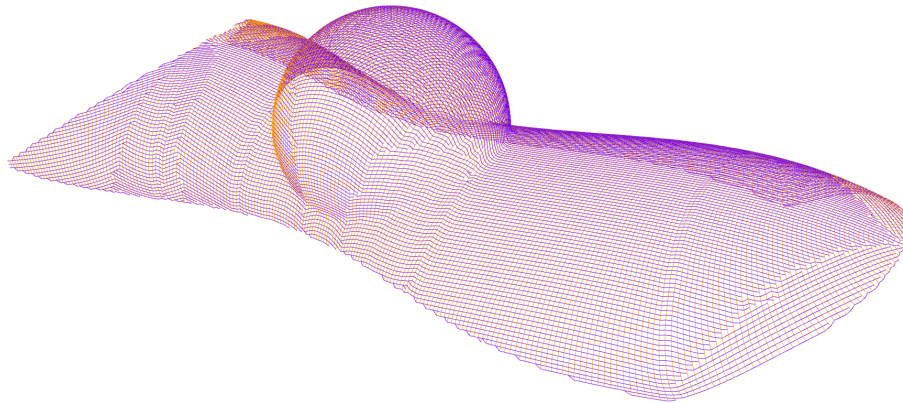


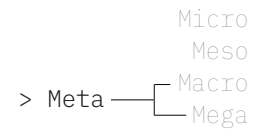
Fig 64. Step 6: stitchPaths

Having already established the height dimension of a single knit stitch, the next step was to create the width dimension. To do this, I divided each contour by a distance determined by the user's input in a custom C# Grasshopper component I created. This was the part of the global Grasshopper script that I began building myself, thus augmenting Piker's work. The component that divides the contour also allowed me to control where the contour was divided. I could create different types of symmetries, which ultimately altered how the 3D surface was constructed. This provided some flexibility in fabrication, in case I needed to modify the paths in which stitches were formed.

In addition to creating courses, I needed to create wales. This more complex process involved linking the course rows together. Inspired by Popescu's work, I developed an additional custom C# component to



Fig 65. Step 7: incompleteStitchMesh



create linkages between the course rows to form wales. I was motivated by the conditions she used in her own work. Initially, I connected each start and end point of each stitch to the stitch in the next row with the shortest distance. I plan to add a second condition to connect stitches with the least amount of angle change. Furthermore, in the time I had, I was only able to create linkages for plain stitches and decreasing stitches, stitches that denote the surface getting smaller at that locality. I have not yet been able to create increase stitches.

Creating a fully fashioned mesh from these linkages is perhaps the most complex part of the process. With the correct base geometry as input, the answer is there. However, it requires some algorithmic know-how to produce a fully fashioned mesh from the previously mentioned linkages. With the time I had for this research, I only sought to create mesh faces from the decreases in the geometry. With more time and investment, it would be achievable to create a full stitch mesh. This would allow me to consider this entire method as tier 6 of the workflow, which would complete this workflow for the aims of this research.

## Research Review

# RESULTS

Through this research, I've come to understand that digitizing the textile-making process can be rather complex to fully realize. Firstly, this problem is a niche issue. Research in this realm easily touches disciplines that seek to understand material assembly, textile fabrication, computer science, and graphical representation. Using experimental tools, creating one's own tools, and finding areas for exploration takes understanding of these knowledge bases. It requires that one understands their angle of approach and their gaps in knowledge.

Secondly, it's worth noting that the textile industry and some research spaces that address this issue are less reluctant to share their tools and knowledge. This creates a diversity in approaches to digital fabricating textiles. This forces researchers to create their own approaches as there are not commonly available tools for sole designers. Researchers coming from architecture backgrounds tend to be more focused on developing digital fabrication workflows for the sake of design and material exploration of textiles. Therefore, tool-making doesn't seem to be their primary goal but a product of the lack of access. I find that tools developed by architects would democratize access to these digital fabrication workflows because ease of use for the sake of collaboration and teaching is often a baked-in goal of theirs. However, it proved more difficult to access their tools, as they deem them proprietary. Furthermore, if the research was also funded by a larger corporation, digital fabrication tools also proved to be inaccessible. This was true with the Visual Knit project.

