Structural Optimisation Methods as a New Toolset for Architects

Sebastian Białkowski

Lodz University of Technology
sebastian.bialkowski@p.lodz.pl

The paper focuses on possibilities of already known engineering procedures such as Finite Element Method or Topology Optimisation for effective implementation in architectural design process. The existing attempts of complex engineering algorithms implementation, as a form finding approach will be discussed. The review of architectural approaches utilising engineering methods will be supplemented by the author's own solution for that particular problem. By intersecting architectural form evaluation with engineering analysis complemented by optimisation algorithms, the new quality of contemporary architecture design process may appears.

Keywords: topology optimization, design support tools, complex geometries, finite element method, CUDA

INTRODUCTION

A continuous development of Computer Aided Design tools complement by computational design methods highly influence architectural design development. Architects are looking for new approaches which may support their form finding process with new and unique solutions for particular problem. As an extraordinary tool in the architectural design practice, Topology Optimisation methods might be pointed. That algorithm is very widely used in the industrial product design such as aerospace and automotive where mechanical parts efficiency and its material usage is crucial. Also civil engineering is not lagging in this field (Guest and Moen 2010). Many scientific discourse and researches have been made, implementing Topology Optimisation for various purposes. In majority of them, only separate structural elements are taken into account such as beam, column or truss layout (Amir and Bogomolny 2011). In this studies the emphasis is put on structural or material efficiency optimisation.

BACKGROUND

Methods

Topology Optimisation of structures can be divided on two main subfields: Layout Optimization (LO) and General Shape Optimisation (GSO) (Kutylowski 2004). The first approach concerns grid-like or trust-like structures having very low volume fraction. Opposed to LO method, GSO is concerning higher volume fractions, optimising the topology and shape of material continuum. The three subtypes of GSO method can be pointed, as shown on Figure 1, based on material properties applied as a ground structure: Isotropic Solid-Empty (ISE), Anisotropic Solid-Empty (ASE) and Isotropic Solid-Empty Porous (ISEP). The force distribution in the material is the main criteria for that classification. GSO can be also categorized based on...
optimisation algorithm implemented on. Two most popular approaches are Solid Isotropic Microstructure with Penalization (SIMP) developed in 1989 and Evolutionary Structure Optimisation (ESO) firstly applied in 1992. The principle of SIMP methodology is material density distribution by applying heuristic optimization algorithm such as Optimal Criteria or Methods of Moving Asymptotes (Bendsøe and Sigmund 2004). Indecently which algorithm will be chosen, the optimisation process is based on design sensitivities. The ESO approach introduced firstly by Xie and Steven in 1992 were detail described in paper from 1997. Through time, method derived into different approaches such as:

- **ESO** - base method, which only remove material based on stress criteria. Algorithm works until maximum stress is achieved.
- **AESO** - Additive ESO - procedure starts with small amount of material, and adds it near the points with high stress until maximum stress criteria is met.
- **BESO** - Bi-directional ESO - this method combine principles of basic ESO and AESO approaches. Algorithm is able to add or remove material in every point in space during any step of optimisation. This methodology decrease the possibility of hitting local optimum which can led to wrong solutions.

Both algorithms are based on Finite Element toolset.

**Topology Optimisation in architectural design**

Topology Optimisation as an engineering tool is rarely applied in the architectural design process. Commonly it is caused by a complex and time taking process to achieve results which would satisfy a designer. However, that obstacle should not discourage some architects from experimenting with these tools in the field of architecture. The first practical application of structural optimisation methods was
made in 2004 in the project named Akutagawa River Side in Takatsuki in Japan. Two adjacent walls in the four story office building nearby Takatsuki Japanese Railway station were optimised by two dimensional implementation of ESO method (see Figure 2). During the whole process multiple types of load were taken into account. Except from typical dead and live load, dynamic earthquake loads were also applied (Ohmori 2010).

Related approach, but in a bigger scale was presented by Stromberg (Stromberg et al. 2011) towards a high raised building. This theoretical application of Simple Isotropic Microstructure with Penalisation (SIMP) method developed by Bendsøe and Sigmund (Bendsøe and Sigmund 2004) is exploring design for constructability of high-raised building. As authors pointed, "The optimality comes from the idea of understanding how the forces are "moving" through the structure to the foundation and embrace this flow with the structural members", the research are conducted on external skin of designed building with compliance minimization as an objective function (see Figure 3).

