Proceedings of the IASS 2024 Symposium
Redefining the Art of Structural Design
August 26-30, 2024, Zurich, Switzerland
Philippe Block, Giulia Boller, Catherine DeWolf,
Jacqueline Pauli, Walter Kaufmann (eds.)

A Volumetric Finite Element Method Software for Analyzing Joints in Spatial Structures

Marcin LUCZKOWSKI[*], Sverre Magnus HAAKONSEN, Lars Olav TOPPE[a], Vegard ØYRE[b]

[*] NTNU (Norges teknisk-naturvitenskapelige universitet)
Richard Birkelands vei 1A, 7034 Trondheim
marcin.luczkowski@ntnu.no

[a] Norconsult ASA
[b] Aas-Jakobsen Trondheim AS

Abstract

This paper presents a novel volumetric finite element method (FEM) software for analyzing joints in spatial structures. The software is designed to provide an efficient and accurate analysis of complex joint structures, which are commonly found in large-scale spatial structures such as gridshells and shells. The FEM software utilizes a three-dimensional meshing technique that enables the modelling of complex geometries and the accurate simulation of joint behaviour under various loading conditions. The software's capabilities are demonstrated through several case studies, including the analysis of a cable-stayed bridge joint and a stadium roof joint. The paper describes the FEM software's key features, including its great implementation into a parametric modelling environment (Rhino/Grasshopper), material modelling, and load application capabilities. The software's accuracy is validated through a comparison with alternative FEM software, demonstrating its capability to predict joint behaviour accurately. The software's efficiency is demonstrated through its ability to analyze timber spatial structures (also gridshell) joints efficiently and accurately, thus reducing the analysis time and costs significantly.

Keywords: Finite Element Method, Finite Element Analysis, Parametric Modelling, Algorithm Aided Design, Joint Design, Knowledge Based Design

1. Introduction

The architecture, engineering and construction (AEC) sectors face increasing demands for designing and executing complex spatial structures. Particularly challenging are the requirements for sophisticated connections capable of accommodating significant forces while also being easily assembled and disassembled to promote sustainable building practices [1]. This urgency stems from a growing emphasis on sustainability within general building systems, underscoring the necessity for connections that are not only efficient to assemble but also environmentally considerate [2].

A pivotal concern in the industry has been the reliance on welding for creating structural steel connections, a method now recognized for its limitations, especially in terms of disassembly and environmental impact. Furthermore, in specialized truss systems, the connectors must withstand exceptionally high forces, making their accurate calculation a cornerstone of structural design and analysis. Alternatively, additive manufacturing has started to play a bigger and more significant role in today's AEC market [3]. Although printing in steel (or other metals) is still a unique technology, the glueing or customization of metal casting is becoming more economically rational. From a design perspective, we will need a better or just different digital workflow [4]. A workflow is not oriented on fabrication limitation but more on structural performance and architectural appearance.

The introduction of parametric modelling and generative design, facilitated by tools such as Grasshopper and Dynamo, has changed architectural and structural engineering practices. These methodologies, alongside the integration of Building Information Modeling (BIM), signal a shift towards algorithmically informed design processes. The natural next stage would be to increase the implementation of generative design. When most of the articles and research focus in this field purely on geometrical (more architectural) and BIM data (managment/contstruction) we would like to go into the field of generative elements which can mimic the structural behaviour. Till now, we have several great tools already implemented in the AAD platforms, such as Karamba3D [5] or IdeaStatica [6]. They can analyze the elements and populate the information from the analysis into the object.

Although existing Finite Element Methods (FEM) software can work in AAD platforms, we see that we have a gap in analyzing elements with specific geometry. Karamaba3D focuses only on beam and shell objects; IdeaStatica, for example, is only for standardized steel connections. When we design connectors for additive manufacturing or just connections with not trivial geometry, we very often have to go to volumetric FEM analysis. Software such as ABAQUS [7] gives us the opportunity to test geometry and material specifications freely. Unfortunately, transferring geometry and data from the AAD platform to the FEM solver was very often the bottleneck for the implementation of such a solution in the concept stage of design.

