11_environmental analysis

"Don't fight forces, use them".

R. Buckminster Fuller

Several software applications are available in the field of environmental design; most however, are back-end analysis and give small contributions in exploring holistic design solutions. Due to difficulties in making changes during advanced stages of modelling/drawing designers often rely on personal judgment deduced from experience, over analysis. Thereby, narrowing the possibilities to explore a large set of environmental-informed designs.

The combined use of environmental and parametric packages can guide the generation of form from the initial sketches to the final design. This chapter introduces plug-in software used to bridge between parametric modelling and environmental analysis. The software find forms through real-time analysis based on insolation, lighting etc.. These techniques, supported by a deep understanding of data can lead to environmental-conscious designs.

11.1 Tools

The Grasshopper ecosystem provides users with several plug-ins for environmental analysis; some plug-ins perform complete environmental analysis, while others bridge between specific environmental analysis software. Among the latter is the plug-in **GECO**, which directly links between Grasshopper and one of the most powerful analysis software: **Autodesk® Ecotect® Analysis**, developed in 1996 by Andrew Marsh and subsequently acquired by Autodesk in 2008. This chapter will demonstrate the use of the plug-in GECO to solve environmental analysis problems.

11.2 GECO and Ecotect

GECO, developed by [uto], an Innsbruck-based design team and research collective founded by Ursula Frick and Thomas Grabner[37], contains a set of components for Grasshopper that enables data to directly flow between Grasshopper and Ecotect, making environmental optimization possible.

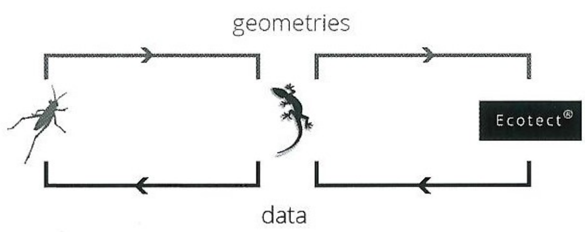


FIGURE 11.1
The GECO – Grasshopper – Ecotect Workflow.

The Ecotect interface is both graphically intuitive and analytically comprehensive. The software has the ability to organize data related to thermal performance, solar radiation, daylighting, shadows, and acoustical analysis. Ecotect also features a complete 3D modelling environment, however is not conducive to creating complex forms. GECO fills the void between shape modelling and analysis, allowing users

NOTE 37
Ursula Frick and Thomas Grabner are the founding directors of [uto]. Their research focuses, along with the generation and analysis of complex geometries, on the development of formal strategies, data driven design approaches and generative structures, implementing and evaluating the physical matter of a design. They have led studios and run workshops worldwide. They are also recognised for their development of digital tools of digital tools, namely PhysX, Geco, Flowlines, MeshPaint and MeshEdit, which seek to improve workflow in a parametric environment. Both graduated with distinction from the University of Innsbruck, with degrees in Architecture.

to parametrically model geometries in Grasshopper and export them to Ecotect generating real-time feedback loops used to inform and optimize the geometries of facades, covers, buildings, etc.
In summary, GECO's components facilitate:

- The exporting of points or geometries from Grasshopper to Ecotect;
- Setting analysis grids;
- Performing daylighting calculations (*EcoLightCal*);
- Calculating solar radiation (*EcoSolCal*);
- Importing feedbacks from Ecotect to Grasshopper;
- Importing Grasshopper solar diagrams calculated by Ecotect.

GECO component *EcoLua* enables users to interact directly with the Ecotect scripting language *Lua*.

11.2.1 Interface

Three software's are required to be installed to use GECO: Ecotect, GECO itself and Mesh Edit (see 6.3.3). Ecotect is a commercial software available as a 30-day trial or 3-years student license available at www.autodesk.com. GECO and Mesh Edit are both available as a free download from www.food4rhino.com. Once installed, a new panel "GECO" will appear in the Extra tab.

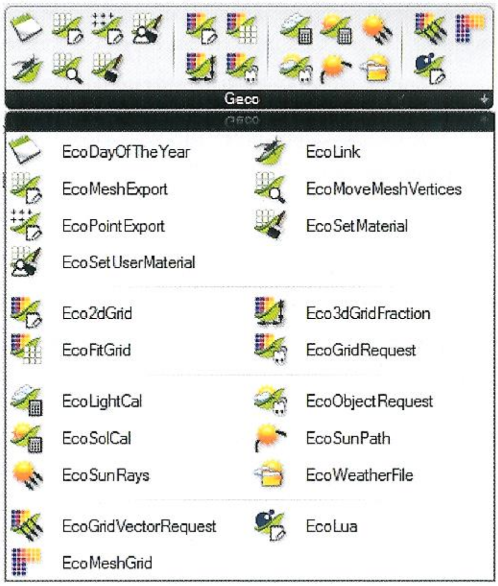


FIGURE 11.2
The GECO panel, hosted within the Extra tab.

GECO's components can be grouped into four categories:

- Link and export;
- Analysis grids;
- Calculations;
- Feedbacks.

Once a GECO component is place on the canvas, the component's name changes from what is displayed in the GECO panel; for example once added on the canvas: *EcoLink* becomes *Link Ecotect*, *EcoSolCal* becomes *Insolation Calculation*.

11.2.2 Link and export

Two components are used to link and export: first, the component *EcoLink* (*Link Ecotect*) is a switch that establishes a connection between Ecotect and Grasshopper, and second, the component *EcoMeshExport*, which exports objects modelled in Rhino or generated in Grasshopper into Ecotect, in order to perform analysis. Similar to other analysis software **Ecotect works exclusively on mesh geometries**.

11.2.3 Analysis grid and mesh-analysis

Ecotect performs analysis either on grids or directly on mesh-objects. **Grids** are a set of points within a space that can be imagined as *sensors* where Ecotect generates results; for example, lighting analysis is performed at several points within a space not only on objects. In other cases, such as insolation, analysis can be directly performed on the faces of the mesh-geometry. If analysis grids are necessary, their position in space and the number of cells is required to be defined. The larger the number of cells the more accurate the final result. The main components to set grids are *2dAnalysisGrid* and *FitGrid*. The component *2dAnalysisGrid* draws a grid according to a spatial domain and to a number of cells, while the component *FitGrid* sets a grid according to the dimensions of the current object in Ecotect.

11.2.4 Calculations

GECO's components calculate:

- **Solar diagrams** by using the two components: *EcoSunPath* and *EcoSunRays* (see 11.4);
- **Insolation** by acting on a mesh or a grid using the component *InsolationCalculations* which

returns the following values (see 11.6):

1. Incident solar radiation;
2. Absorbed and transmitted solar radiation;
3. Sky factor and photosynthetically active radiation (PAR);
4. Shading, overshadowing, and sunlight hours.

- **Lighting comfort,** by using the component *LightingCalculations* the following can be returned (see 11.8):
  1. Daylight factor;
  2. Daylight levels;
  3. Sky component.

WEATHER FILES

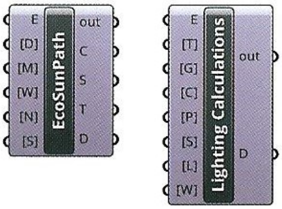
To execute environmental calculations, valid climate data is required to be set for the **geographic location** that hosts the object to analyze. Geographic data is embedded in specific files called *Weather Files*. A **Weather File** contains the climatic data for a location such as: solar radiation, prevailing winds, temperatures, rainfall, humidity etc. Ecotect has Weather Files embedded in the software for major world cities; other locations can be download directly from the U.S. Department of Energy (http://apps1.eere.energy.gov/buildings/energyplus/weatherdata_about.cfm). The USDOE website provides files in the .EPW format file (Energy Plus Weather File) which cannot be read by Ecotect which requires .WEA files (Weather Data File). The .EPW file can be converted into a .WEA file using the *Weather Manager* in Ecotect (Tools > Run the WeatherTool) by opening the .EPW file and saving it as a .WEA file.

11.2.5 Feedback / import

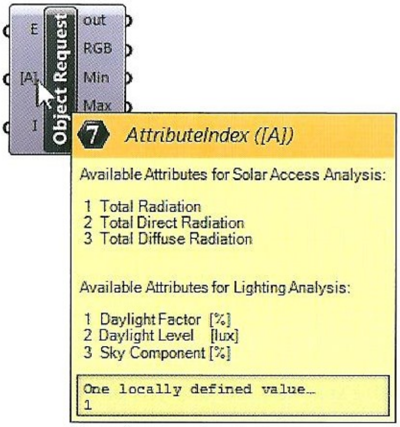
Once the analysis is performed in Ecotect, the results can be imported into Grasshopper to control and inform geometric transformations. For example, data returned from solar radiation analysis can be used to control the dimension of apertures in a cover. We can distinguish between two import modes depending whether the analysis is performed on a grid or onto a mesh. In the first case (analysis on grid) we will use the *EcoGridRequest* which allows to transfer into Grasshopper all the data calculated by Ecotect. In the second case (analysis on mesh), we rely on the *EcoObjectRequest* component able to import calculations made directly on mesh like radiation or lighting.

11.3 About GECO's components

Most GECO components have a similar structure facilitating **cascading connections**. To better understand this concept, two components, *EcoSunPath* (used to draw solar diagrams) and *LightingCalculations* (used to calculate lighting comfort) will be examined.

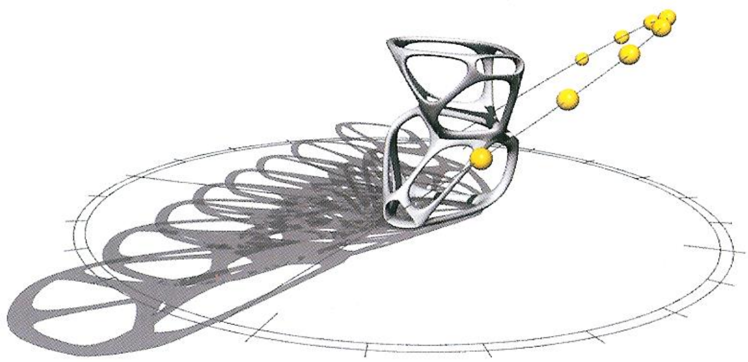


Both components have an input (E) that is satisfied by a Boolean statement which can be provided by a *Boolean Toggle*. A *Boolean Toggle* set to True will execute the specific calculation instructed by the component in Ecotect. All GECO components have: an output (out) which displays information received from Ecotect, and an output (D) which returns a Boolean value False if Ecotect is still running or True if the Ecotect calculation is complete. The output (D) and the input (E) can be used to establish a cascade connection between several components such that the Boolean value from the output (D) will activate the second component input (E). Similar to Grasshopper standard components, hovering the mouse over an input or output will provide specific information.

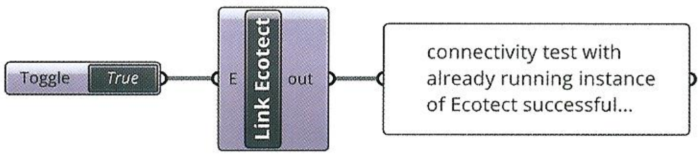


11.4 Solar diagram and shadows

Using the Components *EcoSunPath* and *EcoSunRays* a solar path diagram and the solar ray vector respectively, can be generated for a specific location and time. The solar ray vector can be used to draw the projected shadows of an arbitrary object onto a plane, as illustrated in the following image.



The necessary part of any GECO definintion is to establish a link between Ecotect and Grasshopper using the component *Link Ecotect*. The link can be established or disconnected using a *Boolean Toggle* connected to the input (E). The output (out) generates a text log which displays the result of the connectivity test. *EcoLink* is always an isolated component that is not connected to other components except the *Panel*.



Occasionally the component *EcoLink* (*Link Ecotect*) could be in a warning state, displayed by an orange component colour, meaning the connection with Ecotect has failed. If this happens the connection can be re-established by switching the *Boolean Toggle* from True to False and again from False to True. If *EcoLink* is in an error state, displayed by a red component colour, Ecotect or GECO are not installed properly.

SUN PATH

Once the connection with Ecotect is established the component *EcoSunPath* can extract sun path data for a specific day input [D], month input [M] and location as defined by a Weather File input [W]. The input [D] must be fed by a slider ranging between 1 and 31 and input [M] by a slider ranging between 1 and 12 and input [W] by a Weather File. Weather Files can be loaded using the component *File Path* (Params > Primitive) – renamed as *weather file* in the following images – by right-clicking the component and selecting the option *Set One File Path* from the contextual menu. The .WEA files for specific locations can be found in the default folder: *c:\programs(x86)\Autodesk\Ecotect Analysis\Weather Data.* If the E-input is set to True the analysis starts and a sun path diagram will appear in Rhino. The component *EcoSunPath* returns as its output two curves representing the *sundial* output (C) and the sun path output (S). The output (T) is a number that indicates the *sunrise and sunset*. The scale of sun path and sundial curves can be controlled by the [S]-input.

SOLAR RAY

Once the sun path is defined for a specific day, the component *EcoSunRays* returns a solar ray for a specific time. The inputs [D] and [M] are required to be connected to the same sliders used for the component *EcoSunPath.* The input [T] specifies the time and is connected to a slider ranging from sunrise to sunset (e.g. from 8.00 to 18.00). The V-output of the component *EcoSunRays* is the solar ray vector, which can be displayed using the component *Vector Display* (Display > Vector). The P-output is the vector's start-point.

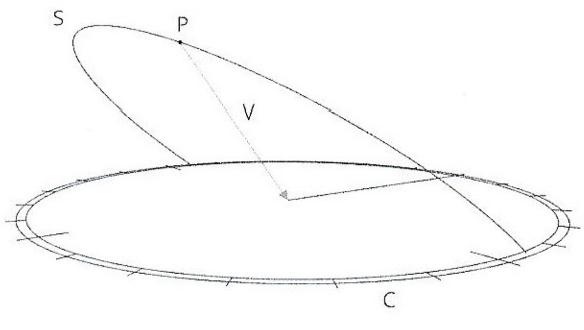


FIGURE 11.3
EcoSunPath and *EcoSunRays* draw in Rhino the sun path curve (S), the sundial (C) and the solar ray vector (V) for a specific place and time.

SHADOWS

The data so far calculated can be used to draw the shadows of an arbitrary mesh-object onto an arbitrary plane. The component *MeshShadow* (Mesh>Util) calculates the cast shadows of a mesh geometry input (M) from a solar ray vector input (L) satisfied by the V-output of *EcoSunRays* component.

