2020-05-18

Homework 5

CESG506

KRISTINN HLÍÐAR GRÉTARSSON

All code used for solving the homework can be found under course github.

Problem 1

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

$$\varepsilon_0 = u_0' + h'v' + \frac{1}{2}(v')^2 \qquad \text{and} \qquad \phi = v'' \,, \tag{1,2}$$

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

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

with, in the linear elastic case, constitutive relations

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

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 $d\,\delta\Pi(u_0, v; du_0, dv; \delta u_0, \delta v)$

5-1

$$\varepsilon_o = u_o' + h'v' + \tfrac{1}{2}(v')^2 \qquad \phi = v''$$

$$\delta\pi(u_o, v) \equiv -G(u_o, v, \delta u_o, \delta v)$$

$$= \int_L \delta\bar{W}(\varepsilon_o, \phi)\,dx + \int_L w(x)\,\delta v(x)\,dx - \bar{P}u_o(L) + \bar{R}v(L) - \bar{M}\delta v_L'$$

$$= \int_L \big(F(\varepsilon_o, \phi)\,\delta\varepsilon_o + M(\varepsilon_o, \phi)\,\delta\phi\big)\,dx + \int_L w(x)\,\delta v(x)\,dx - \bar{P}\delta u_{o_L} + \bar{R}\delta v_L - \bar{M}\delta v_L'$$

$$\delta\varepsilon_o = \delta\big(u_o' + h'v' + \tfrac{1}{2}(v')^2\big) = \delta u_o' + h'\delta v' + \delta v' v'$$

$$d\delta\varepsilon_o = d(\delta u_o' + h'\delta v' + \delta v' v') = dv'\delta v'$$

$$\delta\phi = \delta(v'') = \delta v''$$

$$d\delta\phi = d(\delta v'') = 0$$

$$d\delta\pi = d\Big(\int_L \big(F(\varepsilon_o, \phi)\,\delta\varepsilon_o + M(\varepsilon_o, \phi)\,\delta\phi\big)\,dx + \int_L w(x)\,\delta v\,dx - \bar{P}\delta u_{o_L} + \bar{R}\delta v_L - M\delta v_L'\Big)$$

$$= \int_L \Big(\frac{\partial F}{\partial \varepsilon_o} d\varepsilon_o \delta\varepsilon_o + \frac{\partial F}{\partial \phi} d\phi\,\delta\varepsilon_o + \frac{\partial M}{\partial \varepsilon_o} d\varepsilon_o \delta\phi + \frac{\partial M}{\partial \phi} d\phi\,\delta\phi + F(\varepsilon_o, \phi)\,d\delta\varepsilon_o \ldots$$

$$\ldots + M(\varepsilon_o, \phi)\underbrace{d\delta\phi}_{=0}\Big)\,dx$$

Linear elastic case with $F(\varepsilon_o, \phi) = EA\varepsilon_o$ & $M(\varepsilon_o, \phi) = EI\phi$

$$d\delta\pi = \int_L EA\big(du_o' + (h' + v')dv'\big)\big(\delta u_o' + h'\delta v' + \delta v' v'\big)\,dx \ldots$$

$$\ldots + \int_L EA\big(u_o' + h'v' + \tfrac{1}{2}(v')^2\big)\,dv'\delta v'\,dx$$

$$\ldots + \int_L EI\,dv''\,\delta v''\,dx$$

Problem 2

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 functions[1] 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_i 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

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

L^e

Make sure to feed global $x = x_i + x^e_{local}$ 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.

5-2

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

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

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

w/ $\xi = x/L$

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

1. $-\delta u \cdot \underline{R}(\underline{u}) := \delta\pi(u, \delta u)$

$$-\delta\underline{u}\cdot\underline{R}(u) = \int_L \left(F(\varepsilon_0,\phi)\,\delta\varepsilon_0 + M(\varepsilon_0,\phi)\,\delta\phi\right)dx$$

$$= \int_L \left(F(\varepsilon_0,\phi)\right)\left(\delta u_0' + h'\delta v' + v'\delta v'\right)dx + \int_L M(\varepsilon_0,\phi)\,\delta v''\,dx$$

$$= \int_L \left(F(\varepsilon_0,\phi)\left([\underline{N}^u]'\{\delta\underline{u}\} + (h'+v')[\underline{N}^v]'\{\delta\underline{v}\}\right)dx \;\ldots\right.$$

$$\ldots + \int_L M(\varepsilon_0,\phi)\,[\underline{N}^v]''\{\delta v\}\,dx$$

$$= \begin{Bmatrix} \int_L F(\varepsilon_0,\phi)\,[\underline{N}^u]'\,dx \\ \int_L M(\varepsilon_0,\phi)\,[\underline{N}^v]''\,dx + \int_L \underbrace{(h'+v')}_{F(\varepsilon_0,\phi)}[\underline{N}^v]'\,dx \end{Bmatrix}^T \begin{Bmatrix} \delta\underline{u} \\ \delta\underline{v} \end{Bmatrix}$$

5-2

2.

