Parametric Models in Hyper-Space

One of the key aspects in computational design is the capacity of computers and computational procedures to create multiple solutions to a design problem in a short time. This becomes evident if designers use parametric models during the early stages of the design process, exploratory design, where many ideas and solution candidates are generated and evaluated rapidly.

CARLOS R. BARRIOS
Clemson University

The use of parametric models in the initial phases of design brings to the surface two important questions: how to find and effectively display the possible design solutions generated by a parametric model; and how to navigate thorugh the solution space with efficiency. This paper presents a research in progress on a possible solution to this problem. This paper will address the issues of effective display and efficient navigation of the possible solutions produced by a parametric model prior to the selection process of the final candidate for a design problem.

PARAMETRIC MODELING AND DESIGN

A parametric model can be defined as a group of geometrical components with attributes (properties) that are variable and others attributes that are fixed. The variable attributes are called *parameters* and the fixed attributes are called *explicit*. Variable attributes will allow changes when a designer wishes to perform these changes. Changes can occur within the limits set by the parametric model. Theoretically speaking, parameters can be added, changed or deleted by the designer at will at any point of the modeling process.

Parametric Design is defined as the process where designers use Parametric Models.[1] Designers use them to explore multiple solutions by performing variations to the variable components of a geometrical. Parametric model offer multiple solutions to a single problem with great deal of flexibility. Navigation through the solution space, the collection of all possible solutions provided by a parametric model, is an issue that deserves more attention. In particular when a designer finds difficult to predict the possible outcomes of a parametric model.

Although the design process is inherently parametric activity, *Parametric Design* occurs when a designer uses computer geometrical models that are parametric.

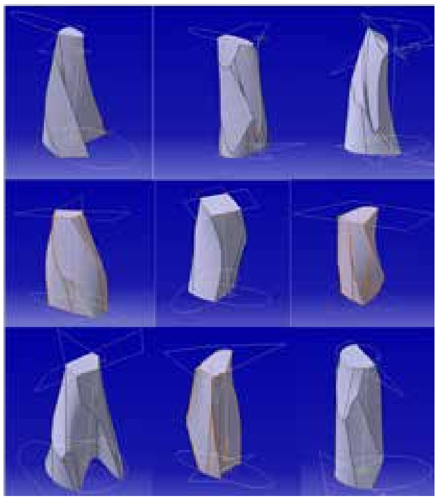
1 

Figure 1: Typical arrangement of instances of parametric models.

TYPICAL ARRANGEMENTS OF PARAMETRIC MODELING INSTANCES

It is quite evident that a parametric model could generate a large array of possible designs or *design instances.* To organize them, designers resort to arrays that will display the individual instances next to each other, sometimes in an seamilgly orderly manner. Design instances can be grouped in either a linear fashion or in table (Figure 1). A table is a 2D structure with cells used as placeholders for each parametric design instance. This is a typical array when models are simple and easy to follow, but it can create confusion when models increase in complexity. Optimum navigation through the solution space is one of the most overlooked problems in parametric computational design. Proper navigation cannot exist without proper visual display of the design solutions. This is particularly true when a solution set gets larger as the design model becomes complex, and typical display arrangements can become inefficient.

THE HYPER-MATRIX AS A FORMALISTIC MODEL

The formalistic model is adapted from the periodic arrangement in hyper-structures method developed by H. Lalvani in the 1980s.[2] The formalistic modelpresented here allows the logical arrangement and display of all design solutions from a parametric model in a tesseract, hyperspace cube, lattice, hence the name hyper-metric design. The hypermetric model uses a hyper-lattice growth in every direction where a parameter is needed. It also allows the incorporation of new parameters on demand at any point of the development of the design.

The Hyper-Matrix, constructed as a multidimensional lattice structure based on a hyper-cube of four dimensions, is conceptually built on the following premises:

1) Parametric variations are based on a single parametric model only, thus each Hyper-Matrix contains the instances of one and only one parametric model.

2) In the Hyper-Matrix, each axis or direction indicates the variations of a single parameter on the model. The number of parameters will determine the number of axes. If a model has two parameters, then it will require two axes. If the model has three parameters the matrix will have three axes, so forth for any number of parameters.

3) Each node in the Hyper-Matrix is a placeholder for a single parametric instance. Each node shows the actual values of the parameters that are varied at that particular point, or the result of the interactions when multiple parameters are simultaneously varied.

4) Any instance travelling between nodes will have infinite variations of the values of the parameters that are being varied along the line.

5) When new parameters are introduced, a new axis will be introduced. These new axes can be retraced back to any previous node in the Hyper-Matrix at the designer's will. This axiom is considered an advanced feature of the formalistic model and will be discussed in depth in a following paper.

CASE STUDY

For the purpose of illustrating how the hyper-matrix is used for arrangement of the parametric instances we will use a parametric model that generates the forms of the columns of the Sagrada Familia. The multidimensional hyper-matrix lattice will use a single parametric model to create all of the shapes that generate the aforementioned columns and produce a logical arrangement in a periodic tessellated form.