Additionally, the textile industry, specifically the industrial knitting industry, is very conservative with their software. There is no common software used in industrial knitting, as the software one uses is linked to the brand of machine one is using. Industrial knitting machines and their linked software are priced in the tens of thousands, creating a steep barrier of entry for research and nascent knit programmers without institutional or industrial backing. With disparate knowledge bases and a diverse range of experimental tools, the challenge is to find a way to link available tools and build a workflow unique to your needs. The pipeline I've created was developed with a "hacker" mentality, wherein I'm trying to understand how to get different systems to communicate with each other. Due to the less predictable nature of textiles, different

research aims, and a diversity of disciplines in this research field, there is no single solution. The pipeline I've created is the culmination of getting different moving parts to communicate and work together. The process of creating this pipeline was both challenging and rewarding. It required a deep understanding of various fields and a willingness to experiment with different tools and systems. Despite the obstacles, I was able to create a unique workflow that enabled me to effectively digitize the textile-making process, create textiles with complex knitted structures that are informed by geometric data, and planted the seeds to explore how biometric information can inform textile structure.

## Research Review

# CONTRIBUTIONS

Throughout my research, I have made several significant contributions to the computational knitting research field, exploring various scales and aspects of the knitting process. The research spans from the micro scale, where I've investigated the properties of yarn, to the macro and mega scales, where I've developed innovative design processes and techniques for creating textiles and garments. These findings yield a model for understanding of the potential challenges and opportunities when working to create a digital fabrication pipeline for knitted textiles.

At the micro scale, I have gained an understanding of yarn, highlighting the importance of controlling material properties even at the smallest of scales. This knowledge is crucial for developing digital design systems that accurately simulate and predict textile behaviors, encouraging designers to understand material behavior .

At the meso scale, I've developed an organizational scheme for classifying different programming languages and tools based on their level of encoding. By integrating these tools and establishing communication between them, I've created a comprehensive workflow for digitally processing knitting machine instructions. This workflow can accommodate various stitch types and design complexities, offering flexibility and adaptability in the design process.

At the macro scale, I've employed a Grasshopper script to create complex textiles with different stitch types and arrangements that would be difficult to achieve through traditional methods. I have also treated the human body as a site, developing an algorithm that assigns stitch types to specific localities on the body to create textiles and garments that address various performative needs.

At the mega scale, I have explored various computational knitting systems, from pipelines for designing and fabricating 4D garments to automatic methods for converting 3D shapes into machine-knitable instructions. Although I've encountered some limitations and difficulties in integrating these systems into my workflow, the exploration has provided valuable insights into the possibilities and constraints of

computational knitting.

Finally, at the meta scale, I have revisited garment construction and assembly, seeking a balance between computational efficiency and innovative design. Drawing inspiration from the works of Rickard Lindqvist and Mariana Popescu, I have developed a method for creating stitch meshes using the heat method for distance computation, laying the groundwork for a new approach to garment construction.

Overall, my research has made significant strides in advancing the computational knitting field, uncovering new methods and techniques while addressing challenges at various scales. By exploring the relationships between yarn properties, stitch patterns, design processes, and garment construction, I have provided valuable insights for future research and innovation in textile design and fabrication. My contributions not only help to democratize access to these workflows but also inspire further exploration and development in this interdisciplinary field.

## Research Review

# NEXT STEPS

The research presented in this thesis offers valuable insights into the potential of computational knitting at various scales. However, there are limitations in the current pipeline and opportunities for further development.

**Micro Scale:** One limitation at the micro scale is the difficulty in acquiring specialized yarns, such as fire-resistant Nomex yarn. Future work could involve collaborations with manufacturers to develop customized yarns for specific applications. Additionally, the handspun yarn created in this research was not used in further experimentation. Investigating the potential of handspun yarns in computational knitting could provide novel material properties and design possibilities.

**Meso Scale:** The current workflow does not account for stitch types that require racking and intricate transfers. Further research is needed to develop methods for digitally representing and accommodating more complex stitch types. Improving the interoperability of different programming tools and languages could streamline the design process and enhance the capabilities of the system.

**Macro Scale:** More testing and research are needed in the development of multi-modal maps and the optimization algorithm for assigning stitch types based on data points. Integrating real data, such as body temperature or abrasion probability, could further refine the design process and result in more performance-driven garments.

**Mega Scale:** The computational bottlenecks encountered in this research highlight the need for more efficient algorithms and software for generating stitch meshes. Future work could focus on optimizing the Autoknit process and exploring other computational knitting tools. Additionally, a better understanding of the relationship between the scale input parameter and the final knitted form could improve the accuracy and predictability of the design process.

**Meta Scale:** The pivot in research direction towards the Meta scale reveals the potential for incorporating

sewing and construction processes into computational knitting workflows. More time and investment are needed to develop a fully fashioned mesh from the linkages created at this scale. Exploring alternative methods for generating stitch meshes and optimizing the process of creating courses and wales in areas of high curvature could enhance the capabilities of the pipeline.