Addressing these challenges, our research focuses on developing and applying various design methodologies in structural engineering and architecture in the concept stage of the design process. We leverage finite element method (FEM) plugins for parametric modelling environments, enabling real-time volumetric analysis of complex geometries. This approach allows for a nuanced examination of stresses and geometrical dependencies, offering insights into structural vulnerabilities not readily apparent with traditional calculation methods. Our contribution is two plugins for a parametric modelling environment (Rhino/Grasshopper) that facilitates real-time volumetric finite element analysis. The first plugin is native, built with C# directly in Grasshopper [8]. The second one uses pre- and post-processing in Grasshopper, but the main solver uses the FEniCSx [9] engine on a Linux system. Although still in development, those tools have demonstrated their potential through the successful analysis of complex case studies, which will be discussed in subsequent chapters. This initiative represents a step forward in our quest to refine the methodology for designing spatial structures, ultimately enhancing both their sustainability and structural integrity.

2. Methodology

Development Environment and Language Choice

For the development of our plugin, we opted for the C# programming language due to its robust features and seamless integration capabilities within the Grasshopper environment. A primary goal during development was to prioritize calculation speed and performance. It was imperative that our plugin could process a vast number of finite elements efficiently to deliver results promptly, catering especially to the dynamic needs of conceptual stage design discussions involving architects and other stakeholders. You can find the source code for the Grasshopper plugin in the GitHub repository. Explanation of the code, algorithms and some benchmarks can be found in two theses written in 2023[10] and 2022 [11]. The code for algorithms working with FEniCSx is not open source, but a good explanation can be found in the thesis written in 2023 [10].

Speed and Performance

Speed was paramount, given the plugin's intended use at the conceptual design phase. This phase often requires rapid feedback during meetings with architects and design teams, where structural engineers are expected to provide immediate structural analysis results. Our approach was to balance the generalization of calculations with the need for delivering robust structural proposals, ensuring that designs developed at this stage could be refined and executed in subsequent design phases. For both cases, with the solver inside Grasshopper and with the solver using FEniCSx, we measure data about time. We also test the performance of the exchange data process. We want to know how fast you can design geometry and get clear information about stresses and deformation.

Parametric Modeling and Output Documentation

The plugin was designed to accept parametric modeling inputs, directly interfacing with Grasshopper to facilitate user interaction. Beyond mere calculation, the plugin needed to present FEM calculation results. Such data as material description, load, and support were added to the analysis using specific plugin components. In the first plugin, software architecture was assumed to make every step of the analysis in the Grasshopper. The brief software architecture and design workflow are presented in Figure 1.

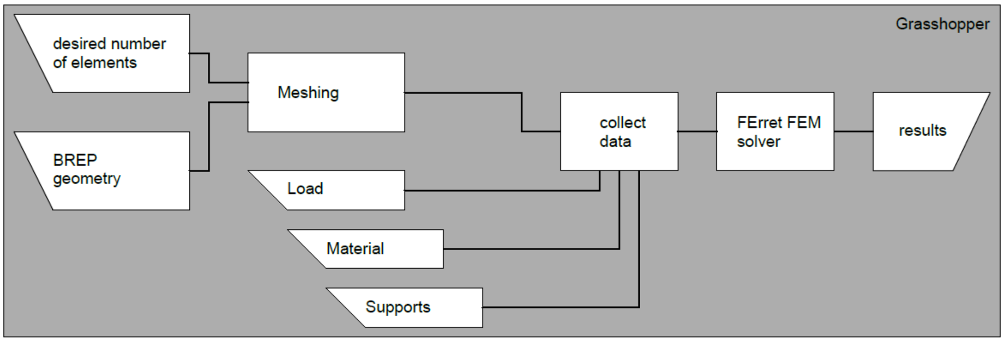

Figure 1: The graph represents the digital workflow for structural analysis with the FErret plugin.

In the second type of plugin, we decided to export the analysis to the FEniCSx solver. This solver runs under a Linux system, which demands that we create importers and exporters for the data. The software architecture and workflow are presented in Figure 2. In both scenarios, the user could read the results in Grasshopper and didn't have to manually open any other software. The connection with FEniCSx was done as a background process while using the plugin.

