

Assignment #5 – A Gently-Curved Beam Element

This assignment is designed to carefully review the slightly curved beam/shallow arch formulation from Monday's lesson. The simple yet powerful non-linear curved beam theory is obtained by modeling a beam along the x -axis and adding a given elevation function (or imperfection function) $h(x)$.

Incorporating the elevation function in the reference and the deformed configuration leads to the kinematic relations

$$\varepsilon_0 = u'_0 + h'v' + \frac{1}{2}(v')^2 \quad \text{and} \quad \phi = v'' , \quad (1,2)$$

where $u_0(x)$ is the displacement parallel to the x -axis (not necessarily parallel to the now curved beam axis), and $v(x)$ is the deflection measured perpendicular to the x -axis.

The proper weak form for the development of a finite element follows as

$$\begin{aligned} \delta\Pi(u_0, v) &\equiv -G(u_0, v; \delta u_0, \delta v) \\ &= \int_{\ell} \delta\tilde{W}(\varepsilon_0, \phi) dx + \int_{\ell} w(x) \delta v(x) dx - \bar{P}u_0(\ell) + \bar{R}v(\ell) - \bar{M}\delta v'_\ell \\ &= \int_{\ell} (F(\varepsilon_0, \phi) \delta\varepsilon_0 + M(\varepsilon_0, \phi) \delta\phi) dx + \int_{\ell} w(x) \delta v(x) dx - \bar{P}\delta u_{0,\ell} + \bar{R}\delta v_\ell - \bar{M}\delta v'_\ell \end{aligned} \quad (3)$$

with, in the linear elastic case, constitutive relations

$$F(\varepsilon_0, \phi) = EA\varepsilon_0 \quad \text{and} \quad M(\varepsilon_0, \phi) = EI\phi . \quad (4,5)$$

Only variations $\delta u_0(x)$ and $\delta v(x)$ of the primary functions $u_0(x)$ and $v(x)$ are independent variations.

Problem 5-1: Linearization of the weak form.

Using the above weak form, derive the linearized weak form $d\delta\Pi(u_0, v; du_0, dv; \delta u_0, \delta v)$

Problem 5-2: Finite Element for the curved beam

Design a simple curved beam element that uses two end-nodes only. For each node, the values for x and $y = h$ are given as coordinates. Use linear shape functions for $u_0^h(x)$ and cubic shape functions¹ for $v^h(x)$ such that

$$u_0^h(x) = \left(1 - \frac{x}{L_x}\right) u_i + \left(\frac{x}{L_x}\right) u_j$$

and

$$v^h(x) = N_1(x) v_i + N_2(x) \theta_i + N_3(x) v_j + N_4(x) \theta_j ,$$

¹ $N_1(\xi) = 1 - 3\xi^2 + 2\xi^3$, $N_2(\xi) = L(\xi - 2\xi^2 + \xi^3)$, $N_3(\xi) = 3\xi^2 - 2\xi^3$, $N_4(\xi) = L(-\xi^2 + \xi^3)$, with $\xi = x/L$.

where u_i and u_j are the horizontal nodal displacements, v_i and v_j are the vertical nodal displacements, and $\theta_i = dv^h/dx|_{x_i}$ and $\theta_j = dv^h/dx|_{x_j}$ are the nodal rotations at nodes i (left end) and node j (right end) of the element. This way, each node will have 3 degrees of freedom.

The elevation function may be approximated as

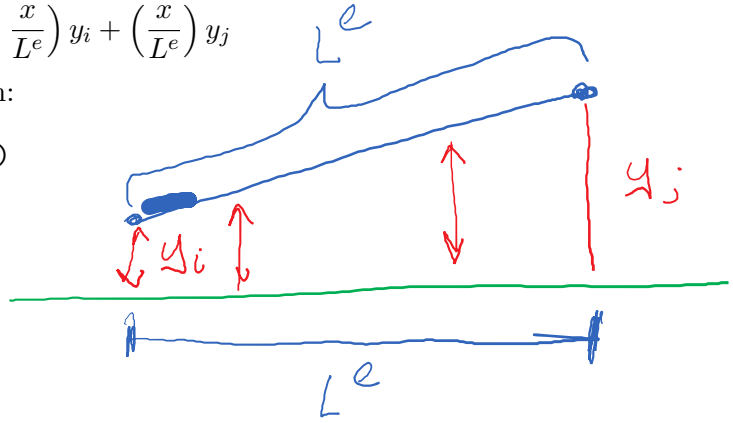
$$h(x) \approx \left(1 - \frac{x}{L^e}\right) y_i + \left(\frac{x}{L^e}\right) y_j$$

or provided via a user-defined matlab function:

```
function [h, dh, ddh] = elevation(x)
    h = ... % h(x)
    dh = ... % dh/dx(x)
    ddh = 0.. % d^2 h/dh^2(x)
end
```

or using python:

```
def elevation(x):
    h = ... # h(x)
    dh = ... # dh/dx(x)
    ddh = ... # d^2 h/dh^2(x)
    return (h, dh, ddh)
```



