Proceedings of the ASME 2015 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2015
August 2 – August 5, 2015, Boston, MA, USA

DETC2015-46137

TOPOLOGY OPTIMIZATION ON THE CLOUD: A CONFLUENCE OF TECHNOLOGIES

Krishnan Suresh
suresh@engr.wisc.edu

Department of Mechanical Engineering
UW-Madison, Madison, Wisconsin 53706, USA

ABSTRACT

Topology optimization is a systematic method of generating designs to meet specific engineering requirements. It is exploited today in several industries including aircraft, automobile, and machinery, and it strongly complements the emerging field of additive manufacturing. Yet, the wide-spread use of topology optimization has been deterred due to high computational cost and significant software/hardware investment.

In this paper, we propose a cloud based topology optimization (CTO) framework to overcome these challenges, thereby promoting the wider use of topology optimization. CTO requires a confluence of several methods and technologies, each of which is discussed in this paper.

First and foremost, CTO requires a fast 3D topology optimization method that can respond rapidly to multiple clients. Here, PareTO, a topological sensitivity based method is used as the backbone of the framework. PareTO relies on limited-memory finite element analysis with a deflated linear solver that is designed to exploit multi-core and many-core architectures. At the client-end, the framework relies on JavaScript based WebGL and ThreeJS technologies to display 3D geometry and formulate structural problems within a browser. Finally, Ajax, php and HTML5 technologies are exploited to achieve asynchronous and robust user experience. An implementation of this framework is available at www.cloudtopopt.com; to use this free service, JavaScript must be enabled within the browser.

INTRODUCTION

Topology optimization has rapidly evolved from an academic exercise into an exciting discipline with numerous industrial applications [1], [2]. Applications include optimization of aircraft components [3], [4], spacecraft modules [5], automobiles components [6], and compliant mechanisms [7]–[10].

A finite element based structural topology optimization problem may be posed as (see Figure 1):

$$\begin{aligned}&\underset{\Omega\subset D}{Min}\,\varphi\\&g_i(u,\Omega)\le 0\\&\text{subject to}\\&Ku=f\end{aligned}\qquad(1.1)$$

where:

$$\begin{aligned}&\varphi:\ \text{Objective such as compliance and volume}\\&\Omega:\ \text{Topology to be computed}\\&D:\ \text{Domain within which the topology must lie}\\&.\,u:\ \text{Finite element displacement field}\\&K:\ \text{Finite element stiffness matrix}\\&f:\ \text{External force vector}\\&g_i:\ \text{Constraints}\end{aligned}\qquad(1.2)$$

In other words, the objective is to find the optimal topology, within a given design space, that minimizes a specific objective and satisfies certain design constraints. Typical objectives include volume fraction, compliance, etc., while typical constraints include stress, buckling, and manufacturing constraints.

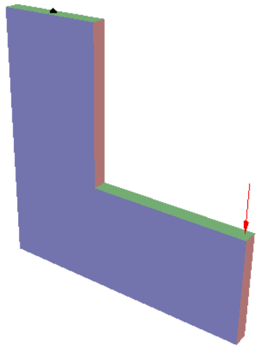


Figure 1: A STRUCTURAL PROBLEM OVER DESIGN SPACE *D*.

Various topology optimization methods such as homogenization [11], Solid Isotropic Material with Penalization (SIMP) [12], level-set [13]–[16], and evolutionary methods [17]–[19], have been proposed for solving such problems; please see [20], [21] for recent reviews. For the problem posed in Figure 1, if the objective is compliance, the optimal topology for a volume fraction of 0.5, in the absence of other constraints, is illustrated in Figure 2a. On the other hand, if the objective is the p-norm von Mises stress [22], an optimal topology is illustrated in Figure 2b.

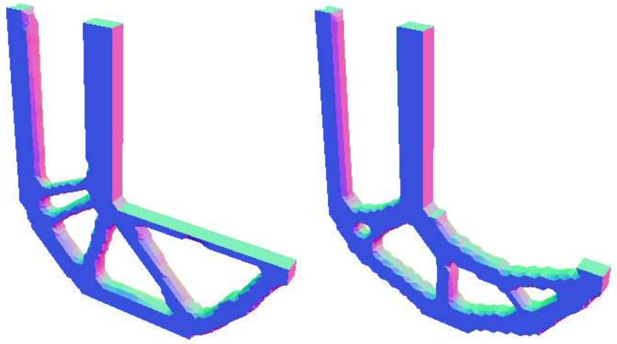


Figure 2: TOPOLOGIES THAT MINIMIZE (A) COMPLIANCE, (B) STRESS.

While the theory of topology optimization has reached a high level of maturity and has significant potential [23], its widespread use has been deterred for several reasons including high-computational cost and hardware/software investment.

Researchers have made several inroads towards popularizing topology optimization. For example, an interactive topology optimization 'app' was proposed in [24] to minimize compliance in 2D and 3D.

Along similar lines, we explore here the use of cloud computing to further promote the wider use of 3D topology optimization. Cloud computing, in essence, is time-sharing of hardware and software to deliver low-cost service. Through cloud computing, duplicated infrastructure costs are avoided, software updates are easier and service demands are evened out. For all these reasons, cloud computing in Information Technology (IT), for example, is experiencing growth rates of up to 50% per year.

Implementing a cloud based topology optimization (CTO) poses several challenges that have not been addressed by the IT industry. For example, CTO requires manipulation of 3D geometry within a browser, posing structural problem, repeated 3D finite element analysis, and robust computation of optimal topologies. This paper describes one particular strategy on how these challenges can be addressed.

