The current issue and full text archive of this journal is available on Emerald Insight at:
https://www.emerald.com/insight/0168-2601.htm

Teaching parametric design: fostering algorithmic thinking through incomplete recipes

Elena Vazquez
School of Architecture, Umeå University, Umeå, Sweden

Received 2 June 2023
Revised 5 September 2023
11 October 2023
Accepted 15 November 2023

Abstract

Purpose – Algorithmic and computational thinking are necessary skills for designers in an increasingly digital world. Parametric design, a method to construct designs based on algorithmic logic and rules, has become widely used in architecture practice and incorporated in the curricula of architecture schools. However, there are few studies proposing strategies for teaching parametric design into architecture students, tackling software literacy while promoting the development of algorithmic thinking.

Design/methodology/approach – A descriptive study and a prescriptive study are conducted. The descriptive study reviews the literature on parametric design education. The prescriptive study is centered on proposing the incomplete recipe as instructional material and a new approach to teaching parametric design.

Findings – The literature on parametric design education has mostly focused on curricular discussions, descriptions of case studies or studio-long approaches; day-to-day instructional methods, however, are rarely discussed. A pedagogical strategy to teach parametric design is introduced: the incomplete recipe. The instructional method proposed provides students with incomplete recipes for parametric scripts that are increasingly pared down as the students become expert users.

Originality/value – The article contributes to the existing literature by proposing the incomplete recipe as a strategy for teaching parametric design. The recipe as a pedagogical tool provides a means for both software skill acquisition and the development of algorithmic thinking.

Keywords Parametric design education, Parametric design, Digital design education, Architecture education, Design pedagogy, Algorithmic thinking

Paper type Conceptual paper

1. Introduction

Digital technologies have changed how buildings are designed and made, shifting alongside how architecture is taught. Parametric design, an algorithmic method to construct designs based on parameters and relationships between these, has increasingly become part of contemporary architecture practice. The literature on parametric design teaching has mostly focused on curricular discussions, descriptions of case studies, or studio-long approaches; day-to-day instructional methods, however, are rarely discussed. As a result, most parametric design teaching has been centered around the vast online community of computational designers that post video tutorials of their parametric scripts. A quick online search reveals a myriad of video channels dedicated to teaching parametric design, where you are invited to follow along the process of creating parametric scripts. However, how can students acquire parametric design skills in architecture schools with strategies that move beyond video tutorials?

© Elena Vazquez. Published by Emerald Publishing Limited. This article is published under the Creative Commons Attribution (CC BY 4.0) licence. Anyone may reproduce, distribute, translate and create derivative works of this article (for both commercial and non-commercial purposes), subject to full attribution to the original publication and authors. The full terms of this licence may be seen at http://creativecommons.org/licences/by/4.0/legalcode

The author thanks Prof. Blaine Brownell (UNC Charlotte) for the insightful discussions around teaching parametric design.

Since submission of this article, the following author(s) have updated their affiliations: Elena Vazquez is at the School of Architecture, UNC Charlotte, Charlotte, USA.

Open House International
Vol. 49 No. 4, 2024
pp. 736-751
Emerald Publishing Limited
0168-2601
DOI 10.1108/OHI-06-2023-0135

Downloaded from http://ftp.nowpublishers.com/ohi/article-pdf/49/4/736/10204942/ohi-06-2023-0135en.pdf by guest on 15 May 2026

In this article, we first conduct a literature review on parametric design pedagogy, articulating the need for developing pedagogical methods to teach parametric design in architecture schools. Current methods for introducing parametric design include providing students with ready-made scripts that they learn to manipulate to create designs. However, there is a difference between modifying a script and creating one; the latter requires students to decompose their designs into steps thus encouraging algorithmic thinking. Algorithms are step-by-step solutions to solve a problem. Recipes, an everyday example of an algorithm, provide instructions on how to achieve a specific goal, such as baking a cake. Parametric models can be thought of as recipes that construct three-dimensional elements using different ingredients: geometry, dimensions and euclidean transformations. Drawing inspiration from baking shows, this article outlines parallels between recipes, algorithmic thinking and the use of those in parametric design education. The first part of the paper is a descriptive study, consisting of a literature review on current methods for teaching parametric design. The second part of the paper is a prescriptive study, where the teaching method of the incomplete recipe is developed.

2. Methods

This paper describes research for bringing forwards a pedagogical tool for teaching parametric design in architecture schools. In this study, the development of the instructional method is considered a design problem. Consequently, the research method implemented relies on design, pedagogy and instructional design research methods. This section provides an overview of the methods implemented, summarized in Table 1.

The selected method is based on the Design Research Method (DRM) by Blessing and Chakrabarti (2009). The method is a framework in four stages: Research Clarification, Descriptive Study I, Prescriptive Study I and Descriptive Study II. In the first two stages, researchers conduct an initial literature review to get a first account of the current situation, followed by an in-depth review. In the third stage, researchers elaborate a description of a desired situation, which represents their approach on how to address different aspects identified as lacking in the first stages. The vision or approach is then tested in the fourth and last stage. This paper addresses the first two research stages: a descriptive study which consists of a literature survey, followed by a prescriptive study, in which an instructional method is proposed.

The descriptive study, shown in the first part of this article, is conducted through a systematic qualitative review of the literature on teaching digital design in general and parametric design in particular. The survey is conducted by searching in several databases with the following terms: ("teaching method" OR "pedagogical approach" OR "teaching strategy") AND ("digital design" OR "computational design" OR "parametric design" OR "visual programming language"). After identifying the main articles in the area, a narrative

Research Stages and Methods				*Reference*
Research Clarification → *Literature review*	*Descriptive Study I* →	*Prescriptive Study I* → *The incomplete recipe method. (Developmental evaluation).*	*Descriptive Study II* → *Testing the method (Summative evaluation)* *Future studies*	*Blessing and Chakrabarti (2009)*
		Scope of this article		

Table 1.
Research methods

Downloaded from http://ftp.nowpublishers.com/ohi/article-pdf/49/4/736/10204942/ohi-06-2023-0135en.pdf by guest on 15 May 2026

analysis was conducted, identifying themes. The outcome of the descriptive study is a list of requirements for a pedagogical tool for teaching parametric design, based on the learning outcomes identified as critical.

The prescriptive study is centered on proposing a new instructional method to teach parametric design, addressing some of the areas for improvement found in the literature of digital design education. The prescriptive study is based on a research method in pedagogy called Action Research, where (teaching) practitioners understand and improve their own practice by centering their actions and reflections on them (Nind *et al.*, 2016). This method relies on educators to draw on their own experience and provide hypotheses on how to improve their practices (Corey, 1954). In the DRM method, the prescriptive study is followed by a second descriptive study that aims to implement and test the proposed approach. While the method was developed, implemented and refined during a semester-long project, it was not thoroughly tested yet, which will be done in further studies.