From 5-1 we have

$$d\delta\pi = \int_L EA(du_0' + (h'+v')dv')(\delta u_0' + (h'+v')\delta v')\,dx \ldots$$

$$\ldots + \int_L EA(u_0' + h'v' + \tfrac{1}{2}(v')^2)\,dv'\delta v'\,dx + \int_L EI\,dv''\delta v''\,dx$$

Sub in our approximations

$$d\delta\pi^h = \int_L EA([\underline{N}^u]'\{d\underline{u}\} + (h' + [\underline{N}^v]'\{\underline{v}\})[\underline{N}^v]'\{d\underline{v}\})([\underline{N}^u]'\{\delta\underline{u}\} + (h' + [\underline{N}^v]'\{\underline{v}\})[\underline{N}^v]'\{\delta\underline{v}\})\,dx \ldots$$

$$\ldots + \int_L EA([\underline{N}^u]'\{\underline{u}\} + h'[\underline{N}^v]'\{\underline{v}\} + \tfrac{1}{2}([\underline{N}^v]'\{\underline{v}\})^2)[\underline{N}^v]'\{d\underline{v}\}[\underline{N}^v]'\{\delta\underline{v}\}\,dx \ldots$$

$$\ldots + \int_L EI[\underline{N}^v]''\{d\underline{v}\}[\underline{N}^v]''\{\delta\underline{v}\}\,dx$$

$$= \int_L EA([\underline{N}^u]'\{d\underline{u}\})([\underline{N}^u]'\{\delta\underline{u}\})\,dx + \int_L EA((h' + \overbrace{[\underline{N}^v]'\{\underline{v}\}}^{v'})[\underline{N}^v]'\{d\underline{v}\})([\underline{N}^u]'\{\delta\underline{u}\})\,dx \ldots$$

$$+ \int_L EA([\underline{N}^u]'\{d\underline{u}\})((h' + [\underline{N}^v]'\{\underline{v}\})[\underline{N}^v]'\{\delta\underline{v}\})\,dx \ldots$$

$$+ \int_L EA((h' + [\underline{N}^v]'\{\underline{v}\})[\underline{N}^v]'\{d\underline{v}\})((h' + [\underline{N}^v]'\{\underline{v}\})[\underline{N}^v]'\{\delta\underline{v}\})\,dx \ldots$$

$$+ \int_L F(\varepsilon_0, \phi)[\underline{N}^v]'\{d\underline{v}\}[\underline{N}^v]'\{\delta\underline{v}\}\,dx \ldots$$

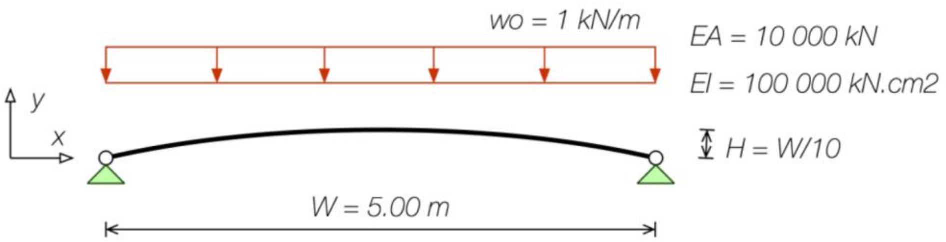
$$+ \int_L EI[\underline{N}^v]''\{d\underline{v}\}[\underline{N}^v]''\{\delta\underline{v}\}\,dx$$

$$[d\underline{u}, d\underline{v}]\underbrace{\begin{bmatrix} \int_L EA[\underline{N}^u]'^T[\underline{N}^u]'\,dx & \int_L EA(h'+v')[\underline{N}^u]'^T[\underline{N}^v]'\,dx \\ \int_L EA(h'+v')[\underline{N}^v]'^T[\underline{N}^u]'\,dx & \int_L (EA(h'+v')^2 + F(\varepsilon_0,\phi))[\underline{N}^v]'^T[\underline{N}^v]'\,dx + \int_L EI[\underline{N}^v]''^T[\underline{N}^v]''\,dx \end{bmatrix}}_{[\underline{K}_T]}\begin{Bmatrix} \delta\underline{u} \\ \delta\underline{v} \end{Bmatrix}$$

Problem 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.



