Algorithmic Design and Analysis Fusing Disciplines

Rita Aguiar
Instituto Superior Técnico/
INESC-ID

Carmo Cardoso
Instituto Superior Técnico/
INESC-ID

António Leitão
Instituto Superior Técnico/
INESC-ID

1

1 Representative image of a tridimensional truss.

ABSTRACT

In the past, there has been a rapid evolution in computational tools to represent and analyze architectural designs. Analysis tools can be used in all stages of the design process, but they are often only used in the final stages, where it might be too late to impact the design. This is due to the considerable time and effort typically needed to produce the analytical models required by the analysis tools. A possible solution would be to convert the digital architectural models into analytical ones, but unfortunately, this often results in errors and frequently the analytical models need to be built almost from scratch. These issues discourage architects from doing a performance-oriented exploration of their designs in the early stages of a project.

To overcome these issues, we propose Algorithmic Design and Analysis, a method for analysis that is based on adapting and extending an algorithmic-based design representation so that the modeling operations can generate the elements of the analytical model containing solely the information required by the analysis tool. Using this method, the same algorithm that produces the digital architectural model can also automatically generate analytical models for different types of analysis. Using the proposed method, there is no information loss and architects do not need additional work to perform the analysis. This encourages architects to explore several design alternatives while taking into account the design's performance. Moreover, when architects know the set of design variations they wish to analyze beforehand, they can easily automate the analysis process.

INTRODUCTION

The spread and popularity of algorithmic tools has changed the way architects design (Kalay 2004). These tools allow architects to generate model variations in an almost effortless fashion and to obtain more elaborate and innovative shapes (Yusuf 2011). These shapes represent new challenges in terms of fabrication, construction, and performance evaluation (Kolarevic 2001).

When done manually, performance evaluations can be a time-consuming task, particularly for the complex designs that emerge from algorithmic approaches. Fortunately, current performance-analysis tools can replace the extensive error-prone manual calculations, allowing a faster evaluation of a design's performance (e.g., within its thermal, lighting, structural, and other elements). As a result, several architecture firms have taken advantage of them to explore design iterations and select the best-performing ones (Burry and Burry 2010).

Typically, performance analysis requires analytical models that only preserve the aspects of the original model that are relevant for the intended analysis; for example, surfaces for radiation analysis, or node-bar connections for structural analysis. Unfortunately, these specialized analytical models might not be trivial to produce and they might require extensive manual work, thus making it difficult to evaluate several design iterations. In this paper, we present a solution to this problem: we propose to extend algorithmic design to include automatic generation of analytical models and further processing of the performance evaluation, thus speeding up the evaluation of both complex designs and multiple design iterations.

RELATED WORK

In the traditional design process, the initial stages focus on aesthetics and functional aspects. It is only during latter stages of design that other performance factors are analyzed. This happens because it takes a lot of time and effort to produce the required analytical model (Turrin, von Buelow and Stouffs 2011), even when taking advantage of existing digital models. Often, exporting a 3D model to a different format may result in errors (Moon et al. 2011), especially if it contains complex geometry. There are also cases where the analysis tool requires a simplification of the model (e.g., transforming curved surfaces into arrangements of planar ones, which might not be a rigorous translation). As a last resort, it might be necessary to fully rebuild the model according to the requirements of the analysis tool.

There are some tools that already overcome some of these interoperability issues by exporting the 3D model into a format that is understood by the structural analysis tool, such as Industry Foundation Classes (IFC). However, sometimes information does not have a corresponding definition or is missing (Wan, Chen, and Tiong 2004). Having to input this information can still consume a great amount of time if done for all variations of a building. Another setback is that when doing design changes, the architect needs to repeatedly convert the 3D model to be evaluated by the analysis tool.

A different approach to deal with this issue is to integrate the modeling and analysis tools. This is the case with Grasshopper, which is a parametric design tool that, directly or via plug-ins, combines parametric 3D modeling with different kinds of analysis.

Ladybug and Honeybee are Grasshopper plug-ins that provide a series of analyses related to lighting, radiation, and thermal efficiency (Roudsari, Pak, and Smith 2013). When introducing design changes, these tools provide feedback on how these choices affect the building's performance, which encourages architects to integrate these tools in their design workflow. However, one of the biggest setbacks of these tools is that for very complex models, the analysis becomes too time-consuming, which severely hinders the highly interactive nature of Grasshopper.

Karamba is a plug-in for Grasshopper, which is an efficient tool to evaluate the structural performance of a design (Preisinger and Heimrath 2014). However, this tool has some limitations and lacks the ability to extract detailed information (Wallin and Wasberg 2016).

Geometry Gym is another plug-in for Grasshopper that makes it possible to export a Karamba model to Autodesk Robot Structural Analysis software (Mirtschin 2011). GH2Robot is an alternative to Karamba and Geometry Gym that enables a direct link between Rhino and Robot (Christensen, Parigi and Kirkegaard 2014). Unfortunately, both Geometry Gym and GH2Robot lack the ability to retrieve and display the results from Robot back into the 3D modelling software.

In order to solve these limitations, it would be advantageous to have an isomorphism between the parametric and analytical models. Ideally, the analytical models should coexist simultaneously with the parametric model, so that they are immediately usable by the corresponding analysis tool without requiring additional work from the architect. By facilitating this process, we are encouraging a performance-oriented design exploration between the parametric and analysis tools.