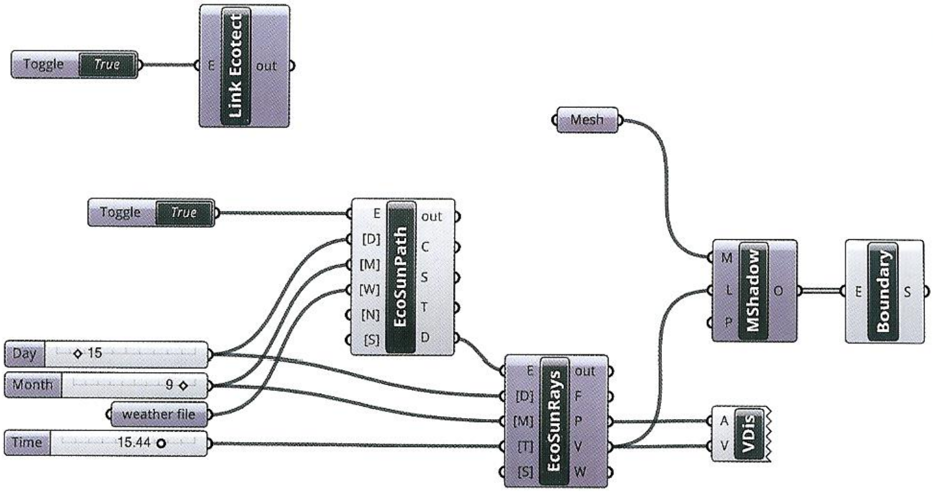
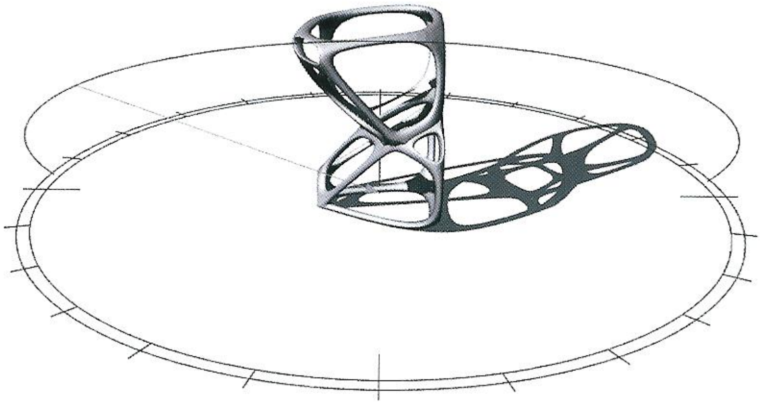


FIGURE 11.4

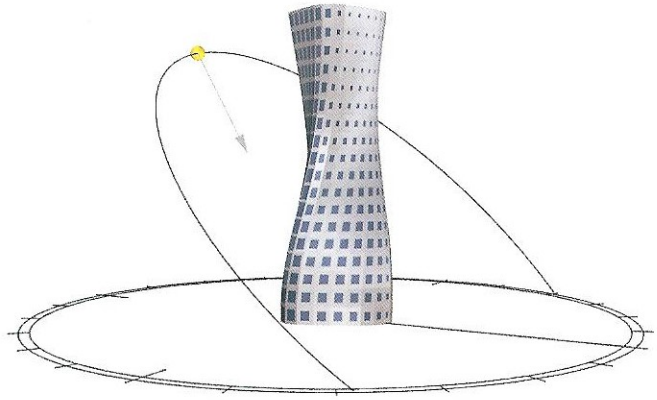
The complete algorithm that calculates sun path, solar ray and shadows for an arbitrary mesh geometry.

The output (O) stores the shadow contour data as projected on an arbitrary plane input (P). To better visualize the shadows a planar surface can be created using the component *Boundary Surfaces* (Surface > Freeform).

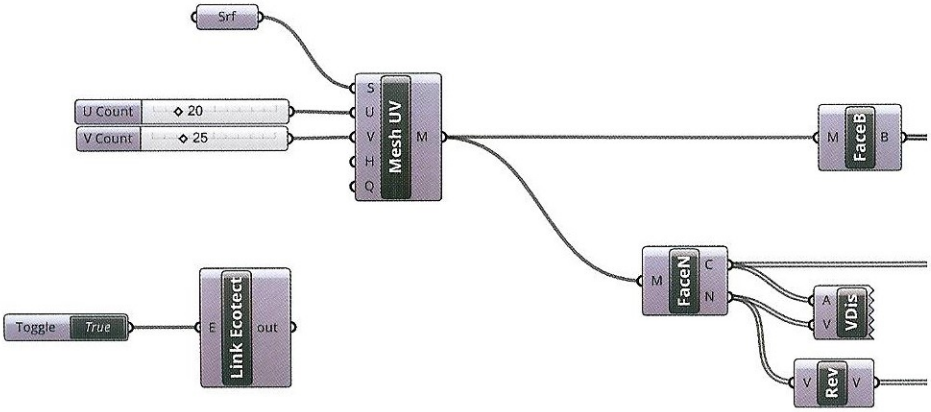


11.4.1 Responsive skin

Environmental data calculated in Ecotect filtered through GECO can be useful in the conceptual stages of design. For instance, the solar ray vector can be used to control a facade's window dimensions, by comparing the angles between the solar ray vector and the surface's normals to inform a geometric transformation.



The responsive skin algorithm starts by defining a single freeform surface and then converting it to a mesh using the component *Mesh Surface*. The defined number of U and V subdivisions becomes the dimensions of the facade panel matrix. As second step the normal vectors at each mesh-face center are calculated, using the component *Face Normals* (Mesh > Analysis). Then, we extract the mesh edges using the component *Face Boundaries* (Mesh > Analysis) outputting a set of closed polyline curves.



After the normal vectors are calculated it is crucial to verify the normal vectors direction using the component *Vector Display*. The vectors must have the same sense as the solar ray vectors, thus the vectors must be directed towards the inside of the surface. In the example, the normal vectors are directed toward the outside of the surface and are required to be inverted using the component *Reverse* (Vector > Vector). After the vectors have the correct sense, the component *EcoLink* (*Link Ecotect*) can be used to establish a connection between Ecotect and Grasshopper.

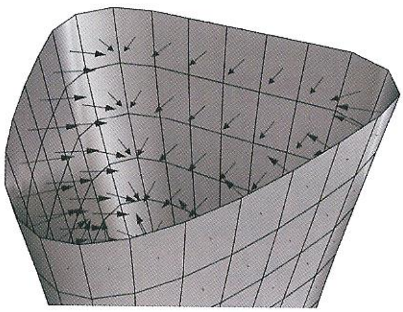
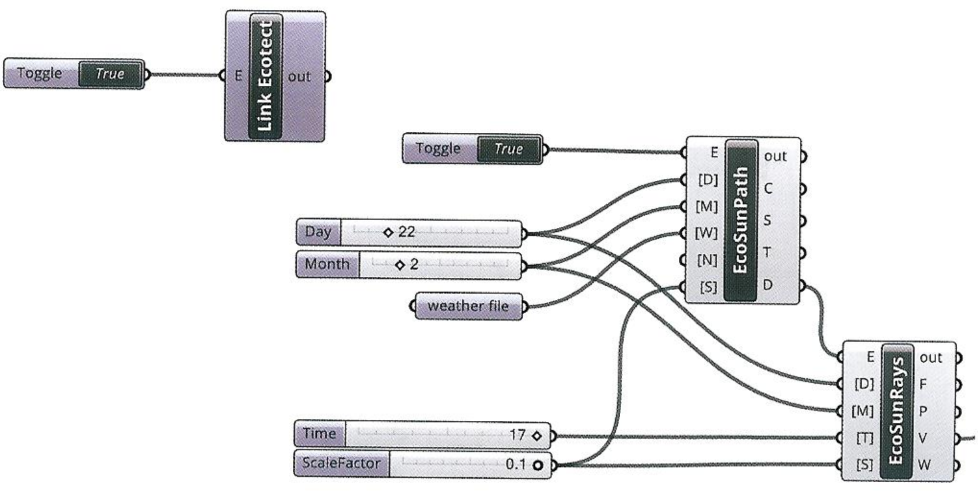
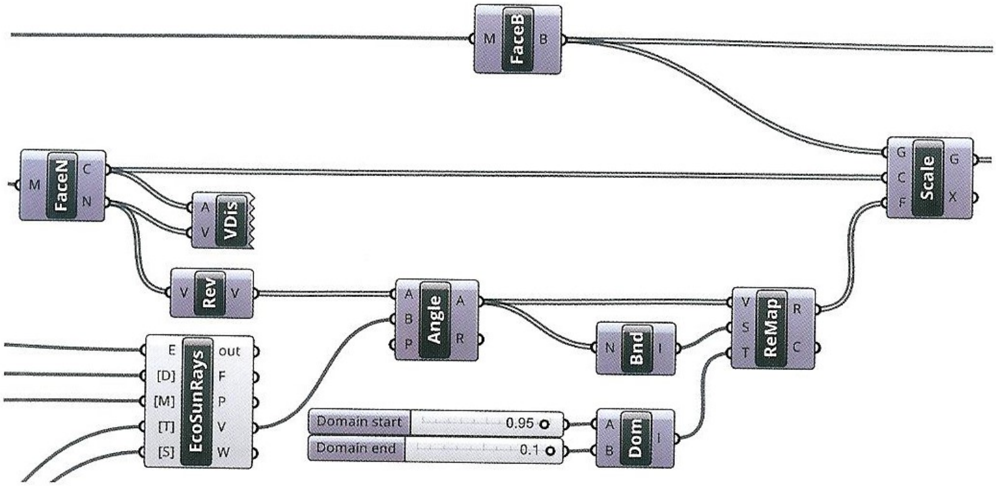


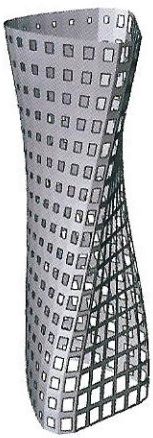
FIGURE 11.5
Faces' normals must be oriented according to the solar rays, meaning they must be directed toward the inside of the geometry to analyze.

The second step is to generate the solar diagram and the solar ray vector using the components *EcoSunPath* and *EcoSunRays* respectively for a specific day and time.

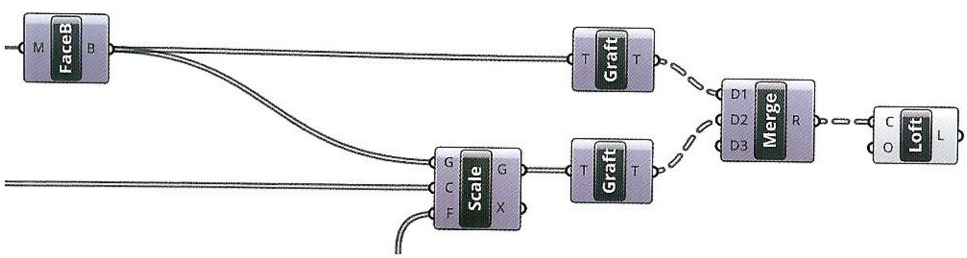


The component *Angle* (Vector > Vector) is used to measure the angle in radians between the solar ray vector and faces' normals. To vary the size of the apertures the mesh edges are scaled using the component *Scale*; such that, the lower the angle value the lower the scale factor. In order to inwardly scale the mesh edges the scale factor must be within the domain (0,1]. As a result, the component *remap* (see 2.5) is used to remap the angle values to a target domain.





Lastly, the variable holed skin is generated by a *Loft* between each original edge item to the respective scaled edge item. As explained in chapter 5, the two data flows are required to be *grafted* before they are merged. After the skin is defined, modifying the [T]-input of *the component EcoSunRays* will yield varying aperture dimensions.



11.5 Exporting geometries and importing data

As discussed in the previous section, Grasshopper can calulate the shadow of a mesh modelled in Rhino using sun path data from Ecotect, through GECO. Some types of analysis, for instance lighting and insolation, require that mesh geometries be exported from Grasshopper to Ecotect to perform analysis then the results are returned into Grasshopper.

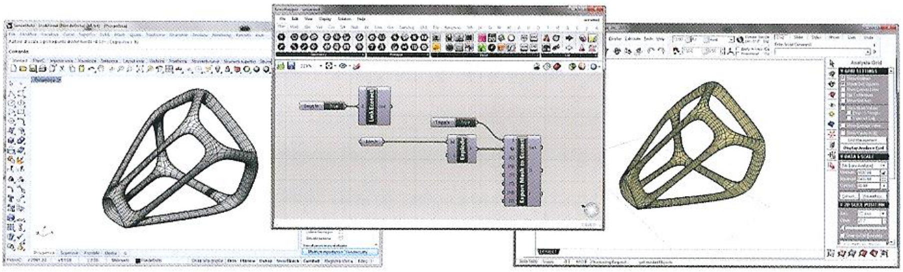
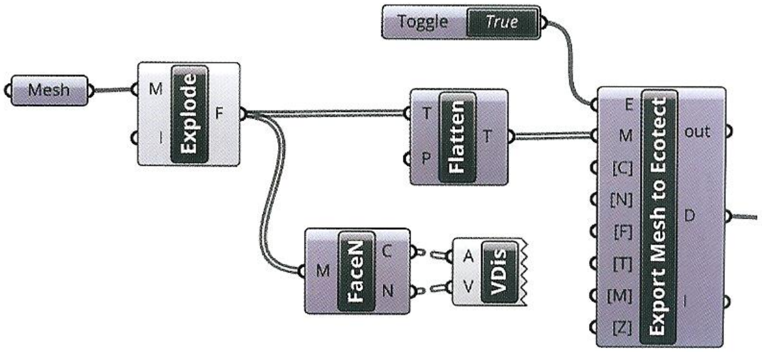


FIGURE 11.6
A mesh geometry modeled in Rhino and exported into Ecotect using GECO as a bridge.

11.5.1 Exporting geometries from Grasshopper to Ecotect

The main export component is *EcoMeshExport* (*Export Mesh to Ecotect*).



When exporting mesh geometries, some important *guidelines* should be followed:

- **Exploding meshes**: to improve compatibility meshes must be exploded using the component *Mesh Explode* (Mesh > Analysis);

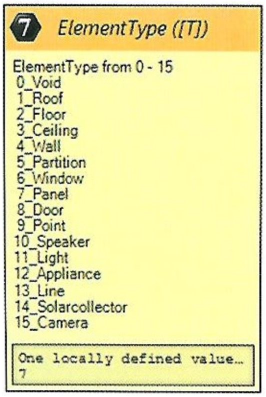
- **Checking faces' normal vectors**: normal vectors must be directed towards the outside. The direction of normals' can be visualized using the components *Face Normals* and *Vector Display*;
- **Flattening data**: the output (F) of component *Mesh Explode* must be in *Flatten* mode before connecting it to the M-input of the component *Export Mesh to Ecotect.* Otherwise, Ecotect will import just the faces hosted within the last branch.

The E-input set to True exports the mesh to Ecotect. It is important to point out that the export procedure is controlled by the N-input which can assume three values:

- if **[N] = 0** GECO exports the current Grasshopper meshes and deletes any geometry present in Ecotect;
- if **[N] = 1** GECO exports the current Grasshopper meshes and deletes just the meshes exported by the latest *Export Mesh to Ecotect* (mesh selected in Ecotect);
- if **[N] = 2** GECO exports the current Grasshopper Meshes and maintains all geometry present in Ecotect (this option may lead to overlapping errors).

The remaining inputs are:

- **[C]**, is the scale factor of the exported mesh. Since Ecotect usually operates in millimeters we can set the scale factor to 0, the default value, if the Rhino environment is also set to millimeters. Otherwise, if the Rhino environment is set to meters we must set [C] = 1000;
- **[F]** is an input related to the analysis grids. Through a *Boolean Toggle* set to True we can fit the analysis grid to the exported mesh;
- **[T]** and **[M]** set an element type and a material to a mesh respectively;



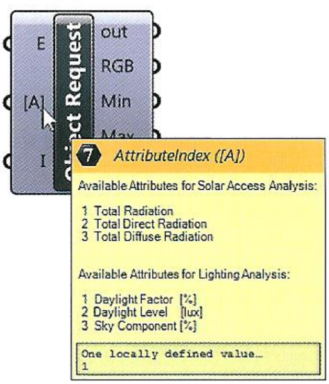
- **[Z]** defines the Ecotect's **zone**. Zones are defined in Ecotect as homogenous enclosed areas or volumes and often but not always coincide with "rooms". Zones are managed similar to CAD-layers, names can be acquired by typing text into a *Panel* connected to the input [Z]. If multiple *Export Mesh to Ecotect* components are used, the same input [Z] is required in order to export different meshes into the same zone.