Section 2 provides an overview of topology optimization methods with an emphasis on the underlying computation challenges, followed by a review of cloud computing in engineering. In Section 3, the proposed CTO framework is elaborated, and illustrated through case-studies in Section 4. Conclusions and future work are summarized in Section 5.

LITERATURE REVIEW

Topology Optimization

Among various topology optimization methods, Solid Isotropic Material with Penalization (SIMP) is perhaps the most widely used [25]. In SIMP, the domain is typically discretized via a finite element mesh, and a (pseudo) density variable is assigned to each element [12], [26]. Material properties are linked to these density-variables, and optimized to meet the desired objective. Most commercial topology optimization systems such as Optistruct [27], Genesis [28], and Atom [29] are based on SIMP.

The primary advantages of SIMP are that it is easy to implement and the theoretical foundation is well established. However, the ill-conditioning of the stiffness matrices , due to presence of low-density elements, can lead to high computational costs for iterative solvers [30], [31] in 3D, and can lead to instabilities during Eigen-mode and buckling analysis, requiring special treatment [32].

To illustrate the high computational cost, consider the edge-cantilever beam illustrated in Figure 3a; the domain is discretized using 180x60x30 8-node brick elements, resulting in about a million degrees of freedom. The objective is to find the compliance-minimizing topology of 50% volume fraction; a typical solution is illustrated in Figure 3b.

- In [30], with a specialized iterative solver with SIMP, computing the optimal topology required over 45 hours, on an AMD Opteron (2 core, with 8 GB memory).
- More recently, this problem was solved using SIMP and a direct solver in Optistruct 12.0 [27], on an Intel Xeon 12 core, with 96 GB memory, in 20 hours.

Using SIMP in a cloud based 3D topology optimization framework is therefore challenging.

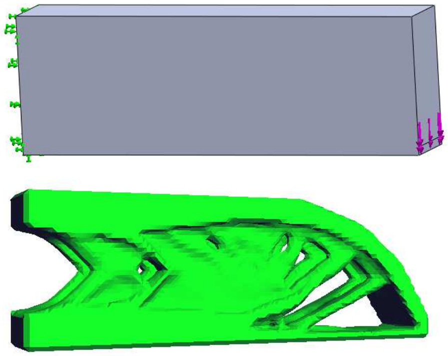


Figure 3: (A) A STRUCTURAL PROBLEM. (B) OPTIMAL TOPOLOGY.

The second strategy for topology optimization (as opposed to SIMP) is to define the evolving topology via a level-set function that is typically controlled via Hamilton-Jacobi equations [33]. An important advantage of level-set methods over SIMP is the unambiguous description of the boundary. Consequently, level-set based methods are particularly effective in boundary-dependent problems and stress-constrained topology optimization. Numerous authors have demonstrated the success of level-set methods; for example, see [34], [35], [36].

In the proposed CTO framework, we rely on a particular type of level-set method, namely PareTO [31], [37] that exploits the concept of topological sensitivity with additional computational advantages; this method is discussed later in Section 3.

Cloud Computing in Engineering

Cloud computing is essentially time sharing of computing resources, a concept that is as old as computer technology [38]. According to National Institute of Standards and Technology (NIST) [39]:

> "*Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction*."

Over the last few years, the concept of cloud has been applied across multiple engineering disciplines including (1) cloud based design, (2) cloud based finite element analysis, and (3) cloud based manufacturing. These are briefly reviewed next.

Cloud based Design

The objective in cloud based design is to provide an environment for designers to create, edit and *share* 3D CAD designs. An example of a cloud-based design environment is Autodesk's 123D [40] that provides a platform for transforming 2D photos, sharing and printing 3D designs. AutoCAD 360 [41] is a cloud based embodiment of the popular desktop AutoCAD software. Similarly, other CAD vendors such as Dassault Systems are offering cloud based design services; please see [42] for additional examples of cloud based design.

Cloud based FEA

On the other hand, in cloud based finite element analysis, the objective is to provide a high-performance finite-element computing service over the network. The advantages of a cloud based finite element service over traditional desktop computing are primarily cost and convenience. For example, the authors of [43] state that "*the installation and large-scale maintenance of these FEA tools over continuously evolving operating system (OS), processor and cluster technologies can be costly and cumbersome for the end users"* … justifying the use of cloud based FEA. Specifically, in [43], the authors characterize the performance of linear and nonlinear mechanical structural analysis workloads over multi-core and multi-node computing resources using CalculiX, an open-source FEA software. They also propose a smart scheduler for dynamic resource allocation on MPI controlled parallel architectures.

Cloud based Manufacturing

Cloud based manufacturing encompasses different services including instant quoting engines, competitive quoting from multiple manufacturers, specialized 3D printing cloud service, etc.; see [42] for a detailed discussion on cloud based manufacturing.

Cloud based Topology Optimization

While there are several desktop implementations of topology optimization, we are not aware of a cloud based implementation. However, as mentioned earlier, an interactive topology optimization 'app' was recently proposed in [24] to minimize compliance in 2D and 3D.

