Computational Design of Parametric Scripts for Digital Fabrication of Curved Structures

Sergio Araya Goldberg

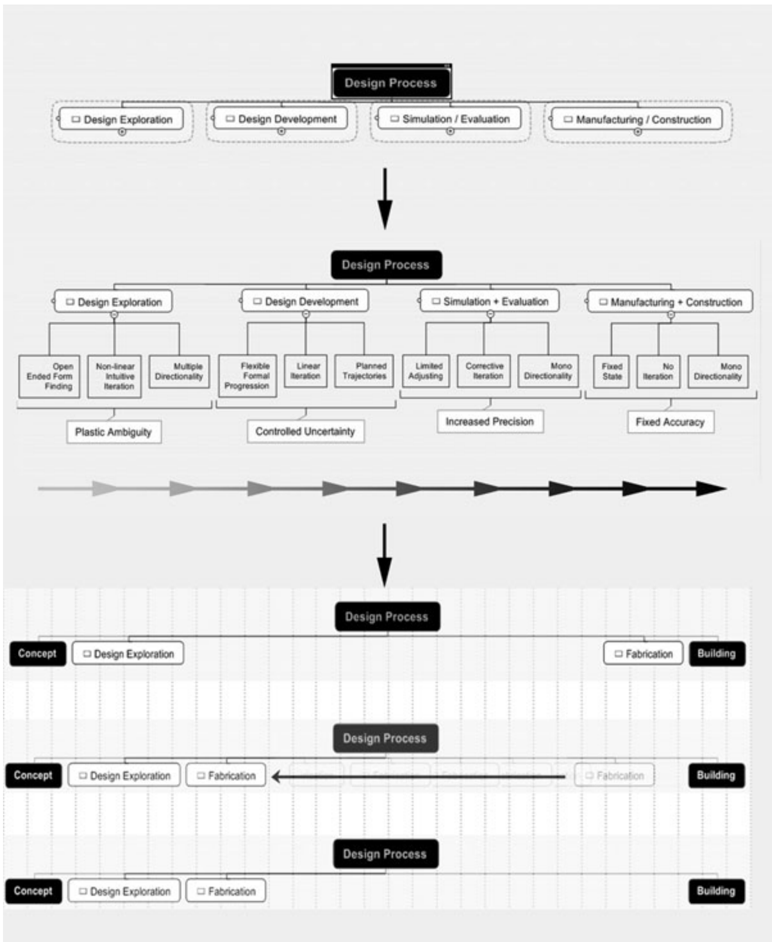
Computational Design of Parametric Scripts for Digital Fabrication of Curved Structures

Sergio Araya Goldberg

This paper explores strategies for building toolchains to design, develop and fabricate architectural designs. It explains how complex curved structures can be constructed from flat standard panels. The hypothesis of this research is that by embedding ruled based procedures addressing generative, variational, iterative, and fabricational logics into early phases of design, both design techniques and digital fabrication methods can merge to solve a recurrent problem in contemporary architectural design, building double curved structures. Furthermore it achieves this using common fabrication methods and standard construction materials. It describes the processes of programming computational tools creating and developing designs to fabricate continuous complex curved structures. I describe this through a series of experiments, using parametric design environments and scripted functions, implementing certain techniques to fabricate these designs using rapid prototyping machines. Comparing different design and fabrication approaches I offer a discussion about universal application of programmed procedures into architectural design.

1. INTRODUCTION

An architectural design process usually comprises a series of phases of development, going typically from conceptual development and sketching, to programming development and then to design development[1], later in the process, detailing drawing takes place and finally construction planning and documentation occurs. Usually, this implies that some refinement has to be done to the design in order for it to be manufactured using the available construction technologies. Figure 1



▶ Figure 1 Phases in a design process gradually shift from flexibility and ambiguity to fixation and definition. Incorporating Fabrication logics and procedures to enhance design research and exploration

The degree of refinement will vary according to the particular design and the fabrication processes involved. In some cases when the refinement has to be more radical, the design has to be simplified, reduced and transformed in order to be fabricated, usually meaning the attributes that made that design valuable in the first place. The objective of this research project is to improve the design process by incorporating manufacturing logics at early stages of the design process to tackle these issues.

Parametric Design implies a whole new paradigm of non standard design through the propagation of the difference, the repetition of the variation. The ability to control variation and adaptation to local conditions on designs allows more precise yet complex solutions. Computer Numerically Controlled Manufacturing (CNC) allows to mass produce non standard elements at almost the same pace as standard industrial processes do with identical repetitive elements.

"We already had a digital revolution, we don't need to keep having it"[2]
N. Gershenfeld

The Director of the Center for Bits and Atoms, Neil Gershenfeld, points now to the imminent *Revolution of Making*, the programming and fabrication of the real world, not just the electronic virtual worlds. During the last two decades, computers and digital technologies have infiltrated our lives. Computers have populated our work environments, including architecture offices. The power of computers relies partially on their ability to calculate very fast complex mathematical formulas. This has allowed that complex geometries, based on increasingly more complicated equations, became possible tools for design, introducing a whole new family of shapes and curved topologies, splines, and spline surfaces. Nevertheless, these tools, as present in almost every CAD package available in market today, still offer just discrete control and limited options; as the relations and dependencies between points and curves and curves and surfaces are usually fixed by default, we cannot fully grasp the latent potency of such geometric constructions. New computational tools however, called parametric design environments, allow programming these dependencies, with variables called parameters, between one point and another, and build the rules to trace a particular curve or geometry, defining the "degrees of intelligence" of these points and the relationship between them, and therefore the curves later derived from them, thus enabling the creation of controlled curved surfaces. Parametric environments then are usually associated with "smart geometries"[3].