The prescriptive study is the design or development of an instructional material. As such, we relied in current methods in the literature of instructional design. Traditional models of instructional design such as ADDIE (analyze, design, develop and evaluate) are linear in nature; However, current discourse postulates that the process is not linear, rather, developmental and iterative (Crawford, 2004; Willis, 1995). As such, the instructional method presented in this manuscript undertook iterative cycles of evaluation, based on the constant revision of the literature described in the next section and in day-to-day student interaction. This type of evaluation is called developmental evaluation, which involves responding to changes based on theory, testing and own experience in implementation (Williams *et al.*, 2011). The developmental evaluation and refinement took place during a semester-long course titled "Performance-Based Parametric Design".

In terms of the software selected for the development of the teaching method, parametric design is commonly associated with Visual Programming Languages (VPL) embedded in modeling software. Novice architectural students appear to obtain better results with VPLs than with Textual Programming Languages (Celani and Vaz, 2012), although they can be easily combined to extend the functionality of the former. Software applications of VPLs include Generative Components (Bentley Systems), Dynamo (Autodesk) and Grasshopper (McNeel and Associates). A survey by Touloupaki and Theodosiou (2017) found that the most popular software for performance-driven design optimization was Grasshopper for Rhinoceros 3D, followed by Dynamo for Revit. Dynamo and Grasshopper are also known for their added functionalities through plugins. For instance, there are currently close to a thousand plugins available for Grasshopper (foodforhino.com), some of the most popular ones being Ladybug and Honeybee (environmental performance analysis), Karamba (structural analysis) and Pufferfish (complex shape manipulation). Both software programs, however, follow a similar logic in the construction of geometry. In the two software, the user designs with nodes representing geometry/Euclidean transformations/data operations and how they are connected. Because of the preponderance of these types of parametric design software, this study is centered on developing strategies to teach how to construct parametric models with their logic.

3. Results

3.1 A review of teaching parametric design

Before diving into the literature on teaching parametric design, it is useful to set a working definition of parametric design. Parametric design has adopted several definitions as it has become increasingly part of architectural education and practice. Parametric design is described as a method with an algorithmic approach that contains parameters and rules that codify the designers' intent (Casini, 2021). It has also been defined as a process based on the

Downloaded from http://ftp.nowpublishers.com/ohi/article-pdf/49/4/736/10204942/ohi-06-2023-0135en.pdf by guest on 15 May 2026

exploration and editing of associative relations in a geometric solution space (Oxman, 2017; Woodbury, 2010). Other authors describe parametric models as a design representation of geometries that have properties that are fixed and others that can vary (Hernandez, 2006). What these definitions have in common is that they describe a design method based on constructing geometry with parameters (fixed or variable) and the rules between these. For the purposes of this article, parametric design is defined as a method to construct designs with parameters that convey algorithmic logic and rules. Furthermore, for this study, the working definition of parametric design is limited to its application as a digital design tool.

Having defined parametric design, the following is a survey on parametric design education. For at least a decade now, parametric design has been a part of the curricula of some architecture schools and has even been the subject of graduate degrees in architecture. The question that arises is then, how has it been taught and what methods have been developed? The following paragraphs summarize pedagogical approaches to teaching parametric design found in the literature, organized by themes:

A trend in the literature is to use non-computerized methods in teaching parametric design. For instance, Alalouch (2018) proposed an analog method to encourage parametric thinking in students. The method follows the problem-solving strategy of breaking down a complex problem into pieces or steps. With the approach, students follow a sequence of steps starting with designing a generative base plane, articulating a rule to the base plane, layering the planes, applying the rule, composing the form and evaluating it. The process is an iterative one that students can engage with physical planes and form structures based on rules for the initial plane and how they repeat to form a building. The analog method proposed by the author follows similar steps found in parametric design scripts of simple parametric pavilions or parametric towers. Not relying on specific software, students learn the process of designing through a set of rules for form manipulation.

Howe (2011) also highlights the importance of using analog methods to introduce scripting languages. The author suggests starting the process with plane manipulations through folding material to understand how simple rules create geometry. Students are then asked to populate a surface with variations of the single unit, using parametric software. Finally, they are asked to reverse engineer the forms to ensure that the designs can be fabricated. Computing without computers can be an effective way of introducing parametric thinking in students. Nevertheless, performing complex form manipulation as well as conducting simulations or analyzing building data might require the use of parametric modeling software.

A second theme is to incorporate material constraints or alternate between digital and material explorations in teaching. Abdelmohsen and Massoud (2021) propose a material-based computational framework for teaching parametric design. They argue that parametric design education is typically focused on universal strategies where students blindly follow scripts and procedures, thus negating a bottom-up form finding embedded in material logic. The main idea is to adopt a two-step process that begins with material exploration and extracting embedded parameters, followed by a second step that synthesizes the parameters into a parametric model. The students need to deduct material rules and constraints from hands-on prototyping, to later translate those into algorithmic constructs. The methodology proposed intends to encourage students to develop their parametric models from the bottom up, based on material logic. Howe (2011) also suggests alternating between physical and digital models, to encourage students to think about how the designs can be fabricated and thus, ensure buildability.

Another concern among scholars in digital design education is how to incorporate digital tools in design studio settings, or how a digital studio can be conceived. Oxman (2008) proposes an experimental design studio, where the exploration of digital architecture concepts is presented as a pedagogical framework for digital design education. The goal in

Downloaded from http://ftp.nowpublishers.com/ohi/article-pdf/49/4/736/10204942/ohi-06-2023-0135en.pdf by guest on 15 May 2026

the described approach is to integrate concepts, experimental methods and digital literacy. The author argues that digital design has transformed the concept of form into the concept of formation: processes, rather than final design ideas, become relevant. Instead of following the traditional sequence of site analysis, program definition, conceptual design and so on, a principle that transcends conventional design methods is proposed: no program or specific site is needed. Instead, teaching is model-oriented and based on material exploration. The didactic process proposed by the author has four steps: conceptualize a digital material, define a digital design model, select a context and develop a taxonomy that describes the process.

An experience implementing parametric design in a studio setting can also be seen in the work by Schnabel (2013). The author proposes implementing digital design from very early in the design process. In the first step, the site analysis stage, students were encouraged to focus on two key parameters they want to address. These parameters should thereafter inform their initial rules that address the design problem. An intense three-week training on digital tools followed, where students gained software literacy to implement their design ideas. In the following stages, students scripted, designed and fabricated their digitally created design. The study centers on how to structure a parametric design studio; the issue of software literacy acquisition is not discussed in detail.