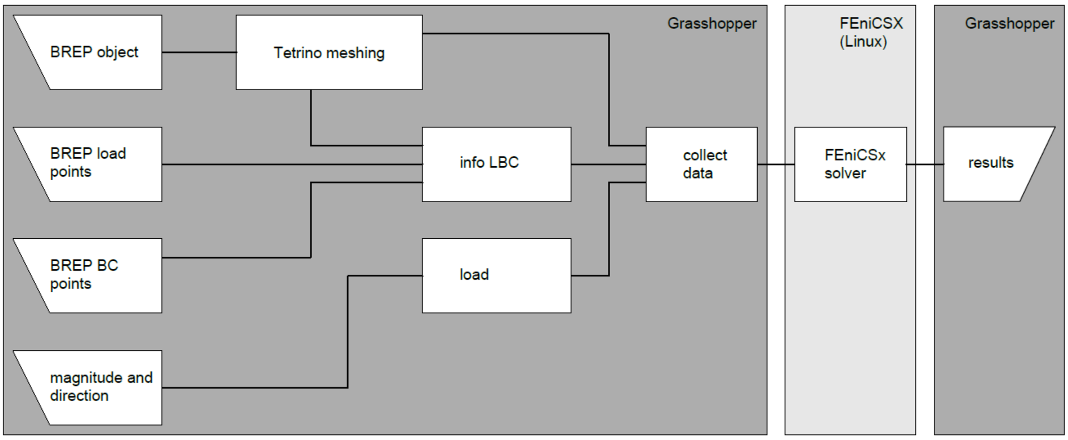

Figure 2: The graph represents the digital workflow for structural analysis with the FEniCSx software.

Finite Element Method Distribution and Mesh Data

To perform finite element distribution, the plugin utilizes mesh data types. These meshes must be constructed with precision, adhering to specific detailed requirements, such as the necessity for each mesh to have either eight vertices for hexahedral elements or four vertices for tetrahedral elements. Both plugins are able to handle classic mesh object from Rhino API. Those meshes can be done manually by the user or generated by other plugins. In figure 3, you can find a mesh preview of the case study analyzed with the FEniCSx solver.

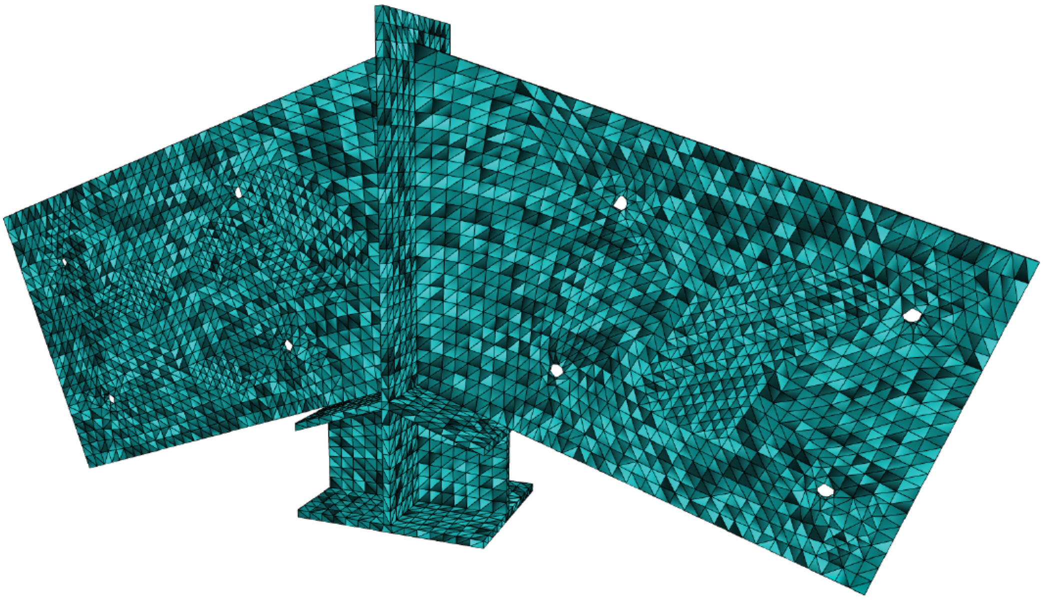

Figure 3: Geometry with mesh for the case study solved with FEniCSx solver.

Material Description

Initially, our plugin supported isotropic, homogeneous materials, with ongoing expansions to include orthotropic materials like timber. However, increasing material complexity and introducing nonlinear elements significantly impact solver performance, necessitating trade-offs between accuracy and computational speed. Although both plugins are able now to handle orthotropic material, in this article, we will focus only on isotropic materials with properties similar to steel S355.

