HyparActive: A Design Exploration Tool for Multi-segment HyPar Shells

Winston DAVIS 1, Tyler S. SPRAGUE 2, Brian R JOHNSON 3

1 MS Arch, Design Machine Group
   University of Washington
   jwd09@u.washington.edu

2 Assistant Professor, Design Machine Group
   University of Washington
   208 Gould Hall, Box 355720
   Seattle, WA 98195
   tyler2@u.washington.edu

3 Associate Professor, Design Machine Group
   University of Washington
   208 Gould Hall, Box 355720
   Seattle, WA 98195
   brj@u.washington.edu

Abstract
While shell structures offer efficient solutions to many contemporary design problems, designer familiarity with the discipline has declined in recent years. Interest in shells is growing again due to modern fabrication techniques, but is hampered by the difficulty of balancing form manipulation and structural analysis concerns. In this paper we describe HyparActive, a software tool developed specifically to help designers explore conceptual shell forms. HyparActive provides simple 3D form-manipulation tools and visual FEA feedback in the same digital model. The tool allows users to define, manipulate and visualize a parametric form consisting of multiple hyperbolic paraboloid (hypar) shell segments. The quick feedback means the designer can interactively explore how form and stress distributions interact, informing design decisions and enhancing their design "intuition." This research project investigates the challenges of developing an interactive modeling tool and the benefits of including performance analysis in the conceptual design phase.

Keywords: conceptual design, concrete shells, education, form finding, hyperbolic paraboloid, morphology, parametric design, performance analysis, prefabrication, simulation

1. Introduction
Shell structures have been developed for over a hundred years, providing economical solutions to long-span design problems. While the period of wide-spread shell popularity and development ended in the 1970’s, development of the design and construction technology has continued. A research interest in shells has been brought about by innovations in fabrication technology coupled with the availability of algorithmic design environments that allow designers to produce highly-efficient forms. The apprehension that initially ended the wide-spread use of shells, identified by Meyer and Sheer [1] to be predominately high construction costs and limited formal expressivity, has been alleviated by developments in design and fabrication technology that can produce previously-impossible structural forms (Meyer and Sheer [1]).
The caveat is that taking advantage of these processes is difficult for the typical architect developing a concept for a new project. These digital workflows are heavily front-loaded with complex programming and debugging. This may intimidate and deter some architects, who traditionally have minimal education in computer programming. Simultaneously managing formal exploration and performance analysis feedback can be both obscure and meticulous, which certainly limits the success of any concept model. HyparActive was developed as a solution: a tool that allows designers to freely explore and design shell forms all while getting a good idea of how well the current solution would perform.

2. HyparActive

Prior to development, some important objectives were identified to maximize the usefulness of the tool. The tool should offer intuitive inputs and controls for changing variables, such as shape and material. It is important for the user to be able to create interesting forms with a high level of control and flexibility. On the same note, the tool would need to communicate performance feedback in an easily-understood format, so the user fully understands the consequences of their design decisions. Just as important as knowing exactly what is being changed is understanding that change’s consequences.

Early in development, after experimenting with various shell geometries, the decision was made to generate exclusively hypar forms. The hypar shows the most promise geometrically by affording a wide range of shapes with minimal parameters all while maintaining its anticlastic form, and therefore its unique structural behaviors. Peerdeman [2] offers a concise summary of why hypars are desirable forms for shells: “The success of the form rests for the architect in its appealing [geometry], for the structural engineer in its simple structural analysis, and for the contractor in its economical formwork…” (Peerdeman [2]).

2.1. Control Schemes

Many geometric explorations were conducted to find the most user-friendly method to generate a multi-segment hypar form. The first exercise in generating single hypar forms was to “project” a quadrilateral “onto” a hyperbolic paraboloid. The process involves extracting a grid of points from the initial quad surface, and calculating new Z coordinates (heights) for each point by using their X and Y coordinates as parameters for the hypar formula. This produces a new grid of points that is used to generate a best-fit surface in the form of a trimmed hypar.
Using this algorithm, the hypar geometry can be controlled with 6 parameters. This control scheme maximizes the range of forms available to the user by letting them control every attribute of the hypar. The user can individually change the degree of curvature in each direction of the hypar, as well as the “rotation” of the curvature relative to the quad. Coupled with the capacity to shift the “balance” of the quad to one side of the hypar, a wide range of asymmetrical geometries can be achieved.