Agkathidis (2015) also describes an approach of introducing generative design in an architectural studio setting. The author proposes a framework composed of three stages: Analysis, Morphogenesis and Metamorphosis. The analysis stage consists of data collection and rule definition, including defining the projects' context, program, material and so on. The morphogenesis stage targets the generation of abstract prototypes, and the metamorphosis stage includes translating those into architecture. In their approach, the authors also encourage switching between physical and digital model-making, to reduce unneeded geometric complexity.

Other pedagogical approaches include Agirbas (2018), who proposes using metaphors in parametric design education, to encourage students to create their designs with deductive reasoning. The method was implemented through a 14-week semester elective course, in which students first were introduced to parametric design software (Rhinoceros and Grasshopper) and then had to select a metaphor to create their own architectural design. The authors argue that the use of metaphors in their project provided a framework for exploration which guided and limited the definition of algorithmic rules (parameters) in their models. The software acquisition strategy described by the authors entailed providing students with ready-made scripts, that they could start modifying. The authors are concerned with a methodology that can encourage students to create their own designs; there is little discussion, however, on software acquisition strategies.

A study by Maldonado (2014) described a method for teaching digital modeling using recipes. While it is not a method for parametric design in particular, it is included in this literature survey as it is a precedent for using recipes in design education. The author described a course in which students were provided with recipes for spatial studies such as "recipes for a carved, sculpted, space" and "for a folded, tiling space". These recipes were provided as step-by-step instructions for modeling these structures in Rhinoceros for no specific site. The main idea is that students should first follow and learn the recipe, and then they can use it and adapt it to their own needs. Much like cooking recipes, the authors argue, these digital recipes are procedural diagrams that can produce substantially different outputs when adapted by students. In line with the study, we see the potential of recipes as a powerful pedagogical tool to develop not only algorithmic thinking but to introduce parametric modeling skills for architecture education.

A group of studies in parametric design education is centered on describing the current structures of architecture schools' curricula. For instance, Cudzik and Radziszewski (2019) describe a case in which parametric design education is divided into two main courses called

Downloaded from http://ftp.nowpublishers.com/ohi/article-pdf/49/4/736/10204942/ohi-06-2023-0135en.pdf by guest on 15 May 2026

descriptive geometry and parametric design and advanced Computer Aided Design (CAD) and additional electives and workshops. However, there is little discussion on teaching methodologies and approaches to the subject. On a similar note, Romcy *et al.* (2015) discuss the curricula of different architecture schools and their experiences introducing parametric design. The authors also characterize the challenges that entail shifting from a single final design in traditional studio settings to the design of a process in parametric design. In other words, using parametric design, students typically design a family of possible design solutions, rather than a single design solution, which is the case in traditional design studios.

Other authors present case studies of introducing parametric design in short workshops, describing the challenges and opportunities found. For instance, Wanan (2016) describes a design-to-fabrication experience in which a group of students designed and built a waffle-structure pavilion. Hanna and Turner (2006) argue that one of the frustrations that students encounter when exposed to parametric tools is related to the explicitness required to build a model. In the design process, however, sketches and schemas do not require high definition, particularly in the early stages. The authors, however, point out that the procedural nature of parametric modeling where the designers define the steps that rule desired geometries presents an opportunity to ground those steps to the reality of the project. That is, defining the algorithmic relationships in the design concerning project constraints, fabrication, site, material and so on.

One of the challenges in parametric design education appears to be helping students reach the application stage, i.e. moving beyond learning how to construct parametric models to applying this knowledge to architectural design problems (Agirbas, 2020). To address this issue, authors have proposed a methodology using ready-made parametric scripts, as a tool for design exploration to create digital sketches. The issue of acquiring the knowledge of how to construct the scripts, however, is not addressed. Implementing ready-made scripts can guarantee a finished architectural design, in the same way that a ready-made baking mix can ensure a finished cake. From a pedagogical point of view, however, it might not ensure that students will develop algorithmic thinking and be able to construct their own scripts.

Overall, there is limited literature on pedagogical strategies for parametric design education. Table 2 synthesizes the studies that propose methods for teaching parametric design.

Author	Approach
Alalouch (2018)	Analog "serial of planes" method based on defining rules for manipulating planes
Howe (2011)	Method based on introducing algorithmic concepts through simple physical plane manipulations
Abdelmohsen and Massoud (2021)	The material-based framework, a bottom-up method to construct parametric designs starting from material constraints
Oxman (2008)	Studio setting – an experimental digital studio in four steps: conceptualize a digital material, define a digital design model, select a context, and develop a taxonomy
Schnabel (2013)	Studio setting – encourage students to focus on the parameters and rules that conform their designs
Agkathidis (2015)	Studio setting – a method in three stages: Analysis, Morphogenesis, and Metamorphosis
Agirbas (2018)	A method based on using metaphors to guide parametric models
Maldonado (2014)	Studio setting – A method for teaching digital modeling and providing a framework for early design explorations using recipes
Agirbas (2020)	Provide students with ready-made scripts, students make "digital sketches"

Table 2. Literature review summary

Downloaded from http://ftp.nowpublishers.com/ohi/article-pdf/49/4/736/10204942/ohi-06-2023-0135en.pdf by guest on 15 May 2026

Taken together, the studies highlight the importance of encouraging students to think algorithmically about their designs, i.e. to define a set of rules that confirm processes of formation. Another aspect of parametric design highlighted by the authors is that students need to adopt a problem-solving strategy that breaks down the design problems into parts which can be translated into geometric operations. Several studies focus on analyzing case studies of design-to-built projects where parametric design is one of the components of the project. Studies suggest that there is a growing concern among scholars on how to introduce digital tools in a studio setting in architecture school (Oxman, 2008; Agkathidis, 2015; Schnabel, 2013). Studio courses are traditionally the core component of architecture curriculums around the world, there is therefore a need to develop and propose strategies for incorporating digital technologies. Few studies, however, propose day-to-day instructional materials that tackle software literacy. Additionally, there is less discussion of how to teach parametric design in semester-long courses, which provide more opportunity for in-depth explorations by the students into digital design.

In terms of skills acquisition, a common strategy is to provide ready-made scripts to students for them to start modifying them and understanding the relationship between the scripts, variables and output. However, there is a concern that ready-made scripts can only be powerful design tools insofar as they are profoundly understood and can be thus adapted to custom needs (Riekstins, 2018). There is a difference between being able to use the script and being able to design the script. Generally, there appears to be a need to develop strategies for teaching parametric design in ways that encourage algorithmic thinking. There is a need for day-to-day instructional materials and methods, rather than general approaches to the subject, to better equip educators with tools for teaching parametric design and its associated software.

