

# Interactive Hyperelasticity Simulation

Jon-Michael Deldin  
Simulation and Modeling

Spring 2012

# 1 Introduction

Hyperelastic materials are materials that can experience large elastic strain from small forces and retain their original properties after the applied forces are removed [3]. Materials are hyperelastic if a scalar function exists to relate a material's strain energy-density to its deformation gradient [4]. These are nonlinear models because the deformation is not directly proportional to the stress applied. Due to the non-linearity, it is difficult to understand the impact of various input parameters without "seeing" them applied. This paper addresses this by developing and describing an interactive website.

## 2 Methods

In modeling hyperelasticity for a domain  $\Omega \subset \mathbb{R}^d$ , where  $d$  is the spatial dimension, the goal is to find a displacement field  $u : \Omega \rightarrow \mathbb{R}^d$  that minimizes the total potential energy  $\Pi$ . We can represent this problem as

$$\min_{u \in V} \Pi,$$

where  $V$  is a function space that satisfies boundary conditions on  $u$ . Total potential energy is defined as

$$\Pi = \int_{\Omega} \psi(u) dx - \int_{\Omega} B \cdot u dx - \int_{\partial\Omega} T \cdot u ds,$$

where  $\psi$  is the elastic stored energy density function,  $B$  is the body force per unit reference volume (like gravity), and  $T$  is the traction force per unit reference area[2] (e.g., pushing/pulling force).

The elastic stored energy density function  $\psi$  requires a number of parameters to be defined:

**deformation gradient**  $F = I + \nabla u$

**right Cauchy-Green tensor**  $C = F^T F$

**scalars**  $J = \det(F)$ ,  $I_c = \text{trace}(C)$

**Young's modulus**  $E$  is the stiffness of a material

**Poisson ratio** This is a measure of the Poisson effect, which describes how a material tends to expand in the directions perpendicular to the direction of compression (e.g., pressing a finger into a balloon).

**Lamé parameters**  $\mu = \frac{E}{2(1+\nu)}$ ,  $\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}$

These parameters lead to the function definition [2]

$$\psi = \frac{\mu}{2}(I_c - 3) - \mu \ln(J) + \frac{\lambda}{2} \ln(J)^2.$$

The directional derivative of the total energy with respect to change in  $u$  is given by the following variational equation[2]:

$$L(u; v) = D_v \Pi = \left. \frac{d\Pi(u + \epsilon v)}{d\epsilon} \right|_{\epsilon=0}.$$

This is equal to zero for all  $v \in V$ . To solve this equation with Newton's method, we need to define the Jacobian of  $L$  as

$$a(u; du; v) = D_{du} L = \left. \frac{dL(u + \epsilon du; v)}{d\epsilon} \right|_{\epsilon=0}.$$

Fortunately, the model is easily expressed in terms of finite elements and can be easily translated to a finite elements solver, such as FEniCS [1] for this project.

### 2.1 Interface

With the model in hand, we can wrap a user-interface around FEniCS. The completed website is shown in Fig. 1–2. This interface features three plots: the primary is a tricontour plot of the solution, and the lower plots show the shape of the material before and after deforming it. The outline plot shows each coordinate in the mesh before and after deformation, and the shape plot shows just the corner points to generate a crude shape. Below the plots are controls for the hyperelasticity problem, and each input is documented to minimize a user's memory load. The most recent plot is saved each time the user deforms the material, so he or she can compare the effects of each parameter.

The interface is implemented as a Python server-side script that runs on a web server. Specifically, it uses Flask, a small web library and server, to handle requests and plotting. This is a long-running script that loads FEniCS, so it is able to render plots without an additional delay. Plots are generated as PNG files in Matplotlib and included in the HTML as `IMG` elements.

When the user submits a set of parameters, they are sent as a POST request to the script, which then hands them off to FEniCS/DOLFIN. Next, the previous figure is renamed, and a new figure is saved.

