

Performance Based Interactive Analysis

Odysseas Georgiou, Paul Richens, Paul Shepherd

Department of Architecture & Civil Engineering, University of Bath, Bath BA2 7AY, UK
odysseas.georgiou@hub.com.cy, p.n.richens@bath.ac.uk, P.Shepherd@bath.ac.uk

Abstract

This paper re-approaches structural engineering through an interactive perspective by introducing a series of tools that combine parametric design with structural analysis, thus achieving a synergy between the architectural shape and its structural performance. Furthermore, this paper demonstrates how the design can be realised into an efficient structural form by applying novel techniques of form-finding through the exploitation of the generated analytical output. The combination of these tools and their parametric control contributes to a new design approach that outrides the generation of single solutions and enables a deeper exploration of the design parameters leading to multiple performance-based outcomes. This paper describes the integration between a Parametric Design software, *McNeal's Grasshopper 3D* and a Finite Element Analysis software, Autodesk's *Robot Structural Analysis*. The generated synergy between form and structure is demonstrated through a series of case studies through which the interactive control of the parameters enables the designer to iterate between a range of form-found solutions.

1. Introduction

The rapid advance in CAD technology has enabled architects to overcome the traditional design boundaries and to transform any imagined shape into a persuasive building. In this context, structural design is lagging behind and engineering's engagement with architecture is still restricted. This traditional approach cannot keep up with the modern design process and the engineer is unable to give feedback to the architect's design, often stalling the design process. While a large variety of tools serving architectural geometry, such as parametric modelling, is available for use by architects, allowing limitless capabilities and speed in design, the engineering industry remains adherent to traditional structural analysis and design techniques.

This paper introduces a novel design procedure through a series of tools that interactively manage and form-find structures. An overview of combining a parametric design software, Grasshopper 3D [1] with a structural analysis

software, Autodesk Robot Structural Analysis [2], through the use of computer programming is presented. This combination enables the engineer to retain better control over his designs by employing a performance based approach and it speeds up the design process while allowing for the exploration of new optimum structural solutions [3, 4]. By extending the capabilities of parametric design to include and implement structural analysis, the engineer can move away from the traditional ways of structural thinking and relax the technical boundaries. The results deriving from a structural analysis need no longer be single solutions to problems but parameters that feed into the architectural form and conclude to an optimum shape.

To demonstrate the capabilities of a structural extension to parametric design, two examples of distinct design cases are presented in this paper. Both cases employ the performance based approach by first interactively visualising structural analysis results and then utilising them to iterate between a range of structurally optimum solutions which respond to the designer's control.

2. Interactive Structural Modelling and Analysis

The interactive framework that enables a performance based analysis was achieved by linking together McNeal's Grasshopper 3D and Autodesk's Robot Structural Analysis using C# programming language. Grasshopper 3D is a plugin built in .NET framework to access McNeal's core software, Rhinoceros 3D in order to control and manipulate geometry in a generative manner. The functionality of Grasshopper 3D (GH) can be extended by writing code in C# or VB DotNet programming language to create custom components. In parallel, Robot Structural Analysis (RSA) allows the interaction with other software and the use of its Calculation Engine through an Application Programming Interface (API) [5].

2.1 Methodology

A major part of the process of generating the framework is translating the geometric model to an FEM model. A distinction should be first made between the way that CAD software and FEM software understand and control geometry: when modelling a structure, a set of notions should be taken into account in order to facilitate the process of defining and analysing its performance. These notions can be defined as the structural modelling entities and include nodes, bars and panels and match real building elements as foundations, beams, slabs and so on. Some of these entities also exist in the pure geometric model, sometimes with different naming, and in fact determine a similar notion. For example, two points can describe a line and in the same way, two

connected nodes describe a structural bar. What makes a significant difference in the representation of a structural model is the need to attach structural attributes to each of its elements, while the geometric model can be purely described by its topology. In addition, the numbering of each element and its global orientation in relation to its local axis definition are crucial points in the definition of a structural model (Figure 1). A simple bar element for example, is labelled with a number and is defined by its two interconnected nodes i and j , each one numbered individually in the global system. The accurate definition of nodes is important in structural analysis since that is where all the calculations occur.

