Parametric Models and Algorithmic Thinking in Architectural Education

Alexandros Peteinarelis¹, Socrates Yiannoudes²
¹,²Technical University of Crete
¹ng5ten@yahoo.com ²soc.yiannoudes@gmail.com

Part of our research and teaching agenda at the School of Architecture of the Technical University of Crete focuses on algorithmic design with parametric models, its methodological characteristics and the study of applied and theoretical work that defined this architectural design thinking. Our work challenges architectural design processes, through the systematic study of parametric models. This paper presents three projects from the undergraduate elective course “Special Topics in Architectural Design”, which took place during the spring semester of 2017, that investigated parametric models for a given architectural problem, inspired, to some extent, by precedents in 20th century architecture where students traced algorithmic design thinking. Although students understood well the concept and function of parametric models and in many cases applied them successfully for their design objectives, several of them did not fully assimilate some critical aspects of computation. This allowed us to determine areas of improvement and points of complete reevaluation in our educational strategy approach.

Keywords: algorithmic thinking, parametric model, computational thinking, architectural education, Frei Otto

INTRODUCTION

It has been more than a decade that architecture schools offer required and/or elective computer-aided design (CAD) courses. In recent years they are becoming increasingly popular amongst architecture students, while as Ahlquist and Menges (2011) argue there is a transition from CAD to computational design in architecture, with one critical aspect being the processing of information algorithmically. Furthermore it seems that in current professional practice CAD and 3D modeling alone are not sufficient to address the need for more economically and ecologically sustainable buildings (Senske 2013), as well as the increasing complexity and demands of architectural projects. Thus it can be argued that teaching algorithmic design requires more than just explaining software interface, its often used commands and design procedures. Algorithmic design courses ought to provoke and cultivate a change in design thinking itself (Senske 2011).

Part of our research and teaching agenda at the School of Architecture of the Technical University of Crete focuses on algorithmic design with parametric models, its methodological characteristics and the
study of applied and theoretical work that defined this architectural design thinking. We attempt to challenge established architectural design processes, through the systematic study of parametric models. This paper presents three projects from the undergraduate elective course Special Topics in Architectural Design, which took place during the spring semester of 2017 that investigated parametric models for a given architectural problem, inspired, to some extent, by precedents in 20th century architecture, where students traced algorithmic design thinking. Taking into account Senske’s observations (2017) we present not only what was designed but, to the same extent, the design process acknowledging that in an undergraduate algorithmic design course the process and the technical aspects are of the same, if not greater, importance than the design outcome. We reflect on our pedagogical strategy, where we put forward and promote the development of logic and method matching the learning and use of the design tool. Within the presented projects we search for areas of improvement, points of complete reevaluation and to what extent students comprehended computational design thinking.

PARAMETRIC MODELS IN THE CONTEXT OF ALGORITHMIC DESIGN

Gaudí used inverted suspension models with strings and birdshot weights to simulate, by analogue means, the route of the forces and thus study multiple variations of the form of the Colònia Güell chapel. Adjusting the model, string length, anchor point location and birdshot weight allowed the architect to calculate experimentally the catenary curves, chains converged in pure tension, which were subsequently inverted to reveal structures in pure compression. Gaudí’s models are physical equivalents of digital parametric models (Burry 2007), since they automatically compute outcomes without consecutive manual evaluations of catenary curve’s parametric formula. Davis (2013; [2]), based on the definition of the parametric equation in mathematics, which is a set of quantities expressed as an explicit function of a number of parameters, states “A parametric model is set of equations that express a geometric model as explicit functions of a number of parameters”. Changing these parameters causes a coordinated overall update, thus allowing designers to constantly adjust and interact with the model, to generate and classify discreet variations, to “search” within a wide range of virtual results.