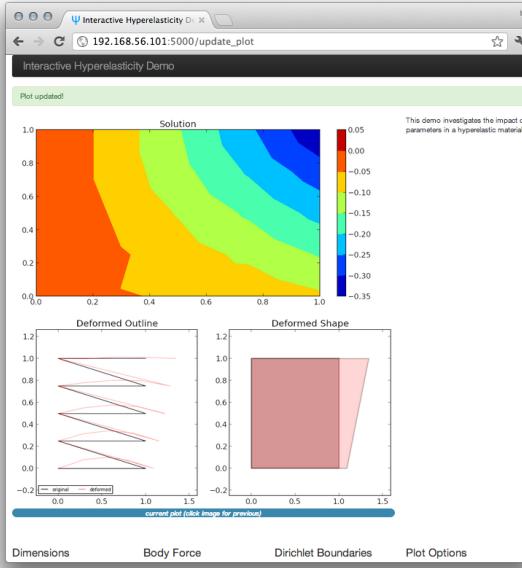


Figure 1: Screenshot of the interactive website.

Finally, the user is redirected back to the main page where he or she can examine the results and change other parameters.

## 2.2 Verification

This simulation is user-defined, so the method of verification is visually inspecting the resulting deformed outline or the solution plot. The model was adapted from [2], and it is assumed to be correct.

## 3 Analysis

To assess the effect of each parameter, all parameters are set to 0, except for  $E = 10$ ,  $\nu = 0.3$ , and the unit square is  $4 \times 4$ . This results in a deformed material identical to the original, shown in Fig. 3.

### 3.1 Body Force

The body force is simply a force that acts throughout the volume of a body. With the boundaries set to zero, the body force can only impact points in-between. This parameter has a direct effect on the positions of each mesh point, which is expected (Fig. 4). Positive values represent compression due to a force from the top of the material.

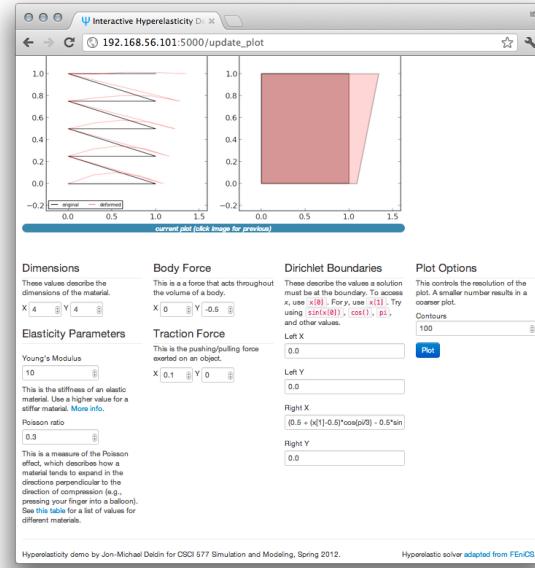


Figure 2: Screenshot of the website's hyperelasticity controls.

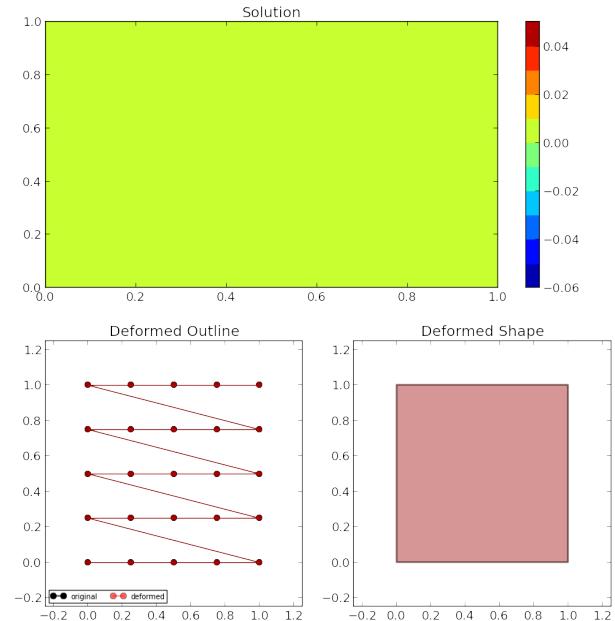


Figure 3: Baseline plot showing no deformation.