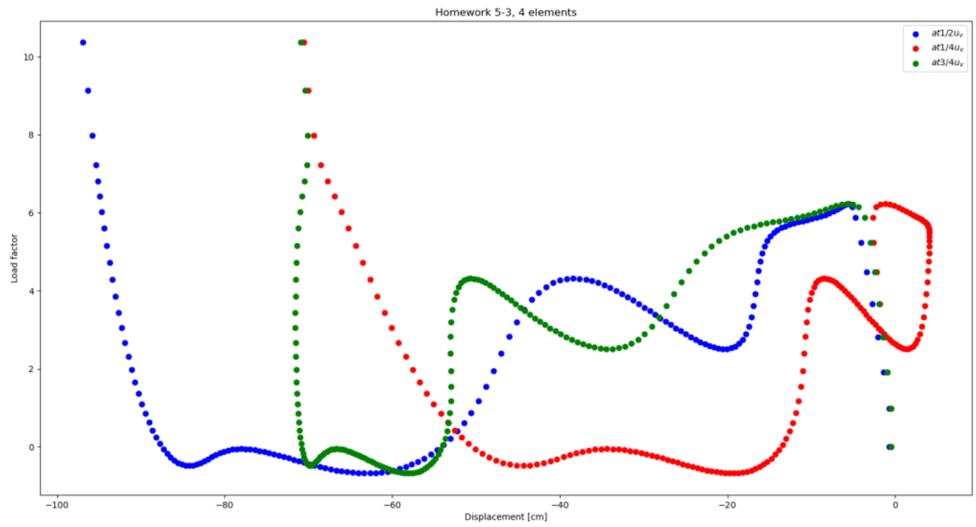
1. You will need a tiny imperfection to discover the full complex response. I suggest to vary the load pattern linearly to $0.99\,w_0$ on the left end and $1.01\,w_0$ on the right end of the arch. Note, that this doesn't change the load much over a single element and you may use the mean value over each element and assume the load constant for each element. That way, you can pre-integrate the load vector.

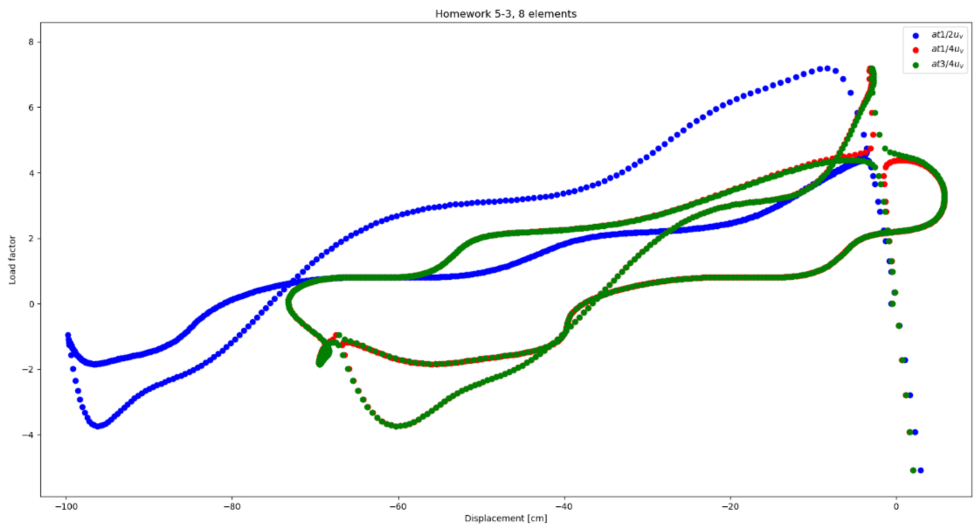
2. Perform a mesh refinement study: start with 4 elements, then double to 8, 16, 32, ...until the solution paths converge.[2] Check convergence for mid-point and quarter points.

