



## A Boolean Method to Model Knit Geometries with Conditional Logic for Additive Manufacturing

James I. Novak<sup>1</sup> 

<sup>1</sup>Deakin University, Australia, [james.novak@deakin.edu.au](mailto:james.novak@deakin.edu.au)

**Abstract.** As additive manufacturing (3D printing) technology becomes increasingly mainstream, new tools are required to enhance creative endeavors in disciplines such as fashion design, which have been slow to embrace digital technologies and computer-aided design software. Additionally, new techniques are required based on an understanding of the limitations of common desktop extrusion-based 3D printers, with 2.5D printing presenting new opportunities to rapidly produce complex forms, such as fashion, which can be assembled by hand as a hybrid approach to digital manufacturing. This paper presents a parametric system through which a new type of digitally fabricated, hand-assembled knit can be customized using a constrained selection of interactive controls. Novice users may rapidly iterate a pattern of loops and floats at a scale matching their Fused Deposition Modeling (FDM) 3D printer, creating a series of knit courses which can be assembled into a textile of expandable dimensions. More advanced users may modify the geometry or logic of the system, building new forms of knits that could not be manufactured through traditional means. This paper will guide designers through the process of developing a new typology of textile appropriate for production on ubiquitous FDM machines, following a workflow from 2D to 2.5D to 3D. Examples were printed at various scales and contribute to discourse on customization, parametric design and visual programming languages for additive manufacturing.

**Keywords:** 2.5D printing, 3D printing, Algorithm, Fashion and textiles, Fused deposition modeling (FDM), Grasshopper, Rhinoceros 3D, Visual programming language.

**DOI:** <https://doi.org/10.14733/cadaps.2020.659-673>

### 1 INTRODUCTION

Additive manufacturing (AM), known popularly as 3D printing, provides new opportunities to imagine the way products are designed, manufactured and distributed [5, 33]. An active and highly visible area of research is fashion, with numerous studies exploring the ability to produce novel textiles directly through AM [17, 23], as a hybrid by 3D printing onto fabric substrates [13, 24], or as complete garments made up of numerous components printed as a fully assembled object [15, 28]. Despite advances being made by some leading designers, Computer-Aided Design (CAD) software has been slow to accommodate for the new capabilities offered by AM [9, 29], making it challenging

for designers to maximize their use of the technology. This is particularly evident in fashion design, an industry that has been slow to adopt three-dimensional (3D) CAD software suitable for designing parts for 3D printing [27], with the technical aspect of projects often led by designers with backgrounds in architecture or industrial design where advanced CAD software is a common part of practice [25]. Iconic 3D printed outcomes of collaboration include the *Escapism Dress* by Iris van Herpen (fashion designer) and Daniel Widrig (architect), the full-length gown for Dita Von Teese by Michael Schmidt (fashion designer) and Francis Bitonti (architect), and the *Dive Into Me* gown by Melinda Looi (fashion designer) and Samuel Canning (industrial designer). Such collaborative efforts require extensive communication and translation of ideas over time [15], highlighting the gap in appropriate software tools for fashion designers to translate concepts to 3D printable designs.

However, complex 3D forms requiring advanced AM technologies, such as Selective Laser Sintering (SLS) used in all the previously mentioned examples, are also being exhibited alongside garments produced on common desktop Fused Deposition Modeling (FDM) machines. Such machines typically have a build volume <300mm in each orientation, which alone cannot make an entire garment. However, through novel hybrid approaches to printing smaller panels and manually assembling these, designers such as Danit Peleg [14] and XYZ Workshop [7] have produced new forms of 3D printed garments. The technique they employ is often called 2.5D printing [20, 36], with individual parts representative of a 2D shape that has been extruded in a single orientation. According to Novak et al. [20] such 2.5D prints can be easily optimized in CAD software to 3D print much faster than more complex 3D forms, and fashion designers can retain hand-based skills in the way smaller panels are assembled together.

This paper applies such 2.5D printing to a common textile geometry; the knit. Traditionally, knit structures are formed by “the interlooping of a continuous thread to create symmetrical loops which create the structure. These loops can be extended in different directions giving knitted structures their inherent stretch and elasticity” [4]. The focus of this study is on customization of knitted textile structures using popular CAD software *Rhinoceros 3D* with the *Grasshopper* plug-in, allowing others to apply and develop the methods described due to the widespread availability of the software. Utilizing 3D printing, new geometries are possible that could not be manufactured using a traditional knitting machine. Furthermore, by building off similar research into hybrid textile fabrication processes [10, 22, 35], and applying it to common FDM technology, a broad range of designers and researchers will be able to implement the method described in this research compared with studies dependent on more expensive 3D printing hardware.

The practical component of the study developed a simple control interface allowing novice designers to customize a weft knit pattern, with parametric controls ensuring that all permutations possible connected correctly together. This included provision of inputs for users to enter basic settings from their specific FDM machine, automatically modifying the knit geometry to ensure printability, and acting as a contained system for a new typology of knit textile fabrication. The secondary aim was to establish a system that designers could expand and modify as their CAD skill increased, with discussion and examples indicating how new knit geometries can be developed beyond the constraints of traditional thread-based knitting.

## 2 RELATED WORK

The first computer system employed within the textile industry was from IBM in 1967, and personal computer systems for textile designers did not become available until the 1980’s [34]. Since then, development of CAD systems for the textile industry have flourished alongside computer systems controlling manufacturing processes. For the purposes of this study, literature is divided into two categories:

1. Those using CAD systems to advance traditional knitting with thread.
2. Those using CAD systems to develop new knit knowledge for additive manufacturing.