The limitations in the current pipeline present opportunities for further development and refinement. Enhancing the interoperability of programming tools and languages, optimizing computational knitting algorithms, and refining the stitch mesh generation process could result in a more streamlined and efficient pipeline. The ongoing research and development in computational knitting has the potential to revolutionize the design and fabrication of textiles and garments, offering new possibilities in both the textile industry and beyond.

## Bibliography

- Alfaris, Anas, and Riccardo Merello. "The Generative Multi-Performance Design System." Thesis, Acadia, 2008.
- *ancient knitting*. February 14, 2014. *The History of Knitting Pt 1: Mysterious Origins*. <https://sheepandstitch.com/the-history-of-knitting-part-1-mysterious-origins/>.
- Crane, Keenan, Clarisse Weischedel, and Max Wardetzky. "The Heat Method for Distance Computation." ACM Transactions on Graphics, 2017.
- Davis, Felecia, Farzaneh Oghazian, and Sam Moradzadeh. "Form-Finding of Architectural Knitted Tensioned Structures," 2022.
- Dumitrescu, Delia. "Relational Textiles: Surface Expressions in Space Design," 2013.
- Eschenbach, Max. "GHAutoknit." Computer software. *A Prototypical Bridge between Grasshopper and Autoknit*. , October 2021. <https://github.com/fstwn/ghautoknit>.
- *Flame resistant meta-aramidic yarn*. n.d. *NOMEX Yarns*. <https://www.micspa.com/en/yarns/260/nomex->.
- Gaffney, Patrick J., The NFPA 1975 standard on station/work uniforms: A necessity or a luxury? § (1990).
- Horteborn, Erica, and Malgorzata A Zboinska. "Exploring Expressive and Functional Capacities of Knitted Textiles Exposed to Wind Influence," 2021.

- Jones, Benjamin, Yuxuan Mei, Haisen Zhao, Taylor Gotfrid, Jennifer Mankoff, and Adriana Schulz. "Computational Design of Knit Templates." *ACM Transactions on Graphics* 41, no. 2. 16, 2021.
- Karmon, Ayelet, Yoav Stermann, Tom Shaked, and Eyal Sheffer. "KNITIT: A Computational Tool for Design, Simulation, and Fabrication of Multiple Structured Knits," 2018.
- Knight, Terry, and George Stiny. "Classical and Non-Classical Computation." *Architectural Research Quarterly* 5, no. 4, 2001.
- Lindqvist, Rickard. "Kinectic Garment Construction," 2015.
- Liu, Zishun, Xingjian Han, Yuchen Zhang, Xiangjia Chen, Yu-Kun Lai, Eugeni L Dubrovski, Emily Whiting, and Charlie C. L. Wang. "Knitting 4D Garments with Elasticity Controlled for Body Motion." *ACM Transactions on Graphics* 40, no. 4. 62, 2021.
- Majumdar, Abhijit, Anindya Ghosh, Prithwiraj Mal, and Debamalya Banerjee. "Multi-Objective Optimization of Air Permeability and Thermal Conductivity of Knitted Fabrics with Desired Ultraviolet Protection." *Journal of the Textile Institute*, 2017.
- Narayanan, Vidya, Kui Wu, Cem Yuksel, and James McCann. "Visual Knitting Machine Programming." *ACM Transactions on Graphics* 38, no. 4. 63, 2019.
- Narayanan, Vidya, Lea Albaugh, Jessica Hodgins, and James McCann. "Automatic Machine Knitting of 3D Meshes." *ACM Transactions on Graphics*, 1, no. 1. 1, 2018.
- Popescu, Mariana. "KnitCrete," 2012.
- *Spinning with a Suspended Spindle. Spindle (Textiles)*. Wikipedia, n.d. [https://en.wikipedia.org/wiki/Spindle\\_%28textiles%29](https://en.wikipedia.org/wiki/Spindle_%28textiles%29).
- Tamke, Martin, Paul Nicholas, and Mette Ramsgard Thomsen. "A New Material Practice," 2012.
- Yu, Catherine, Willa Yunqi Wang, and Jaiqi Liu. SoftNote, n.d. <https://catherineyu.com/SoftNote-2021-1>.
- Yu, Tianhong Catherine, and James McCann. "Coupling Programs and Visualization for Machine Knitting," 2020.

- Yuksel, Cem, Jonathan M Kaldor, Doug L James, and Steve Marschner. "Stitch Meshes for Modeling Knitted Clothing with Yarn-Level Detail." *ACM Transactions on Graphics* 31, no. 3, 2012.