Linking Early Design Stages with Physical Simulations using Machine Learning

Structural Analysis Feedback of Architectural Design Sketches

Rasoulzadeh S., Senk V., Kovacic I., Reisinger J., Füssl J., Hensel M.
TU Wien, Austria
shervin.rasoulzadeh@tuwien.ac.at

Abstract. State-of-the-art workflows within Architecture, Engineering, and Construction (AEC) industry are still caught in sequential planning processes. Digital design tools founded in this domain often lack proper communication between different stages of design and relevant domain knowledge. Furthermore, decisions made in the early stages of design, where sketching is used to initiate, develop, and communicate ideas, heavily impact later stages, offering the need for fast feedback to the architectural designer to proceed with adequate knowledge regarding design implications. Accordingly, this paper presents research on a novel AEC workflow based on a 4D sketching application targeted for architectural design as a form-finding tool coupled with two modules: (1) *Shape Inference* module, which is aided by machine learning enabling automatic surface mesh modelling from sketches, and (2) *Structural Analysis* module which provides fast feedback with respect to the mechanical performance of the model. The proposed workflow is a step towards a platform integrating implicit and explicit criteria in the early stages of design, allowing a more informed design leading to increased design quality.

1. Introduction

Architecture, Engineering, and Construction (AEC) shapes our built environment, having substantial environmental, cultural and economic influence on society. However, among the least digitised industries, it is still caught in silo-thinking and sequential planning processes, as experts from different disciplines must work together and communicate with each other throughout different stages of the design. Moreover, the flow of information exchange between these domain-experts with various domain-knowledge within current workflows has been challenging, thus, deflecting optimised progression. In this context, Architecture, Computer Science, Engineering, and Mathematics disciplines can be connected to develop advanced computational design tools to combine implicit (e.g. aesthetical, cultural, or emotional) and explicit knowledge (e.g. functional, environmental, economic), bringing radical innovations. One such innovation can be brought by introducing a workflow that allows fast feedback already in the early stages of design.

Considering the overall workflow, many critical decisions can occur when a design is in its most rough form, namely a sketch (Mahoney, 2018). Sketching is used to initiate, develop, and communicate ideas while allowing the architectural designer to easily tap into his/her intuition to ideate and explore the solution space. In the early stages of design, sketching serves as a visual thinking and architectural form-finding process to the designer and, subsequently, as a visual reference for a computer-aided design (CAD) modelling expert. Moreover, the modelling procedure allows further communication with other downstream processes, such as structural analysis and computer-aided engineering (CAE) simulations (Eissen et al, 2008, Yu et al, 2021). However, the 3D modelling process is often too time-consuming, and simultaneously, the design intention might not be captured accurately and can be misinterpreted. Also, on the other hand, structural analysis is an essential part of the design, which studies and predicts the behaviour of structure's fitness subject to different loads, materials, etc, which is usually not

https://doi.org/10.7146/aul.455.c212

taken into account in the early stages of design. Therefore, reconciling these various domain experts can cause delays, conflicts, and undesirable design compromises.
In this setting, a workflow targeting the early stages of design can address two main challenges. First, the workflow should capture the design intention behind the various parts of the sketch and subsequently translate it into an appropriate (non-) parametric geometry format to be used in different softwares without requiring a modelling expert. Recent advances in Machine Learning (ML) have demonstrated an increasing ability regarding surface modelling and shape reconstruction from sketch data which can be taken into account for this purpose. Second, fast feedback provision with respect to the mechanical performance of the model is of great importance in unfolding the strength and weaknesses of the structure. Such feedback enables the architectural designer to modify his/her design accordingly in the early stages, where the design is still amenable to substantial improvements.
Building upon the above statements, this paper introduces a novel AEC workflow coupled with computational support, enabling the integration of digital design and structural analysis tools in the early stages of design aided by ML, resulting in quick iteration over the design cycle, increased design quality, and better supervision of structural implications of the design in the early stages. This workflow is based on a developed 4D sketching application with Unity Engine targeted for architectural design as a form-finding tool simulating traditional sketching behaviour with paper and pen but allowing the sketch creation in 3D space through tablet and stylus while capturing temporal data as of the fourth dimension. It is noteworthy that the sketch creation is guided by different geometric shapes that are used as canvases for stroke projection. The workflow starts by sketching an architectural element followed by passing the sketch to the *Shape Inference* module, which outputs the reconstructed surface mesh of the sketch. Hence, the reconstructed surface mesh is processed in the *Structural Analysis* module, whereby feedback based on the mechanical performance of the design is sent back to the sketching application. As a proof of concept, the most fundamental architectural element, namely a wall, is used throughout the pipeline to demonstrate the efficiency of the proposed workflow.
This paper is structured as follows. First, a brief overview of the state-of-the-art on novel AEC workflows targeting early stages of design and sketching applications taking the structural analysis part into account is given. Then, the proposed workflow, along with its two main modules, including the *Shape Inference* and *Structural Analysis* and come along- required pre-processing and post-processing steps are explained thoroughly. However, details of the sketching application and its main interaction techniques other than the type of data it provides is not explained as it is not in the scope of this paper. Finally, the current state of the workflow is discussed, and an overview of the future outlooks is given.

