Topology Optimization with FEniCS

Zachary Ferguson
New York University
zfergus@nyu.edu

Francis Williams
New York University
francis.williams@nyu.edu

Abstract

We present an implementation of topology optimization with a linear elasticity solver using FEniCS. We approach the problem of minimizing compliance using solid isotropic material with penalization (SIMP). We generate an optimal structure given a load condition by optimizing the relative density of each element. To verify our solutions, we perform finite element analysis using FEniCS and conduct physical experimentation using fabricated instances of our results.

1. Introduction

In mechanical engineering, we often wish to design physical structures which can withstand certain forces known a priori. For example, when designing a building, an engineer would want to place concrete walls in a way that the structure can support the downward force caused by the roof. In addition to support requirements, engineers are often faced with budgetary requirements on the amount of material they can use. In other words, they must design a structure that can withstand a set of forces while using a limited amount of material.

We solve this design problem algorithmically using *Topology Optimization* [5], an optimization technique where the topology of the final structure is optimized for the given loads. We present a Python implementation of the topology optimization algorithm in [4] as well as finite-element simulations of our generated results using *FEniCS* [2]. In addition, we conduct physical experimentation on results fabricated via 3D printing and laser cutting.

2. Linear Elasticity

Linear elasticity models small deformations under given load conditions. The equations of linear elasticity are given as

$$-\nabla \cdot \sigma = f \tag{1}$$

$$\sigma = \lambda \, \text{tr}(\varepsilon) I + 2\mu\varepsilon \tag{2}$$

$$\varepsilon = \frac{1}{2}(\nabla u + (\nabla u)^\top) \tag{3}$$

where σ is the strain tensor, f is the body forces, λ and μ are Lamé's elasticity parameters, ε is the symmetric strain-rate tensor, and u is the displacement vector field [1]. Equation (2) is known as the constitutive equation, and Equation (3) the strain-displacement equation. Equation (2) and (3) can be combined to give a more concise definition:

$$\sigma = \lambda(\nabla \cdot u)I + \mu(\nabla u + (\nabla u)^\top) \tag{4}$$

These linearized equations describe a small elastic deformation. For larger displacements or more complex materials a full nonlinear equation is required. To solve nonlinear elasticity, Newton's method is used. This is, however, beyond the scope of this paper. For our optimization and simulation we restrict the problem to be isometric material with small displacements. Under these restrictions, linear elasticity models our problem well.

3. Topology Optimization

Our design problem can be viewed as a constraint optimization problem. Informally, we wish to place material in some domain such that the structure formed by the material can withstand a set of forces while keeping the total material usage under some threshold.

Each element e is assigned a density x_e that determines the Young's modulus [1] E_e.

$$E_e(x_e) = E_{\min} + x_e^p(E_0 - E_{\min}), \quad x_e \in [0,1] \tag{5}$$

E_0 is the maximal stiffness of each element, and $E_{\min}$ is a small value assigned to the stiffness of the void region ($x_e = 0$) to avoid a singular stiffness matrix [4]. The penalization

[1] Young's modulus is a numerical constant that describes the tensile elasticity of a solid. The tendency of the solid to deform along an axis with opposing forces applied along the axis.

factor, p, forces x_e to be binary (either 0 or 1). This works well for a penalization factor $p \geq 3$ [8]. [2]

We view the design problem as minimizing *compliance* at each element in the domain subject to constraints on the volume of material placed at each element.

$$\min_{x} : c(x) = U^T K U = \sum_{e=1}^{N} E_e(x_e) \mathbf{u}_e^T \mathbf{k}_0 \mathbf{u}_e \quad (6)$$

$$\text{subject to} : \frac{V(x)}{V_0} = f \quad (7)$$

$$: KU = F \quad (8)$$

$$: 0 < x_{\min} \leq x \leq 1 \quad (9)$$

U is is the displacement vector of each element in the domain, F is the force vector applied to each element in the domain, $V(x)$ is the volume of material placed in each element as a function of the material density, V_0 is the volume of the domain, and K is the global stiffness matrix arising from the linear elasticity finite element problem on the input domain. The variable $x_{\min}$ is a minimum density of material per element designed to avoid numerical issues with zero density material.

