

DEVELOPMENT OF A TOOL FOR INTERACTIVE DETAILING OF AREAS OF OBJECTS FOR THE STRENGTH MODELLING SYSTEM

E. Budlov^a, O. Iakushkin, O. Sedova, A. Gogotov

¹ *Saint Petersburg State University, 7-9 Universitetskaya emb., Saint Petersburg, 199034,
Russia*

E-mail: ^abudlow@yandex.ru

The development of technological progress makes increased demands on the strength properties of structural elements of buildings and structures, machines and mechanisms, and a decrease in their material consumption. This leads to the need for effective use of existing and creation of new methods of solid mechanics and the training of new highly qualified specialists. Additionally, having control over the objects` sizes will allow potential users (designers, engineers etc.) to consider the desired properties in the optimization problem. Otherwise, it would be both quite difficult and inefficient to utilize.

This article describes a system for CAD modelling. Its implementation is based on open-source software, such as Python programming language, Jupyterlab and FEniCS library. The system also supports topological optimization and allows its user to integrate the desired numerical methods, which is currently not supported in the existing solutions. The execution time of solving the example problem was measured and presented together with the result. Special attention was paid to the comparison of the described solution with the currently existing ones.

Keywords: CAD, FEM, Jupyterlab, Modelling, Cloud computing

Egor Budlov, Oleg Iakushkin, Olga Sedova, Alexander Gogotov

Copyright © 2021 for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

1. Introduction

One of the most important tasks in the field of deformable body mechanics is the representation of an existing CAD model in the form of a finite element mesh. Such a transformation will allow for the final calculations and optimization of the mathematical parameters of the object. Our work aims to find the optimal way to solve the following (and other related) problems:

- Preservation of all geometric features of a complex object in its model.
- Ability to make changes of any scale to every part of the model.
- Support for complex operations with the object: local rescaling of big and small details, optimization of parameters.
- Elimination of loss of information about the properties of objects.

2. Related work

Although various popular solutions already exist, both open-source (FreeCAD, Salome) and commercial (Comsol, Ansys [1,2]), they nevertheless have their drawbacks.

In particular, to make the user experience as easy as possible, integration with an intuitive visual interface is required. This approach will provide interactivity, as well as the speed of interaction and execution of operations. Its implementation will increase the speed and productivity of work, as well as reduce the requirements for the necessary user skills.

3. Problem definition

Various existing methods of numerical modelling are based on testing the technology's operability on separate boundary value problems and do not allow the transition from the description of the algorithm to its immediate application by end-users. Our goal is to create an environment suitable for easily integrating custom calculation methods into the underlying system we have implemented. At the same time, an important task is to support local scaling.

4. System description

We believe that the most optimal solution for the described problems is an implementation in the form of a computer-aided design system that will support the modelling and processing of objects. To improve efficiency, the web service format was chosen [3, 4]. At this stage, the following prototype system is being developed in Python:

- The main functionality of numerical calculations was performed using the FEniCS and TopOpt modules;
- To implement the automated design of 3D models, tools of the PythonOCC, VTKjs libraries were used;
- The Jupyter framework, as well as the GMSH and VTK utilities, were chosen as the basis for the interface visualization;
- Containerization is done using Docker functionality.

4.1 FEniCS

FEniCS is a popular open-source computing platform for solving partial differential equations (PDEs). FEniCS enables users to quickly translate scientific models into efficient finite element code. With the high-level Python and C++ interfaces, it is easy to get started, but it also offers powerful capabilities for more experienced programmers. Each component of the FEniCS platform has been

fundamentally designed for parallel processing, and it can run on a multitude of platforms ranging from laptops to high-performance clusters.

4.2 TopOpt

TopOpt is a fully parallel framework for conducting very large scale topology optimization on structured grids. It solves the compliance minimization problem subject to volume and maximum size restrictions. The problem is formulated using the robust design approach based on eroded, intermediate and dilated designs. A quarter of the 3D MBB beam is modelled and the density filter weights are adjusted to simulate an extension of the design domain.

