Solving PDEs with Firedrake: hyperelasticity

Patrick E. Farrell

University of Oxford

February 2024

We will build a solver for a nontrivial problem: compressible hyperelasticity.

We will build a solver for a nontrivial problem: compressible hyperelasticity.

Like linear elasticity, these equations describe how a structure deforms under a load. We set Ω to be the undeformed configuration, and solve for a displacement $u : \Omega \to \mathbb{R}^d$ that describes how each point $X \in \Omega$ maps to the deformed configuration:

$$x(X) = X + u(X).$$

We will build a solver for a nontrivial problem: compressible hyperelasticity.

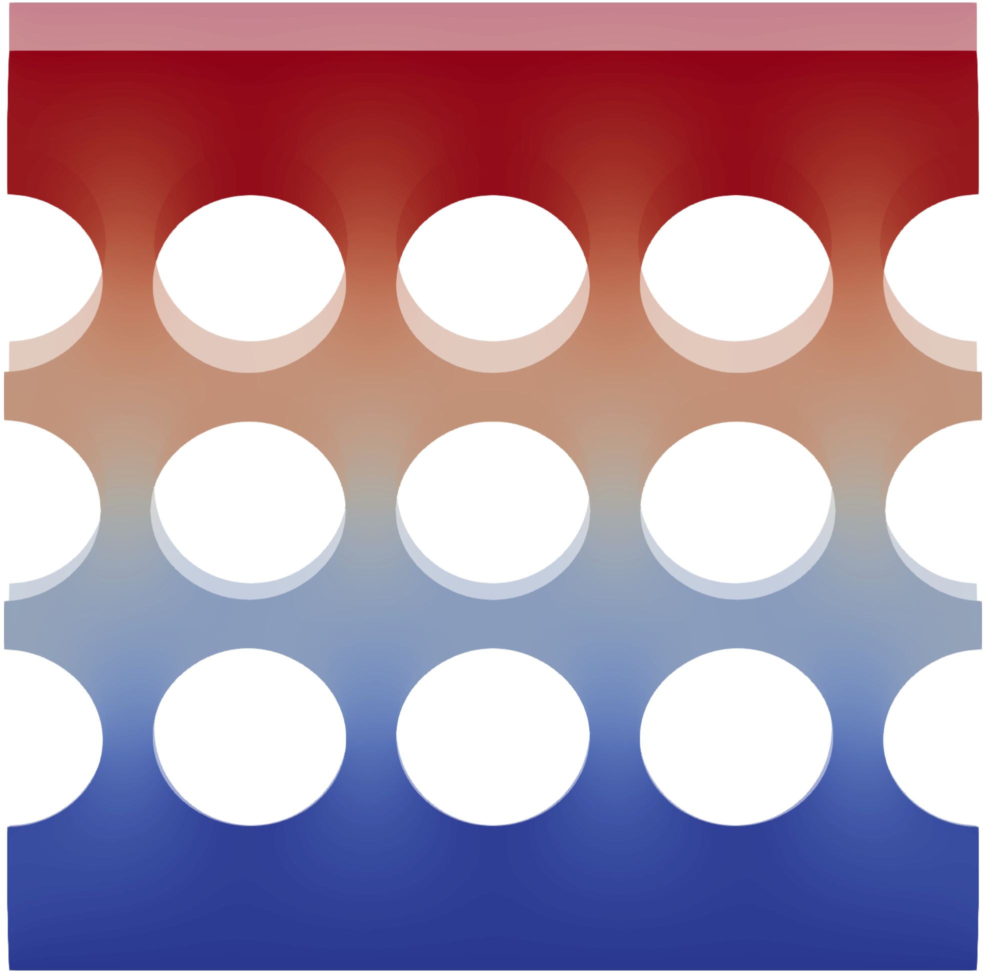
Like linear elasticity, these equations describe how a structure deforms under a load. We set Ω to be the undeformed configuration, and solve for a displacement $u : \Omega \to \mathbb{R}^d$ that describes how each point $X \in \Omega$ maps to the deformed configuration:

$$x(X) = X + u(X).$$

Unlike linear elasticity, hyperelasticity is more realistic because

- (constitutive nonlinearity) the stress-strain curve is not necessarily linear;
- (geometric nonlinearity) the displacements are not necessarily small.

The equations are thus nonlinear.



Challenge!

In this exercise, you will write your own code from scratch.

Good news!

I will tell you everything you need to know.

Section 2

Minimisation and saddle point problems

Many problems can be cast in an optimisation framework.

For example, the Poisson equation arises as the minimisation of the Dirichlet energy

$$J(u) = \frac{1}{2}\int_\Omega \nabla u \cdot \nabla u \ \mathrm{d}x - \int_\Omega f u \ \mathrm{d}x.$$

Many problems can be cast in an optimisation framework.

For example, the Poisson equation arises as the minimisation of the Dirichlet energy

$$J(u) = \frac{1}{2}\int_\Omega \nabla u \cdot \nabla u \, \mathrm{d}x - \int_\Omega f u \, \mathrm{d}x.$$

We can see this by taking its Fréchet derivative and setting it to zero:

$$\begin{aligned} J_u(u; v) &\coloneqq \lim_{\epsilon \to 0} \frac{J(u + \epsilon v) - J(u)}{\epsilon} \\ &= \lim_{\epsilon \to 0} \frac{1}{\epsilon}\left(\epsilon \int_\Omega \nabla u \cdot \nabla v \, \mathrm{d}x + \epsilon^2 \int_\Omega \nabla v \cdot \nabla v \, \mathrm{d}x - \epsilon \int_\Omega f v \, \mathrm{d}x\right) \\ &= \int_\Omega \nabla u \cdot \nabla v \, \mathrm{d}x - \int_\Omega f v \, \mathrm{d}x = 0, \end{aligned}$$

the weak statement of the Poisson equation.