3.2 Criteria for development of the instructional method

So far, in this article we have argued that there is a need to develop instructional materials and methods for teaching parametric design. The hypothesized requirements for the instructional method are summarized as following and expanded in the paragraphs below.

(1) An indirect teaching methodology that emphasizes problem-solving, as defined by Rüütmann and Kipper (2011).

(2) A method centered on the acquisition of parametric design software literacy.

(3) A method that encourages parametric/algorithmic thinking (Alalouch, 2018)

The first criterion is that it should be an indirect teaching method. There are two broad types of learning outcomes, as defined by Rüütmann and Kipper (2011): facts, rules, action sequence – commonly taught in direct instruction format such as a lecture – and concepts, patterns and abstractions – taught through strategies that emphasize problem solving. Learning to construct parametric models implies higher-level cognitive tasks including analysis, synthesis and decision-making. Indirect instruction strategies such as project-based lessons are then required teaching strategies for students to acquire digital design literacy. Following video tutorials in parametric design, while requiring student engagement, is a direct teaching strategy, such as a demonstration and presentation. A pedagogical strategy for abstracting form and decoding the patterns required to build it parametrically must entail an indirect teaching method which encourages problem-solving.

The second criterion is that the instructional material should address software literacy. There are several analog methods to encourage algorithmic thinking and computing without computers (Knight, 2012; Alalouch, 2018). Less known are methods that address software literacy. The most widespread methods for teaching parametric design rely on either

Downloaded from http://ftp.nowpublishers.com/ohi/article-pdf/49/4/736/10204942/ohi-06-2023-0135en.pdf by guest on 15 May 2026

following step-by-step tutorials or using ready-made scripts. Following along with video tutorials, while a valid teaching strategy, is not a method that emphasizes problem solving as described above. Additionally, using ready-made scripts might be a better strategy to implement once students have a solid foundation and can use them for solving specific design problems.

The third criterion is that the instructional method should encourage parametric or algorithmic thinking. An algorithm can be broadly thought of as a procedure that allows one to solve a problem with specific instructions. Algorithmic thinking, consequently, is defined as the ability to analyze problems, specify them, find and construct a series of actions that are adequate to solve it and improve upon the constructed solution (Futschek, 2006). In other words, it is the process to solve a given problem by developing a sequence of steps (Kátai, 2015). Because algorithmic thinking is about developing a solution, it is also a creative process; for every problem, there are multiple possible solutions. Algorithmic thinking, in the framework of architecture schools, can be thought of as a vessel for creative inquiry, rather than merely a technical skill: Algorithmic thinking is the process of designing a solution. Algorithmic thinking is not associated with a particular software literacy; however, we can use specific software to develop algorithmic thinking. The goal would be that when students leave the classroom, they are able to implement that algorithmic thinking of finding the steps to construct the design problem in other software and scenarios.

Finally, it is important to describe the context in which the instructional method is proposed. The incomplete recipe method was developed and refined in the context of a semester-long course on Parametric Design. The instructional material is an assignment thought of as an in-class exercise that would take no more than 80 min from start to finish. It is intended as an instrument for improving students' modeling skills in parametric design software.

3.3 The approach: recipes and algorithms in teaching parametric design

The previous section defined the criteria and context for developing the instructional method. What follows is a description of the development of the method, starting with an examination of recipes as an analogy for algorithm, followed by a discussion of the logics of parametric design that lay the foundation for the instructional material and a description of the teaching tool and the developmental evaluation and refinements done through the semester-long course. Finally, the instructional method is discussed in the context of a semester-long seminar on Parametric Design.

3.3.1 From algorithms to recipes. Thus far, we have argued that algorithms present a way to solve a problem with defined steps. In parametric design, these steps usually describe the construction of three-dimensional elements, from objects to buildings. Algorithms can also be thought of as recipes, or recipes as algorithms. Using cooking as an analogy to algorithmic design is not, however, a new idea. Tedeschi and Lombardi (2018) argue that "An algorithmic software could be imagined as a blender and an algorithm as a recipe" (pg. 36). The ingredients, in this case, are the dimensions, constraints, geometries and so on; the algorithms describe how they come together. Parametric design recipes can then depict steps necessary to construct parametric models, describing ingredients and operations.

Recipes, or step-by-step instructions instead of finished depictions, in architectural design is a concept that has been introduced previously. For instance, Alberti's three treaties are not illustrated but rely on simple diagrams and a series of computational instructions (Carpo, 2011). Introducing recipes for teaching parametric design also draws upon the notion that algorithms are, indeed, a form of a recipe. As Woodbury (2010) explains, algorithms "spell out practical tasks of representing and manipulating design objects" (p. 85). It is perhaps no wonder that one critical command in Grasshopper, a VPL for Rhino, is "bake," which instantiates geometry into the Rhino workspace in its current state (Mode Lab, 2015).

Downloaded from http://ftp.nowpublishers.com/ohi/article-pdf/49/4/736/10204942/ohi-06-2023-0135en.pdf by guest on 15 May 2026

In the context of architectural practice, the problem being addressed is a design one; the design of a window, facade, or building. Consequently, thinking algorithmically, the solution often involves specifying a series of steps to construct the geometry, with the defined constraints and relations between the parts. For instance, if we consider the design of a chair using algorithmic thinking, one must develop the sequence of steps that lead to its conception, from the main geometry to each of its parts. How the steps are defined in parametric design can vary enormously, in ways that are more or less connected to the constructive nature of the elements being designed. While it is true that when we design in a virtual space, there is no gravity or material inherently present, parametric models can be constrained to material realities. For instance, the design of a chair that will be built using a waffle structure can be constructed with planes and joins based on the thickness of the materials. Or, if the chair will be 3D printed, the geometries can be defined taking into account minimizing the required support material. The degree to which the algorithm (group of steps) proposed as a solution to the design problem is connected to material constraints is entirely up to the designer.