While these digital technologies are now able to manage complex geometries, making curved shapes and geometries almost ubiquitous with a notorious presence in the architectural realm for the last decade, the transition to actual fabrication has been difficult. Only few designers in fewer projects have succeeded in building such designs, never without yielding part of the complexity of the geometries to the fabrication processes utilized, and requiring extremely costly and specialized manufacturing. This research project consists in developing a series of tools and techniques that can be used to design and fabricate double curved structures.

1.1. Design Technologies, CAD and Parametrics

Computer Aided Design CAD has become a regular tool, almost omnipresent at designer's offices from all different fields. Its unquestionable efficiencies make it an irreplaceable tool. One of its many comparative advantages is that of efficiency in performing complex repetitive tasks, much faster than if they were drawn by hand. Nevertheless, most CAD packages available in the market today, operate in a linear way, were the order of the operations is performed and stored in the program chronologically, and thus the data is hierarchically ordered based on this sequence. In terms of the data stored, each geometry is defined individually, without any reference to the rest of the geometrical descriptions in the file, therefore, without any kind of associativity between them. This implies that every process that modifies something on the design environment, will transform that object or entity into something else, thus acquiring new properties and loosing the preexisting ones. If a mistake is done or just to go back to its previous state, the program will require undoing the operation, loosing the recently acquired new properties or characteristics and recovering the old ones. It is very hard or impossible to obtain intermediate results between these two "extreme" results, requiring many "trial and error" iterations, doing and undoing constantly. This is a very common operation performed regularly and mechanically by architects during the design process.

Another characteristic of CAD software is that they are based on the assumption that a designer uses the software and its tools as a pallet of functions which are combined differently to produce different designs. Both the tools and the final design are conceived in a static state. The tools can be combined in different ways to produce different effects on the design, but they are always fixed. In this case, different software provides different functions, thus tending to produce a specialization on CADs. This implies that many times, a design undergoes a series of steps in specific pieces of software, and then is imported to another one to use different functions to transform the design according to specific processes, and so on and so forth. The result of this is a design that constantly shifts depending on the environment where is being "transformed", but the overall result is always a fixed state that depends on the sequence of the functions applied to it. Further more, this search for "control" over functions, which are present in different software, implies translation processes, that each time reduced the information stored within the file, to make it "compatible" to other applications where it will be imported. Each translation deprives the model of the possible "smartness" or "intelligence", the parameterization of the objects and functions present in the previous software, but useless and unreadable in the next one.

A parametric approach provides the opportunity to build a set of relations between operations and or functions, in order to be able to allow the design to be "tuned" or "calibrated" without undoing all the process and restarting from scratch. Also, if the functions are incorporated into the parametric "intelligence" of the model, they could be adjusted or tweaked to test different configurations, or just distributed over time or based on local conditions of the geometry, to test different design strategies and possible evolution pathways for each design. This methodology allows building a "smarter" model, where the relations and the hierarchy dependencies are coded by the designer. It also permits to change any of these variables or relations, transforming the geometries, but without loosing the previous relations or dependencies. If a parametric approach is combined with scripting capabilities, all the different functions from different software can be accessed from the same parametric platform, eliminating the issue with exporting and translating the geometries, loosing information and "intelligence" of the model in the process.

2. METHODOLOGY

2.1. Parametric Modeling

Parametric environments provide a platform for design where the user or designer cannot work just with the basic given palette of shapes, in fact, there is not such thing as a palette of basic shapes or solids in parametric software; everything starts always, and has to be built bottom up, from points. But it includes also the ability to integrate the functions and relations between shapes, and even between functions themselves. In a parametric environment, the resulting design is the consequence of setting up a number of conditions regarding the geometry of the design, the relations between those geometries, the functions applied to obtain or derive these geometries, and the relations between those functions. This results in a higher level of control over the resultant design, where the design process can be streamlined in terms of different iterations of the design, as a change of a parameter will affect all the functions that depend on it, modifying the end result, the actual resultant shape. This is an advantage over standard CAAD platforms, as in most cases, these update methods for adjusting particularities of a design happen in real time, allowing the user or designer, to quickly evaluate different alternative solutions for a particular problem.

This aspect of parametric design increases considerably the amount of possible answers to specific problems that which can be reviewed and evaluated, this has been referred by some authors as the enhancement of a "solution space" thanks to these parametric design methods. Parametric design has been usually associated to design for fabrication, as some of the relations that can be "programmed" into the design are related to production and manufacturing. Parametric environments were originally developed and have been used for years in the car industry as well as in the aeronautic industry. It has been introduced to the architectural arena in the last years by firms like Gehry and Partners, Foster and Partners and others.

2.2. Scripting and Parametric Design

Scripting provides a number of advantages when applied as a tool to design. Scripting gives the power of recursion, which allows performing repetitive tasks in a faster and efficient way. Recursion provides speed for calculating large number of functions or operations that are of the same kind. It allows automating iterations, to construct complex objects or methods using smaller and simpler functions, which are repeated several times.

