TOPIC 2: TYPES OF PARAMETERS

Jabi, W. (2013). Parametric design for architecture, Laurence King Publishing.



accordingly. We call this feature of updating the value of one object based on changes in other values *propagation*. Imagine a large network of wired or associated values. A change in one or few parameters would propagate through the whole network, modifying the values of attributes and changing the characteristics of the final design solution. This is the power of an associative parametric system. Objects, attributes and values are associated with one another and parameterized so that a change in the value of one parameter can have ripple effects throughout the design.

### Families and inheritance

Objects that share certain characteristics can be organized as members of a *class* or *family* of objects. A class or family of *doors* **[fig. 5]**, for example, can contain many individual family members (hinged doors, sliding doors, folding doors, etc.). The advantage of grouping several objects into a family is that they can then share certain attributes with their siblings and inherit certain attributes from their parents. It is much more efficient to organize these shared attributes only once, in a parent object, than to have to customize all the attributes and values for each offspring.

### Methods

In an object-oriented system, methods are functions and algorithms that act on an object by modifying its attributes. Rather than have a large set of centralized instructions that specify how to draw circles, squares and triangles, an object-oriented system delegates, encapsulating these instructions in the class or family of each object. How an object is to be constructed or modified is thus encoded as a method in the object itself. In the case of a circle, one such method could be to construct the circle by specifying the position of its centre and the value of its radius attribute. Another method could be to specify three points that circumscribe it. The system can simply tell a circle to draw itself – or it can ask a door to reverse its opening. In a modern parametric system a typical object, even one as simple as a sphere, can have many parameters and methods **[fig. 6]**.

START HERE ----

### Parameters

At the heart of any modern parametric system is the term *parameter* and so it would be wise to define that term at this point. The word *parameter* derives from the Greek for *para* (besides, before or instead of) + *metron* (measure). If we look at the Greek origin of the word, it becomes clear that the word means a term that stands in for or determines another measure. The word *parameter* is often confused with *variable*, but it is more specific. In mathematics *parameter* is defined as a variable term in a function that 'determines the specific form of the function but not its general nature, as $a$ in $f(x) = ax$, where $a$ determines only the slope of the line described by $f(x)$'.[3]

In parametric CAD software, the term *parameter* usually signifies a variable term in equations that determine other values. A parameter, as opposed to a constant, is characterized by having a range of possible values. One of the most seductive powers of a parametric system is the ability to explore many design variations by modifying the value of a few controlling parameters **[fig. 7]**.

The remainder of this book presents a series of parametric design patterns of increasing complexity followed by exemplar case studies that reflect the potential of the associated patterns. The book ends with a discussion of the future of parametric design and its potential to form a language of design. The afterword by Brian Johnson closes the discussion with advice on how to craft new solutions, based on knowledge gleaned from this book.

3 From http://dictionary.com


**fig. 6** The parameters and creation method options of a sphere in Autodesk's *3ds Max.*

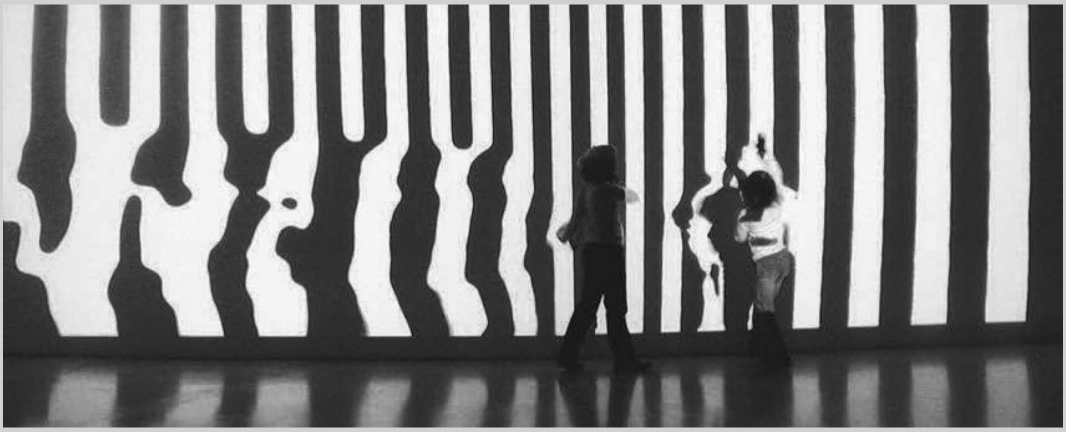

**fig. 7** In the *Matière à Rétro-projeter!* (Material Projects) exhibition at the Centre Pompidou, Paris, France, young visitors affect a regular field of light and shadow patterns as they move in front of it.

# A taxonomy of parameters

Parametric design is 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. It is only natural that computer-based parametric systems focus mainly on geometry and topology. After all, parametric systems are usually attached to, or built on top of, more traditional 3D solids modelling software. It is a good starting point for anyone interested in implementing a parametric approach to form-finding in his or her design workflow. Yet, at times, this can reduce the whole design process to a series of fantastic, self-congratulatory mathematical acts of acrobatics. As any other system, a parametric design system is defined by its input, algorithm, and output. We have matured in the area of geometric algorithms and can invent as well as physically build very complex geometry. The real challenge in parametric design is not how clever the algorithm is, or how complicated the output is, but in the selection of the initial input parameters. What parameters exist beyond the geometric one? Very few architects and software developers have taken on the challenge to classify, let alone invent, systems that can accept fundamentally different types of parameters. In order to truly connect parametric design to the everyday activities of designers, they need to understand and represent the same issues the designers are working with: geometry and topology, but also architectural components, materials, the environment and people. Below is an attempt to classify and explain these parameters, in the hope that it will serve as the foundation of future research projects with the goal of inventing more versatile tools to address this glaring deficiency in the current generation of parametric systems.