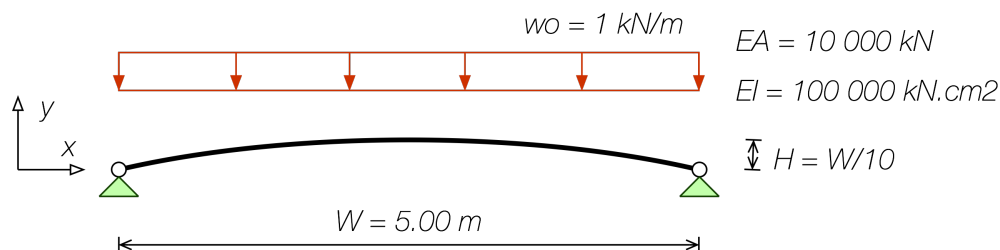
$$-\int_{\Omega} \mathbf{u} \cdot \mathbf{R}(\mathbf{u}) := \delta \pi(\mathbf{u}, \delta \mathbf{u})$$

Make sure to feed global $x = x_i + x_{local}^e$ to your function when calling it from within your element code.

1. Find the residual force vector for both nodes.
2. Find the tangent stiffness matrix (matrices, if thinking of $d\mathbf{f}_i := \sum_{j=1}^N \mathbf{k}_{ij} \cdot d\mathbf{u}_j$).
3. Implement appropriate matlab or python functions/classes/methods that return the nodal residuals and stiffness matrices.
4. Adjust your assembly function from Assignment #4 (hope you have one by now) such that you can assemble a curved beam or shallow arch. You should be able to use your Newton algorithm for load stepping or displacement control or arc-length algorithm with the new element force and element tangent stiffness matrix without much change to your algorithm. The biggest update should be the number of degrees of freedom per node from 2 to 3.

Problem 5-3: Study the snap-through behavior of a shallow arch

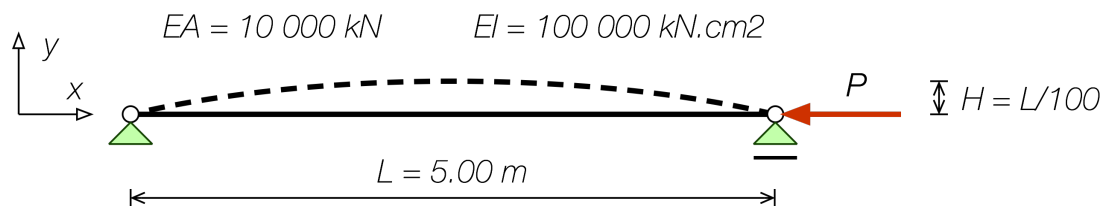
Model the snap-through behavior of the shown shallow arch using your arc-length implementation.



1. You will need a tiny imperfection to discover the full complex response. I suggest to vary the load pattern linearly to $0.99 w_0$ on the left end and $1.01 w_0$ on the right end of the arch. Note, that this doesn't change the load much over a single element and you may use the mean value over each element and assume the load constant for each element. That way, you can pre-integrate the load vector.
2. Perform a mesh refinement study: start with 4 elements, then double to 8, 16, 32, ... until the solution paths converge.² Check convergence for mid-point and quarter points.

Remark: This arch is shaped perfectly to generate a linear solution for which $M = 0$ and the horizontal force $F = \text{const}$. Compute that force to check your element for a sufficiently small load.

Problem 5-4: Buckling of a beam



1. Test your element by modeling a straight beam subjected to an axial load. You may use either load stepping as in Assignment #1 or displacement control as in Assignment #2 or the arc-length method as in Assignment #3 for gradually increasing the load level. Plot $\det \mathbf{K}_T$ over load level λ . You have reached the critical load when $\det \mathbf{K}_T = 0$.

You may need mesh refinement similar to Problem 5-3 for a good result.

2. Add a small imperfection by curving the beam such that the center deflection $\max h = h(L/2) = L/100$. Now perform the same test. Did $\det \mathbf{K}_T$ go to zero? Why? or Why not?

²Consistent refinement means every element gets split in two each refinement step. That means, that the number of elements doubles with a single refinement. Changing a mesh from 8 to 9 to 10 to ... elements will not yield a smooth convergence plot.