Lecture 05

Application to Elasticity

Anders Logg

May 18, 2018

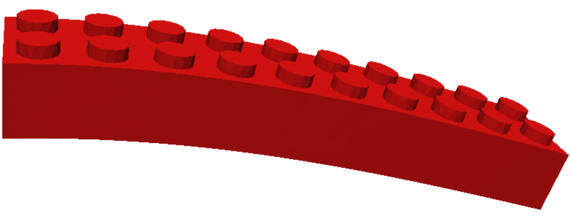
What will you learn?

- ***How to solve linear elastic problems***
- ***How to solve nonlinear elastic problems***
- *How to formulate the linear elastic problem*
- *How to formulate the nonlinear elastic problem*
- *How to derive the variational problem*
- *FEniCS programming*
    - Assembling linear systems
    - Applying boundary conditions
    - Solving linear systems
    - Assembling and solving linear systems
    - Projecting and interpolating solutions
    - Computing functionals
- *Exercises*

Solid mechanics

Compute the deformed configuration $\varphi(\Omega)$ of an elastic body

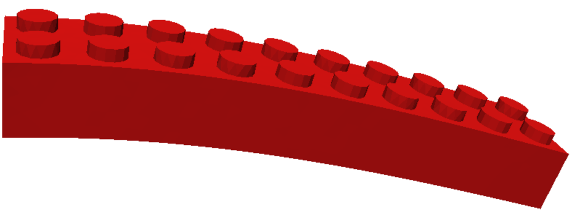
$$\Omega \quad \mapsto \quad \varphi(\Omega)$$



Solid mechanics

Compute the deformed configuration $\varphi(\Omega)$ of an elastic body

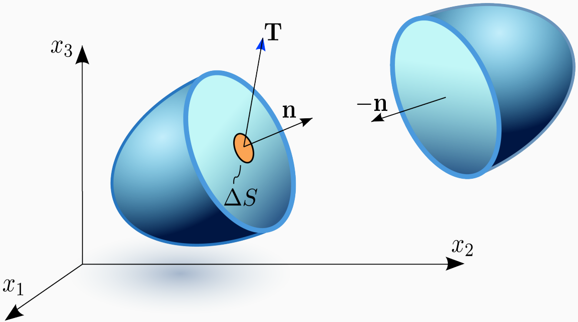
$$\Omega \quad \mapsto \quad \varphi(\Omega)$$



Or equivalently compute the *displacement* $u(x) = \varphi(x) - x$ of each point $x \in \Omega$

How to formulate the linear elastic problem

The stress tensor



σ is the stress tensor [force per unit area]

n is the unit normal [dimensionless]

$T = \sigma \cdot n$ is the boundary traction [force per unit area]

$F \approx T\Delta S$ is the force acting on ΔS

Partial differential equation

The linear elastic PDE

$$-\nabla \cdot \sigma(u) = f \quad \text{in } \Omega$$

Partial differential equation

The linear elastic PDE

$$-\nabla \cdot \sigma(u) = f \quad \text{in } \Omega$$

Expresses balance between internal and external forces

u is the solution to be computed (the displacement)

f is a given source term (the body force)

Ω is the computational domain (the elastic body)

$\sigma(u)$ is the Cauchy stress tensor (linear in u)

Partial differential equation

The linear elastic PDE

$$-\nabla \cdot \sigma(u) = f \quad \text{in } \Omega$$

Expresses balance between internal and external forces

u is the solution to be computed (the displacement)

f is a given source term (the body force)

Ω is the computational domain (the elastic body)

$\sigma(u)$ is the Cauchy stress tensor (linear in u)

Note: u and f are vector-valued

Note: $\sigma(u)$ is matrix-valued

Boundary conditions

Dirichlet boundary condition

$$u = u_{\mathrm{D}} \quad \text{on } \Gamma_{\mathrm{D}} \subseteq \partial\Omega$$

