Design Computing of Complex-Curved Geometry using Digital Fabrication Methods

by

Kenfield A. Griffith

MASTER OF SCIENCE IN ARCHITECTURE STUDIES
AT THE
MASSACHUSSETTS INSTITUTE OF TECHNOLOGY
JUNE 2006

Signed by ____

Kenfield A. Griffith
Department of Architecture and Planning, MIT

Certified by ____

Larry Sass
Associate Professor of Architecture, Department of Architecture and Planning, MIT
Thesis Advisor

Accepted By **Julian Beinart**
Professor of Architecture, Department of Architecture and Planning, MIT
Chairman, Departmental Committee on Graduate Students

READERS

Terry Knight
Professor of Computation, Department of Architecture and Planning, MIT
Thesis Co-Advisor

Fernando P. Domeyko
Senior Lecturer in Architecture, Department of Architecture and Planning, MIT
Thesis Co-Advisor

Design Computing of Complex-Curved Geometry using Digital Fabrication Methods

by

Kenfield A. Griffith

MASTER OF SCIENCE IN ARCHITECTURE STUDIES
AT THE
MASSACHUSSETTS INSTITUTE OF TECHNOLOGY
JUNE 2006

Abstract

The production of design information for digital fabrication is presented in this thesis. This thesis outlines the research of generating information for physical construction as architectural models of complex curved walls built from unique units. A series of computer programs and physical models as examples of orthogonal, non-orthogonal, and complex curved walls as designs were developed. The wall examples here are built of non-uniform, interlocking units using an integral connection approach. This is an exploration of design tools that construct complex curved structures in CAD for fabrication with a 3D printer. The thesis explores the evaluation processes used by architects when evaluating digitally fabricated desktop models. The research involved in this thesis takes the direction of investigating a new methodology for solving a modern and aesthetic approach to architecture. The research conducted investigates design as a way for synthesizing a grammatical (Stiny, 1977) approach as the systematic engine that is used to solve less systematic, curved, non-uniform form (Smithers, 1989).

TABLE OF CONTENTS

Abstract 3
Chapter 1 12
1.0 Introduction 12
Chapter 2 17
2.0 Background and Purpose 17
2.1 Limitation of 3D fabrication devices 17
2.2 Fabricating physical models from systematically generated digital models 17
2.2.1 Shape grammars as the design generator 18
2.2.2 Parametric tools as the design generator 18
2.3 Evolutionary processes and cognition in design 19
2.3.1 Dialogue between architect and computer 19
2.3.2 Processes of Design 20
2.4 Complexity of multi-curved surfaces 20
2.4.1 Computation and aesthetics 20
2.4.2 Synclastic and Anticlastic curvature 21
2.4.3 Standard fabrication tools within Architecture 22
2.5 Fabrication as evaluation mechanism 22
2.5.1 Physical models in the Design Process 22
2.5.2 "Refabrication" 23
Chapter 3 24
3.0 Masonry systems 25
Method of thesis 25
3.1 Creating a Design Description for assembly 25
3.2 Results using a less systematic design approach 26
3.3 Creating a systematic design language 27
3.4 Results using a systematic design approach 27
3.5 Mapping of digital fabrication technologies 29
3.3 STANDARD WALL SYSTEM (orthogonal) 30
3.3.1 How the algorithm works 30
3.3.2 Construct divisions and array of information 30
3.1.3 Construct initial surfaces of each unit 30
3.3.4 Construct initial unit block 31
3.3.5 Construct side tabs for each unit 31
3.3.6 Construct top tabs for each unit 31
3.3.7 Construct notches for Boolean operations 31
3.3.8 Fabricate each discrete piece using ZPrinter printing technology 33
3.3.9 Results of STANDARD WALL SYSTEM 33
3.4 NON-STANDARD WALL SYSTEM (non-orthogonal) 34
3.4.1 How the algorithm works 34
3.4.2 Divide sides of void 34
(figure 3.15 thru 3.24 on page 37) 34
3.4.3 Create outline grid for units 34
3.4.4 Find intersection of lines for surface boundaries 35

3.4.5 Construct the thickness of unit 35
3.4.6 Create tabs and notches 35
3.4.7 Boolean operations 36
3.4.8 Weight factor of units 36
3.4.9 Fabrication and assembly of units 37
3.4.11 Results of NON-STANDARD WALL SYSTEM 39
3.5 **COMPLEX-CURVED WALL SYSTEM** 40
3.5.1 How the algorithm works 40
3.5.2 Divide initial curve for required number of units 40
3.5.3 Construct boundary curves for units 41
3.5.4 Construct initial unit volume 41
3.5.5 Create tabs and notches 41
3.5.6 Fabrication and assembly of units 41
3.5.7 Results of COMPLEX-CURVED WALL SYSTEM 42
3.6 **COMPLEX-CURVED WALL SYSTEM (semi-enclosed form)** 43
AVAILABLE FUNCTIONS 43
GLOBAL VARIABLE DEFINITIONS 44
FUNCTIONS 44
Chapter 4 53
4.0 Results and Limitations 53
4.1 Manageable assembly 53
4.2 Integral connections in the assembly 54
4.3 Variation in design scale 55
4.4 Mapping data to other fabrication devices 58
4.5 Structural Limitations 58
Chapter 5 59
5.0 Contributions and Future Exploration 59
5.1 Contributions 59
5.1.1 Fabrication devices within and design process 59
5.1.2 Complexity of designing with curvature 60
5.1.3 Design computing 60
5.2 Future Exploration 61
5.2.1 Structural investigation 61
5.2.2 Larger structures 62
5.2.3 A seamless design computing process 62

TABLE OF FIGURES

FIGURE 1.0 *COMPLEX-CURVED MASONRY CONSTRUCTION*....13
FIGURE 1.1 *FABRICATION OF STYROFOAM MOLDS*....13
FIGURE 1.2 *COMPUTER MODEL OF STATA CENTER*....13
FIGURE 1.3 *BUILT STATA CENTER*....14
FIGURE 1.4 *COMPUTER DESIGNED MODEL*....14
FIGURE 1.5 *VARIATION OF COMPLEX MODELS*....14
FIGURE 1.6 *DIMENSION OF ZPRINTER BED*....15
FIGURE 1.7 *ORTHOGONAL WALL GEOMETRY*....15
FIGURE 1.8 *NON-ORTHOGONAL WALL GEOMETRY*....15
FIGURE 1.9 *COMPLEX-CURVED WALL GEOMETRY*....16
FIGURE 1.10 *SEMI-CLOSED COMPLEX FORM*....16

FIGURE 2.0 *3D SHAPER APPLICATION*....18
FIGURE 2.1 *INTERLOCKING COMPONENTS*....19
FIGURE 2.2 *TETRAHEDRON APPLICATION*....20
FIGURE 2.3 *SYNCLASTIC FORM*....21
FIGURE 2.4 *ANTICLASTIC FORM*....21
FIGURE 2.5 *2D FABRICATION FOR 3D APPLICATION*....22

FIGURE 3.0 *GUGGENHEIM MUSEUM*....24
FIGURE 3.1 *PUZZLE-PIECES FOR INTERLOCKING UNITS*....26
FIGURE 3.2 *CASE WESTERN UNIVERSITY*....24
FIGURE 3.3 *TRADITIONAL MASONRY CONSTRUCTION*....25
FIGURE 3.4 *FABRICATION TESTING*....27
FIGURE 3.5 *DESIGN LANGUAGE*....27
FIGURE 3.6 *FABRICATION MAPPING: MODEL → LIFE-SIZE*....29
FIGURE 3.7 *COMPUTER MODEL OF ORTHOGONAL FORM*....30
FIGURE 3.8 *CREATING SURFACE FROM BOUNDARIES*....30
FIGURE 3.9 *CALCULATION OF UNIT SIZES*....31
FIGURE 3.10 *CREATING TAB OF UNIT*....32
FIGURE 3.11 *BOOLEAN OPERATIONS (CSG)*....32
FIGURE 3.12 *UNIT AFTER BOOLEAN OPERATION*....33
FIGURE 3.13 *FABRICATED ORTHOGONAL WALL*....33
FIGURE 3.14 *COMPUTER MODEL OF NON-ORTHOGONAL WALL*....34
FIGURE 3.15 *DIVISION OF CURVES*....38
FIGURE 3.16 *CONNECTION OF DIVISION POINTS*....38
FIGURE 3.17 *GRID OUTLINE OF BRICK UNITS*....38
FIGURE 3.18 *SURFACING OF GRID*....38
FIGURE 3.19 *EXTRUSION OF SURFACES*....38
FIGURE 3.20 *SURFACES FOR TOP TABS AND NOTCHES*....38
FIGURE 3.21 *EXTRUSION OF TAB AND SIDE TABS*....38
FIGURE 3.22 *BOOLEAN OPERATION OF UNITS*....38
FIGURE 3.23 *EXTRACT UNITS FOR WEIGHT*....38
FIGURE 3.24 *LABELING OF UNITS*....38
FIGURE 3.25 *FABRICATED NON-ORTHOGONAL WALL*....39
FIGURE 3.26 *COMPUTER MODEL OF COMPLEX WALL*....40
FIGURE 3.27 *GRID OF COMPLEX WALL*....41
FIGURE 3.28 *EXTRUSION OF EACH SURFACE*....41
FIGURE 3.29 *ARRAY OF ACCEPTED WALL TYPES*....42
FIGURE 3.30 *FABRICATED COMPLEX WALL*....42
FIGURE 3.31 *COMPUTER MODEL OF SEMI-CLOSED FORM*....43

FIGURE 3.32 *THICKNESS OF UNITS*...44
FIGURE 3.33 *CREATION OF POINTS ON FORM*...45
FIGURE 3.34 *CREATION OF GRID*...45
FIGURE 3.35 *CREATION OF UNIT BOUNDARIES*...46
FIGURE 3.36 *CREATION OF UNIT SURFACES*...46
FIGURE 3.37 *BOTTOM OF UNIT*...47
FIGURE 3.38 *NOTCHES AND TABS AT BOTTOM OF UNIT*...47
FIGURE 3.39 *TOP OF UNIT*...48
FIGURE 3.40 *LEFT OF UNIT*...49
FIGURE 3.41 *RIGHT OF UNIT*...49
FIGURE 3.42 *BIND EACH UNIT*...50
FIGURE 3.43 *COMPLETE DECONSTRUCTED FORM*...51
FIGURE 3.44 *FABRICATED SEMI-ENCLOSED MODEL*...52
FIGURE 3.45 *SCALED MODELS*...52

FIGURE 4.0 *EVALUATION PROCESS*... 57
FIGURE 4.1 *DIFFERENCE IN MODEL SIZES*...55
FIGURE 4.2 *2" PLASTIC MODEL*...58

FIGURE 5.0 *CONTOUR CRAFTING DEVICE*... 62

ACKNOWLEDGEMENTS

The academic directions instilled in my brothers, sisters, and me at a young age has given me the courage to attempt and pursue the work conducted in this thesis. I am grateful to be able to inscribe this document with the names of two of the greatest life role models: Mr. Winfield Griffith, my late father and Mrs. Shiena Griffith, my wonderful mother.

There are so many other names that I would like to include but may become a thesis within itself. I would like to include the lucky seven: my beautiful and loving daughter Imaune, brothers and sisters: Gary, Glen, Marva, Sharon, Vanessa, and Brad, all who have helped me discover who I am through the many experiences we have all shared.

This work would not have been possible without the great and scholarly direction of a wonderful advisor and friend, Professor Lawrence Sass (Larry). Thanks for your guidance and showing the way "Morpheus." Thanks to Dr. Isaac Colbert, who has always made time and space in his busy schedule to listen to my inquiries and share advice as I acclimated to the environment at MIT.

I would like to thank the SMArchS 2006 graduates whose energy and intelligence stimulated a lot of thought and investigation throughout my exploration at MIT.

For the two wonderful UROP students I had the privilege working with in the summer of 2005: Alexander Vasquez and Kieren Jamaes-Lubin, I enjoyed sharing and tackling the investigation outlined in this thesis.

Finally, I would like to thank the Department of Architecture at MIT for the plethora of possibilities created which have manifested in this thesis.

Chapter 1

1.0 Introduction

1.1 *Design Computing in Architecture*

The integration of technology and current architectural design processes requires non-conventional solutions for fabrication. Computing complex geometries has become common practice for architects, but creates difficulties in the stages of fabrication and assembly. Computation within the design process instigates architects to design less orthogonal systems and more complex juxtaposition of form that requires negotiation for fabrication processes and material for construction. This thesis examines issues of evaluating and fabricating complex geometry using computational processes. The vision is to be able to use this process as a general approach within multiple design systems for constructing architectural representations and eventually towards life-size solutions. This study addresses a process of using desktop fabrication for design evaluation applying computational design, computer programming, and algorithms.

1.2 *Goal of Thesis*

This thesis introduces a system built to analyze and deconstruct curvilinear form into non-uniform components. The purpose is to be able to evaluate complex curved structures by deconstruction into their respective components for fabrication and assembly. Fabrication of form will aid in the investigation of design concepts as architects will be able to create design models which would then be translated to life-size production.

1.3 *Issues Architects face with digital fabrication methods*

Form and shape investigation done by architect Frank Gehry (figure 1.0) demonstrated the complexity of using masonry units as the elements of construction. Gehry's architecture involved the negotiation of form with the use of standard masonry units and Styrofoam molds (figure 1.1) to reach an approximation of the intended design. This method required formwork and creative labor for reaching the design goal. The material presented in this thesis exposes the issue of using current technologies for negotiating the form and production of constructible units.

Figure 1.0 *Complex-curved masonry construction*
PROJECT: Case Western Reserve University
LOCATION: Cleveland, Ohio
ARCHITECT: Gehry Partners

Illustrates fabrication and construction of the curved wall (red) using masonry units

Frank Gehry and Bernard Cache demonstrate ways to physically construct buildings of glass, metal, and masonry from computer models (figure 1.2), and display the continual design intent of current architectural practice to experiment with more provocative forms. Short comings of these designers' work are in the final constructed building (figure 1.3) and understanding the methods for efficiency of production. Higher level questions are centered on methods to design these shapes as physical models from rapid prototyping devices and redesign based on changes found while exploring the design. Based on published information design evaluation and redesign was an abstract process; physical methods to reason with design shapes and assembly strategies were not until the end of the process.