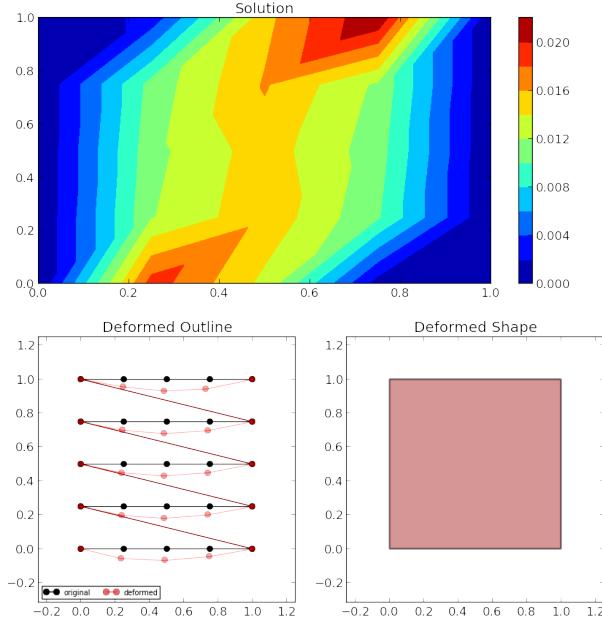


Figure 4: Body force of  $x = 1.5, y = 2.0$ .

### 3.2 Traction Force

Traction force is the pushing/pulling force exerted on an object, and the interface enables us to explore that. This force has a much greater effect on the solution and coordinates than the body force, shown in Fig. 5. This is reasonable due to the traction force being multiplied by the derivative of the area instead of just  $x$ .

### 3.3 Dirichlet Boundaries

Dirichlet boundaries describe the values a solution must be at a given boundary. The website permits assigning values to the  $x$  and  $y$  boundaries of both the left and right subdomains. Figures 6–9 are plots of adjusting individual  $x$  or  $y$  values. It seems as though the boundary conditions are carried through the solution, which may indicate FEniCS does not reset the boundaries at each integration.

### 3.4 Young's Modulus

Young's modulus,  $E$ , is a measure of the stiffness of an elastic material. In this simulation, it does not have an effect on the material's shape on its own or combined with other parameters (confirmed visually and by comparing MD5 hashes of each plot im-

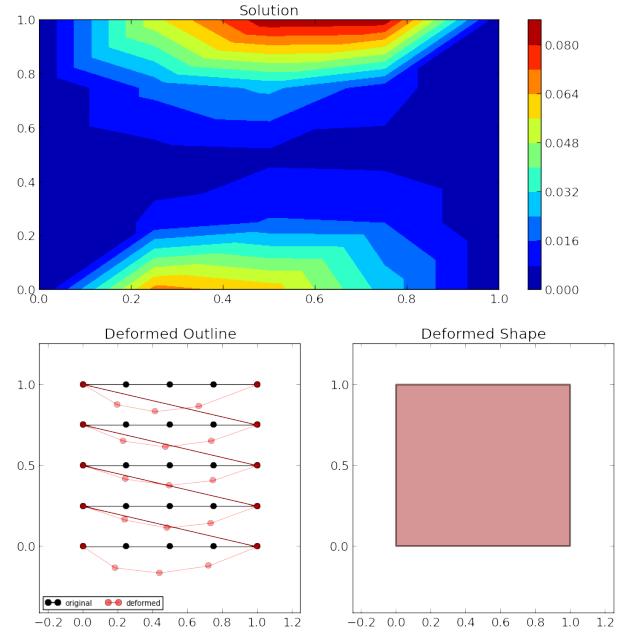


Figure 5: Traction force of  $x = 1.5, y = 2.0$ .

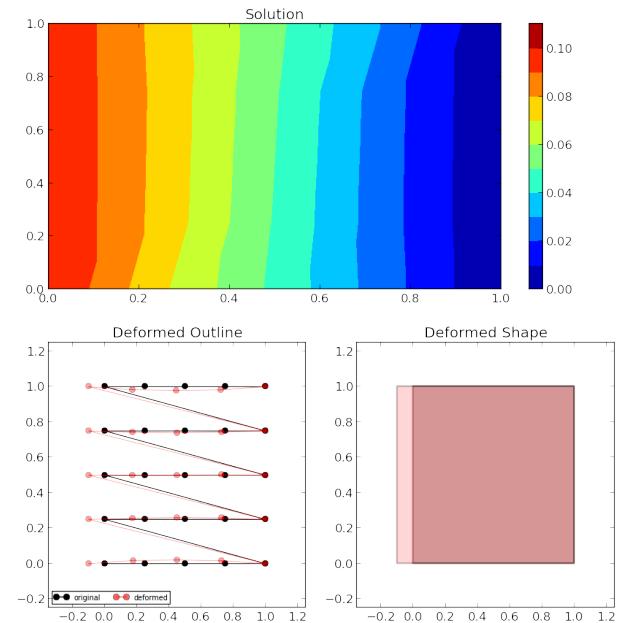


Figure 6: Dirichlet boundary of left- $x$  set to 0.1.