4.3 JupyterLab

JupyterLab is an interactive development environment for working with notebooks, code and data. Most importantly, JupyterLab has full support for Jupyter notebooks. Additionally, JupyterLab enables you to use text editors, terminals, data file viewers, and other custom components side by side with notebooks in a tabbed work area. It is easy to deploy, fast, interactive and highly customizable – so this framework was chosen as the interface of the system [fig. 1].

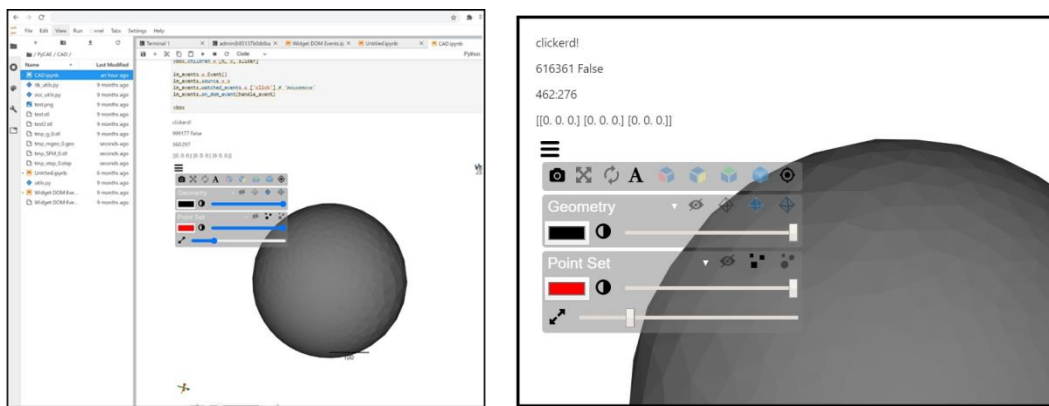


Figure 1. The system interface

More efficient usage of the system and resources, easy scaling, stability and cross-platform are ensured via Docker.

5. Calculations

5.1 The example problem

As an example, let us consider an area of a 3-D cube held at 4 corners on one face of the cube. A pressure load is applied to the outer face of the center piece on the opposite side. A .vtu file is saved for the result of each iteration of the algorithm. The result of the last iteration is the proposed solution by the algorithm [fig. 2].

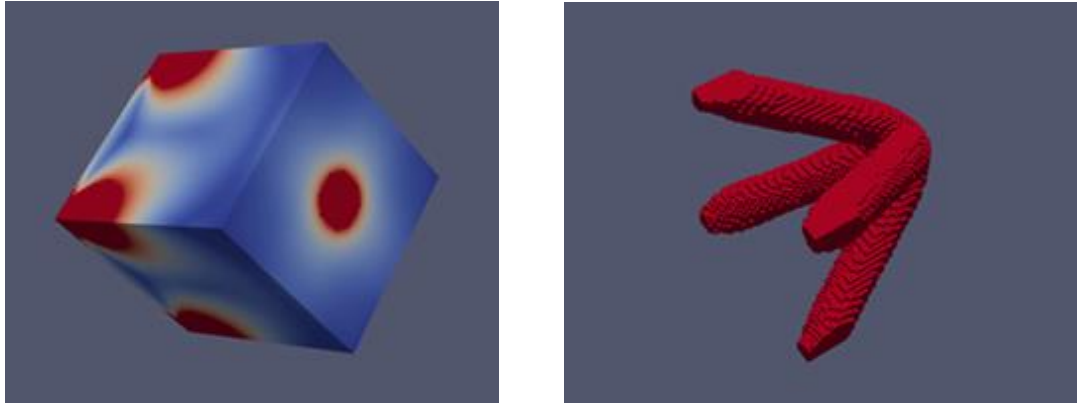


Figure 2. The first and the last iterations of the algorithm

Another possible application – FEM modelling and optimization. The general procedure is as follows:

- Mesh definition via parameters;
- FEM and modelling analysis with FEniCS;
- Topological optimization using TopOpt;
- FEM analysis of the result;
- Comparison with the original;

The console displays information about the execution of the algorithm, indicating the number of the corresponding iteration. The result is an optimally rigid design to solve problems of minimum compliance (maximum stiffness), optimal distribution of two materials for thermal conductivity problems, and optimal distribution of material for efficient mobility.