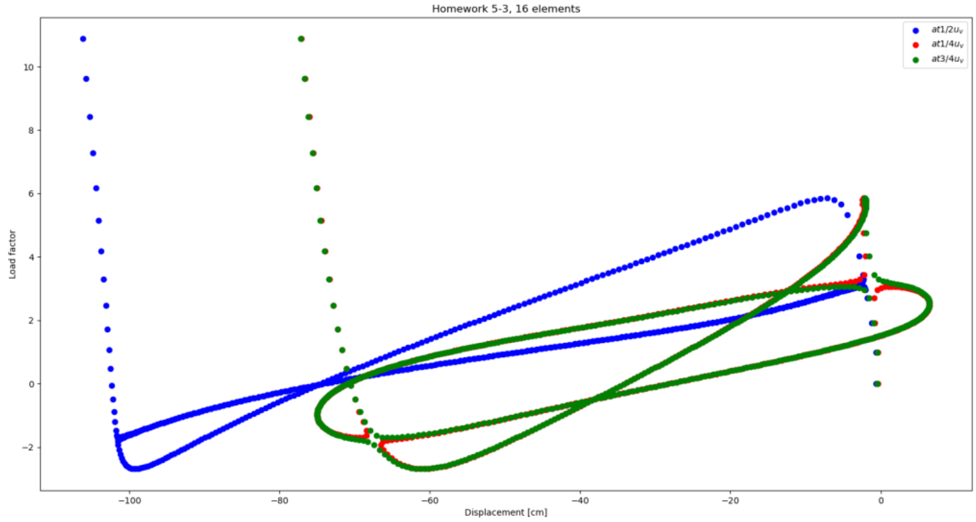
Solution

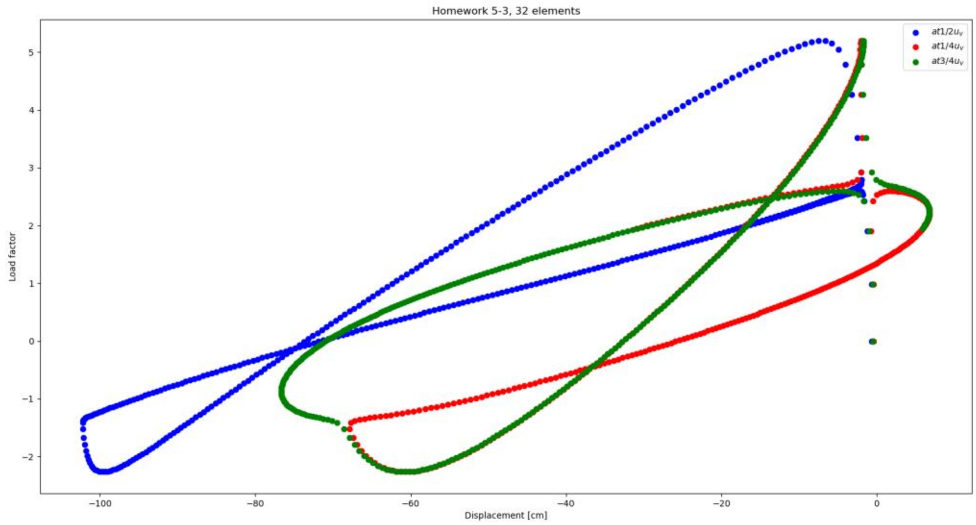
Code was run with increasingly finer mesh and vertical displacement of ¼ point, ½ point and ¾ point were plotted. The results can be seen in the images below.

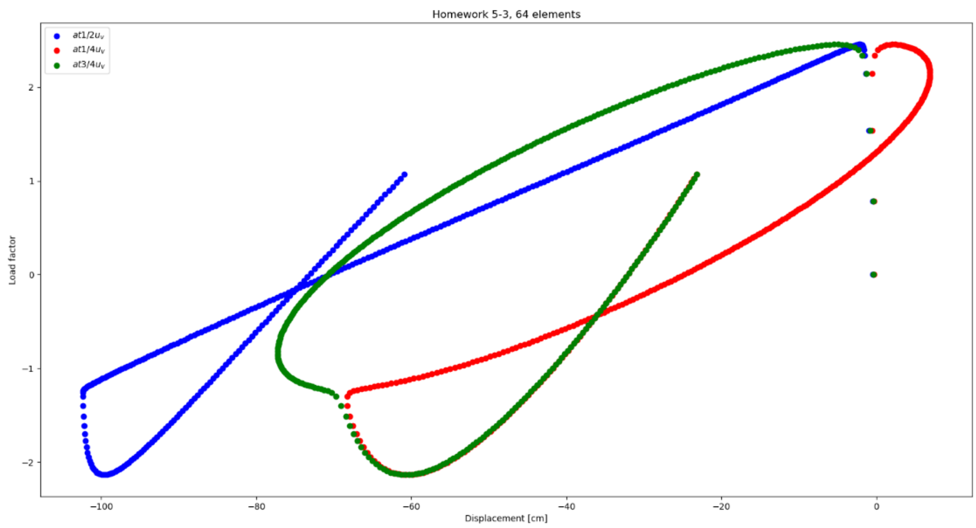
For meshes with 4 and 8 elements the beam behaves in a wildly different manner. Once the mesh refinement hits 16 elements or higher, the solution paths start to converge to a set path.





