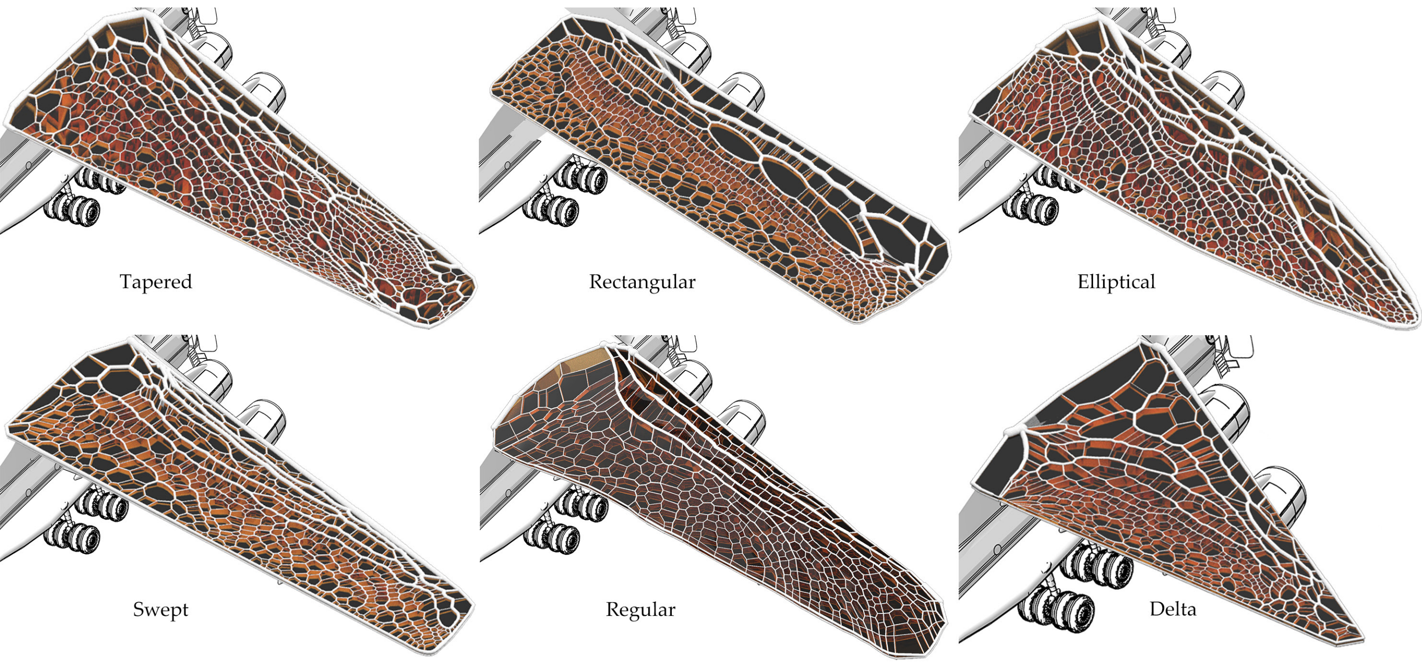
A Web-based Interactive Structural Pattern Generation Tool with Graphic Statics and Machine Learning of Dragonfly Wings

Hao Zheng
Masoud Akbarzadeh

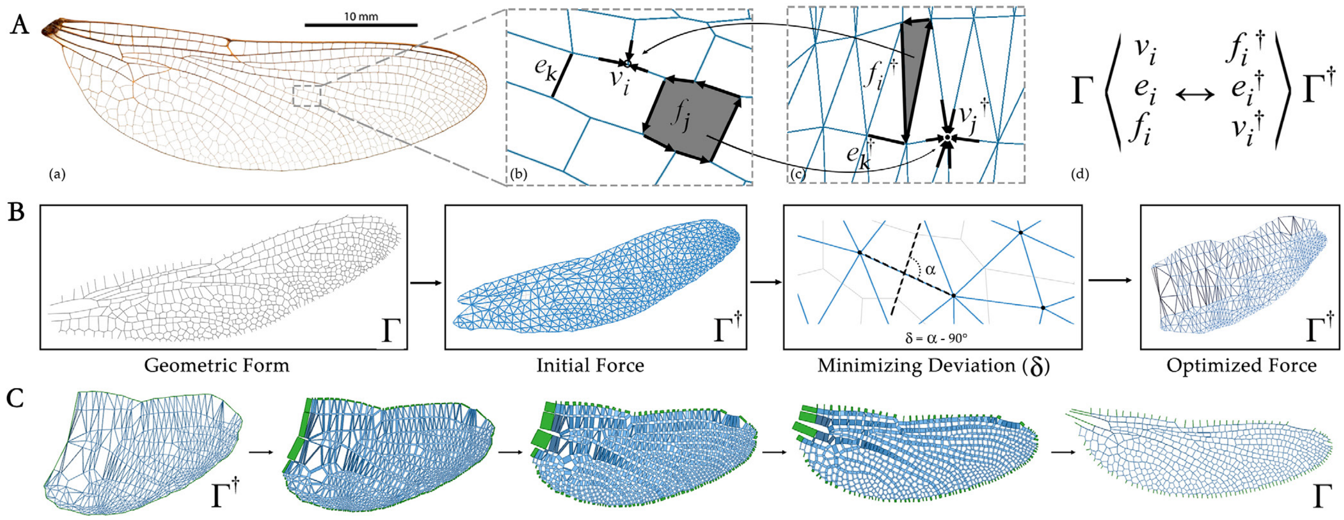
Polyhedral Structures Laboratory,
Stuart Weitzman School of
Design, University of Pennsylvania



1 Airplane wing structures generated by our tool

ABSTRACT

Designers, engineers, and scientists have always drawn inspiration from nature. The dragonfly wing is among the many natural structures that have intrigued many researchers to study its geometry and performance as a bioinspired design. In previous research, we developed a workflow to use Graphic Statics to analyze the dragonfly wing structure and machine learning models to generate the topology and geometry of a dragonfly wing structure. However, the current workflow involves multiple geometric algorithms and the implementation of complex machine learning models, making it is difficult for designers to follow and use. Therefore, in this paper, we introduce a web-based tool that implements the workflow. It includes (1) the input control panel from the user to define the constraints of the structure; (2) the backend server that proceeds the generation of the structure with Graphic Statics and machine learning models; (3) the output control panel to allow the interaction of the result for frontend display; and (4) the file manager to store and restore the generated result. On the web page, designers can easily input the boundary and parameters of a wing/cantilever structure and generate a funicular structural form in our cloud server. Analytical results such as Minkowski Sum and FEM analysis are shown to the user. Finally, the user can export the STL model for other purposes such as aerodynamic analysis or digital fabrication.