3.1. Implementation

To implement the above equations, we discretize our domain as a 2D box with square elements (a uniform grid). Given our domain choice, we use linear elements to compute the stiffness matrix K. The input force field is defined as a set of vectors on the nodes of the grid (corners of each element). To minimize equation (6), we use the Method of Moving Asymptotes (MMA) [9], a convex simplification approach, implemented in NLopt [7].[3]

At each iteration of the nonlinear optimization we compute the compliance by solving for a displacement $U = K^{-1}F$. K is stored as a sparse matrix for memory efficiency. CHOLMOD's [3] sparse linear solver is used to solve this linear equation efficiently.

To direct the MMA solver, we compute the gradient of the compliance. The gradient of (6) is given as [4]

$$\frac{\partial c}{\partial x_e} = -p x_e^{p-1} (E_0 - E_{\min}) \mathbf{u}_e^T \mathbf{k}_0 \mathbf{u}_e \quad (10)$$

3.2. Results

Figures 2 and 3 are the results of our optimization on the MBB beam [8]. The boundary conditions for the MBB beam are a point load with fixed lower corners as described in Figure 1. This design problem models a bridge over a gap where only supports on the ends are achievable.

[2]All of our results use a penalization factor of $p = 3$.

[3]All the source code and simulation models are open source and can be found at *https://github.com/zfergus/fenics-topopt*.

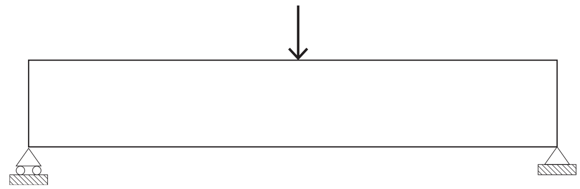

Figure 1. MBB beam with boundary conditions. Arrows are force vectors, and striping indicate fixed points.

Figure 2. An optimized bridge structure at a resolution of 360×60 with a single point load in the top center and constrained supports at the bottom corners. The optimization naturally tends towards to creating strut-like structures.

Figure 3. An optimized bridge structure at a resolution of 600×100 with a distributed load around the center top quarters and constrained supports at the bottom corners. With higher resolution the optimization will place more fine struts.

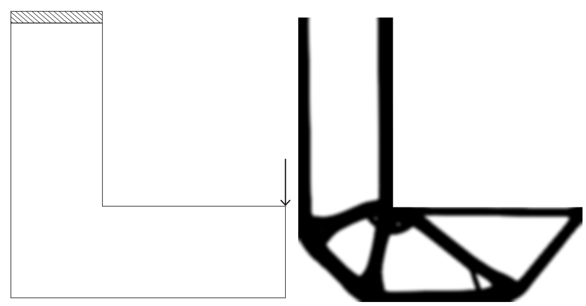

Figure 4. Left: L-bracket boundary conditions with a fixed top boundary and a point load on the right. Results of our topology optimization on the L-bracket boundary conditions with a volume fraction of 36%. We use a passive element in the upper right corner to prevent material from filling the area.

We also ran our optimization on the boundary conditions described in Figure 4, an L-bracket. With a volume constraint of 36% of the design volume available, the optimization finds the optimal placement to minimize compliance due to the load. Our optimization does not account for stress, so sharp features are not avoided (e.g. inside corner).

4. Finite Element Simulation

To verify the output structures from our implementation can withstand the prescribed forces, we ran a finite-element simulation using FEniCS [1], a finite element package.