A parametric study by Oliveira et al. [22] is an example of research in the first category, with a key finding being that “parametric computer modeling can be a valuable tool for proposing new 3D

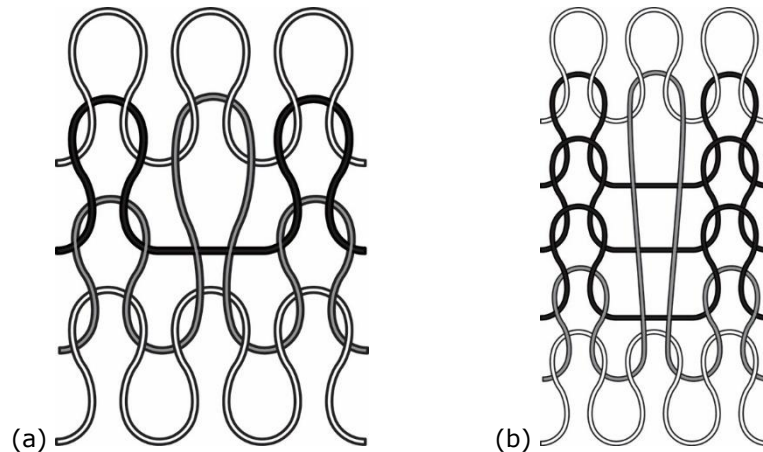
structures, since it can generate several possibilities which can then be selected according to the application needs.” Earlier knit research focused on accurate 3D modeling of knit structures using NURBS surfaces and mathematic formula, resulting in visually accurate knit models [11]. As CAD software and computer hardware improved, such accurate knit models employed simulation to examine strength and flexibility characteristics; for example Guo et al. [12] used ball b-spline curves and Hooke’s Law to simulate various weft knitted structures, while Renkens and Kyosev [26] simulated the mechanical process of knitting warp structures in order to result in an accurate 3D model, rather than directly modeling the end result. Such studies focused on singular materials, and simulation has been found to be more challenging with composite knit structures, with Araújo et al. [2] breaking a fiberglass textile into tetrahedral unit cells in order to obtain simulation data. The challenge for this first category of knit CAD research is that knit manufacturing technology remains relatively unchanged in recent decades, and there are physical limitations to how a continuous thread can be manipulated by a knitting machine. As a result, there are limited opportunities to design new knit textiles, with innovation being incremental, and the benefits for fashion designers who are involved in more creative applications of textiles are limited.

As additive manufacturing technology matures and offers new design freedoms, research in the second category provides designers with new opportunities to harness the flexibility and stretch of knit structures, and solve the issues of rigidity common to many previous exemplars of 3D printed fashion [3]. The earliest research exploring 3D printed knits appeared in 2012 and employed a code-based approach to the design of a “plain knit stitch” structure alongside other textile geometries [8]. This theoretically allowed for customization of the textile geometries by altering code parameters, however, was not demonstrated in the study which focused on qualitative assessments of the 3D printed outcomes. Melnikova et al. [16] used CAD software *Blender* to model a single assembled knit geometry, focusing on a comparison of material and performance qualities between desktop FDM technology and SLS, rather than any ability for designers to utilize CAD software to design knit geometries. Similarly, Beecroft designed and printed two assembled weft knits in 2016 [3], and most recently in 2019 expanded the same concept to tubular knits [4]. *Rhinoceros 3D* CAD software was used for both studies, which on its own provides minimal opportunities for parametric customization. Across the literature on 3D printed knits, CAD has been a means to generate forms to test physical qualities of 3D printed knits, rather than being explored as a tool to enhance the design capabilities of novice CAD users using parametric relationships like studies discussed in the previous section [10, 22, 35]. Therefore, new research is needed to investigate how a knit may be designed differently when there is no longer a single thread being looped by needles into a knitted form, and how it may be easily altered to suit different visual or functional characteristics desired by a user who may have limited CAD expertise.

### 3 KNIT TECHNICAL KNOWLEDGE

#### 3.1 Knit Definitions

As defined by BS EN ISO 4921:2002 standards [31], the basic building block of a knit is the *loop* (also known as a *kink*), with a row of connected loops known as a *course*. As each successive course is knitted together, the loops are interconnected, with columns of interconnected loops known as a *wale*. The craft of knitting features numerous techniques that can be implemented to generate countless visual and functional effects. To limit the scope of this study, customization of the knit was performed by the application of *floats* which occur when a loop in a series is skipped, missing the connection to the loop from the previous course, as shown in Figure 1a. The loop from the following course can then be connected over the float to the preceding loop, and is known as a *held stitch* [31]. Figure 1b shows the ability to build up successive floats, which may or may not be manufacturable depending on the capabilities of the knitting machine and the thread characteristics. However, these restrictions are removed when using 3D printing, and new limitations on the size of a held stitch are related to the size of the 3D printer.



**Figure 1:** Examples of float stitches in black, which may occur in a single course (a), or successively over multiple courses (b). The held stitch is shown in gray and stretches over the float stitches.

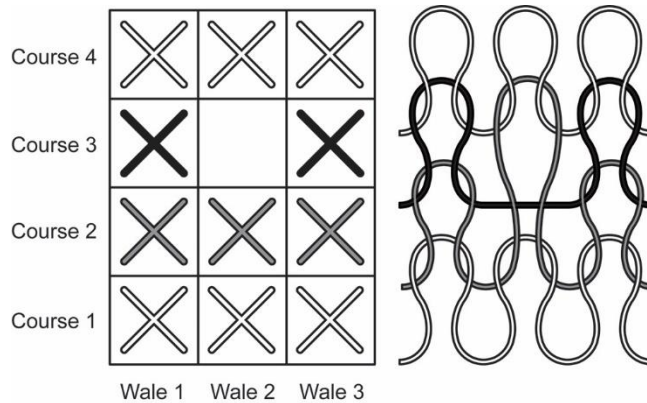
### 3.2 CAD software

The software used for this study was *Rhinoceros 3D V5* which is a popular 3D CAD tool for architects and designers [19]. A reason for its popularity is its open software development kit (SDK), making it modular and customizable, with a large variety of plug-ins enhancing the capabilities of the base software package. *Grasshopper* was the most popular plug-in for V5 of *Rhinoceros 3D* and is now integrated directly within *Rhinoceros 3D* from V6. It is a visual programming language that uses algorithms to control geometry directly within *Rhinoceros 3D*, as well as having the capacity to interact with external software and systems [18], and is the primary parametric tool for *Rhinoceros 3D*. As a result of its visual nature, complex systems that may otherwise require advanced text-based coding can be graphically formulated, with studies suggesting that it is more readily learned by designers due to this visual quality [6, 19]. This was a desirable quality for the study and allowed complex systems to be reduced to a small number of visual controls that promoted interaction.

