BIM2FEM: From Building Information Modelling to Finite Element Analysis – An Automated Open Source-Based Workflow

Julius Emig – Fraunhofer Italia Research, Bolzano, Italy – julius.emig@fraunhofer.it
Dietmar Siegele – Fraunhofer Italia Research, Bolzano, Italy – dietmar.siegele@fraunhofer.it
Dominik T. Matt – Fraunhofer Italia Research & Free University of Bozen-Bolzano, Bolzano, Italy – dominik.matt@fraunhofer.it

Abstract

We propose an open-source-based workflow which connects Building Information Modelling (BIM) with thermal Finite Element (FE) – Analysis. We use the open IFC-standard for data interoperability and leverage open-source FE software packages for both 2D and 3D thermal analysis. The Finite Element Method (FEM) represents a highly flexible state-of-the-art approach for thermal analysis in construction engineering. With the recent increase in computing power, even complex 3D FE models can be analysed within a feasible amount of time. However, the integration of FE-Analysis into BIM-workflows remains an active area of research. Especially, a consistent and automated flow of material and boundary condition information is a challenging task to realize. The aim of this work is to contribute to the advancement of automated and open-source-based solutions for thermal analysis of buildings. The proposed workflow has the potential to decrease the time needed for evaluation of energy efficient building designs, especially in early design-phases. Its open-source nature promotes transparency, reproducibility, and collaboration in the building industry. The implementation of the proposed workflow results in a software prototype, which is tested based on a selected use case.

1. Introduction

Data interoperability is still one of the main issues in the construction industry today. Many data exist already in today's building information models, but they are in many cases not used for tasks that go beyond the pure design of a building or a built structure. Especially the information about the structure of the building envelope like wall stratigraphies or window details with material information are of high value for building performance analysis. Often missing interfaces and tedious manual input of data – that theoretically already exists in building information models – hinder the efficient analysis of building performances.

The Finite Element Method (FEM) is a numerical technique widely used in engineering and mathematical modelling to solve complex physical problems. The FEM is highly suitable for Building Energy Performance Simulation (BEPS), due to its ability to model complex geometries, provide high spatial resolution with low computational times, and adapt to specific accuracy requirements.

It could be shown by multiple recent studies that the integration of FEM into Building Information Modelling (BIM) workflows carries a high potential. Jia et al. (2022) present an innovative approach to automate the conversion from BIM to FE models, leveraging Industry Foundation Classes (IFC) and ontology systems. Their methodology aims to enhance the efficiency and quality of data conversion and information transfer, critical aspects in the structural design phase. Leonardi et al. (2024) demonstrate how open-BIM can be used to facilitated structural analysis of historic masonry structures. Fedorik et al. (2016) explore the integration of BIM and FE analysis to automate engineering design processes for buildings and bridges. The integration of BIM with FEM tools was shown to facilitate the efficient transfer of data, reducing manual inputs and errors, and ultimately improving the quality of engineering designs.

Furthermore, in the field of BEPS the integration of BIM and the exploitation of BIM data for more efficient simulations is a subject of current research. Andriamamonjy et al. (2019) conducted a combined

Part of
Pernigotto, G., Ballarini, I., Patuzzi, F., Prada, A., Corrado, V., & Gasparella, A. (Eds.). 2025. Building simulation applications BSA 2024. bu,press.
https://doi.org/10.13124/9788860462022

scientometric and literature review to understand BIM research trends and its integration with BEPS. They identified a lack of established strategies for interoperability between BIM and BEPS, noting the potential for improvement in building system and control modelling during the operational phase.

Ahmad Mohammad Ahmad et al. (2022) stress the importance of a coherent life cycle information flow within BIM projects for effective energy analysis, developed through literature review and expert interviews. Li et al. (2020) explore BIM-based energy simulation for building operations, specifically addressing interoperability issues between BIM tools and energy simulation software, proposing a technical framework for information transfer to support accurate energy simulations.

Integrating BIM with energy modelling tools, as shown in the works of Carvalho et al. (2021) and Pezeshki et al. (2019), provides valuable opportunities to improve building sustainability and energy efficiency from the early design stages.