Figure 1.1 *Fabrication of Styrofoam molds*
PROJECT: Case Western Reserve University
LOCATION: Cleveland, Ohio
ARCHITECT: Gehry Partners

Digitally fabricated Styrofoam molds for building concrete walls to be used as guides for laying bricks

Figure 1.2 *Computer model of Stata Center*
PROJECT: Ray and Maria Stata Center
LOCATION: Cambridge, MA
ARCHITECT: Gehry Partners

Computer representational model of the intended building design

Design experiments using computational tools exemplify the intentions of this thesis which is to evaluate complex shapes (figure 1.4) into units of assembly and to take steps in understanding fabrication technologies within architecture. The evaluation criteria will be based on the designer's hierarchy of interests in the stages of investigating the design models.

1.4 *Exploring integral connection for digital fabrication*

This system offers both a digital and physical solution for dividing curvilinear form into sub-units with embedded connectors that integrally create a continuous complex curved system. Alternatively this thesis explores the design process at the desktop level with curved surface design as the subject and evolutionary processes, generative processes, and parametric processes as the form generators (Bentley, 1999). Design evaluation for this work falls in the realm of physical testing tolerance, structure, assembly, space relation, and aesthetics. Most important, the work explores the issue of redesign of curved surface models (as desktop models) for the production of a variety of design solutions.

The thesis approach tests a concept to build large physical design models from components. There are multiple challenges in this work, modeling unique objects (geometries) as parts of a larger complex curved surface (figure 1.5), and fabrication of 3D models; managing component strength and assembly tolerance. Modeling and 3D printing non-uniform complex curved surfaces is a challenge for any modeling program. The problem is exacerbated for models built larger than the volume of a typical 3D printing machine (figure 1.6). Models larger than 10" square must be built and printed as individual components (Sass, 2005). Conventional modeling of unique shapes for individual manufacture in standard 3D modeling software is time consuming and laborious. There is a layered thought process when attempting to model complex geometric solutions within most 3D software packages which has an immense impact on the architects (Art Westerberg, 1989). Architects are forced to negotiate with 3D software rather than the ease of transforming design into tangible objects. The work is driven by thesis questions focused on design production and precise manufacturing of computed

Figure 1.3 *Built Stata Center*
PROJECT: Ray and Maria Stata Center
LOCATION: Cambridge, MA
ARCHITECT: Gehry Partners

The completed building which maps to the computer model (figure 1.2)

Figure 1.4 *Computer designed model*
Computer design of a complex-curved form using Rhinoceros 3.0

Figure 1.5 *Variation of complex models*

information. The thesis questions ask if it is possible to generate many design variations of curved walls, as large physical models (some 30" in height) from generated computer models. What types of design tools are needed to generate information and what types of systems are needed for redesign? What does the fabricated model allow designer to evaluate?

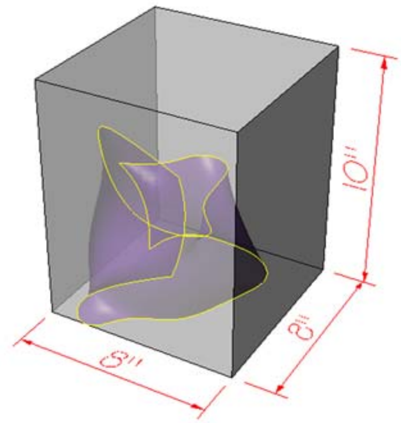


Figure 1.6 *Dimension of ZPrinter bed*
Current limitations of the printing bed of the ZCorp ZPrinter. The bounding box illustrates the size of the bed: 8" x 8" x 10" (203 x 203 x 254)

1.5 *Product of Thesis*

The first program **[Standard Walls]** generates uniform blocks complete with a tab and slot for assembly (Figure 1.7). The program subdivides a rectangular surface (starting shape) as the seed element into evenly shaped units. Each rectangular block is 3D printed as a 1" x 2" block in model complete with a voided space in the back of the block to reduce weight and save on printing time. For example if the starting shape were 8" x 16" the program first subdivides the surface into 64 evenly shaped units, then transforms each unit into a unit with a tab and slot for assembly.

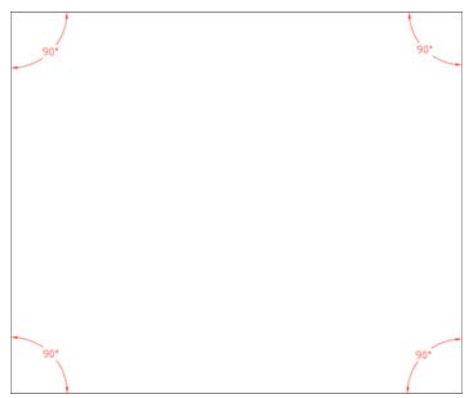


Figure 1.7 *Orthogonal wall geometry*
Standard orthogonal wall that has 90° at all intersection points

The second program **[Non-Standard Walls]** generates non-standard units from an irregular starting shape as the seed for unit generation (Figure 1.8). This program works with a four sided non-orthogonal shape (quadrilateral) as input. Subdivision of the shape creates non-orthogonal units that are transformed to a 3D shape with tabs, slots, and a void in the back. The novelty of this approach is that the tabs and slots are angled based on the shape of the block.

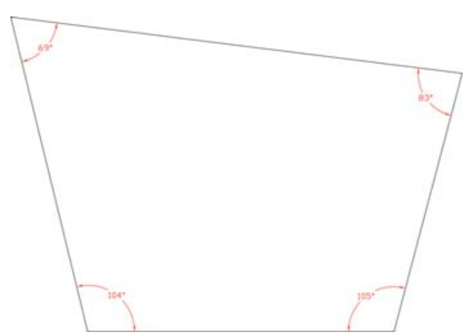


Figure 1.8 *Non-orthogonal wall geometry*
Non-standard wall with angle variances

The third program [**Non-standard Curved Walls]** subdivides complex-curved form into non-uniform components; a process far more complex than the previous two. The third program transforms two curved lines as the initial seed (Figure 1.9). The program evaluates and subdivides the seed, offsets the shape to provide thickness of units.

Finally, the program subdivides the solid into units along the isocurves using a parameter specified by user, and then inserts slots and tabs into each unit. Results of the Non-Standard Curved Wall program are 3D units as part of a larger curved surface with interlocking objects built into each unit. All three methods provide a computational solution that is translated into physical units with embedded connectors for the physical assembly and evaluation of the form.

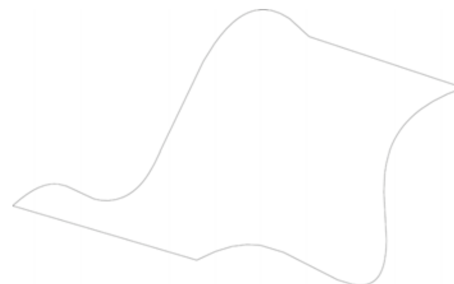


Figure 1.9 *Complex-curved wall geometry*
Complex curved wall that demonstrates a greater degree of invariance in angles and curvature

The fourth program **[Complex curved Form]** uses a design language to divide the surface geometry into respective components of assembly. These geometries unlike the others are semi-closed complex-curved forms (Figure 1.10). The design description includes units that are composed of an integral connecting system that align units for continuity of assembly.

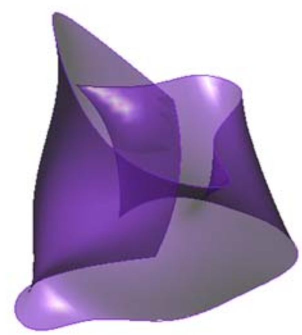


Figure 1.10 *Semi-closed complex form*
Semi-closed complex form that increases the complexity of curvature

1.6 *Fabrication as a Constraint*

The introduction of digital fabrication methods earlier in the design process helps architect to understand ways and methods of rationalizing complex form for creating design models. This approach investigates ways of mapping fabrication methods to life-size solutions easing the final stages of design. This thesis solution addresses the implication of the fabricated solution models which is to map to contour printing technologies for life-size fabrication.

Chapter 2

2.0 Background and Purpose

This research tests a concept of using computing and fabrication as parallel processes for creating accurate design information. This thesis addresses previous investigation to explore some key issues within the design process using recent fabrication technologies and engages in the development of new methods and ways of design computing with complex geometries.

2.1 Limitation of 3D fabrication devices

An obstacle with using 3D printing devices such as the ZCorp printer is the limitation of the printing area (bed size). The ZCorp ZPrinter has a restricted bed size of 8" x 8" x 10" (Figure 1.6) which limits the production of design information larger than 10" inches in height. The purpose of this thesis is to be able to print objects that are bigger than the printing volume of the printer bed and even bigger than the printer. This allows the production of multiple scales of design for the process of evaluation by the architect.

The algorithms presented in this thesis make it feasible for deconstructing form into units at the size of the printing bed as a constraint for assembling into models bigger than the fabricating device.

2.2 Fabricating physical models from systematically generated digital models

2.2.1 Shape grammars as the design generator

Wang and Duarte's (2002) (figure 2.0) approach explained how geometrically complex designs, difficult to produce by hand, can be generated based on a few basic shapes and rules using computational methods. Wang and Duarte's approach explained the transfer of data across platforms from a digitally computed design model to a digitally fabricated solution. With the use of shape grammars for dictating the parameters and combination of shapes, a computational implementation was devised to link the process of design and fabrication. The designs were fabricated using the ZCorp ZPrinter as representations of the computed model.

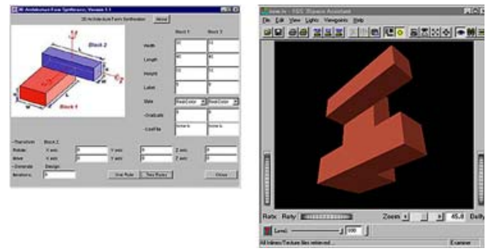


Figure 2.0 *3D Shaper Application*
Wang and Duarte's 3D Shaper design application that uses shape grammars (Stiny, 1980) as the design generator for the production of 3D designs to be printed as solid objects.

Despite the aim to show the integration of technologies from design computing to digital fabrication, this approach fails to show the logic that can be embedded into a discrete component for fabricating larger scale models. The models produced were solid composite representations of the design that had no implication of the life-size fabrication process. The intent of this thesis is to take this methodology a step further by embedding functionality in fabricated components. This thesis addresses the issue of having individual components that can be printed as unique pieces that are assembled together using their connection logic. This thesis illustrates the process of embedding constructability in a model and also offers a solution for dealing with tolerance issues based on material and CAM processes.

2.2.2 Parametric tools as the design generator

Sacks (et al) (2003) explained how parametric variations can drive the design of different solutions for precast concrete based on complexity of the building. They derived solutions based on location of beams and columns and the distribution of weight within the system to calculate the types of precast components needed. This is an important step in creating construction information in the building of architecture, but still creates data as information to be translated by human intervention.

Figure 2.1 *Interlocking components*
3D fabricated models of interlocking assemblies

Their work shows great potential in creating accurate data to be used in the fabrication process. Although this information is viable and necessary, it is still used as information for creating casting mechanisms for producing the precast concrete rather than directly fabricating the units from the data. Furthermore, the solution does not possess any way of interlocking fabricated components (figure 2.1). The process of this thesis illustrates how data can be used as information in the fabrication of composite-like material and creating integral connection (Wasser, 2004) within units for assembly.

2.3 Evolutionary processes and cognition in design

2.3.1 Dialogue between architect and computer

Peter Bentley explains the intersection of disciplines for creating designs (Peter Bentley) in an effort to emphasize the process of evolutionary solutions. He expresses, with the use of examples, the creativity that computers instigate as they introduce possible design solutions or options that were non-existent before applying evolutionary procedures. This study illustrates designs created using computational methods and deconstructed using algorithms within a 3D software environment. The process used to create this thesis involves the integration of computer science and architectural design to solve the fabrication of complex-curved form, providing different design options as results creating a dialogue between architect and computer.

2.3.2 Processes of Design

Kristina Shea (Kristina Shea, 2004) shows the evolutionary process of tiling using grammar rules as the basis for the emergent of tetrahedron shapes. Shea shows a great deal of configuration using adjacency data structures that evolve to build complex 3D form (figure 2.2). Despite the contributions made for evolving new 3D designs, this process is used for visual evaluation of complex form.

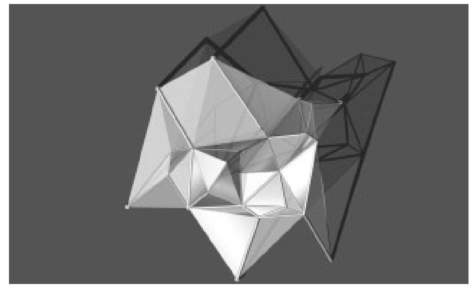


Figure 2.2 *Tetrahedron application*
Kristina Shea tiling software that produces complex tetrahedron shapes as computer models

Shea's designs are based on the concatenation of shapes resulting in tetrahedron structures for intricate design solutions. The design process does not capture connection or the constructability of individual units for physical evaluation. Shea's approach shows advances in design processes but creates design solutions as an abstract artifact. In this thesis the approach uses physical models for evaluating designs. The programs presented in this work use individual units as the assembly modules for the design form. Each unit is constructed based on the integral connection mechanism for producing the end designs from the abstract data.