For this study both *Rhinoceros 3D* and *Grasshopper* work symbiotically in the final system; *Rhinoceros 3D* contains the basic geometry options for the knit as two-dimensional (2D) linework, allowing advanced users to substitute the geometry for their own. Since the geometry begins 2D, users may develop their own system of loops, floats or other forms using any 2D CAD software, which may be more familiar to disciplines like fashion and graphic design. *Grasshopper* then links to the components of the 2D linework, using Boolean operations to control their appearance as a user commands, while also expanding the linework into 3D geometry suitable for 3D printing. Given these two separate yet interdependent systems, the following sections will detail the CAD system course-by-course rather than as separate processes to ensure clarity and the ability for others to replicate the system. Four courses were developed to suitably demonstrate the system but can be expanded by advanced users following the conditional logic that forms the foundation of the algorithm.

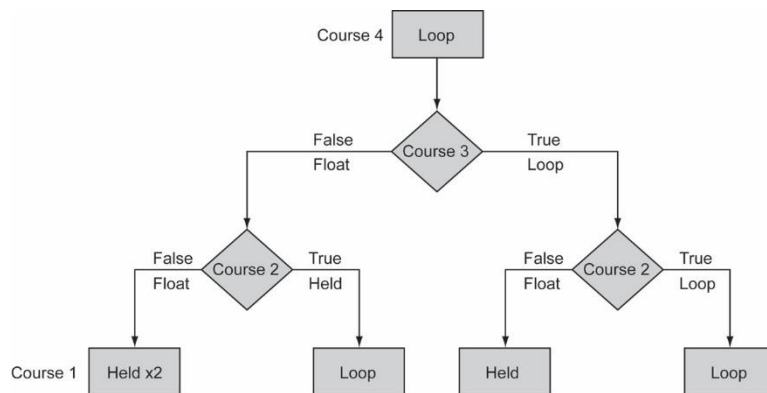
## 4 METHOD: 2D TO 2.5D TO 3D

Traditional knits are often portrayed diagrammatically to communicate the process of knitting specific patterns or programming machines; one such system uses a square grid as shown in Figure 2, which may be likened to Boolean commands when the knit options are reduced to binary loop (X) or float (empty) geometries. A loop appearing adjacent to a float in the following course is assumed to be a held stitch and is not given a different symbol. This reduction of more complex 3D geometry into a series of binary cells forms the foundation of the customizable knit algorithm.



**Figure 2:** Diagrammatic representation of loops (X) and float (empty), as well as course and wale directions, alongside visual representation.

Progressing through wale cells in a four-course system can thus be reduced to a series of Boolean commands, visually represented as a decision tree in Figure 3. Course 4 remains static in this system; this is to ensure that the algorithm remains self-contained and that multiple collections of courses will connect together into a larger textile (i.e. Course 4 of the next collection will connect with Course 1). It would be impossible for the algorithm to know, for example, how far to extend a held stitch in Course 3 over a float in Course 4 without knowing where the previous loop occurred above Course 4 to connect to. Similarly, Course 1 cannot contain floats and must close the collection so that a new collection can be connected below, whether it is identical or different. This will be discussed as a limitation later in the paper. The following sections will detail the method of translating the decision tree in Figure 3 to the Rhinoceros 3D and Grasshopper system.

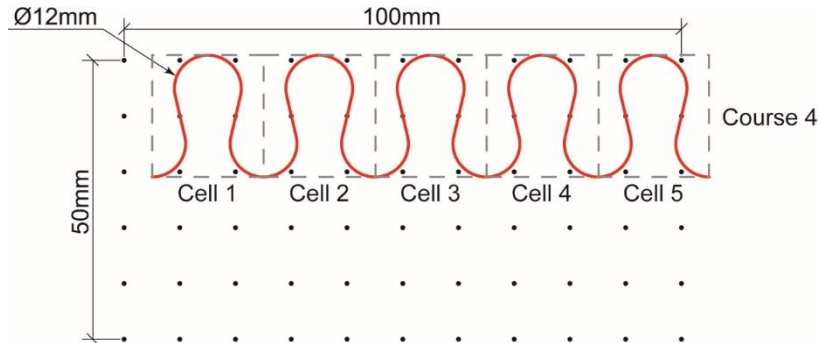


**Figure 3:** Decision tree highlighting Boolean logic for cells in a single wale using loops, floats and held stitches.

#### 4.1 Course 4

Within Rhinoceros 3D, a grid of 10x10mm cells was constructed as shown in Figure 4. The course length of 100mm was chosen to reflect the smallest common FDM 3D printer bed size. The basic loop geometry was constructed from Ø12mm circles and tangentially connected with straight lines between top and bottom circles to create a course of knit. A single loop is defined by the upper portion of loop geometry; therefore, there were five knit cells within a course spanning 100mm as defined in Figure 4. Because Course 4 remains static, this 2D geometry was directly imported into

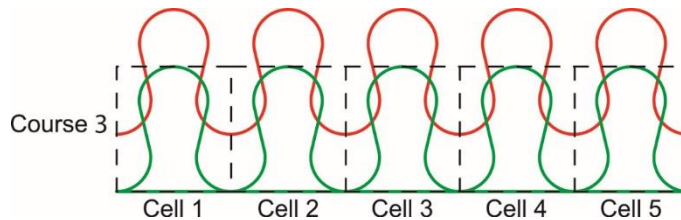
Grasshopper using the *Curve* function, to be later connected at the point the knit geometries are given three-dimensional character.



**Figure 4:** Construction geometry for a basic knit course in Rhinoceros 3D.

#### 4.2 Course 3

With Course 4 providing a foundation, the cells of Course 3 provide customization opportunities as each cell could either be a loop connected to Course 4, or a float, as reflected in the Figure 3 decision tree. These options were drawn in Rhinoceros 3D as a new layer as shown in Figure 5. It is important to note that each arc and straight line remained an individual segment that could be individually selected from Grasshopper, allowing each cell to be manipulated separately. This principle is similar to a seven segment display commonly found in digital clocks.

