The 8th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2022
5-9 June 2022, Oslo, Norway

Large-scale Topology Optimization on a desktop using the GPU

Erik A. Träff[1*], Niel Aage[1,2] and Ole Sigmund[1,3]

[1] Technical University of Denmark, Nils Koppels Allé, B. 404, 2800 Kgs. Lyngby,
eratr@mek.dtu.dk
[2] naage@mek.dtu.dk
[3] sigmund@mek.dtu.dk

Keywords: *Topology Optimization, Large-scale, GPU, OpenMP, MPI*

Topology optimization is a rapidly maturing form of design optimization which already has shown great results in many practical applications. In most industrial applications of topology optimization solving large-scale problems is necessary, as the feature size of the optimized designs is directly tied to the computational effort required. An approach to solve these large-scale problems is the use of HPC clusters [1]. While HPC clusters enable the solution of large-scale problems, there is a large financial cost associated with using HPC systems, which can make topology optimization prohibitively expensive for some would-be practitioners in both industry and academia.

This presentation will discuss the efficient implementation of topology optimization for desktop systems. Enabling the solution of large-scale topology optimization problems on such systems is of high interest, as even high-end desktop solutions are an order of magnitude cheaper to acquire and operate than HPC systems. Three separate implementations of linear elastic topology optimization are presented; a distributed memory implementation based on MPI, a shared memory CPU implementation using OpenMP, and a single GPU implementation using Futhark [2]. All implementations use a geometric multigrid preconditioner, and are able to solve large-scale topology optimization problems of 30 million elements within a reasonable time-frame (less than 24 hours).

The distributed memory implementation is furthermore able to solve unstructured problems which typically arise in industrial settings, shown by an example of a flywheel used in a direct-drive wind turbine, which makes this implementation a suitable reference for the single desktop implementations. These three implementations are used as a basis for discussing the possibilities and limitations of performing large-scale topology optimization on a desktop machine.

REFERENCES

[1] Aage, Niels, Erik Andreassen, Boyan S. Lazarov, and Ole Sigmund. 2017. “Giga-Voxel Computational Morphogenesis for Structural Design.” Nature 550 (7674): 84–86. https://doi.org/10.1038/nature23911.

[2] Henriksen, Troels, et al. ”Futhark: purely functional GPU-programming with nested parallelism and in-place array updates.” Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation. 2017.