Scripting also gives the power of abstraction, as complicated functions and operations can be compiled to provide a "compressed" function or object that can be then called by its methods of implementation. This provides a chance of programming complex imbricate algorithms which can be compiled into program components. Abstraction allows thinking modularly, where different modules perform different tasks and can be combined to create more complex modules to build a large construct. Scripting also provides the ability to program reasoning, through embedding conditionals which will "read" given information and decide how to proceed according to that information. This allows to program "conditioned behaviors" into the design, as it will be able to make decisions according to programmed conditions. Conditionals allow the user to set different responses from different scenarios, as opposed to working linearly with only one possible direction and only one predefined result.

Digital design tools have been incorporating scripting platforms in their latest versions, as a way of facilitating the ability to extend the programs functionality through scripted custom functions. Some of them provide internal script editors; some others read and compile scripts created in external regular text editors. In the last case, the scripts have to be opened in the program and then executed. While they extend and enhance greatly the functionality of these CAAD platforms, a disadvantage is that in these cases, due to the original platform being a standard CAAD platform, when the scripts is not providing the desired functionality in the design, the user has to go back to the script editor or text editor, rewrite the script, save it, reopen it in the design software, and finally re run the script to be able to evaluate the results. This is a great advantage in terms of performance over manual processes, but still lacks the fluidity required to allow fast iterations that can be rapidly subject of design evaluations to inform design decisions. The trial and error method of these current algorithmic and scripting processes can be extremely time-consuming, especially when in a design process the final objective is being developed during the process itself. And it can be even more disappointing and frustrating while debugging. Beyond the time issue involved, this method while powerful in terms of providing an extended functionality for the design process, lacks the smooth relation between computation and representation expected in digital design environments, which delays the evaluation process of the resulting design and therefore slows down the design process itself.

2.3. Parametric Design within Generative Components

For the purpose of this study, I have used an experimental parametric platform called Generative Components, which has been developed by Robert Aish at Bentley Systems, as an extension of their CAAD package Microstation. It is not yet public, except for a small (but growing) group of researchers and professionals who have been testing it and helping in its development, scattered mainly through England, US and Canada.

2.4. Design Strategies

The approach proposed by this methodology targets two different issues, how to design for fabrication and assembly, and how to fabricate these designs.

The first method deals with the aim of designing a complex continuous curved structure, which has to be developed as a whole, but which will be later manufactured by assembling fragments or parts of it. This implies that in parallel it has to be thought both as a whole and as a series of related parts. For this I use a method of subdivision of the structure (the original surface), which returns an object that is not an exact copy of the original one, but a fairly approximated one, where the degree of "approximation" or of "likeliness" can be partially controlled by the resolution of the subdivided output object. This allows to test, during the design development phase, the most appropriate design parameterization in terms of the output for fabrication. A secondary method is implemented to provide a number of choices for the design to be fabricated, regarding formal, material, structural and other design criteria. For this purpose I built a series of functions that address particular manufacturing requirements, which will constrain but simultaneously facilitate the fabrication process.

The methodology that I applied consists of a combination of parametric modeling and scripting, to provide the digital tools required. The processes were evaluated both in their digital environment as through the physical output that they provided through CNC machining and posterior assembly. The method proposed, was divided in two parts. The first addresses the challenge of designing a surface considering its later subdivision and re-assembly; this is a generic method that can be applied to any surface independently of the fabrication process. The second part provides a chance of choosing between different desired output results from the design, therefore choosing a particular fabrication method and materiality. I believe that by doing this at early stages of the design development process it will in same ways constrain the process, but it will also enhance the design solution and material result.

3. TOOLS, TECHNIQUES AND TECHNOLOGIES

In contemporary architectural discourse, within post digital culture, three concepts are used as synonyms, almost indistinctively, while they have different meaning and refer to different stages of cultural knowledge: tool, technique and technology. Digital tools and digital technologies are often used as equivalents, I will explain the differences. Tool is something that is used to perform an action, it is the instrument. Technique on the other hand, is a method or group of methods for accomplishing a particular task. But technology is the body of knowledge, available to a society, which is of use to achieve specific practical purposes. Then the computer is a tool, as it is the software running on it. They are used to execute specific actions or operations, to achieve certain objectives. The specific methods developed to use computers and software, inventing and perfecting creative processes to achieve better results, are recorded as techniques. But technology is achieved, when a new knowledge is produced from the creation and application of certain techniques, running specific tools, to achieve desired objectives. Although this research is still in progress and the results are partial, I will use them to explain how emergent computational tools, parametric design environments and numerically controlled fabrication processes can be implemented in the development of architectural designs.