ALGORITHMIC DESIGN AND ANALYSIS

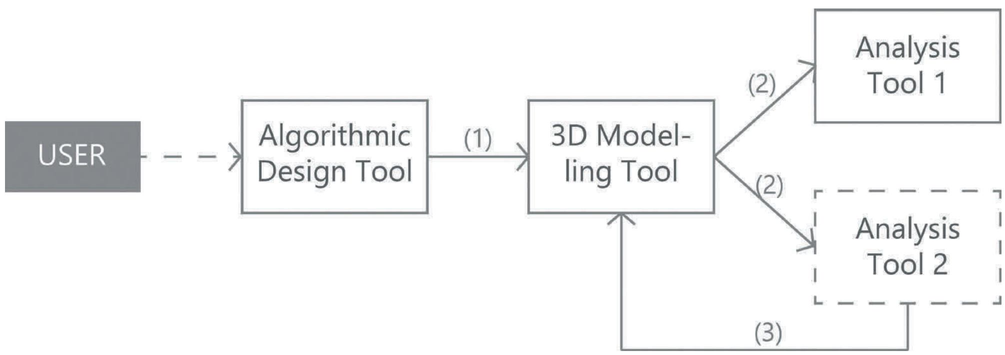
In this paper, we propose an analysis workflow named Algorithmic Design and Analysis (ADA). In this section, we explain the theoretical basis for the workflow and compare it with a typical one. We will also discuss the automation process and the visualization options we provide.

Analysis Workflow

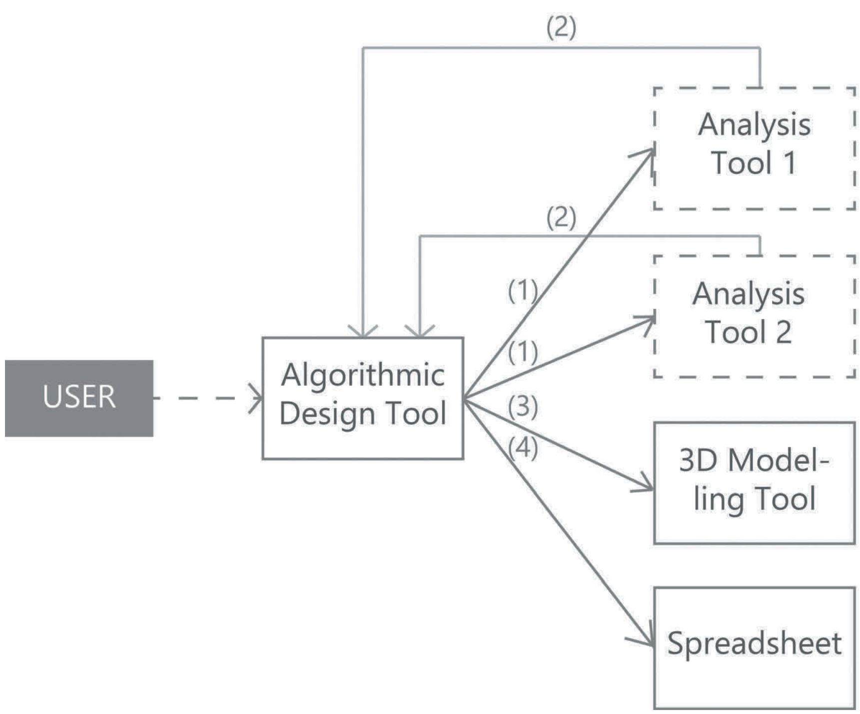
The proposed workflow assumes that the designer is working with an algorithmic design tool where they implement a parametric design. Figure 2 describes the typical workflow of analysis processes (e.g., in Dynamo/Revit/Robot or Grasshopper/Rhino/DIVA).

Taking into account the problems described in section 2, particularly the loss of information that tends to occur in a long pipeline of tools, we propose a different analysis workflow whereby information is relayed directly from the algorithmic design tool to the analysis tool. Moreover, we generate the information in the required format and only containing the required details. Following completion of the analysis, the 3D modeling tools display the results. Figure 3 shows the workflow that we propose for ADA.

We call this process Algorithmic Design and Analysis because it generates the analytical model algorithmically according to the requirements of the analysis. Moreover, it allows the designer to change the analysis features directly on the algorithmic description of the design, thus supporting the automation of multiple different analyses. The objective of this approach is not only to achieve more reliable results, but also to make it easier for designers to perform more analyses.



2 Typical analysis workflow: (1) the algorithmic design tool generates the model in a 3D modeling tool, (2) the model is exported to the analysis tools where the architect can visualize the results or, (3) the results are retrieved for further processing.



3 ADA workflow: (1) the algorithmic design tool generates and sends the analytical model directly to the analysis tools, (2) retrieves the results, (3) and uses them to generate a 3D model in the modeling tool. It is also possible to (4) present the results in other formats, e.g., as numeric data in a spreadsheet.