PROPOSED FRAMEWORK

Objectives and Guidelines

The proposed cloud based topology optimization (CTO) framework was developed using the following set of guidelines:

- Ease of Use: One of the primary goals of the CTO framework is to popularize topology optimization. Therefore the framework is kept simple by exposing only a limited number of common objectives and constraints.
- Fast, Free Service: The CTO framework is envisioned to be a free service delivering fast server response. Since high-end computer clusters are difficult to justify in a free service, CTO currently relies on a single E3-1270 (V3) Xeon workstation, equipped with 8 GB of memory. Yet, fast server response is achieved by: (1) limiting the finite element degrees of freedom to about 150,000, and (2) relying on limited-memory deflation techniques [44] and (3) fine-grain parallelism. Statistics shows that FEA operations are executed in less than 3 seconds, while topology optimization problems are solved in 10 to 150 seconds (depending on the degrees of freedom, desired volume fraction and constraints).
- Robustness: The framework must be robust in that neither the finite element analysis nor the topology optimization process should fail. To ensure robustness, incorrect problem formulations (example, incorrect boundary conditions) are first identified and corrected at the client-side, without consuming valuable server resources. Second, finite element failures, typically associated with meshing, are avoided by replacing traditional methods of mesh generation with structured-mesh generation ('voxelization'). Finally, topology optimization failures, typically due to disconnected topologies, are avoided by tracing the pareto-curve [31]; see below for details.
- Browser Independence: The framework must be accessible from any of the popular browsers including Internet Explorer, Firefox, Chrome, Safari and Opera. This is achieved here by relying on cross-platform HTML5 and WebGL. For these technologies to work, the only requirement is that JavaScript must be enabled within the browser.

In the following sections, we describe how each of these objectives has been achieved. While the current framework is limited both in the number of degrees of freedom, and nature of topology optimization problems, the core server module, namely PareTO, has been tested on problems with millions of degrees of freedom, and a variety objectives and constraints [31].

The schematic interaction between the client and server is illustrated in Figure 4.

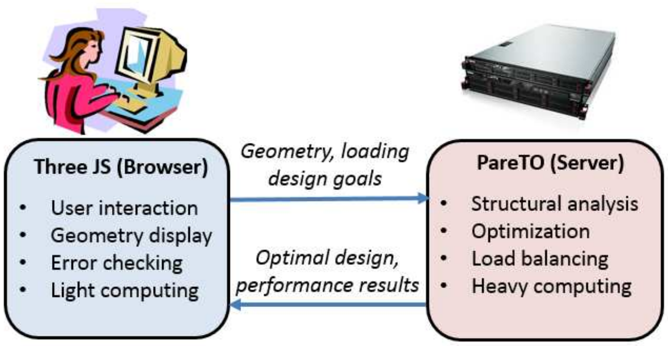


Figure 4: INTERACTION BETWEEN CLIENT AND SERVER.

Topology Optimization Problems

The generic topology optimization posed in Equation (1.1) captures a large class of multi-load, multi-constraint problems. However, as stated earlier, only a limited class of problems is exposed in the CTO framework.

The first class of problems exposed in the CTO framework include compliance minimization problems subject to volume, displacement and stress constraints:

$$\begin{aligned} &\underset{\Omega\subset D}{Min}\, J \\ &\left|\Omega\right| \geq \alpha \mid D \mid \\ &\delta \leq \beta\delta_0 \\ &\sigma \leq \gamma\sigma_0 \\ &\text{subject to} \\ &Ku = f \end{aligned} \tag{3.1}$$

where (also see Equation (1.2)):

$$\begin{aligned} &J: \text{ Compliance} \\ &\delta_0: \text{ Initial max displacement} \\ &\sigma_0: \text{ Initial max von Mises stress} \\ &\alpha,\beta,\gamma: \text{ User specified constants; } \alpha<1;\beta,\gamma>1 \end{aligned} \tag{3.2}$$

Observe in Equation (3.1) that the constraints are imposed relative to the initial volume, displacement and stress. The optimization process will terminate if *any* of the constraints are violated.

As a special case of Equation (3.1), the classic 'unconstrained' problem with a target 0.5 volume fraction, may be imposed by choosing large values for $\beta \,\&\, \gamma$:

$$\begin{aligned} &\underset{\Omega\subset D}{Min}\, J \\ &\left|\Omega\right| \geq 0.5 \mid D \mid \\ &\delta \leq 1000\delta_0 \\ &\sigma \leq 1000\sigma_0 \\ &\text{subject to} \\ &Ku = f \end{aligned} \tag{3.3}$$

To solve the above problem, the pareto-curve involving the compliance and volume fraction is traced starting with a volume fraction of 1.0. For example, Figure 5 illustrates the pareto-optimal curve and the corresponding topologies, for a specific instance of Equation (3.1). Tracing the pareto curve guarantees that the intermediate topologies are also optimal [37], ensuring the robustness of the algorithm.

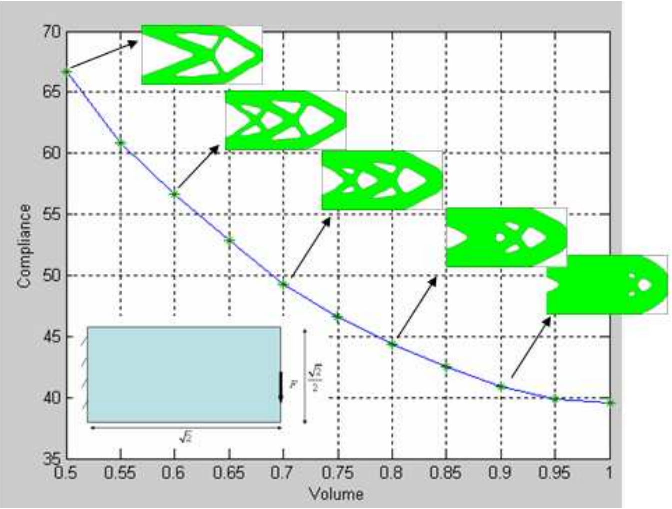


Figure 5: THE PARETO-OPTIMAL CURVE AND TOPOLOGIES.