Defining the parametric model according to mathematical terms and definitions, besides helping to clarify key aspects of the proposed design thinking, allows also to steer away from common views, such as that architectural design has always been parametric, since buildings and cities have always been shaped according to cultural, biological, structural and bioclimatic factors. Along this direction we can distance ourselves from the idea that working with parametric models means contributing to a new architectural style, introduced by PatrikSchumacher (2008; [11]) with the term “Parametricism” in his “Parametricist Manifesto”. To further enhance our understanding of design thinking with parametric models, Davis (2013; [2]) explains that “a parametric model is unique not for what it does but rather for how it was created. A parametric model is created by a designer explicitly stating how outcomes derive from a set of parameters”. Furthermore Jabi (2013) defines “parametric design as a process based on algorithmic thinking that enables the expression of parameters and rules that, together, define, encode and clarify the relationship between design intent and design response”. Thus, we propose that parametric design is the process of describing architectural problems with parameters, and a flexible way to describe digital models, orthogonal or curvy, through visual or text based code, that define associations between elements of the model through specific rules and constraints in the context of algorithmic design.
THE COURSE
The undergraduate elective course Special Topics in Architectural Design is offered to 4th year students to choose as one of two compulsory elective courses of the spring semester. It spans a duration of thirteen, once per week, four hour sessions, without any prerequisite skills requirements, and investigates the application of algorithmic thinking with parametric models in architectural design. The main objectives of this course are to build solid technical skills in parametric modelling with Grasshopper for Rhino (McNeil & Associates; [3], [1]), to establish basic understanding of computational thinking (Senske 2017) and to setup hands on experimentation with parametric models that aim to output answers for specific logic and design problems. The course in 2017 was attended by 20 students with 65% female to 35% male gender balance. Six out of thirteen sessions were used for tutoring Grasshopper and computational thinking (three hours) and lectures followed by discussion with the students (1 hour). The seven remaining were class time with us and the students working in teams on small assignments and the deliverables.

For the spring semester of 2017 the opening argument in the introduction session was that algorithmic design thinking is not a direct result of current advanced digital design tools, nor requires them to be applied (Mateas as cited in Senske 2014), but has specific historical precedents. There is a shift in architectural design thinking, from visual representation of architectural space with drawings and renders, into the systematic representation of design processes, analysis methods and simulations embedded into parametric models. Hence continuing the algorithmic design strategies of the postgraduate spring semester course “Advanced Digital Tools in Design and Fabrication” in 2015 (Peteinarelis and Yiannoudes 2016), we proposed the study of parametric models in relation to their historical precedents, i.e. examples in the history of 20th century architecture in which we traced algorithmic thinking. As Ahlquist and Menges (2011) argue it is important that the critical approach towards computational design extracts knowledge from the historical and practical foundations of design and computation.

To support the selected topic we presented two examples where we traced elements of algorithmic thinking in conjunction with our understanding of parametric modelling. The first one examined brutalist architecture through the work of Marcel Breuer. We chose the IBM Research Center and Forum in La Gaude 1965, the Whitney Museum of American Art / Met Brauer in New York 1966 and Armstrong Rubber Company in New Haven 1968, as case studies, which integrate design concepts that could be embedded and be further processed in parametric models. More specifically we focused on the repetition of identical modular elements that form façades, with distinct rhythmic patterns, which besides being a clear architectural design gesture, was also imposed by factors like prefabrication and manufacturing capabilities, cost and construction challenges. This brought forward the association and differentiation between building elements that could be introduced with a parametric model, which implements geometric rules that describe the aforementioned modules.

The second example was Frei Otto’s analogue experimental form-finding methods, developed at the Institute for Lightweight Structures (ILS), for a new architecture of optimal structures on the basis of a systematic understanding of the self-organizing morphogenetic processes in nature, including the application of minimal detour path systems onto urban networks and settlements, through a close investigation of the natural laws of attraction and repulsion, extension and contraction, and occupation and connection (Otto 2009). Otto used wool thread models, which interacted with the surface tension of water tanks in which they were immersed and then self-organized in a new optimized detour path network. By studying this experiment we had the opportunity to present dynamic simulation methods into parametric modelling environment. The resulting network, which by contrast to the simpler systems of
direct and minimum paths, emerged with a significantly lower total path length and a low average detour factor, set the example of a system in equilibrium state that its entities are not in static isolation, but interact with each other according to values that define attraction repulsion etc.

Students were asked to select similar 20th century cases and trace algorithmic thinking. This “reverse engineering” aimed to promote abstraction and the communication of the findings through free-hand sketches followed by diagrams. The extracted knowledge would later be applied to parametric models which would attempt to transfer and/or multiply the traced design ideas.

**PROJECTS**

**Team01**

Team01 decided to investigate and apply Otto’s analogue computing (Davis 2013; [2]), specifically the minimized paths network into the study of a new spatial network in the university campus of the Technical University of Crete in Chania. Their work consisted of four main steps: the first step was the understanding of the minimized detour network in analogue and computational means for its implementation into a parametric model. The second was the analysis of the existing path network in relation to the buildings and the bus stations of the campus, that would help them form their design strategy, followed by the application of their parametric model to create multiple outcomes, which would attempt to answer to the chosen architectural and logic problem. Lastly an evaluation and comparison of selected outcomes to existing network assessed both their strategy and the outcomes that would be suggested for implementation.