Boundary conditions

Dirichlet boundary condition

$$u = u_{\mathrm{D}} \quad \text{on } \Gamma_{\mathrm{D}} \subseteq \partial\Omega$$

Neumann boundary condition

$$\sigma \cdot n = g \quad \text{on } \Gamma_{\mathrm{N}} \subseteq \partial\Omega$$

Boundary conditions

Dirichlet boundary condition

$$u = u_{\mathrm{D}} \quad \text{on } \Gamma_{\mathrm{D}} \subseteq \partial\Omega$$

Neumann boundary condition

$$\sigma \cdot n = g \quad \text{on } \Gamma_{\mathrm{N}} \subseteq \partial\Omega$$

The Dirichlet condition $u = u_{\mathrm{D}}$ is also called a *strong* boundary condition

The Neumann condition $\sigma \cdot n = g$ is also called a *natural* boundary condition

The function g is the *boundary traction*

Linearly elastic materials

The Cauchy stress tensor

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

Linearly elastic materials

The Cauchy stress tensor

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

The symmetric gradient (of a vector v)

$$\varepsilon(v) = \mathrm{sym}(\nabla v) = \frac{1}{2}(\nabla v + (\nabla v)^\top)$$

The trace (of a matrix A)

$$\mathrm{tr}(A) = \sum_{i=1}^{d} A_{ii}$$

Linearly elastic materials

The Cauchy stress tensor

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

The symmetric gradient (of a vector v)

$$\varepsilon(v) = \mathrm{sym}(\nabla v) = \frac{1}{2}(\nabla v + (\nabla v)^\top)$$

The trace (of a matrix A)

$$\mathrm{tr}(A) = \sum_{i=1}^{d} A_{ii}$$

μ and λ are called the *Lamé parameters*

Material parameters

1st Lamé parameter μ relates shear stress to shear strain

2nd Lamé parameter λ relates pressure to volumetric strain

Material parameters

1st Lamé parameter μ relates shear stress to shear strain

2nd Lamé parameter λ relates pressure to volumetric strain

Relation to Young's modulus E and Poisson ratio ν

$$\mu = \frac{E}{2(1+\nu)}$$

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}$$

Note that $\lambda \to \infty$ when $\nu \to 1/2$

Material parameters

1st Lamé parameter μ relates shear stress to shear strain

2nd Lamé parameter λ relates pressure to volumetric strain

Relation to Young's modulus E and Poisson ratio ν

$$\mu = \frac{E}{2(1+\nu)}$$

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}$$

Note that $\lambda \to \infty$ when $\nu \to 1/2$

Relation to shear modulus G and bulk modulus K

$$G = \mu$$

$$K = \lambda + 2\mu/3$$

How to formulate the nonlinear elastic problem

Partial differentiual equation

The nonlinear elastic PDE

$$-\nabla \cdot P(u) = f \quad \text{in } \Omega$$

Expresses balance between internal and external forces

Expressed in the *reference configuration* Ω

u is the solution to be computed (the displacement)

f is a given source term (the body force)

Ω is the computational domain (the undeformed elastic body)

$P(u)$ is the first Piola–Kirchoff stress tensor (nonlinear in u)

Boundary conditions (same as for the linear case)

Dirichlet boundary condition

$$u = u_{\mathrm{D}} \quad \text{on } \Gamma_{\mathrm{D}} \subseteq \partial\Omega$$

Boundary conditions (same as for the linear case)

Dirichlet boundary condition

$$u = u_{\mathrm{D}} \quad \text{on } \Gamma_{\mathrm{D}} \subseteq \partial\Omega$$

Neumann boundary condition

$$P \cdot n = g \quad \text{on } \Gamma_{\mathrm{N}} \subseteq \partial\Omega$$

Boundary conditions (same as for the linear case)

Dirichlet boundary condition

$$u = u_{\mathrm{D}} \quad \text{on } \Gamma_{\mathrm{D}} \subseteq \partial\Omega$$