The automation of data conversion and information transfer processes, as explored by Jia et al. (2022), represents a critical area for advancement, offering the potential to significantly enhance the efficiency and accuracy of both structural analysis and energy modelling.

Ou et al. (2017) leverage BIM models specifically for thermal FE Analysis to evaluate the thermal building performance. The authors demonstrated the transfer of an IFC model to a commercial FEA software and highlight the capabilities of FE models to accurately represent geometric information from original IFC model. However, one drawback of their approach is the limitation to a specific commercial FE software.

The integration of BIM with FEM and BEPS is a dynamic and evolving field, offering significant potential to advance building design, structural analysis, and energy efficiency. While challenges remain, particularly in interoperability and data exchange, ongoing research and development in this area hold promise for creating more sustainable, energy-efficient buildings.

By proposing an automated open source-based workflow that enables a consistent flow of geometrical and non-geometrical data between BIM and FE analysis we want to address interoperability issues and increase automation for BEPS. A high degree of automation can reduce manual input errors and especially for large building models it has the advantage of reducing the amount of tedious manual input work.

This paper is structured as follows. In the following Section 2 we introduce our developed workflow from a methodical point of view. In Section 3 we demonstrate its application through a selected case-study. In Section 4 we discuss the performance and possible drawbacks that were found conducting the case-study. In Section 5 we draw conclusions and give a brief outlook for possible directions of further research.

2. Methodology

The objective of this study is to create an automated workflow which takes a digital building model as input and returns results of a thermal FE simulation as output. These results can then be used to calculate a variety of different metrics to indicate the performance and necessary dimensioning installations of the corresponding building.

The overall workflow to achieve this is shown in Fig. 1. It consists of three primary steps:

1. IFC-Processing (Section 2.1)
2. Geometry discretization (Section 2.2)
3. Thermal FE simulation (Section 2.3)

We implemented all three steps in the Python programming language and made use of free and open-source Python libraries.

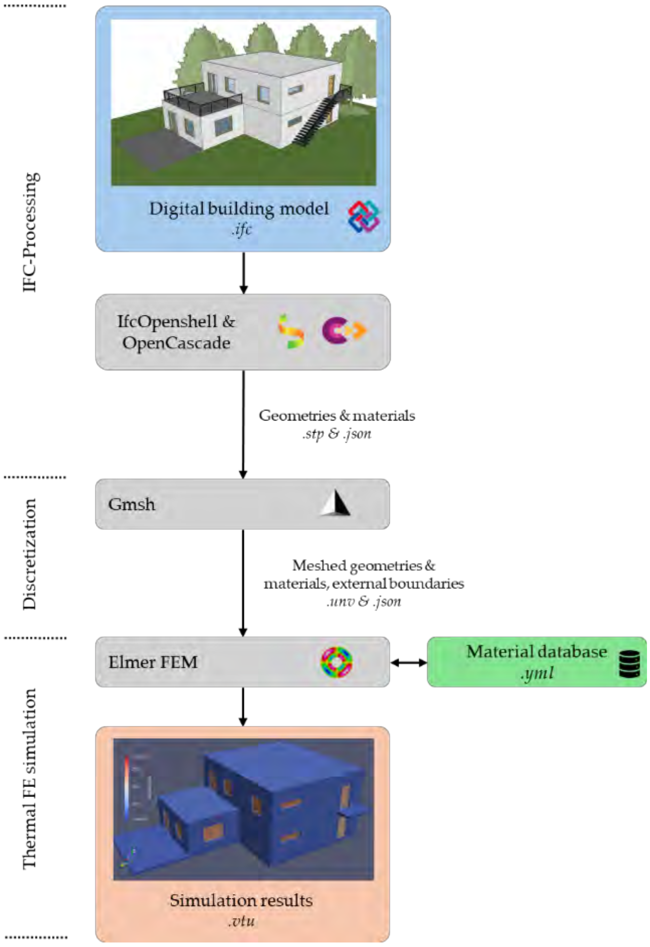

Fig. 1 – Overall workflow with used software, libraries and file formats

2.1 Processing of IFC Model

The starting point of our workflow is an IFC-file containing a digital building model. The IFC file format is an open standard and widely used for exchange of construction data. Hence, every common building design and engineering software like Autodesk Revit, Nemetschek Allplan or Graphisoft Archicad support nowadays the export of models in IFC format.

