FEniCS Course

Lecture 22: Linear elasticity

Contributors
Kent-Andre Mardal

Linear elasticity: continuous equations

- $\Omega_0 \in \mathbb{R}^3$ that is being deformed under a load
- Ω is the deformed Ω_0
- Let $x \in \Omega$ correspond to $X \in \Omega_0$
- Then the deformation is $u = x - X$
- The stress tensor σ is a symmetric 3×3 tensor which is a function u
- Hooke's law states:

$$\sigma = 2\mu\epsilon(u) + \lambda \operatorname{tr}(\epsilon(u))\delta$$

- In equilibrium (i.e. no accelration terms), Newton's second law states:

$$\begin{aligned} \operatorname{div} \sigma &= f, \quad \text{in } \Omega \\ \sigma \cdot n &= g, \quad \text{on } \partial\Omega \end{aligned}$$

- f and g are body and surface forces
- n is the outward normal vector

Linear elasticity: continuous equations, cont'd

- Hooke's law states:

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

- $\epsilon(u)$ is the strain tensor or the symmetric gradient:

$$\epsilon(u) = \frac{1}{2}(\nabla u + (\nabla u)^T)$$

- μ and λ are the Lame constants
- tr is the trace operator (the sum of the diagonal matrix entries), u is the displacement, and

$$\delta = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

From Newton's second law and Hooke's law we arrive directly at the equation of linear elasticity:

$$-2\mu(\nabla \cdot \epsilon(u)) - \lambda\nabla(\nabla \cdot u) = f \tag{1}$$

Linear elasticity: continuous equations, cont'd

The equation of linear elasticity:

$$-2\mu(\nabla\cdot\epsilon(u)) - \lambda\nabla(\nabla\cdot u) = 0 \tag{2}$$

The equation is elliptic, but there are crucial differences between this equation and a standard elliptic equation like $-\Delta u = f$. These differences often cause problems in a numerical setting. To explain the numerical issues we will here consider the three operators:

1. $\Delta = \nabla\cdot\nabla = \operatorname{div}\operatorname{grad}$
2. $\nabla\cdot\epsilon = \nabla\cdot(\frac{1}{2}(\nabla + (\nabla^T))$
3. $\nabla\cdot\operatorname{tr}\epsilon = \nabla\nabla\cdot = \operatorname{grad}\operatorname{div}$

Item 2 leads to the study of rigid motions while item 3 leads to the study of locking

Weak form and finite element method

For the moment we consider the pure Neumann problem
Find $u \in H^1$ such that

$$a(u, v) = f(v), \quad \forall v \in H^1$$

Here

$$a(u, v) = \mu(\epsilon(u), \epsilon(v)) + \lambda(\nabla \cdot u, \nabla \cdot v)$$

$$f(v) = (f, v) + \int_{\partial\Omega} gv ds$$

From the weak form we obtain a linear system $Au = f$ by using the finite element method, where the stiffness matrix involved in linear elasticity is obtained as

$$A_{ij} = a(N_i, N_j)$$

where $\{N_i\}$ are the finite element basis functions

Rigid motions

Rigid motions consists of translations and rotations

Mathematically, we have the following analytical expressions for rigid motions:

$$\mathrm{RM}_{2D} = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} + \begin{bmatrix} 0 & a_2 \\ -a_2 & 0 \end{bmatrix} \begin{bmatrix} -y \\ x \end{bmatrix} \tag{3}$$

$$\mathrm{RM}_{3D} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} + \begin{bmatrix} 0 & a_3 & a_4 \\ -a_3 & 0 & a_5 \\ -a_4 & -a_5 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{4}$$

In other words 3 degrees of freedom (dofs), i.e., a_0, a_1, a_2, in 2D and 6 dofs in 3D.

The strain of a rigid motion

The strain of a rigid motion is zero. CHECK! Consequences:

- Let r be a rigid motion
- then $\epsilon(r) = 0$ and therefore
  - $\nabla \cdot \epsilon(r) = 0$
  - $\operatorname{tr} \epsilon(r) = 0$
  - $\nabla \cdot \operatorname{tr} \epsilon(r) = 0$

Our equation

$$-2\mu(\nabla \cdot \epsilon(u)) - \lambda \nabla(\nabla \cdot u) = f$$

is singular since if u solves the equation then so does $u + r$ because

$$-2\mu(\nabla \cdot \epsilon(r)) - \lambda \nabla(\nabla \cdot r) = 0$$

The same applies to Neumann conditions. Therefore the pure Neumann problem is the hardest!