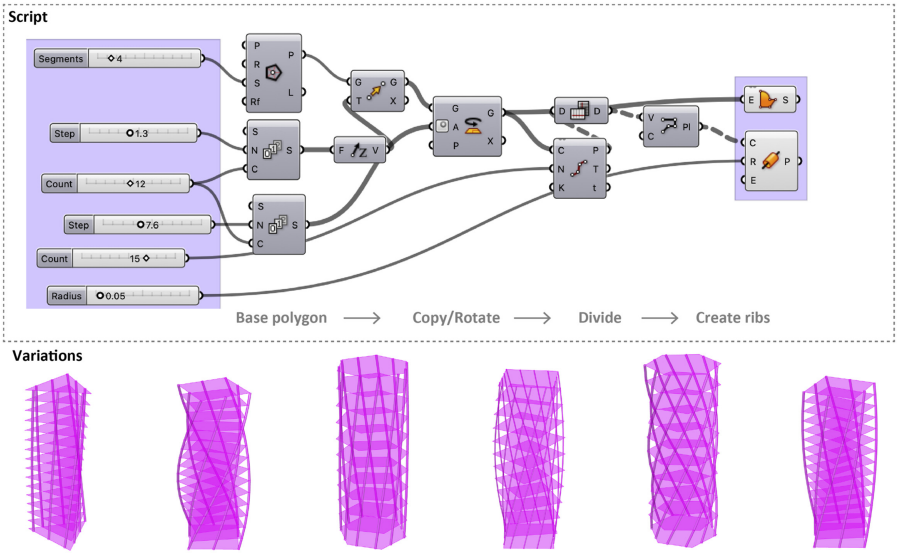
3.3.2 Recipes in Visual Programming Languages (VLPs). The way VPL parametric software works is that designers must explicitly state the components and the relationship between those components, which as noted by Aish and Woodbury (2005) runs counter to the inclination towards ambiguity preferred in the design process. In the design process, the representation of ideas typically follows the definition of them, from initial sketches to detailed floor plans. Parametric modeling, however, requires constructing explicit geometries, with defined dimensions and their topological relations. The design process in parametric design, is then, not about the finished design, but rather, about defining the steps that will comprise the design system. However, the definition of steps in constructing parametric models is not without creativity, as designers can adopt several paths to achieve a similar goal. The way the steps are defined ultimately determines how the model is constructed, its constraints, and therefore, the design solution space – a group of all possible designs with a single parametric model.

Constructing geometry in the two most well-known parametric software, i.e. Dynamo and Grasshopper, involves decomposing the geometry into a series of operations, starting from the most basic geometric unit. Figure 1 depicts a simple Grasshopper script to design a parametric tower and a sample of design alternatives from the design solution space generated with the script. The sequence of constructing the geometry is to (1) create a base polygon, (2) copy and rotate it using a series of numbers, (3) divide the polygons into points and (4) connect the points to create ribs. Learning to use these tools to design then entails thinking about what the steps are to obtain the desired model.

3.3.3 The instructional method: the incomplete recipe. In the great British baking show, an amateur baking competition, one of the challenges the contestant must face is the dreaded technical challenge. All the participants receive a simplified set of instructions to bake the same cake, croissant, or similar in the technical challenge. Depending on the difficulty level of the bake and when the challenge takes place, the recipe can become increasingly fragmentary. For instance, a recipe might state "make the dough" as a step on a semifinal near the show's end. This exercise can be translated as an assignment in parametric design courses, where students receive incomplete recipes to parametric scripts of increasing levels of difficulties, without the time pressures of actual competition, of course.

The idea is quite simple: Students are provided with recipes to construct parametric models. These recipes are step-by-step instructions that guide the construction of parametric scripts. These recipes can be very detailed for beginner students and more general or "incomplete" for advanced students. Providing cues in incomplete recipes helps build what Newell and Simon (1972) call the structure in problem-solving. The presence of structure determines if students will either search indefinitely for the correct answer or search for a solution systematically.

Downloaded from http://ftp.nowpublishers.com/ohi/article-pdf/49/4/736/10204942/ohi-06-2023-0135en.pdf by guest on 15 May 2026



Source(s): Created by the author

Figure 1.
Script to create a simple parametric tower

Therefore, the parametric recipes can provide text or visual clues to indicate what are the steps required to build specific parametric models.

What follows is a description of example assignments developed with the incomplete recipe approach. As a first example, let us consider the recipe for designing a solar shading device consisting of vertical louvers on an existing window, shown in Figure 2. This incomplete recipe is one that can be given to students in the initial process of learning parametric software, because it provides many hints and instructions. The steps will include (1) obtaining the top and bottom curves of the window, (2) dividing these curves into points, (3) creating a line between the two points, (4) extruding the curve to create the louvers, (5) defining the thickness of the louvers and move them to have a rotation axis in the middle and (6) rotating the louvers according to their distance to a point, remapping the values as needed. Note that some words are highlighted in bold, representing specific nodes to be used in parametric modeling software. Since this is an assignment for novice users of parametric design software, hinting at every node or component that needs to be used becomes very helpful.

Other examples of the incomplete recipe assignment are shown in Figure 3. Figures 1–3 presents recipes for arranging data structures, asking students to arrange a list of points in a certain way. In this assignment, the highlighted word is arrange, so that students infer that the component they are looking for is probably in the group that deals with lists and data structures. Figures 2 and 3 shows a section of an incomplete recipe assignment dealing with making window perforations aligned with a tilted facade plane. Providing visual and verbal cues of the steps that might be taken to achieve a specific goal helps students decompose the geometry into what steps need to be taken to arrive at the desired goal. Because there are many ways to arrive at the same result when constructing a parametric script, it is essential to provide textual and visual information that allows students to visualize the goal, without explicitly stating which node to use.

Figure 4 shows image sequences used in recipes for surface analysis and for constructing a curved wall. The recipe for surface analysis shows the sequence of analyzing a surface and coloring the surface fragments by height or by relative position in the *Z* axis. The middle image

Downloaded from http://ftp.nowpublishers.com/ohi/article-pdf/49/4/736/10204942/ohi-06-2023-0135en.pdf by guest on 15 May 2026