![Figure 2: Various forms generated with the six-parameter control scheme](image)

While offering a high degree of expressivity, the six-parameter control scheme had some unavoidable flaws. Some redundancy in control was caused by having two different parameters affect the height of the form. Using these two parameters that affect the same attribute caused some complication in achieving a desired form. Unfortunately, only one of the parameters did not offer sufficient control of height without the other parameter. A second drawback was the level of abstraction between design intent and action. Even design constraints simple in conception proved challenging to realize with this control scheme. Simply keeping a specific corner in contact with the base plane became an exercise in data management. These hindrances abstracted the design intent and could deceive the user, who either might not have been making the changes they believe they were, or couldn’t figure out how to make the changes they wanted. While not particularly bad for the exploration of conceptual geometry, more parametric transparency is required when designing built structures with specific formal requirements. When generating multiple hypars, each of them requiring six individual input values, the control scheme became very cumbersome.

While exploring an algorithm for generating a multi-segment hypar form, a much simpler control scheme was found. In this scheme, a single quadrilateral polygon was used to generate four points that were moved vertically by some user-controlled value, and then used as parameters to create a best-fit surface from these corner points. The resulting surfaces were validated as hypars by checking their isocurves for linearity. This exercise provided some useful insight into generating hypars: the parameters for each segment could be simplified down to four points. This method of generating four points and creating a surface between them is much simpler than the projection method, but some loss of geometric control is observed: this method always generates a standard, untrimmed hypar. However, his method proved helpful in generating a cohesive multi-segment form as each pair of adjacent hypars could share two corner points. This greatly reduced the number of input parameters the user is responsible for controlling.

The polygon used to generate the hypar forms evolved into the “footprint mesh”. This mesh is configured to be the flat footprint of the shell form. The user picks the size, configuration and number of segments to generate, and the corner points of each 4-sided segment are moved vertically and used to generate a hypar section. A grid configuration and a radial configuration were developed to give the user more form options.
Figure 3: A radially-configured multi-segment form above the footprint mesh

The simplicity of these footprint configurations is a product of keeping the control scheme simple. Some explorations into allowing for more freedom in footprint configuration produces interesting results, but convoluted the interface with excessive parameters. Also, when using the four-point method of generating the hypars, the footprint mesh needs to be comprised of four-sided segments. If a robust control scheme could be developed that allowed for generation of trimmed hypars, i.e. using the projection method, it would allow for a less-constrained footprint mesh and allow for more complex forms.

Compromises were made to simplify the control scheme that limited formal flexibility. This was necessary to allow the user to design multi-segment forms with a manageable control scheme. Finding the right balance between the level of control and simplicity of the control scheme proved to be very difficult. From an architectural design standpoint, the control scheme leaves more formal flexibility to be desired. However, the simple scheme makes the tool easier to use and still allows the user to explore a wide range of asymmetrical forms. The most worthwhile development that served both formal freedom and simplicity was getting rid of the controls for the height of each point all together and making the points themselves directly manipulable. This eliminated the need for the user to translate their design intent into a set of height values and allowed them to place each point in 3D space as they saw fit. This development also allows for easy lateral translation of the points, which was not available previously (without the use of two more inputs per point).

Figure 4: The six-parameter control scheme (left) and the direct manipulation control scheme (right)
2.2. Structural Simulation

A structural simulation module was used to analyze the form and provide the resulting material stresses in the model. When the simulation is activated, the shell geometry is first converted into the required format and in the process is subdivided into smaller elements. The density of these elements controls the precision of the analysis, so the value was parameterized so that a quasi-optimized value can be found. This value, referred to as the “mesh sensitivity” value, controls the relationship between the speed and the accuracy of the analysis, and since both are valued, a balance is necessary.

Figure 5: The user’s view of the form with the structural analysis feedback

To find the most appropriate mesh sensitivity value, three different forms were generated and analyzed using mesh sensitivity values that ranged from the lowest available to high values that required more time to process than was deemed acceptable (upwards of one minute between moving a point and seeing the model update). The five highest values of each principal stress were recorded to identify an asymptotic curve that would provide a benchmark value to gauge the accuracy of the individual sensitivity settings. The time it took to calculate the solution was also measured for each iteration to help identify the most useful range. After comparing the results of all three forms, a value range was found that provides the best balance of accuracy and speed.
Best Time Range

Figure 6: Analysis of the mesh sensitivity results for one form. The graphs were used to cross-check appropriate compute times with stress values’ presumed accuracy.
After finding the appropriate mesh sensitivity value, some validation models were used to make sure the analysis tool was configured properly. Two forms of Felix Candela’s design were chosen to analyze for this test as similar analysis results from a study at Princeton University were available for comparison. The analysis returned the largest principal tension and compressive stresses in each form, affording a total of four values to compare. The resulting sets of values are proportional and considerably close which provides a level of validity to the analysis configuration of the tool.