Neumann boundary condition

$$P \cdot n = g \quad \text{on } \Gamma_{\mathrm{N}} \subseteq \partial\Omega$$

The Dirichlet condition $u = u_{\mathrm{D}}$ is also called a *strong* boundary condition

The Neumann condition $P \cdot n = g$ is also called a *natural* boundary condition

The function g is the *boundary traction*

Nonlinear elastic materials (hyperelasticity)

Express strain energy function in terms of strain measures

Standard strain and stress measures

- $F = \frac{\partial \varphi}{\partial x} = \frac{\partial (x+u(x))}{\partial x} = I + \nabla u$ 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

Nonlinear elastic materials (hyperelasticity)

Express strain energy function in terms of strain measures

Standard strain and stress measures

- $F = \frac{\partial \varphi}{\partial x} = \frac{\partial(x+u(x))}{\partial x} = I + \nabla u$ 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

Saint Venant–Kirchoff strain energy function

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

One of many hyperelastic models!

How to derive the variational problem

From strong to weak form: (1) $\to$ (2) (linear case)

Partial differential equation (strong form)

$$-\nabla \cdot \sigma(u) = f$$

From strong to weak form: (1) $\to$ (2) (linear case)

Partial differential equation (strong form)

$$-\nabla \cdot \sigma(u) = f$$

Multiply by a test function v and integrate by parts:

$$-\int_{\Omega} \nabla \cdot \sigma(u) \cdot v \,\mathrm{d}x = \int_{\Omega} \sigma(u) : \varepsilon(v) \,\mathrm{d}x - \int_{\partial\Omega} (\sigma(u) \cdot n) \cdot v \,\mathrm{d}s$$

From strong to weak form: (1) $\rightarrow$ (2) (linear case)

Partial differential equation (strong form)

$$-\nabla \cdot \sigma(u) = f$$

Multiply by a test function v and integrate by parts:

$$-\int_\Omega \nabla \cdot \sigma(u) \cdot v \,\mathrm{d}x = \int_\Omega \sigma(u) : \varepsilon(v) \,\mathrm{d}x - \int_{\partial\Omega} (\sigma(u) \cdot n) \cdot v \,\mathrm{d}s$$

Note that $v = 0$ on Γ_D and $\sigma(u) \cdot n = g$ on Γ_N

From strong to weak form: (1) $\to$ (2) (linear case)

Partial differential equation (strong form)

$$-\nabla \cdot \sigma(u) = f$$

Multiply by a test function v and integrate by parts:

$$-\int_\Omega \nabla \cdot \sigma(u) \cdot v \,\mathrm{d}x = \int_\Omega \sigma(u) : \varepsilon(v) \,\mathrm{d}x - \int_{\partial\Omega} (\sigma(u) \cdot n) \cdot v \,\mathrm{d}s$$

Note that $v = 0$ on Γ_D and $\sigma(u) \cdot n = g$ on Γ_N

$\Rightarrow$ Continuous variational problem (weak form)

Find $u \in V$ such that

$$\int_\Omega \sigma(u) : \varepsilon(v) \,\mathrm{d}x = \int_\Omega f \cdot v \,\mathrm{d}x + \int_{\Gamma_\mathrm{N}} g \cdot v \,\mathrm{d}s \quad \forall v \in V$$

From strong to weak form: (1) $\rightarrow$ (2) (nonlinear case)

Partial differential equation (strong form)

$$-\nabla \cdot P(u) = f$$

From strong to weak form: (1) $\rightarrow$ (2) (nonlinear case)

Partial differential equation (strong form)

$$-\nabla \cdot P(u) = f$$

Multiply by a test function v and integrate by parts:

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

From strong to weak form: (1) $\rightarrow$ (2) (nonlinear case)

Partial differential equation (strong form)

$$-\nabla \cdot P(u) = f$$

Multiply by a test function v and integrate by parts:

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

Note that $v = 0$ on Γ_D and $P(u) \cdot n = g$ on Γ_N

From strong to weak form: (1) $\rightarrow$ (2) (nonlinear case)