While researching detour paths network they benefited from accumulated knowledge found in Grasshopper online forum (McNeil 2009; [3]), where they found two types of definitions. The first type used only native Grasshopper components and the second employed kangaroo plugin (Piker 2009; [10]) simulations. After studying both, they chose to work with the native components definition (Hristov 2015; [9]).

For the analysis of the existing network they used Urban Network Analysis toolbox (UNA) (Sevtsuk and Kalvo 2015; [8]), a Rhino plugin that offers precise control and flexible solutions to spatial network analysis problems. Reach, gravity and closeness are three measures offered by UNA, which capture how many surrounding points, representing buildings, bus stops etc., can be reached in a defined radius using a given path network from given origin points, the accessibility of the destinations and the average distance from an origin point to all destination points within specified radius along the shortest path of the path network respectively. As shown in Figure 1, the quantification of spatial accessibility, for both pedestrian and bus networks, helped Team01 to locate areas for new bus stops and to move existing ones ac-
Accordingly, to serve inaccessible buildings and to reduce overcrowding in some problematic stops. It also revealed that a cluster of buildings was not easily accessible through the existing network, so they decided to prioritize their accessibility with the new one.

Applying the detour paths network definition on the analyzed masterplan brought forward five main design challenges, the division of the masterplan area into subareas, the automatic creation of charts that would inform the design process, the elimination of very small polyline paths segments, the location of steep areas and the physical intersection of the resulting paths with existing buildings (Figure 2). At this point the team used plugins and implemented their own solutions. A voronoi diagram partitioned the masterplan into three regions. The voronoi cells seeds were three average points of the existing buildings polygon centers that their locations form three discrete buildings clusters. Bumblebee plugin (Mans 2013; [7]) creates interoperability between Grasshopper and Microsoft Excel and was utilized to chart design decisions. To omit very small segments of the resulting polyline paths and to further simplify them was very challenging, so the team resorted to Topologizer component (Piker 2012; [6]), which outputs a clean directed graph or network out of a list of lines according to specified distance threshold. The location of steep areas and the paths-buildings intersections were solved using native Grasshopper components.

After confronting the aforementioned challenges Team01 started generating multiple outcomes, path networks that attempt to solve the problems found with the analysis. They worked varying the number of central connections areas, the number of attraction points and the attraction force. They decided to generate two types of outcomes, one type being path curves simplified with Topologizer and the other raw outcomes from minimal paths definition. Each of these types have two subtypes regarding the addition of new bus stops or not. Selected outcomes were analyzed once again with UNA to create similar metrics with the initial analysis of the existing paths and bus stops. To finalize their study Team01 compared the analyzed outcomes of both types and subtypes to the initial network in order to establish their selection of the solution for the new spatial network implementation, as shown in Figure 3.

Team02

Understanding the current urban and residential condition of Venice, Team02 applied a strategic plan to expand, diffuse and decompress the city’s existing networks, as well as tactics to enhance and balance touristic activity and local residence (the team's initial research showed that buildings currently in Venice house 62% touristic infrastructure and 18% local residence). After studying the city’s urban network, its density and territorial layers, i.e. its urban spaces, pavements, canals and regions (Figure 4), the team proposed four main strategic interventions: the connection of Venice with its surrounding islands, the decongestion of the existing urban network, the creation of a secondary network that would designate...
new areas and landmarks in the city. To achieve its goal the team employed Frei Otto’s minimal detour path system, as well as the compositional principle of continuous extensibility implied by the plan of the hospital that Le Corbusier designed at the southern edge of the city (1965), and the layer superimposition strategies used by Peter Eisenman to design the nearby Cannaregio Town Square (1978). Eisenman’s strategies were used to highlight areas in-between the existing and the new proposed network, while Le Corbusier’s grid was applied to extend the city network through the sea.