Constraints can be imposed as follows:

$$\begin{aligned} &\underset{\Omega\subset D}{Min}\, J \\ &\left|\Omega\right| \geq 0.5 \mid D \mid \\ &\delta \leq 3\delta_0 \\ &\sigma \leq 2\sigma_0 \\ &\text{subject to} \\ &Ku = f \end{aligned} \tag{3.4}$$

The optimization process will terminate if: (1) 0.5 volume fraction is reached, *or* (2) if the displacement reaches three times the initial displacement, *or* (3) if the maximum von Mises stress reaches twice the initial maximum von Mises stress.

The second class of problems exposed in the CTO frameworks includes stress minimization problem, subject to similar constraints:

$$\begin{aligned} &\underset{\Omega\subset D}{Min}\, S \\ &\left|\Omega\right| \geq \alpha \mid D \mid \\ &\delta \leq \beta\delta_0 \\ &\sigma \leq \gamma\sigma_0 \\ &\text{subject to} \\ &Ku = f \end{aligned} \tag{3.5}$$

In Equation (3.5), S is the p-norm von Mises stress [22], with the value of p-norm set to 6. Once again, to solve the above problem, the pareto-curve involving the p-norm stress and the volume fraction is traced. The two classes of problems are compared, for example, in [22]; illustrative case studies are discussed later in the paper.

Robustness

As stated earlier, one of the primary reasons today for lack of robustness in finite element analysis is the generation of the underlying mesh. Despite decades of research, conforming mesh generation continues to be a computationally expensive and fragile process [45].

To avoid such failures, we rely here on structured meshes, also referred to as 'voxels' that can be easily computed through *voxelization* [46]. Voxel representation is robust, simple, and is particularly well-suited for fast FEA [43].

However, voxelization can lead to stress fluctuations as the mesh is refined, and is therefore rarely used in product *verification*, i.e., during the final stages of product design. For topology optimization, when product designs are being *conceptualized*, the accuracy requirements are less stringent, and a voxel mesh often suffices as the case studies later support. Thus, the advantages of voxelization far exceeds its disadvantages in the current context.

Fast Finite Element Analysis

The primary computational bottle-neck in topology optimization is finite element analysis (FEA), and much of the computational cost lies in FEA is the solution of linear systems of equations.

Direct solvers [47] are the default choice today for solving such linear systems. They are robust and well-understood, and rely on factoring the stiffness matrix into Cholesky decomposition. However, due to the explicit factorization, direct solvers are memory intensive [48]. Since memory-access is often the bottle-neck in computer architecture, this translates into an increased computational time. To avoid such bottlenecks, we resort here to iterative solvers that do not factorize the stiffness matrix, but compute the solution iteratively [49]. Further, since ParetTO does not rely on pseudo-densities (as in SIMP), all elements are either 'in' or 'out'. This combined with pareto-tracing makes the stiffness matrices inherently better conditioned, leading to faster convergence of iterative solvers; see [50] for a comparison of condition numbers in SIMP and ParetTO.

To further accelerate the iterative solution, we employ here an assembly-free version of the deflated conjugate gradient [44], [51], [52], where neither the stiffness matrix nor the deflation matrix is assembled. The resulting implementation is particularly well suited for parallelization, and can be easily ported to multi-core CPU and GPU architectures. Since the GPU implementation is beneficial for problems with million degrees of freedom or more, it is not supported currently in the CTO framework.

Parallelization on the multicore Xeon CPU was attained through OpenMP commands (www.openmp.org). In the current implementation, typical FEA operation is executed in less than 3 seconds (often less than 1 second) of server time for problems up to 150,000 degrees of freedom.

Browser-Independence

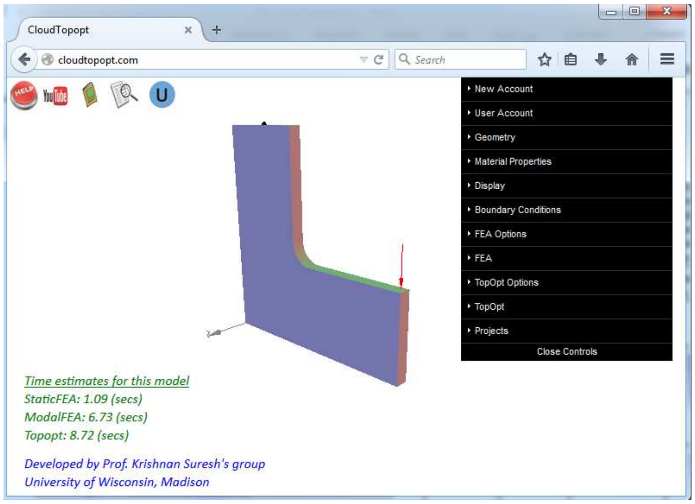
An essential aspect of cloud-based topology optimization is the ability to display and manipulate 3D geometry within a browser. We exploit here JavaScript based WebGL (Web Graphics Library and ThreeJS.

WebGL, a technology similar to traditional desktop OpenGL, is a JavaScript API for rendering interactive 3D graphics within modern browsers. It can be mixed with other HTML elements, and is designed and maintained by the non-profit Khronos Group (www.khronos.org).

Since WebGL is a low-level language, ThreeJS was recently developed as an extension (see www.threejs.org). It provides high-level language constructs, for example, to transform 3D geometry, and is currently supported by almost all browsers (except for a few features in Internet Explorer).

ThreeJS internally represents geometry as a collection of triangles. A standard off-the-shelf software library (such as CADlook; www.cadlook.com) can be used for converting standard computer aided design (CAD) representations such as IGES into triangle-based STereoLithography (STL) representation.

Figure 6 illustrates an implementation of CTO, as viewed within Mozilla Firefox browser; it is currently hosted at www.cloudtopopt.com.