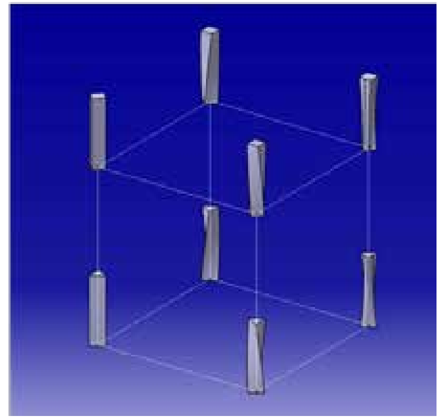
To start, we will build the first level of the hyper-matrix by using three parameters: Initial shape, rotation, and counter-rotation. The use of three parameters will require using three axes (directions) where each axis represents a parameter as follows: A) the first axis will be used for the clockwise rotation parameter; B) the second axis will be used to indicate the counter clockwise rotation; and C) the third axis (vertical axis) will be used to change the topology of the initial shape. As a starting point we chose a triangle as the initial shape for the first node. Figure 2 shows an arrangement of the possible designs from using a triangle as the initial shape on the initial node and the corresponding parametric instances when parameters are changed.

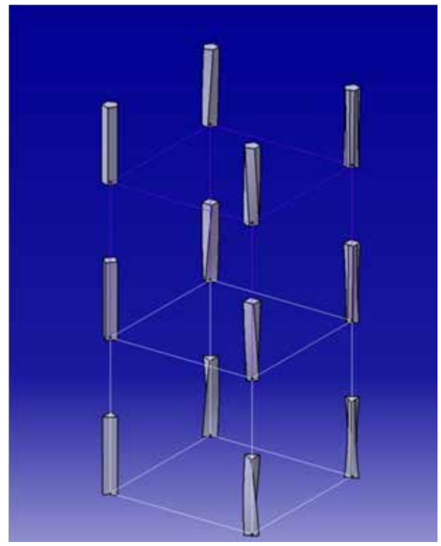
The lower level of the lattice shows the parametric instances when parametric variations are performed on the triangular initial shape. One axis shows the rotation parameter with the corresponding rotated instance at the node, while the second axis shows the counter-rotation parameter and corresponding parametric instance. When the two axes converge, both parametric variations are performed simultaneously thus creating an emergent design in the fourth node, as it is the case for the node on the far right. It is assumed that the simultaneous application of both parametric variations will yield a third emergent parametric instance which is dependent on the values of the parameters of the previous parametric variations. This creates a dependency relationship between nodes that is linear and unidirectional, which is consistent with the accepted notions of parametric variations.

The vertical axis is used to change the topology of the initial shape by increasing the number of sides by 1. Therefore on the second level, the initial shape is a square. The corresponding directions for rotation and counter-rotation are parallel to those on the lower level so the relations between lower level instances and the upper level instances remains logical and consistent. This allows a logical relation between parametric instances and their corresponding variations.

To include more parametric instances in the arrangement there are two ways to increase the levels on the hyper-matrix: 1) continuous tessellations on the existing axes to extend the parametric variations (parametric extension); and 2) increase the number of parameters by adding additional rules or axes as deem necessary (parametric expansion). Figure 3 shows a parametric extension to a third level by adding an additional initial shape as a tessellated operation on the vertical axis. The new level follows the rule of the vertical axis that calls for an increase of the number of sides of the initial shape by 1, therefore the next shape in the extension will be a pentagon. The additional axes remain congruent with the rules of the basic Hyper-Matrix thus creating a logical growth of the parametric instances and showing how the parametric instances of the rotation, counter-rotation and intersection are immediately generated. Further growth in the vertical axis is possible to allow new levels to emerge as the parametric rule for extension is applied to the vertical axis. For the case of the Sagrada Familia column, the next group of columns are 6, 8 10 and 12 sided.

A different solution will be to increase the Matrix to a higher dimension with a new parameter defintion. In Figure 4 we see a projection of a *tesseract* or hypercube (cube of 4 dimensions) where the new axis indicates a transformation that duplicates the number of sides for the original shape. On the first level we see a 3 sided column growing to a 6 sided column. Likewise, on the second level we see a 4 sided column being duplicated to an 8 sided column.