The team worked with the city plan geometry generated by the Elk (Logan 2012; [4]) plugin for Grasshopper which processes open source XLS formatted map data from OpenStreetMap.org. Two computational design procedures were used, organized into two separate Grasshopper definitions. The first (Figure 5) aimed to expand the city network through the sea between Venice and its surrounding islands, and the second to determine, highlight and evaluate areas of intervention in the main part of the city. The first procedure started from determining and evaluating specific aquatic and land start and endpoints, i.e. the canal estuaries and squares along the coastline of both the city and its islands, in order to create connections according to two criteria: a distance between connecting points that was smaller than a specified length and an angle of the connecting lines to the north-south axis than was larger or smaller than a specified degree according to the particular island and the type of network it belonged (aquatic or land). After culling the connections that conformed to the criteria, the final aquatic and land lists of connections were input to the minimized detour paths network Grasshopper definition, which utilized a Kangaroo plugin simulation, using the aquatic connections as attractors for the land connection system, to optimize it. Le Corbusier’s Venice hospital grid was then superimposed to align with the land system and a separate Grasshopper definition was employed to extrude building volumes, columns, paths and squares on the resulting plan.

The second procedure (Figure 6) determined the entrance points to the city from the sea, which were used as start and endpoints to create a series of connecting lines for the new city network. Out of these connecting lines several were culled according to two criteria: the minimum distance between connecting points and whether they were within the coastline border of the land of Venice. Subsequently the connecting lines were input to the minimal detour path network Grasshopper definition to pro-
duce a new optimized network according to set attractor points. These attractor points were located on city squares and open spaces that had a surface area larger than a variable number (currently set to 800m²), as well as distance from the basic touristic road that was smaller than a variable number (currently set to 100m). Then several areas of intervention on the network were selected depending on the amount of intersection points on the new network paths (Figure 7). The points that intersected with the existing urban touristic network as well as those that coincided with the attractor points were excluded from the selection. The surface of the intervention area was determined according to the amount of intersection points on the new network paths. Subsequently Eisenman’s grid was introduced and superimposed on the selected intervention areas on the new network, excluding those grid nodes that were outside the coastline border. Several local intervention structures were considered and proposed on the nodes of the grid according to the fragment of the territorial layer of the city on which they were found (private or public building, a public square, a canal, a pavement or private open spaces).

Team03
Team03 investigated the plan of Igualada Cemetary, in Igualada 1994, a winning proposal of the architectural competition in 1984 by Enric Miralles and Estudio Carme Pinos. They focused on two specific areas, the mortuary and the mausoleum, where the design forms an interplay between the structure and the landscape. The students were specifically fascinated by the distinct plan curves of the structures and immediate surrounding areas. They decided to implement a definition that embeds geometric rules and constraints relevant to the plan curves and to examine if the output forms spatial qualities similar to those in Igualada Cemetary. Soon they realized that they needed to override Grasshopper’s directed
The planar curves consist of lines and arcs and the first segment is always a line, with random starting point, direction and length in a defined domain. Arcs result from circles tangent at the end point of last drawn line, which are evaluated along their length according to random angle values. Their center is either to left or right of the end point, defining the direction of evolution of the curve and the overall length depends on the number of chosen loops. In the initial definitions the loop instructions were to alternate the side of the circle center and to vary the length of each line and the radii of the arcs. The process is presented in Figure 8. Such setup output curves, where Team03 detected spatial distribution of open and closed spaces from small and large circle evaluation angles respectively. To produce more complex curves they experimented with paired arcs directions and then introduced three types of behaviors relevant to one, two and four points that affect the evolution of the curve. In case of one point, the line segments steer every \( n \) defined steps towards it and for two points, they orient parallel or perpendicular to the axis formed by the two points, with the orientation imposed according to the distance of the end point of the curve to the axis. Same applies in case of four points, where the axis is replaced by the boundary of the four points along with a point in curve containment check, which makes sure that curve evolution will steer toward the boundary if found outside of it.

Team03 using loops created multiple curves that attempted to represent the two chosen plan parts of Igualada Cemetery. It is the only team of the course that studied and experimented with emergence and after heavy tweaking of all three types of curves generation methods, they proposed through a manual, admittedly not very clear, selection strategy, a vocabulary of curves, with four subtypes encompassing four templates of spatial organization properties. The resulting curves, according to their type and subtype, are illustrated in Figure 9.

**DISCUSSION**

Investigating alternatives in solution design space, reflecting on, modifying and iteratively refining the design outcome are common aspects of most architectural design processes, but we argue that working with parametric models introduces significant
and qualitative differences in workflow and in design thinking. Like the projects that we presented, such design practices seem to point to a shift from the concept of form and its representation, to the concept of formation, i.e. the mechanism, the performative process of form generation and self-organization (Ahlquist and Menges 2011) (Oxman, 2006). This concept seems to have been well understood by the students, since their final presentation showed that they managed to define parametric relationships, author definitions with native components and plugins, setup kangaroo simulations, etc. They also applied specific, per design challenge, logic combined with collected or created information to form coordinated design processes, that would answer to specific architectural problems. In the end they communicated the design process and results successfully with diagrams and plans in clearly structured presentation boards, supported by video and gif images.