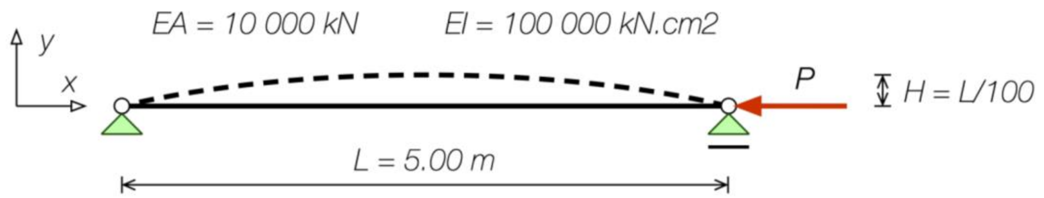

Homework 5-3, 16 elements
$at1/2u_v$
$at1/4u_v$
$at3/4u_v$
Load factor
Displacement [cm]


Homework 5-3, 32 elements
$at1/2u_v$
$at1/4u_v$
$at3/4u_v$
Load factor
Displacement [cm]


Homework 5-3, 64 elements
$at1/2u_v$
$at1/4u_v$
$at3/4u_v$
Load factor
Displacement [cm]

Problem 4

Problem 5-4: Buckling of a beam



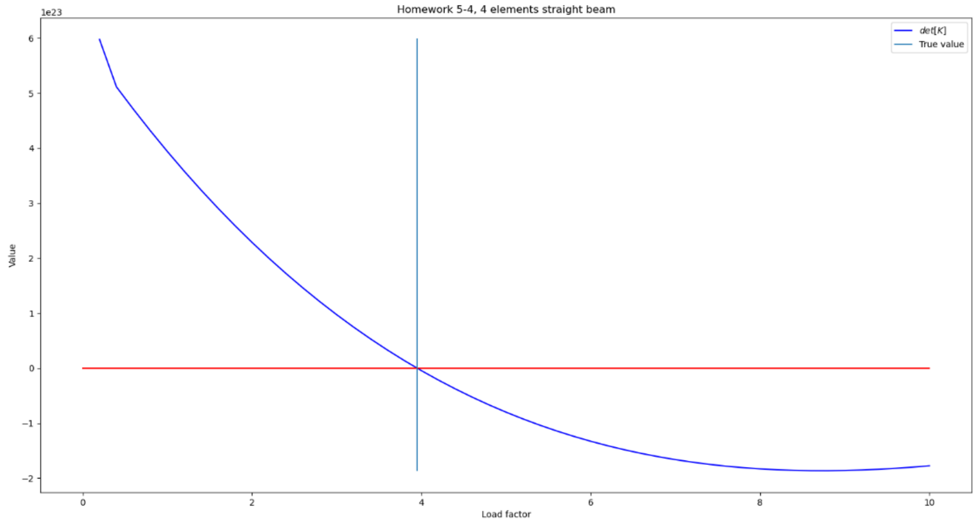
1. Test your element by modeling a straight beam subjected to an axial load. You may use either load stepping as in Assignment #1 or displacement control as in Assignment #2 or the arc-length method as in Assignment #3 for gradually increasing the load level. Plot $\det \mathbf{K}_T$ over load level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

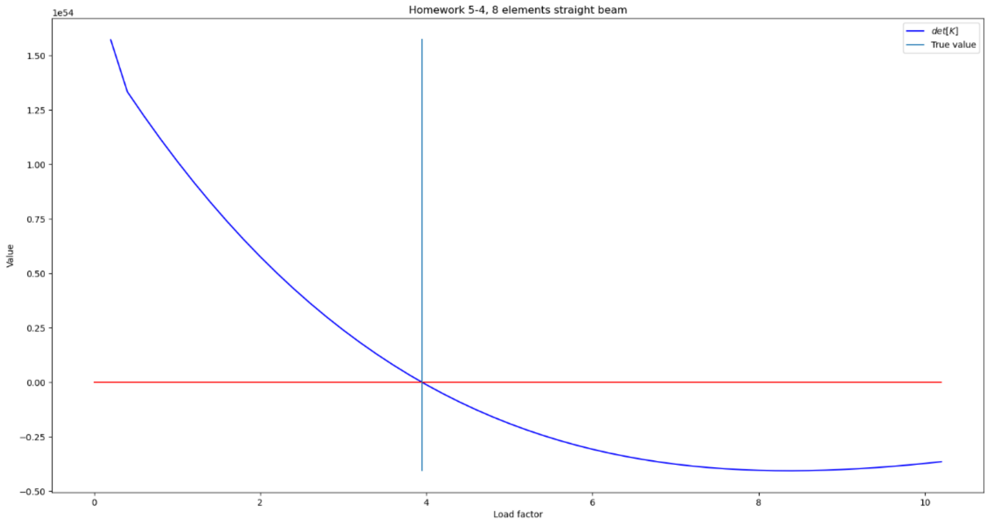
   You may need mesh refinement similar to Problem 5-3 for a good result.