3.1. Geometrical Constructs

Spline curves are traced using the vertices of a control polygon. Modifying the position in space of this poles alter the geometry of the spline. In the same fashion, surfaces can be traced by a grid of poles, resulting in complex curved surface. A spline curve, is always tangent to the first and last poles, but the rest of the poles remain external to the curve. While the continuity of the curvature is one of its main appeals, it is also a big problem. If a spline is cut in two each resulting segment changes, becoming tangent to the new edges and acquiring a new geometry. When this new splines are put together, there is no continuity on the curvature. Usually, when this happens in real construction works, the pieces are stressed to fit, almost always wrinkling and even breaking as a result of these forces, the titanium tiles of the Guggenheim Museum at Bilbao, by Gehry and Partners is an example of this. Another aspect of these continuous complex curved structures is that cutting them in pieces will result in large collections of pieces, almost all equal, but every single one slightly different. In regular CAD this operation is extremely time-consuming. Parametric environments tend to be streamlined towards fabrication, which facilitates building processes like for example, unrolling ruled surfaces. But there is no possible unrolling of double curved surfaces. The technique that I developed for this research uses the unrolling principle to obtain a flat panel from an approximated double curved surface. This paper explains how complex curved structures can be constructed from flat panels, borrowing a concept from mechanical engineering: "flexure structures".

3.2. Manufacturing Curved Structures

Construction industry is founded on standardization and modularity. Most construction materials come in flat sheets or panels. And casting materials that can acquire free forms also require molds that have to be made from flat panel materials. Therefore, the process of translating these continuous complex curved structures is always painful and usually requires translation to a more rational expression. This paper describes a procedure using "flexure" structures developed in parametric environments, to fabricate partially double curved structures from rigid flat panel using rapid prototyping tools and CNC machining. In the larger perspective, this research approaches the notion of generative design tools and their ability to use digital design fabrication logics and processes to extend the actual boundaries of constructability in contemporary design. Furthermore, this approach engages the reality of construction industry and local economies, providing an affordable response to these complexities.

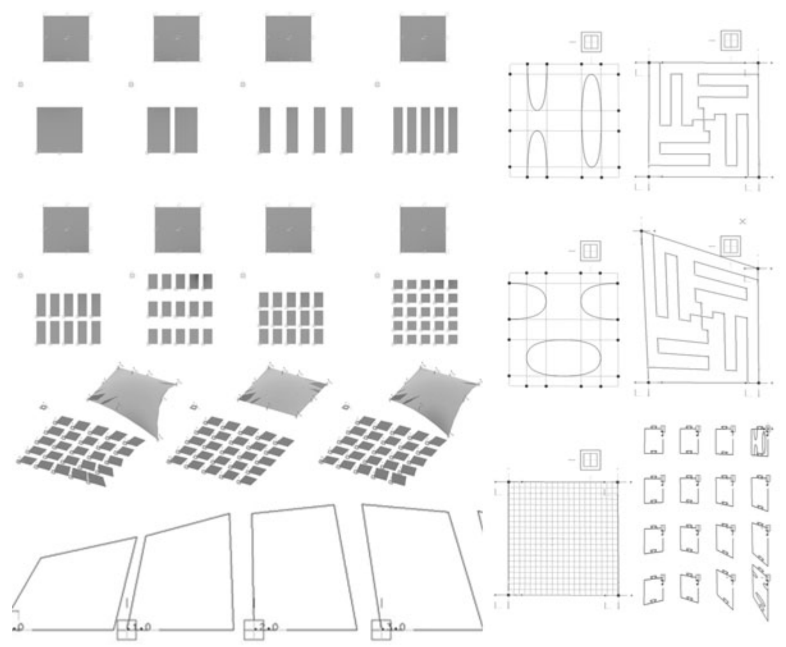
3.3. Flexures

Compliant structures are those who can change its shape when a force is applied to them, and that will return to their previous state if the force is taken out, for example springs. There is a type of compliant structures that behave similarly to springs, called flexures. These structures deform elastically depending both on material properties and on its geometry. This paper describes how to create complex curved structures from flat rigid panels through the flection of these structures. According to Larry Howell, it is a special kind of mechanism, "a mechanical device used to transfer or transform motion, force, or energy" Typically they are made of "rigid links connected at movable joints."[4]

A compliant mechanism or flexure, however, while still performs the same basic functions of transferring or transforming energy or force, gains at least part of its mobility "from the deflection of flexible members rather than from movable joints only"[5]

Flexure structures have several advantages as they reduced the number of pieces involved, reducing the assemblies required, and therefore reducing its costs. But more important for this investigation, is that they can be developed from single pieces. Flexure structures are frequently used in machines that require very precise movements, as they have a reliable displacement precision. They can effectively isolate their movements to the axis where the maximum flexibility has been provided from other lateral movements, "reducing the vibration natural to hinged joints, eliminating the friction between movable parts and the backlash from their rigid body and hinged counterparts"6 .

Applying this notion of flexure in this investigation provides a method of material transformation, where solid rigid flat boards can be developed into partially flexible structures. This process was conducted through experimentation on different geometrical patterns and the performance obtained from them when applied to a solid material. The fabrication process chosen was material removal by cutting these designed flexure patterns onto the rigid boards. Figure 2



► Figure 2 Sub-dividing the double curved shape in components and extracting them one by one and unfolding each of them separately. Being a parametric scripted model, any changes to the original surface model are automatically updated in the disassembled-into-components version. Every piece is tagged, which serves as a guide for post fabrication registration, alignment and assembly. The scripted pattern can change orientation according to material requirements. The function is based on drawing flexure patters on a parametric grid. Custom pattern designs can be applied.

4. DESIGN FOR FABRICATION

The computational algorithm utilized, was to abstract and fragment the complex 3D geometry of the surface in order to process it as smaller flat 2D shapes that could be used as fabrication models. Then specific functions developed according to particular manufacturing procedures and machines are applied.