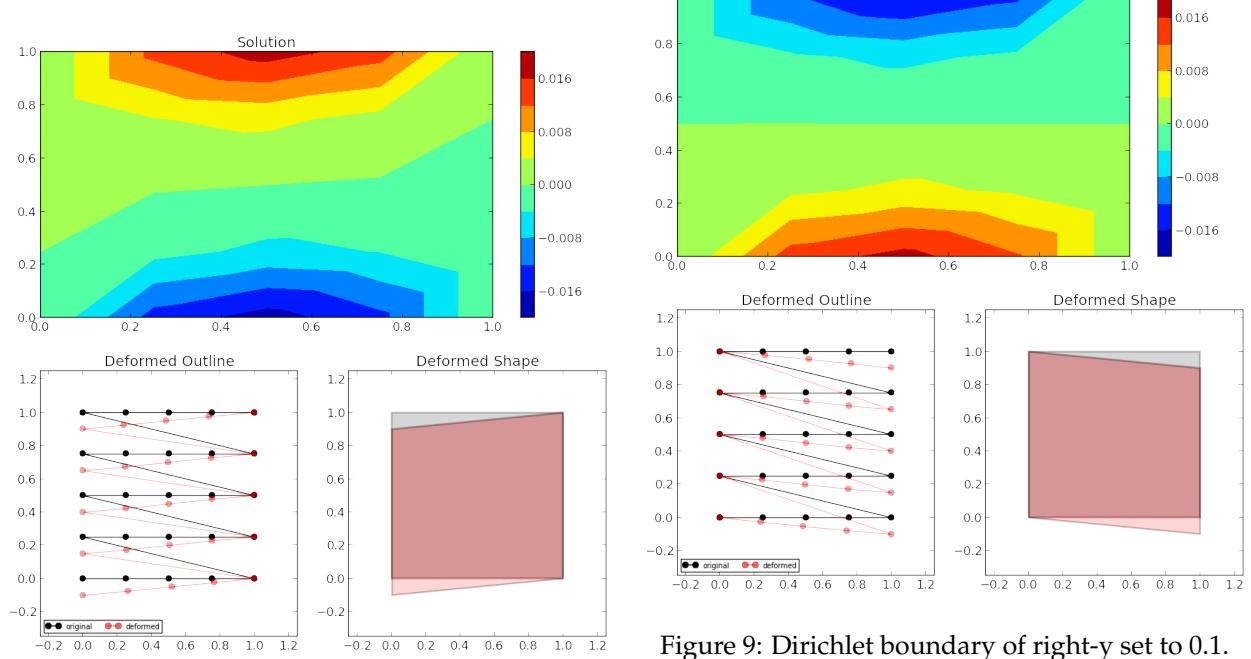


Figure 7: Dirichlet boundary of left-y set to 0.1.

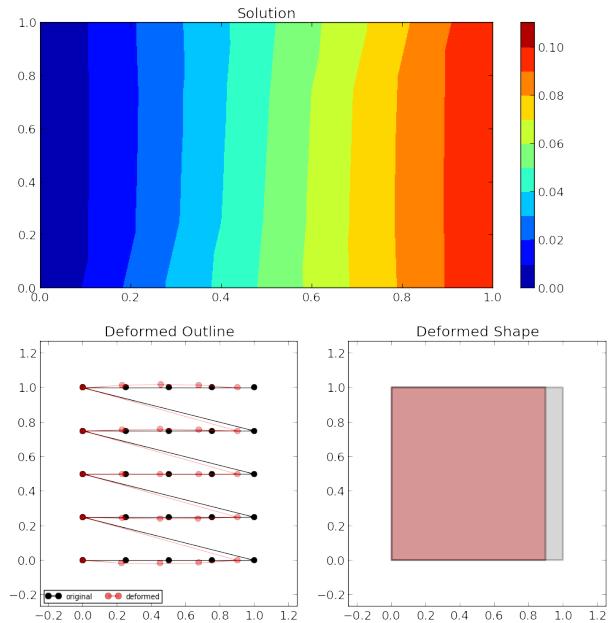


Figure 8: Dirichlet boundary of right-x set to 0.1.

age. This may be due to this parameter's reduced importance in a 2D problem.