2.4 Complexity of multi-curved surfaces

2.4.1 Computation and aesthetics

Complex-curved forms have been evident in design results with the use of computational design tools. Architects are given the freedom to explore an exponential array of design possibilities that usually encompass the current notion of "aesthetically pleasing" (Smyth, 2001) and less orthogonal form. Due to the constraints of material and fabrication methods, the solutions for constructability are limited because of the bending and extrapolation of shapes in multiple directions that are difficult to express in models. The digital representation of the design solution is just the forefront of producing the actual architecture. The complications are amplified in the fabrication of solutions for expressing the designer's intent. Past investigations have introduced approaches such as genetic algorithms (Azariadisa, 2002) in the process for reaching fabrication goals that the digital design representation does not address. The aim is to be able to create tangible artifacts for design investigation before the complication of life-size construction. The ability to address constructability provokes better solutions using fabrication as a design constraint which is often ignored in the design process and later addressed when the design has completely manifested into a more complex problem.

2.4.2 Synclastic and Anticlastic curvature

There are two types of complex-curved surfaces that are prevalent in design form; synclastic (figure 2.3), having positive Gaussian curvature and anticlastic (figure 2.4), having negative Gaussian curvature. Complex-curved forms possess both types which creates great issues in resolving for fabrication. The complexity lies in rationalizing 3D form to 2D surfaces, one having positive Gaussian curvature and the other zero or negative Gaussian curvature respectively. This is not a direct mapping in translating for Computer Numerical Controlled fabrication, therefore requires other approaches for fabricating the accuracy of the computed form. This thesis research looks at other ways of fabricating complex geometry using composite materials. The exploration looked at 3D printing technologies

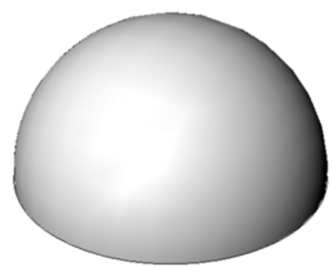


Figure 2.3 *Synclastic form*
Synclastic form having positive Gaussian curvature

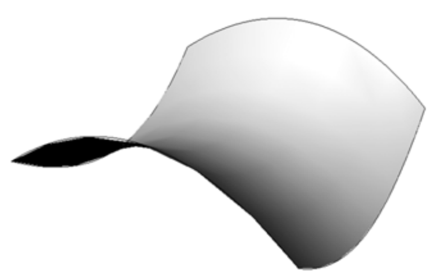


Figure 2.4 *Anticlastic form*
Saddle shape showing anticlastic curvature

for generating units that can be assembled for creating the final form.

2.4.3 Standard fabrication tools within Architecture

Most standard tools used in the stages of fabrication and construction of buildings are those that rely on 2D (figure 2.5) operations for manufacture rather than multi-dimensional operations. These tools usually cut or carve planar surfaces with limited curvature, limiting the production of complex geometries and the architects' exploration of form using computational tools. This thesis examines the use of other fabrication technologies that allow the flexibility of computing and fabricating more complex geometry as 3D information rather than 2D.

Figure 2.5 *2D fabrication for 3D application*
2D fabrication of metal to produce 3D form

2.5 Fabrication as evaluation mechanism

2.5.1 Physical models in the Design Process

Physical design models allow architects to clarify design intent based on design evaluation criteria. Until design process has gotten to the stage of architects understanding and manipulating computed geometry as a tangible medium, it is necessary to produce multiple variations of designs for realizing the complexity of design solutions and exploring the possibilities. Fabrication offers architects the ability to think of ways of rationalizing computed design into tangible means allowing a variety of constructible designs rather than abstract objects.

2.5.2 "Refabrication"

Evaluating design is an ample part of the design that allows the architect to reinsert new findings into the design for creating redesigns. The information presented in this thesis acknowledges the redesign process as a coupling part of the design and redesign process. The algorithms presented allow architects to evaluate fabricated representation model and reinsert information back into the computed model for "refabrication".

The contribution made in this thesis allows architects to think of 3D geometry as it is represented rather than the process of mapping the information to 2D fabrication processes. Mapping 3D to 2D can create inaccuracies in trying to approximate curvature rather than fabricating the actual data as designed.

Chapter 3

Frank Gehry is well known for experimenting with complex-curved shapes at many scales and different types of geometries. The use of computation in his process is mainly to solve geometries for fabrication versus design. From the juxtaposition of form in the Guggenheim (figure 3.0) to the use of masonry units in the Case Western (figure 3.1) for building curved surfaces, Frank Gehry has teased technology within the process of design fabrication. His process takes geometry and rationalizes the form based on fabrication tools, which usually map to 2D operations for cutting sheet metal. The initiative taken to look at other 3D technologies made a notable presence in the production of molds for creating complex-curved form using masonry units.

Figure 3.0 *Guggenheim Museum*
PROJECT: Guggenheim Museum
LOCATION: Bilbao, Spain
ARCHITECT: Gehry Partners

Articulates the juxtaposition of shapes to create complex architecture

Gehry Partners created complex-curved masonry forms in three steps. The digital files were converted into machine data that was used to drive a multi-axis CNC mill. The mill was used to fabricate molds which were in turn used as the form drivers for the concrete that was poured to create the shape. The concrete walls were then used as guides for the masons to follow when laying the masonry. This process instigated a new way of thinking using different methods of fabrication. Despite the lengthy and expensive method and the excess of material, it motivated some of the research carried out in this thesis. This thesis looks into a direct fabrication of the assembly units instead of molds that require excess material, time, and labor.

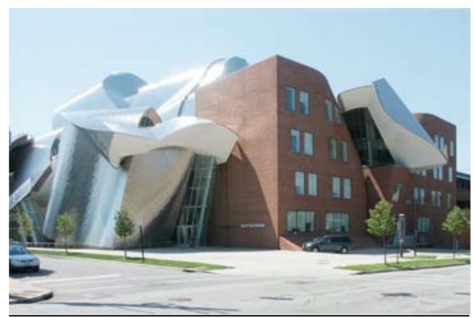
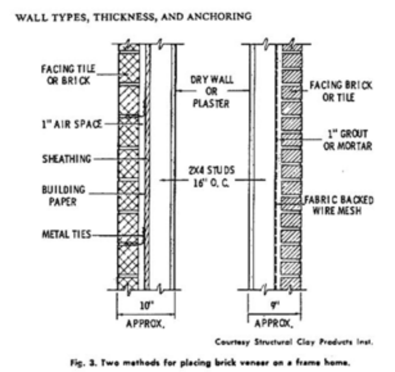


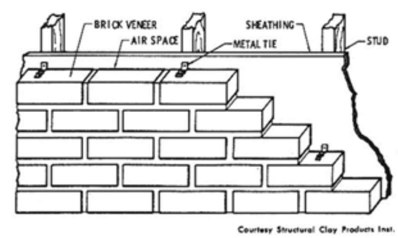
Figure 3.2 *Case Western University*
PROJECT: Case Western Reserve University
LOCATION: Cleveland, Ohio
ARCHITECT: Gehry Partners

The use of bricks for building complex-curved walls

3.0 Masonry systems

Conventional ways of constructing masonry wall units require mason to align masonry (figure 3.2) units along a defined topology. This topology is usually planar, built to follow the formwork with zero Gaussian curvature producing highly orthogonal and Euclidean geometry. The research conducted in this thesis addresses the issues of constructing complex-curved geometries that do not follow the same assembly process as conventional methods. The method constructs the geometry as the building units with connection mechanics in each piece. The units are actually the shape of the form rather that used to shape the form.





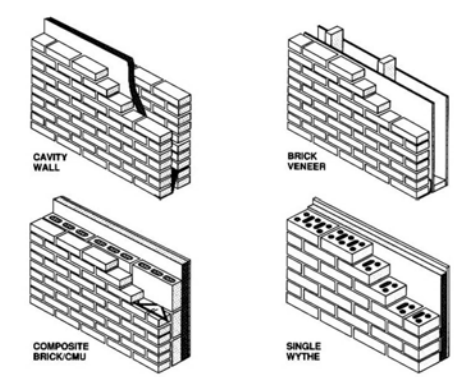


Figure 3.3 *Traditional masonry construction*
Traditional methods of constructing masonry walls that demonstrates the limited use of curved forms

Method of thesis

The geometries built for this thesis were constructed for demonstrating curvature complexity. The form consisted of both synclastic and anticlastic curvature. The algorithms for constructing the non-uniform components described were written in Visual Basics for Rhinoceros® 3.0 and Rhinoceros® 4.0 Work In Progress environments. Fabrication was achieved with the use of the Z Corporation ZPrinter 400, and testing of design description exploration was achieved using AutoCAD 2005.

3.1 Creating a Design Description for assembly

The aim of this thesis was to create a generic method for evaluating complex form into sub-units of assembly. The process explored many puzzle-like configurations for creating a universal system (figure 3.3). Sketches were developed to create interlocking connections that would serve as the integral connections between units. After the sketches were created through conventional hand and pencil methods, information was migrated computationally by importing the sketched images into AutoCAD 2005 for tracing. The unit pieces were then extruded for creating 3Dimensional geometry that would interlock based on the 2D sketch. The geometry was then exported as stereolithography (*.stl) files to be printed and tested for assembly and tolerance using ZPrinter technologies.

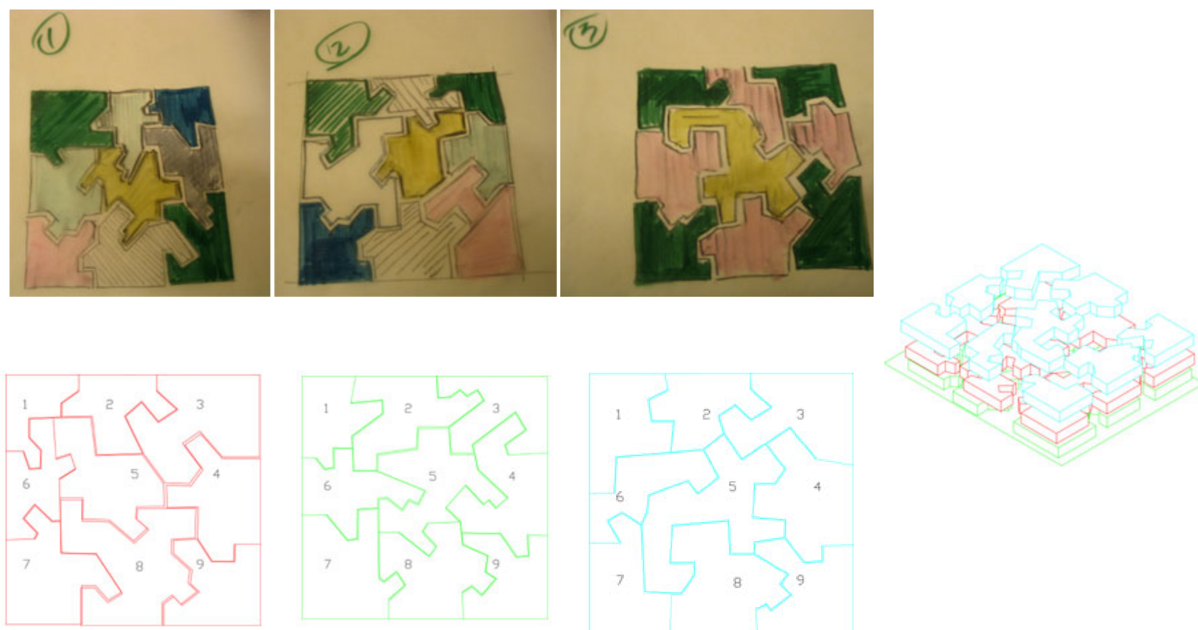


Figure 3.1 *Puzzle-pieces for interlocking units*
A study of puzzle-like pieces for creating interlocking units. Study was developed from sketches, computer models, and fabricated models

3.2 Results using a less systematic design approach

The solutions did not offer a universal system as the configuration was too random to be used for deconstructing a collection of other forms. The solutions were prescribed to the form which they solved with no real logic to the configuration. The aim was to find a more generic approach for solving the design problem.

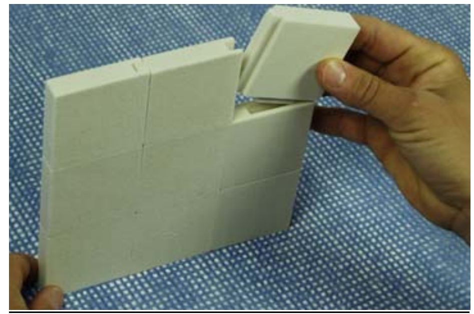


Figure 3.4 *Fabrication testing*
Physical test of design language that was used to deconstruct the form. Test included feasibility, functionality, and simplicity for computation

3.3 Creating a systematic design language

The other approach was modeling a description in a 3Dimensional environment that would offer a generic logic for deconstructing any form. The solution took into consideration "tabs" (extrusions) and "notches" (subtractions) that would connect together to assemble the forms. The tabs and notches served as the integral geometry for assembly. Experiments were conducted to test the connectivity, the stability, and tolerance within the system (figure 3.4).

3.4 Results using a systematic design approach

The tests proved to address the integral connection desired as well as offered flexibility in tolerance variances. The solution gave nine different typologies (figure 3.5) that were repeated based on location of unit and resolution of units per form:

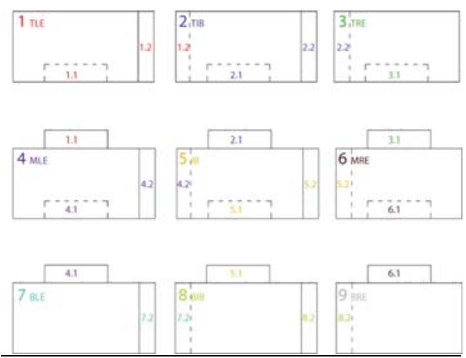


Figure 3.5 *Design language*
Nine different typologies from puzzle studies

1. **TLE**: Top Left End

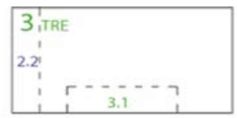
Units that are top most left

2. **TIB**: Top In-Between

Units that are positioned between TLE and TRE

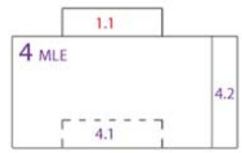
3. **TRE**: Top Right End

Units that are top most right



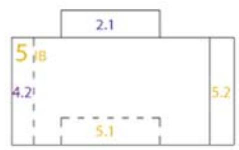
4. **MLE**: Middle Left End

Units that are positioned between TLE and BLE from top to bottom



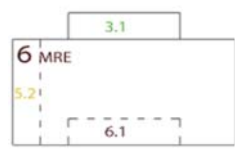
5. **IB**: In- Between

All units that are positioned between TLE, TIB, TRE, MLE, MRE, BLE, BRE, and BIB



6. **MRE**: Middle Right End

Units that are positioned between TRE and BRE from top to bottom



7. **BLE**: Bottom Left End

Units that are bottom most left

8. **BIB**: Bottom In-Between

Units that are positioned between BLE and BRE

9. **BRE**: Bottom Right End

Units that are bottom most right

3.5 Mapping of digital fabrication technologies

The exploration done in this thesis used ZCorp ZPrinting technologies. The future exploration of this work would investigate other methods of fabrication using different types of materials that would accept the geometrical data presented. As seen in the direct translation of desktop fabrication methods such as lasercutting technologies mapping to CNC technologies (Botha M, Sass L, 2006) (figure 3.6), the goal is for future processes to emulate this process with the use of concrete printing technologies. The use of ZCorp printing helped in refining the problem within a controllable environment.