2. Related Works

Various researches have been targeting the early stages of design to automate and integrate modelling and performance simulations. In the context of 3D sketches, (Mahoney et al, 2018) presents the prototype of a 3D sketching system to enhance the design exploration in the early stages. The prototype is aided by machine learning which enables the translation of sketch into an intermediate description followed by a reconstruction function that translates this description into a 3D form. The reconstruction function and a set of libraries containing various geometric elements enable the designer to refine the solution space and produce different outputs from the same sketch. Another recent work is CASSIE (Yu et al, 2021), a conceptual modelling system leveraging freehand mid-air gestures coupled with a neatening framework producing a connected 3D curve network from the sketch. The curve network is subsequently surfaced, providing an output amenable for presentation, structural analysis, or manufacturing.

A few sketching systems integrating structural analysis have been introduced as well. However, they only work with 2D sketches and are developed for mainly educational purposes. The FEAsy (Murugappan et al. 2007), is a sketch-based environment for structural analysis in the early stages of design in which users can transform, simulate, and analyse their finite element models through freehand sketching within this environment. The tool is coupled with a beautification module, which simplifies and represents the sketch in a more meaningful way prior to the finite element analysis. STRAT (Peschel et al, 2008) is another tool for solving truss problems. Using the freehand sketch of the truss, this tool allows the designer to determine unknown forces in it with the aid of a sketch recognition system.
Many researches address supporting the architectural designer in the early stages of design throughout structural recommendations and performance simulations. (Ampanavos et al, 2021) trained a Convolutional Neural Network (CNN), which iteratively generates structural design solutions for the sketches of the plans accompanied by real-time guidance before formalisation into CAD software. Other works used ML as surrogate models to speed-up simulations that are time-consuming to be employed in the early stages of design. (Nie et al, 2020) uses a CNN for predicting stress fields in 2D linear elastic cantilevered structures. Successively, (Jiang et al, 2021) achieved a fast mechanical analysis by introducing a new network architecture called StressGAN for predicting 2D von Mises stress distributions in solid structures. However, in contrast to recent works in this area that either focus on a specific stage of the design flow such as modelling, or only work with certain types of (non-) parametric geometries in terms of dimension, this paper presents a workflow where automation of modelling and structural analysis happens within 3D space which is relatively more challenging. Simultaneously, connectivity between different stages of design is prioritised to reach a harmonious integration of sketching, modelling, and structural simulations.

3. Proposed Workflow: From Shape Inference to Structural Analysis Feedback

The proposed workflow begins with 3D sketching within the developed sketching application by an architectural designer. The sketch's 3D polylines and the associated temporal data are recorded throughout the sketching procedure. Afterwards, as the sketching finishes, the sketch's constituent 3D polylines are sent into the *Shape Inference* module for surface mesh inference. The employed reconstruction algorithm in this module digests point cloud data as input. Thus, the 3D polylines must be converted into a 3D point cloud in the pre-processing step. Subsequently, the *Shape Inference* module outputs the reconstructed surface mesh of the original sketch. Given that the surface mesh will be further processed in the *Structural Analysis* module, the outputted surface mesh must be post-processed to create an appropriate input format for the aforementioned module. As the obtained surface mesh is in 3D format, it must be transformed into a volume mesh prior to being sent to the *Structural Analysis* module. Furthermore, boundary conditions, loads, and material information are added to the volume mesh before the finite-element analysis (FEA) starts off. Ultimately, the output of this late module, including the displacements and stresses, is linked and sent back to the sketching application to be further visualised by the designer. The visualisation assists the architectural designer so he/she can improve the design in an acknowledged way regarding the structural implications of his/her design in the early stages. Depicted in the Figure 1, the general overview of the key steps in the proposed workflow can be seen.
In the subsequent sections, the surface reconstruction algorithm employed in the *Shape Inference* and its theoretical background are explained thoroughly. Additionally, the pre-processing and post-processing steps that come along with the *Shape Inference* module are discussed. Finally, the *Structural Analysis* module operations and linkage of its output back to the sketching application is described subsequently.