3. Case study

The first case study is using the plugin using a solver done only with C# and fully integrated into the Grasshopper. The main question here is the size of the model that can be analyzed. We tested different mesh sizes to investigate when the time lag of the calculation will not be acceptable for so-called ”real-time” analysis.

For creating the plugin, we built our own classes, but such inputs as loads were, for example, controlled by the Vector3d class (Rhino class) and Point3d (Rhino class). Thanks to integrating our solver in the Rhino/Grasshopper environment, the slowest algorithm was easy to predict. Inverting the global stiffness matrix took 98% of all calculation time. Our plugin uses MathNet numerics C# library for algebra operations. The calculation time in this situation depends mostly on the number of degrees of freedom. Table number 1 presents the relation between a number of elements, degrees of freedom and time.

Table 1: The performance of plugin (integrated in grasshopper) calculation

Number of elements	288	10296	23520
Number of DOFs	189	60012	212070
Total time[s]	0.089	40.0	199.1

The second goal of the first case study was to measure the quality of the analysis. We compare the same model analysis with the Abaqus analysis. We used the same tetrahedron finite elements with linear shape functions. The relative error changes and decreases with the increasing number of elements, from 35% for 315 elements to 6% for 23 918 elements. Figure 4 presents the case study 1 with the deformation map on it.

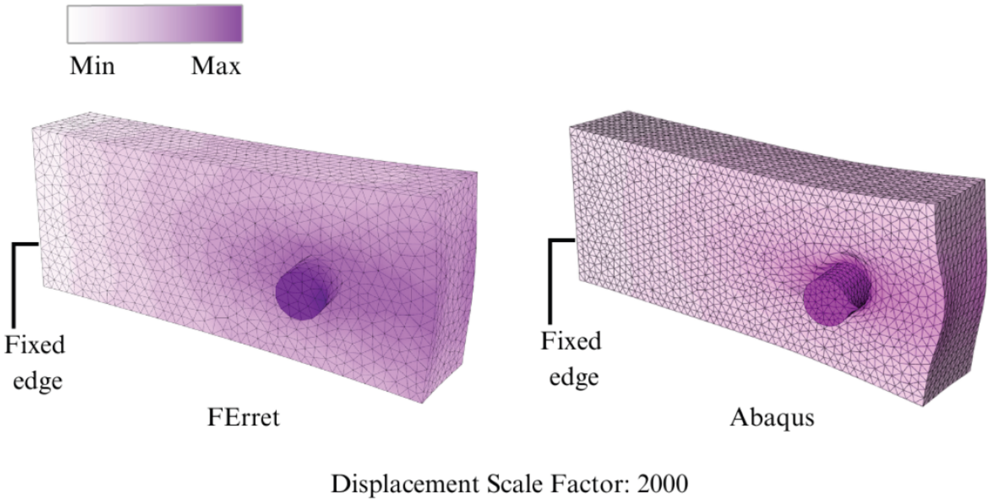

Figure 4: Geometry with deformation map of the case study 1, solved with FErret.

From study case 1, we clearly see that, unfortunately, for the bigger models (bigger meshes, more elements), calculation time crossed several seconds, which, in our opinion, could be understood as ”real-time.” Also, our plugin needs to converge faster to correct results. On the other side, one of the applied solutions, which was developing hexahedral finite elements, revealed insufficient tools for correct meshing. In this situation, we start developing the possibility to outsource the main FEM algorithms to another platform. Project FEniCSx is a great set of FEM tools, but it works on different platforms, so the big question was how to solve the transfer of the data. Case study 2 shows a solution to this question.

In our research, we aim to demonstrate the efficacy of a robust method for calculating complex and practical structural connections. To achieve this, we selected a connection type that balances practical applicability and geometric complexity: the slotted steel plate and dowel combination commonly found in timber construction worldwide. This is why we decided to use such a common but not so geometrically trivial connection in case study 2 (see Figure 5).

The calculation times presented in Table 2 clearly show that despite outsourcing calculation to another platform, we were able to solve a linear elastic task with isotropic material for 37 627 elements (212 070 degrees of freedom) in less than 20 seconds. The computation time was about 15 seconds, and the rest was data transfer. The relative error between Abaqus and FEniCSx didn’t cross 5% in all mesh cases.