2 Previous research on the Graphic Statics of dragonfly wing structures: (A) The form and force diagrams; (B) The iterative solution of Graphic Statics; (C) The Minkowski Sum between the force diagram and the force diagram

INTRODUCTION

Graphic Statics on Natural Structures

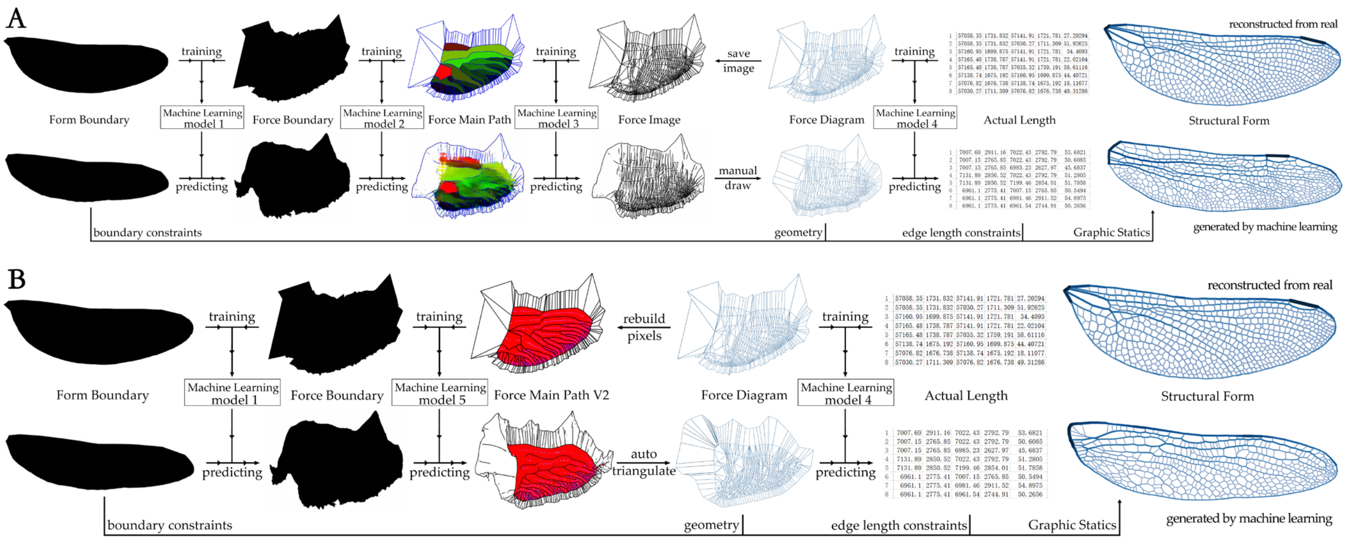
The geometry of a structural system with axial-only internal loads can be related to its force equilibrium using two geometric diagrams as proposed by J. Clerk Maxwell (Maxwell 1864): (a) the form diagram that represents the form of the structure, the length of members, and the locations of supports and applied loads; and (b) the force diagram that consists of closed polygonal faces and shows the equilibrium of forces in each node of the structure. The numbers of the edges of the two diagrams are equal, and the magnitude of the force in each member of the structure is proportional to the lengths of the corresponding edge in the force diagram. This method is known as Graphic Statics (Akbari, Akbarzadeh, and Bolhassani 2019; Akbari et al. 2020; Akbari, Lu, and Akbarzadeh 2021; Akbari et al. 2022) and has been used previously to describe the force equilibrium of convex-only, natural networks such as a spider web (Whiteley et al. 2013). In the case of a dragonfly wing, the internal 2D pattern bounded by the boundary edges mainly consists of convex polygons. Thus, the method of graphic statics can be used to analyze the static equilibrium of forces in the system (Figure 2A).

The geometry of the force equilibrium for the network of the wing is found using iterative methods. The wing's network will be referred to as the form diagram from which the topology of its dual force diagram is extracted. Each polygon in the form diagram is reciprocal to a vertex in the force and vice versa. Subsequently, the edges of the force diagram are rotated iteratively to become normal to the edges of the form diagram. The difference measured from the right angle of 90° and the angle of the two corresponding edges is defined as the deviation. The iteration process minimizes the value of the deviation to derive a solution of the force diagram within the predefined tolerance (Figure 2B). The PolyFrame plugin (Nejur and Akbarzadeh 2021) was used for this iterative operation which is a free plugin for Rhinoceros software (Robert McNeel & Associates 2018)..

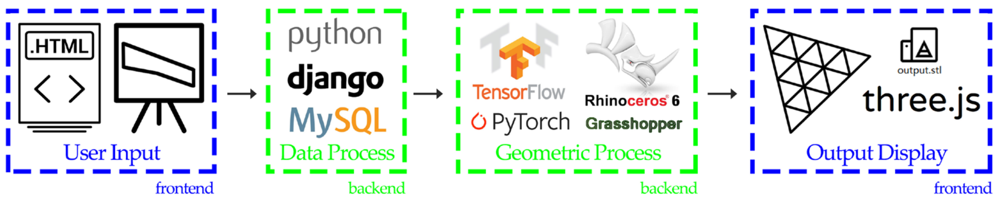
Machine Learning of Topology

Based on the above research, in the following phase, we developed two methods using Machine Learning models capable of generating the entire structural form of the wing from a user-input boundary with an intermediate product of the force diagram. The implemented machine learning techqnues inlcude Generative Adversarial Network (GAN) to generate patterns as images, and Artificial Neural Network (ANN) to predict structural properties as vectors. In the first method (Figure 3A), to build the training and testing dataset for the machine learning models, geometries in different stages are transformed into images. Inspired by the identification of the main veins by (Hoffmann et al. 2018), a similar method is developed to extract the main path of the force diagram of the dragonfly wing. Then, image-to-image machine learning models (Isola et al. 2017) are used to learn the mapping between each stage of the dragonfly wing data and generate the force diagram from the form boundary as an image. In addition, a vector-based machine learning model is trained to predict the edge lengths of the form diagram using a dataset of the edge lengths extracted from the dragonfly wing geometries. Therefore, the four machine learning models can predict all information needed to generate the structural form of the wing.

However, in the first method, the geometries of a force diagram need to be reconstructed from the image manually



3 Previous research on the machine learning of the topology of the dragonfly wing structures: (A) Machine learning models with a manual process to draw the force diagram and generate the structural form; (B) Machine learning models with an automatic process to directly generate the structural form



4 The proposed workflow of the web tool

to maintain precision. To automatically generate the force geometry, we develop the second method that represents the vertex information as the pixel values in the main path image and triangulates the regions in the main path according to the recognized vertexes (Figure 3B). Thus, the force geometry can be directly generated and proceeded into the structural form. The first method produces more visible information for a human to understand, while the second method generates abstract information for the machine to rebuild the geometry.