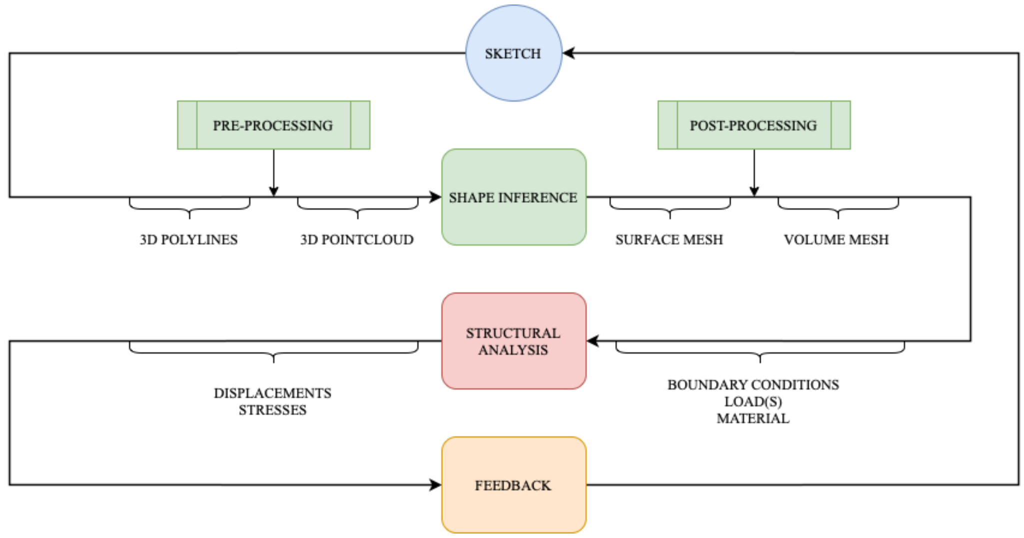


Figure 1: Overview of the proposed workflow containing two main modules allowing fast structural analysis feedback in the early stages of design.

3.1 Shape Inference

Once the sketching part is finished, the artist-drawn sketch must be converted into the intended 3D geometric format suitable for other downstream (non-) parametric and engineering applications. Extensively Artificial Intelligence (AI)-based surface mesh reconstruction algorithms have been developed for such a scenario, aiding 3D shape reconstruction from 2D and 3D sketches (Xu et al, 2014, Guillard et al, 2021). The 3D polylines drawn in the sketching application in our framework can be easily converted into a point cloud. Therefore, surface mesh reconstruction algorithms using point cloud data as input are straightforward and reasonable choices to embark on. To do so, a surface mesh reconstruction algorithm called Points2Surf (Erler et al, 2020) is leveraged to obtain a solid surface mesh of the sketched wall. In contrast to other Machine Learning (ML)-based surface reconstruction algorithms (Vakalopoulou et al, 2018, Park et al, 2019), Points2Surf is patch-based and independent from classes, leading to a better generalisation ability on unseen inputs, making it a reasonable option to get started.
In order to create a proper input format from the sketch data to be fed to this algorithm, specific steps need to be performed as pre-processing. In addition, a post-processing procedure must generate suitable mesh data for structural analysis. Each of these is described in more detail in the following.