**Mathematical parameters** are the most basic type of parameter that are already understood by 3D modelling software: numbers, logical values and even strings of characters (which are represented internally using numbers). Many parametric systems, such as spreadsheets (which are undeniably powerful parametric systems in their own right), only need this level of parametric input in order to calculate very useful outputs.

**Geometric parameters** are higher-level entities that are built out of the lower-level mathematical parameters. Examples include points, lines, surfaces and solids. Most current 3D modelling software can represent and parametrically modify geometric constructs of various types.

**Topological parameters** describe how two or more entities relate to each other: connected to, above, below, is near to, looking at, is within, is outside of, etc. Most modern parametric systems excel at precisely these types of parameters. For example, a diagrid pattern is a topology that divides a surface in a consistent manner regardless of the exact geometry of the parent surface or the resulting pattern. This allows us to disassociate topology from geometry while maintaining the consistency of our design intents. Most of the examples in this book fall under this category. Topological parameters allow us to consider issues of form, composition and fabrication, and they open the possibility of further analysis as they more precisely define our design intent for how the parts relate to each other and to the whole.

**Representational parameters** describe and abstract entities outside themselves. Examples include computer representations of walls, windows or columns. Building Information Modelling (BIM) was invented in large part to address the need to represent 'real' objects. In BIM, a distinction is made between an isolated geometric construct such as a cuboid, and a brick wall, which knows how many bricks it has, its own weight, structural strength, cost, etc. Representational parameters allow us to describe some if not all of the physical properties of what we are modelling. They also allow us to aggregate that information so we can report overall values and quantities.

**Material parameters** build on mathematical, geometric, topological and representational parameters by adding and connecting several physical attributes: weight, tension, friction, elasticity, structural strength, U-value, reflection, refraction, etc. This class of parameter begins to remove us from the realm of self-referential geometric games and into the physical world of materiality. Good examples of parametric systems that accept and consider topological parameters are tensile membrane form-finders, biomimetic explorations, and particle and physics engines that can encode, almost at a cellular level, the physical properties, collision, velocity, gravity and structural stresses that a system is undergoing. Future systems for parametric design in architecture should encode materiality and physical parameters, as this will allow us to model, predict and thus parametrically explore the performative aspects of our design proposals before they are actually physically built. Analysis software that precisely models structural or thermal properties should more fully integrate the essential material and physical properties in our geometric and representational constructs, such that they fluidly react to, propagate and give us feedback on constraints and interactions within the overall parametric system in real or near-real time. For example, very few current parametric systems can represent the time-based effect of prolonged exposure to fire on a structural system or a particular building material. Physics-based computer games and bioengineering research, however, have reached that stage and we would be wise to learn from their techniques in the field of architecture.

**Environmental parameters** include the frequently invisible and fluid forces that surround us. Time, wind, thermal variation, vistas and views, the movement of light and shadow, magnetic fields, Wi-Fi and GPS signals, growth and erosion are all examples of environmental parameters. Not many of us can easily imagine the path of a shadow as it travels during the day or the undulations of a field of sunflowers as they follow the path of the sun; this is why we find time-lapse photography so fascinating. Interactive façades that respond to environmental conditions (usually the path of the sun) are a good start. However, we need a deeper understanding of the totality and complexity of environmental factors so that we can optimize our design solutions, given complex and competing constraints.

**Human parameters** form the seventh and most challenging class of parameters. Architecture's purpose is, after all, to shelter humans from the elements. While we share many physical attributes and needs, we also differ in profound ways both ergonomically and psychologically. If we are to create humane architecture and one that creates truly customizable spaces, we need to be able to model our clients, their intents and desires, and incorporate that information as parameters in our design systems. It is truly shocking that, in many cases, incorporating the human parameter in our design projects and renderings does not go beyond the inclusion of a scale model of a person. That is only a start; masterful architects know how to address and resolve multivalent parameters (fig. 112). We truly need to learn from the field of ergonomics and especially the advanced systems that office furniture, automobile and medical equipment manufacturers use to model human beings. A good development in our field is the increasing incorporation of simulated crowd systems. Sadly, while effective at simulated fire egress, the simplifying assumptions of such systems, which reduce the complexity of human behaviour to that of a robot, render them useless to predict common human behaviour such as where clusters of people might gather or pause on a sunny afternoon. As the computational capability of our systems grows, so will the sophistication of these simulations.

fig. 112 Le Corbusier's *Le Modulor* cast in concrete at the Unité d'Habitation (1965), Firminy, France.

Incorporating all seven classes in a parametric design system is not only a tall order, but not always advisable given the design situation. Knowing how to abstract a situation and build a conceptual model in which extraneous parameters are excluded but essential ones are included is part of our irreplaceable skill as designers. Parametric modelling lets you capture that conceptual model and make it explicit. This shifts the conversation. The challenge is not one we can shy away from if we aim to be precise about our design intent and, perhaps more importantly, strive to understand the consequences of our decisions before we actually build them. Parametric systems are only one step in that endeavour.