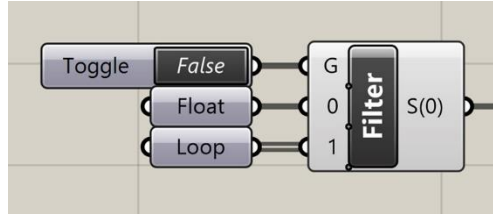


**Figure 5:** Cells in Course 3 (green) can either be loops or floats.

Beginning with Course 3, Cell 1, the lines describing a float and a loop were separately imported into Grasshopper using the *Curve* function, shown in Figure 6 and labeled accordingly. These were then connected to a *Stream Filter*, with a *Boolean Toggle* used to switch between the two curves. In Grasshopper, a Boolean Toggle acts like a digital switch which can be turned True or False by clicking on it. Therefore, when the Boolean Toggle was False (or 0, corresponding to an empty cell in a knit diagram), only the float geometry was sent through the Stream Filter to the next stage of the algorithm. Correspondingly, when the Boolean Toggle was made True (or 1, corresponding to a X in a knit diagram), only the Loop geometry was sent through the Stream Filter. This may also be written as:

**if** Course 3 Toggle = False **then** Stream Filter: Float **else** Stream Filter: Loop.

Within Grasshopper, each cell of Course 3 was connected to its own set of code like Figure 6, providing five Boolean Toggles that could be used to customize the course.

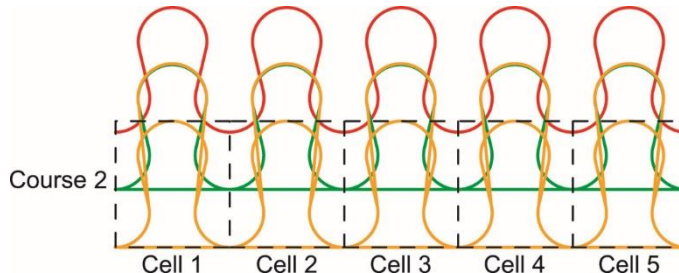


**Figure 6:** Grasshopper code that imports float and loop geometry from a cell and uses a Boolean Toggle to switch between them.

### 4.3 Course 2

Course 2 increased coding complexity as its geometry was parametrically affected by the decisions made for each cell in Course 3, whilst also providing its own opportunities for customization. The parametric relationship to Course 3 was essential to ensure the system always resulted in a knit structure that connected together during later assembly. Given that Course 3 provided the opportunity for floats, Course 2 required added intelligence to use a held stitch rather than a loop when the cell in the previous wale used a float. However, a user simply decided whether a cell is a float or a loop, and the algorithm utilized the Boolean logic of Figure 3 to automate the selection of a loop or held stitch.

Given the increased options for Course 2, the 2D linework in Rhinoceros 3D increased as evidenced in Figure 7. The base geometry of loops and floats were the same as Course 3, however, to accommodate for a held stitch, geometry including the top arc of Course 3 in situ was included and connected down to the lower arcs of Course 3, meaning the geometry extended beyond the cell boundaries.



**Figure 7:** Cells in Course 2 (yellow) may be loops, floats or held stitches that extend to replace geometry in Course 3.

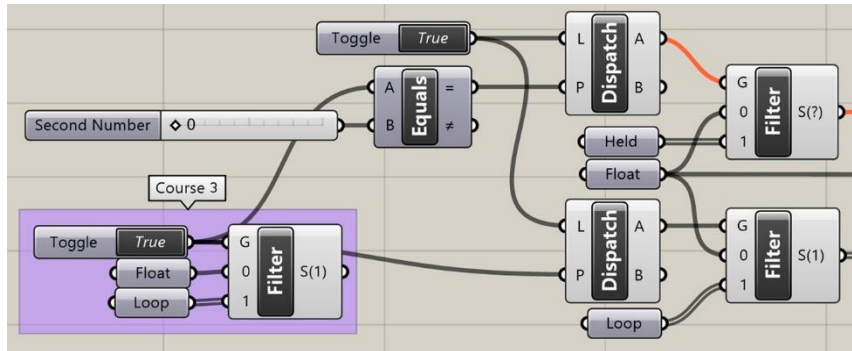
Following the logic in Figure 3 there were two pathways for each cell in Course 2 based on the choices made in the previous wale cell: If a float was selected for Course 3, the cell in Course 2 may also be a float or will become a held stitch to stretch over the Course 3 float and connect with the loop in Course 4. Alternatively, if a loop was selected for Course 3, the choices for Course 2 became float or loop. This can be expressed as:

```

if Course 3 Toggle = False AND Course 2 Toggle = False then Stream Filter: Float
else if
Course 3 Toggle = False AND Course 2 Toggle = True then Stream Filter: Held
else if
Course 3 Toggle = True AND Course 2 Toggle = False then Stream Filter: Float
else
Stream Filter: Loop

```

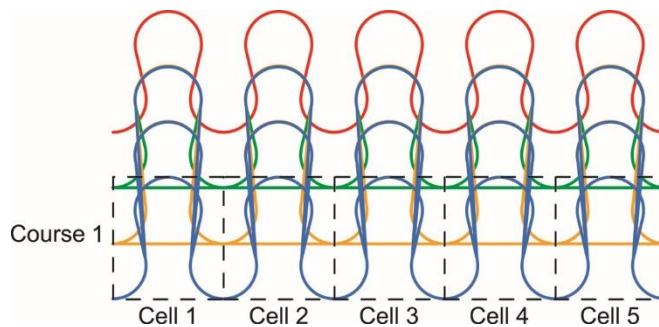
Figure 8 shows the Grasshopper code that represents this logic, and while visually more complicated than the decision tree, can be read from left to right in a similar way. Feeding from the left is the grouped code of a Course 3 cell from Figure 6, which influences whether the 2D curves to be dispatched were those of a float or held stitch, or a float or loop stitch. In Figure 8, this distinction is shown through the upper and lower Dispatch commands, connected to the upper and lower Stream Filters; only one of the Dispatch commands can be active at a time, which in Figure 8 is the lower one. This then dispatches to the Stream Filter the result of the second Boolean Toggle which the user controls, and sends out the loop geometry since Course 3 = True (loop) and Course 2 = True (loop). The code of Figure 8 was repeated for each cell of Course 2 and connected to the corresponding code for Course 3 cells.



**Figure 8:** Grasshopper code for a single cell in Course 2 that dispatches a Loop.

#### 4.4 Course 1

Within this four-course system the final course must connect all loops and ensure any later collections can be connected. Therefore, the user could not customize the pattern of loops and floats, with the geometry automatically building based on decisions made for earlier courses. The 2D linework expanded from previous courses to include geometry from Course 3 and Course 2, as shown in Figure 9, with the possibility that a held stitch would need to stretch across two courses to connect with a loop in Course 4 if Course 3 and Course 2 wale cells contained floats.