We leveraged the capabilities of IfcOpenShell (Krijnen, 2023) and PythonOCC (Paviot, 2023) libraries to develop a Python module for the efficient processing and manipulation of 3D geometries derived from IFC models. This module encompasses two primary functions:

Firstly, the function designated for extracting geometrical representations selectively filters and refines 3D objects from the IFC model and extracts material information. The output is a curated Python dictionary with 3D geometries, each associated with specific object and material names.

Objects of type *IfcSpace* play a special role, as heat sources will be placed into them at later stages of the thermal simulation. To identify these elements later, their names are prefixed with 'Air_body' followed by a sequential number.

Secondly, we implemented a function aimed at exporting the processed geometries into a STEP file format. The STEP format is extensively recognized for its compatibility with various Computer-Aided Design (CAD) and meshing software for FE analysis. All geometries apart from the *IfcSpace* objects carry their original name as in the input IFC model. Due to limitations regarding STEP file importing in the consequent discretization step (see Section 2.2) the material information of the geometric objects is stored in a separate JSON-file and not directly in the STEP-file itself. The JSON-file contains a mapping between the geometry names and their material names.

2.2 Discretization

For the execution of FE simulations, it is imperative to discretize geometries into finite elements, a process commonly referred to as meshing. One commonly used free and open-source meshing software is Gmsh (Geuzaine & Remacle, 2009). Gmsh encompasses a comprehensive set of robust meshing algorithms suitable for both 2D and 3D discretization. In addition, it is equipped with a Python programming interface, thus enabling access to all functionalities via Python code. Additionally, Gmsh supports processing the STEP file format as an input.

In the context of this study, we have developed a Python module that harnesses the meshing capabilities of Gmsh. We have also devised an algorithm that automatically identifies all external boundaries of the processed geometrical model. The algorithm iterates through entities of the geometric model, which are one dimension lower than the model itself (surfaces in 3D models or lines in 2D models).

For each entity, it examines the adjacent higher-dimensional entities. If an entity is adjacent to only one higher-dimensional entity (e.g., a surface adjacent to only one solid in 3D), it is considered part of an external boundary. This is based on the premise that external boundaries will not be shared by multiple higher-dimensional neighbours.

Gmsh offers several configurable parameters to tailor the meshing process. The parameters of paramount importance include the meshing algorithm, the size of the resulting mesh elements, and the mesh output file format. For the sake of file format compatibility with Elmer FEM, the software employed for thermal FE simulation (refer to Section 2.3), we have opted for the UNV as mesh output file format. The remaining parameters are designed to be adaptable, allowing for customization based on specific use-case scenarios. Detailed discussions on the parameters selected for the case-study examined within this study are provided in Section 3.

In this step, our objective is still to maintain the material information preserved during the IFC processing (see Section 2.1), while also incorporating details about the external boundaries. The UNV file format is limited to representing only the geometrical aspects of the stored mesh, lacking the capability to include metadata such as materials, object names, or tags identifying external boundaries. To overcome this limitation and ensure the retention of this essential metadata, we once again employ a JSON file. This file facilitates a linkage between the 3D mesh geometry IDs and their respective materials, in addition to enumerating the IDs of any detected external boundaries. Both the 3D geometry IDs and the boundary IDs are cross-referenced in the UNV file.