Automated Analysis

Using ADA we can automate the analysis process, allowing for the evaluation of a range of design variations belonging to the design space set by the architect. To this end, the architect selects the relevant parameters of the design, which can range over a numerical interval or a set. Considering that an analysis process may be very time-consuming, it is not realistic to assume that we have enough time to analyze all the variations in a large interval. In this case, it might be necessary to analyze just a sample of the design space. There are several different approaches to do this: (1) we can select equally spaced parameter values in an interval, or (2) we can randomly choose parameter values within their interval, producing random design variations. In either case, for each parameter value, we algorithmically generate and evaluate the corresponding analytic model, and we save the results for further processing (e.g., to graphically present them to the designer for comparison). Another strategy is to use the Monte Carlo method, which iteratively generates a random design, analyzes it, and keeps the best one found to date. Whatever the strategy, it might still be necessary to limit the number of iterations or the computation time.

In the next section, we compare the typical analysis workflow with that of ADA. We also discuss automating analysis when a variable ranges over a small set or when it ranges over a large interval.

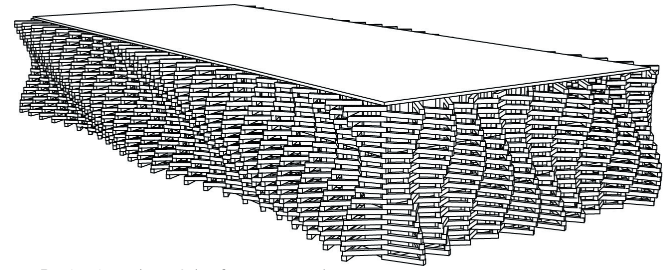
EVALUATION

Case Studies

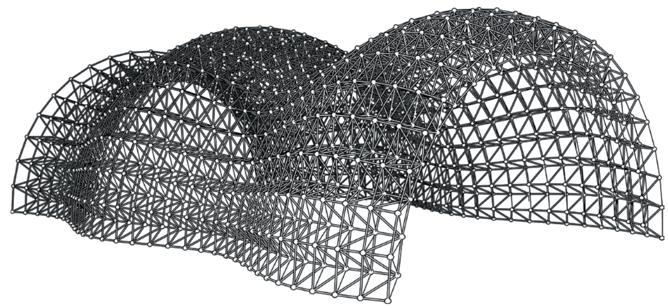
The first case study is a façade design for an office building in Shanghai (Figure 4). It is a modular façade, where each module is composed of stacked elements. The elements' length is given by a sinusoidal curve. By increasing the sinusoid amplitude, the length of the façade elements also increases.

The second case study is a pavilion truss (Figure 5); its shape is conceptually idealized to create a visual effect of a wavy structure. The objective is to reduce the maximum global stress of the bars by changing its wave parameters (Ferdinand et al. 2009).

To use the ADA workflow, a portable algorithmic design tool (AD) is required. For this reason, we used Rosetta (Lopes e Leitão 2011), which is a programming environment that is able to generate a 3D model in various CAD and BIM tools that we refer to as backends. To evaluate ADA, we introduced two new analysis backends to Rosetta, one for lighting analysis using Radiance and DAYSIM, which we compare with DIVA, a plug-in for Rhino that uses Radiance, DAYSIM, and Energy Plus; and another



4 Design iteration of the first case study.



5 Design iteration of the second case study.

for structural analysis using Robot, which we compare with Grasshopper's Karamba connected to Robot using Geometry Gym.

In both cases, we start by using Rosetta as a parametric tool to generate 3D models in Rhino, which we then analyze using Rhino's plug-ins. Then, we repeat the process but this time directly connect Rosetta to the analysis tools.

First Case Study Workflow

For the lighting analysis, we use DIVA to analyze the generated model in Radiance. The information flow is illustrated in Figure 6. Looking at the diagram on Figure 6, we can see that the information travels back and forth through different tools, which may cause information loss. This is a well-known problem affecting most tools: each one has its own data format and the export/import mechanisms have difficulties handling all the details of foreign formats. For example, DIVA is only capable of analyzing surfaces, which means that a model composed of masses must be exploded into surfaces before running the analysis. On the other hand, DIVA distributes sensor nodes for heat and radiation along each surface at a given distance, for which it computes the normal vector of each surface.

Differently from the previous workflow, ADA is based on a direct connection between the algorithmic-design tool and the analysis tool. Figure 7 illustrates the workflow for the case where we connect Rosetta directly to Radiance.

Although planar surfaces have a single normal vector, non-planar surfaces need a vector field, and when the user forgets to mesh

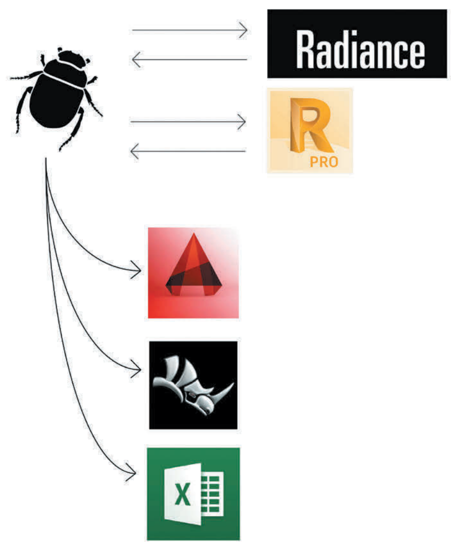
6 Workflow where Rosetta generates the model in Rhino, which DIVA adapts for Radiance to analyze, and finally DIVA displays the results back on Rhino.

non-planar surfaces, DIVA will use just one normal vector for the entire surface. This, unfortunately, has dramatic consequences in the analysis (Leitão, Branco, and Cardoso 2017), as is visible in Figures 8A and 8B, which show that, due to incorrect computation of normal vectors for two curved surfaces, the sensor nodes are incorrectly placed and the radiation analysis becomes unreliable.

