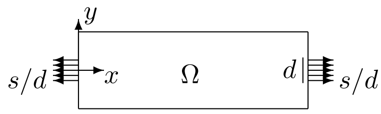
Mechanics I
Homework Mar. 13th; due Apr. 10rd

1. Consider the following rectangular 2D geometry of horizontal width w and vertical height h,



where a surface load s/d is applied in a d thick region at the center of both sides (denoted $\partial\Omega_1$; there are no body forces). The material has (using linearized elasticity) the stress-strain law $T = 2\mu\varepsilon(u)$. We ask ourselves how large the Gibbs free energy of the displacement u,

$$E^d[u] = \int_\Omega \mu|\varepsilon(u)|^2 \,\mathrm{d}x - \int_{\partial\Omega_1} \tfrac{s}{d} \cdot u \,\mathrm{d}a\,,$$

is in equilibrium. This is obviously a difficult task, since we cannot directly find a formula for the equilibrium displacement u (the minimizer of E).

- For $d = h$ we can find the equilibrium displacement u^h explicitly. State u^h, show that it is the equilibrium displacement, and compute its energy $E^h_{\min} = E^h[u^h]$.

For $d < h$ we will find a lower bound on the energy via convex duality, which requires a little work.

- Show that any twice weakly differentiable function $\phi : \Omega \to \mathbb{R}$ induces a stress field T via

$$T = \begin{pmatrix} \phi_{,22} & -\phi_{,12} \\ -\phi_{,21} & \phi_{,11} \end{pmatrix}$$

which satisfies $\mathrm{div}T = 0$ weakly.
In fact, one can show that any statically admissible stress field T can be expressed via the above formula for some ϕ, which is called the corresponding Airy stress function. The Airy stress function obviously is only determined up to an affine function.

- Compute the Airy stress function ϕ^h for the equilibrium stress in the case $d = h$, fixing $\phi^h(0,0) = 0$, $\nabla\phi^h(0,0) = 0$.
- Find an Airy stress function $\hat{\phi}^d$ belonging to a statically admissible stress (not necessarily the equilibrium stress—this is too difficult to obtain) for $d < h$, fixing $\hat{\phi}^d(0,0) = 0$, $\nabla\hat{\phi}^d(0,0) = 0$.
- We will now find a lower bound of the form

$$E^d_{\min} \leq E^h_{\min} - f(d,w,h,s)\,.$$

First, we construct an Airy stress function ϕ^d which belongs to a statically admissible stress for $d < h$ (and which probably comes closer to the true equilibrium stress than $\hat{\phi}^d$): We take $\phi^d = \phi^h + \phi$, where

- $\phi = \hat{\phi}^d - \phi^h$ at $x = 0$ and $x = w$,
- ϕ is rotationally symmetric on $\{(x, y) \,|\, x^2 + y^2 < (\frac{h}{2})^2\}$ around $(0, 0)$,
- ϕ is rotationally symmetric on $\{(x, y) \,|\, (x - w)^2 + y^2 < (\frac{h}{2})^2\}$ around $(w, 0)$,
- ϕ =const. elsewhere.

Compute the corresponding stress field and use it to find the above-mentioned bound via convex duality.

2. Find the equilibrium displacement from the previous question numerically, e. g. in Matlab, using finite elements and the parameters $w = 3$, $h = 1$, $d = 1/3$, $s = 1/10$. Note: You can find the meaning and documentation of all Matlab commands on `www.mathworks.com`.

   - First create a finite element mesh:

```
x=linspace(0,w,37);
y=linspace(0,h,13);
[X,Y]=meshgrid(x,y);
boundaryIndicator=zeros(13,37);
boundaryIndicator(:,[1,end])=1;
boundaryIndicator([1,end],:)=1;
vertices=[X(:),Y(:),boundaryIndicator(:)];
triangles=delaunay(vertices(:,1),vertices(:,2));
boundaryConditionXComponent=zeros(13,37);
boundaryConditionXComponent(5:9,1)=-0.1;
boundaryConditionXComponent(5:9,end)=0.1;
boundaryCondition=[boundaryConditionXComponent(:),zeros(13*37,1)];
```

     Each row of `vertices` now represents the (x, y)-coordinate of a vertex of the mesh; the third entry is 1 on $\partial\Omega$. `boundaryCondition` is a matrix whose k^{th} entry is the surface load on the k^{th} node if this node lies on the boundary.