Finite elements methods require a discretization of the domain into elements. FEniCS restricts the domain to simplices

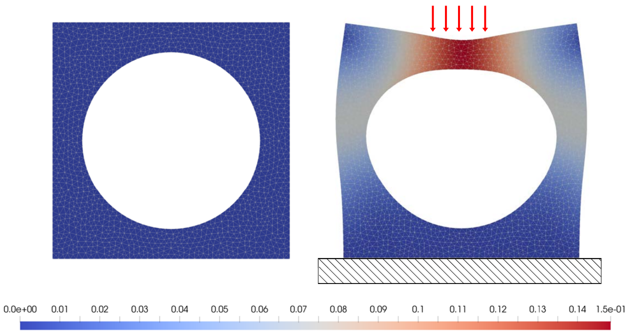

Figure 5. Left: Initial geometry meshed using GMSH. Right: The same geometry under a distributed load (red arrows) with constrains on the position of the bottom edge (stripe pattern). The displacements are drawn after running the linear elasticity finite element simulation.

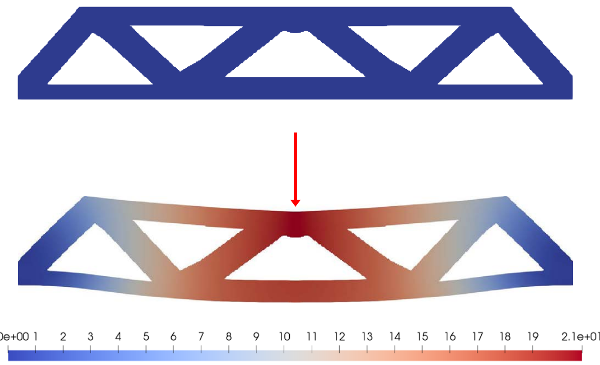

Figure 6. Top: Results of the topology optimization meshed from a black and white raster image. Bottom: The same geometry under a single point load (red arrow). The displacements are drawn after running the linear elasticity finite element simulation.

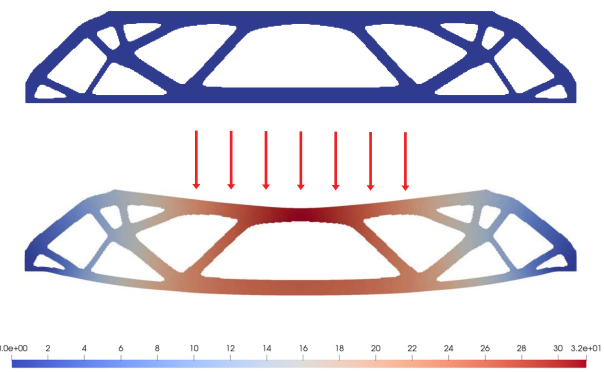

Figure 7. Top: Results of the topology optimization meshed from a black and white raster image. Bottom: The same geometry under a distributed load (red arrows). The displacements are drawn after running the linear elasticity finite element simulation.

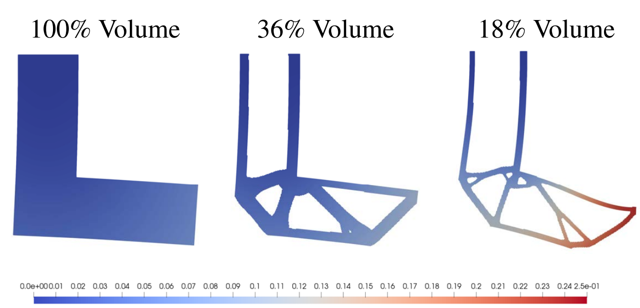

Figure 8. The results of simulating the L-bracket shape. Left: 100% material fill, a costly domain, as a point of comparison. Center: meshed results of our topology optimization with a constrain of 36% of the total volume. Right: meshed result of our topology optimization with a constraint of 18% of the total volume. As the amount of material decreases the structure becomes less resilient.

(triangles in 2D and tetrahedron in 3D). Additionally, the problem is defined by boundary conditions. In the case of linear elasticity these boundary conditions are the fixed nodes and the loads applied to nodes. Fixed nodes are enforce as Dirichlet boundaries with a displacement value of 0. Loads, or forces, are defined as Neumann boundary conditions.