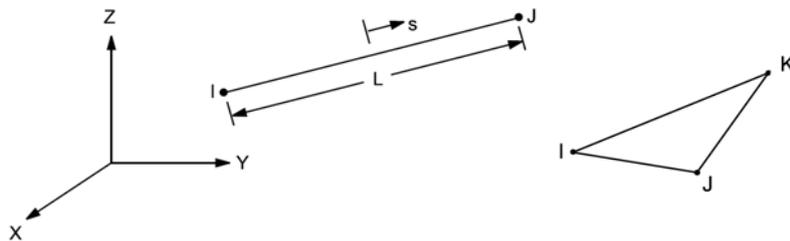


Figure 1- 3D Line element and 3-Node triangular element

Thus, each node that defines geometry in GH needs to be modified accordingly to be read by RSA. For example in order to translate a line to a structural bar, the line's *start* and *end* point coordinates are extracted from GH and are used to define two new nodes for RSA. Those nodes are then used to define a structural bar since there does not exist an implicit way of translating a geometric line to a bar. Each structural member in RSA is defined or controlled using an appropriate interface, which is included in the API library, called Robot Object Model. An interface is a software structure comprising a set of data, defined as attributes or members, and operations that can be performed which are called functions. These interfaces and their functions match the operations that a user follows to model and analyse a structure in the actual software environment. A node for example can be represented by the *RobotNode* interface, which among others includes three real numbers for x, y and z global coordinates. Each node can be managed using the *RobotNodeServer* interface, which for example includes the *Create* function for creating a new node.

3. Applications

A series of application examples are presented in order to illustrate the capabilities of the generated framework.

3.1 Three-dimensional grillage analysis

In this example a three-dimensional structural grillage is linked to a free-form surface that is modelled parametrically in Grasshopper 3D (Figure 2). By this means, any complex architectural skin can acquire a structure where its performance attributes are visualised interactively responding to the change of geometry.

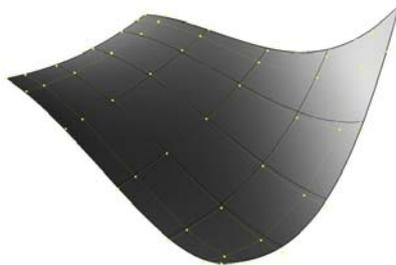


Figure 2 - Free-form surface

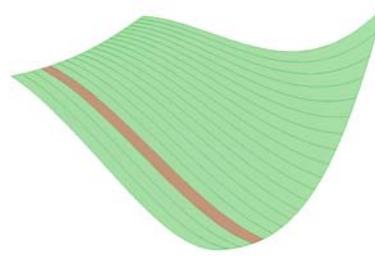


Figure 3 - Surface division in regions

In this case the parametric surface is split into regions according to the user's selection and is then transformed to structural data to be read by the analysis software (Figure 3). These regions also form the loading strips of the structure, according to the load-cases set by the user (Figure 4).

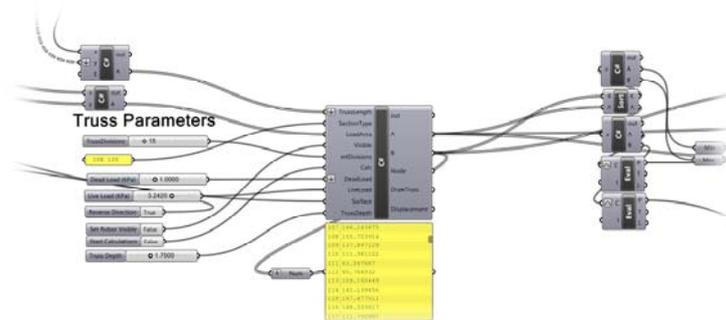


Figure 4 - User parameters in Grasshopper 3D

The edges of the generated regions are subdivided into the desired grillage spacing bays generating structural nodes. The nodes are then offset normal to the surface according to a controllable parameter which defines the height of the grillage. The set of nodes are then connected with structural bars in space, to generate the 3-dimensional grillage. The surface loading is translated to nodal loading to be applied to the structural nodes. The constrain conditions, section types and material properties are also parameters that can be defined in a generative manner by the user, either using numerical sliders or text components. It is the user's choice to compute the analytical model inside the

Structural Analysis software, in order to perform more rigorous analytical tasks (as this tools' capabilities are limited to conceptual analysis), or to visualise the current results in Grasshopper. The following images illustrate the analytical model in RSA including the point loads assigned on to structural nodes and the constrain conditions (Figure 5) and the visualisation of the stress distribution along the bars of the grillage in Grasshopper's envorment (Figure 6).