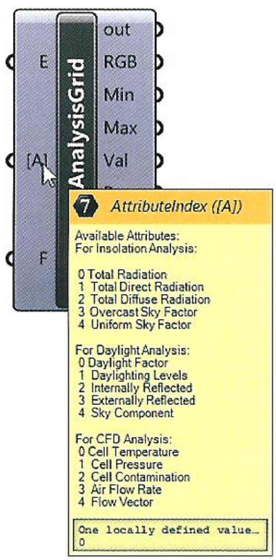
11.5.2 Importing data from Ecotect

There are two methods to import data from Ecotect into Grasshopper; the selected method depends on whether the analysis is performed on grids or on meshes.

The component *EcoObjectRequest* (*Object Request*) is used for meshes. The component returns data calculated at center of each mesh face and it requires as I-input the indices of mesh faces. The I-output of *EcoMeshExport* provides the indices. The component *EcoObjectRequest* has two main outputs: (Val) which provides numeric values resuting from the specific analysis and (RGB) which assigns a color gradient to a mesh according to the numeric values. The component *EcoGridRequest* (*ImportAnalysisGrid*) is used to import grids and data performed on grids.

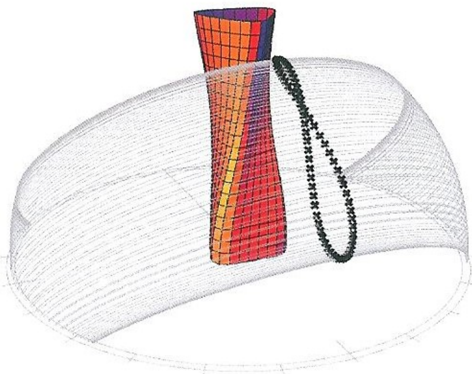
Both components have an input [A] that allows to select specific results (attributes) from the entire set of data returned by an analysis. For example to select the attribute *Total Radiation the* [A]-input must be set to 0. The available attributes can be read on the help menu which appears hovering the mouse over the input.



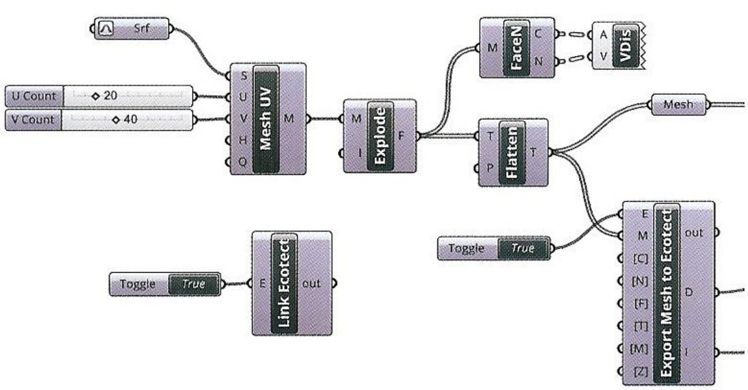


11.6 Insolation analysis

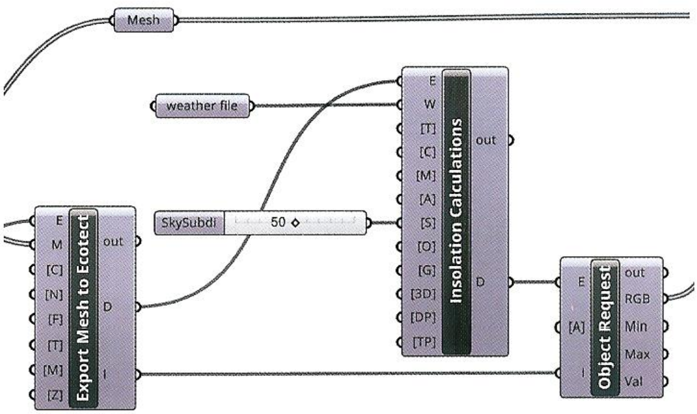
One of most important parameters of *passive energy efficiency* is **insolation**, or the energy received on a given surface during a given time measured in *watt* per *square meter* (Wh/m^2). To calculate insolation, Ecotect uses the **Lambert's cosine law**.



Buildings' geometry and materials should be defined to maximize insolation during the winter and minimize it during the summer. Energy optimization usually focuses on the concept of *passive solar gain*. Passive solar gain is defined as the increase in temperature within a space due to the solar radiation. Insolation is also pivotal in defining the correct position and dimensions of solar collectors and photovoltaic cells. In order to calculate insolation on an arbitrary given mesh, the component *EcoSolCal (Insolation Calculations)* is used. The definition is composed of several steps: define the mesh-geometry to analyze, run Ecotect by *EcoLink* (*Link Ecotect*), and export the mesh using the export guidelines explained in (11.5).



The insolation calculation performed by the component *EcoSolCal (Insolation Calculations)* is cascade-connected (see 11.3) to the component *EcoMeshExport* (*Export Mesh to Ecotect*).

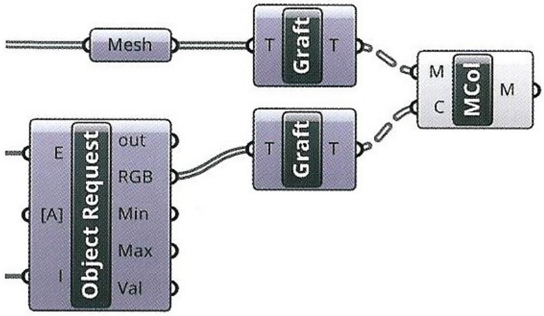


The main inputs are:

- The **Weather File [W]** as explained in 11.2.4.
- The **Terrain Types [T]**: 0 for a location exposed to the wind, 1 for rural settings, 2 for suburban settings and 3 for dense urban settings.
- The type of **Calculation**s to perform **[C]**: 1 for incident solar radiation, 2 for solar absorption-transmission, 3 for sky factor and photosynthetically active radiation.
- The **Sky Subdivision** is a technique that Ecotect uses to subdivide the sky dome in order to perform calculations. The smaller the value, the higher the accuracy. The input [S] must be fed by a number ranging between 1 and 15; [S] can also be set to 50 to run a fast calculation.
- **[G]** Switches analysis between Objects and Grid. By default is set to Object.
- **[DP]** Determines the start and end day of the year for the calculation, and is set using the component *Construct Domain* where A and B range between 1 and 365 (Julian date). By default the domain is [1,365].
- **[TP]** Determines the starting and ending time for the calculation, and is set using the component *Construct Domain* where A and B range between 0.00 and 23.99.

In order to import the results calculated by Ecotect, the component *EcoObjectRequest* (*Object Request*) is connected to the *Insolation Calculation* component using the cascade logic. The input (I) or the faces' indices is connected to the output (I) of the component *EcoMeshExport.*

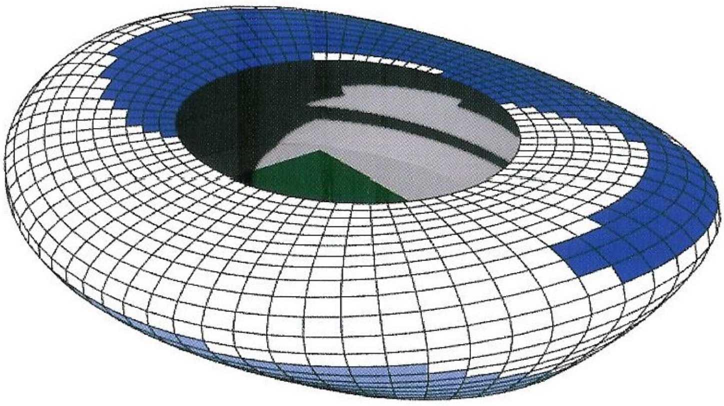
As discussed earlier, the component *EcoObjectRequest* has two main outputs: *(RGB)* and *(Val)*. (RGB) is useful to generate a colour gradient illustration which is helpful in the concpetual stage of design. To display the colour gradient the component *Mesh Colours* (Mesh > Primitive) is connected to the output *(RGB)* of the component *EcoObjectRequest*. The numeric values are returned by the output (Val).



11.6.1 Practical Exercise: Positioning photovoltaic panels

The following exercise demonstrates how insolation analysis can guide the positioning of photovoltaic panels on a cover. The algorithm divides the cover into a set of panels and identifies the panels where insolation is maximized. The complete exercise can be acessed using the QR code.

7



11.7 Analysis Grids

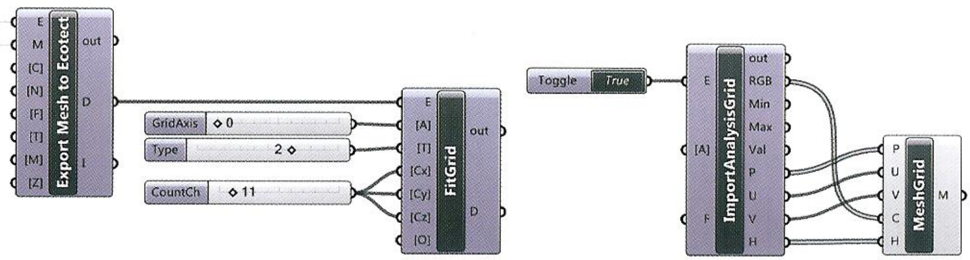
Many calculations, in particular those related to lighting, are performed on analysis grids. As we already explained, grids are set of points which can be considered sensors placed in the space. Grids can be set according to the following methods:

- Grids can be fitted to the extents of selected objects in Ecotect, by the component *EcoFitGrid* (*FitGrid*);
- Grids can be defined using the component *Eco2DGrid* (*2dAnalysisGrid*) which sets the dimensions and the position of grids.

In both cases, two additional components are required to be connected: *EcoGridRequest* (*ImportAnalysisGrid*), which imports analysis values and *EcoMeshGrid* (*MeshGrid*), which draws the grid in Rhino.

FIT GRID

In order to adapt a grid to a mesh geometry exported into Ecotect, the *EcoFitGrid* component is connected to *EcoMeshExport* using the cascade logic.



EcoFitGrid requires:

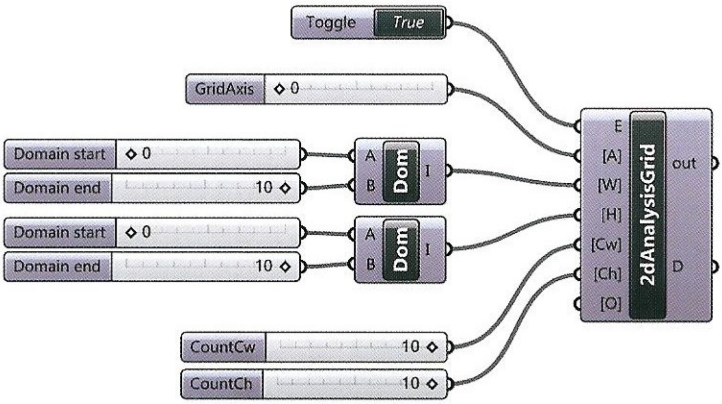
- The reference plane [A] can be defined by a *Number Slider* set to: 0 (XY plan), 1 (YZ plane) or 2 (XZ plane);
- The type of fit [T] which specifies whether the grid will be created *within* the mesh geometry (slider set to 0), *around* the geometry (slider set to 1), *adapted* to the actual shape (slider set to 2) or *larger* than the object (slider set to 3).

The three inputs [Cx], [Cy], [Cz] control the count of faces in three main directions. Input (O) defines the offset from the reference plane.

Once a grid analysis is completed, the results can be imported using the component *EcoGridRequest* (*ImportAnalysisGrid*). The input [A] sets selected attributes for the specific analysis. For instance, for a daylight analysis (see 11.8) the *daylight factor* will be returned when A=0 and the *daylighting levels* will be returned when A=1. Other attributes can be set specifying numeric values as specified in the contextual menu. Lastly, the component *EcoMeshGrid* (*MeshGrid*) generates a colored mesh according to the analysis values.

2D ANALYSIS GRID

The component *Eco2DGrid* (*2dAnalysisGrid*) is useful when manually defining a grid's characteristics; in particular: the plane [A], the domain width [W] and height [H] and the count of faces in W and H directions.



It is recommended that grids are created such that domains in W and H directions are both positive or negative. In other words the grid must lie entirely in one of four quadrants of Rhino's space.

11.8 Light Control

Light distribution plays a large role in a building's energy use as well as visual comfort. As a result, the optimization of light distribution is a key strategy in passive design. The component *EcoLightCal* (*LightingCalulations*) analyzes illumination levels and distribution of light using the parameters:

- Daylight Factors;
- Daylighting Levels;
- Internally Reflected;
- Externally Reflected;
- Sky Component.

Light distribution calculations requires the light levels outside the buildings, including the direct sunlight and the indirect illumination of the sky. The ***sky illuminance*** value (lux), is derived from statistical analysis of dynamic outdoor sky illuminance levels. The following image shows design sky illuminance levels for different latitudes. Several tools available online can be used to determine recommended Lux levels for particular locations.

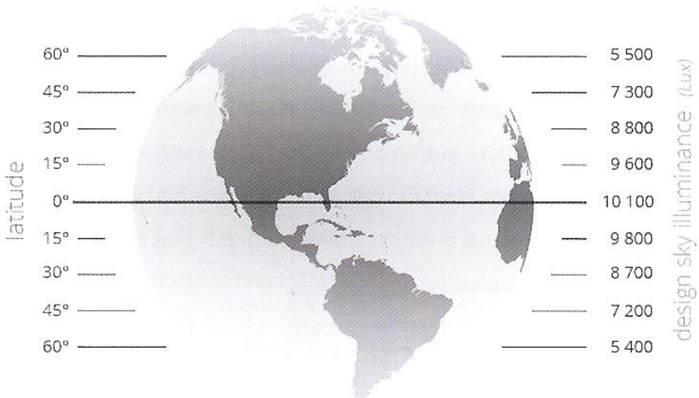
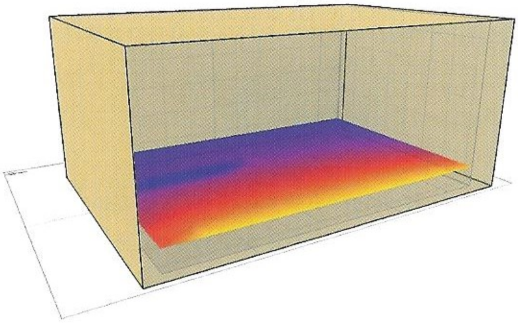


FIGURE 11.7
Sky illuminance (in lux) according to latitude.