Partial differential equation (strong form)

$$-\nabla \cdot P(u) = f$$

Multiply by a test function v and integrate by parts:

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

Note that $v = 0$ on Γ_{D} and $P(u) \cdot n = g$ on Γ_{N}

$\Rightarrow$ Continuous variational problem (weak form)

Find $u \in V$ such that

$$\int_\Omega P(u) : \nabla v \,\mathrm{d}x = \int_\Omega f \cdot v \,\mathrm{d}x + \int_{\Gamma_{\mathrm{N}}} g \cdot v \,\mathrm{d}s \quad \forall v \in V$$

From weak form to system of equations: (2) $\rightarrow$ (3) $\rightarrow$ (4)

(2) $\rightarrow$ (3): Let V_h be a discrete finite element subspace of V

(3) $\rightarrow$ (4): Let $u_h(x) = \sum_{j=1}^{N} U_j \phi_j(x)$

From weak form to system of equations: (2) $\rightarrow$ (3) $\rightarrow$ (4)

(2) $\rightarrow$ (3): Let V_h be a discrete finite element subspace of V

(3) $\rightarrow$ (4): Let $u_h(x) = \sum_{j=1}^{N} U_j \phi_j(x)$

The linear elastic problem gives a symmetric linear system

The nonlinear (hyperelastic) problem gives a nonlinear system

FEniCS programming

Assembling linear systems

Matrices and vectors are assembled using the `assemble()` function:

```
A = assemble(a)
b = assemble(L)
```

`A` is a sparse matrix of size $N \times N$

`b` is a vector of size N

Applying boundary conditions

Dirichlet boundary conditions are applied to an assembled linear system by calling the `apply()` function:

```
bc.apply(A, b)
```

Applying boundary conditions

Dirichlet boundary conditions are applied to an assembled linear system by calling the `apply()` function:

```
bc.apply(A, b)
```

Multiple boundary conditions are applied using either a loop or a list comprehension:

```
for bc in bcs:
    bc.apply(A, b)
```

```
[bc.apply(A, b) for bc in bcs]
```

Solving linear systems

Linear systems are solved using the `solve()` function:

```
U = Vector()
solve(A, U, b)
```

Solving linear systems

Linear systems are solved using the `solve()` function:

```
U = Vector()
solve(A, U, b)
```

Use `u.vector()` to access the vector of degrees of freedom:

```
solve(A, u.vector(), b)
```

Solving linear systems

Linear systems are solved using the `solve()` function:

```
U = Vector()
solve(A, U, b)
```

Use `u.vector()` to access the vector of degrees of freedom:

```
solve(A, u.vector(), b)
```

The linear solver can be controlled by optional arguments:

```
solve(A, U, b, 'gmres', 'ilu')
solve(A, U, b, 'cg', 'amg')
```

Assembling and solving linear systems

Relation to the `solve()` function:

```
def solve(a, L, u, bcs)

    # Assemble linear system
    A = assemble(a)
    b = assemble(L)

    # Apply boundary conditions
    for bc in bcs:
        bc.apply(A, b)

    # Solve linear system
    solve(A, u.vector(), b)
```

Projecting and interpolating solutions

Function projection $P : V \to W$

Find $Pu \in W$ such that

$$\int_\Omega Pu \, w \, \mathrm{d}x = \int_\Omega uw \, \mathrm{d}x$$

for all $w \in W$

Projecting and interpolating solutions

Function projection $P : V \to W$

Find $Pu \in W$ such that

$$\int_\Omega Pu\, w \,\mathrm{d}x = \int_\Omega uw \,\mathrm{d}x$$

for all $w \in W$

Projections are computed by calling the `project()` function:

```
Pu = project(u, W)
```

Projecting and interpolating solutions

Function projection $P : V \to W$

Find $Pu \in W$ such that

$$\int_\Omega Pu\, w \, \mathrm{d}x = \int_\Omega uw \, \mathrm{d}x$$

for all $w \in W$

Projections are computed by calling the `project()` function:

```
Pu = project(u, W)
```