Pre-Processing

In order to utilise the Points2Surf algorithm, firstly, the sketched wall must be converted into a point cloud from its constituent 3D polylines. To locate 3D points along the drawn 3D polylines, at every frame at which the Unity Engine's Update function gets called, the contact point of the ray originating from the camera intersecting the drawing surface is recorded and stored as a 3D point coordinate. Accordingly, a 3D point cloud of the whole sketch is obtained by merging the individual 3D polylines' point clouds into one (see Figure 2). Also, prior to feeding the point cloud into the Points2Surf network, as part of the pre-processing, the point cloud must be centred at the origin and scaled uniformly to fit within the unit cube. Additionally, within the developed sketching application, while sketching 3D polylines, the designer can set the width parameter for the brush based on his/her willingness, giving thickness to the sketched element. However, due to the nature of the point sampling strategy of the sketch, this thickness parameter is discarded. The absence of the explicit thickness in the point cloud creates a problem for the Points2Surf reconstruction algorithm since it is trained on solid objects with front and back sides. To solve this issue, normal vectors along the point cloud of the sketch are estimated, and points are moved along them to give the point cloud a slight implicit thickness. Normal estimation is done by finding adjacent points and calculating their principal axis using covariance analysis. The covariance analysis algorithm outputs two opposite directions as the normal candidates making them not consistently oriented across the point cloud. In that case, normals are aligned with respect to the tangent planes computed from the point cloud (Hoppe et al, 1992).

(a) (b)

Figure 2: (a) The sketch of a wall drawn in the sketching application, and (b) The corresponding obtained point cloud.

Points2Surf

As previously stated, the aim is to reconstruct a surface mesh from a 3D point cloud $P = \{p_1, p_2, \dots, p_N\}$. The authors of Points2Surf take zero-set of the Signed Distance Function (SDF) f_S as a suitable representation for surfaces for training a neural network:

$$S = L_0(f_S) = \{x \in R^3 \mid f_S(x) = 0\} \tag{1}$$

The approach taken in Points2Surf consists of the point cloud fed into a neural network with an encoder-decoder architecture, generating a latent vector and approximating the SDF through encoder and decoder, respectively:

$$f_S(x) \approx \widetilde{f_P}(x) = s(z),\ with\ z = e(P) \tag{2}$$

where z is the latent vector obtained through the encoder e from the point cloud P, and s is the decoder conditioned on the latent vector z. However, encoding a surface with a single latent vector affects the accuracy and generalisation ability of the network. Consequently, the authors propose factorising the SDF into two factors: absolute distance f_S^d and sign of the distance f_S^s, where each of them is estimated through separate encoders e^d and e^s, respectively. To estimate the absolute distance at a query point x, local neighbourhood p^d from the point cloud P is chosen and fed to encoder e^d. Furthermore, since the interior/exterior of the surface cannot be reliably determined from a local neighbourhood, encoder e^s is trained on a global subsample p^s computed from the point cloud P for every query point x to estimate the sign of the distance. Moreover, the authors found out that instead of having two separate decoders for each, sharing information between the two latent vectors z^d and z^s benefits the network, resulting in the following formulation:

$$\left(\widetilde{f_P^d}(x), \widetilde{f_P^s}(x)\right) = s(z^d, z^s),\ with\ z^d = e^d(p^d)\ and\ z_x^s = e^s(p^s) \tag{3}$$

where s is the decoder containing the z^d and z^s as its inputs, outputting the distance $\widetilde{f^d}$ and the sign of the distance $\widetilde{f^s}$. Afterwards, the surface S can be reconstructed by applying the Marching Cubes (Lorensen et al, 1987) to the estimated SDF $\widetilde{f^d} \times \widetilde{f^s}$. Figure 3 demonstrates the overview of Points2Surf architecture.

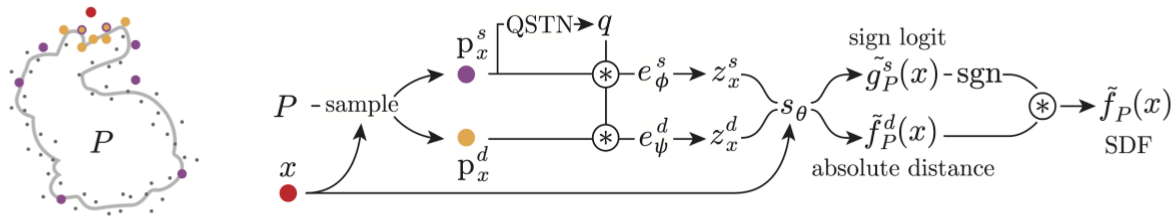


Figure 3: Points2Surf architecture. Image taken from (Erler et al. 2020).