When using the Rosetta's Radiance backend there is no need for extra steps, such as exploding masses or meshing surfaces. Rosetta automatically generates the analytical model containing solely the information Radiance requires for the analysis, which makes the whole process faster. Another advantage of this direct connection is that it minimizes information loss. For example, when using curved shapes, Rosetta produces an independent vector for each sensor node, which allows Radiance to correctly compute the radiation amount, as is visible in Figures 8C and 8D. A final advantage is that this connection also allows for visualizing the analysis results on other backends, such as Rhino or AutoCAD, or exporting the numerical data to a spreadsheet.

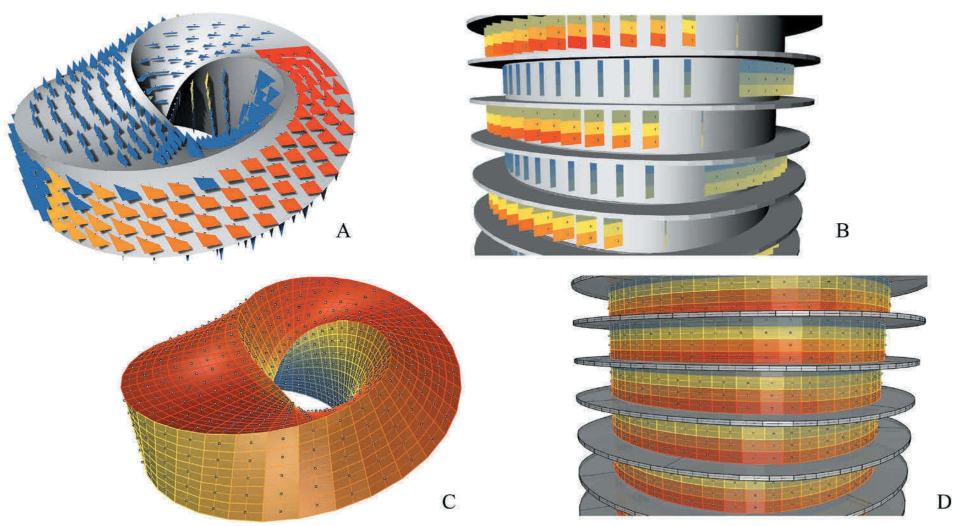
Second Case Study Workflow

In the second case study, we focus on using Robot, a structural analysis program associated with Revit, for the analysis of the truss visible in Figure 5. Robot does not support certain modelling capabilities that are meant to be performed by modelling software, and so Robot requires a previously designed model input such as a Revit model (Autodesk 2014). Ideally, a Revit model should be directly analyzable by Robot, but in fact, it is necessary to build a fully structural model. In addition, every time we introduce a change to the architectural model, we have to introduce it to the structural model separately, which is a potential source of inconsistencies. To solve this problem, we use the workflow in Figure 9, which takes advantage of the connection between Rosetta and Rhino to generate an analytical model of the truss. The model is then fed to Grasshopper's Karamba components, and the results are displayed back in



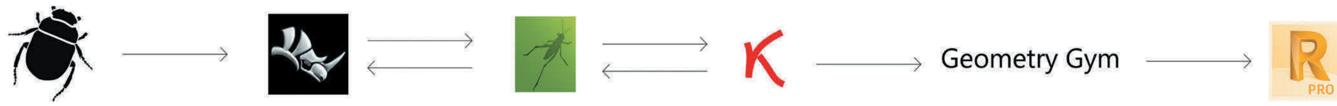
7

7 Rosetta in direct connection with Radience for lighting analysis and with Robot for structural analysis. After retrieving the analysis results, the architect can visualize them in the supported backends such as AutoCAD and Rhino.



8

8 Analysis generated by DIVA (A and B) and by Rosetta (C and D).



9 Grasshopper-Karamba connection, using Geometry Gym to export the model to Robot.

Rhino. However, since Karamba is a Rhino plug-in, only a model supported by Rhino can be used with Karamba.

In order to fulfill Karamba's requirements, it is necessary to: (1) adapt the truss model, so that the bar elements become lines and the nodes become points; (2) assign properties to the previous elements, which can be done using a particular layer organization; and (3) specify the support conditions, material properties, and loads applied to the structure. This information is normally not part of an architectural model, which means that the architect has to add it manually. However, in our case, we used the capabilities of both Rosetta and Grasshopper to do this in a parametric way.