Time estimates for this model
StaticFEA: 1.09 (secs)
ModalFEA: 6.73 (secs)
Topopt: 8.72 (secs)

Developed by Prof. Krishnan Suresh's group
University of Wisconsin, Madison

Figure 6: WWW.CLOUDTOPOPT.COM IN MOZILLA FIREFOX.

CASE STUDIES

The cloud based topology optimization (CTO) framework is illustrated here through a few case studies. The material properties used in all examples are those of steel, with:

$$\begin{aligned} &E = 2.1*10^{11} \quad N/m^2 \\ &\nu = 0.28 \end{aligned} \tag{4.1}$$

The server is a quad-core 3.4 GHz E3-1270 (V3) Xeon workstation, equipped with 8 GB of memory. On the client side, Mozilla Firefox, Version 34.0.5 was used for all the experiments.

Compliance Minimization

The first example is that of a compliance minimization problem over a standard L-bracket with cross-sectional dimensions shown in Figure 7, and 0.006 meters thickness. The bracket is fixed on the top face, and a load of 5000 N is applied as shown.

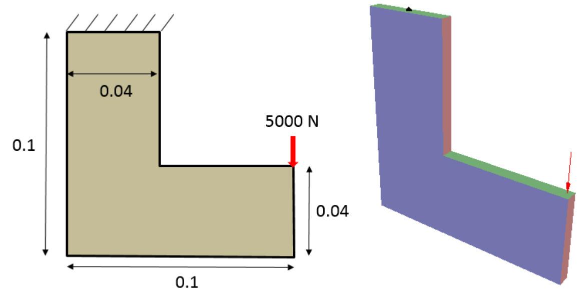


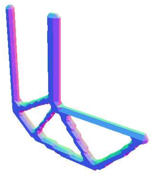
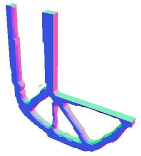
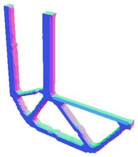
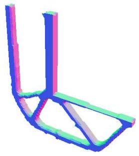
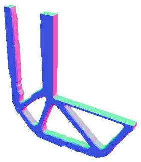
Figure 7: A STRUCTURAL PROBLEM OVER L-BRACKET.

The topology optimization problem is:

$$\begin{aligned}
&\underset{\Omega\subset D}{Min}\, J \\
&\left|\Omega\right| \geq 0.01\,|\,D\,| \\
&\delta \leq 2.5\delta_0 \\
&\sigma \leq 1000\sigma_0 \\
&\text{subject to} \\
&Ku = f
\end{aligned} \qquad (4.2)$$

Thus, in practical terms, a displacement constraint is imposed, and the topology with the lowest volume fraction is desired. The geometry is discretized into structured hexahedral elements (voxels) ranging from 5,000 to 50,000 elements, and the problem solved for each of the mesh sizes. The results are summarized in Table 1; the consistency in the computed topology can be observed. Thus, the computed topology is relatively insensitive to the voxelization error.

Table 1: NUMBER OF MESH ELEMENTS, FINAL VOLUME FRACTION, TIME TAKEN AND COMPUTED TOPOLOGIES.

Number of Mesh Elements	Final Volume Fraction	#FEAs and time taken	Computed Topology
5,000	0.28	83 FEAs 8.96 secs	
10,000	0.32	82 FEAs 19.4 secs	
15,000	0.30	91 FEAs 26.7 secs	
20,000	0.27	89 FEAs 36 secs	
50,000	0.29	90 FEAs 151 secs	

Stress Minimization

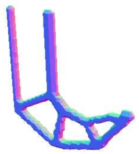
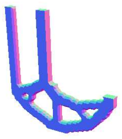
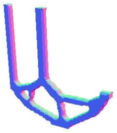
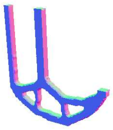
Next, instead of minimizing compliance, the p-norm von Mises stress [22] was minimized with the following constraints:

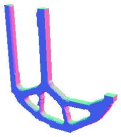
$$\begin{aligned}
&\underset{\Omega\subset D}{Min}\, S \\
&\left|\Omega\right| \geq 0.01\,|\,D\,| \\
&\delta \leq 1000\delta_0 \\
&\sigma \leq 1.25\sigma_0 \\
&\text{subject to} \\
&Ku = f
\end{aligned} \qquad (4.3)$$

Here a stress constraint is imposed, and the topology with the lowest volume fraction is desired. The final results for various mesh sizes are summarized in Table 2.

For stress minimization, an adjoint problem must be solved at each step of the optimization process, doubling the number of FEAs and computational cost.

Table 2: NUMBER OF MESH ELEMENTS, FINAL VOLUME FRACTION, TIME TAKEN AND COMPUTED TOPOLOGIES.

Number of Mesh Elements	Final Volume Fraction	#FEAs and time taken	Computed Topology
5,000	0.33	224 FEAs 13.2 secs	
10,000	0.38	176 FEAs 20.9 secs	
15,000	0.37	194 FEAs 28.8 secs	
20,000	0.37	206 FEAs 48 secs	

50,000	0.38	203 FEAs 231 secs	

Pareto Optimal Designs

Recall that the ParеTO algorithm generates multiple topologies of decreasing volume fractions; such topologies can provide key insights to the designer. As an illustrative example, consider the structural problem illustrated in Figure 8a. The following topology optimization is posed, where the objective is to generate optimal topologies up to a volume fraction of 0.2, with no other constraint:

$$\begin{aligned} &\underset{\Omega\subset D}{Min}\, J\\ &\left|\Omega\right| \ge 0.2\,|\,D\,|\\ &\delta \le 1000\delta_0\\ &\sigma \le 1000\sigma_0\\ &\text{subject to}\\ &Ku = f \end{aligned} \tag{4.4}$$

The geometry is discretized with 20,000 elements; an intermediate topology for a volume fraction of 0.3 is illustrated in Figure 8b. The final topology at a volume fraction of 0.2 is illustrated in Figure 8c.

