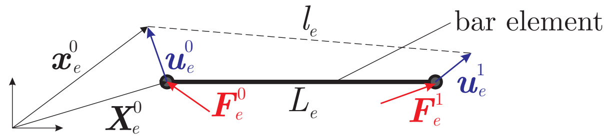
Homework Set #4

assigned: Friday, October 28th, 2016
due: Friday, November 11th, 2016, 3 pm
drop boxes outside 374 Firestone

Problem 1: bar element (15 points).



Let us consider a *2-node bar element* with energy density $W(\varepsilon)$ depending only on the axial bar strain ε, so the total energy of a single bar element e of original length $L_e = \|\boldsymbol{X}_1 - \boldsymbol{X}_0\|$ and cross-sectional area A_e is

$$I_e = \int_{\Omega_e} W(\varepsilon)\,\mathrm{d}V = \int_0^{L_e} W(\varepsilon) A_e\,\mathrm{d}X.$$

For the implementation of $W(\varepsilon)$, let us re-use the 1D linear elastic material model of problem set #2.

For the 2-node bar element, we use the linear interpolation with X running along the bar's axis:

$$\boldsymbol{u}_e^h(X) = N_e^0(X)\boldsymbol{u}_e^0 + N_e^1(X)\boldsymbol{u}_e^1,$$

whose shape functions N_e^a are obtained from the Kronecker identity $N^a(X_b) = \delta_{ab}$. Since the shape functions are linear, the strain in the bar is constant inside each bar element. Therefore, we have

$$I_e^h = \int_0^{L_e} W(\varepsilon^h) A_e\,\mathrm{d}X = W(\varepsilon_e^h) A_e L_e \qquad \Rightarrow \qquad \boxed{I_e^h = W(\varepsilon_e^h) A_e L_e}$$

with the approximate axial bar strain in element e,

$$\varepsilon_e^h = \frac{l_e - L_e}{L_e} = \frac{\|\boldsymbol{x}_e^1 - \boldsymbol{x}_e^0\| - L_e}{L_e} = \frac{\|\boldsymbol{X}_e^1 + \boldsymbol{u}_e^1 - (\boldsymbol{X}_e^0 + \boldsymbol{u}_e^0)\|}{L_e} - 1,$$

where $\boldsymbol{x}$ and $\boldsymbol{X}$ are, respectively, the deformed and undeformed nodal positions of the bar.

(a) Show that the internal force vector $\boldsymbol{F}_e^a = \partial I_e / \partial \boldsymbol{u}_e^a$ acting on node a follows as

$$\boxed{\boldsymbol{F}_e^a = \pm A_e \sigma(\varepsilon_e^h)\hat{\boldsymbol{l}}_e} \qquad \text{where} \qquad \hat{\boldsymbol{l}}_e = \frac{\boldsymbol{l}_e}{l_e} = \frac{\boldsymbol{x}_e^1 - \boldsymbol{x}_e^0}{\|\boldsymbol{x}_e^1 - \boldsymbol{x}_e^0\|}$$

is the axial unit vector along the deformed bar, and $\sigma(\cdot) = W'(\cdot)$.

(b) For the iterative solution, we need the element stiffness matrix $\boldsymbol{K}$ with $K_{ij}^{ab} = \partial F_i^a / \partial u_j^b$. Show that

$$\boxed{\boldsymbol{K}^{ab} = \pm A_e \left[\frac{\mathbb{C}(\varepsilon_e^h)}{L_e} - \frac{\sigma(\varepsilon_e^h)}{l_e}\right] \hat{\boldsymbol{l}}_e \otimes \hat{\boldsymbol{l}}_e + A_e \frac{\sigma(\varepsilon_e^h)}{l_e}\boldsymbol{I}} \qquad \text{with} \qquad \mathbb{C}(\cdot) = W''(\cdot).$$

(c) In simulations, one often assumes linearized kinematics in which case the above framework simplifies significantly. Here, the approximate axial strain is obtained from projecting the deformed bar onto the undeformed bar axis, so that

$$\varepsilon_e^h = \frac{(\boldsymbol{x}_e^1 - \boldsymbol{x}_e^0)\cdot\hat{\boldsymbol{L}}_e - L_e}{L_e} = (\boldsymbol{x}_e^1 - \boldsymbol{x}_e^0)\cdot\hat{\boldsymbol{L}}_e - 1 \qquad \text{with} \qquad \hat{\boldsymbol{L}}_e = \frac{\boldsymbol{X}_e^1 - \boldsymbol{X}_e^0}{\|\boldsymbol{X}_e^1 - \boldsymbol{X}_e^0\|}.$$

For this case, derive the nodal forces and stiffness matrix. Show that the latter is constant.