Problem Statement and Objectives

With the above previous research (Zheng, Hablicsek, and Akbarzadeh 2021; Zheng and Akbarzadeh 2022), we have successfully developed the workflow of generating funicular wing/cantilever structures with Graphic Statics and machine learning. However, technically speaking, implementing the workflow locally on a computer is particularly difficult for a design without much computational and programming knowledge.

Therefore, to better help designers access our method, we develop a web-based tool that accepts user inputs and feedback generated structures. The web tool is implemented as an online resource and open for designers to visit as a web page. Meanwhile, a local server proceeds the input data, generates the output structure, and sends the model file to be displayed on the web page. With this web tool, designers can easily adjust the input parameters and boundary conditions, and obtain the structure model online, without going through the complex local computing process.

To be specific, Figure 4 shows the proposed workflow of the web tool, which contains the frontend and backend. It follows four steps: 1) The user inputs the boundary and parameters in the web page, and HTML and Javascript (Eich 2020) transform the data from Canvas (Fulton 2011) into a formatted database; 2) The server stores and proceeds the data (Holovaty 2022; Widenius 2022), and updates an indicator file; 3) The geometric components (Robert McNeel & Associates 2018; Rutten 2020) and the machine learning components (Google 2019; Paszke 2018) in the server work together to generate the structural model as CSV file, and send it back to the web page; 4) the web page (Cabello 2022) displays the model, and allows the user to generate and download the STL file.

METHODOLOGY

User Input

First, the web page should receive the input boundary and parameters from the user. Figure 5 shows the icons, names, ranges, and buttons in the "Input Control Panel." The boundary is defined as a closed polygon of several vertexes. The user can select the number of vertexes and adjust the positions of each vertex by dragging them on the canvas. The boundary

Icon	Name and range	Icon	Name and range
	Input Boundary (canvas input)		Subdivision Density (0.01-1.00)
	Sharpness (0.00-1.00)		Length Constraint Multiplier (0.01-2.00)
	Boundary Constraint Multiplier (0.01-9.99)		Iterations (10000-30000)
	Total Length of the Wing (10-50000)		Machine Learning Model (selection)
	Default Settings (button)	Submit	Submit (button)
help	Instruction on how to use (button)	about	Developer information (button)

5 The control parameters and functional buttons in the user input panel and the compute panel

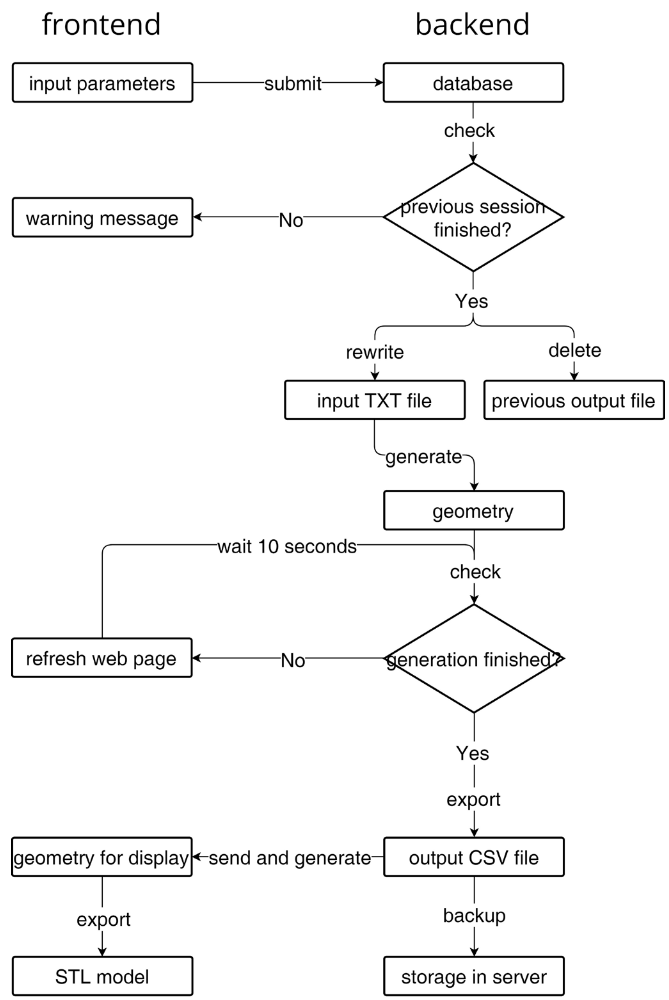
vertexes are transformed into numeric values based on their coordinates. Noted that, the most left curve in the boundary is the place where the structure is anchored. Therefore, additional marks are made as slash lines to represent the anchor.

Besides, the “Input Control Panel” also includes parameters that are defined as numeric values. A set of six parameters can be input from the user and sent to the server. To be specific, “Subdivision Density” defines the density of the structural members. By increasing it, more members will be generated with a longer time cost. “Sharpness” defines the upper and lower bounds of the length constraint for each edge. Increasing it will give more freedom to the geometric generation process in graphic statics. “Length Constraint Multiplier” defines the relaxation of the edge length constraints. Increasing it will cause more rectangular cells than circular cells. “Boundary Constraint Magnitude” defines the magnitude of the boundary constraint to the form. Increasing it will make the structure attach closer to the boundary. “Iterations” defines the number of iterations in the geometric generation process in graphic statics. Increasing it will generate a more accurate structure but with a longer time. “Total Length of the Wing” defines the size of the generated model in millimeter. Last, the user can select the machine learning model of different species, the default is set as the dragonfly wing model.

Noted that, the web page will restore the input parameters when the user successfully submitted the last time, thus the user can more easily adjust the parameters. There is a button “Default Settings,” by clicking it, the input parameters will be set as the default values. If a new user does not understand the meaning of each parameter, he/she can move the mouse cursor to the button of each parameter to see its name and click the “help” or “about” button to see the detailed instructions.

Backend for Geometric Processes

When finishing adjusting the six input parameters, the user can click the “Submit” button in the “Compute” panel to send the first set of input parameters to the server. Figure 6 shows



6 The workflow in the backend and the frontend of the server after the user submits the input parameters

the algorithm in the backend and the frontend of the server after the user submits the input parameters. First, in order to avoid duplicated submissions from multiple users before the server completely responds to the current request, an indicator value in the server is firstly loaded by the web page. If it indicates that a request is being proceeded, a warning message will be shown to the user when he/she clicks the “Submit” button, and no values will be submitted to the server. Besides, the entire generation process in the server usually takes three to seven minutes. The web page will be automatically refreshed every 10 seconds until it receives the result from the server.