2.3 Thermal FE Simulation

Leveraging the discretized geometry model and associated metadata obtained from the preceding step, we employ Elmer (CSC - IT Center for Science, 2023), an open-source finite element software for multiphysical problems to solve a steady state heat equation (1) with at least one internal heat source.

$$-\nabla \cdot (k \nabla T) = \rho h \qquad (1)$$

Where T is the temperature, k is the thermal conductivity, ρ is the density and h is the heat source. Heat sources are placed in all 3D objects that carry the 'Air_body' tag placed during the IFC processing step (see Section 2.1). In our workflow, these objects are meant to represent the indoor air volume inside of the building model and should simulate the heating of a building. Elmer exposes a functionality called *Smart Heater Control* that adjusts the intensity of the prescribed heat source. This adjustment ensures that a given target temperature is reached at a designated point within the heated object. Utilizing this feature, we aim to regulate the desired indoor room temperature effectively. The initial value of the heat source is set to a predetermined default in our code. However, the target indoor temperature remains adjustable, allowing the flexibility to accommodate various use-case requirements. We simulate the indoor air with equation (1) which is a simplification. Instead of applying an indoor heat transfer coefficient, we model the indoor air as a volume with high thermal conductivity (see Section 3 for the specific value).

As an external boundary condition, we apply a boundary condition which simulates the heat transfer between the external boundaries and the surrounding environment of the building model using a heat transfer coefficient (2).

$$q = \alpha \cdot (T - T_{ext}) \qquad (2)$$

Where q is the heat flux, α denotes the heat transfer coefficient (combined convective and radiative) and T_{ext} refers to the external temperature.

Due to the fact that Elmer does not provide a built-in Python interface, we made use of the library pyelmer (Wintzer et al., 2023) which enables the setup of the Elmer simulation through Python code. Elmer processes UNV files and converts them into mesh files readable by the Elmer solver. With pyelmer we were able to assign the boundary condition to all detected external boundaries from Section 2.2. The material parameters for the different geometries are queried from a YAML file that contains the density and thermal conductivity of a predefined set of materials. This material database can be easily extended by inserting the desired material information into the YML-file.

When the materials and boundary conditions are set, the Elmer solver is triggered. The output of the solver is a VTU file that contains a calculated temperature for every mesh point. Furthermore, the Elmer solver also returns the tuned heating power of the heat sources. These results can be used to identify thermal bridges, to calculate heat and cooling loads and seasonal heating or cooling demands. For visualization and postprocessing purposes tools like ParaView (Ahrens et al., 2005) can be used.

2.4 Two-Dimensional Case

In addition to the 3D case, we implemented the possibility to run a 2D thermal simulation on a horizontal cross-section of the digital building model. For this we added a function to the IFC processing module that takes a given height as input parameter and creates a horizontal cross-section of processed geometries at this height. The rest of the algorithm then works analogously to the 3D case following the workflow described in Sections 2.1 to 2.3.

3. Application and Case Study

To test the developed workflow, we modelled a simple 3D building which serves for demonstration purposes in this study. The model was created with Autodesk Revit (Autodesk, 2022) and exported in IFC-format (see Fig. 2). It is a two-storey residential building with two separated apartments (one per storey). The outer walls consist of 30 cm of sand-lime bricks, followed by 12 cm of external insulation (EPS), and a 1-cm thick layer of plaster on both the interior and exterior surfaces. All external horizontal structural elements like slabs and roof consist of 20 cm reinforce concrete, 10 cm of external insulation (EPS) and in case of the roof also of a 1-cm thick exterior finishing layer. There are six one-winged wooden windows and two double-winged wooden terrace doors. Both windows and terrace doors are modelled with simple rectangular wooden frames. The glass is modelled as a 5 cm thick monolithic block with equivalent thermal conductivity to simplify the modelling of a triple-paned window and avoid the modelling of the gas layers. For simplicity reasons internal walls and rooms were not modelled. The model contains two elements of type *IfcSpace* (one per storey) which represent the air volumes inside ground floor and first floor storey respectively.

Fig. 2 – Digital building model in IFC file format

We use this model as input for our automated workflow. In the first step the trees, the hand railing and the external stairs are filtered out then the STEP file with the remaining processed geometries is generated.

As meshing algorithm, we conducted the Delaunay algorithm (Gmsh Developers, 2024) which generates a mesh of linear tetrahedral elements. The target mesh size was set to 5 cm. Fig. 3 shows the result of the meshing process: the discretized model with all meshed geometries. The mesh consists of nearly 3 million elements with a total of approximately 510 000 nodes.