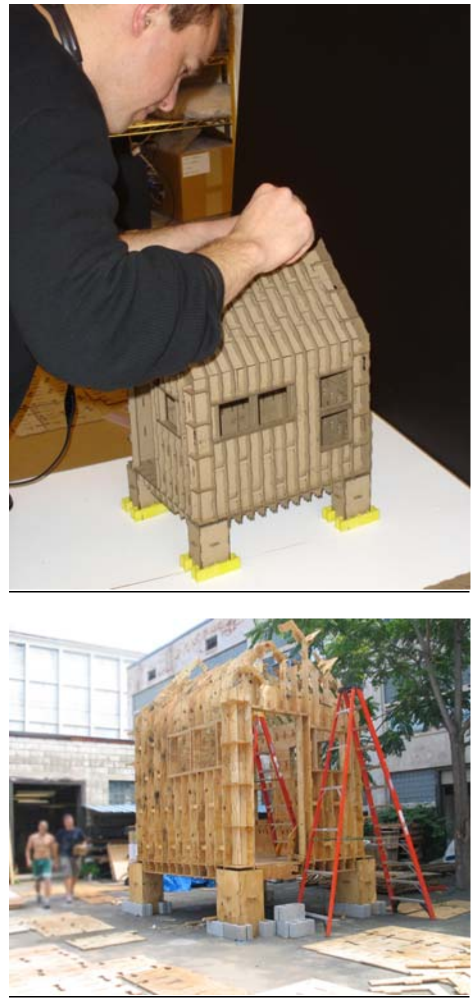


Figure 3.6 *Fabrication mapping: model→ life-size*
The direct mapping of computer information that drives both desktop fabrication devices (lasercutter) as well as life-size devices (CNC)

3.3 STANDARD WALL SYSTEM (orthogonal)

Constructing orthogonal shaped walls with non-uniform masonry units

Standard wall system denotes orthogonal wall system that posses no complex curvature (figure 3.7). This initial approach used the design description of the units for creating the division algorithm used for building the units of assembly

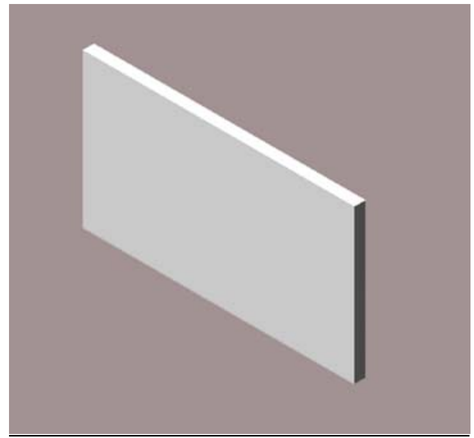


Figure 3.7 *Computer model of orthogonal form*
A computer model of an orthogonal wall. The algorithm creates units for building walls of this type

3.3.1 How the algorithm works

From a single straight line, the user defines the number of vertical and horizontal components desired for constructing the wall. The integer input is used as parameters within a divide function to inform the system of the number of unit columns required from the initial curve.

3.3.2 Construct divisions and array of information

After divisions, a copy translation of the line and points is done to account for thickness of wall. The copies are mapped to a database in the form of arrays for keeping track of all of the geometric information for future operations. The array makes it possible for iteration allowing selection of points for making curves from N point of one array to N point of the array containing the parallel curves.

3.1.3 Construct initial surfaces of each unit

The surface is created for constructing the faces of the units. Iteration is done through the arrays selecting
N curve, N+1 curve, initial curve, translated curve (figure 3.8).
With these boundaries selected, a surface is created based on the boundary that is formed. The surface takes its form from an interpolation between curves. After the creation of each surface, the surface was inserted into an array for future operations.

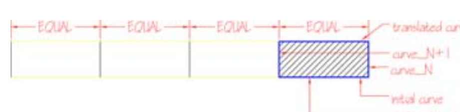


Figure 3.8 *Creating surface from boundaries*
Method algorithm uses for creating a simple surface

3.3.4 *Construct initial unit block*

An iteration was performed through the surface array accessing each surface performing an extrusion that defined the height of the unit. This height is a parameter obtained as an input from the user. The extrusion created a new object that needed to be tracked, therefore was inserted into an array for future operations.

3.3.5 *Construct side tabs for each unit*

Each tab is constructed from a ratio of the thickness of each unit.

Tab:unit thickness ratio → 1:2 (figure 3.9)

In order to get the proportional value of the ratio each curve is determined by the calculation.

Once division of curves was created the points were joined with a new curve to create a boundary. This boundary acted as the perimeter for creating the tab surface for extrusion.

3.3.6 *Construct top tabs for each unit*

Using the same ratio principle each curve was divided into 8 for defining width, height, and thickness of top tabs. Boundaries were constructed for each surface by connecting

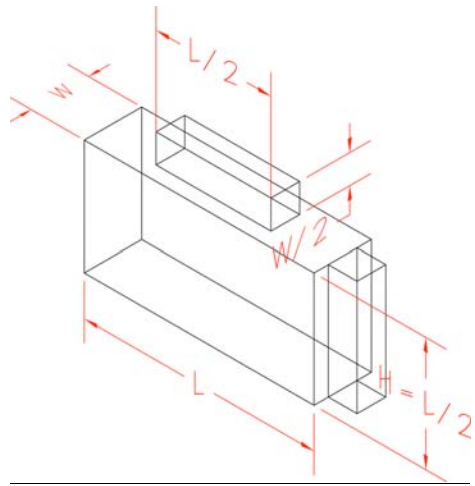
[Array1]N(point)= 2 to [Array2]N(point) = 2

and

[Array1]N(point)= 6 to [Array2]N(point)=6 (figure 3.10) of parallel lines. Surfaces were constructed using curves as edges and inserting each surface in an array for future operations. From surface array, tabs were constructed via extrusion that took the initial surface as input and defines a volume for the depth of the tab.

3.3.7 *Construct notches for Boolean operations*

Both sides and top tabs were copied to original location saving copies to different arrays. The copies were accessed and scaled



L : length of curve along form
W : width of curve along form
T_w : Tab width
T_L : Tab length
N_w : Notch width
N_L : Notch length
T_m : Tolerance determined by the material
For each L or W $\Rightarrow$ L/8 or W/8
This divides L and W into 8 equal pieces thus giving an array of 9 points
$\therefore$ L/8 = 9i $\Rightarrow$ 8 equal divisions along the curve
i : steps or index in array starting at i = 0
$\rightarrow$ i = 0, 1, 2 …….8
$\therefore L/8 = d[i \rightarrow i+1]$ such that i = 8
$d_{(i+6)} - d_{(i+2)} = L/2$
$\therefore$ L/2 = ½ length of curve
$T_L = L/2$
AND
$W/8 = d[i \rightarrow i+1]$ such that i = 8
$d_{(i+6)} - d_{(i+2)} = W/2$
$\therefore$ W/2 = ½ width of curve
$T_w = W/2$

T_m is introduced for notches
$N_L = T_L + 2(T_m)$
AND
$N_w = T_w + 2(T_m)$

Figure 3.9 *Calculation of unit sizes*
Equation used for computing sizes for notches and tabs of each unit based on a proportional ratio

by a percentage which was a calculation of the tolerance needed for fabrication. A tolerance variable is important within the program for parameter manipulation according to the printing device used. The ZPrinter used was a 300dpi resolution ZCorp printer, therefore printed an excess of material (0.0154") on the perimeter of the units giving a thicker tab that needed to be calculated in the equation as tolerance. Other ZCorp machines such as the Spectrum Z510 System offer a resolution of 600 x 540 dpi which will reduce the tolerance because of the added printing accuracy. The tolerance variable will have additional variance with contour crafting technologies. Tolerance can therefore be manipulated to produce the individual units accurately to make assembly a more efficient process.

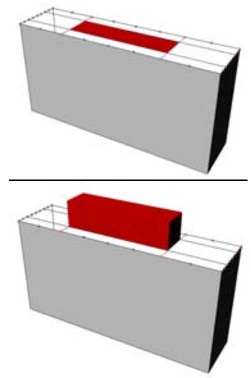


Figure 3.10 *Creating tab of unit*
Process algorithm uses for creating tabs on top of units

Boolean operations are very important for combining objects using a Constructive Solid Geometry (CSG) principle. CSG methods find the intersection of shapes for the purpose of combining into one or subtracting one from the other. Due to the nature of this thesis, the concentration is based on the integral connections within the system. This was done by creating notches (slots) for both side tabs and top tabs.

Both side tabs and top tabs were combined with the initial block to construct an assembly unit (figure 3.11). This was done using a union principle approach. The newly created objects were then inserted back into the respective arrays. Boolean operations create new objects based on the intersection of the originals. In order to keep an updated list of the units, it was necessary to refresh the array with the appropriate objects to keep a global array of each unit after each operation.

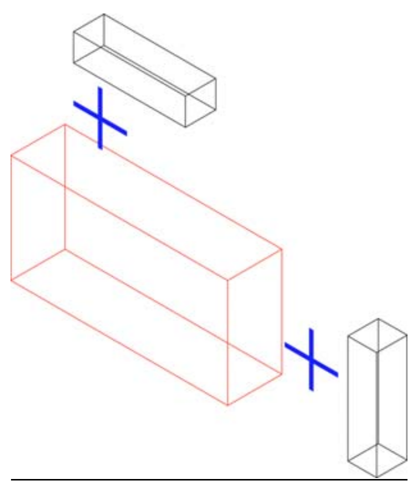


Figure 3.11 *Boolean operations (CSG)*
Boolean operation of solid geometry (CSG) for creating a unit

3.3.8 *Fabricate each discrete piece using ZPrinter printing technology*

The purpose behind creating algorithms within this thesis is for deconstructing complex form to facilitate in the production of models in a physical environment. The physical model gives the designer a tangible product for evaluating. Each individual piece was a solid piece of geometry that was fabricated using ZCorp printing and in extended research, advanced methods such as concrete fabrication. The digital information is transferred to printing devices to produce study models that work up to the final model. The benefits of having a program that can efficiently construct individual components for producing form, makes the process less monotonous in the case of designer having to rebuild the form because of tolerance issues or complications with the components. The computed information addressed the possibility of scaling. Designer is given the opportunity to re-execute the program changing the necessary variables or parameters that apply to a more feasible and acceptable solution.

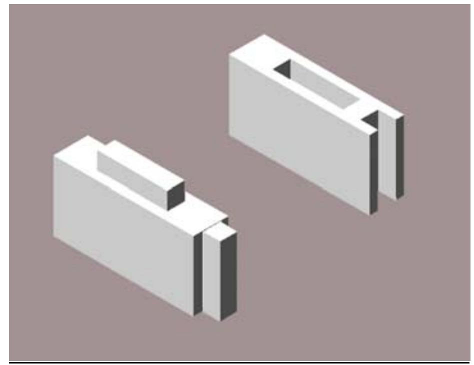


Figure 3.12 *Unit after boolean operation*
Two units after Boolean operations

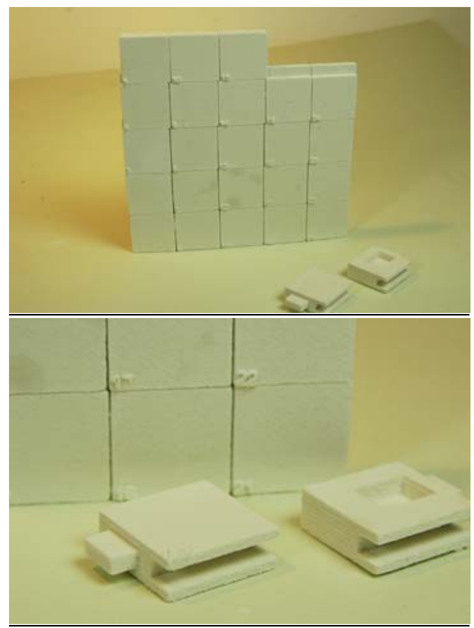


Figure 3.13 *Fabricated Orthogonal wall*
Complete model of wall printed and assembled based on labeling instructions embedded in each unit

3.3.9 *Results of STANDARD WALL SYSTEM*

The approach was the initial approach of the research for testing the design description of the units using the algorithms without human intervention for modeling (figure 3.12). The tests mapped directly to the hand- modeled versions of the walls (figure 3.13) which proved that the algorithm worked for fabricating the assembly units. The setback that led to the other stage of the research was the way in which the algorithm was structured. The algorithm was not generic enough to handle a wider variation of planar shapes. It lacked the embedded intelligence of dividing the shapes according to its topology and accepted every input shape as an orthogonal object.