When the server finishes the generation process, it will send back a CSV file to the web page, which contains the geometric information of the generated structure, the graph information for implementing the Minkowski Sum, and the numeric information of the FEM analysis. The geometric information contains the coordinates of the start and end points of each

Icon	Function	Icon	Function
	View the generated model		Remove the generated model from the scene
	View the Minkowski Sum		Remove the Minkowski Sum geometry from the scene
	Show the external forces		Hide the external forces
	Show color-coding		Hide color-coding
	Turn on the FEM result under the load of self-weight		Turn off the FEM result under the load of self-weight
	Turn on the FEM result under the point load on the farthest vertex		Turn off the FEM result under the point load on the farthest vertex
	Material of the Wing (selection)	FEM	Request the recomputation of the FEM results from the server

7 The functional buttons in the output control panel and the FEM control panel

edge, as well as its corresponding force magnitude. The graph information stores the connectivity matrix of the form and the force diagrams. The FEM analysis result contains the deformation magnitude for each edge in the structure.

Frontend for Display

Next, when the web page receives the data file, the user can choose a different display mode in the “Output Control Panel” and “FEM Control Panel” (Figure 7) to turn on or off the generated model and the Minkowski Sum. Also, the user can turn on or off the FEM results under the load of self-weight or under the point load in the farthest vertex.

In the case of the normal display mode, the web page regenerates the structural members according to the information from the file and the second set of the user input parameters of the minimum radius and the maximum radius. An additional transparent box geometry is shown to indicate the anchor of the structure. The user can control the camera with the mouse in the main display window to better view the generated model. The generated 3D model is displayed on the web page with pre-set lighting environment. However, to reduce the computational load from the local device, the shadow is represented as a series of static geometries on the ground with gray lines. The color setting for the main geometry keeps constant with that in PolyFrame. In the display control panel, the user can also change to turn on or off the display of the external forces.

In the case of the Minkowski Sum mode, the web page reads the user input of the Minkowski Sum indicator (MSI) and calculates the corresponding status in the form-to-force transformation. The graph information in the feedback file contains the following items: (1) the coordinates of the vertexes in the form diagram; (2) the index of neighbor cells of each cell in the force diagram; (3) the index of the shared edges in the neighbor cells of each cell in the force diagram; (4) the index of edges in each cell; and (5) the coordinates of the start and end points of each edge in the force diagram.

Icon	Function	Icon	Function
STL	Export the current scene as an STL model	1	Restore the pre-generated sample 1 from the server
CSV	Download the vector-based data file to save the result	2	Restore the pre-generated sample 2 from the server
CSV	Upload the vector-based data file to restore the result	3	Restore the pre-generated sample 3 from the server

8 The functional buttons in the file manager panel

By scaling the cells in the force diagram with the MSI value and moving them to each corresponding vertex in the form diagram, each edge in the force diagram will become an area with thickness. Therefore, with a gradually-changed MSI value from 0 to 1, the areas shift from the edges in the form diagram to the edges in the force diagram, thus showing the transformation between the form and force. Still, the user can turn on or off the external forces in the Minkowski Sum mode.

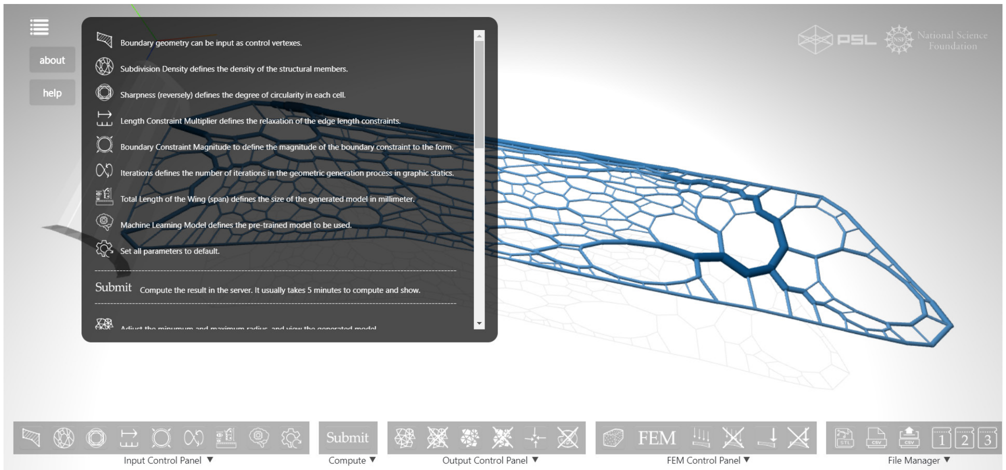
For the FEM analysis, Karamba (Karamba 2022) is used to calculate the deformation of edges based on the user input of the span and the material of the structure. Applicable materials include steel, wood, concrete, and aluminum, and the material property is embedded in Karamba. We provide two types of loading: (1) self-weight loading for all vertexes; and (2) point loading for the farthest vertex to the anchor with the magnitude of half of the self-weight. In the CSV data file, the FEM analysis part includes the following information for each edge: (1) coordinates (*x, y, z*) for the start and end points (deformation in *z*-axis included); and (2) color-coding value (R,G,B). The user can adjust the multiplier in the frontend to increase or decrease the deformation magnitude, and view the color-coded edges and color scales. When the user changes the material setting or the structural thickness, a recomputation request can be sent to the server and the FEM results will be updated in round ten seconds.

Server-free Data Files

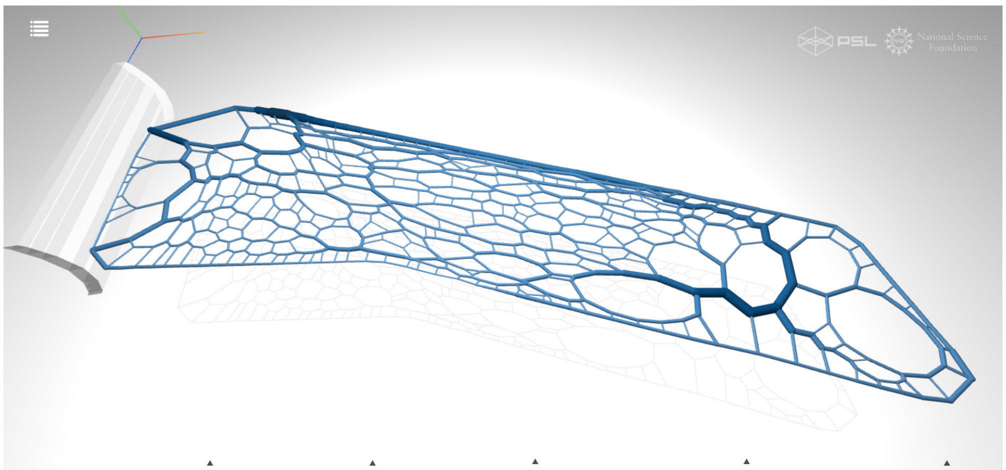
In addition, to better help users restore the previously generated results, in the “File Manager” panel (Figure 8), users can download the data as a CSV file and save it on their local computer. If the user input an email address when submitting a request to the server, when the computation is finished, the server will send an email to the user with the CSV data file attached. By uploading the file to the web page, users can restore the input parameters, the output structure, the Minkowski Sum geometry, and the FEM analysis results. Restoring the previous result only requires the local computer from the user, thus it is an offline process and does not require a connection to the server. Also, the user can export and download the STL file for 3D printing or other purposes.