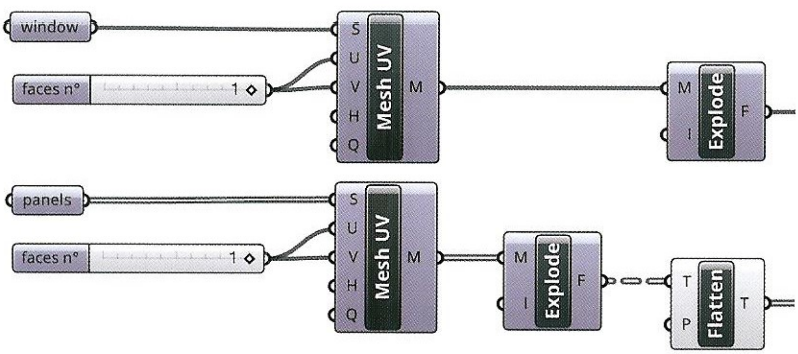
11.8.1 Visual comfort of a room

This example will cover a visual comfort analysis of a prismatic room with an east-facing glass wall.



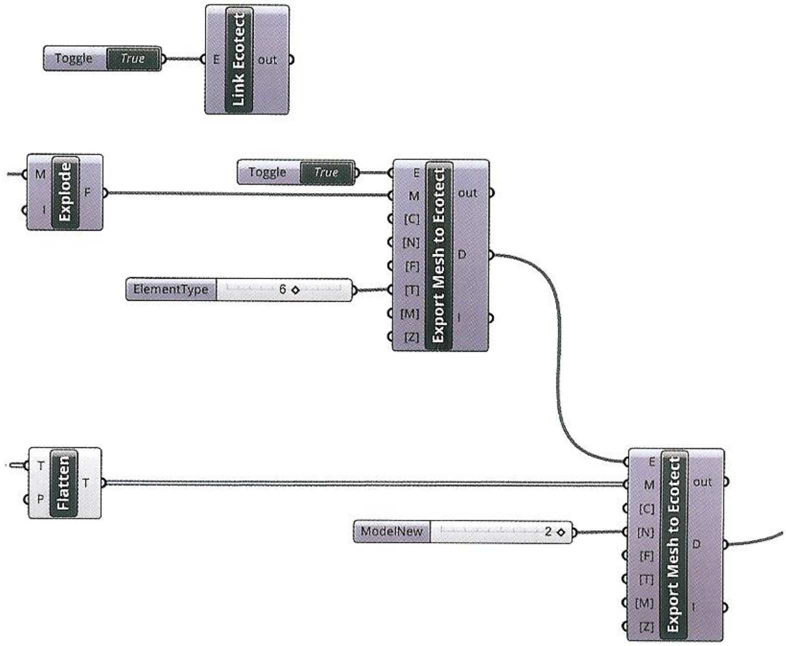
SETTING THE GEOMETRY

The first step in the definition is to define a box in Rhino or Grasshopper with the dimensions: 12000 millimeters length, 6000 millimeters width and 6000 millimeters high. Subsequently, the box is exploded into individual faces, the east face is collected by a *Surface* container component renamed as *window* and the remaininig faces are collected by a *Surface* container component renamed as *panels*. The second step is to convert the two set of surfaces into two sets of meshes, so different materials can be assigned to the window and panels. Each surface is converted into mesh using just one subdivision for U and V directions, to output one face for each mesh. Lastly, the two sets of meshes are exploded using the component *Mesh Explode,* and the *panels* data is then flattened.



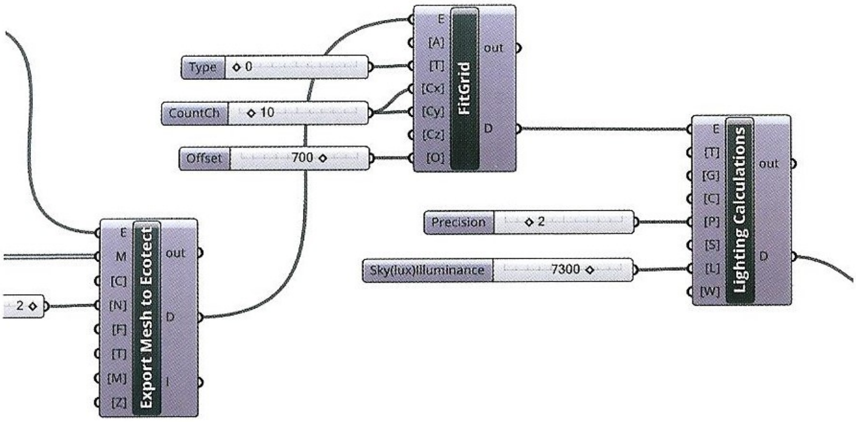
EXPORTING MESHES INTO ECOTECT

Meshes are exported using two *EcoMeshExport* components. In order to assign the *window* type to the east-facing meshes we have to set the [T]-input to 6. The *panels* meshes are exported with the default [T] value (7).



SETTING THE ANALYSIS GRID

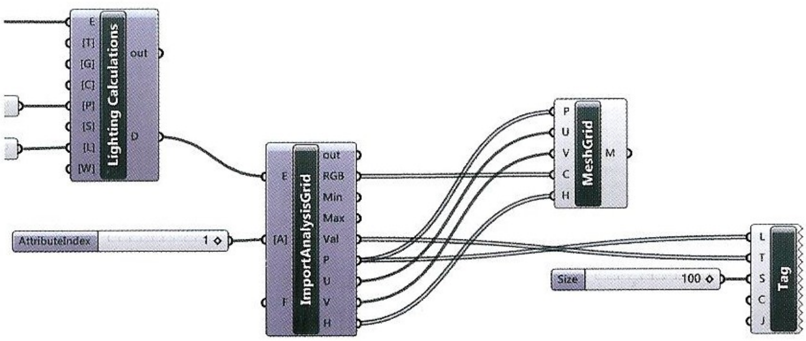
In order to define the analysis grid the component *EcoFitGrid* is used, setting the inputs such that: the grid will be parallel to the XY plane (input [A] set to 0), with an offset of 700mm (input [O]). The grid will be created within the object by specifying [T]=0) and that the cells count is 10 in [Cx] and [Cy] directions. The input [Cz] is irrelevant in this case since the analysis will be performed on the 2D grid of points at 700mm from the reference plane.



LIGHTING ANALYSIS

Lighting analysis is performed by the component *EcoLightCal* (*Lighting Calculations*), in the example the precision input [P] is set to 2 (High Precision) and the *sky illuminance* (see figure 11.6) input [L] is set to 7300 lux. The input [T] specifies whether the calculation is performed over the analysis grid (0, default value) or over point objects (1). Input [C] sets the available lighting calculations.

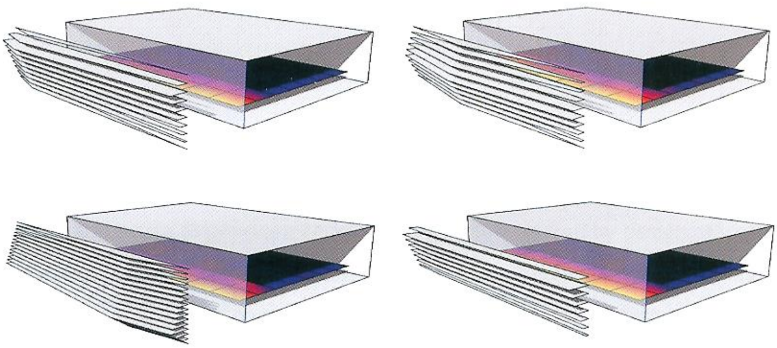
In order to import results the component *EcoGridRequest* (*ImportAnalysisGrid*) is used, while the component *EcoMeshGrid* (*MeshGrid*) generates a colored mesh according to the analysis values. As discussed in 11.7, the component *EcoGridRequest* (*ImportAnalysisGrid*) enables selected attributes to be set for the specific analysis through the A-input.



The calculated values can be directly displayed on the grid using the *Text Tag 3D* component (Display > Dimensions), feeding the L-input with the P-output (grid's points) of *EcoGridRequest* (*ImportAnalysisGrid*). The T-input needs the calculated values and (S) the size of text.

11.8.2 Practical Exercise Galapagos + GECO, visual comfort optimization

The following exercise demonstrates how to optimize - using Galapagos - the visual comfort of a space by modifying number and shape of a set of sun shading devices. The complete exercise can be acessed using the QR code. [8]



This chapter is part of a research developed by **Maurizio Degni**, a specialist in the field of energy and environmental analysis related to complex systems with a particular focus on parametric and optimization strategies. Maurizio Degni worked for several offices in Italy such as Studio Kami and J.M. Schivo where he collaborated on many international competitions, projects and research activities. With Arturo Tedeschi he designed the NU:S Installation within the Cloister of Bramante (Rome) and he had a main role in the computational design of the NU:S Parametric Shoes an avant-garde project that matched advanced design techniques and prototyping technologies. From 2012 he is a tutors assistant for the AA Rome Visiting School.

Post Digital Strategies

Pragmatic Computation in Grasshopper

Brian Vesely

makelite design studio

Data: big data, small data, ubiquitous data. Algorithms fed by data are indifferent to output. The practice of organizing data to generate geometry has been explored by design offices since the early 1980's, discussed in academic texts, and demonstrated at institutions. Beyond the commonly perceived formal scope, the advantage of parametricism is the iterative calculability to formulate pragmatic data. Parametricism is often used and abused as a justification for design decisions without close study of the inputs; decisions are often made without integrity to produce a formal output. *Post digital strategies*[1] considers the organization of data as a tool to produce a pragmatic output. Parametricism and pragmatism are seemingly polar concepts; however, a pragmatic parametric workflow can expose latencies in even mundane problems.

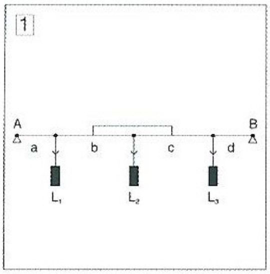
As discussed in Chapter 9, *Digital Simulation,* a hanging chain can be illustrated in Kangaroo following Hooke's Law. As an alternative, a hanging chain model can be calculated using computation graphical statics in Grasshopper. Graphical statics is an equilibrium method of visualizing forces acting on a rigid body by graphically representing forces with vectors that are proportional in magnitude and equivalent in direction to the final geometry of the body. Graphical statics is a pragmatic approach to form-finding that, when used in conjunction with optimization solvers, can produce intelligent output. Intelligent graphic statics applied to a reversed cantenary structure can be defined by following the subsequent algorithmic workflow diagrams[2]. Workflow diagrams – similar to spatial, conceptual, and structural diagrams – are an essential means to define the procedural logic of complex multi-part algorithms. Workflow figures 1-12 decompose a graphical static definition into its respective sub-parts; within each part, variables that are required to be calculated are noted. Figure

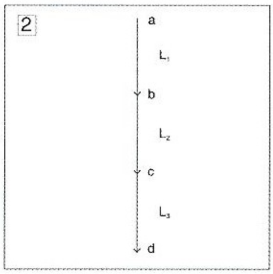
1. The graphical statics algorithmic logic presented was researched in the independent study, *Post Digital Strategies, at the Illinois School of Architecture with students Catherine Lie and Ailin Wang.*

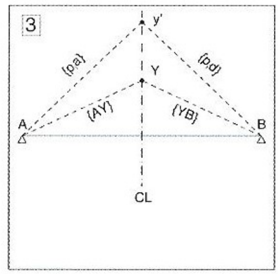
2. For additional information on this topic, see: Allen, Edward, and Wac Zalewski. *Form and Forces: Designing Efficient, Expressive Structures*. Hoboken, N.J.: John Wiley & Sons, 2010.

13 illustrates how the iterative calculability of Grasshopper can be used as optimization input. The workflow diagrams used to organize the data of a generic static definition of a reversed cantenary are illustrated and discussed in the following pragmatic example.

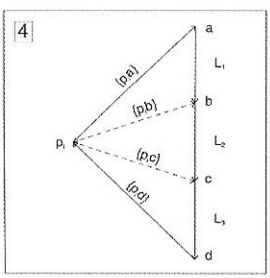
Loads calculated for the given span and material are converted into vectors and placed incrementally along the loading line (figure 1). The loads are transposed to the initial force diagram as the load vectors {L_1, L_2, L_3} (figure 2). The vertical height point (y') is set at the centerline of the loading line two times the desired funicular height (Y), and tangent vectors {p_ia} and {p_id} are constructed by connecting points (A) and (B) to point {y'} (figure 3).

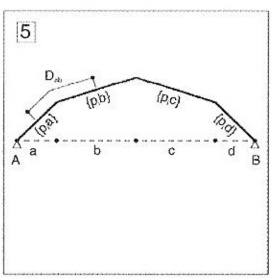


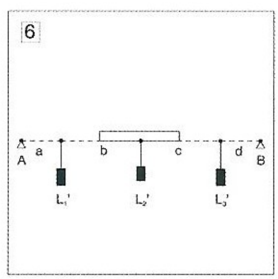




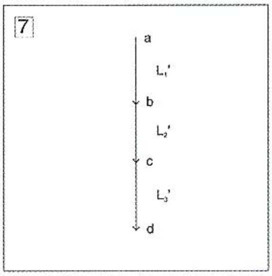
The form diagram tangent vectors {p_ia} and {p_id} are translated to the load line and intersected to define the initial pole (p_i). Points (b) and (c) are connected to (p_i), defining vectors {p_ib} and {p_ic} (figure 4). The length of each loading segment is measured (i.e. D_{ab}) on the initial loading diagram (figure 5), and the loads are updated for each segment (figure 6).

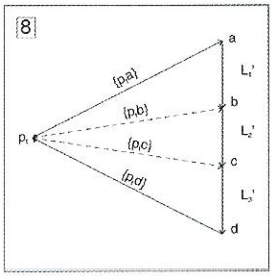


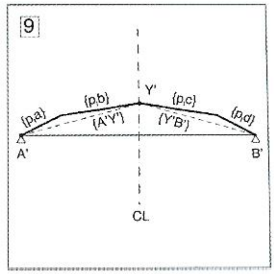




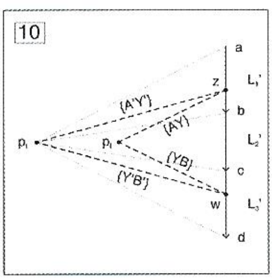
The loads are transposed to the final force diagram as the load vectors {L_1', L_2', L_3'} (figure 7). The trial pole (p_t) is defined as a point in proximity to the final load line. Points (a), (b), (c), and (d) are connected to the point (p_t) defining vectors {p_ta}, {p_tb}, {p_tc}, and {p_td} (figure 8). The defined vectors are used to construct the trial form polygon. The intersection of the geometric centerline and the trial form polygon defines point (Y'). Vectors {A'Y'} and {B'Y'} are defined by connecting points (A') and (B') to point (Y') (figure 9).

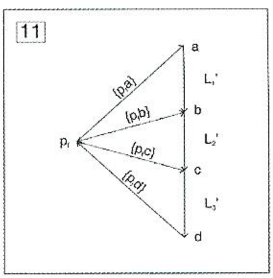






Vectors {A'Y'} and {B'Y'} are translated to originate at point (p_t) and intersected with the load line defining points (z) and (w). Vectors {AY} and {YB} are translated to originate at points (z) and (w) respectively and intersected to define point (p_f) (figure 10). Points (a), (b), (c), and (d) are connected to the point (p_f) defining vectors {p_fa}, {p_fb}, {p_fc}, {p_fd}, and the final force diagram (figure 11). The calculated vectors are translated to construct the final form diagram through point (Y) (figure 12).