(Coding) Problem 2: bar element (25 points).

Let us implement the above *2-node bar element* in 3D in our code as a new class `Elements::TwoNodeBar::Finite` with the following functionality in analogy to the material models:

- The class constructor receives the *undeformed nodal positions* $\{\boldsymbol{X}_e^0, \boldsymbol{X}_e^1\}$, a *1D material model* (for W, σ, $\mathbb{C}$), and the *element properties* (here, simpy the cross-sectional area A_e).
- Method `computeEnergy` turns nodal displacements $\{\boldsymbol{u}_e^0, \boldsymbol{u}_e^1\}$ into the total bar energy I_e.
- Method `computeForces` turns nodal displacements $\{\boldsymbol{u}_e^0, \boldsymbol{u}_e^1\}$ into nodal forces $\left\{\boldsymbol{F}_e^0, \boldsymbol{F}_e^1\right\}$.
- Method `computeStiffnessMatrix` turns displacements $\{\boldsymbol{u}_e^0, \boldsymbol{u}_e^1\}$ at nodes into the element stiffness matrix $\boldsymbol{K}$. Note that we return a matrix $\boldsymbol{C} \in \mathbb{R}^{6\times 6}$ whose components are such that $C(d\cdot a+i, d\cdot b+j) = K_{ij}^{ab}$ (with node numbers $a, b = \{0, 1\}$ and coordinates $i, j = \{0, \dots, 2\}$), as discussed in class.
- Methods `computeBarStress` and `computeBarStrain` turn nodal displacements into the (constant) tensile/compressive bar stress $\sigma(\varepsilon_e^h)$ and strain ε_e^h. These methods are needed only for visualization.

You can test the implementation of your element by using the available method `testElementDerivatives`, which is analogous to the material model test you wrote on homework set #3 and checks if derivatives are consistent. Please use a general element for testing, i.e., pick some non-trivial nodal locations and not, e.g., simply $(0, 0, 0)$ and $(1, 0, 0)$ since the latter could accidentally lead to a passing of the test.

(Coding) Problem 3: linearized bar element (10 points).

Let us implement the *linearized 2-node bar element* in 3D in our code as a new class `Elements::TwoNodeBar::Linear`.

Hint: You can simply duplicate and modify your finite-strain bar model (the changes are indeed very small).

Problem 4: assembler (20 points).

We ask you to write a class `Assembler` that realizes the assembly of finite element quantities into their respective global quantities, as discussed in class. The new class should have the following methods:

- The *constructor* receives and stores the vector of all elements and the number of nodes.
- Method `assembleEnergy` computes the total energy $I = \sum_e I_e$ from individual elements.
- Method `assembleForceVector` computes the global force vector $\boldsymbol{F}_{\text{int}}$ by assembling force vectors $\{\boldsymbol{F}_e^0, \boldsymbol{F}_e^1\}$ of all elements.
- Method `assembleStiffnessMatrix` computes the global stiffness matrix $\boldsymbol{K}$ by assembling stiffness matrices $\boldsymbol{K}_e$ of all elements.

Each method receives the global displacement vector $\boldsymbol{U}^h$, which must be broken down into element displacement vectors $\{\boldsymbol{u}_e^0, \boldsymbol{u}_e^1\}$ for each element. Next, you can "ask" each element for its contributions and assemble those.

(Coding) Problem 5: boundary value problem (20 points).

Let us solve our first boundary value problem. This requires the following:

- creation of a mesh (nodal locations and element connectivities are read in from a text file);
- creation of a material model (let us use the 1D linear elastic model from set #2);
- creation of an `assembler`, which interfaces with all elements to assemble the global energy, nodal forces and stiffness matrix by requesting the individual element quantities from the respective elements and assembling them into large (global) vectors and matrices;
- creation of `essentialBoundaryConditions`: let us restrict us here to only essential boundary conditions. That is, to a few selected nodes we assigned prescribed displacements (each such BC requires a triple of `nodeId`, `coordinate`, and `prescribedDisplacement`).
- creation of an iterative `solver` (we provide a Newton-Raphson solver that uses the assembler and the essential BCs to find the sought displacement at all nodes).

Use all of the above along with your material model and bar element to solve a static boundary value problem. Create a truss consisting of at least ten bars, apply admissible boundary conditions, and solve for the displacements. We provide functionality to plot your undeformed and deformed configurations (along with stresses and strains) using *ParaView*.

(Coding) Problem 6: boundary value problem (10 points).

Repeat your boundary value problem from the previous problem, this time using the linearized-kinematics bar elements. Show that for small imposed displacements the solutions agree, whereas they disagree for large deformation.

total: 100 points