UI Design of the Web Page

Therefore, with this web page implementation (http://www.



9



10

9 The implemented web page with panels unfolded

10 The implemented web page with panels folded

ai-gs.com/frontend/DFW-GH.html) (PSL 2022), users even without much knowledge of machine learning and graphic statics could easily generate lightweight and high-performance structures within given boundaries. Figure 9 shows the web page with control panels unfolded, and Figure 10 shows the web page with control panels folded. The user can freely decide to fold or unfold each panel to better balance the UI and the model display.

APPLICATIONS

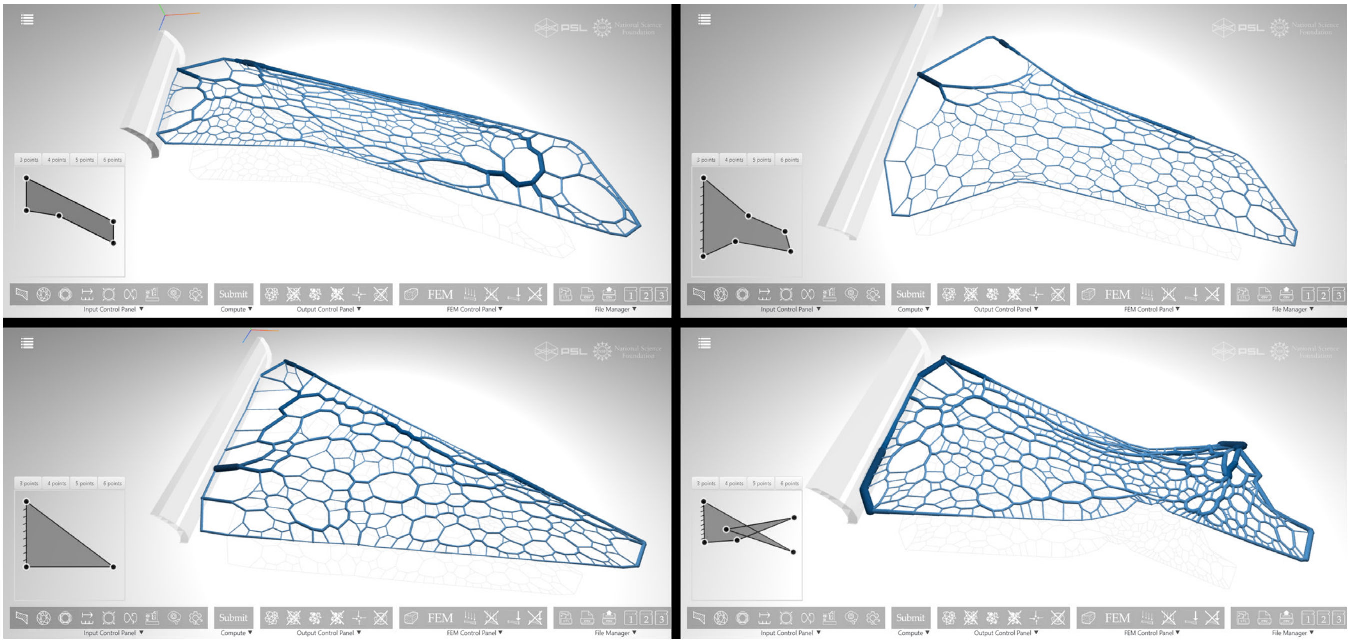
In this section, several cases are generated and shown using our web tool. The user can either submit multiple requests to our backend server to generate structures with different input-related parameters or adjust the output-related parameters in the frontend to view and export results.

Form-finding with Various Boundaries

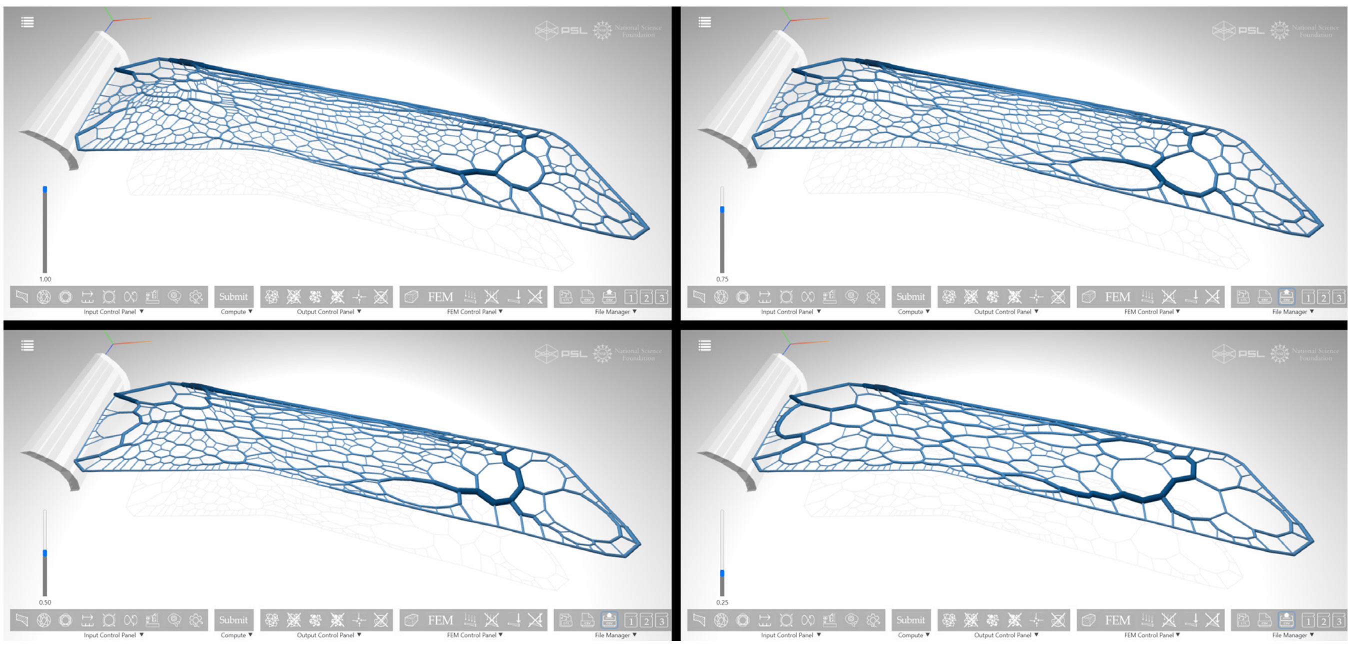
In the first case (Figure 11), structures with different input boundaries are generated. The user can select the number of control points in the input boundary, and drag the points to adjust their positions. Even if the user inputs an invalid boundary such as crossing curvatures (Figure 11 bottom right), our geometric script will merge it into a pixel-based black-and-white image, and send the image to machine learning models. In addition, the user is not required to input a wing-like boundary. As long as the anchor is on the left, any cantilever structure can be generated with the input boundary, which contains the features of dragonfly wings.