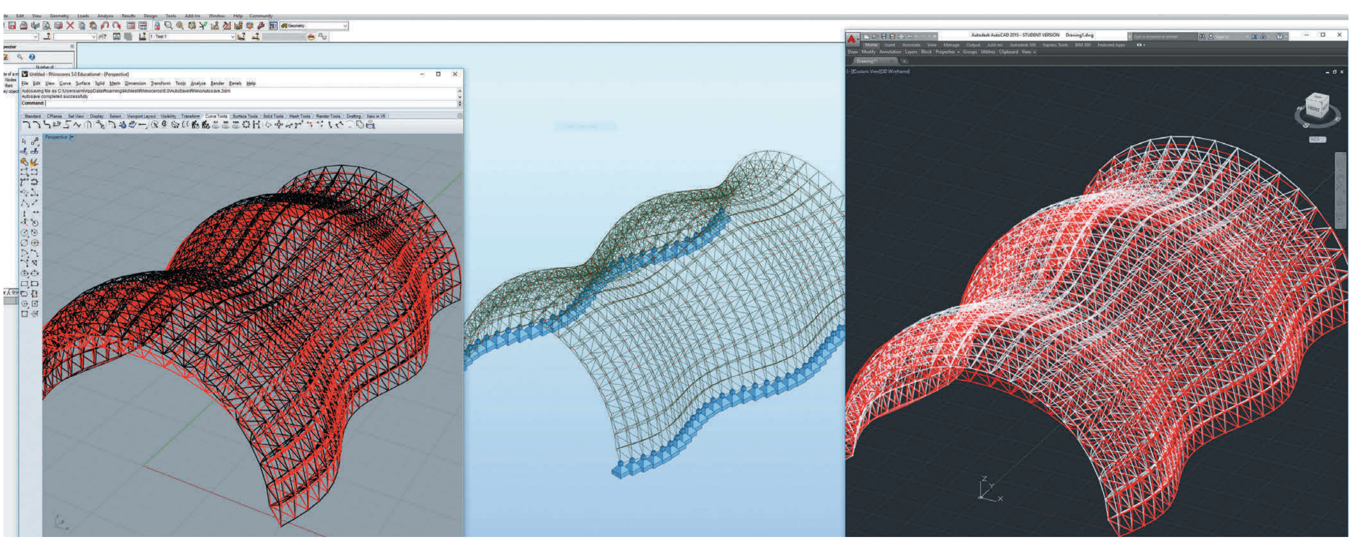
Karamba allows us to visualize the behavior of a structure, but since it has limitations with regards to accessing the analysis results (Wallin and Wasberg 2016), we decided to export the 3D Karamba model to Robot. To this end, we had to use Geometry Gym, making the workflow even more complex. Unfortunately, this only makes the results visualizable in Robot.

To overcome these problems, we directly connected Rosetta to Robot, as is visible in Figure 7, which also shows that Rosetta can present the results of the analysis either in Robot or in the other supported backends.

With our ADA approach, Rosetta is capable of generating an analytical model containing the exact information Robot requires. The script that produces the 3D model is the same that generates the analytical model and there is no need to convert the model elements every time the architect wishes to produce a new analysis. Moreover, we can specify the loads and support conditions directly using the script, which means that it is possible to rerun the analysis on every change in the model without any additional preparations. This approach not only results in a faster analysis process, but also ensures outputs that are more reliable, which encourages architects to explore more design variations. This workflow also enables architects to visualize the analysis results on their preferred modeling backend, as well as retrieving the numerical data to a spreadsheet. Figure 10 illustrates a truss deformation analysis done by Robot and visualized in Rhino and AutoCAD.

Algorithmic Analysis

Usually, architects are interested in exploring more than one variation of a design, thus an automation of the analysis process is desirable. Grasshopper and its various analysis plug-ins already provide a mechanism to observe the impact that design changes have on performance. The issue with Grasshopper's approach is that it becomes very time-consuming for buildings with high levels of complexity and detail. What should be a highly interactive procedure turns into a situation where the designer changes a parameter and has to wait a long time for the results. After the analysis is ready, the architect needs to take notice of the results



10 Truss deformation analysis done by Robot (in the back) over an algorithmically generated truss and visualization of the results on Rhino (on the left) and AutoCAD (on the right).

before trying another option.

To optimize this workflow, architects should be able to set, in advance, all the design variations they wish to perform. For this reason, Rosetta provides the option to automate a series of analyses: the architect just needs to specify the varying set and Rosetta will execute the intended analysis and collect the results. This approach can then be easily extended to optimize the design. We will now illustrate different possibilities for this optimization using the previous case studies.

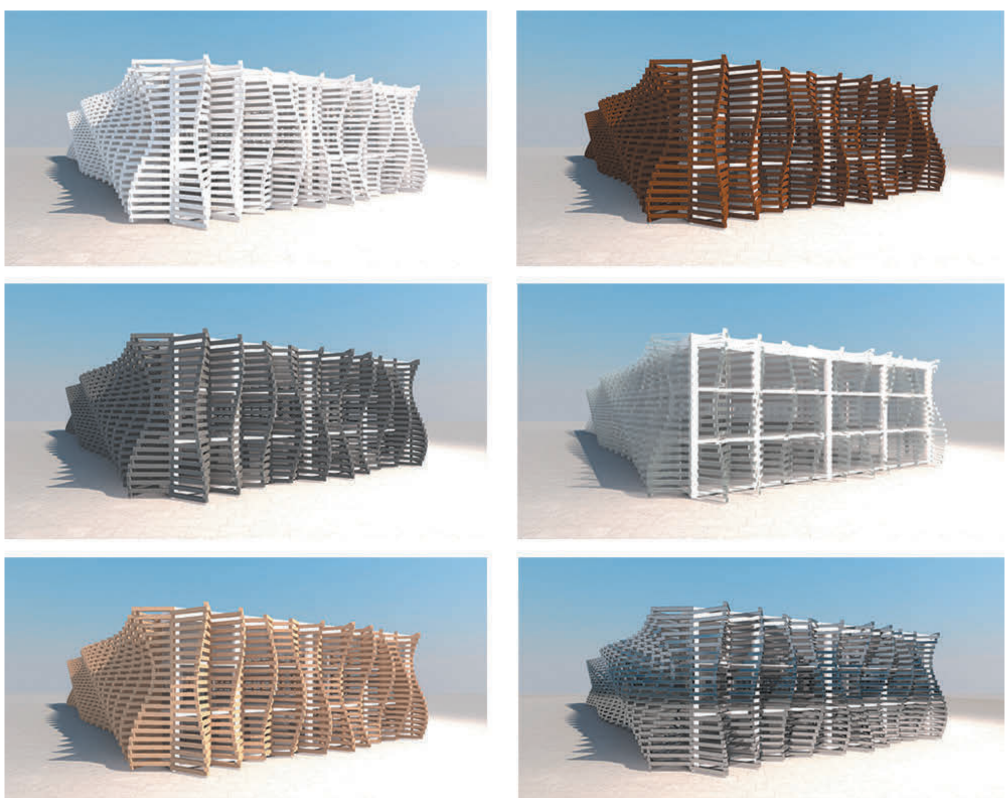
In regards to the first case study, we will improve the useful daylight illuminance (UDI) (Nabil and Mardaljevic 2005) by changing the façade materials and the wave's amplitude.

