PARAMETIC FORM FINDING OF SHELLS SUBJECTED TO SEISMIC FORCE LOADING

Teodora Mihajlovska, [1]Ana Trombeva-Gavriloska,[2] Bojan Karanakov,[3] Liljana Dimevska Sofronievska[4]

Abstract: The goal of parametric design should not be the use of algorithms to design and manipulate architectural form in new freeform shapes without an underlying cause for their complexity. Although many guiding parameters can be defined and taken into consideration, this research focuses on strategies that help us shape forms by using forces, more specifically seismic loading. It explores different approaches of implementing these parameters in the Rhinoceros modeling software through the Grasshopper virtual programing environment that enable us to simulate the behavior of shells subjected to seismic force loading. The implementation of such a method in the specific software package would allow architects, at an early stage of design, to define a form that has improved performance under the action of seismic force

Key words: parametric design, form-finding, seismic force

1. INTRODUCTION

When considering horizontal seismic forces, the geometry of shells plays a key role in their behavior, primarily due to their light mass and great geometric stiffness. Nevertheless, seismic loading is rarely taken into account in the initial design of shells, although it could significantly improve material efficiency and seismic performance. In recent research, methodologies have been developed that generate forms for shells designed to carry their own weight and seismic load through 3D models of suspended networks of cables obtained by implementing a dynamic relaxation method [1]. By applying this approach, the geometry of a single-layered shell with variable thickness or the geometry of two-layer interconnected thinner and lighter shells is defined.

The aim of this research is the implementation of the method in a software that allows great flexibility in the creation of 3D models and is used in the early stages of the design during the genesis of the very form of architecture. Rhinoceros3D [2], through its Grasshopper virtual programming environment, is used mainly for parametric design. In this paper, the application of the method in the Rhinoceros 3D software package is presented. Through implementation in the virtual programming environment Grasshopper and the programming language Python, an algorithm is built, which, for a given layout shape, defined boundary conditions and seismic force intensity, generates a double-layer shell. Dynamic relaxation is performed using Kangaroo [3], which is an integrated plug-in in Grasshopper, after which the proposed algorithm from the obtained shells of different load cases defines the envelope of the double shell.

The algorithm is a visual algorithm developed through Grasshopper, based on the principle of components and connections between them. Grasshopper components are preprogrammed, ready-made segments of code that perform specific tasks. Such components connect with each other to build an algorithm that performs the desired task. However, not all necessary functions are available as ready-made components, certain specific functions must be written as classical code, through a programming language. This code is entered as a new component that builds on the visual algorithm.

Developing methods within this framework means that the form that this research examines in terms of its bearing characteristics, through the same 3D model, can be further considered in terms of various parameters that are of interest to the designer.

[1]Teaching assistant, M.Arch, Faculty of Architecture, "Ss. Cyril and Methodius" University, mihajlovska.teodora@arh.ukim.edu.mk

[2]Professor, Ph.D., Faculty of Architecture, "Ss. Cyril and Methodius" University, agavriloska@arh.ukim.edu.mk

[3]Professor, Ph.D., Faculty of Architecture, "Ss. Cyril and Methodius" University, karanakov.bojan@arh.ukim.edu.mk

[4]Teaching assistant, M.Arch, Faculty of Architecture, "Ss. Cyril and Methodius" University, dimevska.liljana@arh.ukim.edu.mk

2. OBTAINING INDIVIDUAL SHELLS BY DYNAMIC RELAXATION METHOD (VISUAL ALGORITHM)

In order to obtain the envelope of a double shell, first the individual shells of different load cases are defined. In this research, 5 different load cases were taken into consideration: a load case representing gravity loading without any horizontal acceleration and 4 combinations of gravity and horizontal seismic acceleration in both orthogonal directions, in the positive and negative directions with an acceleration intensity of 0.45g.

The algorithm used by Kangaroo to perform the operation is a form of dynamic relaxation, a numerical method based on solving a set of nonlinear equations. Dynamic relaxation follows the movement of the structure over time at a given loading and just like the Verlet method [4], it is used to integrate Newton's second law over time [5].