The network architecture used for encoders e^d and e^s are the same as the PointNet (Qi et al, 2017), where a feature representation for each point is computed through a 5-layer Multi Layer Perceptron (MLP) neural network . Also, the decoder s consists of a 4-layer MLP that takes as input the concatenated feature vectors z^d and z^s and outputs the two aforementioned SDF factors. Assuming the ground-truth surfaces are available during the training for supervision, the above network is trained in an end-to-end manner regressing the distance and classifying the sign as positive or negative based on the interiority and exteriority, respectively. Two separate loss functions are used for the distance and the sign of the distance for the training procedure. Firstly, L_2-based regression is used for the distance:

$$L^d(x, P, S) = |tanh\left(\left|\widetilde{f^d}(x)\right|\right) - tanh\left(|d(x, S)|\right)|_2^2 \tag{4}$$

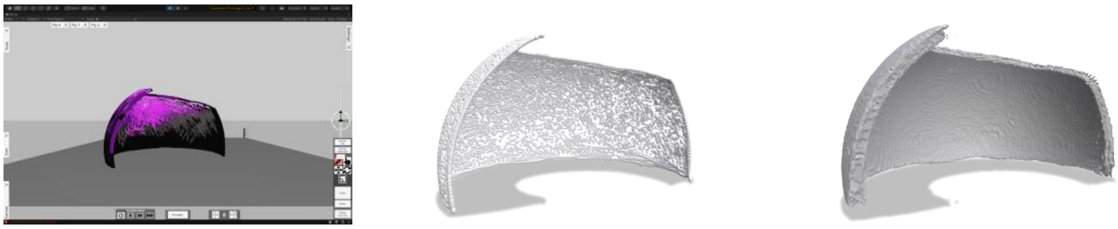
where $d(.,.)$ is the ground-truth distance between the query point x and surface S. Secondly, for sign of the distance classification, the binary cross-entropy loss H is used as follows:

$$L^s(x, P, S) = H\left(\sigma\left(\widetilde{f^s}(x)\right),\ [f_S(x) > 0]\right) \tag{5}$$

where σ is the logistic function converting the sign logits to probabilities, and $[f_S(x) > 0]$ is equal to 1 when x resides in the exterior of the surface and is equal to 0 otherwise. Altogether, the network is optimised with the following loss function compromising of the summation over these two losses for all shapes and query points of the training set:

$$\sum_{(P,S)\in S} \sum_{x \in X_S} L^d(x, P, S) + L^s(x, P, S) \tag{6}$$

The dataset that has been chosen to train the network on is the ABC dataset (Koch et al, 2019). This dataset includes a collection of one million CAD models. The authors of the Points2Surf have picked 4950 clean watertight meshes for training and 100 meshes for validation and test sets. Ultimately, for the inference on the point clouds of the sketches drawn in the developed sketching application, we have chosen and utilised the best pre-trained model based on the ablation results trained for 250 epochs. Depicted in Figure 4, a curved wall sketch's 3D polylines, its 3D point cloud representation, and the reconstructed surface mesh by the Points2Surf algorithm can be seen.



(a) (b) (c)

Figure 4: (a) The sketch of the curved wall drawn in the developed sketching application, (b) The point cloud of it, and (c) The reconstructed surface mesh.

Post-Processing

To translate the obtained surface mesh via Points2Surf into a format suitable for structural analysis, several post-processing steps must be employed, as discussed next. The reconstructed surface mesh may not be smooth enough and contain noise over its entirety, given that the density of the sampled points might differ in different parts of the sketch as the drawing speed varies. Due to the nature of the sampling strategy within the UnityEngine that is employed in the pre-processing step, very fast sketching of 3D polylines may leave empty areas in the 3D point cloud of them, and thus, the reconstructed surface mesh might be rugged. To solve this challenge, Laplacian smoothing (Vollmer et al, 1999) is employed to remove the noise and smooth the surface mesh. After smoothing, verifying the watertightness of the surface mesh is required. A watertight mesh can be defined as a mesh that is edge manifold, vertex manifold and not self-intersecting. A non-manifold triangle mesh is necessary to be able to carry out the mechanical structural analysis using finite element-based methods. Thus, the approach introduced in the ManifoldPlus (Huang et al, 2020) is adopted to convert the reconstructed surface mesh into a watertight one. ManifoldPlus extracts watertight manifolds from surface meshes using the exterior faces between the occupied and the empty voxels and a projection-based optimisation method. Subsequently, the reconstructed surface mesh is prepared for the application of suitable boundary conditions required for the structural analysis. Within this context, it is considered that the virtual ground that exists in the sketching application is the surface where boundary conditions are prescribed. Therefore, the surface mesh has been levelled at its bottom part, a feature that might not automatically exist due to the absence of domain knowledge encoded in the Points2Surf reconstruction method. To enforce planarity, the reconstructed surface mesh is sliced with a plane and capped subsequently, see Figure 5, for the reconstructed surface mesh and its corresponding smoothed variant with a planar bottom.