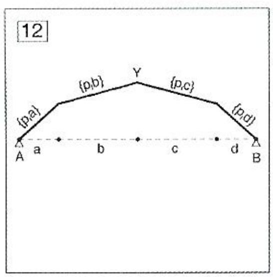
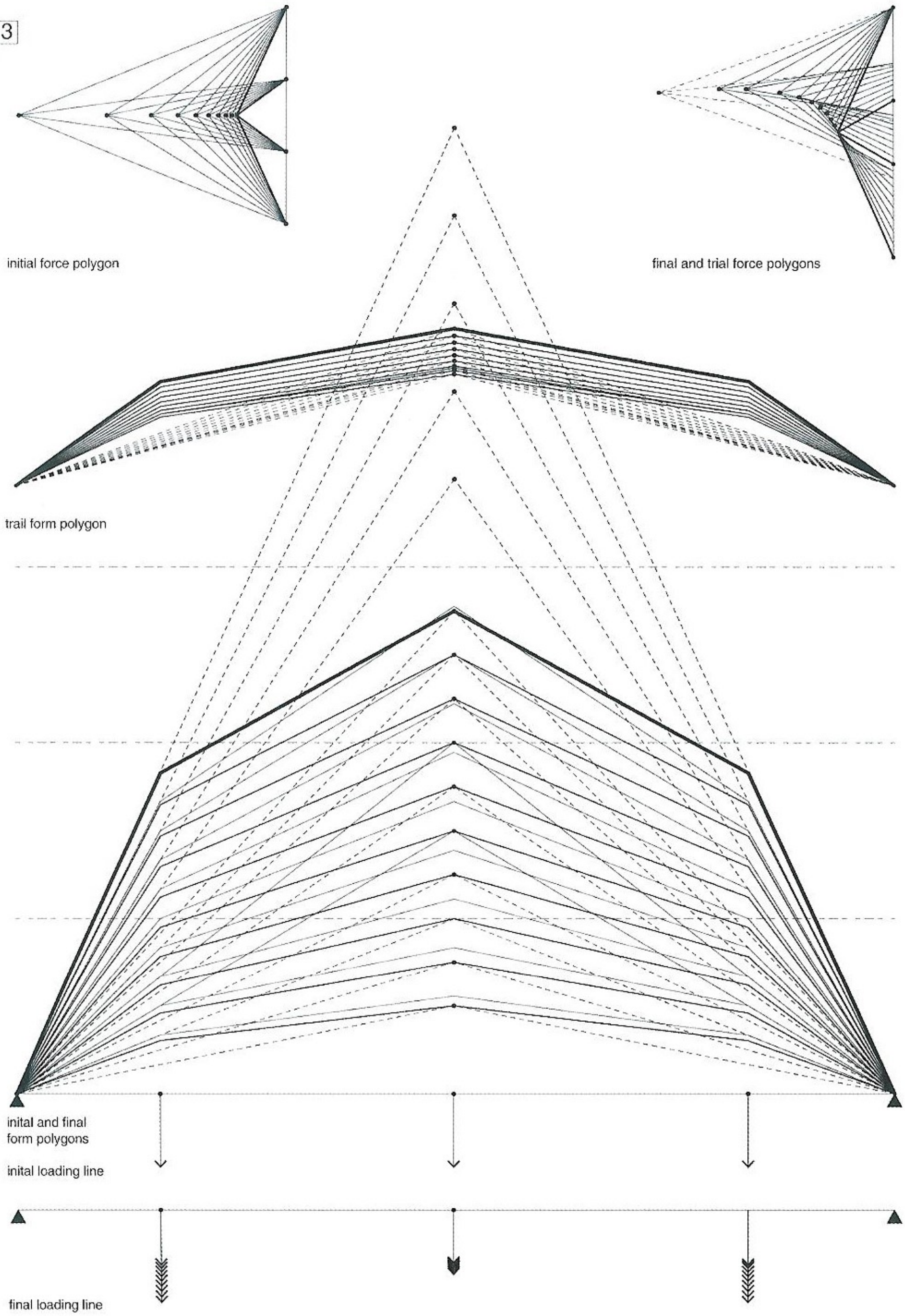


Figure 13 displays an abbreviated illustration of solutions in a composite diagram. The solver Goat can be used to minimize the *objective*, axial force, as calculated by measuring the length of the vectors respectively in the final force diagram. The calculated axial force can be used to determine the thickness of the vault using the stress equation:

$$F_{allow}=P/A \text{ expanded to } \mathbf{t}=(\mathbf{P}\text{ lb})/((\mathbf{12}\text{ in})*(\mathbf{F_{allow}}\text{ lb/in}^2))$$

13


initial force polygon
final and trial force polygons
trail form polygon
inital and final
form polygons
inital loading line
final loading line

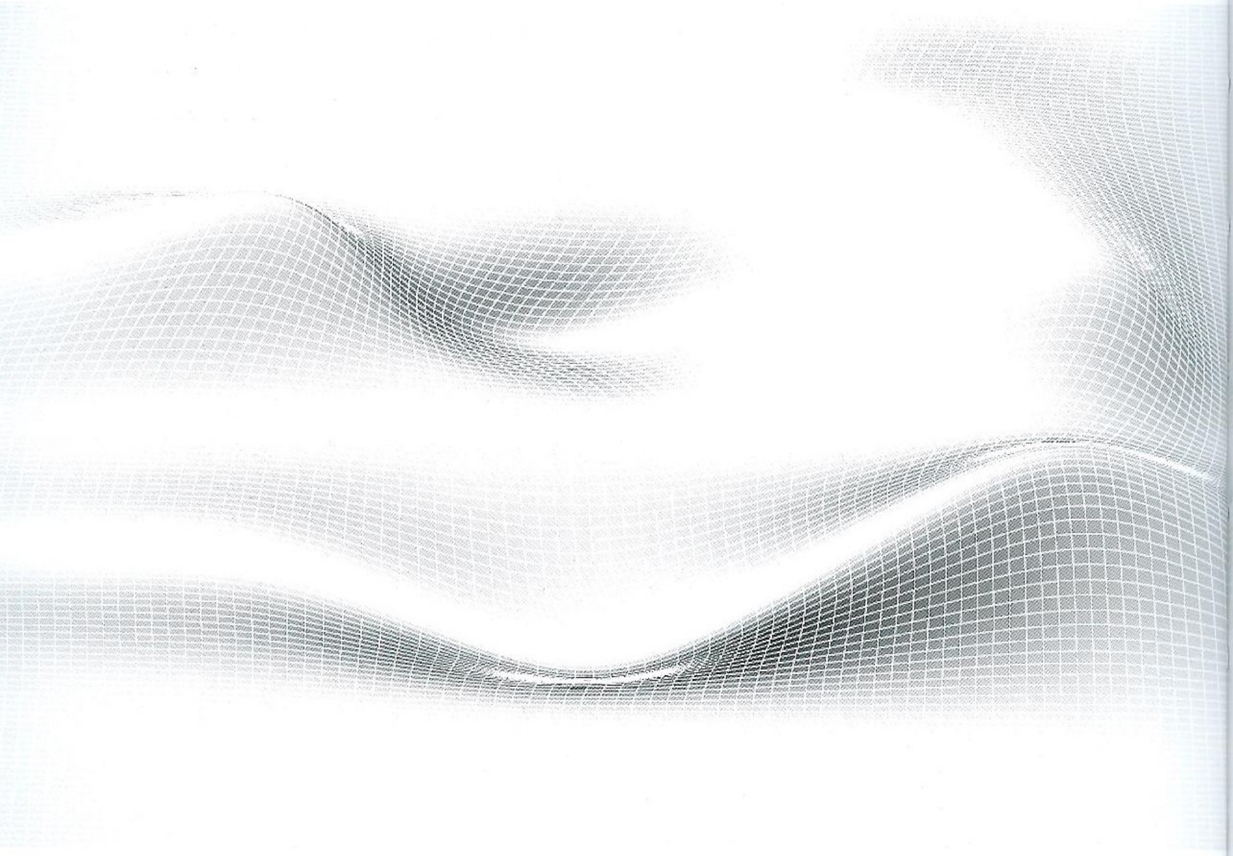
Consideration within the definition must be given to the minimum and maximum rise-to-span ratios for issues of stability as well as the minimum thickness[3]. These considerations can be overcome by knowledgeably setting the range of the sliders as well as the conditional statements with respect to the axial force-thickness relationship. Once the generic workflow has been defined in Grasshopper, the sliders for the number of segments and the span can be expanded, the position of the restraints can be manipulated, and series components can be added to define a multitude of vaults. The complete funicular definition as well as other Grasshopper definitions and tutorials can be viewed and downloaded from [www.make-lite.com].

This pragmatic afterword is intended to amplify a point that the author Arturo Tedeschi makes throughout the text: Grasshopper, and more generically, algorithms, are tools to organize data; it is up to the user to determine the output. The text's compilation of examples should not be seen as how-to manual to complete formal exercises; instead, the knowledgeably crafted examples should be seen as a means to understand the way data is organized. In the end, understanding the way data is organized unleashes the power of Grasshopper.

Brian Vesely

Brian Vesely is the lead Designer at *makelite*. He graduated with a Masters Degree from the University of Illinois School of Architecture in 2012. As a Visiting Lecturer at the Illinois School of Architecture, he was part of the Detail and Fabrication Concentration (d+F), teaching courses in fabrication, computation, digital workflow, and design. He operates from experience in the fields of architecture, land surveying, and engineering. His work occurs at the intersection of speculative narratives and anecdotes of construction, exposing tectonic latencies. He creates fun-rigorous-smart assemblies exploring the potentials of fabrication, computation, and emerging infrastructural and ecological potentials.

3. Additional attention is required to consider all loads imposed on any structure and the results should be reviewed by a qualified engineer; this example is used to demonstrate pragmatic data logic only.



I am City, we are City

Francesco Lipari
Founding principal at OFL Architecture

When I think about the future, my mind comes back to the 80's to the dawn of the new millennium. I clearly remember that period, no doubt that it seemed like the future to me: a time when the scale between technology and humanity was perfectly balanced. The future, in fact, always coincides with something simple that simplifies our life. Today, we live in complex times which are a sort of borderline for humanity. In response to rapid changes of lifestyles and habits, humans must adapt and change their own nature, influencing the urban environment, understood as the ideal ground of sharing and exchange. Everything changes at an unprecedented speed that we are not prepared for, dragged by a two-speed rhythm. On the one hand there is *technology*, which proceeds inexorably as it had suddenly realized to be in late on original plans; on the other hand, there is *humanity*, which gathers those who are similar in feel an estrangement in which they must live, as the current stage doesn't make them protagonists. These are unprecedented times, we have never been so close and so distant at the same time. The natural, adaptive human-mechanism seems to be jammed, or simply inadequate to face the deep changes which happen cyclically at ever shorter intervals.

The shape of earth and circularity of events leads to idea that our world is finite, predetermined. Humans are naturally inclined to interpret signs, trying to anticipate changes that actually seem to have already happened. Time variations and continuous adaptations are just the effects of something which has already happened and recurs, in a new shape.

The man of today, for instance, is closer to a man who lived in fifth century than to a man who lived in nineteenth century. The latter, in turn, is similar to a man lived in roman age. In this scenario, apparently anachronistic – in which is difficult to find humanity – cities grew up. Cities are a representation of our society and, today we are losing control.

The typical model of independent and self-sufficient cities no longer exists. We are experiencing a new "ageographical" city, without precise spaces and forced to demonstrate its strength to face and bear new rhythms. Each city will become a *follower* of a bigger city within a process which is similar to the editing of an infinite book, with an infinite number of pages, hosted in a universal library. Cluster-cities will become nodes of an algorithmic-generative definition, tailored to support a global vision. Moreover, the current technology-overflow is generating two different effects on the same actors, i.e. the human beings. By increasing the level of interaction and communication two kinds of societies emerge. The first one is a *high-communication* society with no territorial and expressive boundaries, while the second one is a *low-communication* society, which reinterprets obsolete technologies with the dual aim to save memory and identity; such a society also uses the technology-divide as an elitist and less controllable type of communication.

All this leads to a disaggregation of consolidated models of urban management and opens the doors to the individual contribution as a model for generation and implementation. We will experience a new age where humans will be a part of a complex bio-mechanical mechanism structured and integrated with urban environment. Technology will further simplify its hardware and also the city will be smaller and smaller, personal, up to coincide with the humans themselves.

Francesco Lipari

Francesco Lipari is a Sicilian architect based in Rome. Recipients of several Prizes for Young Architects Francesco has been lecturer at the MAXXI and MACRO museum and curator of several architecture projects. He's the founding principal of OFL architecture and was formerly senior architect at the Fuksas office in Rome and Mad in Beijing. He's also founder of CityVision (http://www.cityvisionweb.com/) an innovative architecture platform with the aim of generating a dialogue between the contemporary city and its future image – http://www.oflstudio.com/.

Parametric Urbanism: a New Frontier for Smart Cities

Paolo Fusero, Lorenzo Massimiano, Arturo Tedeschi, Sara Lepidi[1]

Planum. The Journal of Urbanism, no. 27. vol. 2/2013, pp. 1-13.

[...] In only a few short years the concept of Smart Cities has evolved from an evocative representation of futuristic digital metropolises into an overblown term used to indicate all manner of virtuous processes: economic, environmental, technological, social, etc. It now appears that any human activity we wish to qualify in positive terms cannot avoid being tagged with the adjective "smart". This induces a reflection: on the one hand the fact that smart thinking applied to cities and territories is becoming "trendy" can be considered positive as it contributes to raising public awareness about such issues as environmental sustainability and technological innovation. On the other hand, the smart phenomenon induces a form of disorientation for the abusive use of the term and the consequent dilution of its importance to research. [...] From our point of view, a less explored, and thus even more interesting frontier, is that which can be defined as Parametric Urbanism. In other words, the use of parametric software in urban design, not only to three-dimensionally represent projects at the urban scale [...], but precisely as part of the processes of developing the tools of urbanism, as an instrument for assisting the planner in evaluating diversified scenarios and making informed decisions. For example, it would be interesting to understand what contribution can be made by parametric tools to the construction of effective models of compensation (options on permutations, flexible distribution, etc.), or what assistance they can bring to the rationalisation of the layout of services within a territory, based on the real needs of users [...]. Or further still, the simulation of alternative scenarios to urban transformations based on a choice of diverse building typologies or densities of inhabitation.

1. Paolo Fusero, full Professor at the G. d'Annunzio University of Chieti-Pescara, Faculty of Architecture; Lorenzo Massimiano, researcher in Urbanism at the G. d'Annunzio University of Chieti-Pescara, Faculty of Architecture; Arturo Tedeschi, independent researcher, co-director since 2012 of the Architectural Association Rome Visiting School; Sara Lepidi, Graduate in "Territorial and Environmental Planning", 2012-13, G. d'Annunzio University.