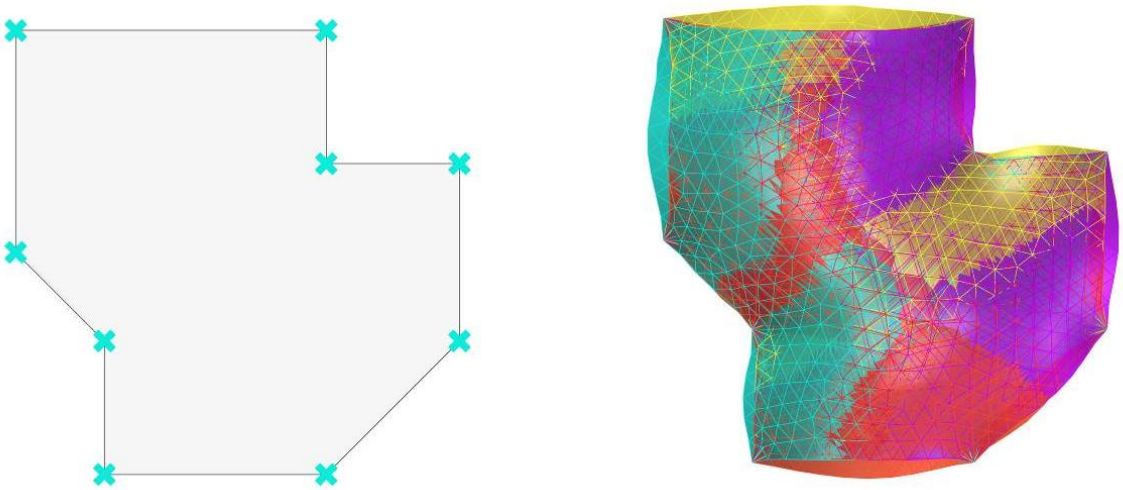

Figure 1– Overlapping of the shells obtained by modeling with different loading cases

By combining all the forces acting on each point, the equilibrium is determined by the constant displacement of all points through several iterations until the forces are balanced and the movement stops, that is, until the structure stops in static equilibrium. This happens in a dynamic way so that the motion will oscillate around the equilibrium and damping is used to remove the energy and ensure convergence. In typical engineering applications of dynamic relaxation, the desired output is a static equilibrium configuration for given boundary conditions.

In this implementation, the shell surface is represented digitally by discretizing the surface into a network of linear elastic springs connected at nodes with free rotation. Gravity and earthquake accelerations (a) are applied to each node, generating loads. The discretized model is given to the Kangaroo solver which models the shells by way of dynamic relaxation. The resulting model behaves like a cable net that stands in pure compression under a combined loading of gravity and horizontal acceleration, as it is the inverse of a net that hangs in pure tension under such loading.

The process of modeling the shell is entirely performed in the Grasshopper virtual environment and is an algorithm created through the visual programming language of the same name. The shells obtained in this way represent input information for the algorithm which generates the envelope of the double layered shell as output information. Considering that the specific functions needed to generate the envelope are not available in Grasshopper, a classic algorithm was programmed in the Iron Python programming language (version 2.7.3) [6]. Iron Python is integrated into Grasshopper and directly builds on its visual algorithms, thus enabling the formation of a continuous process.