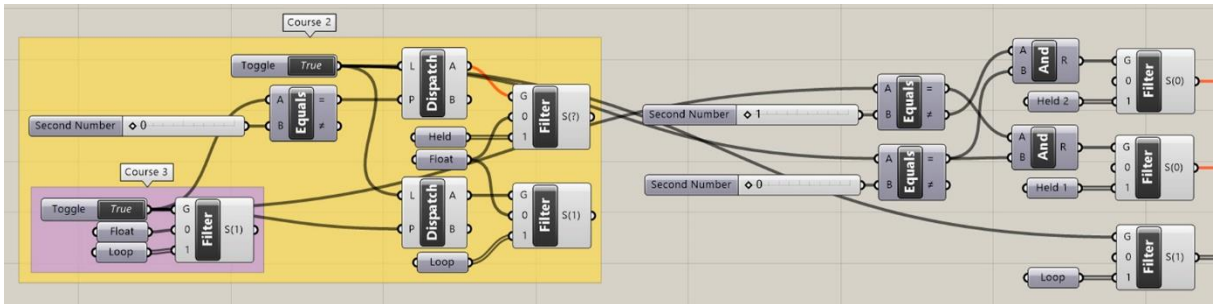


**Figure 9:** Cells in Course 1 (blue) may be loops, floats or single- or double-held stitches that extend to replace geometry in Course 2 or Course 3.

Based on the decision tree in Figure 3, the Grasshopper algorithm continued to build on the information from the previous courses, however, without the Boolean Toggle for user customization. A Stream Filter was used to control the appearance of a loop, single held stitch or double held stitch, with AND gates checking the Boolean Toggles of the previous wale cells to compare against True (1)



or False (0) conditions. As an example, Figure 10 shows Course 3 = True (loop) and Course 2 = True (loop), therefore a loop is being sent through the lower Stream Filter for Course 1. This completed the Boolean logic component of the Grasshopper algorithm and resulted in a collection of 2D lines describing a customized knit pattern.

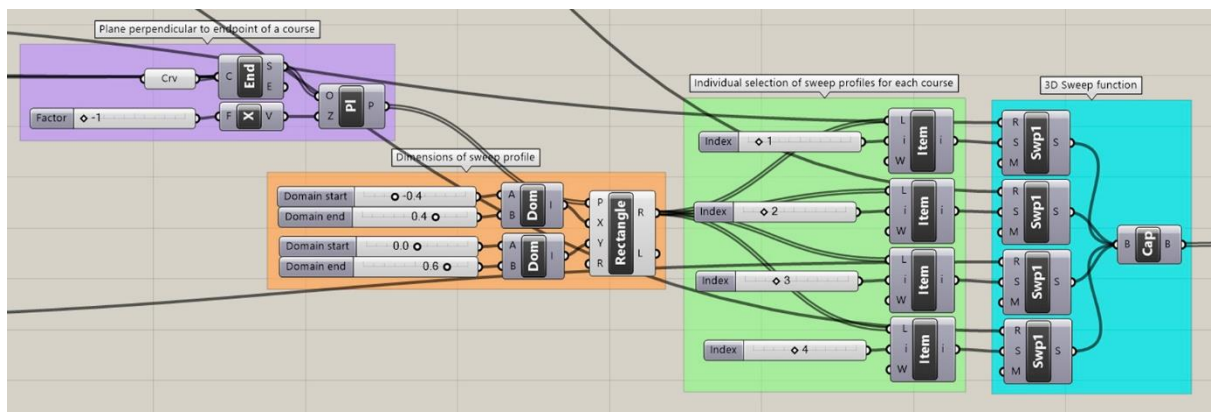


**Figure 10:** Grasshopper code for a single cell in Course 1 that dispatches a Loop.

#### 4.5 2D to 3D Geometry

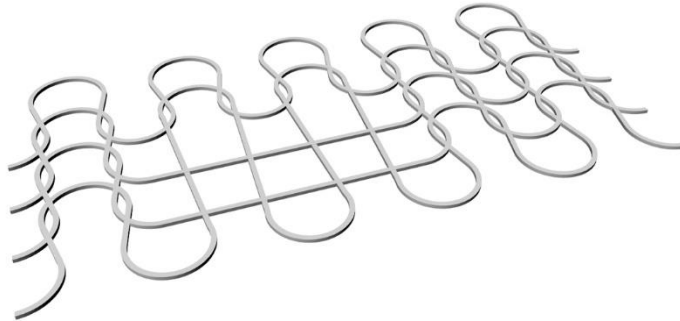
Grasshopper was also used to translate the 2D linework into 3D geometry suitable for 3D printing. Considering the 2D course lines as open pathways, the addition of a closed profile allowed a sweep function to be used to create 3D geometry following a process familiar to operators of other 3D CAD software. The processes to achieve this have been grouped in Figure 11 and can be read from left to right, detailed below:

1. Purple: Creation of a plane perpendicular to the endpoint of each course.
2. Orange: Creation of a closed 2D rectangle on the four planes, using the dimensions described by the domains for X and Y. For example, a domain of -0.4–0.4 resulted in a X dimension of 0.8mm, centered on the course pathway. Users could optimize these for their specific 3D printer using equations such as those described by Novak et al. [20], or for more aesthetic concerns.
3. Green: Each rectangle was separated using a *List Item* function.
4. Blue: Each rectangle was swept along the corresponding course pathway and finished with an end cap.