Ways to remove the singularity

There are several ways to remove the singularity of the elasticity stiffness matrix A in the case of a pure Neumann problem:

- Pin-point the displacement in selected points (6 degrees of freedooms must be removed)
- Penalize rigid motions
- Orthogonalize solution and input data with respect rigid motions and use iterative solvers capable of ignoring the kernel
- Use the method of Lagrange multipliers to deal with the rigid motions

The first three methods here often require tuning and care. We will therefore not go through them in detail. The last method is the easiest and cleanest from a mathematical point of view, but the elasticity problem then changes from an elliptic problem to a saddle-point problem

Removing the rigid motions with the method of Lagrange multipliers

Find $u \in H^1$ and $r \in \mathrm{RM}$ such that

$$\begin{aligned} a(u,v) + b(r,v) &= f(v), \quad \forall v \in H^1 && (5) \\ b(s,u) &= 0, \quad \forall s \in \mathrm{RM} && (6) \end{aligned}$$

Here,

$$\begin{aligned} a(u,v) &= \mu(\epsilon(u),\epsilon(v)) + \lambda(\nabla \cdot u, \nabla \cdot v) && (7) \\ b(r,v) &= (r,v) && (8) \\ f(v) &= (f,v) + \int_{\partial\Omega} gv ds && (9) \end{aligned}$$

Code

(complete code found in the lecture notes)

```
V = VectorFunctionSpace(mesh, "Lagrange", 1)
R = FunctionSpace(mesh, 'R', 0)
M = MixedFunctionSpace([R]*6)
W = MixedFunctionSpace([V, M])
u, rs = TrialFunctions(W)
v, ss = TestFunctions(W)

# Establish a basis for the nullspace of RM
e0 = Constant((1, 0, 0))
e1 = Constant((0, 1, 0))
e2 = Constant((0, 0, 1))
e3 = Expression(('-x[1]', 'x[0]', '0'))
e4 = Expression(('-x[2]', '0', 'x[0]'))
e5 = Expression(('0', '-x[2]', 'x[1]'))
basis_vectors = [e0, e1, e2, e3, e4, e5]
```

Code, cont'd

```
e0 = Constant((1, 0, 0))
e1 = Constant((0, 1, 0))
e2 = Constant((0, 0, 1))
e3 = Expression(('-x[1]', 'x[0]', '0'))
e4 = Expression(('-x[2]', '0', 'x[0]'))
e5 = Expression(('0', '-x[2]', 'x[1]'))
basis_vectors = [e0, e1, e2, e3, e4, e5]

a = 2*mu*inner(epsilon(u),epsilon(v))*dx +
    lambda_*inner(div(u),div(v))*dx

# Lagrange multipliers contrib to a
for i, e in enumerate(basis_vectors):
  r = rs[i]
  s = ss[i]
  a += r*inner(v, e)*dx + s*inner(u, e)*dx
```

Shear vs compression

The equation of linear elasticity:

$$-2\mu(\nabla \cdot \epsilon(u)) - \lambda\nabla(\nabla \cdot u) = 0. \tag{10}$$

- the Lamé parameter λ represents how easily the material compressed.
- often $\lambda \gg \mu$
- physically this means that it is a lot easier to twist (cause shear) than to compress the material
- incompressible liquids are extreme examples that easily deform but are never compressed
- simple numerical schemes do not distinguish between compression and shearing
- if $\lambda \gg \mu$ standard schemes underestimate the deformation
- the phenomen is called *locking*

A mixed form linear elasticity to avoid locking

Let us introduce an artifical uknown "the solid pressure"

$$p = \lambda \nabla \cdot u$$

The equation of linear elasticity,

$$-2\mu(\nabla \cdot \epsilon(u)) - \lambda \nabla(\nabla \cdot u) = f$$

can then be written as

$$\begin{aligned} -2\mu(\nabla \cdot \epsilon(u)) - \nabla p &= f \\ \nabla \cdot u - \frac{1}{\lambda} p &= 0 \end{aligned}$$

This is just a re-write, we have done nothing wrong! But we would like to stress that the "solid pressure" is not a physical pressure in the normal sense

A mixed form linear elasticity to avoid locking

The strong form:

$$-2\mu(\nabla \cdot \epsilon(u)) - \nabla p = f \tag{11}$$

$$\nabla \cdot u - \frac{1}{\lambda} p = 0 \tag{12}$$

The weak form: Find $u \in H^1$ and $p \in L_2$ such that

$$\begin{aligned} \mu(\epsilon(u), \epsilon(v)) + (p, \nabla \cdot v) &= f(v), \quad \forall v \in H^1 \\ (\nabla \cdot u, q) - \frac{1}{\lambda}(p, q) &= 0, \quad \forall q \in L_2 \end{aligned}$$

This looks like the Stokes problem we looked into earlier! In fact the term $c(p, q)$ is non-harmful in this setting as it is a negative term. It even stabilize the system

Normal Stokes elements can be used here and we will optain optimal convergence rates

Exercise: check the convergence of a known scheme

We had the error estimate:

$$\|u - u_h\|_1 + \|p - p_h\|_0 \leqslant Ch^k \|u\|_{k+1} + Dh^{l+1} \|p\|_{l+1}$$

Check if this is correct by manufacturing a right-hand side (and bc) from a known solution. Assume that $u = \nabla \times sin(\pi xy)$ and compute the right-hand side as $\mathrm{f} = -2\mu(\nabla \cdot \epsilon(u)) - \lambda \nabla(\nabla \cdot u)$. The $\nabla\times$ operator is defined as $(-\frac{\partial}{\partial y}, \frac{\partial}{\partial x})$. Compute numerical solutions u_h and refinements of the unit square with various λ and check it the error depends on λ.