**Arata Izosaki - the pioneer of Structural Optimization methods applied in Architecture**

All previous approaches implementing Structural Optimisation methods were focused on structural aspects of this tool diminishing their spatial appearance. One of the first attempts to use shapes, obtained from Topology Optimisation algorithms as an architectural form, was proposed by Japanese architect Arata Isozaki with collaboration of a structural engineer Matsuro Sasaki. They designed a multifunctional object for culture and tourism, called Illa de Blanes at the seaside of Blanes (Costa Brava, Spain) developed in the years 1998-2002 (Januszkiewicz 2013).

This enormous complex, covering 75.000 square meters of mixed functions is supported by a large structure generated by ESO algorithm. In combination with tree like, organic shaped columns, building was equipped with double curved roof created by the application of 3D Extended method of ESO. Despite its iconic and unique shape (see Figure 4), due to budgetary constraints, the project has never been build. Nevertheless, that approach has been the first comprehensive application of topology optimisation algorithms towards architectural form finding process.
One year after Illa de Blanes project, Arata Isosaki together with Matsuro Sasaki took part in an international architectural competition for the largest train station in Italy - Santa Maria Novella in Florence. Japanese designers proposed a huge structure generated by Topology Optimisation algorithms (see Figure 5), namely 3D Extended Evolutionary Structure Optimisation (EESO) (Januszkiwicz 2013). A 400 meters long and 42 meters wide flat roof, designed as a land strip for lightweight aircrafts, elevated 20 meters above the ground, was supported by massive columns in a few points. On the ground, those columns had only four main roots to grow from, which reminiscent of trees branches (see Figure 6). Unfortunately, the project won the second prize, and it has never been build.

The vision of application of engineering tools, such as Topology Optimisation in architectural design, finally came true in 2008. The Qatar National Convention Centre (QNCC) in Doha became a chance for Arata Isosaki to implement his innovative vision of architecture driven by engineering computational methods. In cooperation with Buro Happold he designed a tree-like structure (see Figure 7), which supported 250 meter long and 110 m wide lobby roof (see Figure 8). As in a previous approach, Izosaki implemented iterative EESO algorithms. Until today, it is the biggest structure created by generative tools based on Topology Optimisation (Zwierzycki 2013).

**Optimisation outcome and manufacturing process.**

A result of Topology Optimisation is typically represented as a spatial boundary of material distribution. Three dimensional versions of algorithms return complex geometries, highly difficult to manufacture by a common production process. Especially, when the result of optimisation is not intended for mass production, but is unique for particular problem. Some solutions for solving that problem is described in researches done by Arup. They implemented an additive manufacturing process named SLS for producing irregularly shaped structural ele-
ments made from steel for tensegrity structure (see Figure 9). Generally shaped element (by authors named AM Node) were optimised by SIMP method and as authors claimed "The topology optimization resulted in an organic form with less material while the original functions as cable connectors are still ensured." (Block et al. 2015). Compared to traditional design, presented approach not only provide optimisation of material, but also caused increase of node stiffness. That proves usefulness of Structural Optimisation Methods such as Topology Optimisation, but cost of production based on 3D printing technology is still too high.

Figure 9
Optimised structural element prototyped by utilizing SLS 3D-printing technology.

NEW TOOLSET

tOpos - new Topology Optimisation approach towards architecture

Existing architectural adaptation of Topology Optimisation are the result of deep research and time consuming experiments often aided by specialists in civil engineering. Furthermore, a majority of engineering software which are implementing Topology Optimisation algorithm, needs high abilities and knowledge to operate on. Their explicit user interface and, moreover, plethora of options and decisions which user has to make, put it as a highly specified software dedicated to the limited range of users. That limitation decrease a possibility of usage Topology Optimisation by architects for enriching their design process. All presented arguments affirm the author about the need for developing a new tool for designers. Topology Optimization as a material distribution method based on numerical approach can be successfully enhanced by contemporary computing tools. The aim of the author research is to create a form finding real-time tool for architects based on the engineering method of Topology Optimization. The project, named "tOpos", unlike the complicated and expensive commercial tools for engineers such as Altair or Abacus software’s is planned to be easy to use for architects, fast and efficient. None special knowledge about continuum mechanics will be needed to use it. By developing this tool, it is highly intended to give the opportunity to variety of architects and designers to use the exceedingly complex and compound process to improve their designs. What is more, for a fast and effective design process, the designed software needs to provide an immediate feedback to a designer.