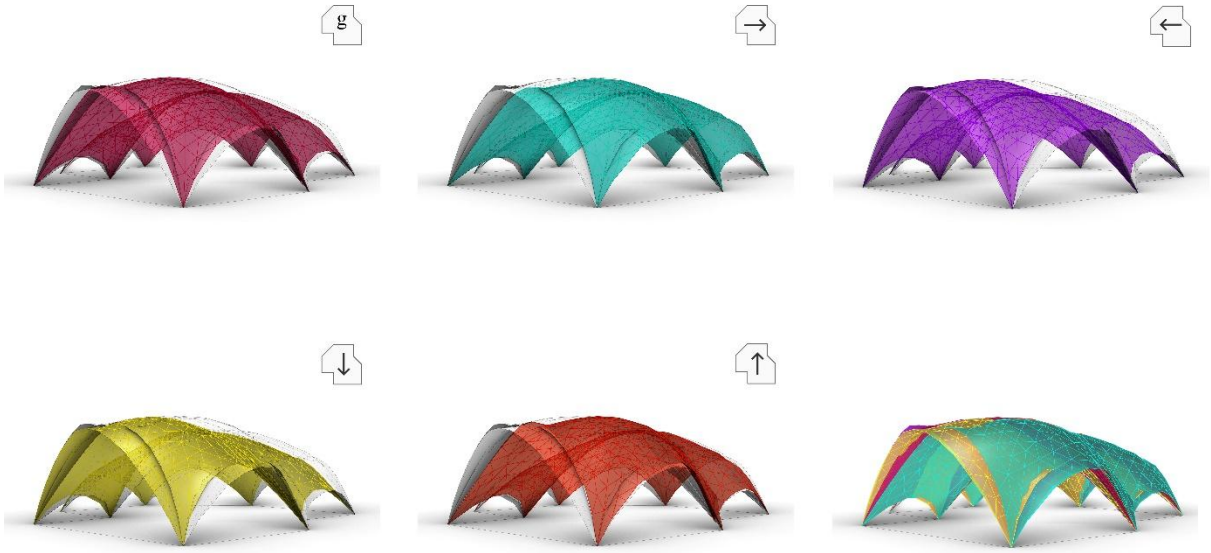

Figure 2– Superimposed funicular nets that are obtained through dynamic relaxation for earthquake loading in different directions.

3. PYTHON COMPONENT FOR SELECTION OF HIGHEST AND LOWEST POINTS (CLASSICAL ALGORITHM)

The functions of the algorithm described so far are defined through a visual language, without classical writing of code, because so far the necessary operations can be done by already existing Grasshopper components. However, the grouping of points according to their xy coordinates and the selection of the local highest and lowest point according to the z coordinate are not available as a ready-made component in the virtual environment and must be scripted as a new native component. Grasshopper supports the use of several programming languages: Python, C# and VB.Net. For the purposes of this paper, it has been decided that the code will be written in the Python programming language.

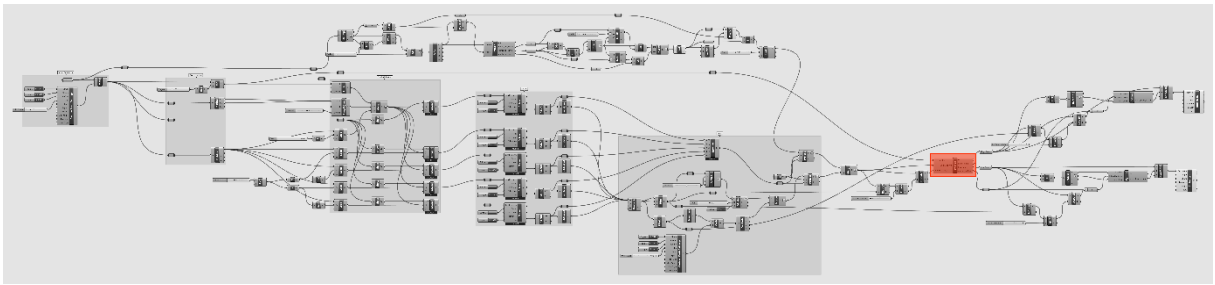

Figure 3 – The visual algorithm built from available components (the new component programmed through Python is highlighted in red)

In the new Python component, three lists of points are defined as input data, a list of points obtained by projection of the points within the union of the projections (union_points); a list of points that belong to the intersection of the projections (intersection_points) and a list of points that are stationary during the simulation – pinned supports (anchor_points). Three lists of points are obtained as output data, which define the upper and lower layers of the shell, i.e. the upper and lower limits of the envelope, and the stationary points - the supported - are separated again as separate ones.