We start by considering a list containing six different materials: white enamel paint, dark wood, grey paint, translucent paint, light wood, and metal sheet. Each material has different properties that result in more or less reflection, changing the light levels inside the building. Using ADA, we iterate over the list of materials, automatically setting up a new Radiance simulation for each one. Once all analyses are finished, Rosetta provides a list containing the different variations analyzed and respective performance values. We also automate the creation of a render for each design variation so that the architect has a more realistic image of the design (Figure 11). This allows the designer to select the best solution, taking into account not only the UDI but also the aesthetics of the result.

Regarding the second case study, eight trusses were automatically generated and analyzed using different frequencies and amplitudes of the sinusoidal shape, and with a different number of nodes and bars, with the aim of reducing the maximum global bar stress of the truss. Figure 12 shows the different trusses and their respective stress values.

Algorithmic Optimization

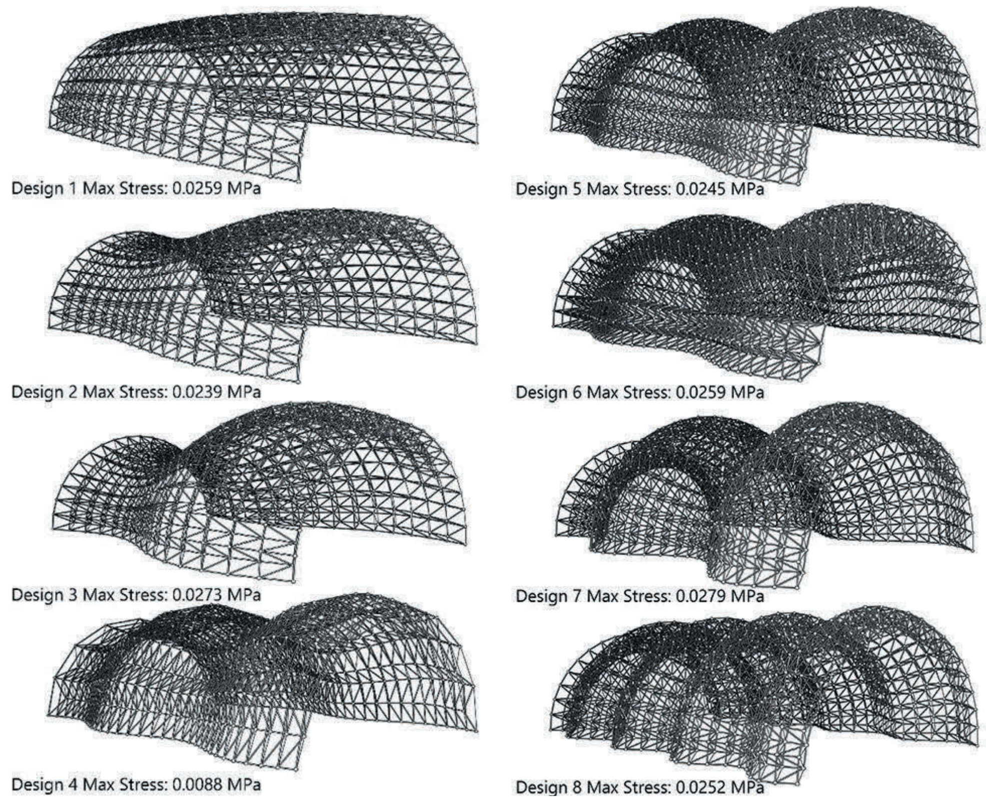
Having an automated process for generating and analyzing design variations opens the door for automating the search for the best design; in fact, there are many optimization algorithms that could be easily combined with the ADA approach. To demonstrate this, we tested the approach with the Monte Carlo method (Motwani and Raghavan 2010). We tested it in the first case study to improve the UDI and we allowed it to run with a time limit of two days, during which we were able to produce 270 design iterations, at a rate of around eleven minutes per analysis. By contrast, for the same case study, DIVA takes about forty minutes per analysis. The increased analysis speed is a consequence of the direct connection between Rosetta and Radiance, which allows Rosetta to automatically prepare each