Precise Control of Generated Structures

The second case shows one example of controlling other input parameters such as the subdivision density (Figure 12). Among the input parameters, the subdivision density is the most important one since it directly controls the complexity of the generated structure. A smaller subdivision density can significantly simplify the structure while keeping the features of dragonfly wing patterns. In our recent research, we found that simplifying the structure to a certain degree would increase the structure performance, and make it easier to fabricate in the real world. Thus, the user can consider the fabrication ability and adjust the subdivision density. Other parameters such as sharpness and iterations can also greatly affect the generated result, but changing the input parameters requires a re-computation from our server and it usually takes around five minutes to respond.



11 The generated structures with different user-input boundaries



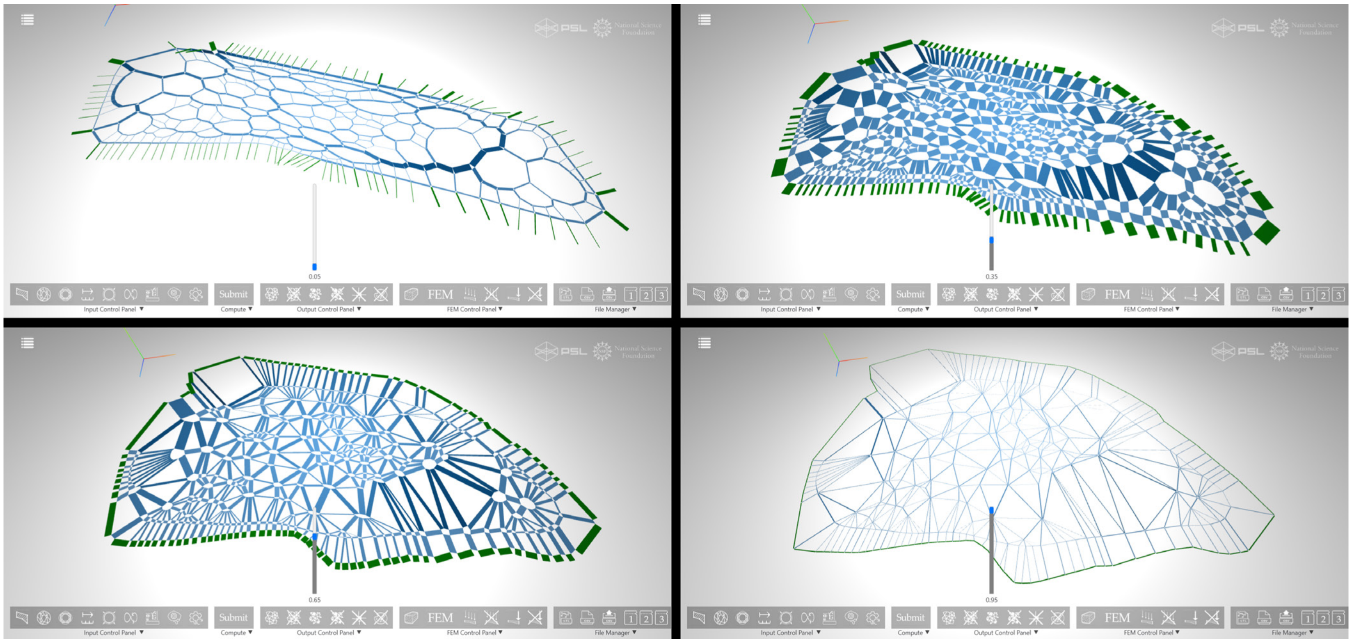
12 The generated structures with different subdivision densities: (top left) SD=1.0; (top right) SD=0.75; (bottom left) SD=0.5; (bottom right) SD=0.25

Minkowski Sum Display

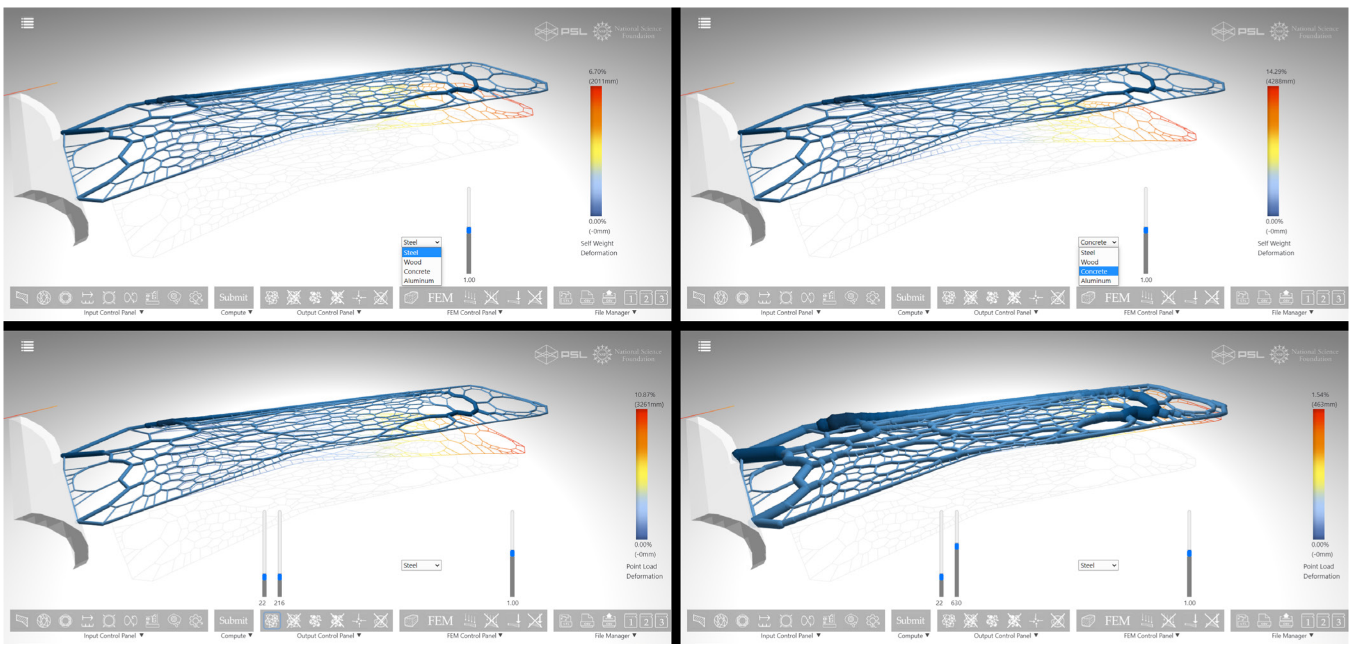
Besides changing the input parameters, the user can also adjust the output display modes and the related indicators to show or hide the generated structures and the analytical results. Figure 13 shows the case of Minkowski Sum in different stages with different values of the indicator. The indicator defines position in percentage of the current Minkowski Sum in the form-to-force transformation. The result is closer to the form diagram with a smaller value of the indicator, while it is closer to the force diagram with a larger value of the indicator. The user can adjust the value of the indicator and view the smooth transformation from the form to the force.