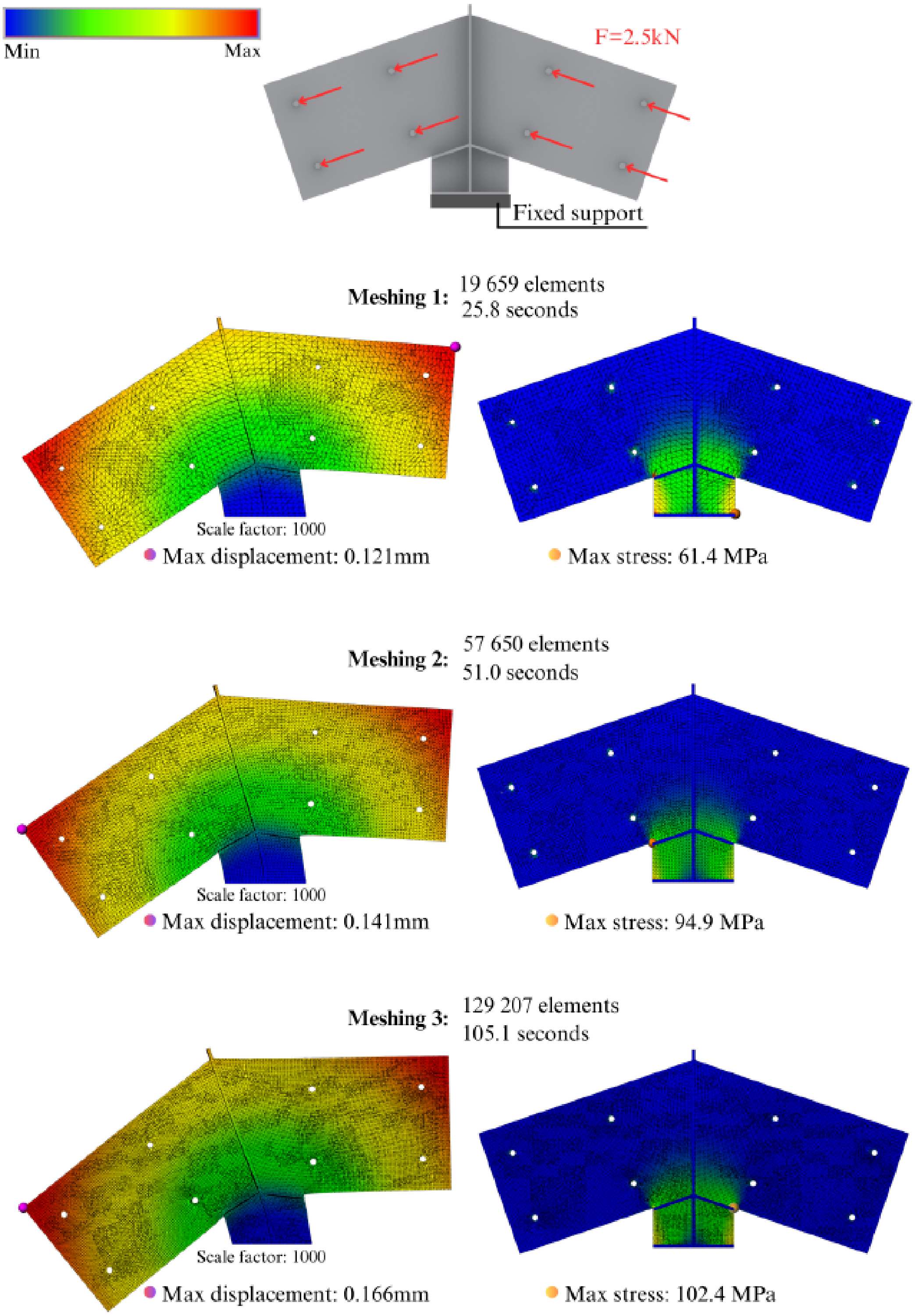

Figure 5: The results for case study 2, solved with the FEniCSx solver.