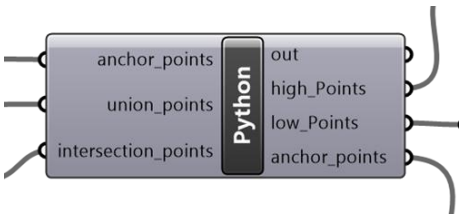

Figure 4– Thecomponent scriptedin Python, which performs the selection of the highest and lowest points

4. REBUILDING THE MESH ENVELOPE

From the three lists obtained by the Python component in the next step, two separate meshes are formed again, one represents the upper limit of the envelope (created by the points with the highest coordinates), the other represents the lower limit of the envelope (defined by the points with the lowest coordinates). However, in this step, the numerical algorithm from Python provides as output a list of points, but not their configuration in mesh faces. In mathematics and computational geometry, one of the most commonly used methods for defining meshes is the c [7]. A Delaunay triangulation for a given set P of discrete points in general position is a triangulation DT(P) such that no point in P is inside the circumscribed circle of any triangle in DT(P) (Fig. 5). Delaunay triangulation maximizes the minimum of all the angles of the triangles it forms; and tends to avoid the so-called "thin" triangles.

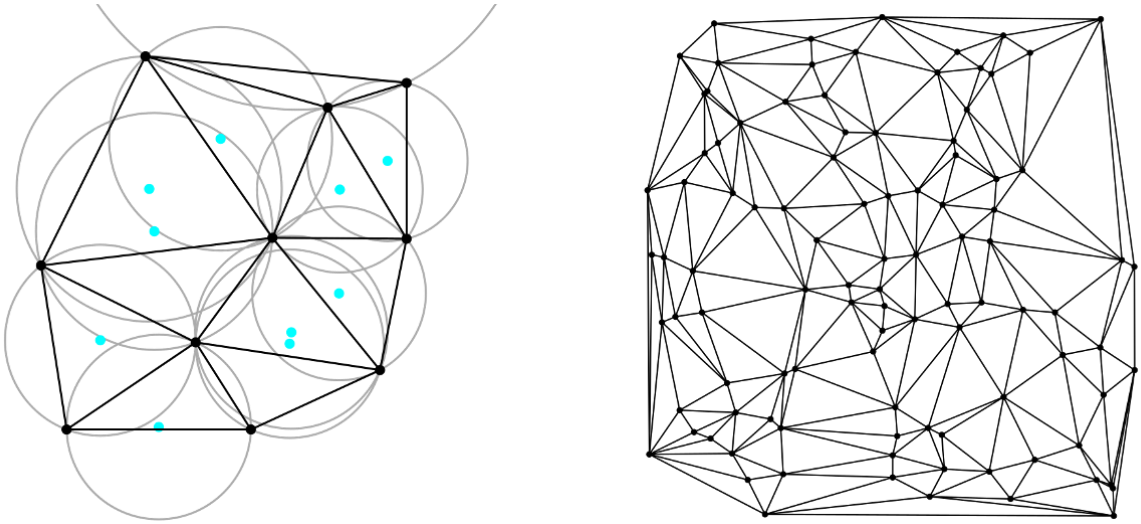

Figure 5 – Delaunay triangulation

Delaunay triangulation manages to successfully form the network of coordinates, however, since the tendency is to connect as many points as possible, it defines faces outside the defined boundaries of the envelopes themselves (Fig. 6). Such unwanted connections are formed when triangulation establishes connections between points belonging to predefined envelope boundaries.

To overcome this problem, the list of mesh faces was refined using the criterion: if three of the vertices forming the triangle (the mesh face) are part of the boundary of the envelope, the mesh face is deleted.This criterion for eliminating faces from the mesh successfully removes unwanted connections from its definition, however, in certain cases, it may result in the deletion of faces from the mesh that are part of the envelope itself (Fig. 6).