3.4 NON-STANDARD WALL SYSTEM (non-orthogonal)

Constructing quadrilateral shaped walls with non-uniform masonry units

This program is based on non-uniform masonry units that can be used to fill arbitrary quadrilateral voids. Most often, masonry units are used to construct orthogonal forms because of the set orthogonal dimensions of the units, i.e. 4" x 8", 8" x 12", etc. The introduction of new methods of fabricating instigates this research for constructing non-orthogonal shapes from individual, non-uniform masonry units. The aim is to fabricate the individual components and assembling to create non-orthogonal form.

3.4.1 How the algorithm works

The algorithm starts with user defining a voided space with four curves which is used as input for construction. The voided space can be strictly orthogonal or less orthogonal leading to quadrilateral spaces (figure 3.14).

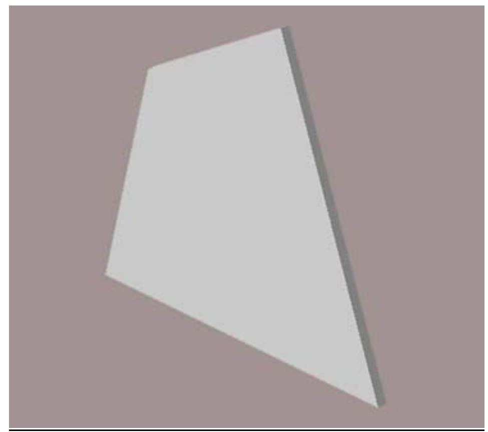


Figure 3.14 *Computer model of non-orthogonal wall*
Computer model of a non-orthogonal wall. The algorithm creates non-uniform units for building walls of this type

3.4.2 Divide sides of void

(figure 3.15 thru 3.24 on page 37)

According to user's input, the integer is read into the system for the purpose of dividing the sides of the void. The division is based on how many discrete units the user wishes to have *length* ***X*** *width* to fill the quadrilateral void. Divisions were made of each edge by using a division function that takes the curve as input and outputs points of division. The division points (figure 3.15) were saved in respect to the curve it divides.

3.4.3 Create outline grid for units

By using array data structures, the objects are stored for future access making it easier to perform any amount of required operations, updating the database with the current object each time an operation is executed. Having saved the points in their respective data structure, an iteration algorithm accesses the database connecting [Array1]N(index) to [Array2]N(index) (figure 3.16) of parallel curves for creating a cross grid. The cross grid creates an outline for the placement of the units. The new objects are stored in new data structures for further operations.

3.4.4 Find intersection of lines for surface boundaries

The intersection of the grid lines dictates the boundary of each unit (figure 3.17). Each line within the grid intersects their perpendicular counterpart. The intersection of each line constructs a boundary used for forming a surface for each cell within the grid. Each surface is the primary face of the unit which is placed in a data structure (figure 3.18).

3.4.5 Construct the thickness of unit

Each surface can be accessed in the constructed database for creating the volume for each unit, by iterating through the array. As each surface is selected within the array, an extrusion is performed to give volume to each unit that will be used to construct the final form. The volume thickness is also specified by the user based on structural requirements or aesthetics once the system is fully fabricated and assembled (figure 3.19).

3.4.6 Create tabs and notches

As a result of each unit being its unique component of the form, it is necessary to construct a methodology of connection between each unit; this methodology it done with tabs and notches to conform to the design description. Tabs are meant to slip into grooves of adjoining units for creating stability within the structure, which creates an integral continuity in the system. By doing a ratio division calculation, each tab is created proportional to the size of the unit, which creates a platform for recursive and systematic execution of code on each unit. Each unit is generically approached based on its simple geometric proportions and relations to each other, the tabs, and the grooves. The tabs are constructed on the sides and tops (figure 3.20) of each unit for union operations. Scaled replications of the tabs are inserted into the bottom of above units, and inserted into the side of the adjoining unit. The scaled tabs are constructed for the purpose of extraction (figure 3.21) from the adjoining units for generating the integral connection within the system.

3.4.7 Boolean operations

In order to construct the embedded connection within the units, two different types of Boolean operations are necessary one for the union of objects, and the other for subtracting objects from other objects. Each non-scaled side and top tab is combined with the constructed volume for forming the unit structure (figure 3.22). The union operation updates the current geometry, which is reinserted in the unit array for maintaining addresses of the geometry within the system. The array is accessed accordingly to extract the necessary tabs for constructing the notches within the units. Each N unit should have a side notch (groove) up to N-1 unit. From N+1 to N row of units should have a notch (groove) in the bottom that connects to the row beneath.

3.4.8 Weight factor of units

Masonry units are usually laborious in nature to construct structures. As a result, it makes it a monotonous and tedious challenge, with longer construction times, considering the laborer's health and motivation after a complete day of assembly. The weight has a great contributing factor, therefore can be lessened by subtracting unnecessary geometry from the units (figure 3.23). Portions of the units are extracted using a subtraction/difference method. This portion is also based on a size ratio [Lenth/4] that can be executed with an equation on each unit for extracting the proportional amount.

3.4.9 Fabrication and assembly of units

The non-uniformity of the units within the system makes the fabrication processes less conventional, as one unit can not be swapped for another; each unit is unique in dimension. To address this issue, a labeling system was devised to identify each piece and to ease the labor and assembly process of the structure. The number system is based on the index of each unit within the data structure. As each unit is inserted into the data structure, it is identified with an index (i=0,1,2,…) number that creates the simplicity of iteration and numbering each unit (figure 3.24). Each unit is inscribed with the correct number in the form of a text object which is combined with the unit. This eases the fabrication and identification of pieces in the event of misprint or fracture of a unit. Once printed, any complications with the system, fabricator/installer can specify in detail the exact piece that is required, rather than having to remeasure and recalculate the gap within the structure.

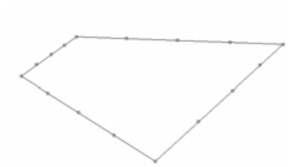


Figure 3.15 *Division of curves*

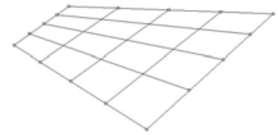


Figure 3.16 *Connection of division points*

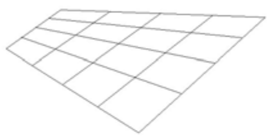


Figure 3.17 *Grid outline of brick units*

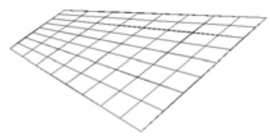


Figure 3.18 *Surfacing of grid*

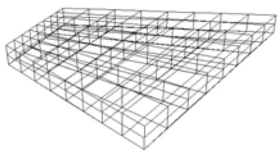


Figure 3.19 *Extrusion of surfaces*

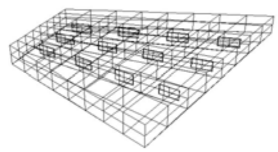


Figure 3.20 *Surfaces for top tabs and notches*

Figure 3.21 *Extrusion of tab and side tabs*

Figure 3.22 *Boolean operation of units*

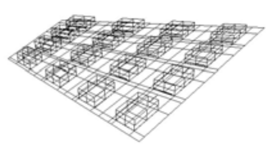
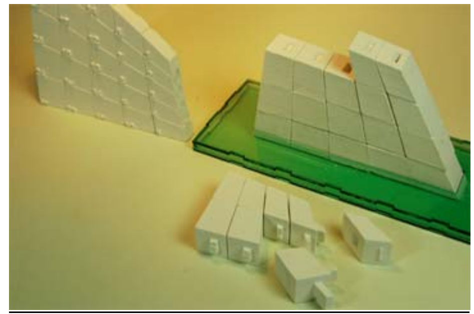


Figure 3.23 *Extract units for weight*

Figure 3.24 *Labeling of units*

3.4.11 Results of NON-STANDARD WALL SYSTEM

The non-standard wall system was an ideal environment for experimenting and using the design description for creating individually different units. This approach tested the ratio calculation that allowed for the integral connection computation of the adjoining units (figure 3.25). The connections were proportional to the adjoining units. Although the algorithm accepted a larger range of surfaces, they were still restricted to planar geometry. The research wanted to utilize the division qualities of the algorithm to deconstruct a greater scale of complex-curved geometry which led to the development of the following algorithm. The progression of algorithms became more generic as they began to accept both standard and non-standard wall systems as input.



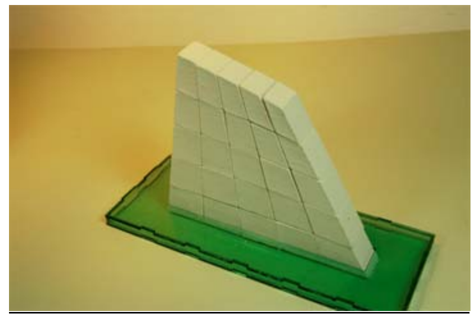


Figure 3.25 *Fabricated Non-orthogonal wall*
Complete printed and assembled non-orthogonal wall system

3.5 COMPLEX-CURVED WALL SYSTEM

Constructing curved walls from discrete connected components

Conventional construction of masonry units entails using standard dimensioned units that are bonded together using a concrete mix. In some instances to obtain the curved nature of a masonry wall, smaller units would have to be used to increase the resolution in order to give a more curved form. A programming approach constructs a complex-curved masonry wall (figure 3.26) with user specified unit sizes, which allows freedom and different aesthetic value determined by the size of units used. The architect is therefore not limited to one size that constructs the required curve, but has a plethora of options that would satisfy their preference.

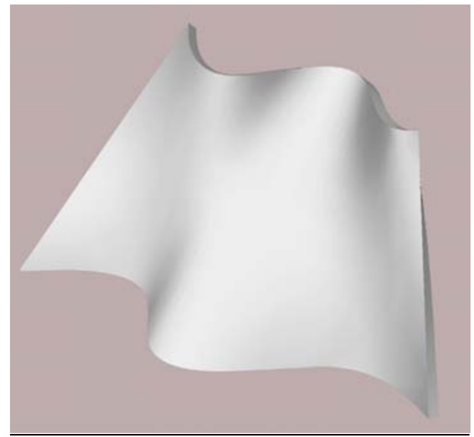


Figure 3.26 *Computer model of complex wall*
Computer model of complex-curved wall. The algorithm creates non-uniform units to build complex walls of this type

Curves are more and more being computed by current architects as they use more computational tools for design processes. The problem lies in the fabrication solution of the curves which are computed for physical rationalization. The aim is not to extract the information from the computed environment to be solved in a physical environment, but rather solve the fabrication methods within a purely computational and digital framework.

3.5.1 How the algorithm works

The algorithm for this program is based on complex-curved geometry that introduces more complexity than the previously mentioned.

3.5.2 Divide initial curve for required number of units

From two independent curves drawn by user, the system prompts to specify how many units are required to construct the curved wall. The entered integer is stored as a variable and transferred to a function for dividing the curve respectively. The division is performed while storing the division points into an array for bookkeeping.

3.5.3 Construct boundary curves for units

In order to create initial surfaces to construct units, proper boundaries should be achieved. The boundaries are the shared edges of units. Boundary curves are formed by iterating through array of points and connecting N(point) of curve one extracting each isocurve that corresponds with the point on the surface. The same method is executed in the other direction that constructs the grid of the surface. Each surface is then constructed using the shared edges as input to create surfaces that are allocated to arrays (figure 3.27).

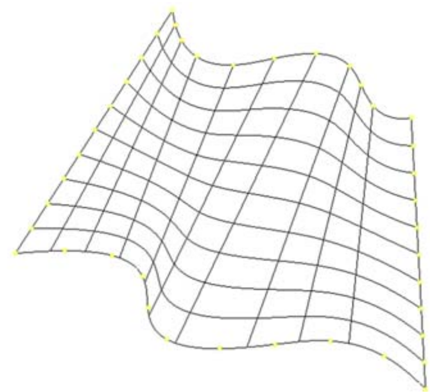


Figure 3.27 *Grid of complex wall*
Algorithm creates grid that outlines each unit

3.5.4 Construct initial unit volume

To create the initial volume for the units, each surface is extruded normal (figure 3.28) to the surface which follows the direction of the curvature. The thickness is given by the user. The thickness of each unit follows the direction and curvature of initial curves to retain the designer's intent. As each volume is created it is inserted into a unit array for keeping track of for future operations.

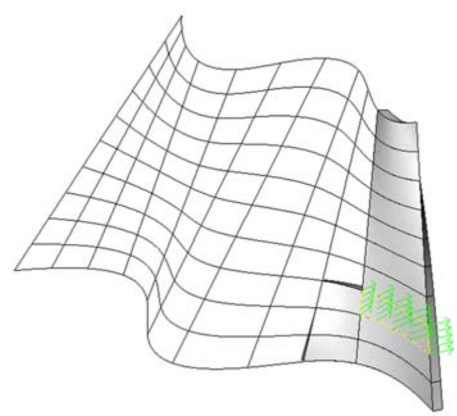


Figure 3.28 *Extrusion of each surface*
Each unit's surface is created from the boundaries of the grid. An extrusion is created based on the surface's normals

3.5.5 Create tabs and notches

Similar to planar structures, tabs and notches should follow the curvature of the initial curve to attribute to the complex geometrical shape that is required within the form. The integral connections created from notches and tabs allow the continuity of the shaped geometry as they align with the parent surface structure. The connections are created from a ratio of the height of the wall units (figure 3.9), which makes it flexible for creating wall systems of various heights and shapes.

3.5.6 Fabrication and assembly of units

Each unit is fabricated with embedded connectors that attach to the adjoining units for building the final curved structure.

Depending on the size of the unit and the size of the printing bed and model size, each unit can be printed separately if needed. This system of construction and assembly provides a systematic approach to assembly. Each piece is labeled according to their position within the system. This labeling allows fabricator to assemble each piece based on a numbering pattern.