From Typological to Procedural Thinking

The use of the computer in the world of design has accelerated a direction of research culturally rooted in the avant-gardes of the 1960s. This branch recently arrived at the elaboration of theoretical apparatuses constructed around a notion that compares architecture to systems in evolution and mechanisms of self-regulation. The research focuses substantially on the pragmatic passage from the concept of the *type* to one of *process*. This involves overcoming the logic of composition to the advantage of a "neo-positivist" vision founded on a multiplicity of interconnected elements (objects, materials, data). Through a propagation of effects, the variation
of one single element can bring about a modification to an entire architectural or urban organism. Hence the final form is an output generated by a procedure, almost as if it were unknown inherent to the system. Design is thus transformed into a sort of "definition of intelligent rules".
[...] The reciprocal fecundation between architectural theories and the possibilities offered by digital technologies consented the rapid extension of the utilisation of the computer. From a simple tool of production (focused on increasing the speed of operations) it has evolved into a refined system of control that permits previously unimagined formal explorations. The introduction of complex programming techniques and parametric software offers designers unexpected possibilities, making it almost impossible to predict the effects these tools will have on design simulations.
Simplifying to a great extreme, parametric software can be considered a programming platform – working within three-dimensional CAD environments – capable of generating form through the definition of a conceptual diagram that becomes the only "drawing" developed by the designer. This diagram explicates the associative ties between a range of input data, and generates an output that is a system of dynamic and modifiable forms.

From Reactivity to Proactivity

Networks of communication, sensors and *smart objects* are able to gather consistent masses of data. This data is in turn filtered through specific software created precisely to organise this material and facilitate its comprehension. A challenge to multinational digital companies of the future will lie precisely in the development of systems with an ability to define relations between heterogeneous data and create innovative forecasting models. Models will no longer be elaborated according to statistic methods, but instead through the *real-time* evaluation of significant parameters and indicators capable of influencing the design process at the urban scale.
For example, the overlapping reading of data as information alphabetisation or the offering of on-line services and relative user feedback, may suggest the territorial decentring of services that no longer require direct relations with users. To the same degree, data related to *co-working*, when compared to

correlated parameters, may offer important indications on urban mobility and energy consumption. Or, data from external sensors used to measure air quality, solar heat gain, ventilation, acoustic pollution, etc., may indicate solutions that optimise the energy efficiency and comfort of settlements. Within scenarios of this type, parametric software may even serve as a tool for experimenting with "new models of Urban Plans". No longer comprised solely of a series of "routine" drawings produced to satisfy normative requirements, they become a dynamic three-dimensional representation. These models can be constantly updated by *smart data*, which thus assumes a "proactive" role, anticipating phenomena and future changes in order to implement rapid and opportune actions and decisions. No longer a traditional "reactive" system controlled by mechanisms of consequential decision-making, but almost a new paradigm of planning supported
by a collective intelligence that is the fruit of choices, decisions and interactions supported and guided by technology. [...] The use of parametric software may thus offer designers a very interesting tool for experimenting with new methods of designing. Projects employing parametric logics are distinguished, in their form and content, from those developed according to traditional methods. The first important difference is conceptual, as mentioned: the final result is not established by the designer *a priori*, but is the result of a process of elaborating selected *smart data*. The second difference lies in the vivacity of the system that structures it: the passage from a static to a dynamic system. The formal result is no longer the definitive crystallisation of a particular line of reasoning, but instead a "snapshot" that captures the status of a process in continuous evolution. It is generated to react to variations, autonomously adapting to stimuli it receives in accordance with the rules established by the designer during the phase of concept design. Projects thus evolve on their own, almost demonstrating a capacity for *selforganisation*[2]. Despite their adaptive capacities, it is clear (and this is directed at those sceptics already thumbing their noses at the thought of substituting the designer during the "creative" process) that parametric platforms always require an *a priori* selection of data to be processed. It is precisely through the control of *input data* that designers are able to evaluate alternative solutions, utilising a "snapshot" of a *work in progress* to satisfy desired qualitative performance values. The phase of data selection and reactive control thus represents a crucial moment within the entire process [...].

2. Brian Team Consulting, "Teoria della complessità", from http://braint.net, last view on 5th July 2013.

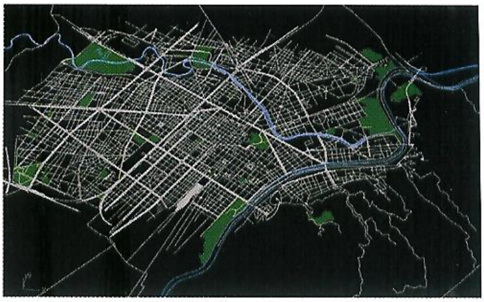
Tools and methods for parametric urbanism

Andrea Galli
Architect at Carlo Ratti Associati

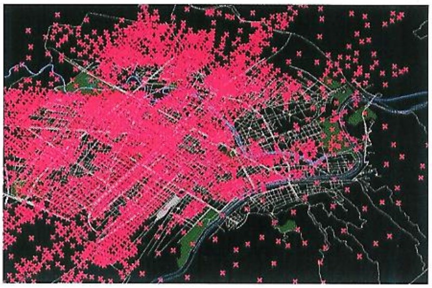
The modern city is a complex and dynamic system that, today, can benefit from the huge possibilities of technology scattered ubiquitously, changing the way the urban context talks to us, and how we live and interact within it. All this represents an incredible opportunity in planning the city of tomorrow, but it is not enough. To find the right answers to the questions that "big data" generates we can rely on the potential offered by parametric tools in elaborating – with total control– the new resources available, which would be unmanageable and redundant in traditional analysis and planning processes. In order to prepare our cities to become "Smart" it is necessary to adapt the design process that rules the change. In fact, many famous examples suggest the opposite, the parametric approach is not based on *shape*, but instead on the relationship between every elementary part of the complex system, where if an element changes, all the other elements will self-organize. This behaviour, called "adaptive" allows to design systems where – once the rules that describe the relationships between every element are set down – if the system changes unexpectedly (emergent behaviour[1]), it will transform itself according to these rules, defined as parametric variables. Parametric Urbanism allows us to understand and control the behavior of complex systems, such as our cities or parts of them, in order to plan his reaction to the real time change of the data context around it. To describe this context it is necessary to make a selection of data; this inherently represents an important design choice, because it deeply influences the final result. In addition to data typologies, the source of them is in the same way very important: data for example can be open-data released by public administrations, freely available to everyone; this data can be easily found on the Internet and embody the social and economic aim of unlocking the potential value of a huge quantity of information, usually under-used. Vector data related to a particular region of interest can be easily found on the portal *OpenStreetMap.org*[2], downloading the .osm file associated

1. Emergent behaviour: can appear when a number of simple entities operate in an environment, forming more complex behaviours as a collective.
2. Open Street Maps: is a collaborative project to create a free editable map of the world. The major driving forces behind the establishment and growth of OSM have been restrictions on use or availability of map information across much of the world and the advent of inexpensive portable satellite navigation devices.

to any geographical region. This file can be imported inside a Grasshopper algorithm using *Elk*[3] and it incorporates inside his framework metadata useful for a differentiated treatment of the geometries according to their respective tags. Some recurring tags are: *minor roads, major roads, waterways, railways, railway: station, highway: bus stop, highway: pedestrian, parking, buildings, amenity, leisure: park, leisure: garden, landuse: industrial, area, etc.*



This database can be furthermore increased with other information regarding, for example, the geographical distribution of people, the position of commercial hubs, schools, libraries, etc. These information come from the extended basin of geo referenced datasets published by public administrations and they are organized in tables or shapefile[4] which can be imported into Grasshopper, respectively through *Lunchbox* and *Finches*[5]. An exact overlapping can be accomplished using *gHowl*, thanks to the perfect correspondence with the coordinate system of the Open Street Maps geometries.

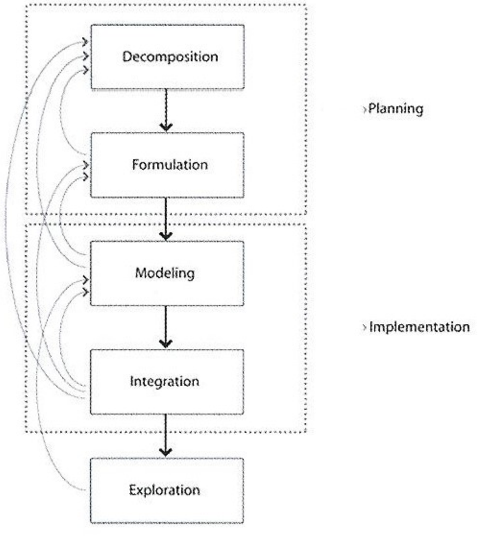


3. Elk: a plugin to generate map and topographical surfaces using open source data from OSM and Shuttle Radar Topography Mission, Timothy Logan, www.food4rhino.com/project/elk

4. Shapefile: or Esri shapefile, is a popular geospatial vector data format for geographic information system software. Shapefiles spatially describe vector features: points, lines, and polygons, representing, for example, water wells, rivers, and lakes.

5. Finches: a component developed by Nicholas De Monchaux to import, export and batch processing shapefiles.

The result is a hybrid model made of vector data and metadata that allows to exploit parametric modelling tools in association with a new informative dimension. A thinkable use of a certain model could be the visualization and the interrogation of the database, as with GIS. However it is possible to explore some more interesting direction, processing the database in order to generate an unlimited range of potential design proposal on different scales. Taking inspiration from the approach suggested by Anas Alfaris within his PhD thesis *Emergence through Conflict: The Multi-Disciplinary Design System (MDDS),* developed at MIT under the supervision of W. J. Mitchell[6] in 2009, we can think to structure the entire design process using a dynamic and holistic framework divided in five phases according to reversible connections, but meanwhile, maintaining a robust hierarchic organization. The five phases are *decomposition, formulation, modeling, integration* and *exploration* and can be referred to different design stages: *decomposition* and *formulation* belong to the planning stage, *modeling* and *integration* belong to the implementation stage, the *exploration* represents the final verification stage where the previous phases can be validated or rejected. The *decomposition* requires to divide the global project vision into elementary units and goals. The *formulation* contemplates the specific analysis of every single goal so to independently define the processes to realize it, but also the way in which they influence each other. During *modelling* every goal is developed in a cluster in order to transform into algorithms (thanks to the parametric tools) what was defined during *formulation*. The *integration* consists in implementing the relationships between the different clusters, until now described as isolated elements. The *exploration* is the phase that allows to verify the match between results and what was expected during *formulation*. In fact, the *integration* can easily generate unexpected consequences, that will eventually require the process to be reviewed starting from *modelling*.



The described flow chart represents a structured way to manage the design process of complex entities such as our cities. Following it Grasshopper can assume the role of an extremely flexible platform to organize into a single data stream large scale data, which come from different sources, describing their relationship through mathematic rules but also counting on the huge 3D modelling possibilities offered by the Rhinoceros environment.

The opportunity of a real time comparison of many alternative scenarios, obtained by changing the mutual influence of the system parameters, is a scientific method for an organic design of our cities considering their real necessities and ensuring consistency with the project constrains and goals.

Andrea Galli

Andrea Galli, independent researcher, graduated in engineering and architecture from the Politecnico di Torino, collaborates with the office Carlo Ratti Associati of Turin from 2012.

6. William J. Mitchell: considered one of the world's leading urban theorists. Through the work of his Smart Cities research group at the MIT Media Lab, he pioneered new approaches to integrating design and technology to make cities more responsive to their citizens and more efficient in their use of resources. He likened tomorrow's cities to living organisms or very-large-scale robots, with nervous systems that enable them to sense changes in the needs of their inhabitants and external conditions, and respond to these needs. W. J. Mitchell died in 2010 at the age of 65.

Playful computation

How Grasshopper3D & its Plugins increased my creativity with five project examples

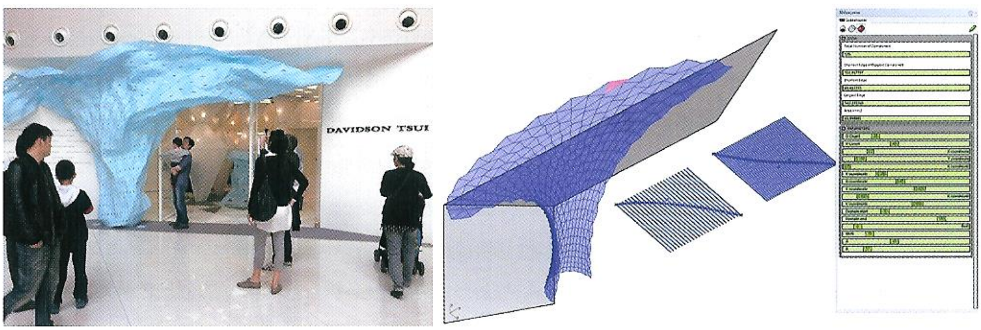
Arthur Mamou-Mani AAdip

Grasshopper3D for Rhinoceros is not just a tool, it is a platform for a community to learn and share computational design projects. It is an educational environment, which together with its numerous plugins will help you to understand mathematics, computer science, physics, fabrication and much more. It will break down barriers between design fields and soon you will be discussing vectors with a jewellery designer, topology with a wheel-chair fabricator or recursion with an engineer (true story). Grasshopper is complicated but it is also friendly and playful. Unlike most design software, it wants you to understand and learn how your tools and items can be linked together to create systems, instead of repetitive and time consuming 3D modelling (A.K.A. CAD monkeying). You will be linking components together with wires and not modelling curves and surfaces in space as you might be used to. Of course Grasshopper can simply accelerate your current workflow, but it can also transform the way you design in a much deeper way. You might have to imagine your project as a set of interconnected elements as opposed to a finished sculpted object. This will encourage you to see the reason behind forms and perhaps understanding these rules may even bring you closer to understanding how forms occur in nature. After doing a couple exercises within this book, you might see scary things like {0;0} N=1, or users around you could start asking if you have "flattened or grafted your list?" This is strange and unfamiliar territory for any early user, so ... DO NOT PANIC! We have all been through that phase of confusion. You will soon understand how the data trees will help you organise and manipulate the information going through the "fancy wires" and how you can apply successive operations to an initial input, which is the beginning of automating your tasks. Why are we all excited about this tool? It has opened up coding to visual people. It is a door to a complex world that we would have not dared to enter if Grasshopper was not here. Moreover it is a community of friendly people that you can find on the buzzing forum Grasshopper3d.com. Please note that if you ask for a new tool it is very likely that you will be asked to write it yourself and add it to the many plugins available on Food4Rhino.com. Users often write the plugins in their free time and out of interest, learning programming from nothing - so anyone can do it. This is the ultimate proof of Grasshopper's success, it grows with the users and soon you will be one of them, wanting to add your creative brick to the parametric building. Following are details of five projects I worked on and used Grasshopper within; to inspire you and show you how

Grasshopper can be used in a practical way. Arturo Tedeschi, a teacher like myself, knows that you will probably not follow tutorials unless you are first inspired. You need to see not just the theory but how it can be applied, so let's look at these five projects of different scale and see how they benefited from the use of Grasshopper.

Project 1
RIBA Windows Project for Davidson Tsui in Xintiandi Style Shanghai[1]
with James K. Cheung of ARUP Associates

A parametric or associative approach to 3D modelling means that your model can be changed based on different input or parameters. By changing the parameters you are creating variations in the system not changing it. This makes Grasshopper3D very useful for quick changes in the geometry of a project or to produce many variations of a design system. This project illustrates how useful Grasshopper has been for these variations.