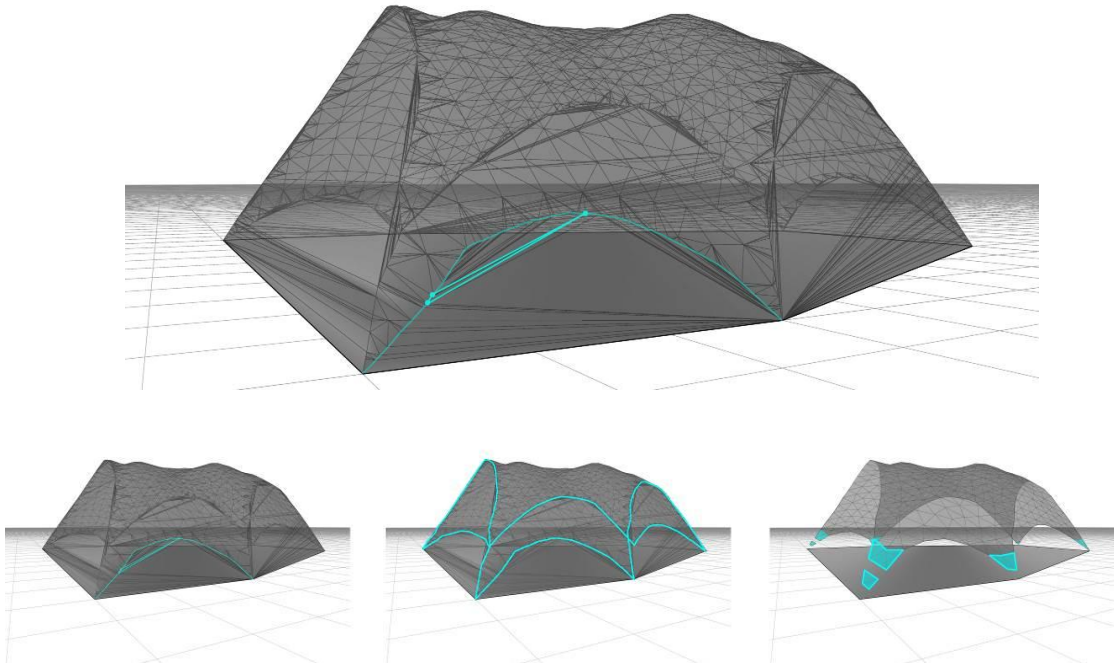

Figure 6 – Mesh faces outsife the envelope border

This problem usually occurs at the lower limit of the envelope near the support. In this segment, the curves that define the border converge to the support and as a result, a smaller number of reference points are obtained, which can also result in a case where the only points are those of the borders. In order to define additional reference points even when defining new entry points through the projection of points on the shells, the list of reference points is supplemented with points from the bisectors of the angles formed at the support (Fig. 7).

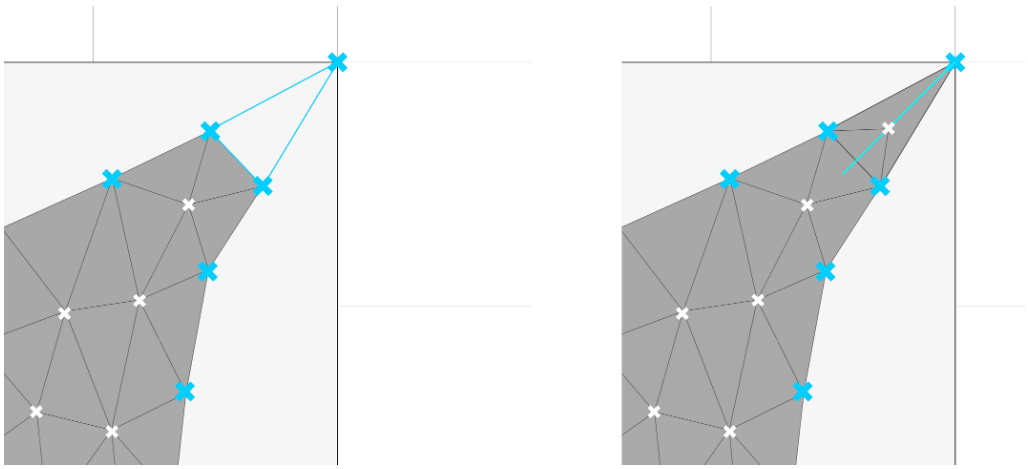

Figure 7 – Defining additional reference points in order to avoid unwanted break of the shell geometry