Structure of the algorithm

To obtain high performance optimisation process it is essential to understand basis of this algorithm. Topology Optimization is a mathematical approach that optimises material layout within a given design space, for a given set of loads and boundary conditions, such that the resulting layout meets a prescribed set of performance targets. The mathematical formulation of the optimization problem can be state as follows:

\[
\min_{x} \quad c(x) = U^T K U = \sum_{e=1}^{N} E_e(x_e) u_e^T k_0 u_e \\
\text{subject to} \quad V(x)/V_0 = f \\
K U = F \\
0 \leq x \leq 1
\]

where \(c\) is the compliance of the design, \(U\) and \(F\) are the global displacement and force vectors, respectively, \(K\) is the global stiffness matrix, \(u_e\) is the element displacement vector, \(k_0\) is the element stiffness matrix for an element with unit Young's modulus, \(x\) is the vector of design variables (i.e. the element density...
ties), \( N \) is the number of elements used to discretize the design domain, \( V(x) \) and \( V_0 \) are the material volume and design domain volume, respectively, and \( f \) is the prescribed volume fraction. To solve equitation \( KU = F \) finite element methods (FEM) (Zienkiewicz and Taylor 2005), referred to as finite element analysis (FEA) by some authors, has to be implemented.

From the algorithmic point of view, the whole optimisation process can be divided into three phases presented on Figure 11. Prepocess is the first stage, where Finite Element model is build and initial design variables are defined. User need to define design domain, all loads and supports conditions and target volume fraction. The next step is named Processing, where the main optimisation process is performed. This step is iteration loop where specified subroutines are executed as follow:

- Based on current distribution of densities, compute by FEA subroutine the design resulting displacement.
- Compute the compliance and sensitivities of the design.
- Check the improvement over last design. If only marginal improve achieved, terminate the loop. Else continue.
- Update the densities by chosen algorithm (Optimality Criteria or MMA)
- Repeat the iteration loop.

The last part of the algorithm is Post processing. The outcome densities can be represent as a 3D spatial model of optimised material boundary. As shown in Figure 11, if proper and efficient FEA subroutine is available, only few functions need to be added to perform Topology Optimisation. The performance of analysing subroutines is fundamental for whole optimisation process.
Code implementation

Based on scientific researches (Schmidt and Schulz 2011) authors came up with an idea of developing and implementing own Topology Optimization algorithm enhanced with the GPU acceleration. Contrary to current scientific implementation of Topology Optimisation presented by Schmidt and Schulz (2011) or Liu and Tovar (2014), tOpos has possibility to define irregularly shaped ground structure. The module responsible for creation of Finite Element model, creates elements and nodes only inside user defined boundary represents as a mesh (look Figure 12). The basic environment for tOpos is Rhinoceros3D software with Rhinocommon libraries. McNeel software is used as 3D engine for data display and what is more, as a modelling environment necessary for the algorithm to create input data. tOpos also utilises Rhinocommon library to process meshes and vectors.

The core of tOpos is based on the host code (performed on CPU) implemented on DotNet C# with acceleration kernels based on CUDA API (Sanders et al. 2012), the extension of C language. To execute real time Finite Element Method subroutine, an algorithm needs to be parallelized on many threads. For easy mathematical operations, which need to be repeated many times, GPU demonstrates higher performance by speeding computation dozen of times. GPGPU (General Purpose computing on Graphics Processing Units) allows to exploit parallel architecture pipeline to bring a new approach to a real time application. On the other hand, GPU has major disadvantage which need to be concern during algorithm design. Memory capacity embedded on Graphic Cards is smaller than amount of CPU memory installed in modern computer. This limitation has a big impact on data structure and it management.

Based on Martinez-Frutos and Herrero-Pérez (2015) researches as a main FEM solver iterative Conjugate Gradient (CG) algorithm has been chosen. Opposed to direct solvers such a Cholesky or QR factorization, iterative solvers are highly adaptable for specific needs such a memory consumption or time re-
served for solving particular problem. During tOpos developing process, two CG solvers were design concerning memory usage on certain graphic card. The main difference is storage model of global stiffness matrix. The first solver, called PreAssembly, creates stiffness matrix $[K]$ in CRS format before iterative process will run. This approach reuse once computed matrix, therefore algorithms execute faster, but needs additional memory resources on GPU. The second solver, named MatrixFree calculate Matrix-Vector product in each iteration and never stores global stiffness matrix. That cause slower code performance, but saves many resources on graphic card (see Section: Numerical Results).