2.3. Transient Data Spikes
Some data spikes are seen in the stress values when compared to mesh sensitivity. While localized, the value spikes are unpredictable and could possibly blur the users understanding of the form’s structural behavior. While investigating the cause of the spikes, it was observed that when a far-outlying value was returned, the surface area of the element producing that value was irregular and far smaller than the average element size. In one instance, the average element size was .85 square feet and the element responsible for the outlying value was .0003 square feet. It was also observed that these “problem elements” were consistently positioned at the bottom of the form, supporting the theory that the current method of trimming the form mesh with the ground plane is producing miniscule elements that produce highly inaccurate values.
Different solutions for this issue range from simply sensing that if problem elements exist in the model to preventing them altogether. Currently, the user is made aware if any problem elements currently exist in the model. The sham stress values provided by these elements are culled from the list of values displayed on screen. These elements are also rendered differently in the viewport to make the user aware of their presence and location in the model. At this point, the user is advised to try locally altering the form where the problem element is located. Minor adjustments are usually all that is needed to return a more uniformly-subdivided mesh after splitting it with the ground plane. This temporary solution consistently solves the value spikes while a system to prevent them is developed.

There are many opportunities for improving the effectiveness of the structural simulation. As observed, a workflow for optimizing the input mesh for analysis would yield fewer glitches in the feedback data. Using multiple meshes could also reduce analysis faults by allowing the meshes to be checked against each other to identify and cull unreliable data. Another possibility would be to pre-record the feedback data and store it in a database for future use. This would make it possible to preview the data and cull any unreliable values beforehand, but also make the data immediately available for visualization instead of having to be recomputed after every change. This would afford an instantaneous feedback loop, providing a superlative user experience. This separation of the actual simulation from the user’s experience could provide a more effective learning tool by maximizing feedback speed and reducing the range of exploration.

3. Conclusion

The acceptable standards for the mesh sensitivity and validation models are relatively low-fidelity, but that is all this tool requires. Since it is meant to help develop sound conceptual forms and not provide high-precision analyses, the user only needs to get an idea of how their most recent design move affected the solution’s performance. The timeframe in which the user gets this information is the strength of HyparActive.

To build experience and design knowledge, the user needs to see if their design decisions play out how they expect. Before any sort of digital performance analysis tools were available, the only way to get this information was to see the project through to completion and only then be able to judge its performance. The introduction of these tools shortened the time it took the user to get this information by offering them a very good guess of how the project would perform.

As these analysis tools become more available and easy to use, they can be implemented very early in the design process. In terms of thin shell construction, no longer does the architect need a personal mastery of geometry or to stick to symmetrical forms and rules of thumb to start with a concept that has some proof of viability. Furthermore, compacting the feedback loop to instantaneous allows the user to build intuition and “get a feel” for how the structures work.
3.1. Further Opportunities

One feature that a tool like HyparActive would benefit from is a complementary fabrication workflow. Using the form output from the exploration tool, this additional workflow could generate formwork or rationalize the form into components to be prefabricated and assembled. This could also be useful during the form exploration, and allow the user to explore how different materials and methods of subdivision and construction affect the performance just as the geometry does. Recent trends show that prefabrication of components is becoming the routine strategy for erecting shell structures, and incorporating this functionality into the design process will allow it to be more efficiently managed. This additional workflow will be necessary to efficiently construct the complex, asymmetrical forms generated with tools like HyparActive.

As computational analysis tools develop and become easier to use, their effect on the design process grows more profound. Their inclusion earlier in the process and ability to provide increasingly accurate results makes them incredibly valuable to architects. By compacting the design feedback loop, not only is the user afforded the ability to develop sound concepts without the need for extensive expertise, but also the ability to gain this expertise. It is hoped that more learning tools are developed that make it easier for complex forms and metrics to be explored and that afford designers an increasing number of efficient solutions to investigate.
Acknowledgements
The authors would like to acknowledge the Department of Architecture at the University of Washington and the Precast Concrete Institute Foundation for their support of this research.

References