Table 2: The performance of FEniCSx calculation

Number of elements	18	10361	37627
Number of DOFs	189	60012	212070
Total time[s]	2.20	4.40	17.3
FEniCSx-computation time[s]	0.897	4.00	15.7
Rebuild mesh in FEniCSx time[s]	0.00203	0.0198	0.102
Assemble model time[s]	0.000825	0.194	0.758
Solver time[s]	0.00242	2.54	11.4
Total time in FEniCSx[s]	0.110	3.19	13.8
Write and read time[s]	0.134	0.286	1.11

4. Discussion and conclusion

We conducted comparative analyses, including displacement error assessments for simple test cases. Results indicated that while our system demonstrated small errors in displacement, it also revealed areas for improvement, particularly in computational speed. Even with smaller meshes, our system exhibited longer processing times than desired, prompting us to explore alternative solutions.

Our findings revealed that while FErret showed promising accuracy, its computational time was a limiting factor, especially for larger meshes. Integration with the FEniCSx solver provided results comparable to Abaqus's while significantly reducing computational time. This integration marked a significant advancement in our plugin's performance, demonstrating its potential for practical application in structural engineering projects.

This project is a part of ongoing research about better integration between structural and architectural analysis. We strongly believe that with developments in computational power and artificial intelligence, the volumetric analysis of objects can give us a better understanding of the structural behaviour of building elements.

For future work, it is already planned to develop better 3D meshing techniques, which could lead to the implementation of more sophisticated types of finite elements. Also, the implementation of a better numerical tools for inverting stiffness matrices for FErret is planned as the next step.

References

[1] I. C. Secretary, "Sustainability in buildings and civil engineering works — design for disassembly and adaptability — principles, requirements and guidance," International Organization for Standardization, Tech. Rep., 2020.

[2] M. Cabaleiro, B. Conde, C. González-Gaya, and B. Barros, "Removable, reconfigurable, and sustainable steel structures: A state-of-the-art review of clamp-based steel connections," *Sustainability*, vol. 15, no. 10, 2023, ISSN: 2071-1050. DOI: `10.3390/su15107808`. [Online]. Available: `https://www.mdpi.com/2071-1050/15/10/7808`.

[3] N. Labonnote, A. Rønnquist, B. Manum, and P. Rüther, "Additive construction: State-of-the-art, challenges and opportunities," *Automation in Construction*, vol. 72, pp. 347–366, 2016, ISSN: 0926-5805. DOI: `https://doi.org/10.1016/j.autcon.2016.08.026`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0926580516301790`.

[4] S. Aziz, J.-S. Kim, D. Stephan, and C. Gengnagel, "Generative structural design: A cross-platform design and optimization workflow for additive manufacturing," in *Third RILEM International Conference on Concrete and Digital Fabrication*, R. Buswell, A. Blanco, S. Cavalaro, and P. Kinnell, Eds., Cham: Springer International Publishing, 2022, pp. 357–363, ISBN: 978-3-031-06116-5.

[5] C. Preisinger, "Linking structure and parametric geometry," *Architectural Design*, vol. 83, no. 2, pp. 110–113, 2013.

[6] IDEA StatiCa s.r.o., *Idea statica - software for structural engineers*, Accessed: 2024-04-10, 2024. [Online]. Available: `https://www.ideastatica.com/`.

[7] Dassault Systèmes, *Abaqus - product page*, Accessed: 2024-04-10, 2024. [Online]. Available: `https://www.3ds.com/products/simulia/abaqus`.

[8] S. M. Haakonsen, S. H. Dyvik, M. Luczkowski, and A. Rønnquist, "A grasshopper plugin for finite element analysis with solid elements and its application on gridshell nodes," *Applied Sciences*, vol. 12, no. 12, 2022, ISSN: 2076-3417. DOI: `10.3390/app12126037`. [Online]. Available: `https://www.mdpi.com/2076-3417/12/12/6037`.

[9] *Fenics*, `https://fenicsproject.org/`, Accessed: 2024-04-14.

[10] L. O. Toppe and V. Øyre, *Integrating finite element method with solid elements in algorithms-aided design for the conceptual phase of projects*, 2023. [Online]. Available: `https://hdl.handle.net/11250/3090917`.

[11] B. L. Aakre, I. F. Bødal, and A. H. Elmer, *Exploring finite element analysis with higher-order solid elements in algorithms aided design and optimization with help of machine learning*, 2022. [Online]. Available: `https://hdl.handle.net/11250/3024149`.