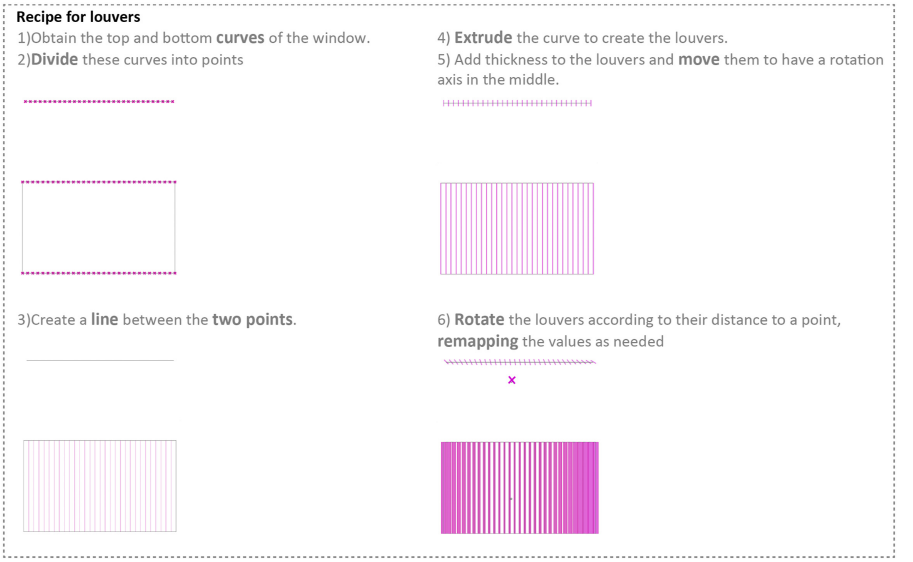


Figure 2.
A recipe for louvers

Source(s): Created by the author

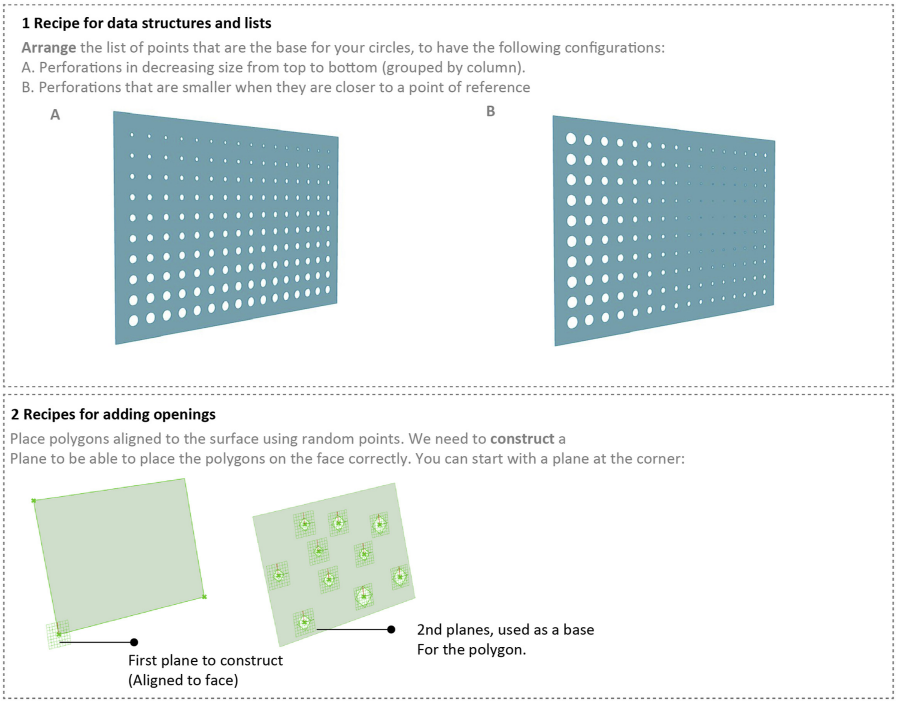
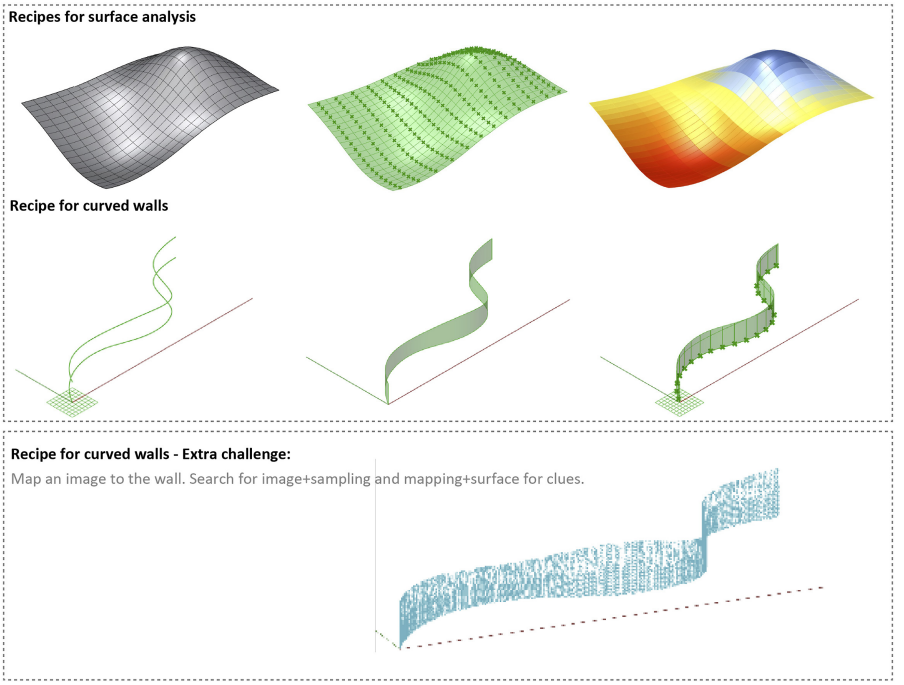


Figure 3.
Parts of recipes for organizing data structures (3–1) and for adding openings (3–2)

Source(s): Created by the author

Downloaded from http://ftp.nowpublishers.com/ohi/article-pdf/49/4/736/10204942/ohi-06-2023-0135en.pdf by guest on 15 May 2026



Source(s): Created by the author

Figure 4.
A recipe for analysis and visualization and a recipe for a curved wall (top). An example of an extra challenge (below)

suggests finding a center point for each surface fragment to obtain the necessary data (in this case, the Z value of each point). Image sequences depicting intermediate steps become helpful as signposts on the path so that if students become lost, they can retrace their steps to the last known place. Image sequence also helps strengthen the skill of breaking down the geometry that often entails constructing a parametric script. When constructing a parametric script, it is often necessary to use the universal problem-solving strategy of "divide and conquer," which means defining the sequence of steps needed to construct the targeted design.

The recipe assignment is designed to be flexible enough to be used in teaching a group of students with different levels of expertise in the subject. It is often the case that there are different levels of exposure to parametric design software in a classroom. While some students might have explored parametric design tools in internships or other classes, others may come to the class without prior knowledge. Adjusting to different levels becomes particularly relevant for mixed groups of undergraduate and graduate students. Dividing the recipes into parts allows for adding extra challenges to advance students, such as the example shown in Figure 4. In the assignment shown, students are asked to map a figure into a surface and create perforations with circles according to the values extracted. As these are extra steps, only a few indications are provided.

The instructional method undertook developmental evaluation and refinement throughout a semester-long course. One refinement that took place is that the wording of the recipes is adjusted to the actual names of the commands in the software. Directly specifying the names of the commands that should be utilized in constructing the models provides a solid guide, especially for novice users. Another refinement that took place for the method was adding more visual cues. Students appeared to respond positively to a combination of image and text.

Downloaded from http://ftp.nowpublishers.com/ohi/article-pdf/49/4/736/10204942/ohi-06-2023-0135en.pdf by guest on 15 May 2026

Finally, another adjustment was to leave space for playful exploration at the end of the recipes. This was done by suggesting new ways in which the parametric model could be modified, either by changing the base geometry, or adding new constraints.