(a) (b)

(c)

(d)

Figure 5: The reconstructed surface mesh of a curved wall viewed from different angles (top-row) and its corresponding smoothed variant with a planar bottom (bottom-row).

Tetrahedralization

Finally, the smoothed, watertight surface mesh with a planar bottom is translated into an analysis-ready volume mesh using the TetWild (Hu et al, 2018) engine, a quite robust engine that does not require any user interaction. The quality of the resulting volume mesh is a direct function of the target mesh size, controllable by a tolerance parameter, denoting how much deviation from the initial surface mesh is permitted. Figure 6 shows the curved wall discretised with first-order linear tetrahedral elements obtained by the TetWild engine.

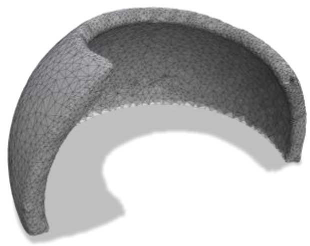


Figure 6: The volume mesh of the curved wall generated by the TetWild engine.

3.2 Structural Analysis

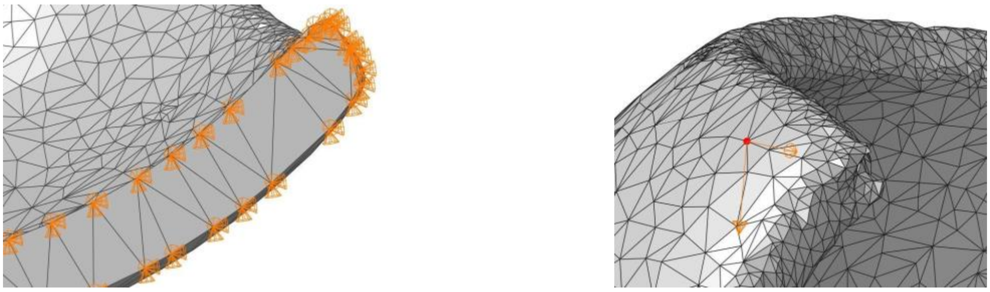
The finite element method is used to analyse the response of the sketched structure and assess its mechanical performance. Apart from the geometrical features, which are well defined by the post-processed volume mesh, the definition of material behaviour, boundary conditions and external loads are required together with the selection of an appropriate FE solution strategy.

Problem Definition

Regarding the definition of the material behaviour, we are currently working on implementing a material database in the sketching application as well as developing a micromechanics-based model to predict elastic properties for a wide range of bio-composites. In future, the user will have the possibility to select a broad range of different materials. So far, the materials used for our testing scenarios are described with isotropic, linear elastic constitutive models.

The material behaviour is thus fully defined by two parameters only, the Young’s modulus along with the Poisson’s ratio. The limitation to linear elastic material models is made to provide FE solutions almost in real-time and, thus, allowing for rapid feedback to the designer. Naturally, it is possible to extend the material database and the underlying models at any time, e.g. by taking direction-dependency and nonlinear material behaviour into account.

Displacement boundary conditions, as displayed in Figure 7(a), are automatically applied at the bottom surface, which is therefore made planar, as described in the previous section. In more detail, a fixed boundary is considered, where all displacement degrees of freedom in the 3D space (u_x, u_y, u_z) are restricted and therefore set to zero for every node. All other surfaces are considered free surfaces.
At last, external loads have to be defined. For this purpose, we focused on deadweight first, where only the density of the material in the database has to be additionally taken into account. Furthermore, in the first step, point loads are added at predefined positions to resemble selected loading states of interest, as illustrated in Figure 7(b). At this point, this is done manually within the FE software ABAQUS, for reasons to test the consistency and functionality of the proposed workflow. Subsequently, an automatic linking of the sketching tool and the FE program should take place via Python code. This allows the designer to define any loading and to mechanically pre-test his design already in the sketching application.