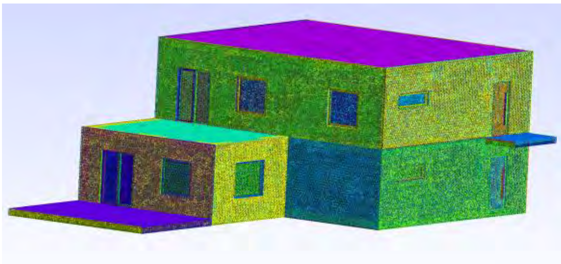

Fig. 3 – Discretized model with meshed geometries in UNV file format

For the Elmer FE simulation we set the external temperature T_{ext} to 277.15 K (4 °C). We applied a uniform value of 25 W/(m² K) for the heat transfer coefficient α, which represents linearised radiative and convective heat transfers. The internal temperature at the control point of Elmer's *Smart Heater* was set to 293.15 K (20 °C). The thermal conductivity k of the most relevant materials that appear in the model are listed in Table 1.

In Fig. 4 we show a visualization of the obtained results after the simulation finished. The figure shows a screenshot from the visualization tool ParaView with the temperature distribution of the simulated model. Through this, a first plausibility control of the results can be done. The temperature on the surface of the building model is almost constant in all points. Only the window frames have a slightly higher temperature compared to the rest of the surface. We chose a well-insulated building as use case because this is how new buildings today are designed and planned. However, another interesting use case would be an existing historic building with poor thermal insulation characteristics and extensive thermal bridges.

A more detailed view is shown in Fig. 5. Here a cross-section of a window in the upper storey of the model is depicted. The temperature distribution inside the insulated wall and the connection to the window can be observed. The concrete roof slab has a significantly higher thermal conductivity than the sand-lime brick wall, which can also be observed in the figure.

The total heat source power was calculated to 3850 W and the computing time of the whole workflow was 408 seconds (6.8 minutes). In Table 2 the induvial computing times for every workflow step can be found. It was run on a machine with an Intel Core i7-1165G7 (2.80 GHz) processor.

Table 1 – Primary materials and their thermal conductivity

Material	**Thermal conductivity (W/(m K))**
Sand-lime-brick	0.560
Reinforced concrete	2.300
EPS	0.035
Plaster	0.500
Window wooden frame	0.130
Window glass (modelled as monolithic block)	0.033
Air volumes (ground and first floor)	1000

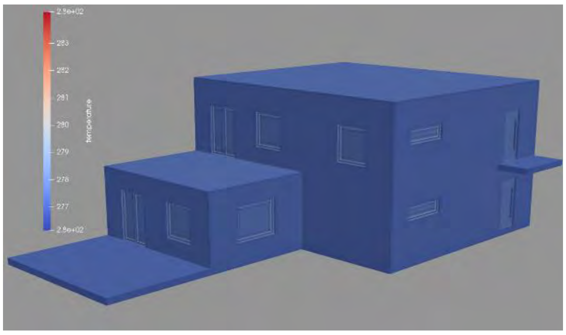

Fig. 4 – Visualized simulation results (temperature distribution on building envelope) in VTU file format