Figure 5 is an example linear elastic problem. The domain is meshed by triangles. The forces are applied to some top points, and the bottom boundary is fixed. Displacements are then solved for and the magnitude of the displacement vectors are drawn as colors.

To verify our optimization, we constructed a triangle mesh of our black and white density topology optimization results by splitting each quadrilateral in the optimization into two triangles. We then ran the linear elasticity simulation in FEniCS. Firgues 6 and 7 show the results of our MBB beam (bridge) optimization. In Figure 6, only a single point load is applied to the center of the structure. In Figure 7, a distributed load is applied to the center half of the bridge. Because the increased force caused more deformation, we increased the Young's modulus in Figure 7. The Young's modulus controls the stiffness of the material, and increasing the value corresponds to a stiffer material less susceptible to deformations.

An important part of topology optimization is the volume constraints. Without the volume constraint the optimal solution is a solid, 100% volume usage, structure. By enforcing a limitation on the amount of material, the optimization finds an optimal structure with the allowed volume. Figure 8 illustrated the effects of material usage on topologically optimal structures.

5. Fabricated Experiments

As a final verification, we fabricated the two bridge models. The first by slightly extruding it in 3D and 3D-printing it using an Ultimaker 3 [10]. The second by laser cutting a sheet of $\frac{1}{8}$" acrylic using an Epilog Mini 24 Laser [6]. Figures 9 and 10 show that the fabricated objects can withstand a downward force as designed.

Figure 9. The 3D printed bridge model generated by our algorithm both with (bottom) and without (top) additional mass.

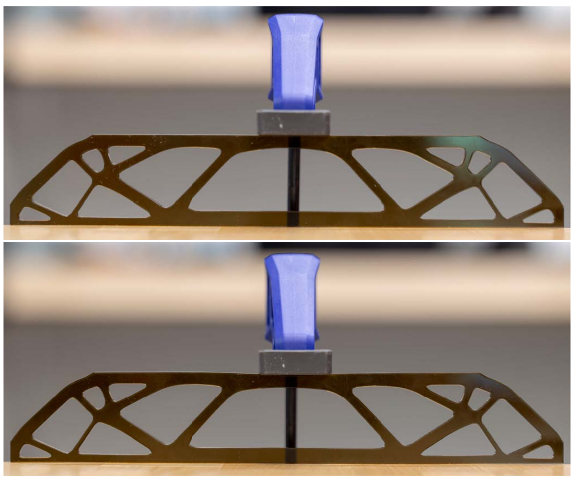

Figure 10. Laser cut bridge model generated by our algorithm under no load (top) and under a load provided by a clamp in the middle (bottom).

References

[1] M. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, and G. N. Wells. The fenics project version 1.5. *Archive of Numerical Software*, 3(100):9–23, 2015.

[2] M. S. Alnaes, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, and G. N. Wells. The fenics project version 1.5. *Archive of Numerical Software*, 3, 2015.

[3] M. S. Andersen, J. Dahl, and L. Vandenberghe. Cvxopt: A python package for convex optimization, version 1.1.5, 2012. Available at http://cvxopt.org/.

[4] E. Andreassen, A. Clausen, M. Schevenels, B. S. Lazarov, and O. Sigmund. Efficient topology optimization in matlab using 88 lines of code. *Structural and Multidisciplinary Optimization*, 43(1):1–16, 2011.

[5] M. Bendse and O. Sigmund. *Topology optimization*, pages 161–194. World Scientific, 2007.

[6] Epilog. Epilog mini 24 laser, 2018. https://goo.gl/iGTZZc.

[7] S. G. Johnson. The nlopt nonlinear-optimization package, 2018. http://ab-initio.mit.edu/nlopt.

[8] O. Sigmund. A 99 line topology optimization code written in matlab. *Structural and Multidisciplinary Optimization*, 21(2):120–127, Apr 2001.

[9] K. Svanberg. The method of moving asymptotesa new method for structural optimization. *International journal for numerical methods in engineering*, 24(2):359–373, 1987.

[10] Ultimaker. Ultimaker 3, 2018. https://goo.gl/MrAjHe.