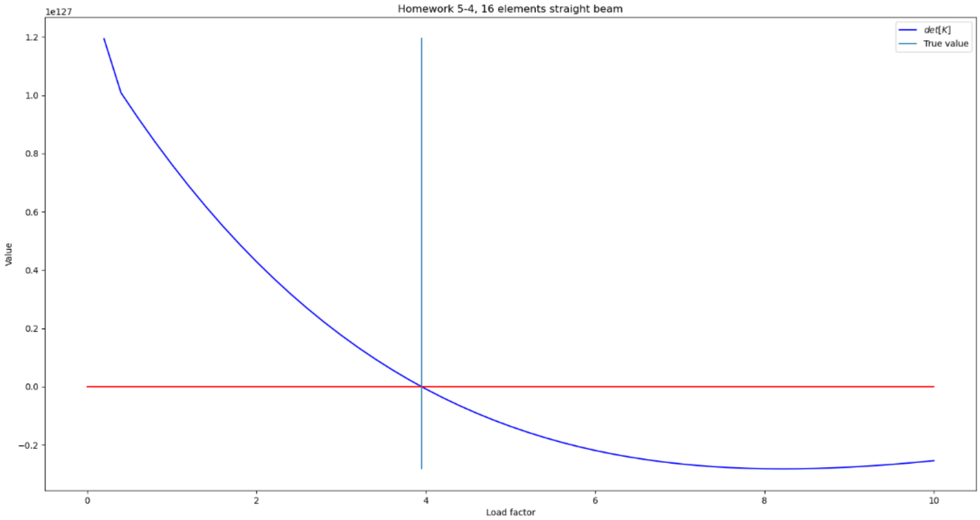
2. Add a small imperfection by curving the beam such that the center deflection $\max h = h(L/2) = L/100$. Now perform the same test. Did $\det \mathbf{K}_T$ go to zero? Why? or Why not?

Solution – Straight beam

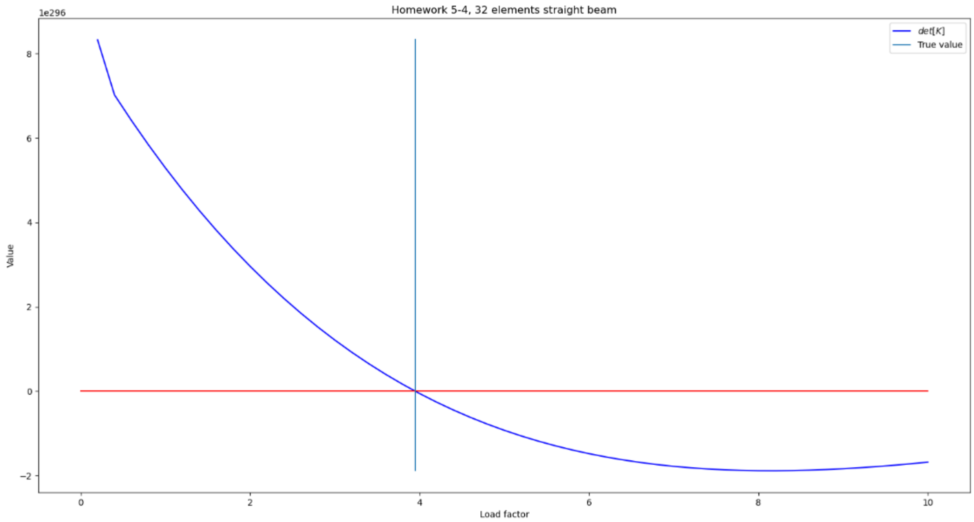
Plotting determinant of the stiffness matrix over load factor shows that the stiffness matrix becomes singular at the theoretical buckling capacity.