4.1. Parametric Components.

Two different strategies were explored and compared to model these curved designs: One was writing a script in Visual Lisp to be executed in Autocad, to create a parametric flexure patterned structure, the other approach was by creating several nested parametric features in Generative Components, and then populating an unrolled approximated double curved surface with them.

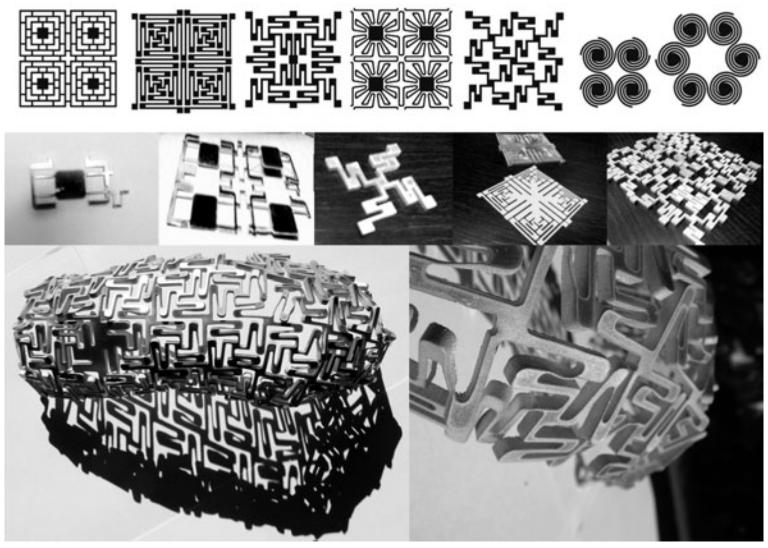
The first strategy developed a Visual Lisp script that draws simple flexure patterns. Two variables control the size of the cut figure, and another two secondary variables control the spacing between them. This first script works only in two dimensions, but allowed me to test different material behaviors, manipulating the variables, studying the tolerances required for a machine, and a specific material. A second script, still in progress, works on three dimensional curved surfaces. The first step was to create an approximated spline surface, because Autocad does not have spline surfaces or nurbs. So I wrote a script that uses splines as inputs. Then I used a spline subdivision function written by Takehiko Nagakura[7], to create lists of points which are used to create another series of splines, going in the opposite direction, connecting the first point on one of the original splines, with the first point on the second original spline, and so on. This second list of splines provides the resolution on U direction on the future spline surface. This list is subdivided again , obtaining the resolution on V direction, creating a dense point grid in 3D space.

Based on this script, I wrote a couple of functions that create ruled surfaces between the splines obtained, producing an approximated spline surface, even though each segment is a developable surface. Based on these properties, each of these segments could be unrolled or developed, and then used as a base for reproducing a two dimensional flexure pattern.

According to different materials and design requirements, this procedure of decomposing the surface into quad-patches could be useful, providing a precise layout for each facet to be fabricated out of a flat panel. In other cases, it might be better to decompose the surface into strips to be cut from larger panels. As the objective of this research is to enable possibilities for design, through the embedding of fabrication logics, these options were kept as parameters in the script. By facilitating the unfolding of the bspline surface points into a flat bidimensional representation, this procedure transforms the design geometry into a fabrication layout, but at the same time, allows to conceive, adjust and perfect such design according to the procedures that would be later involved in its construction.

The second strategy was by nesting parametric components in Generative Components (GC). This first procedure turns a bspline surface with double curvature into a parametric faceted surface by applying a triangulation method via a script. The script locates a series of points on the surface based on its UV values. These points sampling the surface are translated, rotated and aligned in another plane, effectively unfolding the points. Figure 3

As each facet is created by grouping four non collinear UV points on the surface, the resulting facets are quadrangular, which can be reduced to two triangles sharing a side in common. The first triangle is translated into a plane, then the second triangle, takes the common edge of the first unfolded triangle and aligns its third vertex to be coplanar with the vertices of that first triangle. The result is the unfolded quadrangular facet. An optional procedure is included to unfold the second facet, using the common edge with the first facet, either in the U or V directions, creating a continuous facetted strip.



► Figure 3 Samples of Flexure patterns designed and evaluated in this research and some flexure patterns cut from steel and aluminum plates in a CNC WaterJet. Structure showing flexure patterns on double curved structures obtained using rigid aluminum panels.

I created a flexure pattern feature, based on an array of points supported by a basic quadrangular shape. The feature is populated over a point grid placed on a spline surface. A global variable controls the number of points used to subdivide the surface, adjusting the *resolution* of the flexure surface.