5. CONCLUSION

The biggest benefit of implementing dynamic relaxation, as a method of creating the form, in the early stages of the design can significantly improve the final result of the form in terms of load-bearing characteristics. Such algorithms are primarily intended for architects and planners and reduce the time

required for the development of the project, enabling an efficient process through which a variety of possible solutions of the form are quickly and simply considered, which always take into account the given form of the base, the defined boundaries conditions and intensity of seismic force. The added value of the forms obtained in this way, in contrast to the arbitrarily designed ones, represents a much better basis for further development and constructive analysis, which means fewer changes in the design process and better control from the beginning.

Defining the envelope is a function that is built into different software packages, but for the set purposes it is necessary to implement the method in the appropriate software, so its entry into Rhinoceros3D and Grasshopper makes it available as a tool for projecting the shape even when defining the very concept for the project, and by applying the specific software, analyzes from the aspect of aseismic design are possible. In addition, working in parametric software allows the generated forms to be immediately considered from other aspects in relation to the program requirements, the environment, the lighting, so that the optimal form can simply be adopted.

Future research should be aimed at refining the proposed algorithm and further testing it on different base shapes, geometric irregularities in the base, and defining the support points in terms of how they affect the shape.

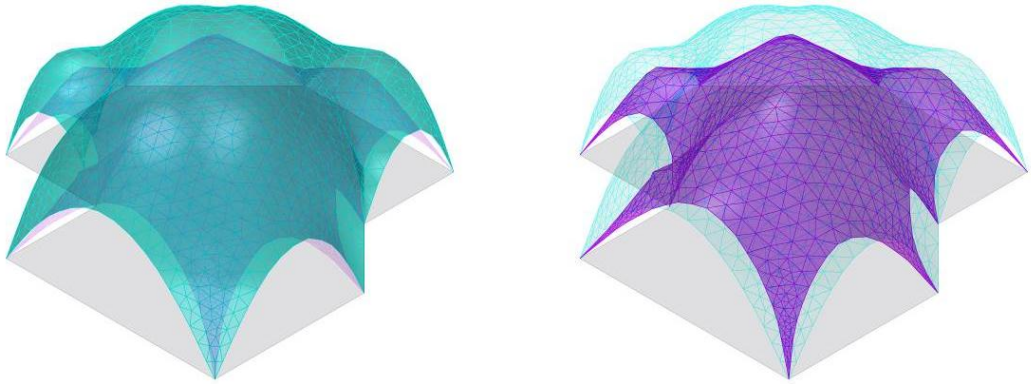

Figure 8– Top and bottom layer of shells obtained using the presented algorithm

6. REFERNCES

[1] Michiels T. L. (2018). Form finding of arches and shell structures subjected to seismic loading. PhD thesis. Princeton University, New Haven, USA.

[2] McNeel, R., & others. (2010). Rhinoceros 3D, Version 6.0. Robert McNeel & Associates, Seattle, WA.

[3] Piker D. (2013). Kangaroo: form finding with computational physics. Architect Des 83(2), pp.136–137.

[4] Verlet L. (1967). Computer experiments on classical fluids, I. Thermodynamical properties of Lennard-Jones molecules. American Physical Society vol.159.

[5] Adriaenssens S., Block P., Veenendaal D. and Williams C. (2014). Shell Structures for Architecture: Form Finding and Optimization. New York, USA.

[6] Harris, A. (2010). Scripting via IronPython. In: Pro ASP.NET 4 CMS. Apress. https://doi.org/10.1007/978-1-4302-2713-7_7

[7] Delaunay, B. (1934) ‘Sur la sphère vide.’, Bulletin de l’Académie des Sciences de l’URSS. VII. Série, 1934(6), pp. 793–800.