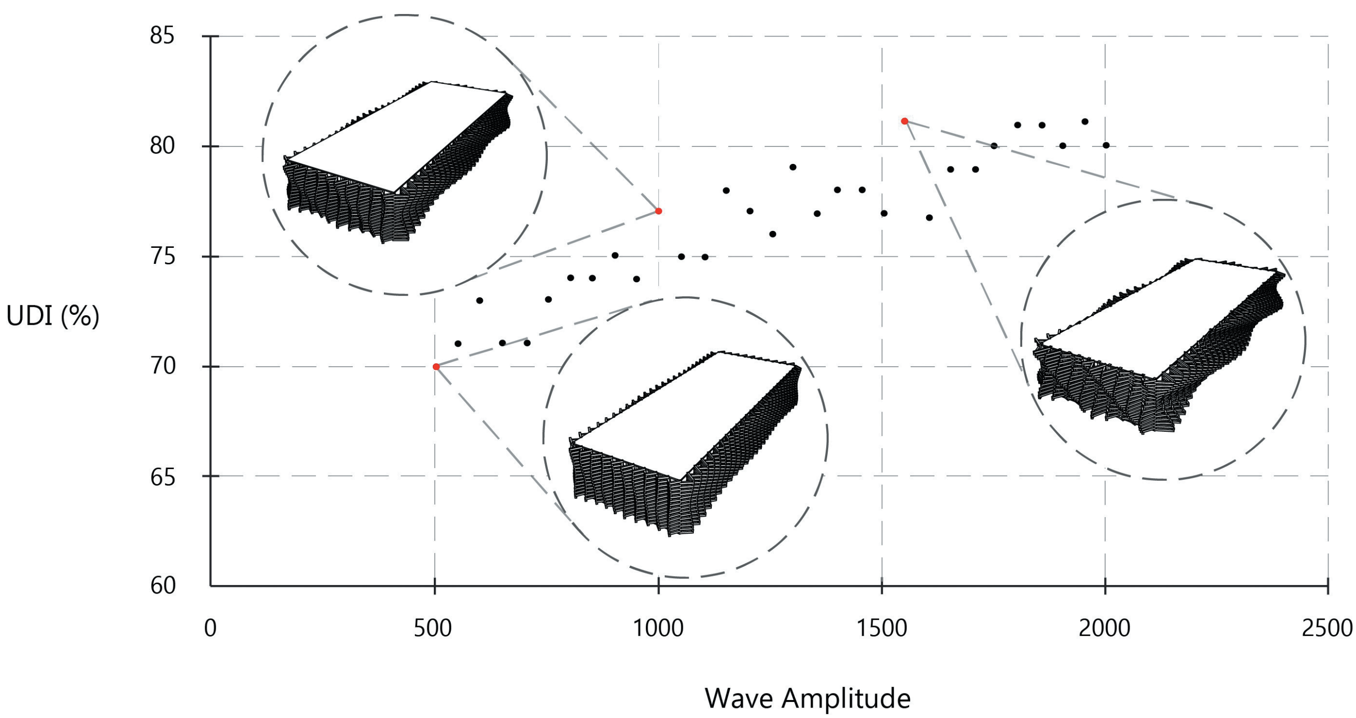
11 Different materials and respective UDIs of the first case study.

11

12 Eight truss designs and their maximum stress value (MPa).



12



13

13 Different designs found by the Monte Carlo method combined with ADA (right).

analysis and to provide Radiance with only the information needed for the analysis, excluding unnecessary details.

Figure 13 depicts the scatter plot of some design solutions, where each dot represents a different design.

CONCLUSION

This paper presents Algorithmic Design and Analysis (ADA), a method to automate performance analysis of algorithmic-based design representations. ADA facilitates the analysis process and helps architects understand the impact of their design choices on the performance of their model.

We based this method on adapting existing modeling operations that generate digital architectural models, so that they only generate the elements of the analytical model, which in turn solely contain the information required by the analysis tool. To develop this approach, we used Rosetta, an AD tool, to which we added analysis backends.

For each backend, we implement the modeling operations in a way that fulfills the backend requirements. For example, a slab that is represented as a box in a CAD backend might be represented as a pair of top and bottom surfaces in a lighting analysis backend. Similarly, a truss in a structural analysis backend might require just the location of the nodes and the edges connecting those nodes, while in another backend they might be represented as spheres and cylinders.

Using the proposed method, the designer creates just one AD model, which is then used to generate analytical versions of the model for each type of analysis. This significantly simplifies the process of analysis, thus encouraging a wider performance-oriented design exploration.

Finally, by making the analysis process algorithmic, we are able to automate the analysis of a set of designs. This way the designer does not need to set up a new analysis for each design, thus reducing the time spent in analysis tasks. Another advantage is that we can analyze designs at different scales, as it is possible to use the same AD approach to generate both a detailed model of a building or a rough representation at the urban scale.

Limitations

Algorithmic Design and analysis requires an AD representation of the design, and thus it is not applicable to manually created digital models. This can be seen as an advantage, particularly if there are plans to optimize the design, or as a disadvantage, when the effort needed to produce an AD representation does not pay off.

Future Work

Despite the positive results already obtained, ADA is still in the early phases of development. In order to make it more useful, we plan to also support design optimization and, particularly, multi-objective optimization. To this end, we will expand the number of analysis backends and we will include different optimization algorithms that the architect can select, depending on the situation.

ACKNOWLEDGEMENTS

This work was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2013.