<table>
<thead>
<tr>
<th>Benchmark No.</th>
<th>Elements</th>
<th>Nodes</th>
<th>Dofs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3440</td>
<td>4663</td>
<td>13989</td>
</tr>
<tr>
<td>2</td>
<td>27472</td>
<td>32260</td>
<td>96780</td>
</tr>
<tr>
<td>3</td>
<td>92654</td>
<td>103291</td>
<td>309873</td>
</tr>
<tr>
<td>4</td>
<td>219953</td>
<td>238849</td>
<td>716547</td>
</tr>
</tbody>
</table>

**Table 1**
Benchmark setup.

The performance evaluation was based on four benchmarks performed on numerus mesh density presented in Table 1 and visualised on Figure 13. During benchmarks CPU was fully overloaded. GPU kernels reported 89% of computational occupancy. It is caused by adaptation of FEM on irregular shapes.

The results are presented in three group of graph. The Figure 15 represents the general execution time need to solve particular problem. It is clearly visible the huge divergence of the results for CPU and GPU implementation. Enlargement of mesh densities caused rapid decrease of CPU performance. The GPU PreAssemble implementation shows the efficiency up to 110 times over CPU MatrixFree implementation. The detailed information about benchmarks components are presented on Figure 16. Each diagram represent one benchmark for all solvers, divided by main subroutines, to evaluate it execution time. The considerable disproportion in execution time between CPU and GPU solvers, effected the graphs to be illegible. For appropriate data representation, supplementary charts were generated only for GPU solvers (see Figure 17).

Properly scaled charts for PreAssemble GPU solver revealed high disproportion in execution time between creation of FE model and solving the equation. That behaviour was unnoticed in other solvers, independently of the FE model density. That result points the weak part of TO implementation and highlight the parts of code which need more attention during further application development. This issue is clear guideline for the next steps for code optimisation. Another important factor is memory consumption by the solvers. Depends on GPU architecture and memory resources available for the computation, proper solver need to be chosen. The last

**Numerical Benchmarks**
High performance of Topology Optimisation can be achieved only by usage of well profiled and optimised FEM solver code. The numerical benchmark were performed to check Conjugate Gradient solvers efficiency, both on CPU and GPU. The benchmark was run on 4-core Intel i5-2500 3.3 Ghz CPU with 8 GB of RAM and nVidia GTX 970 as a GPU (1664 CUDA cores).
Figure 16
All benchmark scenarios with additional division on basic components.

Figure 17
Detailed data for PreAssemble (top) and MatrixFree Solver (bottom).

Figure 18
Memory usage depend of solver type and FE mesh density.
FURTHER WORK
At the time the paper is written, an intensive work is proceed on next steps of algorithm to archive satisfactory performance of Topology Optimisation. The GPU implementation of Optimality Criteria (Bendsoe and Sigmund 2004) or Method of Moving Asymptotes provided by Svanberg (Svanberg 2007) is under develop. It is predicted to achieve the final implementation of TO before ECAADE 2016 conference will run.

CONCLUSION
Fast spreading of generative design techniques allows to extend existing form finding processes with engineering solutions. Based on author’s research potentiality to adhibit structural optimisation methods towards architecture is noticeable. In conjunction with high performance of Finite Element model and assumed easiness of tool, Topology Optimisation might be indispensable part of form finding process in architecture practise. Spatial structures, as an outcome of those algorithms, can bring new and undiscovered forms to contemporary architecture. Additionally, in collaboration with additive manufacturing known as 3d printing, it moves a design process on the next level of freedom.

REFERENCES
Januszkiewicz, K 2013 'Evolutionary digital tools in designing nonlinear Shaping of concrete structures in current architecture', Concrete structures in urban areas, Wroclaw, pp. 1-6
Kutylowski, R 2004, Optymalizacja topologii kontinuum materialnego, Oficyna Wydawnicza Politechniki Wroclawskiej, Wroclaw
Martinez-Frutos, J and Herrero-Perez, D 2015, 'Efficient matrix-free GPU implementation of Fixed Grid Finite Element Analysis', Finite Elements in Analysis and Design, 104, pp. 61-71
Zwierzycki, M 2013, 'Ewolucyjne narzędzia cyfrowe w formowaniu struktur przestrzennych', Archivolta, 58(2), pp. 54-61