3.3.4 Beyond the recipe and design challenges. The incomplete recipes proposed here are thought of as short in-class assignments, which were provided. As such, they can be paired up with week-long design challenges where students can apply what they learned to a subject of their interest. A recurring concern found in the literature was related to the lack of opportunity for students to create their own designs in parametric design courses. To address this, students can be provided with a week-long design challenge, where they use any of the recipes learned through the course to create their designs. These week-long challenges present an opportunity for students to pursue their interest in parametric design since this is a tool that can be applied across scales and fields. As an example, some students might be interested in parametric modeling in urban design, while others in shoe design. Providing an open-ended assignment can then encourage students to start using parametric tools for creative design explorations.

In addition, the proposed design challenges can address different common implementations of parametric design once students have mastered the basics. For instance, a design challenge can focus on form finding, where students need to use physics simulation plugins. Students will then choose a recipe from the module on form-finding and adapt it to design a building component or object of their choosing. Another design challenge can be focused on environmental performance, where students can select what performance criteria drives their design. An example of how to articulate incomplete recipes and design challenges is shown in Figure 5. The figure shows a schema of implementing the recipes and design challenges in a semester-long topical course on parametric design, alternating both. The main idea behind the schema is to alternate more structured skill acquisition assignments with open-ended design explorations, based on the recipes (structured) and design challenges (open ended).

4. Discussion

This article describes research to develop an instructional method for parametric design education. The first step in the research was a descriptive study in the form of a literature review, followed by a prescriptive study, where the incomplete recipe assignment is proposed.

The literature survey highlighted themes in digital design education. One recurring theme was an interest in incorporating material constraints into parametric and algorithmic modeling. Researchers proposed strategies such as a bottom-up approach that starts with material explorations and iterative cycles of physical to digital in a digital design course. A second theme is to incorporate analog making to introduce parametric processes to students. The idea is to induce algorithmic thinking, with paper and simple folding/cutting operations, defining rules for material manipulation. More general approaches to teaching a digital studio include moving from a traditional architectural design process to a site-less model-based design process.

In general, previous studies have not proposed specific strategies for acquiring software literacy in parametric design. One reason might be that software tends to rapidly evolve,

Figure 5.
A schema for alternating incomplete recipes and design challenges in a semester-long course on parametric design

Source(s): Created by the author

Downloaded from http://ftp.nowpublishers.com/ohi/article-pdf/49/4/736/10204942/ohi-06-2023-0135en.pdf by guest on 15 May 2026

so educators might prefer not to focus on a single specific software. Nevertheless, in a digital design studio or a parametric design course, educators select and utilize specific modeling software. Therefore, there is a need for developing methods to teach them. A second reason might be that educators might be centered in students developing algorithmic thinking, which goes beyond software literacy. However, with this study, it is argued that parametric modeling software can be used in teaching parametric design as a way to encourage algorithmic thinking.

The incomplete recipe assignment was designed to address the issue of software literacy in parametric design education, while encouraging algorithmic thinking in students. It was also developed as an indirect teaching methodology centered in problem-solving. The instructional method needs to be formally tested to test these assumptions; however, valuable insights can be gathered from the semester-long process during which the instructional material was developed and refined. The discussed improvements and adjustments to the instructional method provide insights into how to better guide students with the recipe while still providing adequate levels of difficulty.

Some of the learnings from implementing the incomplete recipe assignment during the course are that first, this type of assignment encourages students to start coding at early stages in their learning process. The first incomplete recipe assignment can be provided in the first parametric design class, after a 20-min introduction to the software interface. Assuming students have previously used 3D modeling software, they are able to start constructing parametric models from the beginning of the course. Second, an advantage that recipes have is that once they are mastered, students can adapt them to different contexts and needs. In design challenges, students have adapted the recipes developed to design more complex parametric models. One challenge observed, particularly in the initial sessions, is to find adequate levels of difficulty within the assignment, because of the difference in software literacy among students when they start the course.

5. Conclusion

This article set out to examine current pedagogical strategies for teaching parametric design and argue for the need to develop methodologies for educators. Current strategies include ready-made scripts, which can be limiting for students to learn how to think algorithmically, i.e. identify and develop appropriate steps to construct parametric designs. In addition, educators have proposed analog strategies for inducing algorithmic thinking; digital and software-based strategies, however, remain underdeveloped. Recipes—a type of algorithm—, can provide a useful pedagogical strategy for building digital skills and developing algorithmic thinking in parametric design education. Providing students with incomplete recipes that match their software literacy can be used in parametric design education as an alternative to ready-made scripts and video tutorials, which are prevalent in parametric design education. Incomplete recipes encourage students to start creating their scripts right from the start, they provide structure for problem-solving and can be used as starting points for creative exploration.

This manuscript was limited to reviewing the literature on parametric design education and arguing for recipes in parametric design as an instructional method. The recipe method, however, needs to be further tested. Comparing incomplete recipes and other commonly used strategies such as ready-made parametric scripts would provide insights into their strengths as educational resources. Future work could implement the proposed method to assess whether it leads to faster software literacy and the development of algorithmic thinking. Notwithstanding these limitations, this study contributes to the literature on parametric design education by proposing a pedagogical strategy: the incomplete recipe. The recipe as a method aims to provide a means for both software skill acquisition and the development of algorithmic thinking.

Downloaded from http://ftp.nowpublishers.com/ohi/article-pdf/49/4/736/10204942/ohi-06-2023-0135en.pdf by guest on 15 May 2026

Competency in parametric design only comes with practice. However, a challenge for instructors is to find adequate assignments that are challenging enough without driving the learner away. The recipe-type assignments described here aim to allow students to practice parametric design without having to emulate a step-by-step tutorial for their first parametric scripts. As the lessons progress, students can fill in increasingly larger blanks in the recipe, eventually learning how to build their recipes for their parametric architectural projects.

References

Abdelmohsen, S. and Massoud, P. (2021), "A material-based computation framework for parametric design education", *Open House International*, Vol. 46 No. 3, pp. 459-475.

Agirbas, A. (2018), "The use of metaphors as a parametric design teaching model", *Design and Technology Education: an International Journal*, Vol. 23 No. 1, pp. 40-54.

Agirbas, A. (2020), "A teaching methodology for parametric design: a case study with parametric bench", *SIGraDi 2020 Proceedings of the 24th conference of the Iberoamerican Society of Digital Graphics*, pp. 720-725.

Agkathidis, A. (2015), "Generative design methods implementing computational techniques in undergraduate architectural education", *Proceedings of eCAADe*, Vol. 2 No. 33, pp. 47-55.