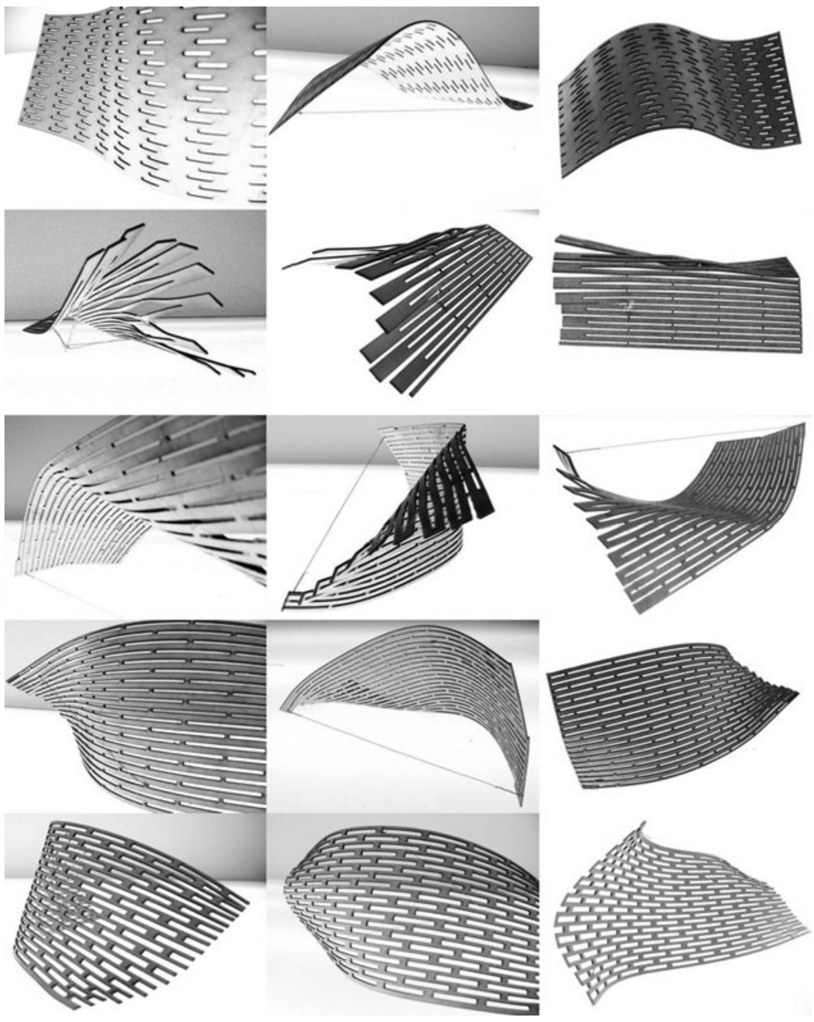
The shape grid used as vehicle for population is unfolded in a different model. After re importing it to GC, the unfolded shape array can be promoted to GC and can then be used to reconstruct a shape grid. Finally the flexure pattern feature is applied to this unfolded surface, obtaining the cut sheet which can now be exported as a regular 2D CAD drawing to be machined.

4.2. Pattern Script

The strategy explored to model these flexure pattern designs consisted of creating a parametric feature in Generative Components (GC), and populated an unrolled portion of the surface with it. The creation of the flexure pattern "*feature*" was based on four points, supporting a basic quadrangular shape. The feature is populated on a shape grid placed over the bspline surface. The shape is used as a vehicle to group and order sets of 4 points in order to insert the parametric feature on the surface, or in the unfolded version of it in this case. A global variable controls the number of points used to subdivide the surface, adjusting the resolution of the flexure surface. The shape grid used for population is unfolded in a different model. Finally the flexure pattern feature is applied to this unfolded surface. Global variables were exposed to be able to control the nesting of the unfolding shapes as a cutsheet, optimizing the material use and reducing the cutting time. Figure 4

4.3. Tagging

A common difficulty when dealing with large number of different pieces to assemble is how to match the pieces, usually requiring an assembly diagram. The script includes a tagging feature that labels each quad patch obtained and unrolled from the original surface. The tag also works as registration mark as it is located always in the lower left corner of the piece, facilitating its alignment. Furthermore the tag can also be applied to the subdivided surface, which acts itself as assembly diagram explaining where each tagged patch goes.



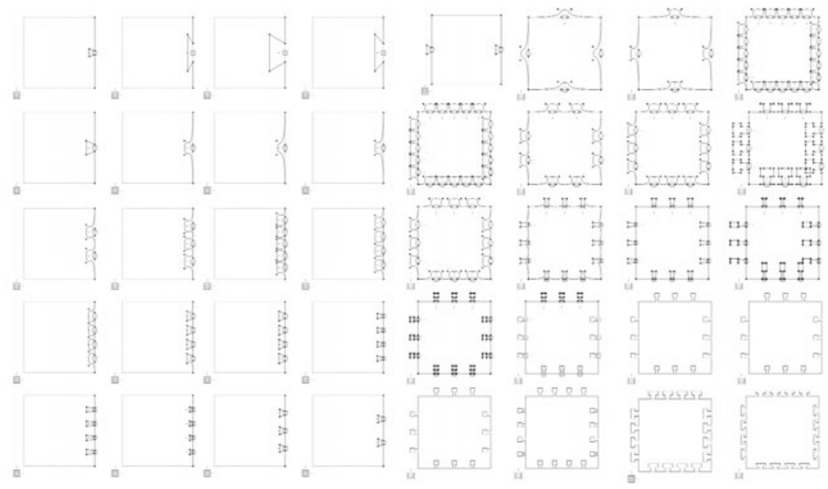
◀ Figure 4 Laser cutted models evaluating material performance of parametric variations of flexure pattern in plywood panels

4.4. Scripted Joints

If a design is to be fabricated as parts or components to be assembled, careful consideration has to be given to the way these components will be joined. The algorithm proposed for the series of exercises conducted for this research uses a common starting join concept, which is developed and adapted according to each specific fabrication method. I started using the common dove-tail joint detail, usually used in carpentry and woodwork. I use this detail as it provides an efficient yet simple press fit joint which could be later modified to different extents as different designed assemblies would require.