Interpolations are computed by calling the `interpolate()` function:

```
Iu = interpolate(u, W)
```

Computing functionals

Functionals are special forms ("forms of arity 0"):

$$\mathcal{M} : V \to \mathbb{R}$$

Computing functionals

Functionals are special forms (“forms of arity 0”):

$$\mathcal{M} : V \to \mathbb{R}$$

Functionals are computed by calling the `assemble()` function:

```
u = Function()
M0 = assemble(u*u*dx)
M1 = assemble(u*dx)
M2 = assemble(u*ds)
```

Exercises

Exercise 5a: Application to Elasticity (linear)

In this exercise, we will solve the equations of linear elasticity with FEniCS:

$$
\begin{aligned}
-\nabla \cdot \sigma(u) &= f && \text{in } \Omega \\
u &= u_{\mathrm{L}} && \text{on } \Gamma_{\mathrm{L}} \\
u &= u_{\mathrm{R}} && \text{on } \Gamma_{\mathrm{R}} \\
\sigma \cdot n &= g && \text{on } \Gamma_{\mathrm{N}}
\end{aligned}
$$

Write a FEniCS program to compute and plot the displacement u by manually assembling and solving the linear system. Use a projection to compute the maximum and average von Mises stress.

Exercise 5b: Application to Elasticity (nonlinear)

In this exercise, we will solve the equations of nonlinear elasticity (hyperelasticity) with FEniCS:

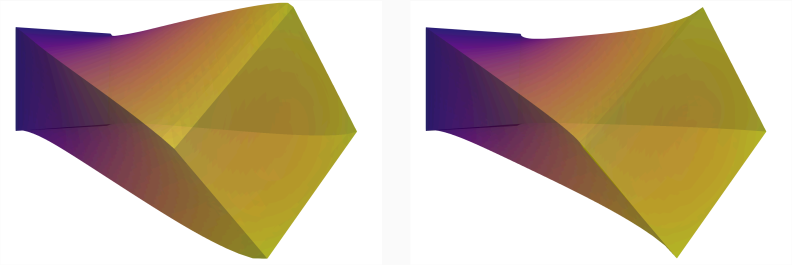
$$
\begin{aligned}
-\nabla \cdot P(u) &= f && \text{in } \Omega \\
u &= u_{\mathrm{L}} && \text{on } \Gamma_{\mathrm{L}} \\
u &= u_{\mathrm{R}} && \text{on } \Gamma_{\mathrm{R}} \\
\sigma \cdot n &= g && \text{on } \Gamma_{\mathrm{N}}
\end{aligned}
$$

Write a FEniCS program to compute and plot the displacement u. Compare the solution with the linear elastic solution from exercise 5.a.

Exercise 5: Problem data

- $\Omega = (0,2) \times (0,1) \times (0,1)$
- $G = 79300\,[\mathrm{Pa}]$ (shear modulus)
- $K = 160000\,[\mathrm{Pa}]$ (bulk modulus)
- $f = (0,0,0)$
- $\Gamma_{\mathrm{L}} = \{(x,y,z) \in \partial\Omega \mid x = 0\}$ (the left boundary)
- $\Gamma_{\mathrm{R}} = \{(x,y,z) \in \partial\Omega \mid x = 2\}$ (the right boundary)
- $\Gamma_{\mathrm{N}} = \partial\Omega \setminus (\Gamma_{\mathrm{L}} \cup \Gamma_{\mathrm{R}})$
- $u_{\mathrm{L}} = (0,0,0)$ (fixed)
- $u_{\mathrm{R}} = (1, 0.5 - y + (y-z)/\sqrt{2}, 0.5 - z + (y+z-1)/\sqrt{2})$ (stretched and rotated)
- $g = (0,0,0)$

Exercise 5: Solution



Solution to Exercise 5 plotted in Paraview using the “Warp By Vector” filter. The left figure shows the linear solution and the right figure shows the nonlinear solution.