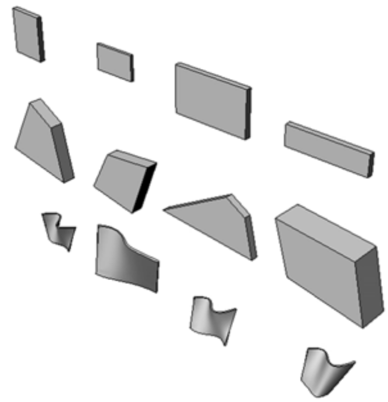


Figure 3.29 *Array of accepted wall types*
An array of wall types accepted by algorithm

3.5.7 Results of COMPLEX-CURVED WALL SYSTEM

The algorithm addressed a vast array of different complex surfaces. As input it can accept orthogonal, non-orthogonal, and complex-curved walls (figure 3.29). The solution tested the durability of the design description logic as rules for dividing the complex surface. The tested assembled models showed that the design description was generic enough for the different systems (figure 3.30).

The algorithm as this stage was still responsible for building the wall geometry before analysis and deconstruction which was not likely to occur in the design process. The prediction of this system was that input geometry will be a constructed wall design which would be evaluated and deconstructed into its respective units. The research continued with the exploration of form design (semi-enclosed complex surfaces) and creating an algorithm for creating the units of assembly.

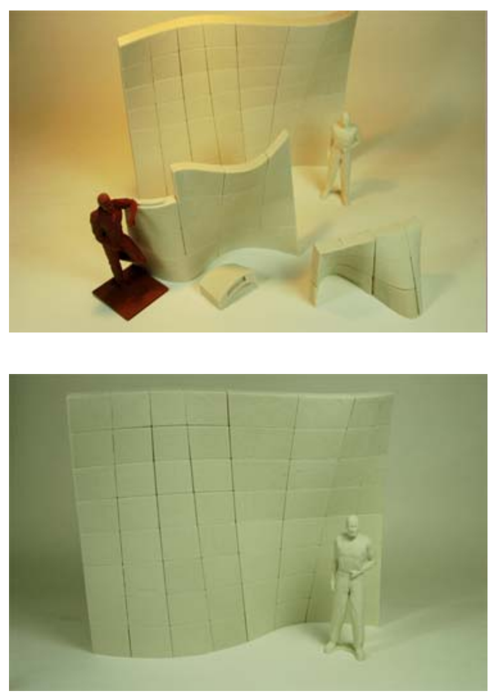


Figure 3.30 *Fabricated Complex wall*
Printed and assembled curved wall system

3.6 COMPLEX-CURVED WALL SYSTEM (semi-enclosed form)

The algorithms discussed tested the feasibility of the design language for deconstruction which needed to be applied in a more generic setting. The following addresses a more generic approach. For this stage of the research, several different semi-enclosed forms were designed as inhabitable structures. These forms were influenced and constrained as transitional, functioning, geometrical spaces. The forms had to exhibit entrances, light, interior space, height, and meet the overall footprint constrain (figure 3.31).

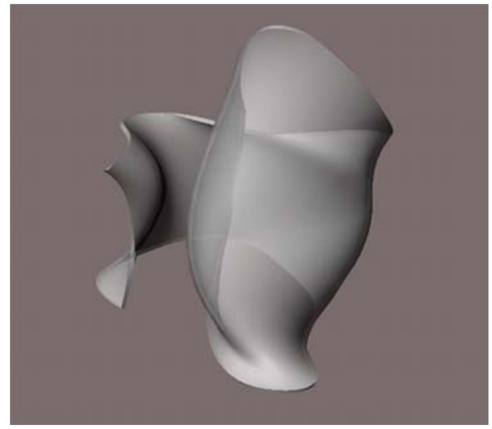


Figure 3.31 *Computer model of Semi-closed form*
Computer model of a semi-closed form. The semi-closure increases the complexity of the curvature

This solution offers a more detail explanation of the complete process and the functions that played the most important roles in the algorithm.

AVAILABLE FUNCTIONS

MakeUnits()
getUserInput()
getGeometry(originalSrf,rows, cols)
createUvPoints(rows,columns,surface)
createIsoCurves(srf, pts, rows, cols)
splitIsoCurves(srf, crvs,ptsArray)
splitSurface(srf,horCrvs,vertCrvs,rows,cols)
createBottomSurfaces(innerSrfs, outerSrfs, horOrig, horOffset, rows, cols)
bottomSrfHelper(bot, arr, rows, cols)
createTopSurfaces(innerSrfs, outerSrfs, horOrig, horOffset, rows, cols)
topSrfHelper(top, arr, rows, cols)
createLeftSurfaces(origSrf,origOffsetSrf,vertOrig, vertOffset, rows, cols)
createRightSurfaces(origSrf,origOffsetSrf,vertOrig, vertOffset, rows, cols)
bondUnits(inner,top,left,bottom,outer,right,rows,cols)

GLOBAL VARIABLE DEFINITIONS

TOLERANCE- variable stores the tolerance amount necessary based on the fabrication device used

FLIP_FLAG- a boolean variable to test the direction in which the surface was constructed for correct offset of surface (outwards)

THICKNESS- variable stores the thickness required of units within the system

originalSrfPoints, offsetSrfPoints- variables store the collection of surface points

FUNCTIONS

3.6 .1 MakeUnits()

Function instigates process by delegating *getUserInput()* function for getting integer input

3.6.2 getUserInput()

Function accepts input from user to apply to the geometry. Input includes the thickness of units, the number of grid columns, and the number of grid rows to divide surface. Input is passed onto *getGeometry(originalSrf,rows, cols)* for creating the necessary geometry

3.6.3 getGeometry(originalSrf,rows, cols)

Function takes three parameters as input.

originalSrf: this variable holds the original surface generated by the user. This surface is copied and translated outwards to give thickness of shell by creating an offset surface (figure 3.32).

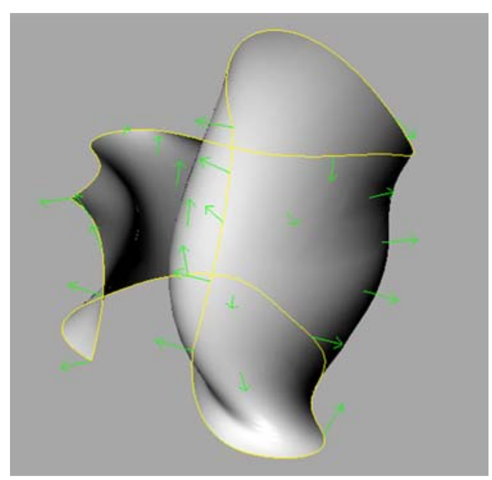


Figure 3.32 *Thickness of units*
Thickness is created based on user's input

rows: this variable holds an integer which defines the number of rows needed. This variable propagates throughout code. It is first used for the extraction of points on the surface, then the extraction of isocurves that follow the surface, and is ultimately used for allocating space in arrays for storing the units and helper geometry

cols: this variable holds an integer which defines the number of columns needed . This variable propagates throughout code. It is first used for the extraction of points on the surface, then the extraction of isocurves that follow the surface, and is ultimately used for allocating space in arrays for storing the units and helper geometry

3.6.4 createUvPoints(rows,columns,surface)

Function takes three parameters/variables as input.

surface: surface to be divided

rows: this variable is used to walk along the surface in the U direction for creating points up to number of *rows*

columns: this variable is used to walk along the surface in the V direction for creating points up to number of *columns*

rows and ***columns*** are also used for creating 2D arrays for storing all the points on the surface: ***Array(rows, columns)*** (figure 3.33).

3.6.5 createIsoCurves(srf, pts, rows, cols)

Function takes four parameters as input.

srf: surface to be evaluated

pts: 2D array of points used to for extracting isocurves from surface

rows: the iteration limit of i

columns: the iteration limit of j

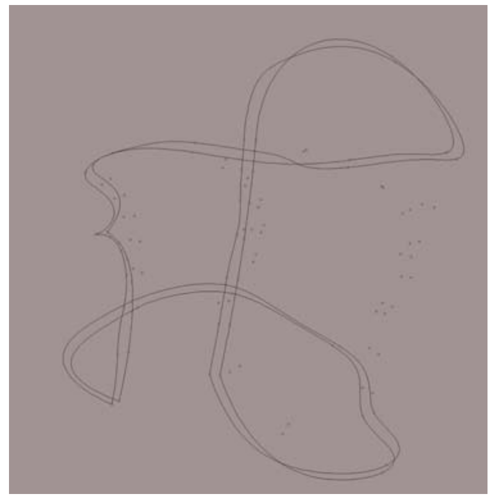


Figure 3.33 *Creation of points on form*
Points are created on form for dividing rows and columns

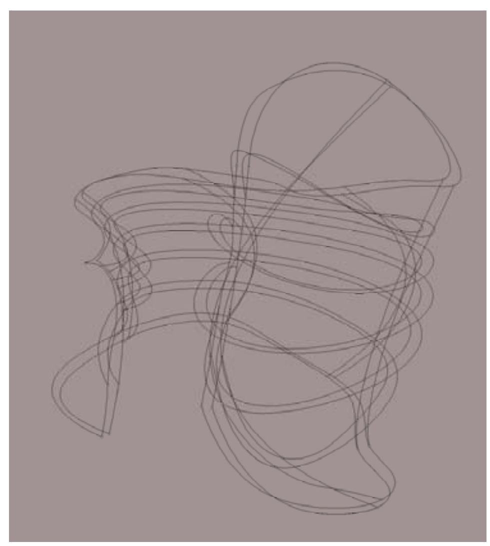


Figure 3.34 *Creation of grid*
Isocurves are extracted from the form for creating a grid that defines each unit

for i = 0 to ***rows***

for j = 0 to ***columns***

extract isocurves from surface.........

next

next
(figure 3.34)

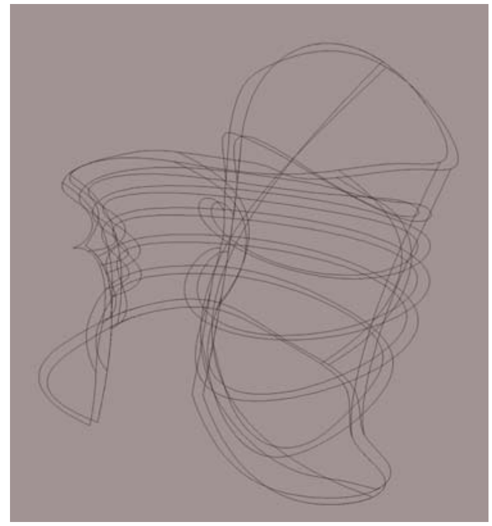


Figure 3.35 *Creation of unit boundaries*
Isocurves are split to create each separate unit. Curves are used for creating unit surface

3.6.6 splitIsoCurves(srf, crvs,ptsArray)

Function takes three parameters as input

srf: surface to be evaluated

crvs: isocurves extracted from surface

ptsArray: array of points on surface used for splitting curves that defining the boundary of the units

(figure 3.35)

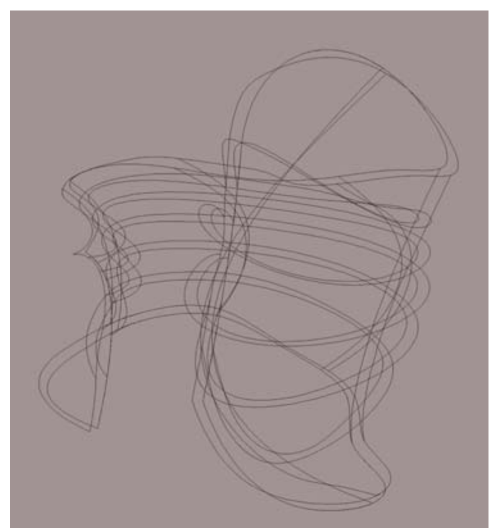


Figure 3.36 *Creation of unit surfaces*
Isocurves are used to slice surface into respective units

3.6.7 splitSurface(srf,horCrvs,vertCrvs,rows,cols)

Function takes four parameters as input

srf: surface to be split

horCrvs: horizontal curve array used to split surface

vertCrvs: vertical curve array used to split surface

rows: the iteration limit of i

cols*:* the iteration limit of j

(figure 3.36)

3.6.8 createBottomSurfaces(innerSrfs, outerSrfs, horOrig, horOffset, rows, cols)

Function takes six parameters as input and creates a bottom surface

innerSrfs: an array of all the split surfaces of the original surface that define the inner side of the units

outerSrfs: an array of all the split surfaces of the offset surface that define the outer side of the units

horOrig: an array of all the horizontal curves from the original surface that define the boundary of the units to be constructed

horOffset: an array of all the horizontal curves from the offset surface that define the boundary of the units to be constructed

rows: the iteration limit of i

columns: the iteration limit of j

(figure 3.37)

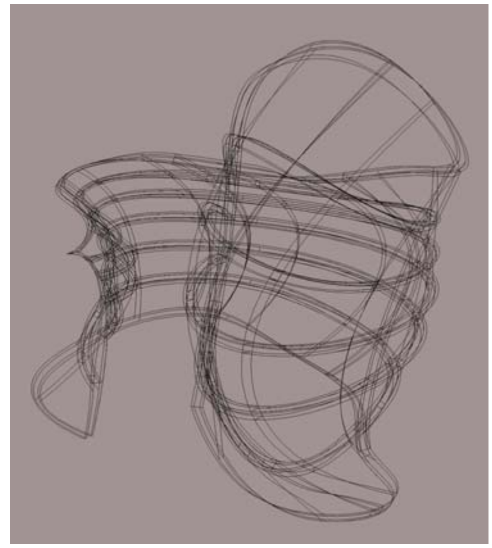


Figure 3.37 *Bottom of unit*
Bottom surfaces are created based on each unit's outline

3.6.9 bottomSrfHelper(bot, arr, rows, cols)

Function takes four parameters as input and creates a notch as part of the bottom surface

bot: the newly created bottom surface used for creating notch

arr: an array of curves that define location of notch

rows: the iteration limit of i

columns: the iteration limit of j

(figure 3.38)