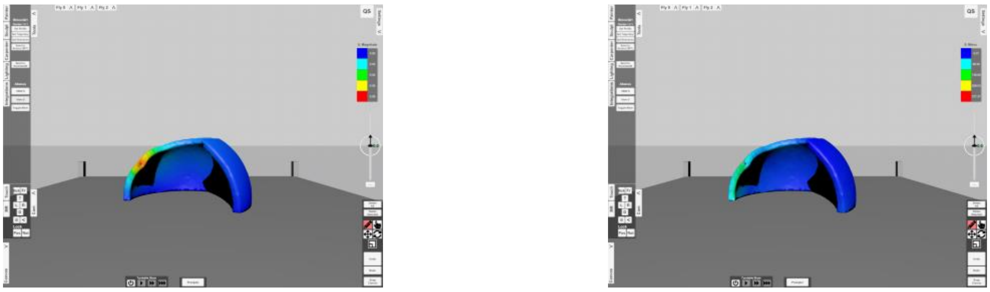


(a) (b)

Figure 7: (a) Visualisation of the displacement boundary conditions at the bottom surface and (b) placement of a concentrated load on a predefined position (node) on the wall. So far, for the calculations, the implicit ABAQUS/Standard solver is used. The computed output variables involve the 3D nodal displacements as well as 3D stress- and strain fields of the whole body, which can be transferred back to the sketching application.

Linking the Output Back to the Sketching Application

Computation results, such as the comparison between initial undeformed and deformed configuration (obtained by adding the computed displacements to the original nodal coordinates) are again visualised in the sketching application, as seen in Figure 8. For this purpose, the structure is converted back to a surface mesh. A heat map and a legend are also added to highlight the zones with large displacements. Other modelling results which can be visualised are stresses and strains in every direction, principal stresses, plastic strains (if plastic material behaviour is considered) or any other output fields of common FE software. Figure 8 shows the visualisation of the von Mises stresses – a common stress quantity governing yield for ductile materials such as steel.
This graphical illustration helps the designer to immediately assess the mechanical performance of the sketched structure. If certain displacements are particularly large, or stresses exceed the material strength, the designer can immediately react, e.g. by changing the geometry, the used material, or by adding additional support to the structure.



(a) (b)

Figure 8: (a) Deformed configuration with highlighted displacements as well as (b) Deformed configuration with visualised von Mises stresses.

4. Conclusion and Future Work

The early stages of design are mainly the process of exploration and ideation. It is also well understood that the decisions made at this stage have a significant impact down the line. However, lack of proper communication among disciplines as well as data and information losses between design stages makes it time-consuming and simultaneously results in a decreased design quality. As a step toward improving state-of-the-art workflows within AEC industry, this paper introduces and showcases a novel workflow automating the geometry creation followed by fast structural analysis feedback provision at the early stages of design where sketching is used for form-finding. Further improvements are envisioned at different parts of the proposed workflow. Primarily, regarding the *Shape Inference* module, a more sophisticated approach based on machine learning capturing design intention directly from 3D polylines and its coupled attributes such as timestamp, pressure, tilt, etc, generating parametric geometry is quite interesting and challenging to be further researched on and developed. Moreover, regarding the *Structural Analysis* module, just the boundary conditions are detected automatically and transferred to ABAQUS so far, while additional information such as material and loads can be recognised and carried to this module as well. In the future, we will be able to draw fixed boundaries, sketch loads and select materials from a database already within the sketching tool. Also, the integration of warnings shall be implemented, for instance, if displacements or stresses exceed certain critical values. It is also intended to integrate the *Shape Inference* and *Structural Analysis* modules and codes directly into the sketching application, to not rely on third-party programming interfaces and softwares, making it more of a unified tool. At last, the geometry can be exported to be further used in different CAD and computer graphics tools for tasks such as lighting simulations, paneling, and structural optimization.

Acknowledgements

The research is supported by grant F77 (SFB “Advanced Computational Design”, subprojects SP2 and SP8) of the Austrian Science Fund (FWF). Moreover, the authors thank the following project partners for providing valuable feedback:

- SFB, Subproject SP2, Integrating AEC Domain Knowledge - Synthesis 2.0, Peter Ferschin, Ingrid Erb, and Balint Istvan Kovacs
- SFB, Subproject SP8, Markus Lukacevic, and Markus Königsberger.