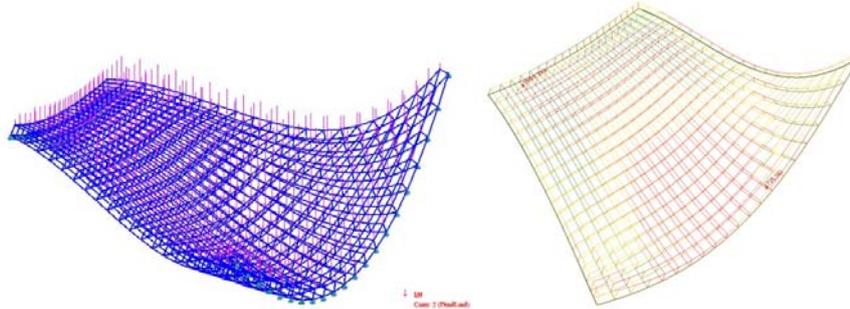


Figure 5 - Analytical model in Robot Structural Analysis **Figure 6** - Visualisation of stress ratios in Grasshopper 3d.

3.1.1 Three-dimensional grillage form-finding

The generated results can further be used to find an optimum form that can coincide with the initial surface. For clarity and in order to prove the feasibility of the proposed optimisation technique, a rectangular planar surface is used instead of the free-form surface shown in the previous paragraph. The stress results are used in an iterative process in order to balance the stress distribution in the bars. Each participating node is translated according to the stress distribution of its interconnected bars following a vector normal to the surface. The user can interactively change the initial parameters to achieve a controlled optimisation result. The optimised shape is then returned into Rhino's graphical environment for assessment or further post processing (Figures 7 and 8).

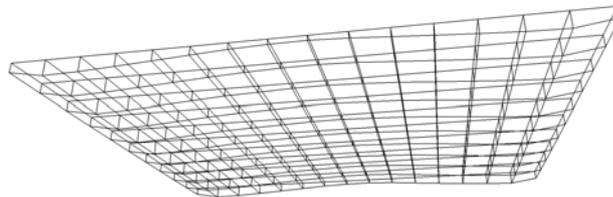


Figure 7- Optimised grillage geometry in Rhino 3D



Figure 8- Optimised section through the grillage

3.2 Free-form surface Analysis

In a similar manner as in the problem above, a free-form surface's geometric representation is linked to the structural analysis software through a framework that is able to visually present results as well as to further exploit them for the generation of efficient forms. Although the only way to structurally analyse a surface is through the use of discretized or finite elements (planar triangles or quads) it was found practical to first represent it as continuous (i.e. as NURBS) in order to freely control it in three-dimensional space. This surface is then translated to an FEM mesh by surface approximation.

The approximate surface or mesh is built in Grasshopper, consisting of a matrix of vertices, edges and faces. This information needs to be decomposed in elements that can be read by the structural analysis component. The mesh vertices are translated in a list of points in 3D space, which are then translated in structural nodes inside the analysis component. The connectivity of each face is then used to create arrays of nodes by selecting the sets that comprise each face. Finite elements can then be created by utilizing the arrays created using appropriate methods in RSA's finite element interface. Structural properties are then to be applied, such as loads, support conditions and material properties. These parameters are also treated in a way that allows them to be controlled parametrically by the user through the GH graphical interface.

After the structural information database is complete, the calculation interface is called, analysing the structure in RSA and its results are sent out to GH for each FEM element. The analysis results are mapped to a faceted surface that was created to resemble the FEM mesh in GH for visualization purposes. Each element values can be tagged accordingly or coloured using a *GH Gradient* component. This generates the interactive representation of the impact of change on the geometry or the constraints of the initial NURBS surface in Rhino's Viewport (see Figure 9).

