FEniCS Course

Lecture 6: Static hyperelasticity

Contributors
Anders Logg

Static hyperelasticity

$$
\begin{aligned}
-\operatorname{div} P &= B && \text{in } \Omega \\
u &= g && \text{on } \Gamma_{\mathrm{D}} \\
P \cdot n &= T && \text{on } \Gamma_{\mathrm{N}}
\end{aligned}
$$

- $X \in \Omega$ is a coordinate in the reference domain
- $x = x(X)$ is the deformation
- $u = x - X$ is the displacement
- $P = P(u)$ is the first Piola–Kirchoff stress tensor
- B is a given body force per unit volume
- g is a given boundary displacement
- T is a given boundary traction

Variational problem

Multiply by a test function $v \in \hat{V}$ and integrate by parts:

$$-\int_\Omega \operatorname{div} P \cdot v \,\mathrm{d}x = \int_\Omega P : \operatorname{grad} v \,\mathrm{d}x - \int_{\partial\Omega} (P \cdot n) \cdot v \,\mathrm{d}s$$

Note that $v = 0$ on Γ_D and $P \cdot n = T$ on Γ_N

Find $u \in V$ such that

$$\int_\Omega P : \operatorname{grad} v \,\mathrm{d}x = \int_\Omega B \cdot v \,\mathrm{d}x + \int_{\Gamma_\mathrm{N}} T \cdot v \,\mathrm{d}s$$

for all $v \in \hat{V}$

Stress–strain relations

- $F = \frac{dx}{dX} = \frac{d(X+u)}{dX} = I + \operatorname{grad} u$ is the deformation gradient
- $C = F^\top F$ is the right Cauchy–Green deformation tensor
- $E = \frac{1}{2}(C - I)$ is the Green–Lagrange strain tensor
- $W = W(E)$ is the strain energy density
- $S_{ij} = \frac{\partial W}{\partial E_{ij}}$ is the second Piola–Kirchoff stress tensor
- $P = FS$ is the first Piola–Kirchoff stress tensor

St. Venant–Kirchoff strain energy function:

$$W(E) = \frac{\lambda}{2}(\operatorname{tr}(E))^2 + \mu \operatorname{tr}(E^2)$$

Useful FEniCS tools (I)

Defining subdomains/boundaries:

Python code

```
class MyBoundary(SubDomain):
    def inside(self, x, on_boundary):
        return on_boundary and \
               x[0] < 1.0 - DOLFIN_EPS
```

*How should this be modified to **not** include the upper and lower right corners?*

Marking boundaries:

Python code

```
my_boundary_1 = MyBoundary1()
my_boundary_2 = MyBoundary2()
boundaries = FacetFunction("size_t", mesh)
boundaries.set_all(0)
my_boundary_1.mark(boundaries, 1)
my_boundary_2.mark(boundaries, 2)
ds = Measure("ds", subdomain_data=boundaries)
a = ...*ds(0) + ...*ds(1) + ...*ds(2)
```

Useful FEniCS tools (II)

Computing derivatives of expressions:

Python code

```
E = variable(...)
W = ...
S = diff(W, E)
```

Computing form derivatives:

Python code

```
u  = Function(V)
du = TrialFunction(V)
F  = ...u...*dx
J  = derivative(F, u, du)
```

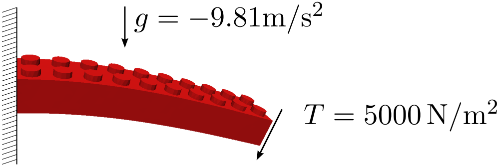
Solving nonlinear variational problems (Newton's method):

Python code

```
solve(F == 0, u, bcs)
solve(F == 0, u, bcs, J=J)
```

The FEniCS challenge!

Compute the deflection of a regular 10×2 LEGO brick. Use the St. Venant–Kirchhoff model and assume that the LEGO brick is made of PVC plastic. The LEGO brick is subject to gravity of size $g = -9.81$ m/s^2 and a downward traction of size 5000 N/m^2 at its end point.



To check your solution, compute the average value of the displacement in the z-direction.

The student(s) who first produce the right answer will be rewarded with an exclusive FEniCS surprise!

Mesh and material parameters:

`http://fenicsproject.org/pub/course/data/`