**Figure 11:** Grasshopper steps to translate the 2D linework into 3D geometry.

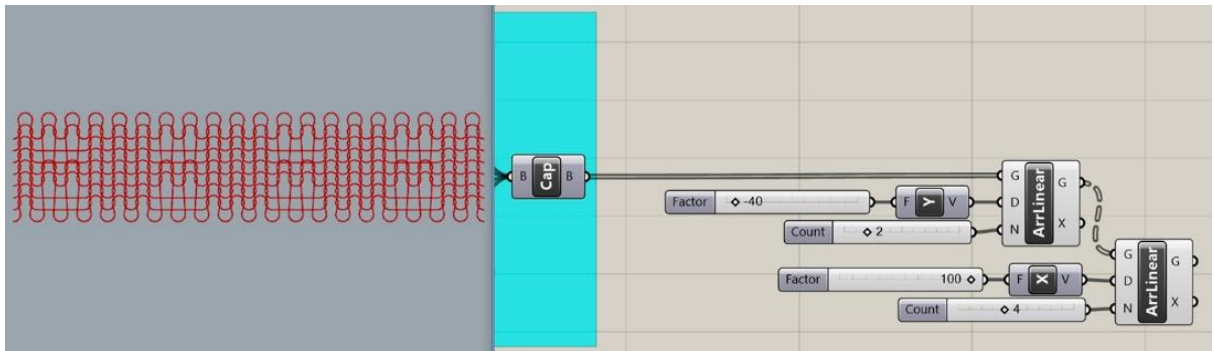
The result of this step was 3D geometry in Rhinoceros 3D that was customizable in cross-sectional size, which when *Baked* (shifted from a visualization linked to Grasshopper to functional geometry in Rhinoceros 3D), resulted in each course being a separate solid object that could be exported as an individual STL for 3D printing, as shown in Figure 12.



**Figure 12:** 3D geometry in Rhinoceros 3D.

#### 4.6 Scalability

A final opportunity for user customization was incorporated into Grasshopper and could be adjusted prior to the Bake step. While the algorithm was designed with a 100mm course length to suit small desktop FDM printers, many printers offer larger build volumes and users may wish to maximize this space. Using a series of *Linear Array* functions in Grasshopper, the user chooses how many times to copy the original pattern in the X and Y directions, for example Figure 13 shows a collection of two courses in the Y direction, with four repetitions in the X direction resulting in a 400mm course length. The Y Linear Array is for visualization purposes only since production with 3D printing is simply a repetition of the initial STL files, and modification to the cells (loop or float) will update the visualization in real-time prior to baking. However, the end-to-end arrangement of courses in the X direction means they can be joined together to form a single long course, examples of which are shown in the Results section.

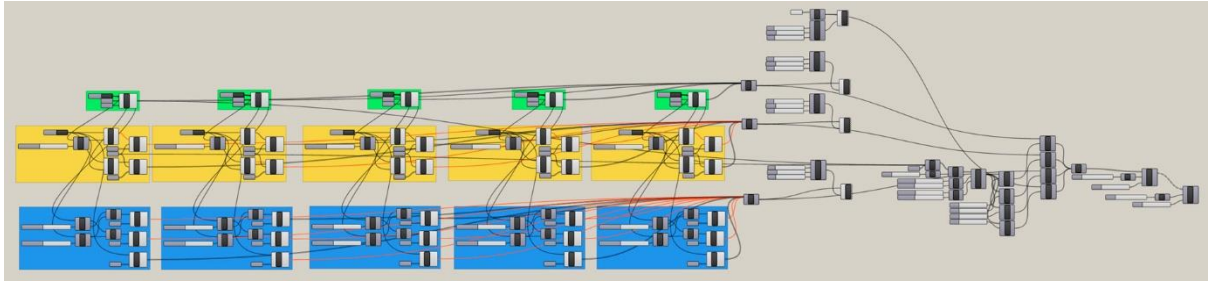


**Figure 13:** Larger swatch of knit pattern in Rhinoceros 3D (left) and the concluding Grasshopper code used to create it (right).

#### 4.7 User Interface

The initial intent of this system was to allow novice CAD users a simple method of customizing a knit pattern for 3D printing. Given the complexity of the Grasshopper code in its entirety, shown in Figure 14, a novice user may be overwhelmed by the complexity of the system and may not locate the specific blocks that allow customization. However, unlike traditional text-based code which must be

structured in a specific order, the blocks of code in Grasshopper can be moved around the workspace and connections hidden, creating a simplified space for interaction as shown in Figure 15.



**Figure 14:** Complete Grasshopper code for the knit system with colored groupings representing cells in a course (please see previous figures for code details of colored groups).



**Figure 15:** Modified user interface for customizing the knit.

Decisions for the user were reduced to loop or float for cells in Course 3 and Course 2, followed by basic 3D knit geometry decisions, and finally an option to extend the size of the courses for larger build plates (in increments of 100mm) and visualize the larger textile. The True/False decisions for cells removed the opportunity for errors in the system, with the clickable switch a basic digital means for controlling the knit using logic similar to traditional knit diagrams. Numerical sliders were then used to customize the 3D geometry, with minimum and maximum values constrained to values that could be manufactured using FDM without erroneous geometry. This is comparable to online systems employed by Nervous System to customize a 3D printable dress (<https://n-e-r-v-o-u-s.com/kinematicsCloth/>) or the NUT JOB customizer on Thingiverse that allows people to easily customize CAD models of nuts and bolts for 3D printing ([https://www.thingiverse.com/apps/customizer/run?thing\\_id=193647](https://www.thingiverse.com/apps/customizer/run?thing_id=193647)).

Additionally, the overall course length was limited to 1000mm to represent build plates for large FDM printers such as the BigRep ONE, which features a build plate of 1005mm in both the X and Y dimensions [21]. The algorithm was also limited to a maximum of 10 collections repeating in the Y direction due to the computer graphics hardware requirements to generate this geometry on screen in real-time, with each additional course reducing system performance. By hiding the more complex code behind such decisions, the simplified interface encourages experimentation and play using familiar control mechanisms, allowing rapid iteration without the need to manually manipulate 3D CAD geometry or risk creating geometry unsuitable for 3D printing (such as being non-manifold). Future testing with users such as fashion designers will evaluate the effectiveness of these parameters.

## 5 RESULTS

To evaluate the system several 3D printed samples were produced. The first was optimized for a cheap desktop FDM machine called the Duplicator i3 (Wanhao, China). A 100mm course length was used, along with a 0.8mm cross-section width optimized using data from Novak et al. [20] for the specifications of the printer, and a 0.6mm thickness. The four courses were 3D printed individually in Acrylonitrile Butadiene Styrene (ABS) plastic as shown in Figure 16, and then assembled by hand by interconnecting loops. Two collections of courses were connected to validate the need for a static Course 4 and parametrically controlled Course 1, ensuring the system was expandable to produce any desired length of textile. As a further test, a longer 900mm course length was created for production on a BigRep ONE machine (BigRep, Germany), with the result of several collections assembled together also shown in Figure 16.