Team01 showed a good grasp of the function of the grasshopper definitions they used (though not all created from scratch), in relation to their design objective. This was evident by a fair application of Frei Otto’s minimized detour path system (its computational version), combined with a productive bidirectional process of analyzing and interpreting data which were embedded in the parametric model to assess and analyze targeted features of the design results, thus achieving the campus network optimization goal. The project of Team02 showed that designing with parametric models can solve complex architectural problems, in various scales, from urban strategy plans to urban equipment. Again, having a good grasp of the computational version of Frei Otto’s system, the team implemented a design strategy that allowed for both global and local interventions to take place, using parametric models that were fed with information and constraints from the urban network and the varying spatial qualities of the city. The project of Team03 did not lead, as intended, to a final set of output forms that resembled the plan curves of Miralles and Pinos’ Igualada Cemetary; yet the iterative process they implemented showed a further potential of computational design, i.e. its explorative nature which “is not limited but enabled by the inherent finiteness of algorithmic procedures” (Ahlquist and Menges 2011). The team’s definitions relied on feedback loops, to allow for emergence, i.e. multiple unpredictable outcomes, admittedly having less control over the final form that however complied with the embedded instructions of the parametric model.

CONCLUSION

Compared to our previous experience in the master’s course in 2015 we observed that undergraduate students have less preconceptions and are more open to learn methods, different from the traditional ones (see also Iordanova 2008). We recognized two types of student accumulated knowledge. The first type are students that can navigate through and modify implemented by others Grasshopper definitions, adjust parameters and generate outcomes according to the design process. They can work with computational analysis tools and interpret the results. The second type are students that can additionally design code from scratch and feel more comfortable expressing themselves computationally. This raises the question how these two types occur and how can this dichotomy be resolved. Going through the delivered definitions and analysis files of the course we noticed that first type students did not fully understand the concept of data trees in grasshopper and did not understand completely how many algorithms work and where to apply them. This also became evident with cases in the initial small assignments, where students applied voronoi and octree components to create “interesting” geometry rather than partitioning space, faster spatial queries or data structure representation. Like Austin and Qattan (2016) argue “the issue with algorithmic andragogy within architecture is that quite advanced programming concepts are required to produce quite basic visual outputs”, something that grasshopper components can bypass, only to further contribute to the problem. To address this we are planning for the next course to focus, with newly implemented assignments, more to the
understanding of concepts like data structures, program composition, and logic, aiming to diminish the “production without comprehension” (Senske 2014) (Pea 1984 cited in Senske 2014). We are not sure if these assignments should exclude architecture completely, but we agree that in an algorithmic design course students should, within reasonable expectations, practice solving problems rather than following through a problem (Austin and Qattan 2016), that are up to a certain degree connected to architecture (Iordanova 2008). We are considering to allow more than three hours to the first three course sessions, that establish the understanding of basic computational thinking. In smaller sessions we observed that students needed a long recap of previously explained and understood concepts, which were not applied in practice due to lack of class time. Lastly, we plan to adopt Senske’s (2017) pair programming method, where class time is an active, rather than passive, experience with students working in pairs, one scripting and the other guiding the process.

REFERENCES
Austin, M. and Qattan, W. 2016 ‘I’m a visual thinker’: Re-thinking algorithmic education for architectural design, Proceedings of CAADRIA 2016, Hong Kong, pp. 829-838
Logan, T. 2012, Elk (Grasshopper Plugin)
Mans, D. 2013, Bumblebee (Grasshopper Plugin), neoarchaic.net
Piker, D. 2009, Kangaroo (Grasshopper Plugin)
Piker, D. 2012, Topologizer/Network CleanUp (Grasshopper Plugin)
Senske, N. 2014 ‘Confronting the challenges of computational design instruction’, Proceedings of CAADRIA 2014, Hong Kong, pp. 821-830
Sevtsuk, A. and Kalvo, R. 2015, Urban Network Analysis Toolbox for Rhinoceros 3D, City Form Lab
Zwierzycki, M. 2013, Anemone (Grasshopper Plugin)
[1] https://www.rhino3d.com/