5.2 The execution time

Parallel computing is of utmost importance when we are talking about modelling and optimization. The calculations were performed on a virtual machine with six physical cores [fig. 3]. Because topological optimization is a relatively complex operation, even when using MPI technologies, it still takes a significant amount of time.

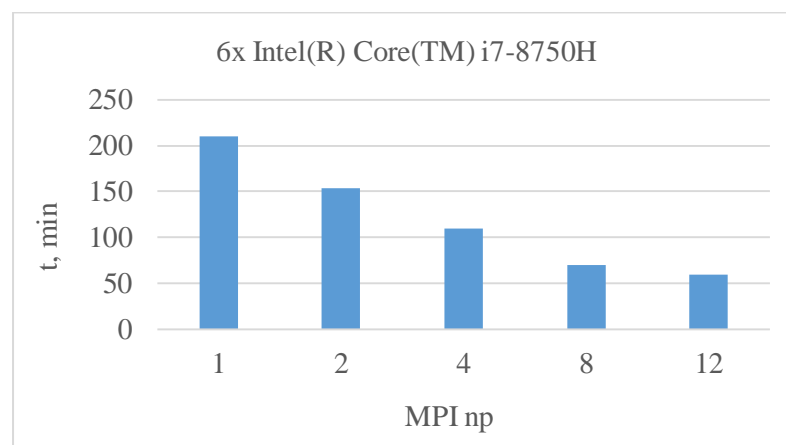


Figure 3. The execution time

5.3 The differences from the existing solutions

Although some open-source projects can sometimes offer quite decent performance, they

usually have very limited scope and lack an intuitive interface. Commercial solutions are expensive, complex and hardly customizable. In described solution, we try to solve most of these problems through integrating modern open-source solutions into a unified system that will support customization, be scalable and intuitive [fig. 4].

Solutions	Pros	Cons
Simple open-source projects	Free, fast, easy to use	Quality is not enough at all, lack of interface
Complex open-source solutions	Free, good quality	Time-consuming, limited functional, lack of interactive interface
Commercial solutions	Good quality, more supported operations	Expensive, time-consuming, hardly customizable, difficult to master
Proposed solution	Free, customizable, broad functionality support, intuitive interface, cross-platform	Time-consuming (for complex operations)

Figure 4. The pros and cons of different solutions

6. Conclusion

The proposed solution made it possible to introduce a step for interactive adjustment of the local mesh resolution between the step of modelling the object and its division into finite elements. This gives us many opportunities to fine-tune the characteristics of the processed object. The system also supports the topological optimization of the object. At the same time, the implementation of tools for managing the local mesh resolution will allow users to integrate their modern numerical methods, such as XFEM and GFEM, the use of which is not available in existing modelling packages. In future, we plan to work on the integration of more functional libraries and further interface improvement.

References

- [1] Iakushkin O., Sedova O., Grishkin V. Jupyter extension for creating CAD designs and their subsequent analysis by the finite element method // CEUR Workshop Proceedings, vol 1787. RWTH Aachen University, 2016. P. 530-534.
- [2] Iakushkin O., Sedova O. Creating CAD designs and performing their subsequent analysis using open source solutions in Python // AIP Conference Proceedings, vol 1922.2018. P. 140011.
- [3] Iakushkin O., Kondratiuk A., Eremin A., Sedova O. Development of a Containerized System to Build Geometric Models and Perform Their Strength Analysis // ICAIT'2018: Proceedings of the 3rd International Conference on Applications in Information Technology. 2018. P. 146-149.
- [4] Sedova O., Iakushkin O., Kondratiuk A. Creating a tool for stress computation with respect to surface defects // CEUR Workshop Proceedings, vol 2507. RWTH Aachen University, 2019. P. 371-375.