The joint detail was scripted as another parametric feature in GC, which could be nested on top of the series of other features to provide a complex modular component design yet maintaining control over the individual features that control the component. The points used to drive the dovetail detail are placed using the planes defined by the triangles obtained by the subdivision function. Every dovetail tenon has a correspondent dovetail mortise. For the joint to work on different materials and with different machining processes, a tolerance value was included, reducing the tenon in size, in a ratio that can be controlled globally. Global variables were exposed to control the size and shape of these details.

Another concern was the location and frequency of these joints, therefore the parameter that controls the number of joints populated on each side of the quadrangular patch was exposed as a variable, redrawing the bspline line that defines that particular edge of the patch to take any number of tenons or mortises. This method provides a flexible solution that ranges from individual joint details for each quad patch up to a continuous joint seam. Figure 5



► Figure 5 Parametric variations of the join system. 16 joint types were hardcoded, but user can add custom joint patterns extending the tool

4.5. Parametric Constructs

Removing material from a flat board weakens its structure. If this operation is performed according to the material flexibility of the panel, it is possible to obtain degrees of flection from it, even from the most thick and rigid panels. Different patterns can be applied to cut and or remove material from flat panels to be able to flex them and obtain curved geometries. Furthermore, I will show how different patterns can be applied to obtain different performance from the original panels and therefore, different surface effects. Two different materials have been tested for this purpose, to investigate the incidence of the material properties in relation to the material removal process.

I used 2 mm plywood on a laser cutter to test different results derived from a script. The wood panels were semi rigid, allowing to be slightly bent along the longest dimension, but completely rigid (for hand applied force) in the short dimension. The cut sheets were imported directly from the dwg file where the script was executed. The speed and power of the cutting were related to the resolution of the pattern, so they had to be adjusted every time to avoid burning the wood and creating flames. Even though the script was very simple, the results obtained were very different, depending on the values of the variables that control the script. The wood acquired the desired material elasticity in order to bend not just in one direction, but in both, obtaining double curved structures, from original, semi rigid panels. When the deformation force was ceased, the panels return to its flat state.

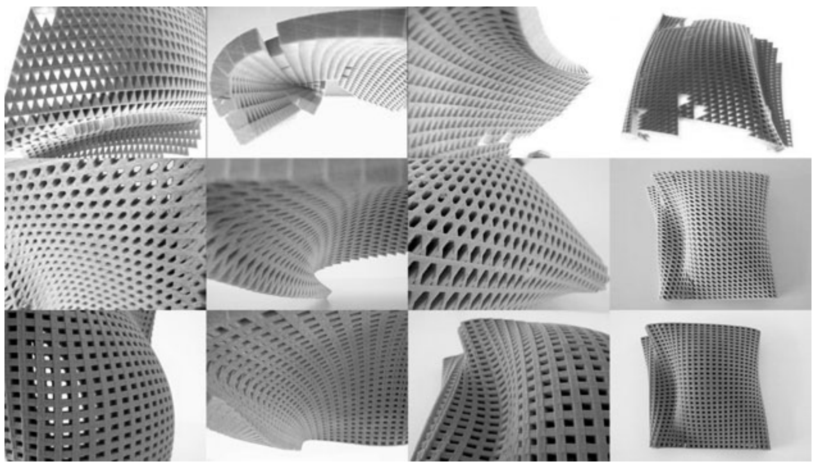
A second test was performed on an Omax Water Jet machine, to cut flexure patterns on 4 mm aluminum. This panel was absolutely rigid in both dimensions. Several test using different patterns were conducted to study the results of varying lengths and thicknesses, comparing removed material percentages and elasticity acquired. A more complex but still regular 2D pattern was applied. The result is that material elasticity was achieved, although the ranges of plastic deformation were several times smaller, given the material properties. Nevertheless, the aluminum panel became ductile enough in order to be shaped by hand into a double curved structure. The flexure pattern applied, gave enough elasticity to the material to allow plastic and non plastic deformation. When the plastic threshold is overcome, the structure does not go back to its original state.

4.6. Machining

The actual fabrication of the components required adjustments according to the machines and the materials utilized. These adjustments were granted by the variables scripted in order to control the tolerance of the joints and the nesting of the shapes in the cutsheet. The tolerances for the laser cutter had to be bigger than the other machines like the waterjet, where given the toolpath it is possible to specify the side of the cut, requiring only small adjustments in the tolerances to obtain press fit precision. The ratio between the speed of the machining and the cutting power was used to provide a smoother or rougher finish, which can also be used to increase the friction between components ensuring a better assembly. It was hard to calibrate the tolerance when pieces to be matched were processed in different machines. In general the strategy that proved to work best was to provide at least on one of the pieces, a rough edge in order to help the press fit assembly.