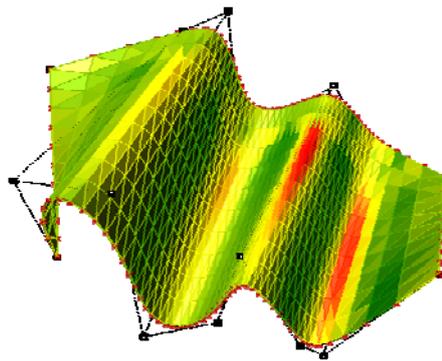


Figure 9 - Graphical representation of surface stress ratios in Grasshopper 3D

3.2.1 Principal stress grid on a free-form surface

An alternative technique for designing freeform grids is introduced in this paragraph. This is based on the directions of principal stresses that occur in a continuum shell. Principal stresses are the components of the stress tensor that occur at each point of a continuum, which are purely axial, consequently their shear component equals to zero. The directions at which these stresses occur are called Principal Stress Vectors. These components share the maximum and minimum stress values and ideally, if a grid is aligned along their directions, it can replace the continuum [6, 7, 8]. (Figure 10)

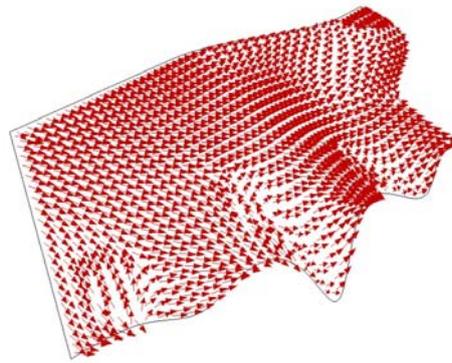


Figure 10 – Principal Stress directions

A series of algorithmic routines are developed inside Grasshopper using C# to plot the principal stress trajectories. These are based on the results obtained using the framework in section 3.2. Some of the basic notions used are described here: The stress data is first mapped on each of the planar surface generated in GH. Selecting an arbitrary face in the mesh, lines are drawn following the direction of the principal stress belonging to the current face. When a line meets a face edge, an intersection occurs which determines the next step and the preferred face at which the plotting would restart. The algorithm goes through all the available mesh faces avoiding the ones already been initialised. The finalised grid is shown in Figure 11.

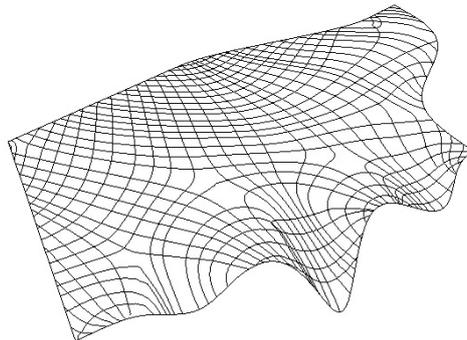


Figure 11 – Plotted principal trajectories

4. Conclusions

This paper presented the generation and application of design tools aiming to achieve a synergy between form and structure while at the same time leading to the creation of efficient structures for complex architectural shapes. This was accomplished by employing the power of parametric design combined with structural engineering software, linking them by computer programming. This combination enabled an interactive approach to structural design, a function which currently is sparingly applied for solving engineering problems. The application of the method was demonstrated through two examples: a space grillage and a surface shell. Both cases were interactively analysed responding to the designers controls having their form adapting to multiple variations of external conditions, as supporting conditions loading and material. The presented studies prove possible that optimum structural solutions need not be singular and could allow the designer to iterate between solutions that respond to their performance criteria. Furthermore, this creates the challenge for future adaptation of systems that enable integration of multi-disciplinary design teams to allow interaction of their dedicated parameters for the generation of multi-objective optimum solutions.

References

1. Grasshopper, Generative modelling for Rhino
<http://www.grasshopper3D.com/>. Accessed 25 June 2010
2. Autodesk, Autodesk Robot Structural Analysis Professional
<http://usa.autodesk.com/adsk/servlet/pc/index?siteID=123112&id=11818169>. Accessed 20 June 2010
3. Kristina Shea, Robert Aish & Marina Gourtovaia (2005) Towards integrated performance-driven generative design tools. *Automation in Construction*.14 (2): 253-264.
4. Shepherd P, Hudson R (2007) Parametric Definition of the Lansdowne Road Stadium. International Association of Shell and Spatial Structures. Venice, Italy.
5. Wikipedia, Application Programming interface. In: Wikipedia: the free encyclopaedia http://en.wikipedia.org/wiki/Application_programming_interface. Accessed 02 July 2010.
6. Kotsovos M D (2005). Reinforced Concrete. Athens: NTUA publications.
7. Winslow P (2009).Synthesis and Optimisation of Free-Form Grid Structures. Thesis (PhD). University of Cambridge.
8. Michalatos P, Kaijima,S (2007) Design in a non homogeneous and anisotropic space. Symposium of the International Association of Shell and Spatial Structures, Venice, Italy, December 3-6.