Finite Element Method Analysis

Also, Figure 14 shows the FEM results under different loading scenarios and materials. As mentioned, in the geometric mechanism of our server, we provide the FEM results of the deformation under the self-weight or a point load. The user can decide to show none/one/both of them by clicking the corresponding buttons in the output control panel, or hide the main structure to better compare the FEM results. The color-coded scale is also shown on the right of the web page when the corresponding FEM result is shown. The scale includes the minimum and maximum values of the percentage of deformation compared with the span, and the real values of



13 The generated Minkowski Sum with different stage indicators: (top left) MSI=0.05; (top right) MSI=0.35; (bottom left) MSI=0.65; (bottom right) MSI=0.95



14 The FEM analysis results with different loading scenarios, materials, and structural thickness: (top left) self-weight load with steel material; (top right) self-weight load with concrete material; (bottom left) point load with steel material; (bottom right) point load with steel material and larger structural thickness

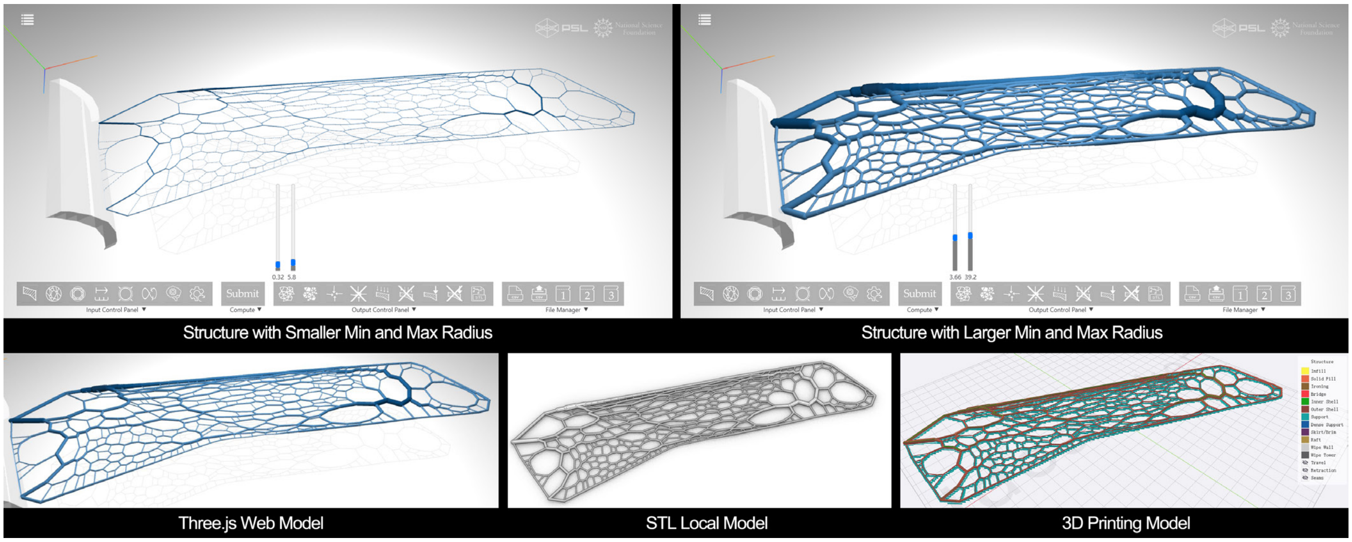
the deformation in millimeter. When keeping the deformation multiplier constant, the user can also directly compare the FEM results from different structures.

Exported Models

The final case shows the application of our web tool in exporting the result to other platforms for various purposes. In the normal display mode, the user can adjust the minimum and the maximum radius to control the range of the thickness for edges, and export the structure as an STL model (Figure 15). The exported STL model can be imported to a variety of platforms, including modeling software for further analysis and 3D printing software for digital fabrication. Therefore, our web tool completes the logic of loop by accepting the user input and exporting the generated result back to the user.

CONCLUSION AND DISCUSSION

This paper introduces a web-based structural design tool that implemented the workflow of Graphic Statics and machine learning of the dragonfly wing structures. It helps designers generate funicular cantilever structures without any installation on the local computer, by inputting the boundary and



15 (top) Structures with different minimum and maximum radius; (bottom) Exported STL model and its 3D printing preview

control parameters directly on the web page. The backend server computes the topology and geometry of the structure, and the frontend web page displays the generated structure, Minkowski Sum, and FEM analysis results. The user can also export the STL model for further purposes, or download and upload the CVS data file to retrieve the previous results.

In addition, this web-based implementation method can not only be used in serving our workflow of generating structures, but also be used in any local computational process involving the geometric generation and machine learning predictions in various design fields, such as architectural geometry generation (Huang and Zheng 2018; Ren and Zheng 2020), urban feature prediction and plan generation (Zheng and Yuan 2021; He and Zheng 2021), and structural evaluation and generation (Zheng, Moosavi, and Akbarzadeh 2020). Especially in the situation that the workflow involves the data flow in multiple platforms and scripts, this implementation method can reduce the risk of errors raised by compatibility issues in local computers, thus serving the input and output without installation requirements to the user.