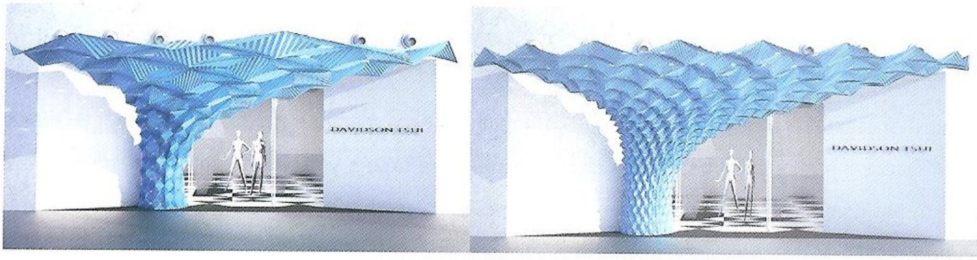


Left: the finished piece in Xintiandi Shanghai - Photo by XiaoHui Chen. Right: The parametric model for the main tree with the remote control panel for display outside the GH interface

The project's main element is a giant folded tree made of 576 laser-cut origami components. By folding the component, the sheet bends into a hyperbolic paraboloid (known as hypar). This is a ruled surface that can be described as two non-parallel lines in space connected with adjacent straight lines and forming two parabolas when cut in section. We used Grasshopper to populate the

1. The Fitting Room - More information: http://mamou-mani.com/RIBAShanghai/ - Client: Davidson Tsui and Shui-On Land - Team: Lead Architects: James K. Cheung (ARUP Associates London), Arthur Mamou-Mani (Mamou-Mani) - Architects: Suzanne Li (ARUP Associates Shanghai), Benita Tan (ARUP Associates Shanghai) Engineer: Eric Sturel (ARUP Associates Shanghai) Collaborators: Arup Associates London: Paul Jeffries, Daryl Miles, Stephen Philips – ARUP Associates Shanghai: Sunglin Tsai, Marta Colas, Milo Gu, Wonder Wu, Vicky Feng – Mamou-Mani: Laetitia Sfez – Tongji and Shanghai university: Gamzar Lee, Alexander Gösta, Tracy Zhang, Rachel Zheng, Zhang Licheng, Zhou Yejun, Liu Xun, Zheng Raven, Cao Sophie, Shi Ji, Chen Kaiyu, Xu Lei, Hu Zhixuan, Zhang Ying.

double-curved geometry of the tree with the folded hypars. To do so, we used the UV parameters of the surface but divided the resulting trimmed surfaces into sets of four in order to get mirroring components. To control the size of the panels, we changed the density of the division and used a custom "graph mapper" tool which projects values from the x axis to the y axis using a user-defined curve. This is how the gradient from small to large and from flat to folded was created.



One system with different output, two renders showing how the parametric model adapted to fit the fabrication.

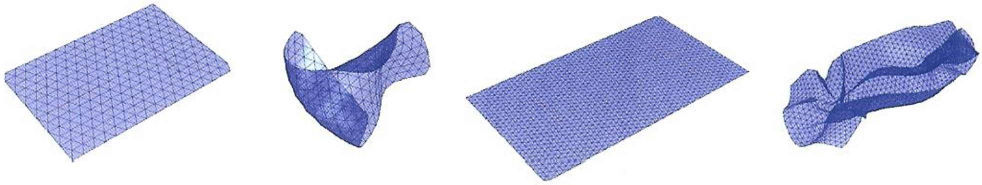
For fabrication, we were limited by the size of the machine used. At first the largest components were around 2m wide. This is possible with a large laser-cutter but in Shanghai we only had access to an A2-sized bed. Instead of drawing the whole project again we simply changed the density and the graph. The graph mapper is one of the GH components however it cannot be changed through sliders and therefore cannot be connected to the evolutionary solver "Galapagos". To make sure the longest edge would fit within an A2 laser-cut bed size we used Galapagos to change most of the sliders (UV density, graph mapper, distance of corner from original surface) until it met the required length of 594 mm.

Project 2
The Magic Garden, RIBA Regent Street Windows Project for Karen Millen[2]

The Magic garden - A 30 m long dress on Regent Street - Photo by Agnes Sanvito.

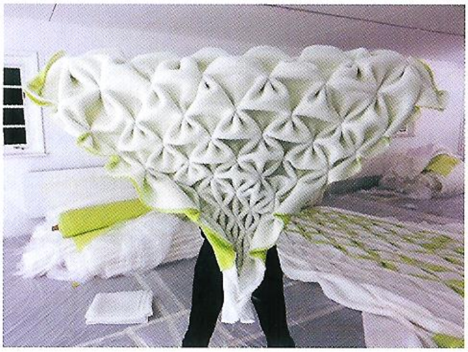
2. The Magic Garden – More information http://mamou-mani.com/KarenMillen/ – Client: Karen Millen Fashions Ltd.

The Magic Garden is a giant and architectural dress that flows along the thirty metres of Karen Millen's flagship store. The material is a cheap fabric that can be found in sport shoes called 3D Spacer Mesh. The mesh has a thickness that scatters light very well but also resists to bending. Both properties are used in this piece. The varying "smocking" pattern used all along the fabric creates different levels of strength and changes the width of the piece itself.



Pinching fabric using hinge forces in Kangaroo.

The physics engine Kangaroo (by Daniel Piker) was used to study the types of patterns that can be used. In this case the plugin "folds" the patterns from a flat sheet instead of drawing an additional geometry with extra area onto an existing surface. In a few words, Kangaroo works by plugging forces and anchor points into a physics engine. In this case we used the hinge force. Points will move in space constrained and defined by a rest angle. The anchor points can be moved to simulate someone pushing or pulling the fabric. When the anchor points move but the rest angle wants to stay at zero it creates a resistance, which is what was used here. Since we had several pinch points, we replaced anchor points by spring forces between two points to pinch together. Spring forces will move to a given rest length therefore the pinch points are linked with springs of a rest length of zero.

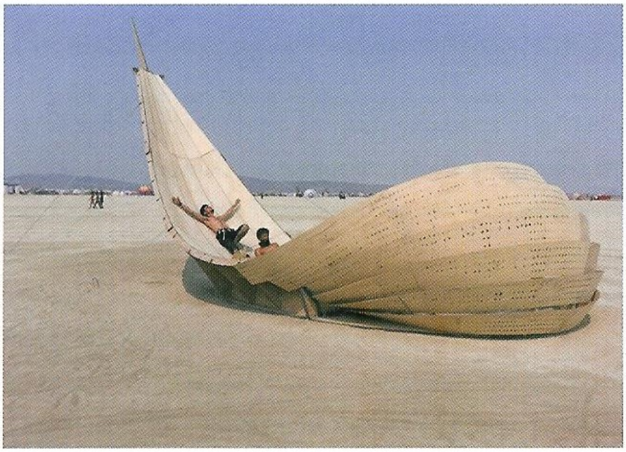


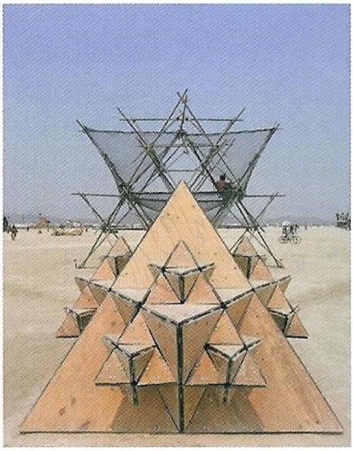
By changing the distance between the smocking patterns we could change the width of fabric.

Contributors: Gemma Metheringham, Andrew Hardy Architect: Arthur Mamou-Mani Lead Collaborator: Jack Munro Collaborators: Savvas Havatzias, Madeeha Maham, Megan Sadler, Heloise Delegue, Anna Rootes, Sharon Toong, Sarah Shuttleworth, Dan Dodds, Andrei Jippa, Jessica Beagleman, Adam Holloway, Stephanie Holloway, Timothee Tan, Christina Leung, Philip Hurrel, Christopher Mount, Michael Clarke, Jacob Alsop, Nick Chung.

Project 3
Shipwreck and Fractal Cult at the Burning Man Festival[3]
with Toby Burgess and Diploma Studio 10 at the University of Westminster

These two projects were designed and built with our students at the University of Westminster. Grasshopper3D was used throughout. On Shipwreck the whole project was created from two initial NURBS curves on Rhinoceros, the top and bottom ones. The project's elegance is based on these two curves; they were therefore carefully changed in Rhinoceros while being used as input for the Grasshopper3D model. The fins flowing all along the structure were also changed within the parametric model and the whole cutting pattern changed accordingly.



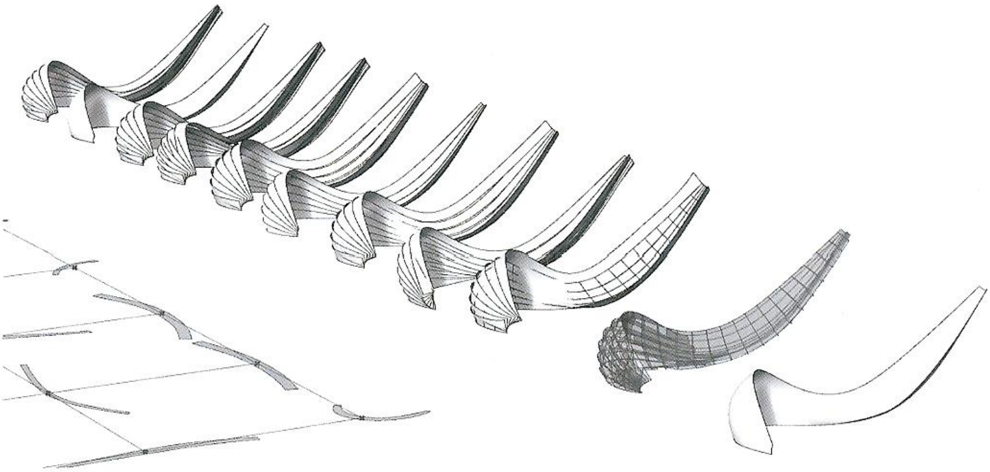


Shipwreck and Fractal Cult Designed by Georgia-Rose Collard-Watson and Thanasis Korras of Diploma Studio 10.

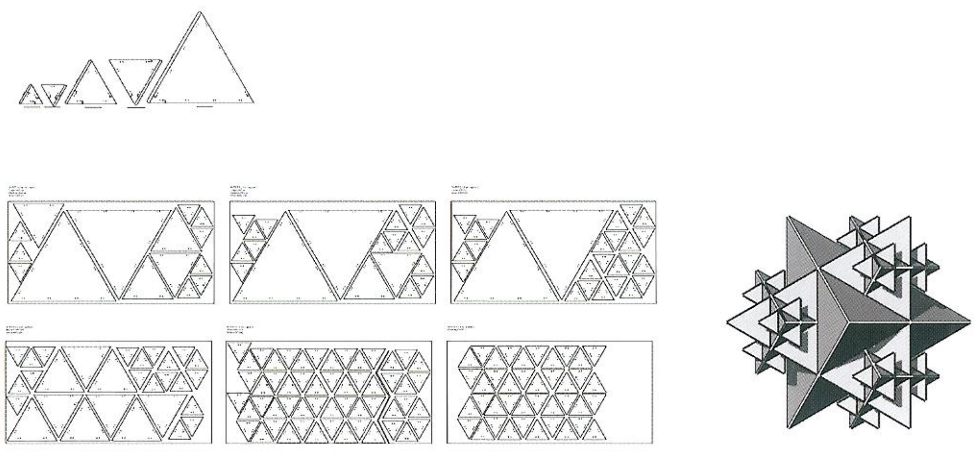
The form of Fractal Cult emerged from a three-dimensional Koch snowflake, which is well-known recursive fractal geometry. This was done using the plugin Hoopsnake, by Yannis Chatzikonstantinou, which allows the user to plug the end of a Grasshopper definition with its own beginning, hence the recursion. Once the triangles were generated we positioned hinges along the triangle edges so that as many triangles as possible could be used any way up and around. We then placed bolt holes for

3. Shipwreck and Fractal Cult at the Burning Man Festival More Information: http://wewanttolearn.wordpress.com/2013/09/12/building-fractal-cult-and-shipwreck-at-burning-man-2013/ Team: Toby Burgess and Arthur Mamou-Mani (Project Directors), Thanasis Korras (Designer of Fractal Cult), Georgia Rose Collard-Watson (Designer of Shipwreck), Jessica Beagleman (Food & Meals), Natasha Coutts (Camp and Rentals), Sarah Shuttlesworth, Andy Rixson, Luka Kreze, Tim Strnad, Philippos Philippidis, Nataly Matathias, Marina Karamali, Harikleia Karamali, Antony Joury, Emma Whitehead, , Jo Cook, Caitlin Hudson, Dan Dodds and Chris Ingram. Engineers: Ramboll Computational Design (RCD) – Stephen Melville, Harri Lewis, James Solly.

hinges with different offset according to whether the hinge is open at an obtuse or acute angle. This maintains a constant gap between the triangles throughout the whole structure. The definition then outputs 2D drawings for laser cutting.



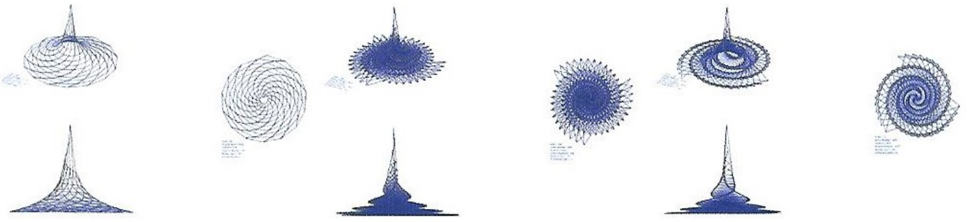
Subtle variation of the fins in the shipwreck within the same parametric model.



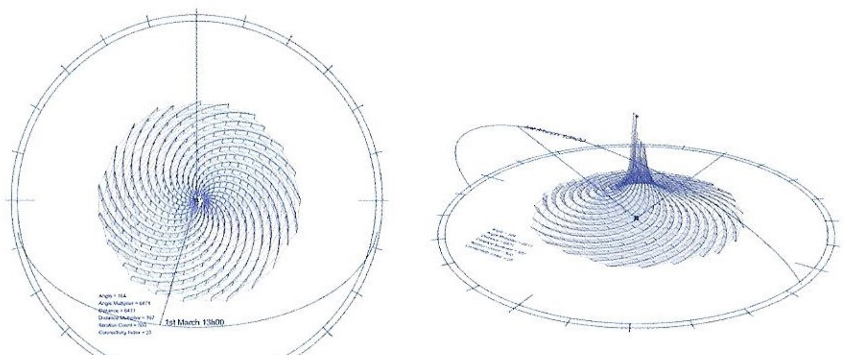
The acute and obtuse angles of the hinges defines the size of the triangles parametric model by Dan Dodds.

Project 4
Eco-Resort in the desert of New-Mexico[4]
with Stephen Melville, Harri Lewis and Neil Clements of Ramboll RCD

Recursion is the process of repeating items in a self-similar way. For this project we moved and rotated vectors in space to form a roof that responds to sunlight and resembles a galaxy. One of the great aspects of Grasshopper3D is that you can link all kinds of tools and simulations to your parametric model directly. In this case, we linked the recursion to a sun angle that was imported to Grasshopper from Ecotect using the plugin Geco by Thomas Grabner and Ursula Frick. This allowed the roof to grow while making sure that sunlight stays out during summer and gets in during winter.