Aish, R. and Woodbury, R. (2005), "Multi-level interaction in parametric design", *Smart Graphics: 5th International Symposium, SG 2005*, Germany, August 22-24, 2005, Frauenwörth Cloister, pp. 151-162, Proceedings 5, Springer Berlin Heidelberg.

Akos, G. and Parsons, R (Mode Lab) (2015), "Talking to Rhino. Talking to Rhino | the grasshopper primer third edition", available at: https://modelab.gitbooks.io/grasshopper-primer/content/1-foundations/1-1/3_talking-to-rhino.html (accessed 15 November 2022).

Alalouch, C. (2018), "A pedagogical approach to integrate parametric thinking in early design studios", *Archnet-IJAR: International Journal of Architectural Research*, Vol. 12 No. 2, pp. 162-181.

Blessing, L.T. and Chakrabarti, A. (2009), *DRM: A Design Research Methodology*, Springer, London.

Carpo, M. (2011), *The Alphabet and the Algorithm*, Mit Press, Cambridge.

Casini, M. (2021), *Construction 4.0: Advanced Technology, Tools and Materials for the Digital Transformation of the Construction Industry*, Woodhead Publishing, Duxford.

Celani, G. and Vaz, C.E.V. (2012), "CAD scripting and visual programming languages for implementing computational design concepts: a comparison from a pedagogical point of view", *International Journal of Architectural Computing*, Vol. 10 No. 1, pp. 121-137.

Corey, S.M. (1954), "Action research in education", *The Journal of Educational Research*, Vol. 47 No. 5, pp. 375-380.

Crawford, C. (2004), "Non-linear instructional design model: eternal, synergistic design and development", *British Journal of Educational Technology*, Vol. 35 No. 4, pp. 413-420.

Cudzik, J. and Radziszewski, K. (2019), "Parametric design in architectural education", *World Transactions on Engineering and Technology Education*, Vol. 17, pp. 448-453.

Futschek, G. (2006), "Algorithmic thinking: the key for understanding computer science", *Informatics Education–The Bridge between Using and Understanding Computers: International Conference in Informatics in Secondary Schools–Evolution and Perspectives, ISSEP 2006*, Vilnius, Lithuania, November 7-11, 2006, Springer Berlin Heidelberg, pp. 159-168.

Hanna, S. and Turner, A. (2006), "Teaching parametric design in code and construction", *Sigradi*, pp. 158-161, 2006.

Hernandez, C.R.B. (2006), "Thinking parametric design: introducing parametric Gaudi", *Design Studies*, Vol. 27 No. 3, pp. 309-324.

Howe, N. (2011), "Algorithmic modeling: teaching architecture in digital age", *Proceedings of ACADIA Regional 2011: Parametricism*.

Downloaded from http://ftp.nowpublishers.com/ohi/article-pdf/49/4/736/10204942/ohi-06-2023-0135en.pdf by guest on 15 May 2026

Kátai, Z. (2015), "The challenge of promoting algorithmic thinking of both sciences-and humanities-oriented learners", *Journal of Computer Assisted Learning*, Vol. 31 No. 4, pp. 287-299.

Knight, T. (2012), "Slow computing: teaching generative design with shape grammars", in Gu, N. and Wang, X. (Eds), *Computational Design Methods and Technologies: Applications in CAD, CAM and CAE Education*, IGI Global, pp. 34-55.

Maldonado, M.P. (2014), "Digital recipes: a diagrammatic approach to digital design methodologies in undergraduate architecture studios", *Fusion-Proceedings of the 32nd eCAADe Conference*, pp. 333-342.

Newell, A. and Simon, H.A. (1972), *Human Problem Solving*, Prentice-Hall, Englewood Cliffs, NJ.

Nind, M., Curtin, A. and Hall, K. (2016), *Research Methods for Pedagogy*, Bloomsbury Publishing, London, New York.

Oxman, R. (2008), "Digital architecture as a challenge for design pedagogy: theory, knowledge, models and medium", *Design Studies*, Vol. 29 No. 2, pp. 99-120.

Oxman, R. (2017), "Thinking difference: theories and models of parametric design thinking", *Design Studies*, Vol. 52, pp. 4-39.

Riekstins, A. (2018), "Teaching parametricism as a standard skill for architecture", *Journal of Architecture and Urbanism*, Vol. 42 No. 1, pp. 34-39.

Romcy, N.M.E.S., Tinoco, M.B.M. and Cardoso, D.R. (2015), "Reflections on the introduction of parametric thinking into design education", *V!RUS*, São Carlos, n. 11, 2015. [online], available at: http://www.nomads.usp.br/virus/virus11/?sec=4&item=2&lang=en

Rüütmann, T. and Kipper, H. (2011), "Teaching strategies for direct and indirect instruction in teaching engineering", *2011 14th International Conference on Interactive Collaborative Learning*, IEEE, pp. 107-114.

Schnabel, M.A. (2013), "Learning parametric designing", in Khosrow-Pour, M. (Ed.), *Industrial Engineering: Concepts, Methodologies, Tools, and Applications*, IGI Global, Hershey, PA, pp. 197-210.

Tedeschi, A. and Lombardi, D. (2018), "The algorithms-aided design (AAD)", in Hemmerling, M. and Cocchiarella, L. (Eds), *Informed Architecture: Computational Strategies in Architectural Design*, Springer, pp. 33-38.

Touloupaki, E. and Theodosiou, T. (2017), "Performance simulation integrated in parametric 3D modeling as a method for early stage design optimization—a review", *Energies*, Vol. 10 No. 5, p. 637, doi: 10.3390/en10050637.

Wanan, S. (2016), "Teaching parametric design in architecture: a case study", *Parametricism Vs. Materialism: Evolution of Digital Technologies for Development [8th ASCAAD Conference Proceedings*, London, 7-8 November 2016, pp. 357-366, ISBN 978-0-9955691-0-2].

Willis, J. (1995), "A recursive, reflective instructional design model based on constructivist-interpretivist theory", *Educational Technology*, Vol. 35 No. 6, pp. 5-23.

Williams, D.D., South, J.B., Yanchar, S.C., Wilson, B.G. and Allen, S. (2011), "How do instructional designers evaluate? A qualitative study of evaluation in practice", *Educational Technology Research and Development*, Vol. 59, pp. 885-907.

Woodbury, R. (2010), *Elements of Parametric Design*, Routledge.

Corresponding author
Elena Vazquez can be contacted at: elena.vazquez@umu.se

For instructions on how to order reprints of this article, please visit our website:
www.emeraldgrouppublishing.com/licensing/reprints.htm
Or contact us for further details: **permissions@emeraldinsight.com**

Downloaded from http://ftp.nowpublishers.com/ohi/article-pdf/49/4/736/10204942/ohi-06-2023-0135en.pdf by guest on 15 May 2026