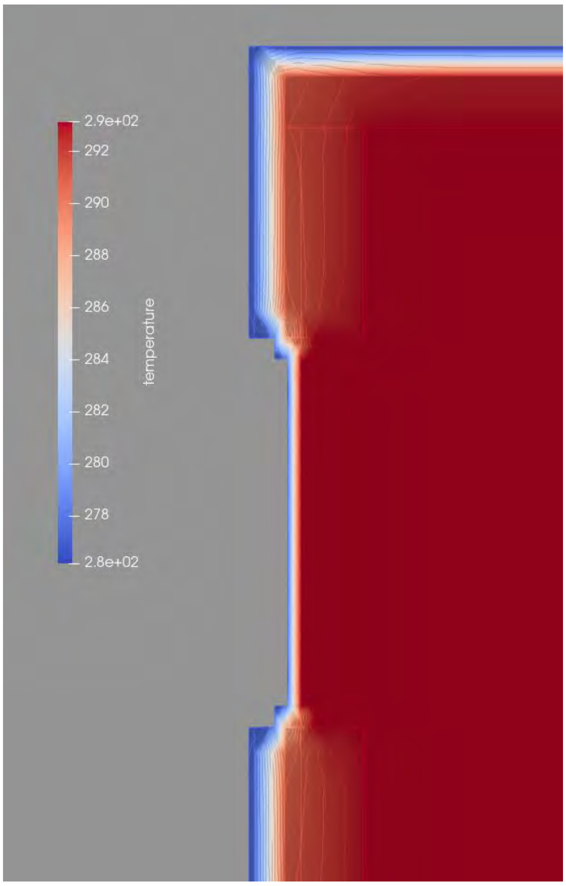

Fig. 5 – Cross-section view of window, wall and roof from the upper storey of the building model, temperature contour lines

Table 2 – Computing times of individual steps

Workflow step	Computing time (s)
IFC Processing	3.89
Discretization	102.44
Thermal FEM simulation	301.93
Total	408.27

4. Discussion

Because we expose relevant parameters both for meshing and FE simulation, the workflow is highly adaptable and can be used for a wide range of different use-cases. However, a drawback that we found was that the solid air volume inside the building, represented by *IfcSpace* elements, must be modelled quite accurately in order to avoid gaps between the internal air volume and the building structure. This can be a tedious modelling task especially when dealing with complex window geometries or other openings. Another possible approach here would be to use the same procedure as for the external boundaries and apply a boundary condition with heat transfer coefficient also to the internal boundaries with an internal temperature instead of using the air volume with a heat source. In this case it would be more complex to detect all boundaries automatically. It might be necessary then to have some sort of user selection to distinguish the internal and external boundaries and to apply the desired temperature.

In terms of computation time, our implementation has still potential for optimizations. The meshing algorithm can be fine-tuned to have adaptive mesh sizes depending on the complexity of geometries. The FE simulations is now running in a single process, but Elmer also supports multiprocessing, so there is potential for parallelising the simulation and to reduce the computation time significantly, always depending the machine the simulation is running on.

5. Conclusions and Outlook

We developed and implemented an automated open source-based workflow to efficiently interface BIM with FE thermal simulations. The corresponding simulation results can be used to evaluate the building energy performance. The key features of our workflow are the consistent data flow of geometry and material information and high degree of automation, in specific the automatic assignment of boundary conditions.

With the conducted case-study we were able to demonstrate that our workflow produces plausible results within a short computing time and little need of manual configuration. Furthermore, we offer a high degree of flexibility since the relevant parameters can be configured based on the needs of specific use-cases. Due to its open-source nature the workflow and algorithms are easily reproduceable and extendable.

The proposed workflow has the potential to be integrated in digital twins in order to run simulations based on real-time sensor-data and assist prediction models. In future work the encountered drawback of missing flexibility in terms of the assignment of different boundary conditions for different areas of the digital model must be addressed. The optimization of meshing and parallel processing could also be subjects. An additional possibility for further enhancements could be to leverage thermal properties or any other properties related to materials that are already integrated in the digital building model instead of retrieving them from a separated material database. Furthermore, the workflows' evaluation and testing on a broader range of different use cases (e.g. existing historic buildings) should be included in future works. In terms of simulation capabilities, the integration of time-dependent dynamic simulations including solar radiation would be a valuable enhancement.

Nomenclature

Symbols

T	Temperature (K)
k	Thermal conductivity (W/(m K))
ρ	Density (kg/m^3)

h	Heat source (W/kg)
∇	Nabla operator (-)
q	Heat flux (W/m²)
α	Heat transfer coefficient (W/(m²K))
T_{ext}	External temperature (K)

References

Ahmad Mohammad Ahmad, Sergio Rodriguez, Mian Atif Hafeez, Nashwan Dawood, Mohamad Kassem, and Khalid Kamal Naji. 2022. 'Drivers for Energy Analysis towards a BIM-Enabled Information Flow'. *Smart and Sustainable Built Environment*, April. https://doi.org/10.1108/sasbe-07-2021-0129