To further develop our web tool, the future research includes the following aspects. In the UI design, free-drawn boundaries will be allowed for users to input any closed area as input images. The intermedia outputs such as the force diagram will be allowed to export for further manipulation of design purposes from users, thus users can freely apply the force diagram to different architectural and structural design cases. In the system design, future improvement includes: (1) parallel computing to serve a larger number of users; (2) user login system and online data storage; (3) precise structural analysis options; and (4) implementation of the machine learning models from other natural structures. Besides, with the generated structural forms, further materialization experiments can proceed, depending on the usage of the structures.

REFERENCES

Akbari, M., Y. Lu, and M. Akbarzadeh. 2021. “From design to the fabrication of shellular funicular structures.” In *ACADIA 21: Realignments: Toward Critical Computation; Proceedings of the 41st Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA)*. Online.

Akbari, Mostafa, Masoud Akbarzadeh, and Mohammad Bolhassani. 2019. “From Polyhedral to Anticlastic Funicular Spatial Structures.” In *Form and Force: Proceedings of the IASS Annual Symposium 2019*. Barcelona, Spain.

Akbari, Mostafa, Armin Mirabolghasemi, Hamid Akbarzadeh, and Masoud Akbarzadeh. 2020. “Geometry-based structural form-finding to design architected cellular solids.” In *SCF '20: Proceedings of the 5th Annual ACM Symposium on Computational Fabrication*.

Akbari, Mostafa, Armin Mirabolghasemi, Mohammad Bolhassani, Abdolhamid Akbarzadeh, and Masoud Akbarzadeh. 2022. “Strut-Based Cellular to Shellular Funicular Materials.” *Advanced Functional Materials* 32 (14): 2109725.

Cabello, Ricardo. *Three.js*. V. 0.151.3. GitHub. PC. 2022.

Eich, Brendan. *Javascript*. V. 2020. Brendan Eich. PC. 2020.

Fulton, Steve. *HTML Canvas*. V. 5. Steve Fulton. PC. 2011.

Google. *TensorFlow*. V. 2.0. Google. PC. 2019.

He, Jingyi, and Hao Zheng. 2021. “Prediction of crime rate in urban neighborhoods based on machine learning.” *Engineering Applications of Artificial Intelligence* 106 (Nov): 104460.

Hoffmann, Jordan, Seth Donoughe, Kathy Li, Mary K. Salcedo, and Chris H. Rycroft. 2018. “A simple developmental model recapitulates complex insect wing venation patterns." In *Proceedings of the National Academy of Sciences* 115 (40): 9905–9910.

Holovaty, Adrian, and Simon Willison. *Django*. V. 4.2. PyPI. PC. 2022

Huang, Weixin, and Hao Zheng. 2018. "Architectural Drawings Recognition and Generation through Machine Learning." In *ACADIA 18: Recalibration; Proceedings of the 38th Annual Conference of the Association for Computer Aided Design in Architecture*. Mexico City, Mexico.

Isola, Phillip, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. "Image-to-image translation with conditional adversarial networks." arXiv preprint.

Maxwell, J. Clerk. 1864. XLV. "On reciprocal figures and diagrams of forces." *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 27 (182): 250–261.

Nejur, Andrei, and Masoud Akbarzadeh. 2021. "Polyframe, efficient computation for 3d graphic statics." *Computer-Aided Design* 134 (May): 103003.

Paszke, Adam, Sam Gross, Soumith Chintala, and Gregory Chanan. *PyTorch*. V. 1.0.0. PyTorch. PC. 2018.

Polyhedral Structures Laboratory (PSL). 2022. "Wing Structure Generator, Machine Learning + Graphic Statics." Accessed April 28, 2022. http://www.ai-gs.com/frontend/DFW-GH.html.

Preisinger, Clemens. *Karamba3D*. V. 2.2.0. Karamba3D. PC. 2022.

PSL. 2022. "Wing Structure Generator | PSL." accessed 04/28. http://www.ai-gs.com/frontend/DFW-GH.html.

Ren, Yue, and Hao Zheng. 2020. "The Spire of AI - Voxel-based 3D Neural Style Transfer." In *Proceedings of the 25th International Conference on Computer-Aided Architectural Design Research in Asia (CAADRIA)*. Bangkok, Thailand.

Robert McNeel & Associates. *Rhino*. V. 6. Robert McNeel & Associates. PC. 2018.

Rutten, David. *Grasshopper*. V. 1.0.0007. Robert McNeel & Associates.

Whiteley, Walter, Peter F. Ash, Ethan Bolker, and Henry Crapo. 2013. "Convex polyhedra, Dirichlet Tessellations, and Spider Webs." In *Shaping Space*, edited by M. Senechal. New York: Springer.

Widenius, Ulf Michael. *MySQL*. V. 8.0. Oracle. PC. 2022

Zheng, Hao, and Masoud Akbarzadeh. 2022. "The Dragonfly Wing Project." *Architectural Design* 92 (3): 132–133.

Zheng, Hao, Marton Hablicsek, and Masoud Akbarzadeh. 2021. "Lightweight Structures and the Geometric Equilibrium in Dragonfly Wings." In *Proceedings of International Association for Shell and Spatial Structures Annual Symposia (IASS)*. Guildford, UK.

Zheng, Hao, Vahid Moosavi, and Masoud Akbarzadeh. 2020. "Machine Learning Assisted Evaluations in Structural Design and Construction." *Automation in Construction* 119 (Nov): 103346.

Zheng, Hao, and Philip F Yuan. 2021. "A Generative Architectural and Urban Design Method through Artificial Neural Networks." *Building and Environment* 205 (Nov): 108178.

IMAGE CREDITS

All drawings and images by the authors.

Hao Zheng graduated from the PhD program at the University of Pennsylvania, specializing in machine learning, digital fabrication, mixed reality, and generative design. He holds a Master of Architecture degree from the University of California, Berkeley, and Bachelor of Architecture and Arts degrees from Shanghai Jiao Tong University. Previously, Hao worked as a research assistant at Tsinghua University and UC Berkeley with a concentration on the robotic assembly, machine learning, and bio-inspired 3D printing. His teaching experience includes workshop tutor at Tongji University, lecturer at the University of Pennsylvania, and teaching fellow at Shanghai Jiao Tong University.

Masoud Akbarzadeh is a designer with academic background and experience in architectural design, computation, and structural engineering. He is an Assistant Professor of Architecture focusing on Structures and Advanced Technologies and the Polyhedral Structures Laboratory (PSL) director. He holds a DSc from the Institute of Technology in Architecture, ETH Zurich, where was a research assistant in the Block Research Group. His main research topic is Three-Dimensional Graphical Statics, a novel geometric method of structural design in three dimensions. In 2020, he received the National Science Foundation CAREER Award to extend the methods of 3D/Polyhedral Graphic Statics for Education, Design, and Optimization of High-Performance Structures.