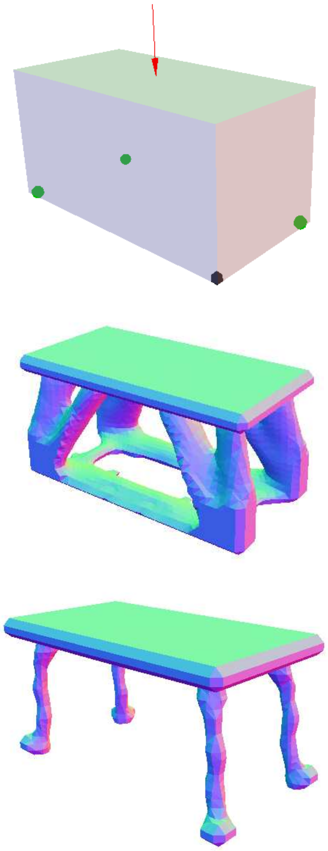


Figure 8: (A) THE TABLE PROBLEM. (B) TABLE DESIGN AT A VOLUME FRACTION OF 0.3. (C) TABLE DESIGN AT A VOLUME FRACTION OF 0.2.

Draw Constraints

One of the options exposed in the CTO framework is the ability to impose 'draw-direction' constraint during optimization. As an illustrative example, consider the edge cantilever problem illustrated in Figure 9a (see [31] for details); the geometry is discretized using 20,000 elements, and the following topology optimization is posed:

$$\begin{aligned} &\underset{\Omega\subset D}{Min}\, J\\ &\left|\Omega\right| \ge 0.5\,|\,D\,|\\ &\delta \le 1000\delta_0\\ &\sigma \le 1000\sigma_0\\ &\text{subject to}\\ &Ku = f \end{aligned} \tag{4.5}$$

The topology computed, in 20 seconds, is illustrated in Figure 9b. Observe that the topology contains 'pockets' in the thickness direction. This may not be desirable in some applications. If one imposes a draw-direction along the thickness direction, the final topology, once again computed in 20 seconds, is illustrated in Figure 9c.

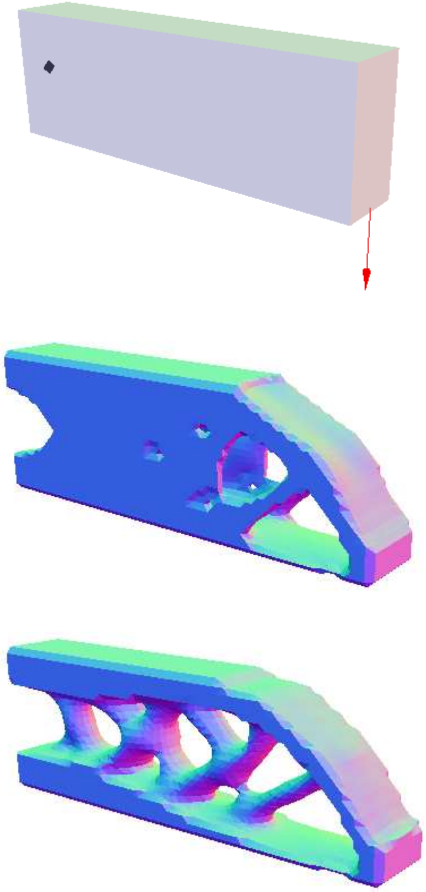


Figure 9: (A) EDGE CANTILEVER PROBLEM. (B) FINAL TOPOLOGY WITH NO CONSTRAINTS, AT A VOLUME FRACTION OF 0.5. (C) FINAL TOPOLOGY WITH DRAW-DIRECTION IMPOSED ALONG THE THICKNESS, AT A VOLUME FRACTION OF 0.5.

Potential Users

Topology optimization brings together computer-aided-design (CAD) and computer-aided-engineering (CAE) users. Thus, the primary users of this service is envisioned to be design engineers who are focused on the CAD/CAE interface. The CTO framework will hopefully reduce the product development cycle time.

Besides, we believe CTO will address a new emerging market of 3-D printing. The new technology of 3D-printing (also referred to additive manufacturing) is revolutionizing the world of fabrication. The most significant benefit of 3d-printing is that geometric-complexity is 'free', i.e., to a large extent, it costs no more time or money to fabricate a very complicated part than it takes to fabricate a simple one. This opens new opportunities in product design in that one can rapidly design parts on the cloud, and directly fabricate these through 3d-printing.

For example, Figure 10 illustrates an example where a design was optimized using the CTO framework, and then directly printed on a low-cost 3D-printer.

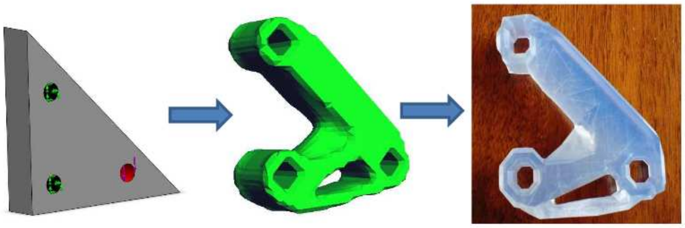


Figure 10: FROM PROBLEM SPECIFICATION OPTIMAL PROTOTYPE.

Just as 3D-printing has leveled the playing-field in the world of manufacturing, cloud-based topology optimization will level the playing-field for designers.