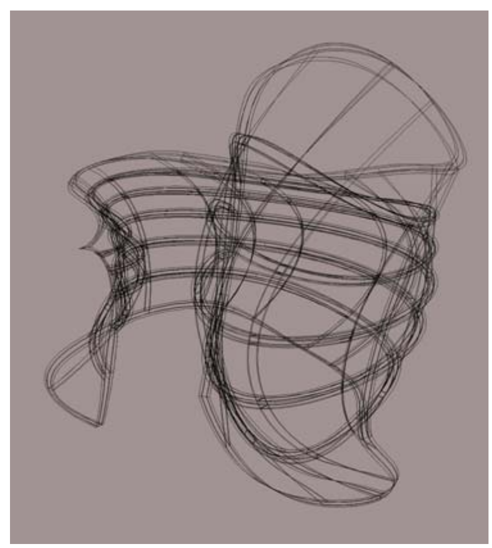


Figure 3.38 *Notches and tabs at bottom of unit*
Each bottom surface is manipulated based on it's location in the system. If the surface is at the very bottom of the form, no manipulation is necessary. If it is above, a tab needs to be created

3.6.10 createTopSurfaces(innerSrfs, outerSrfs, horOrig, horOffset, rows, cols)

Function takes six parameters as input and creates a top surface

innerSrfs: an array of all the split surfaces of the original surface that define the inner side of the units

outerSrfs: an array of all the split surfaces of the offset surface that define the outer side of the units

horOrig: an array of all the horizontal curves from the original surface that define the boundary of the units to be constructed

horOffset: an array of all the horizontal curves from the offset surface that define the boundary of the units to be constructed

rows: the iteration limit of i

cols: the iteration limit of j

(figure 3.39)

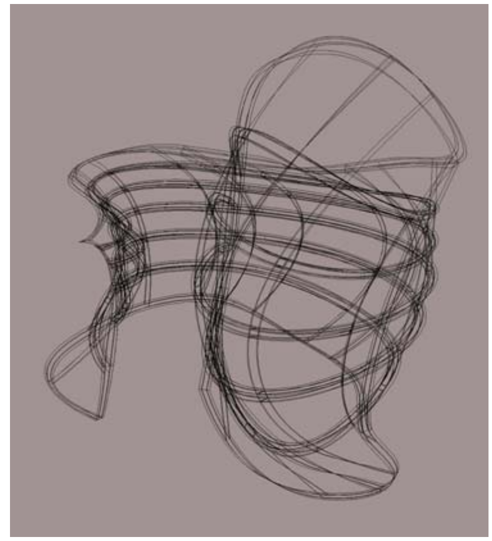


Figure 3.39 *Top of unit*
Top surfaces are created based on outline of units

3.6.11 topSrfHelper(bot, arr, rows, cols)

Function takes four parameters as input and creates a groove as part of the top surface

bot: the newly created bottom surface used for creating groove

arr: an array of curves that define location of groove

rows: the iteration limit of i

cols: the iteration limit of j

3.6.12 createLeftSurfaces(origSrf,origOffsetSrf,vertOrig, vertOffset, rows, cols)

Function takes six parameters as input for creating left side of unit

origSrf*:* the original surfaces

origOffsetSrf*:* the offset surfaces created

vertOrig*:* vertical interior curves that define left surface

vertOffset*:* vertical exterior curves that define left surface

rows*:* the iteration limit of i

cols*:* the iteration limit of j

(figure 3.40)

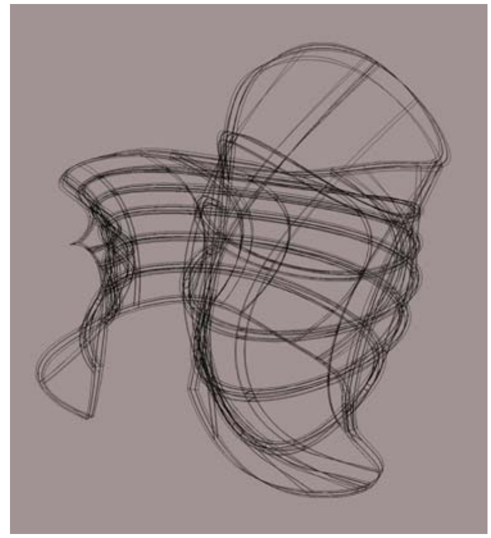


Figure 3.40 *Left of unit*
Left surfaces are created based on left outline of unit

3.6.13 createRightSurfaces(origSrf,origOffsetSrf,vertOrig, vertOffset, rows, cols)

Function takes six parameters as input for creating right side of unit

origSrf*:* the original surfaces

origOffsetSrf*:* the offset surfaces created

vertOrig*:* vertical interior curves that define right surface

vertOffset*:* vertical exterior curves that define right surface

rows*:* the iteration limit of i

cols*:* the iteration limit of j

(figure 3.41)

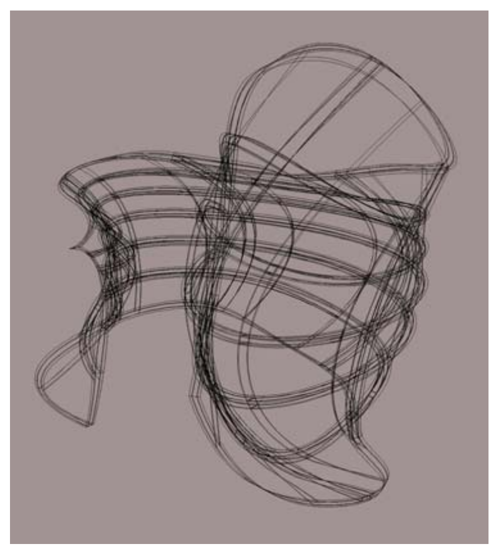


Figure 3.41 *Right of unit*
Right surfaces are created based on right outline of unit

3.6.14

bondUnits(inner,top,left,bottom,outer,right,rows,cols)

Function takes eight parameters for binding units

inner: inner portion of unit

top: top portion of unit

left: left portion of unit

bottom: bottom portion of unit

outer: outer portion of unit

right: right portion of unit

rows: the iteration limit of i

cols: the iteration limit of j

(figure 3.42)

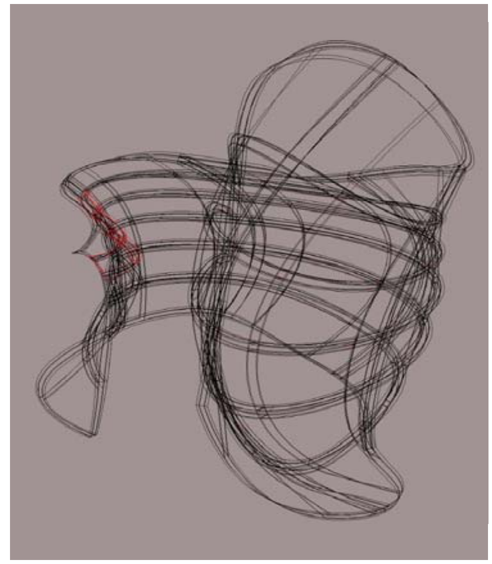


Figure 3.42 *Bind each unit*
All geometry is bound together to create each unit

3.6.15 Results of COMPLEX-CURVED WALL SYSTEM (semi-enclosed form)

The algorithm accepted a wider range of forms and shapes. It tested the design description (figure 3.43) for deconstructing the form into discrete units for a more complex type of geometry. Each piece was printed individually and assembled to give a range of model sizes for evaluation.

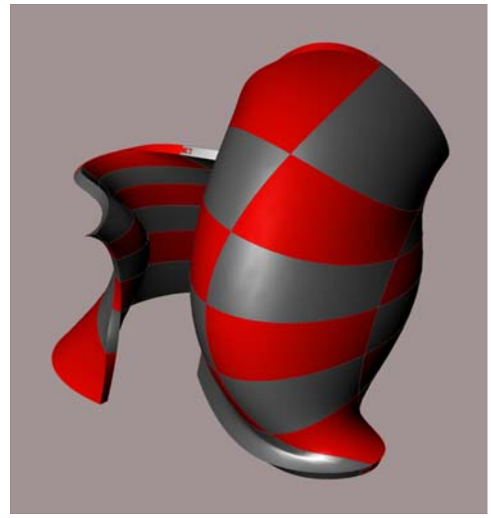


Figure 3.43 *Complete deconstructed form*
Color coded computer model displays the unique properties of each unit. Each unit is printed using the ZCorp ZPrinter

Figure 3.44 *Fabricated semi-enclosed model (~2'-0" in height)*

Figure 3.45 *Scaled models – [(left to right) 1'-0", 0'-8", 0'-2"]*

Chapter 4

4.0 Results and Limitations

The programming methods discussed for fabricating models within this thesis not only contributed to the use of software within design computing, but creates viable information about the geometry and methods used in the design processes by architects.

The research carried out in this thesis demonstrated the ability to deconstruct complex-curved form into smaller manageable units. This thesis addressed the issue of devising a generic design language that can be used with different types of form. The novel approach undertaken in this thesis dealt with the successful manipulation of geometry for constructing integral connections between units of assembly and producing physical models bigger than the fabrication device.

Other research observations and discoveries included mapping the computed data to other types of fabrication technologies. This observation is relevant to future exploration for dealing with different forms of 3D fabrication as well as material composition. This research defined a generic process for an array of forms rather than a specific solution for each form.

4.1 Manageable assembly

Computed models can create great complexity for fabrication. The difficulty is present when trying to negotiate with method and material for producing a physical object that would represent the intent of the design. The results of this thesis introduced a method for creating manageable units for systematically assembly in the production of the design form.

The plaster-like material used in the ZPrinter for this research can sometimes be brittle which may create fractures in the representation model. The imperfections can be limited by creating smaller parts as well as creating an environment for quicker reproduction of a unit rather than the entire model. The research showed how smaller units created a more controllable environment for constructing the design models.

4.2 Integral connections in the assembly

The modeling and production of geometry for fabrication is a very difficult problem because of the layered process of architect communicating an idea and process to the computational tools. It is not natural to model and compute certain forms of geometry without the intervention of the computing power and calculation. The modeling process can become rather inaccurate and tedious for the architect. The solutions offered in this thesis made use of the computational speed and accuracy of the computer for producing the connections between the units. The method as explained in Chapter 3 required a large amount of division calculation based on the design form. The calculations preformed were computed as a relational solution that divided the form proportionally to make the connections work between units.

The connections had to follow the curvature of the form, therefore directions, lengths, and orientations had to be computed to make it possible. The process explained can be computed directly by the architect but can produce inaccuracy in computation due to the repetitious and monotony of some of the calculations. There is also complication in having to recalculate the results based on a change in design intent which slows down design process and production.

The products of this thesis are units that mate to create a continuous form. It is important to understand the space relation that is generally needed between pieces that connect. Through multiple automations and testing of components using fabrication, designer can understand and negotiate the joining of pieces based on tolerances. For the ZCorp ZPrinter Z400, a tolerance of 0.0154" was allowed for the proper alignment and mating of pieces. The introduction of a tolerance parameter that can be manipulated based on the fabrication method and material used makes it possible for computing and fabricating multiple solutions. This approach tested the integral connections between the units. Previously, the designer would have to reconstruct the components by remodeling the form and recalculating the variances for the tolerances versus the expedited method outlined in this thesis.

4.3 Variation in design scale

During the design process, architects create multiple representational models at varying scales for the purpose of evaluation. The evaluation may manifest in different forms but mainly include structure, aesthetics, and space within the design (figure 4.0). The solution contributed in this research offers the flexibility for varying scale for design exploration (figure 4.1). The models can also be disassembled and transported to client for discussion and exploring design concept.

Figure 4.1 *Difference in model sizes*
Three different models showing the variation of scale based on the designer's need for evaluation

A printed model is normally restricted to the bed of the printing device. In the exploration conducted, the bed size of the 3D printer was 8" x 10" x 8" which limited the output to a maximum of 10" in height. The discovery helped in finding a solution for breaking complex curved form into units that can be printed separately at bed size (10") to produce a model possibly larger than the machine used for fabrication.

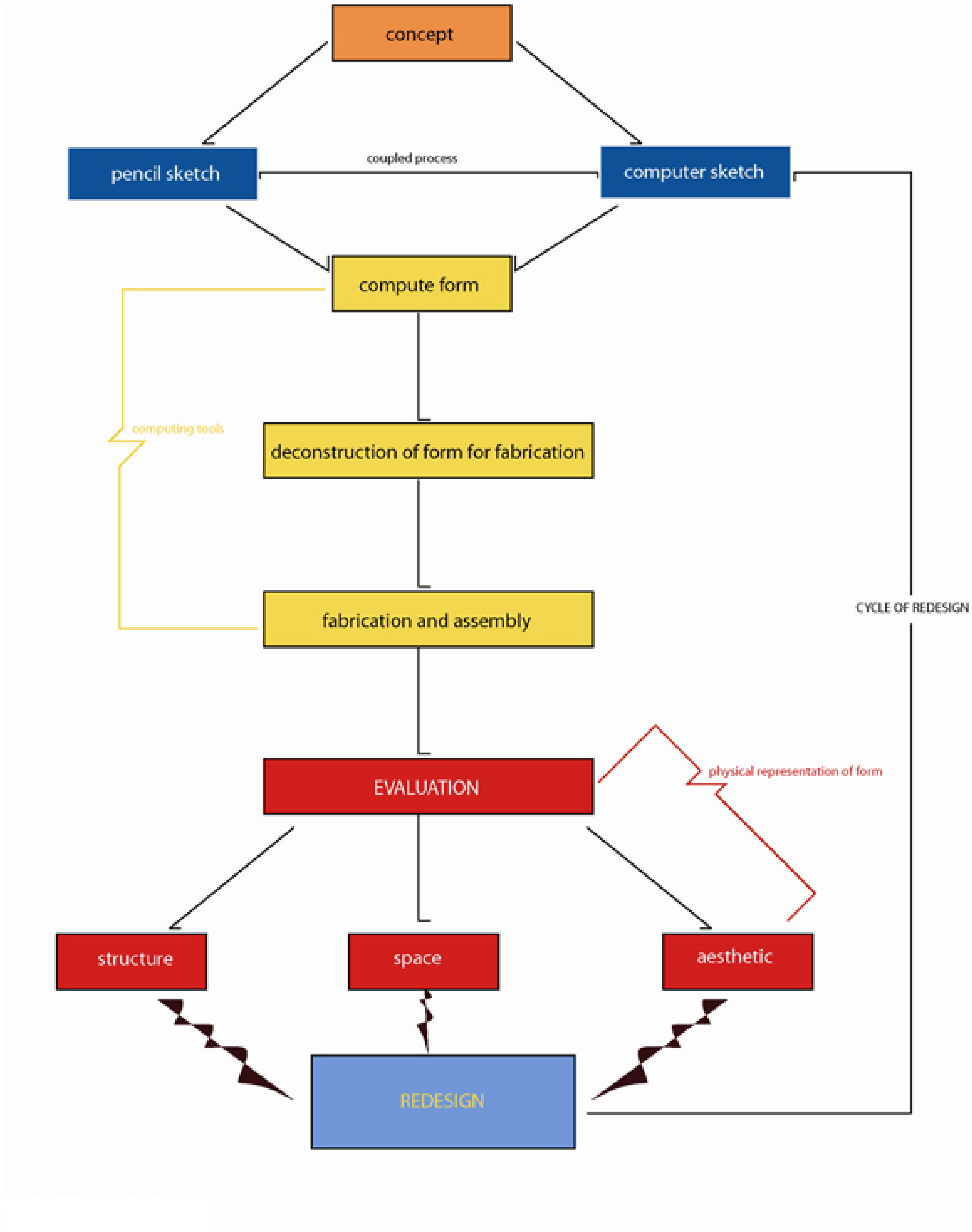


Figure 4.0 *Evaluation process*
Chart showing the evaluation process from concept
through redesign using a computational approach to design