We can get Firedrake to do this calculation for us:

```
# Functional to optimise
J = (
      0.5 * inner(grad(u), grad(u))*dx
    -       inner(f, u)*dx
    )

# Calculate the optimality condition (equation to solve)
F = derivative(J, u)
```

We can get Firedrake to do this calculation for us:

```
# Functional to optimise
J = (
      0.5 * inner(grad(u), grad(u))*dx
    -       inner(f, u)*dx
    )

# Calculate the optimality condition (equation to solve)
F = derivative(J, u)
```

Firedrake uses `derivative` inside `solve` to calculate the Jacobian.

Section 3

Hyperelasticity energy functional

Our *dramatis personae*:

- $\Omega \subset \mathbb{R}^d$, the domain;

Ronald Rivlin

Our *dramatis personae*:

- $\Omega \subset \mathbb{R}^d$, the domain;
- $u : \Omega \to \mathbb{R}^d$, the displacement;

Ronald Rivlin

Our *dramatis personae*:

- $\Omega \subset \mathbb{R}^d$, the domain;
- $u : \Omega \to \mathbb{R}^d$, the displacement;
- $F = I + \nabla u$, the deformation gradient;

Ronald Rivlin

Our *dramatis personae*:

- $\Omega \subset \mathbb{R}^d$, the domain;
- $u : \Omega \to \mathbb{R}^d$, the displacement;
- $F = I + \nabla u$, the deformation gradient;
- $C = F^\top F$, the right Cauchy–Green tensor;

Ronald Rivlin

Our *dramatis personae*:

- $\Omega \subset \mathbb{R}^d$, the domain;
- $u : \Omega \to \mathbb{R}^d$, the displacement;
- $F = I + \nabla u$, the deformation gradient;
- $C = F^\top F$, the right Cauchy–Green tensor;
- $I_c = \mathrm{tr}(C), J = \det(F)$, invariants;

Ronald Rivlin

Our *dramatis personae*:

- $\Omega \subset \mathbb{R}^d$, the domain;
- $u : \Omega \to \mathbb{R}^d$, the displacement;
- $F = I + \nabla u$, the deformation gradient;
- $C = F^\top F$, the right Cauchy–Green tensor;
- $I_c = \mathrm{tr}(C), J = \det(F)$, invariants;
- μ, λ, Lamé parameters.

Ronald Rivlin

Our *dramatis personae*:

- $\Omega \subset \mathbb{R}^d$, the domain;
- $u : \Omega \to \mathbb{R}^d$, the displacement;
- $F = I + \nabla u$, the deformation gradient;
- $C = F^\top F$, the right Cauchy–Green tensor;
- $I_c = \mathrm{tr}(C), J = \det(F)$, invariants;
- μ, λ, Lamé parameters.

Ronald Rivlin

With these, we form the compressible neo-Hookean energy:

$$E(u) = \int_\Omega \frac{\mu}{2}(I_c - d) - \mu \ln J + \frac{\lambda}{2}(\ln J)^2 \, \mathrm{d}x.$$

Stating this in Firedrake:

```
d = mesh.geometric_dimension()
I = Identity(d)
F = I + grad(u)
C = F.T * F
I_c = tr(C)
J = det(F)
```

Section 4

Continuation

Continuation is an extremely powerful algorithm for solving difficult nonlinear problems.

Idea: construct a good initial guess by solving an easier problem.

Continuation

- Solve the problem for easy parameter value.
- While not finished:
  - Use solution for previous parameter as initial guess for next parameter.
  - Increment parameter.

To do continuation in Firedrake, update the parameter in a loop and solve:

```
strain = Constant(0)  # placeholder Constant
# Use the strain as our boundary condition value:
bcs = [... ,
       DirichletBC(V.sub(1), strain, top),
       ...]

strains = ...
for strain_ in strains:
    strain.assign(strain_)  # update parameter value
    solve(F == 0, u, bcs)   # solve for next parameter
```

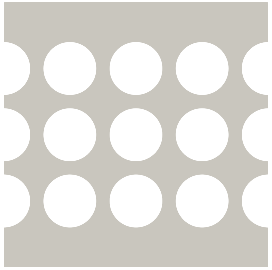
Challenge!

Solve the equations of hyperelasticity on the domain

$$\Omega = (0,1)^2 \setminus \left(\bigcup_{ij} D_{ij} \right),$$

where $i \in \{1, \dots, 3\}, j \in \{1, \dots, 5\}$, and

$$D_{ij} = \left\{ (x,y) \in \mathbb{R}^2 : \left(x - \frac{j-1}{4} \right)^2 + \left(y - \frac{i}{4} \right)^2 \leq 0.1^2 \right\}.$$



Challenge!

Solve the problem with $\mu = 4 \times 10^5$, $\lambda = 6 \times 10^5$, and boundary conditions

$$\begin{aligned} u &= (0,0) \quad \text{on } \{y = 0\}, \\ u &= (0,-s) \ \text{on } \{y = 1\}, \end{aligned}$$

for $s = 0.1$.

Apply natural (i.e. do-nothing, stress-free) boundary conditions on all other boundaries.

Hint: you will probably need to employ continuation.