CONCLUSIONS

In this paper, we discussed various technologies that underlie a cloud based topology optimization framework (CTO), hosted at www.cloudtopopt.com. The CTO framework provides a simple, fast and free 3D topology optimization service, thereby promoting the wider use of topology optimization in product design.

Future work will focus on porting this to a high-performance server [53] such as Amazon Web Service (AWS), and exposing multi-load optimization, with buckling and modal constraints.

Future works will also focus on implementing cloud security controls (some are already in place) into the proposed CTO framework. We will explore the suitability of various controls such as:

- Deterrent controls: These are simple controls that inform potential attackers of adverse consequences; these are obviously the first level and lowest-cost of defense.
- Preventive controls: Preventive controls such as strong authentication of cloud users will make it less likely for unauthorized entry and access to data. SSL technology will be incorporated for strong authentication.
- Detective controls: Detective controls, such as system and network security monitoring, will be put in place to detect and react rapidly to intrusions.
- Corrective controls: Corrective controls, such as temporary cloud lock-down, reduce the consequences of repeated attacks.

Acknowledgements

The author would like to thank the support of National Science Foundation through grants CMMI-1232508 and CMMI-1161474.

REFERENCES

[1] H. A. Eschenauer, V. V. Kobelev, and A. Schumacher, "Bubble method for topology and shape optimization of structures," *Structural Optimization*, vol. 8, pp. 42–51, 1994.

[2] G. I. N. Rozvany, "A critical review of established methods of structural topology optimization," *Structural and Multidisciplinary Optimization*, vol. 37, no. 3, pp. 217–237, 2009.

[3] E. Kesseler, "Multidisciplinary design analysis and multi-objective optimisation applied to aircraft wing," *WSEAS transactions on systems and Control and Cybernetics*, vol. 1, no. 2, p. 221 227, 2006.

[4] J. J. Alonso, "Aircraft design optimization," *Mathematics and Computers in Simulation*, vol. 79, no. 6, pp. 1948–1958, 2009.

[5] V. H. Coverstone-Carroll, "Optimal multi-objective low-thrust spacecraft trajectories," *Comput. Methods Appl. Mech. Eng.*, vol. 186, pp. 387–402, 2000.

[6] L. Wang, "Automobile body reinforcement by finite element optimization," *Finite Elements in Analysis and Design*, vol. 40, no. 8, pp. 879 – 893, 2004.

[7] G. K. Ananthasuresh, S. Kota, and Y. Gianchandani, "A methodical approach to the design of compliant micromechanisms," in *Solid State Sensor and Actuator Workshop*, 1994, pp. 189–192.

[8] S. Nishiwaki, "Topology Optimization of Compliant Mechanisms using the Homogenization Method," *International Journal for Numerical Methods in Engineering*, vol. 42, pp. 535–559, 1998.

[9] T. E. Bruns and D. A. Tortorelli, "Topology optimization of non-linear elastic structures and compliant mechanisms," *Computer Methods in Applied Mechanics and Engineering*, vol. 190, no. 26–27, pp. 3443–3459, 2001.

[10] Z. Luo, "Compliant mechanism design using multi-objective topology optimization scheme of continuum structures," *Structural and Multidisciplinary Optimization*, vol. 30, pp. 142–154, 2005.

[11] M. P. Bendsøe and N. Kikuchi, "Generating optimal topologies in structural design using a homogenization method," *Computer Methods in Applied Mechanics and Engineering*, vol. 71, pp. 197–224, 1988.

[12] O. Sigmund, "A 99 line topology optimization code written in Matlab," *Structural and Multidisciplinary Optimization*, vol. 21, no. 2, pp. 120–127, 2001.

[13] G. Allaire, F. Jouve, and A. M. Toader, “A level-set method for shape optimization,” *Comptes Rendus Mathematique*, vol. 334, no. 12, pp. 1125–1130, 2002.

[14] G. Allaire and F. Jouve, “A level-set method for vibration and multiple loads structural optimization,” *Structural and Design Optimization*, vol. 194, no. 30–33, pp. 3269–3290, 2005.

[15] L. He, C.-Y. Kao, and S. Osher, “Incorporating topological derivatives into shape derivatives based level set methods,” *Journal of Computational Physics*, vol. 225, no. 1, pp. 891–909, 2007.

[16] M. Y. Wang, X. Wang, and D. Guo, “A level set method for structural topology optimization,” *Computer Methods in Applied Mechanics and Engineering*, vol. 192, pp. 227–246, 2003.

[17] H. Hamda, “Application of a multi-objective evolutionary algorithm to topology optimum design,” in *Fifth International Conference on Adaptive Computing in Design and Manufacture*, 2002.

[18] Y. M. Xie, “A simple evolutionary procedure for structural optimization,” *Computers and Structures*, vol. 49, no. 5, pp. 885–896, 1993.

[19] Y. M. Xie and G. P. Steven, *Evolutionary Structural Optimization*, 1st ed. Berlin: Springer-Verlag, 1997.

[20] N. P. van Dijk, K. Maute, M. Langelaar, and F. van Keulen, “Level-set methods for structural topology optimization: a review,” *Structural and Multidisciplinary Optimization*, vol. 48, no. 3, pp. 437–472, 2013.

[21] O. Sigmund and K. Maute, “Topology Optimization Approaches,” *Structural and Multidisciplinary Optimization*, vol. 48, no. 6, pp. 1031–1055, 2013.

[22] K. Suresh and M. Takalloozadeh, “Stress-Constrained Topology Optimization: A Topological Level-Set Approach,” *Structural and Multidisciplinary Optimization*, vol. 48, no. 2, pp. 295–309, 2013.

[23] B. Anna-Lena, A. Brezing, and J. Feldhusen, “The potential of low cost topology optimization,” in *DS 76: Proceedings of E&PDE 2013, the 15th International Conference on Engineering and Product Design Education*, Dublin, Ireland, 2013.