### 3.5 Poisson Ratio

The Poisson ratio,  $\nu$ , is observed in Fig. 10. This version is with a ratio of 0.45, which is similar to clay, which is why the coordinates are quite shifted.

## 4 Interpretation

Overall, this project was successful in exploring hyperelasticity. The interface enables variables to be explored individually or in combination, and it allows the user to investigate hypotheses. It has the potential to be a useful tool in understanding hyperelasticity and FEniCS, and hopefully, it can be placed online on a public-facing server<sup>1</sup>.

Future experiments might explore shading the deformed outlines to provide an accurate representation, instead of shading just the corners. Another area worth investigating is adding mouse interaction, so a user could click-and-drag to change a

<sup>1</sup>I intend to load this onto Amazon's EC2 cloud computing server, which will allow FEniCS, Flask, and Matplotlib to be installed.

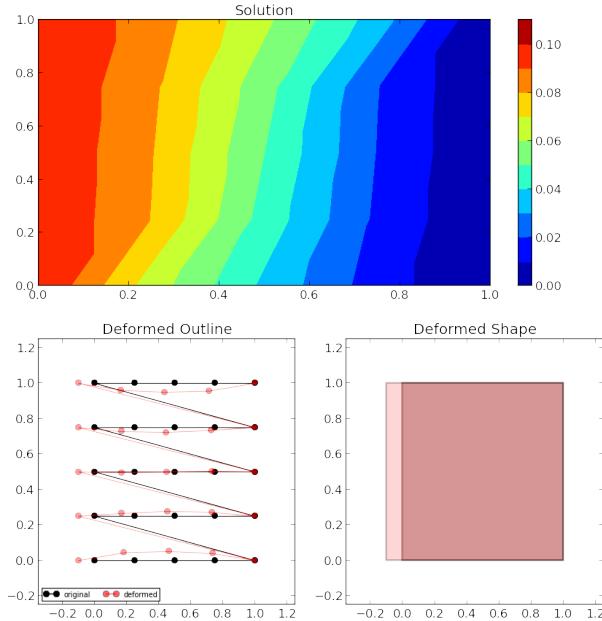


Figure 10: Poisson ratio of 0.45.

boundary. The difficulty will be in creating a responsive interface, and the primary bottleneck is the compilation of boundary conditions in DOLFIN. On a related note, the website could be faster with intelligent caching of model parameters to avoid regenerating DOLFIN parameters. Finally, if the model code is fast enough, Ajax could be used to update the interface immediately after a parameter is changed, so the user would have immediate feedback.

## References

- [1] Anders Logg, Kent-Andre Mardal, Garth N. Wells, et al. *Automated Solution of Differential Equations by the Finite Element Method*. Springer, 2012. ISBN 978-3-642-23098-1. doi: 10.1007/978-3-642-23099-8.
- [2] Anders Logg, Kent-Andre Mardal, Garth N. Wells, et al. Hyperelasticity, 05 2012.
- [3] Manuel J. Garcia R., Oscar E. Ruiz S., and Carlos Lopez. Hyperelastic material modeling. Technical report, Laboratorio CAD/CAM/CAE, Departamento de Ingeneiería Mecánica, Universidad EAFIT, January 2005. URL <http://mecanica.eafit.edu.co/~mgarcia/>

[mg/projects/2004\\_hiperelasticity\\_techReport.pdf](http://mg/projects/2004_hiperelasticity_techReport.pdf).

- [4] Wikipedia. Hyperelastic material — wikipedia, the free encyclopedia, 2012. URL [http://en.wikipedia.org/w/index.php?title=Hyperelastic\\_material&oldid=476230019](http://en.wikipedia.org/w/index.php?title=Hyperelastic_material&oldid=476230019). [Online; accessed 9-May-2012].