**Figure 16:** 3D printed sample of knit using desktop Wanhao Duplicator i3 (FDM) and ABS material on the left, with a 900mm sample printed on a BigRep ONE and PLA material on the right.

More advanced CAD users may utilize Rhinoceros 3D and Grasshopper to generate more complex knit structures and geometries that could not be produced through traditional knitting. For example, rather than the fluid linework used in the previous examples, a more pointed geometry could be substituted as shown in Figure 17. In this case no floats were used, allowing a running bond method of assembly to be trialed that expands the potential size of the textile beyond the limitations of a small 3D printer build plate. Such angular knit geometry is not possible with thread and presents new opportunities for designers to experiment with texture, flexibility and aesthetic pattern using this conditional logic methodology. Further development of the algorithm would be required to enable floats to be incorporated into a running bond assembly.



**Figure 17:** Variation to knit geometry and ability to stagger assembly using a running bond pattern. 3D printed on a Duplicator i3 (Wanhao, China) in ABS material.

## 6 DISCUSSION

The system in its current form does have notable limitations when compared to traditional knitting processes; for example, through the use of binary Boolean Toggles, users only have the choice between two cell features, being float or loop. Knitting contains many varied techniques such as those outlined in BS EN ISO 4921:2002 standards [31], allowing people and machines to build up complex patterns for a variety of aesthetic and functional needs. Increasing cell geometries is possible in Grasshopper, however, would increase the complexity of the code and may have negative effects on the speed at which visualizations occur in real-time. This is an area of ongoing development, and the method should allow researchers to further experiment with 2.5D knits.

Similarly, this study limited experiments to four courses of knit, meaning repetition in collections of courses is visually noticeable. Following the conditional logic presented, advanced users may choose to expand the number of courses that can be customized, creating new held stitch geometries that can cross three or more floats beyond the constraints of traditional knitting hardware. However, it is likely that this will place increased requirements on computer graphics hardware to process, and may also have negative effects on real-time visualizations. Future studies may consider removing the real-time visual element to improve performance, or substituting a simplified diagrammatic method of visualization, with 3D geometry only becoming available when Baked into Rhinoceros 3D as the final step ready for STL export and 3D printing.

While studies have shown Grasshopper to be embraced by design disciplines like Product and Architecture [6, 19], it is not clear whether fashion and textile designers will as readily adopt the system. Online integration of Grasshopper code is possible through providers like Shapediver (<https://www.shapediver.com>), providing an experience more akin to online product customization platforms mentioned previously, and removes the need to access Rhinoceros 3D and Grasshopper on a computer with reasonable processing hardware. This is part of broader shifts in design that focus on parametrically constrained systems that empower prosumers [1, 30, 32], and is an opportunity to shift the physical atoms of fashion and textiles into the digital bits necessary for Industry 4.0.

## 7 CONCLUSION

This paper proposed a novel CAD system to customize knit structures using Boolean conditional logic and parametric design features. Grasshopper 3D was used to develop a visual algorithm that constrained the number of controls available to play with patterns of knit and float geometry visualized in simple 2D linework, before being modified in 2.5D to suit the specifications of a user's 3D printer. Results were validated at both a desktop scale of 100mm and larger industrial scale of 900mm in length, being 3D printed flat without need for support material, and assembled by hand as a hybrid process to creating complex 3D printed knit textiles. Final results also show an ability to substitute the traditional geometry of a knit with new geometries of a user's creation, suggesting new research directions into both CAD and 3D printing of knits.

## 8 ACKNOWLEDGEMENTS

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

James I. Novak, <https://orcid.org/0000-0003-4082-4322>

## REFERENCES

- [1] Ahluwalia, P.; Miller, T.: The prosumer, *Social Identities*, 20(4-5), 2014, 259-261. <https://doi.org/10.1080/13504630.2015.1004830>

- [2] Araújo, M. d.; Figueiro, R.; Hong, H.: Modelling and simulation of the mechanical behaviour of Weft-Knitted Fabrics for Technical Applications, *AUTEX Research Journal*, 4(2), 2004, 72-80.
- [3] Beecroft, M.: 3D printing of weft knitted textile based structures by selective laser sintering of nylon powder, In *Proceedings of the 2016 Global Conference on Polymer and Composite Materials (PCM 2016)*, Hangzhou, China, 2016. <https://doi.org/10.1088/1757-899X/137/1/012017>
- [4] Beecroft, M.: Digital interlooping: 3D printing of weft-knitted textile-based tubular structures using selective laser sintering of nylon powder, *International Journal of Fashion Design, Technology and Education*, 2019, 1-7. <https://doi.org/10.1080/17543266.2019.1573269>
- [5] Campbell, I.; Bourell, D.; Gibson, I.: Additive Manufacturing: Rapid Prototyping Comes of Age, *Rapid Prototyping Journal*, 18(4), 2012, 255-258. <https://doi.org/10.1108/13552541211231563>
- [6] Celani, G.; Vaz, C.: CAD Scripting And Visual Programming Languages For Implementing Computational Design Concepts: A Comparison From A Pedagogical Point Of View, *International Journal of Architectural Computing*, 10(1), 2012, 121-138. <https://doi.org/10.1260/1478-0771.10.1.121>
- [7] Chavez, E.: Blossoming on the Catwalk: XYZ Workshop's 3D Printed inBloom Dress, url: <https://3dprintingindustry.com/news/blossoming-catwalk-xyz-workshops-3d-printed-inbloom-dress-31769/>
- [8] Davis, F.: 3D Printed Textiles from Textile Code: Structural Form and Material Operations, In *Proceedings of the Sigradi 2012*, Fortaleza, Brazil, 2012.
- [9] Diegel, O.; Wohlers, T.: *Design for Additive Manufacturing Presents Opportunities for Software Developments*, Inovar Communications Ltd, 2018.
- [10] Efrat, T. A.; Mizrahi, M.; Zoran, A.: *The Hybrid Bricolage: Bridging Parametric Design with Craft through Algorithmic Modularity*, ACM, 2016.
- [11] Goktepe, O.; Harlock, S. C.: Three-Dimensional Computer Modeling of Warp Knitted Structures, *Textile Research Journal*, 72(3), 2002, 266-272. <https://doi.org/10.1177/004051750207200314>
- [12] Guo, K.; Wang, X.; Wu, Z.; Zhang, T.; Zhou, M.: *Modelling and Simulation of Weft Knitted Fabric Based on Ball B-Spline Curves and Hooke's Law*, IEEE Computer Society, 2015.
- [13] Korger, M.; Bergschneider, J.; Lutz, M.; Mahltig, B.; Finsterbusch, K.; Rabe, M.: *Possible Applications of 3D Printing Technology on Textile Substrates*, IOP Publishing, 2016.
- [14] Krassenstein, E.: Danit Peleg Creates First 3D Printed Fashion Collection Printed Entirely at Home, url: <https://3dprint.com/83423/danit-peleg-3d-printed-fashion/>
- [15] Loy, J.; Canning, S.: Clash of Cultures: Fashion, Engineering, and 3D Printing, In *Creative Technologies for Multidisciplinary Applications*, edited by A. Connor and S. Marks, IGI Global, Hershey, PA, USA 2016, 25-53. <https://doi.org/10.4018/978-1-5225-0016-2.ch002>
- [16] Melnikova, R.; Ehrmann, A.; Finsterbusch, K.: 3D printing of textile-based structures by Fused Deposition Modelling (FDM) with different polymer materials, *IOP Conf. Series: Materials Science and Engineering*, 2014.
- [17] Mikkonen, J.; Kivioja, S.; Suonsilta, H.; Myllymäki, R.; Vanhakartano, S.: *Printed Material and Fabric*, In *Proceedings of the Nordic Design Research Conference*, Copenhagen-Malmö, 2013.
- [18] Novak, J.: *Drawing the Pen: From Physical to Digital and Back Again*, In *Proceedings of the Drawing International Brisbane*, Brisbane, Australia, 2015.
- [19] Novak, J.; Loy, J.: *Recoding Product Design Education: Visual Coding for Human Machine Interfaces*, In *Proceedings of the DESTech*, Deakin University, Geelong, Australia, 2016. <https://doi.org/10.18502/keg.v2i2.620>
- [20] Novak, J. I.; Liu, M. Z.-E.; Loy, J.: *Designing Thin 2.5D Parts Optimized for Fused Deposition Modeling*, In *Additive Manufacturing Technologies from an Optimization Perspective*, edited by K. Kumar, D. Zindani and J. P. Davim, IGI Global, Hershey, PA, USA 2019, 134-164. <https://doi.org/10.4018/978-1-5225-9167-2.ch007>