[24] N. Aage, M. N. Jørgensen, C. S. Andreasen, and O. Sigmund, “Interactive topology optimization on hand-held devices,” *Structural and Multidisciplinary Optimization*, vol. 47, no. 1, pp. 1–6, 2013.

[25] M. P. Bendsoe and O. Sigmund, *Topology Optimization: Theory, Methods and Application*, 2nd ed. Springer, 2003.

[26] E. Andreassen, A. Clausen, M. Schevenels, and O. Sigmund, “Efficient Topology Optimization in MATLAB Using 88 Lines of Code,” *Structural and Multidisciplinary Optimization*, vol. 43, pp. 1–16, 2011.

[27] Altair, *Optistruct; http://www.altairhyperworks.com*. 2013.

[28] Vrand, *Genesis; http://vrand.com/Genesis.html*. 2011.

[29] *ATOM*. Simulia.

[30] S. Wang, E. D. Sturler, and G. Paulino, “Large-scale topology optimization using preconditioned Krylov subspace methods with recycling,” *International Journal for Numerical Methods in Engineering*, vol. 69, no. 12, pp. 2441–2468, 2007.

[31] K. Suresh, “Efficient Generation of Large-Scale Pareto-Optimal Topologies,” *Structural and Multidisciplinary Optimization*, vol. 47, no. 1, pp. 49–61, 2013.

[32] N. L. Pedersen, “Maximization of eigenvalues using topology optimization,” *Structural and Multidisciplinary Optimization*, vol. 20, no. 1, pp. 2–11, 2000.

[33] N. P. van Dijk, K. Maute, M. Langelaar, and F. van Keulen, “Level-set methods for structural topology optimization: a review,” *Structural and Multidisciplinary Optimization*, vol. 10.1007/s00158–013–0912-y, 2013.

[34] G. Allaire and F. Jouve, “Minimum stress optimal design with the level set method,” *Engineering Analysis with Boundary Elements*, vol. 32, no. 11, pp. 909–918, 2008.

[35] K. A. James, E. Lee, and J. R. R. A. Martins, “Stress-based topology optimization using an isoparametric level set method,” *Finite Elements in Analysis and Design*, vol. 58, pp. 20–30, 2012.

[36] Q. Xia, T. Shi, S. Liu, and M. Y. Wang, “A level set solution to the stress-based structural shape and topology optimization,” *Computers & Structures*, vol. 90–91, pp. 55–64, 2012.

[37] K. Suresh, “A 199-line Matlab code for Pareto-optimal tracing in topology optimization,” *Structural and Multidisciplinary Optimization*, vol. 42, no. 5, pp. 665–679, 2010.

[38] C. Strachey, “Time Sharing in Large Fast Computers,” in *Proceedings of the International Conference on Information processing, UNESCO. paper B.2.19*, 1959, pp. 336–341.

[39] P. Mell and T. Grance, “The NIST Definition of Cloud Computing,” National Institute of Standards and Technology, Gaithersburg, Maryland USA, Special Publication 800 - 145, 2011.

[40] Autodesk, *Autodesk-123D*. http://www.123dapp.com, 2014.

[41] Autodesk, *Autocad360*. www.autodesk360.com, 2014.

[42] D. Wu, D. W. Rosen, L. Wang, and D. Schafer, “Cloud-based Design and Manufacturing: A New Paradigm in Digital Manufacturing and Design Innovation,” *Computer Aided Design*, vol. 59, pp. 1–14, 2015.

[43] I. Ari and N. Muhtaroglu, “Design and implementation of a cloud computing service for finite element analysis,” *Advances in Engineering Software*, vol. 60–61, pp. 122–135, 2013.

[44] P. Yadav and K. Suresh, “Limited-Memory Deflated Conjugate Gradient for Solid Mechanics,” in *Proceedings of the ASME 2014 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Buffalo, NY, USA, 2014.

[45] A. V. Mobley, “An object oriented approach to geometry defeaturing for finite element meshing,” in *7th International Meshing Roundtable*, Dearborn, Michigan, 1998, pp. 547–563.

[46] E. A. Karabassi, G. Papaioannou, and T. Theoharis, “A Fast Depth-Buffer-Based Voxelization Algorithm,” *Journal of Graphics Tools*, vol. 4, no. 4, pp. 5–10, 1999.

[47] G. H. Golub, *Matrix Computations*. Balitmore: Johns Hopkins, 1996.

[48] R. Aubry, F. Mut, S. Dey, and R. Lohner, "Deflated preconditioned conjugate gradient solvers for linear elasticity," *International Journal for Numerical Methods in Engineering*, vol. 88, pp. 1112–1127, 2011.

[49] Y. Saad, *Iterative Methods for Sparse Linear Systems*. SIAM, 2003.

[50] S. Deng and K. Suresh, "Multi-constrained topology optimization via the topological sensitivity," *Structural and Multidisciplinary Optimization*, vol. Accepted, 2014.

[51] P. Yadav and K. Suresh, "Large Scale Finite Element Analysis Via Assembly-Free Deflated Conjugate Gradient," *J. Comput. Inf. Sci. Eng*, vol. 14, no. 4, pp. 041008–1: 041008–9, 2014.

[52] K. Suresh and P. Yadav, "Large-Scale Modal Analysis on Multi-Core Architectures," in *Proceedings of the ASME 2012 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Chicago, IL, 2012.

[53] V. Mauch, M. Kunze, and M. Hillenbrand, "High performance cloud computing," *Future Generation Computer Systems*, vol. 29, pp. 1408–1416, 2013.