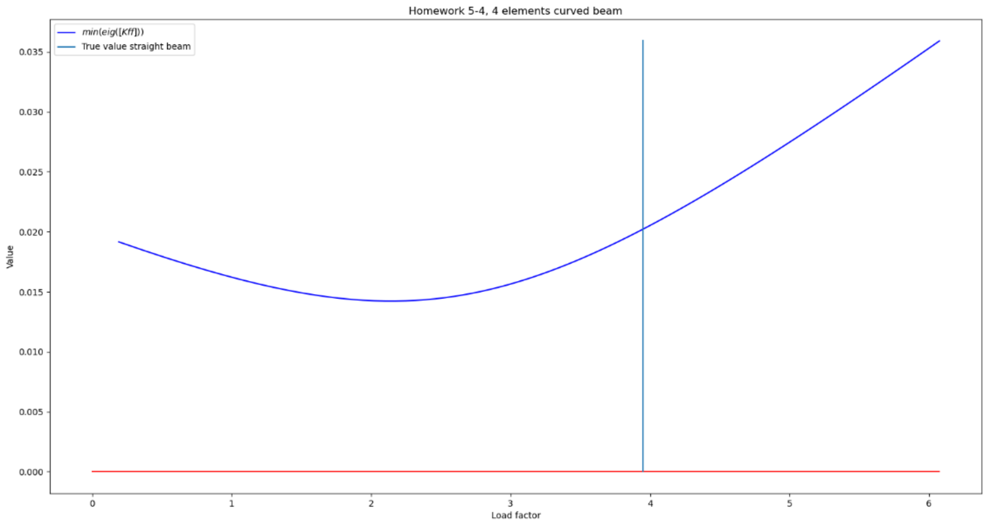
Homework 5-4, 16 elements straight beam
1e127
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
Value
0
2
4
6
8
10
Load factor
$det[K]$
True value

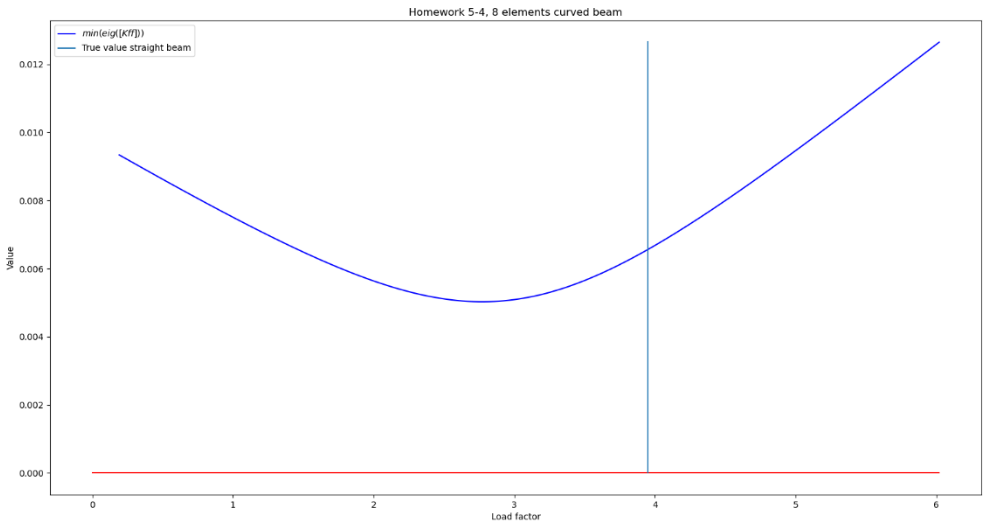


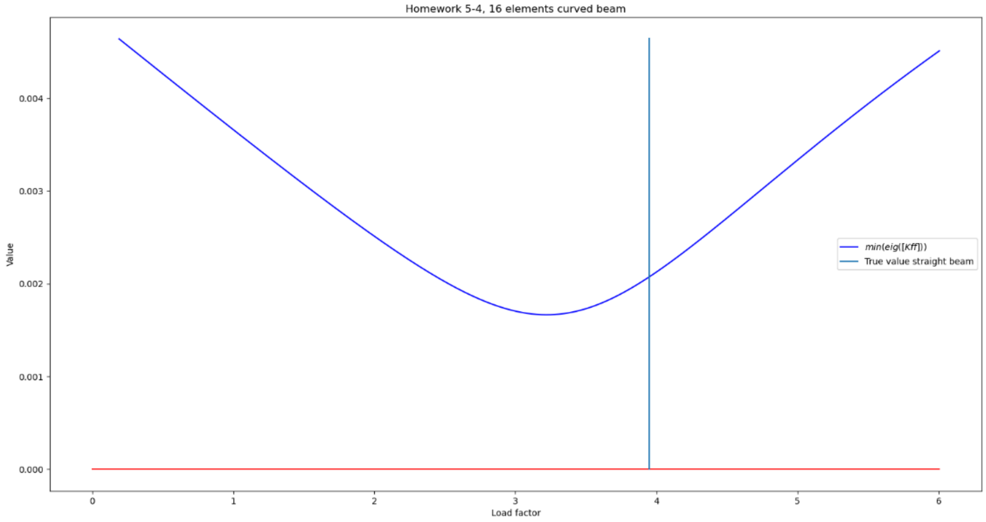
Homework 5-4, 32 elements straight beam
1e296
8
6
4
2
0
−2
Value
0
2
4
6
8
10
Load factor
$det[K]$
True value