REFERENCES

Autodesk. Integrating Autodesk Revit, Revit Structure, and Robot Structural Analysis Professional. 2014.

Burry, Jane, and Mark Burry. 2010. *The New Mathematics of Architecture*. London: Thames and Hudson.

Christensen, Jesper, Dario Parigi, and Poul Henning Kirkegaard. 2014. "Interactive Tool that Empowers Structural Understanding and Enables FEM Analysis in a Parametric Design Environment." In *Proceedings of the International Association for Shell and Spatial Structures*. Brasilia, Brazil: IASS-SLTE.

Ferdinand, Ferdinand P., Johnston E. Russell, John T. Dewolf, and David F. Mazurek. 2009. *Mechanics of Materials*. New York: McGraw Hill Higher Education.

Kalay, Yehuda. 2004. Architecture's New Media: Principles, Theories, and Methods of Computer-Aided Design. Cambridge, MA: MIT Press.

Kolarevic, Branko. 2001. Designing and Manufacturing Architecture in the Digital Age." In *Architectural Information Management: 19th eCAADe Conference Proceedings*, 117–123. Helsinki, Finland: eCAADe.

Leitão, António, Renata Castelo Branco, and Carmo Cardoso. 2017. "Algorithmic-Based Analysis." In *Protocols, Flows, and Glitches: Proceedings of the 22nd Annual Conference of the Association for Computer-Aided Architectural Design Research in Asia*, edited by P. Janssen, P. Loh, A. Raonic, and M.A. Schnabel, 137–147. Suzhou, China: CAADRIA.

Lopes, José, and António Leitão. 2011. "Portable Generative Design for CAD Applications." In *Integration Through Computation: Proceedings of the 31st Annual Conference of the Association for Computer Aided Design in Architecture*, edited by by Joshua Taron, Vera Parlac, Branko Kolarevic and Jason Johnson, 196–203. Banff/Calgary, Canada: ACADIA.

Mirtschin, Jonathan. 2011. "Engaging Generative BIM Workflows." In *Collaborative Design of Lightweight Structures*, 1–8. Sydney, Australia: LSAA.

Moon, Hyeun Jun, Min Seok Choi, Sa Kyum Kim, and Seung Ho Ryu. 2011. "Case Studies for the Evaluation of Interoperability Between a BIM

Based Architectural Model and Building Performance Analysis Programs." In *Proceedings of Building Simulation: 12th Conference of International Building Performance Simulation Association*, 1511–1526. Sydney, Australia: IBPSA.

Motwani, Rajeev, and Prabhakar Raghavan. 2010. *Randomized Algorithms*. London: Chapman & Hall/CRC, 2010.

Nabil, Azza, and John Mardaljevic. 2005. "Useful Daylight Illuminance: A New Paradigm for Assessing Daylight in Buildings." *Lighting Research & Technology* 37 (1): 41–57.

Preisinger, Clemens, and Moritz Heimrath. 2014. "Karamba—A Toolkit for Parametric Structural Design." *Structural Engineering International* 24 (2): 217–221.

Roudsari, Mostapha Sadeghipour, Michelle Pak, and Adrian Smith. 2013. "Ladybug: A Parametric Environmental Plugin for Grasshopper to Help Designers Create an Environmentally-Conscious Design." In *Proceedings of Building Simulation: 12th Conference of the International Building Performance Simulation Association*, 3128–3135. Chambéry, France: IBPSA.

Turrin, Michela, Peter von Buelow, and Rudi Stouffs. 2011. "Design Explorations of Performance Driven Geometry in Architectural Design Using Parametric Modeling and Genetic Algorithms." *Advanced Engineering Informatics* 25 (4): 656–675.

Wallin, Daniel., and Martin Wasberg. 2016. "Parametric Design of Building Structures in Cooperation with Architects: Usage and Evaluation of Structural Plug-ins in 3D Visualization Software." Master's Thesis, KTH Royal Institute of Technology.

Wan, Caiyun., Po-Han Chen, and Robert L. K. Tiong. 2004. "Assessment of IFCs for Structural Analysis Domain." *Journal of Information Technology in Construction* 9: 75–95.

Yusuf, Hauwa. 2011. "The Impact Of Digital-Computational Design On The Architectural Design Process." Master's thesis, University of Salford.

IMAGE CREDITS

All images by the authors.

Rita Aguiar is a Portuguese MSc architecture student at Instituto Superior Tecnico at the University of Lisbon, Portugal, who also studied at the University of Bath, United Kingdom, in the Faculty of Engineering & Design. At the moment she is developing her Master's thesis, which explores and researches the integration of algorithmic design approaches for the design, analysis, and optimization of architectural and structural designs.

Carmo Cardoso is a Master's graduate in architecture from Instituto Superior Técnico. He has also studied at Tongji University, Shanghai, in the faculty of Architecture and Urban Planning, and spent one year researching for Disney Research China. He is currently researching in the area of performance-based design applied to architecture.

António Leitão has a BSc in Mechanical Engineering, an MSc in Electronics Engineering, and a PhD in Computer Science and Engineering, all from Instituto Superior Técnico (IST) at the University of Lisbon. Currently he is Assistant Professor at the same university, Scientific Coordinator of the Software Engineering Group at INESC-ID, and Coordinator of the Architecture and Computation Group, teaching, lecturing, and researching on bringing together the fields of Computer Science and Architecture.