2


3

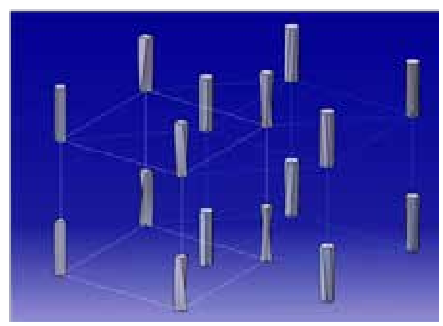
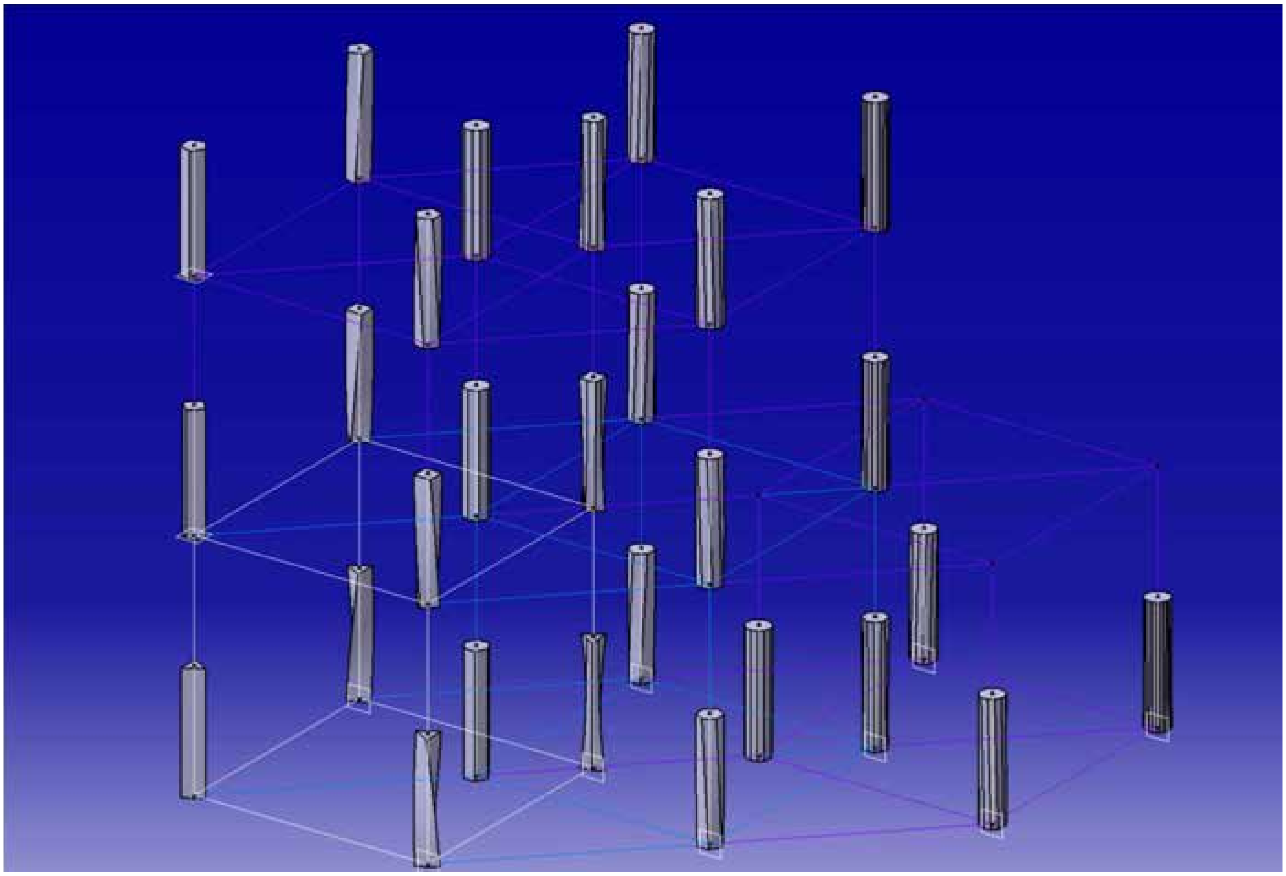

4

Figure 2: Hyper Matrix levels 1 and 2.

Figure 3: Hyper-Matrix expansion to level 3.

Figure 4: Instances organized in a tesseract.



DISCUSSION

The arrangement of the parametric instances in the multidimensional hyper-matrix lattice is periodical, as opposed to the traditional sequential form or 2D arrays. There are several advantages to this methodology. Periodical means that for every direction you chose to follow at every node there is a logical sequence of parametric variations that is consistent with the rest of parallel axes, regardless of the visited node.

In some cases, local parametric variations will be allowed, but this will require nesting of Hyper-Matrices inside the Hyper-Matrix and the possibility that one node can contain more than one parametric instance. This will be the subject of future studies and development.

There is virtually no end to the lattice growth in the hyper-matrix, but the designer can restrict it as deem necessary.

An argument can be made about the importance of *making a proper selection* when a computational process generates a large number of candidates. However, this is a problem that has been addressed in many ways and at different levels through fitting functions, optimization algorithms, genetic algorithms, performance-based evaluation, and many others.

ENDNOTES

1. C. Barrios, Thinking Parametric Deisng: Introducing Parametric Gaudi in Design Studies Vol 27 2006 pp 309-324.

2. H. Lalvani, *Structures in Hyper-Structures. Self published by the author.*