Andriamamonjy, Ando, Dirk Saelens, and Ralf Klein. 2019. 'A Combined Scientometric and Conventional Literature Review to Grasp the Entire BIM Knowledge and Its Integration with Energy Simulation'. *Journal of Building Engineering* 22:513–27. https://doi.org/10.1016/j.jobe.2018.12.021

Autodesk. 2022. 'Revit 2023'

Carvalho, José Pedro Almeida, Manuela Guedes de Almeida, Luís Bragança, and Ricardo Mateus. 2021. 'BIM-Based Energy Analysis and Sustainability Assessment - Application to Portuguese Buildings'. *Buildings* 11 (6): 246. https://doi.org/10.3390/buildings11060246

CSC - IT Center for Science. 2023. 'Elmer - Multiphysical Simulation Software'. https://www.csc.fi/web/elmer/documentation

Fedorik, F., T. Makkonen, and R. Heikkilä. 2016. 'Integration of BIM and FEA in Automation of Building and Bridge Engineering Design'. In 2016 Proceedings of the 33rd ISARC, 735–41. Auburn, USA. https://doi.org/10.22260/ISARC2016/0089

Geuzaine, Christophe, and Jean-François Remacle. 2009. 'Gmsh: A 3-D Finite Element Mesh Generator with Built-in Pre- and Post-processing Facilities'. *International Journal for Numerical Methods in Engineering* 79 (11): 1309–31. https://doi.org/10.1002/nme.2579

Gmsh Developers. 2024. 'Gmsh Reference Manual'. https://gmsh.info/doc/texinfo/gmsh.html

Jia, Jing, Jieya Gao, Weixin Wang, Ling Ma, Junda Li, and Zijing Zhang. 2022. 'An Automatic Generation Method of Finite Element Model Based on BIM and Ontology'. *Buildings* 12 (11): 1949. https://doi.org/10.3390/buildings12111949

Krijnen, Thomas. 2023. 'Ifcopenshell'. https://ifcopenshell.org/

Leonardi, Maria Laura, José Granja, Daniel V. Oliveira, and Miguel Azenha. 2024. 'Scalable BIM Based Open Workflow for Structural Analysis of Masonry Building Aggregates'. *Computers & Structures* 297 (July):107321. https://doi.org/10.1016/j.compstruc.2024.107321

Li, Hong Xian, Zhiliang Ma, Hexu Liu, Jun Wang, Mohamed Al-Hussein, and Anthony Mills. 2020. 'Exploring and Verifying BIM-Based Energy Simulation for Building Operations'. *Engineering, Construction and Architectural Management* 27 (8): 1679–1702. https://doi.org/10.1108/ecam-06-2019-0314

Ou, Yaowen, Yunus Emre Harmanci, Qian Chen, Borja Garcia de Soto, Vasileios Ntertimanis, and Eleni Chatzi. 2017. 'A Finite Element Analysis Approach to Improve Interoperability for Thermal Energy Simulations'. Edited by Eugenio Pellicer, José M. Adam, Víctor Yepes, Amarjit Singh, and Siamak Yazdani. Proceedings of International Structural Engineering and Construction 4 (1). https://doi.org/10.14455/ISEC.res.2017.28

Paviot, Thomas. 2023. 'PythonOCC, 3D CAD/CAE/PLM Development Framework for the Python Programming Language, PythonOCC – 3D CAD Python'. https://github.com/tpaviot/pythonocc.

Pezeshki, Zahra, Ali Soleimani, Ali Soleimani, and Ahmad Darabi. 2019. 'Application of BEM and Using BIM Database for BEM: A Review'. *Journal of Building Engineering* 23 (May):1–17. https://doi.org/10.1016/j.jobe.2019.01.021

Wintzer, Arved, Max Schröder, Anil Kunwar, Mudassir Chapra, Daniel Bates, Aerosmite, and Tom Gillam. 2023. 'Nemocrys/Pyelmer: Pyelmer v1.2.0'. https://doi.org/10.5281/ZENODO.8255634