- [21] Novak, J. I.; O'Neill, J.: A Design for Additive Manufacturing Case Study: Fingerprint Stool on a BigRep ONE, *Rapid Prototyping Journal*, 2019. <https://doi.org/10.1108/RPJ-10-2018-0278>
- [22] Oliveira, N. P.; Maciel, L.; Catarino, A. P.; Rocha, A. M.: Parametric modelling design applied to weft knitted surfaces and its effects in their physical properties, In *Proceedings of the 17th World Textile Conference AUTEX, Corfu, Greece, 2017*). <https://doi.org/10.1088/1757-899x/254/16/162008>
- [23] Partsch, L. N.; Vassiliadis, S.; Papageorgas, P.: 3D Printed Textile Fabrics Structures, In *Proceedings of the 5th International Istanbul Textile Congress, Istanbul, Turkey, 2015*.
- [24] Pei, E.; Shen, J.; Watling, J.: Direct 3D printing of polymers onto textiles: experimental studies and applications, *Rapid Prototyping Journal*, 21(5), 2015, 556-571.
- [25] Reilly, L.: The shift from 3D body scanned data to the physical world, In *Proceedings of the Shapeshifting Conference, Auckland University of Technology, 2014*.
- [26] Renkens, W.; Kyosev, Y.: Geometry modelling of warp knitted fabrics with 3D form, *Textile Research Journal*, 81(4), 2011, 437-443. <https://doi.org/10.1177/0040517510385171>
- [27] Romeo, L. D.; Lee, Y.-A.: Creative and technical design skills: are college apparel curriculums meeting industry needs?, *International Journal of Fashion Design, Technology and Education*, 6(3), 2013, 132-140. <https://doi.org/10.1080/17543266.2013.783629>
- [28] Rosenkrantz, J.; Louis-Rosenberg, J.: Dress/Code Democratizing Design Through Computation and Digital Fabrication, *Architectural Design*, 87(6), 2017, 48-57. <https://doi.org/10.1002/ad.2237>
- [29] Seepersad, C. C.: Challenges and Opportunities in Design for Additive Manufacturing, *3D Printing and Additive Manufacturing*, 1(1), 2014, 10-13. <https://doi.org/10.1089/3dp.2013.0006>
- [30] Shugrina, M.; Shamir, A.; Matusik, W.: Fab Forms: Customizable Objects for Fabrication with Validity and Geometry Caching, *ACM Transactions on Graphics*, 2015.
- [31] British Standard: Knitting - Basic concepts - Vocabulary, 2002.
- [32] Tapscott, D.; Williams, A. D.: *Wikinomics: How Mass Collaboration Changes Everything*, The Penguin Group, USA 2006.
- [33] Thompson, M. K.; Moroni, G.; Vaneker, T.; Fadel, G.; Campbell, R. I.; Gibson, I.; Bernard, A.; Schulz, J.; Graf, P.; Ahuja, B.; Martina, F.: Design for Additive Manufacturing: Trends, Opportunities, Considerations, and Constraints, *CIRP Annals*, 65(2), 2016, 737-760. <https://doi.org/10.1016/j.cirp.2016.05.004>
- [34] Ujiie, H.: Computer technology from a textile designer's perspective, In *Computer Technology for Textiles and Apparel*, edited by J. Hu, Woodhead Publishing 2011, 245-258. <https://doi.org/10.1533/9780857093608.3.245>
- [35] Wang, J.; Lu, G.; Li, W.; Chen, L.; Sakaguti, Y.: Interactive 3D garment design with constrained contour curves and style curves, *Computer-Aided Design*, 41(9), 2009, 614-625. <https://doi.org/10.1016/j.cad.2009.04.009>
- [36] Zhu, K.; Dancu, A.; Zhao, S.: FusePrint: A DIY 2.5D Printing Technique Embracing Everyday Artifacts, *ACM, City*, 2016.