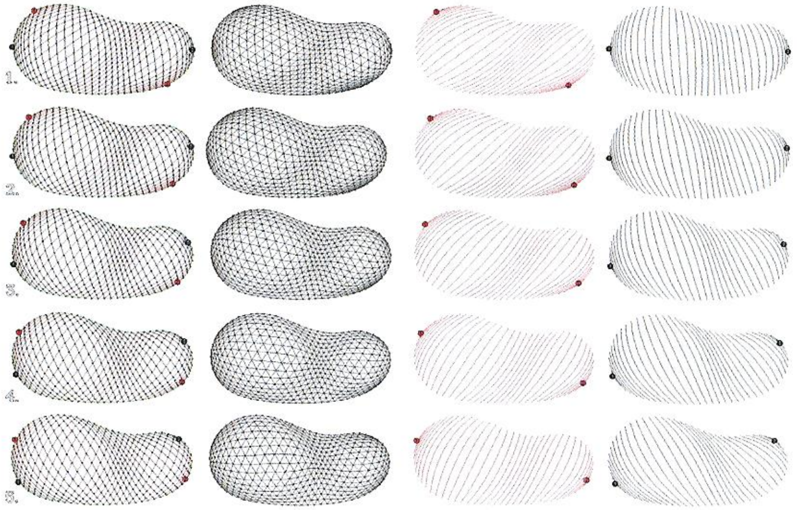
Recursive operation generating the roof mesh using Hoopsnake, slight variations in the angles and distances.



Letting light in winter and keeping it out in summer - Using Geco and Hoopsnake.

4. Eco-Resort in the desert of New-Mexico - More info:http://mamou-mani.com/ecoresort/ - Project by: Mamou-Mani Architects and Ramboll RCD.Team for Mamou-Mani: Arthur Mamou-Mani, Savvas Havatzias, Jack Munro Team for Ramboll RCD: Stephen Melville, Harri Lewis, Neil Clements (environmental engineer).

Project 5
Chester Zoo's Heart of Africa Biodome
by Proctor and Matthews Architects[5]



Chester Zoo is the first project for which I used Grasshopper. It is a 200 meters long biodome hosting animals and plants from central Africa. The directors Stephen Proctor and Andrew Matthews wanted

5. Chester Zoo by Proctor and Matthews Architect – More information: http://www.proctorandmatthews.com/project/heart-africa – Client: Zoological Gardens Chester Project Architect: Gareth Wilkins, Design Team: Stephen Proctor, Andrew Matthews, Stephanie Southward, Constance Leibrock, Benoit Sanson, Kengo Skorick, Arthur Mamou-mani, Steven Davies, Anna Marchant - Help on VB.net from Hyunbai Jun.

a pattern that would respond to the roof's geometry not just an orthogonal grid. We also needed to increase the greenhouse effect by making the openings very large and the structure very thin. Mark Cabrinha, on the Grasshopper3D.com forum, recommended using geodesic curves for the gridshell. These curves are the shortest path between two points on a surface. At first we populated the ring of the roof with points and connected them with these curves. Later on we generated a mesh from the intersecting points, which allowed us to organise all resulting triangles into separate branches. What worked really well is that the resulting structure was larger at the peaks and denser in the valleys. This made the structure very strong and convinced the engineers, who were advocating a more conventional truss and purlin option.

Throughout all these projects, Grasshopper3D with its plugins was used to define, generate, analyse, simulate or fabricate geometry. In all cases, it accelerated the workflow and created unique projects that would not have been created without this fertile platform. GH opens up a world of possibilities and connects designers, architects, engineers, fabricators and artists around topics that go beyond the tool itself. This book is your chance to go beyond the Voronoi, develop your left-brain and be a more complete professional. Now type "Grasshopper" in your command bar and enjoy!

Arthur Mamou-Mani is a registered architect and director of the chartered architecture and computational design practice Mamou-Mani Architects (http://mamou-mani.com). He is unit master of Diploma Studio 10 at the University of Westminster (http://WeWantToLearn.net) and advanced Grasshopper 3D tutor for Simply Rhino Ltd. (http://SimplyRhino.co.uk). Arthur has taught parametric design tools, digital fabrication as well as environmental and structural simulation at many leading academic bodies such as the Architectural Association School of Architecture and the UCL-Bartlett. He has previously worked with Atelier Jean Nouvel, Zaha Hadid Architects and Proctor and Matthews Architects for three years before setting up his own company.

The CloudBridge

"Most of the time, the fastest way to get anywhere is a straight line leading from point A to point B. But as a new conceptual project from Arturo Tedeschi architects, that's not necessarily the most efficient, or most beautiful, way to go about it. [...] CloudBridge links two mountainsides via an ethereal, cloud-like structure. Though bridges are often a visual statement of strength, CloudBridge's latticed form and non-linear path creates a super-stable bridge that appears to float between mountains, blending into its natural surroundings.

The CloudBridge, a concept developed by Arturo Tedeschi and Maurizio Degni.

The surreal concept, bolstered by extreme engineering, is a reflection of Tedeschi's work in general. "Nowadays engineering and architecture are evolving just by improving in small steps the 'state of art,' and designers are gradually losing their visionary attitude" he says. "This is also evident in technology, automotive and product design." Cloudbridge is meant to push the boundaries and be a playful look at how the trajectory and appearance of a bridge can be altered using parametric design".

(From "This cloud-Like bridge was created with an algorithm", WIRED digital magazine 10.04.2013)

References

THEORY

Barrios, C.R., *Parametric Design in Architecture: Fundamentals, Methods, Applications*, Birkhauser, 2014.

Chang, W., Vasilakos, A.V., *Molecular Computing: Towards a Novel Computing Architecture for Complex Problem Solving (Studies in Big Data)*, Springer, 2014.

Pilia, E.J., Fornasari, F., Verso, F., Boschi, L., Monopoli, D., Casolari, S., *Citymakers*, Deleyva, 2013.

AD. Architectural Design, *Computation Works: The Building of Algorithmic Thought*, Volume 222 (March/April 2013).

AD. Architectural Design, *Inside Smartgeometry: Expanding the Architectural Possibilities of Computational Design*, April 2013.

AD. Architectural Design, *Material Computation: Higher Integration in Morphogenetic Design*, Volume 216, No. 2 (March/April 2012).

Giussani, E., *Expressionist Organic Paths in Architecture: From Finsterlin to Gehry*, Lap Lambert Academic Publishing, 2012.

Pizzigoni, A., *Ingegneri e Archistar. Dialogo sul moderno costruire fra miti e mode*, Christian Marinotti Edizioni, Milano, 2011.

Coppola, C., *Attraverso l'Architettura, rappresentazione procedurale e algoritmi per il progetto generativo*, Alinea Editrice, Firenze, 2011.

DETAIL, *Analogue and Digital*, Volume 4, (July/August 2010).

Woodbury, R., Sheikholeslami, M., *Elements of Parametric Design*, Bentley Institute Press, 2010.

Schumacher, P., *The Autopoiesis of Architecture*, Voll. I and II, John Wiley & Sons, London, 2010.

Converso, S., *Il progetto digitale per la costruzione, cronache di un mutamento professionale*, Maggioli, Milano, 2009.

AD. Architectural Design, *Pattern*, Volume 79, No. 6 (November/December 2009).

AD. Architectural Design, *Digital Cities*, Volume 79, No. 4 (July/August 2009).

Otto, F., *Occupying and Connecting: Thoughts on Territories and Spheres of Influence With Particular Reference to Human Settlement*, Axel Menge, 2009.

Bechthold, M., *Innovative Surface Structures: Technology and Applications*, Taylor &Francis, 2008.

Balmond, C., *Informal*, Prestel, Munich, 2007.

Saggio, A., *Introduzione alla Rivoluzione informatica in Architettura*, Carocci, Roma, 2007.

Terzidis, K., *Algorithmic Architecture*, Architectural Press, Oxford, 2006.

AD. Architectural Design, *Techniques and Technologies in Morphogenetic Design*, Volume 76, No. 2 (March/April 2006).

Aranda, B., Lasch, C., *Tooling*, Princeton Architectural press, New York, 2006.

Coppola, C., *Computer e creatività per l'architettura, intelligenza artificiale e sistemi formali*, Alinea, Firenze, 2005.

AD. Architectural Design, *Emergence: Morphogenetic Design Strategies*, July 2004.

Koolhass, R., *Junkspace*, Quodlibet, 2001.

Thompson, D'Arcy W., *On Growth and Form*, Cambridge University Press, 1992.

ALGORITHMS

Wang, X., *Algorithmic Enhancements for Computing the Frame of a Finitely Generated Bounded Polyhedron*, Proquest, 2012.

Deb, K., *Multi-Objective Optimization using Evolutionary Algorithms*, Wiley, New York, 2001.

Goldberg, D., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley, New Jersey, 1989.

Aho, A., Ullman, J., Hopcroft, J., *Data Structures and Algorithms*, Addison Wesley, 1983.

DIGITAL FABRICATION

Jackson, P., *Cut and Fold Techniques for Pop-Up Designs*, Laurence King Publishing, London, 2014.

Jackson, P., *Cut and Fold Techniques for Promotional Materials*, Laurence King Publishing, London, 2013.

Beorkrem, C., *Material Strategies in Digital Fabrication*, Routledge, 2012.

Dunn, N., *Digital Fabrication in Architecture*, Laurence King Pub, 2012.

Nakamichi, T., Ito, C., Miyazaki, Y., *Pattern Magic: Stretch Fabrics*, Laurence King Pub, 2012.

Jackson, P., *Folding techniques for designers from sheet to form*, Laurence King Publishing, London, 2011.

Nakamichi, T., *Pattern Magic*, Laurence King Pub, 2010.

Nakamichi, T., *Pattern Magic 2*, Laurence King Pub, 2010.

Iwamoto, L., *Architectural and Material Techniques*, Princeton Architectural Press, New York, 2009.

Schodek, D., Bechthold, M., Griggs, J.K., Kao, K., Steinberg, M., *Digital Design and Manufacturing: CAD/CAM applications in architecture and design*, Wiley, 2004.

Wolff, C., *The Art of Manipulating Fabric*, Krause Pubns Inc., 1996.

FORM FINDING

Adriaenssens, S., Block, P., Veenendaal, D., Williams, C., *Shell Structures for Architecture*, Routledge, London, 2014.

Descamps, B., *Computational Design of Lightweight Structures: Form Finding and Optimization*, Wiley, 2014.

Gass, W.H., *Finding a Form: Essays*, Knopf, 2013.

Schatz, P., Carline, P., *Rhythm Research and Technology: The Invertible Cube/ Polysomatic Form-finding*, Niggli, 2013.

Knippers, J., Cremers, J., Gabler, M., Lienhard, J., *Construction Manual for Polymers + Membranes: Materials, Semi-Finished Products, Form-Finding, Design*, Birkhauser, 2012.

Pone, S., *Gridshell, i Gusci a graticcio in legno tra innovazione e sperimentazione*, Alinea, Firenze, 2012.

Kourkoutas, V., *Parametric Form Finding in Contemporary Architecture: The Simplicity Within The Complexity Of Modern Architectural Form*, Lambert, 2012.

Otto, F., Rasch, B., *Finding Form*, Deutscher Werkbund Bayern, 2006.

Guccione, M., *Il ponte e la città Sergio Musmeci a Potenza*, Gangemi, Roma, 2004.

Billington, D.P., *Thin Shell Concrete Structures*, McGraw Hill, New York, 1982.

Heyman, J., *Equilibrium of Shell Structures*, Clarendon Press, Oxford, 1977.

Kilian, A., Ochsendorf, J., "Particle-spring systems for structural form finding," *Journal of the International Association for Shell and Spatial Structures*, vol. 46, no. 148 (2005): 77-84.

SCRIPTING

Tang, M., *Parametric Building Design Using Autodesk Maya*, Routledge, 2014.

Pearson, M., *Generative art, a practical guide using processing*, Manning Publications Co., Shelter Island (NY), 2011.

Burry, M., "Scripting Cultures: Architectural Design and Programming," *AD. Architectural Design*, John Wiley & Sons, 2011.

Pisca, N., *YSYT Maya MEL Basics for Designers*, 2008.

Shiffman, D., *Learning processing*, Morgan Kaufmann, 2008.

Beri, M., *Python*, Apogeo, Milano, 2007.

ARCHITECTURAL GEOMETRY

Charleson, A., *Structure As Architecture: A Source Book for Architects and Structural Engineers*, Routledge, London, 2014.

Carrera, E., Briscetto, S., Nali, P., *Plates and Shells for Smart Structures: Classical and Advanced Theories for Modelling and Analysis*, Wiley, 2011.

Gjerde, E., *Origami Tessellations: Awe-inspiring Geometric Designs*, A.K. Peters Ltd., 2009.

Lopez, D., Ambrose, G., Fortunato, B., Ludwig, R., *The Function of Form Farshid Moussavi*, Actar, 2009.

Meredith, M., *From Control to Design: Parametric/Algorithmic Architecture*, Actar, 2008.

Pottmann, H., Asperl, A., Hofer, M., Kilian, A., *Architectural Geometry*, Bentley Institute Press, USA, 2007.

Bensøe, M., Sigmund, O., *Topology Optimization: Theory, Methods and Applications*, Springer Verlag, Berlin and Heidelberg, 2003.

Schock, H.J., *Soft Shells: Design and Technology of Tensile Architecture*, Birkhauser, 1997.

MATHEMATICS

Burry, J., Burry, M., *The New Mathematics of Architecture*, Thames & Hudson, 2012.

Rucker, R., *Infinity and the Mind: The Science and Philosophy of the Infinite*, Princeton University Press, 2004.

Frixione, M., Palladino, D., *Funzioni, Macchine, Algoritmi*, Carocci, Roma, 2004.

Saad, Y., *Iterative Methods for Sparse Linear Systems*, Society for Industrial and Applied Mathematics, 2003.

Lorenz, E.N., *The Essence of Chaos*, University of Washington press, 1996.

MONOGRAPHS

Hiesinger, K.B., *Zaha Hadid: Form in Motion*, Yale University Press, 2011.

De Costa Meyer, E., *Frank Gehry: On Line (Princeton University Art Museum Series) (Princeton University Art Museum Monographs)*, Yale University Press, 2008.

Hadid, Z., Schumacher, P., *Latent Utopias: Experiments Within Contemporary Architecture*, Springer Verlag GmbH, NY, 2003.

Nicoletti, M., *Sergio Musmeci, organicità di forme e forze nello spazio*, Testo & Immagine, Collegno, 1999.

Faber, C., *Candela: The Shell Builder*, Reinhold Publishing Corporation, New York, 1963.

Decoded QR list

QR Code no. 1, p. 242: http://www.lepenseur.it/aad_book/chapter_05_01.jpg

QR Code no. 2, p. 304: http://www.lepenseur.it/aad_book/chapter_07_01.jpg

QR Code no. 3, p. 336: http://www.lepenseur.it/aad_book/chapter_08_01.jpg

QR Code no. 4, p. 390: http://www.lepenseur.it/aad_book/chapter_09_01.jpg

QR Code no. 5, p. 427: http://www.lepenseur.it/aad_book/chapter_10_01.jpg

QR Code no. 6, p. 429: http://www.lepenseur.it/aad_book/chapter_10_02.jpg

QR Code no. 7, p. 458: http://www.lepenseur.it/aad_book/chapter_11_01.jpg

QR Code no. 8, p. 465: http://www.lepenseur.it/aad_book/chapter_11_02.jpg