4.4 Mapping data to other fabrication devices

The instigation caused by the exploration done throughout this thesis is to be able to emulate the desktop fabrication methods to life-size ways of fabrication. For this purpose it was essential to produce information that could be used generically enough for fabricating not only at different scales but with others forms of fabrication devices. The tests included using a Fusion Deposition Modeling (FDM) machine that fabricates plastic instead of plaster in the same layered process as the ZPrinter to produce 3D objects. The model produced was 2" (figure 4.2) in height and snap fitted using the integral connections embedded in the units. The use of both devices (plaster and plastic) proved the feasibility for testing the data in other environments such as future work that will include Contour Crafting, which uses concrete and ceramics as the materials.

Figure 4.2 *2" plastic model*
Model as small as 2" fabricated using Fusion Deposition Modeling (FDM)

4.5 Structural Limitations

This research did not undertake the structural issues of the design rather, it investigated the feasibility of computing complex-curved form for fabrication using 3D devices. The constructed models are limiting for life-size construction, but satisfy the architect as models that usher through the design process. The solution models presented are constructed using a plaster-like material as the building blocks which do not directly map to material used in construction. The systems introduced do not evaluate the form based on stress forces within the form for calculating material thicknesses and distribution; therefore, investigation into other materials and algorithms would be necessary to pursue the designs presented as life-size solutions.

Chapter 5

5.0 Contributions and Future Exploration

The research conducted in this thesis looked into fabrication and complex-curved geometry. Although this is merely an introduction to the possibilities of using 3D fabrication tools for life-size construction, it addressed many important contributions to the design process as well as the mainstream use of fabrication technologies.

5.1 Contributions

5.1.1 Fabrication devices within and design process

The lasercutter has had a dramatic effect on the design process both in academia as well as in the profession of design (Botha M, Sass L, 2006). The use of the lasercutter is slowly becoming a desktop commodity rather than a specialty device. Students are shifting to using lasercutters for building models rather than X-Acto blades. This process is now being mapped to life-size solutions to produce information that use the similar method of construction such as Computer Numerically Controlled routers. This thesis addressed another desktop method of construction using a 3D printer. 3D printers are currently being used to create massing models, rather than addressing issues of construction or mapping to life-size processes. The contribution made in this thesis looked at solving the fabrication of complex-curved geometry using a layered method of printing which can be explored for mapping to future construction devices such as Contour Crafting developed by Dr. Behrokh Khoshnevis at the University of Southern California.

5.1.2 *Complexity of designing with curvature*

As demonstrated in the works of many architects such as Frank Gehry who uses computational methods for solving design problems, it is possible to predict how computation will adapt into the mainstream of the design process and the complexity that will be created as design computed artifacts. The complication lies in translating the designs for negotiating with current fabrication methods as well as with current materials. Most methods rely on rationalizing 3D complex form to be manufactured using 2D processes and if solutions are not derived, it pushes the design intent in other directions. If other methods are not explored and other types of materials are not investigated, these forms will still be abstract designs of architects' dreams and intentions, but will not manifest in the final solution. This thesis concentrated on the implicative ways for testing and exploring different approaches to fabrication and materials. The work done in this thesis does not solve the problem of all complex-curved geometry, but is a step towards understanding what design computing produces and preparing for what is to happen in the future of design.

5.1.3 *Design computing*

The main vein within this thesis concentrated on the current process of design computing and the mechanics of fabrication. This thesis explained the total process of design computing rather than using a computer to solve the problem of design but instead, having a constructive dialogue with the computer. Design computing in the past has been conceived as a method of merely using a computer within the design process. This thesis exhibits the nature of what design computing will be understood as in the future of design. The thesis explained how forms were computed from design constraints (concepts) which had to be computationally negotiated with computer and designer- the designer was responsible for creating multiple examples of the form with the constraints in mind, but had to filter the solutions for one result. The forms were constructed from mapping design cognition of an "idea" or "perception" of the intent and the electronic computation help for driving the idea. The computed form had to further represent the idea and needed to be translated as a physical description to be exploited for evaluation. The power of the computation allowed the development and deconstruction of the form into units that were fabricated and assembled to achieve the physical artifact. The physical artifact in result communicated the design intent from designer using a stream of thoughts, to a computational representation, then to a physical representation, all which explained the design computing process and the design calculating involved.

5.2 Future Exploration

5.2.1 Structural investigation

To be able to build the designs presented in this thesis at life-size scale, future investigation would involve exploring computational methods for calculating the properties of the form for load distribution relative to the material used. The implications made in this thesis points in the direction of composite materials that can be printed in the same fashion as the ZPrinter. For structural feasibility, the investigation would include understanding the properties of concrete and introducing new elements into the mix for structural strength. Past investigation have used glass and fiber within concrete mixture to increase the tensile strength of concrete, but still shows dominance as a compression composite solution. As a result of the nature of concrete the future research will investigate the constructible feasibilities of the process outlined in this thesis.

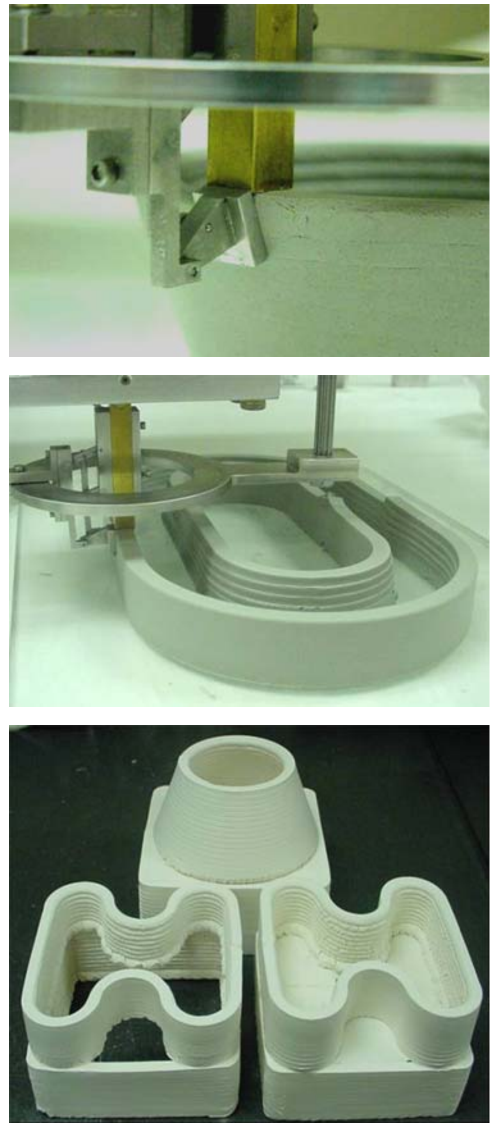


Figure 5.0 *Contour Crafting*
3D printer device developed in the University at Southern California for printing ceramic and concrete

5.2.2 Larger structures

The investigation undertaken in this research involved looking at the feasibility of creating computed data for fabricating with digital devices. This thesis took this goal and trimmed it to a more closed and controllable environment for answering the question. The thesis was successful for creating data through computation for digital fabrication as design models of architect's designs, but not at life-size scale. Currently at the University of Southern California, Dr. Behrokh Khoshnevis is investigating using a contour layering process for fabricating (Contour Crafting) (figure 5.0) physical objects from concrete and ceramic materials. The aim is to map the process investigated in this thesis to his process as further exploration.

Future studies will investigate larger walls built of higher levels of curvature. Investigations will explore means to generate the initial form through computational design processes and constructing the walls as built design examples. The exploration undertaken in this thesis research investigated the fabrication of complex-curved geometries in a controlled and closed experimental environment. The research only included the design computing process and the fabrication as the means of developing this thesis, further exploration would involve mapping this process to a scalable process for creating life-size designs through possible methods like contour crafting.

5.2.3 A seamless design computing process

The process of design will become an integrated dialogue of cognitive computation and electronic computation to aid in the developing and manifestation of design ideas, which will be defined as design computing. Architects will engage in a computational dialogue for creating design form and using higher level ways for understanding and manipulating computed geometry for showing greater success in building their intent rather than shifting to a compromised design. The architect will be equipped with an electronic sketchbook for engaging in design thoughts as seen with paper sketchbooks. This process would emancipate architects with the power to use the technology at hand for creating a greater computational approach for creating a paperless design journey from conception to fabrication. However, the architect will still be the designer as the computational agent. The architect will be responsible for instigating the design computation dialogue through simple embedding and seeing in the electronically computed design for the creation of complex-curved geometrical form.

Bibliography

Agarwal M, Cagan J, Constantine K, (1999) "Influencing generative design through continuous evaluation: Association costs with the coffeemaker grammar" pp. 253-272

Azariadisa P, Nearchoua A, Aspragathosa N, (2002) "An evolutionary algorithm for generating planar developments of arbitrarily curved surfaces", Computers in Industry, 47, pp. 357–368

Botha M, Sass L, (2006) "The Instant House: Design and digital fabrication of housing for developing environments", Proceedings of the 11th Conference on Computer-Aided Architecture Design in Asia pp. 209-216

Chau H, Chen X, McKay A, Pennington A, (2004) "Evaluation of a 3D Shape Grammar Implementation", Design Computing and Cognition pp. 357-376

Friedman M, (1999) Gehry Talks: Architecture + Process, Rizzoli, New York, NY

Heisserman J, Callahan S, Mattikalli R, (2000) "A Design representation to support automated design generation", Artificial Intelligence in Design pp. 540-566

Kilian, A., (2004) "Fabrication of partially complex-curved surfaces out of flat sheet material through a 3d puzzle approach"

Opas J, Bochnick H, Tuomi J, (1994) "Manufacturability Analysis as a Part of CAD/CAM Integration", Intelligent Systems in Design and Manufacturing pp. 261-292

Rudolph S, Alber R, (2002) "An Evolutionary Approach to the Inverse Problem in Rule-Based Design Representations", Artificial Intelligence in Design pp. 329-350

Testa, P., Weiser, D., (2002) "Emergent Structural Morphology," AD Architectural Design, Special Issue, Contemporary Techniques in Architecture, Guest Editor Ali Rahim, Academy Editions (London) Vol. 72, No. 1 pp. 12-16

Sacks R, Eastman C, Lee G, (2004) "Parametric 3D modeling in building construction with examples from precast concrete," Automation in Construction 13 pp. 291-312

Sass L, Shea K, Powell M, (2005) "Constructing freeform designs with rapid prototyping"

Sass L, Oxman R, (2005) "Materializing design: the implications of rapid prototyping in digital design" Design Studies pp. 325-355

Schön, D., (1983) The Reflective Practitioner How Professionals Think in Action. Basic Books

Shea, K, (2004) "Exploration in Using An Aperiodic Spatial Tiling as a Design Generator", Design Computing and Cognition '04 pp. 137-156

Sheldon, D, (2002) Des. Digital Surface Representation and the Constructability of Gehry's Architecture, MIT, Cambridge, USA

Smithers, T, (2002) "Synthesis in Designing", Artificial Intelligence in Design '02 pp. 3-24

Smithers T, Conkie A, Doheny J, Logan B, Millington K, (1989) "Design as Intelligent Behaviour An AI in Design Thesis Programme", Artificial Intelligence in Design pp. 293-334

Soman A, Swapnil, Campbell M.I. (2003) "Toward an automated approach to the design of sheet metal components", Artifical Intelligence for Engineering Design, Analysis and Manufacturing, 17, pp. 187-204

Stiny, G, (1977) "Ice-ray a note on the generation of Chinese lattice designs", Environment and Planning B, volume 4 pp. 89-98

Smyth, E, (2001) Des. Designing Aesthetically Pleasing Freeform Surfaces in a Computer Environment, MIT, Cambridge, USA

Wang Y, Duarte J.P., (2002) "Automatic Generation and fabrication of designs", Automation in Construction II pp. 291-302

Wassef W G, Davis D, (2004) "Integral Steel Box-Beam Pier Caps", National Cooperative Highway Research Program, Report 527

Westerberg A, Grossman I, Talukdar S, Prinz F, Fenves S, and Maher M.L., (1989) "Application of AI in Design Research at Carnegie Melon's University's EDRC", Artificial Intelligence in Design, 335-361

Zhao Z, Shah J.J., (2005) "Domain independent shell for DFM and its application to sheet metal forming and injection molding", Computer-Aided Design 37(9) pp. 881-898