References

Ampanavos, S., Nourbakhsh, M. and Cheng, C.Y., (2021). Structural Design Recommendations in the Early Design Phase using Machine Learning. *arXiv preprint arXiv:2107.08567*.

Eissen, J.J. and Steur, G.N.R.E., (2008). Sketching, drawing techniques for product designers. Ye-Eun Publishing Co.

Erler, P., Guerrero, P., Ohrhallinger, S., Mitra, N.J. and Wimmer, M., (2020), August. Points2surf learning implicit surfaces from point clouds. In *European Conference on Computer Vision* (pp. 108-124). Springer, Cham.

Guillard, B., Remelli, E., Yvernay, P. and Fua, P., (2021). Sketch2Mesh: Reconstructing and Editing 3D Shapes from Sketches. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 13023-13032).

Hoppe, H., DeRose, T., Duchamp, T., McDonald, J. and Stuetzle, W., (1992), July. Surface reconstruction from unorganized points. In *Proceedings of the 19th annual conference on computer graphics and interactive techniques* (pp. 71-78).

Hu, Y., Zhou, Q., Gao, X., Jacobson, A., Zorin, D. and Panozzo, D., (2018). Tetrahedral meshing in the wild. *ACM Trans. Graph.*, *37*(4), pp.60-1.

Huang, J., Zhou, Y. and Guibas, L., (2020). Manifoldplus: A robust and scalable watertight manifold surface generation method for triangle soups. *arXiv preprint arXiv:2005.11621*.

Jiang, H., Nie, Z., Yeo, R., Farimani, A.B. and Kara, L.B.,(2021). Stressgan: A generative deep learning model for two-dimensional stress distribution prediction. *Journal of Applied Mechanics*, *88*(5).

Koch, S., Matveev, A., Jiang, Z., Williams, F., Artemov, A., Burnaev, E., Alexa, M., Zorin, D. and Panozzo, D., (2019). Abc: A big cad model dataset for geometric deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 9601-9611).

Lorensen, W.E. and Cline, H.E., (1987). Marching cubes: A high resolution 3D surface construction algorithm. *ACM siggraph computer graphics*, *21*(4), pp.163-169.

Mahoney, J.M., (2018). THE V-SKETCH SYSTEM, MACHINE ASSISTED DESIGN EXPLORATION IN VIRTUAL REALITY.

Murugappan, S. and Ramani, K., (2009), January. Feasy: a sketch-based interface integrating structural analysis in early design. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (Vol. 48999, pp. 743-752).

Nie, Z., Jiang, H. and Kara, L.B., (2020). Stress field prediction in cantilevered structures using convolutional neural networks. *Journal of Computing and Information Science in Engineering*, *20*(1), p.011002.

Park, J.J., Florence, P., Straub, J., Newcombe, R. and Lovegrove, S., (2019). Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 165-174).

Peschel, J.M. and Hammond, T.A., (2008), September. STRAT: a Sketched-truss Recognition and Analysis Tool. In *DMS* (pp. 282-287).

Qi, C.R., Su, H., Mo, K. and Guibas, L.J., (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 652-660).

Vakalopoulou, M., Chassagnon, G., Bus, N., Marini, R., Zacharaki, E.I., Revel, M.P. and Paragios, N., (2018), September. Atlasnet: Multi-atlas non-linear deep networks for medical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 658-666). Springer, Cham.

Vollmer, J., Mencl, R. and Mueller, H., (1999), September. Improved laplacian smoothing of noisy surface meshes. In *Computer graphics forum* (Vol. 18, No. 3, pp. 131-138). Oxford, UK and Boston, USA: Blackwell Publishers Ltd.

Xu, B., Chang, W., Sheffer, A., Bousseau, A., McCrae, J. and Singh, K., (2014). True2Form: 3D curve networks from 2D sketches via selective regularization. *ACM Transactions on Graphics, 33*(4).

Yu, E., Arora, R., Stanko, T., Bærentzen, J.A., Singh, K. and Bousseau, A., (2021), May. Cassie: Curve and surface sketching in immersive environments. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (pp. 1-14).