   - Assemble the stiffness matrix `L` and the right-hand side `B` of the system, using the procedure from the lecture.
     In detail: First we assemble the stiffness matrix

$$L = \begin{pmatrix} L^{11} & L^{12} \\ L^{21} & L^{22} \end{pmatrix} .$$

     For this purpose, you have to run over all the triangles. Say the k^{th} triangle T_k has nodes $\hat{x}_{i_1}, \hat{x}_{i_2}, \hat{x}_{i_3}$ (ordered counterclockwise), where i_1, i_2, i_3 are just indices in $\{1, \ldots, N\}$. Only the basis functions φ_i^j

with $i \in \{i_1, i_2, i_3\}$ will be non-zero on T_k. First compute the affine transformation matrix $A_k = (\hat{x}_{i_2} - \hat{x}_{i_1} | \hat{x}_{i_3} - \hat{x}_{i_1})$ that transforms T_k onto the reference triangle T. Next compute

$$\tilde{L}^{jl}_{mn} := \int_{T_k} C\epsilon(\varphi^j_{i_m}) : \epsilon(\varphi^l_{i_n})\,\mathrm{d}x \text{ for } j, l \in \{1, 2\}, m, n \in \{1, 2, 3\}\,,$$

which via a pullback onto T can be written as

$$\tilde{L}^{jl} := \begin{pmatrix} \tilde{L}^{jl}_{11} & \tilde{L}^{jl}_{12} & \tilde{L}^{jl}_{13} \\ \tilde{L}^{jl}_{21} & \tilde{L}^{jl}_{22} & \tilde{L}^{jl}_{23} \\ \tilde{L}^{jl}_{31} & \tilde{L}^{jl}_{32} & \tilde{L}^{jl}_{33} \end{pmatrix} = \begin{pmatrix} -1 & -1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \left[\frac{\det A_k}{2} A_k^{-1} \begin{pmatrix} C_{1j1l} & C_{1j2l} \\ C_{2j1l} & C_{2j2l} \end{pmatrix} A_k^{-T} \right] \begin{pmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} \text{ for } j, l \in \{1, 2\}\,.$$

Now add the (m, n)-entry of $\tilde{L}^{jl}$ onto the (i_m, i_n)-entry of L^{jl}. After running over all triangles, the assembly of L is finished. An implementation trick: Instead of initializing the L^{jl} as $N \times N$-matrices full of zeros and then adding up entries in this matrix, one can use the command `sparse(i,j,s,m,n)` which you can look up online. It creates an empty `m`×`n`-matrix and then goes through the entries of `i,j,s`, adding `s(k)` onto the (`m(k)`,`n(k)`)-entry of the matrix, so you only have to produce the vectors `i,j,s`.

Next, we assemble the vector $B = \begin{pmatrix} B^1 \\ B^2 \end{pmatrix}$ which contains the surface loads. Again, you have to run over all triangles. If triangle T_k with nodes i_1, i_2, i_3 has one side on $\partial\Omega$ (i. e. two of its nodes have a nonzero third entry in `vertices`, say i_1 and i_2), then we have to compute

$$\tilde{B}^j_m := \int_{\overline{T_k} \cap \partial\Omega} s \cdot \varphi^j_{i_m}\,\mathrm{d}a \text{ for } j \in \{1, 2\}, m \in \{1, 2\}\,.$$

Writing $s = \sum_{n=1}^N \sum_{l=1}^2 s^l_n \phi^l_n$ (note that (s^1_n, s^2_n) is the n^{th} row of `boundaryCondition`), this becomes

$$\begin{aligned} \tilde{B} &:= \begin{pmatrix} \tilde{B}^1_1 & \tilde{B}^2_1 \\ \tilde{B}^1_2 & \tilde{B}^2_2 \end{pmatrix} = (\tilde{B}^j_m)_{mj} \\ &= \left(\sum_{n=1}^2 \sum_{l=1}^2 s^l_{i_n} \int_{\overline{T_k} \cap \partial\Omega} \varphi^l_{i_n} \cdot \varphi^j_{i_m}\,\mathrm{d}a \right)_{mj} \\ &= \frac{|\hat{x}_{i_1} - \hat{x}_{i_2}|}{6} \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} s^1_{i_1} & s^2_{i_1} \\ s^1_{i_2} & s^2_{i_2} \end{pmatrix}\,. \end{aligned}$$

Now $\tilde{B}^j_m$ has to be added onto the ${i_m}^{\text{th}}$ entry of B^j.

- Solve for the vector of the displacement:

  `U=B\L`

- Visualize the displacement:

```
subplot(2,1,1);
trimesh(triangles,vertices(:,1),vertices(:,2));
hold on;
quiver(X,Y,reshape(U(1:(13*37)),13,37),reshape(U(13*37+1:end),13,37));
subplot(2,1,2);
trimesh(triangles,vertices(:,1)+U(1:(13*37)),vertices(:,2)+U(13*37+1:end));
```

- Compute the equilibrium displacement for a refined mesh and attach the total code as well as a printout of the result to your homework.