4.7. Printing bricks

Even though the purpose of this paper was to explore fabrication on flat panel, I detoured from the original line studying other fabrication methods that could be implemented from the development of the scripted routine. I wrote a function to create solid extruded modules using the point grid obtained from the previous script. Global Variables were defined to control the height of the extrusion. More functions included in the final script were developed to provide a hollow module, and control on the number of sides that the module would have, I tested pyramidal and cubic modules. This result was fabricated using 3D printing process on a ZCorp machine. This line of research has the potential to develop as a separate study, developing complex structures using a non standard modularity approach. Figure 6



► Figure 6 Example of initial results applying the scripted parametric construct to produce 3D printed blocks seeking a custom masonry system

5. CONCLUSIONS

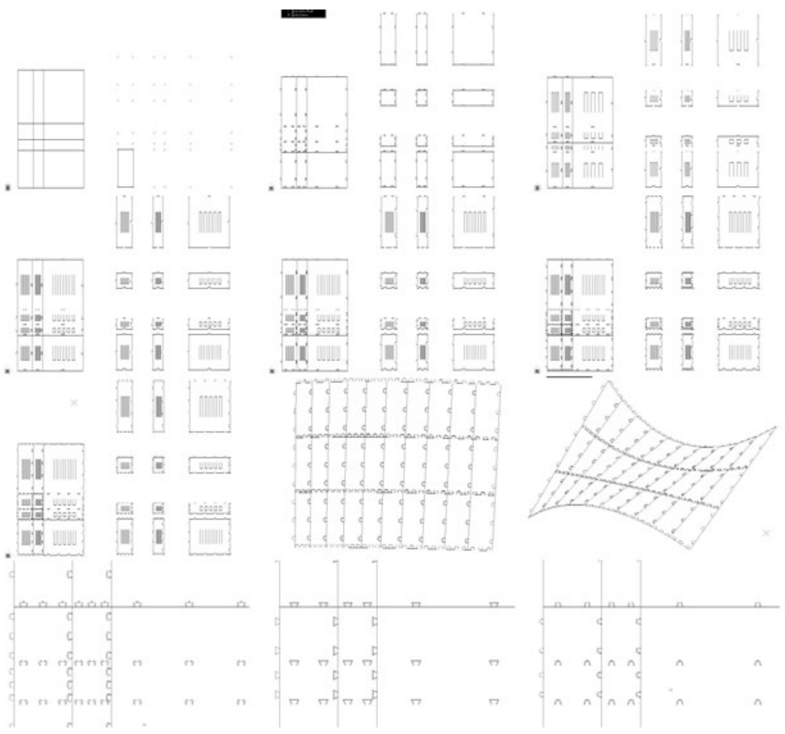
The research described in this paper is still in progress, but the results obtained so far are interesting and promising.

The script files developed prove that the geometrical logics of parametric design environments can be used to drive fabrication models.

The tests performed for this research were at model scale, and further testing is required at real scale with real materials. Nevertheless the results obtained from the model tests, prove that this techniques could be performed using standard materials. The Water Jet test in aluminum shows that it is even possible to obtain complex curved structures form rigid materials, through these techniques.

Comparing the results from scripting to previous tests achieved by modeling, the performance of the script is several times better. Scripting in GC combines the speed and performance of programming and the

adaptability and flexibility of parametric design
Finally it is promising to note that large structures could be develop by this way, reducing the number of pieces required to be assembled, providing a range of deformation to adjust in place the continuity of the curvature between assembled pieces. They could be eventually, while performing with plastic deformation, be unstressed and laid out flat again for transportation advantages. Figure 7



◄ Figure 7 The scripted parametric construct extracts the shape, unfolds each component, creates joints and adjust the proper type of joint systems, applies a flexure pattern, and controls the tolerance between pieces for tight or press fit joints according to different fabrication procedures.

6. FURTHER RESEARCH

The results shown in this paper are preliminary and further research should be done to accomplish more precise and universal results. In depth exploration using parametric components should be performed to demonstrate further functionality of decomposing complex features into cut sheets for fabrication. This opens a wide spectrum of possibilities for architectural design and building technologies, providing a technique to streamline the production of complex components.

7. ACKNOWLEDGEMENTS

All my gratitude to Robert Aish for inviting me and letting me take part of the GC community and for his continuous support, always stimulating this research and other projects.

Sincere thanks to my advisor and readers Ann Pendleton, Lawrence Sass, Takehiko Nagakura and George Stiny, for their support and insight during the development of this research.

Thanks to the RPLab and Media Lab at MIT for making this research possible.

References

[1] Sass, L., *Rapid Prototyping Techniques for Building Program Study*, CAADRIA Conference, 2003.

[2] Gershenfeld, Neil, 2005, FAB the coming revolution on your desktop-From personal computers to personal fabrications, New York: Basic Books

[3] Smart Geometry Group, founded in London in 2005, by Lars Hesselgren (KPF), Hugh Whitehead (Foster and Partners), and J. Parrish (Arup), in collaboration with Robert Aish (Bentley Systems). < http://www.smartgeometry.com/>

[4] Howell, Larry L, 2001, Compliant Mechanisms, New York: John Willey & Sons.

[5] Ibid., 2

[6] Ibid., 2

[7] Nagakura, Takehiko, 2005, 4.207 Formal Design Knowledge and Programmed Constructs < http://cat2.mit.edu/4.207/> (08-11-2005)

Sergio Araya Goldberg
Massachusetts Institute of Technology
Department of Architecture
Master of Science in Architecture Studies (candidate)
284 Vassar Street Apt. G1Cambridge MA 02139 US

sergio_a@mit.edu