Solution – Curved beam

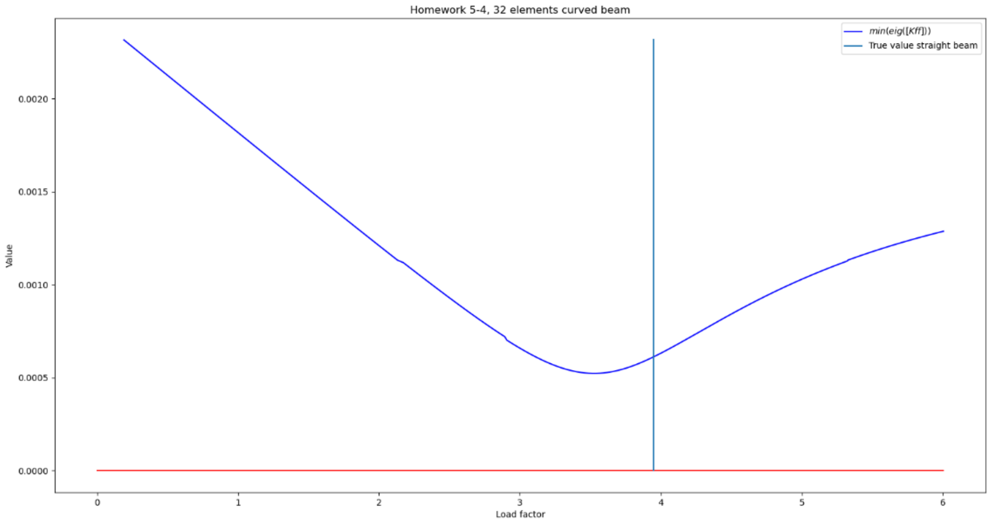
Adding a small curve to the beam changed the behaviour. The matrix no longer turned singular at any point. Furthermore, when the minimum eigenvalue of the stiffness matrix was plotted, it showed it also never became zero.



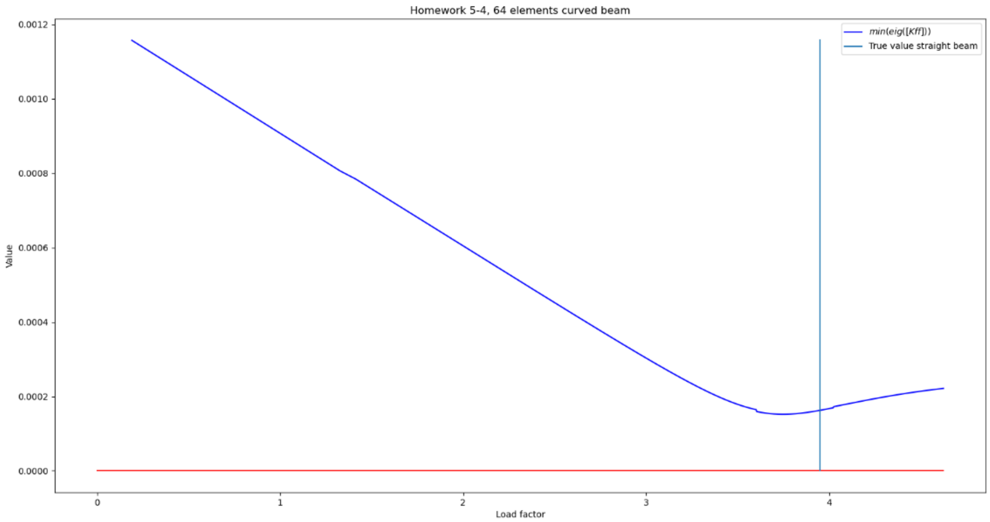




Homework 5-4, 16 elements curved beam
Value
Load factor
0.000
0.001
0.002
0.003
0.004
0
1
2
3
4
5
6
$min(eig([Kff]))$
True value straight beam



Homework 5-4, 32 elements curved beam
Value
Load factor
0.0000
0.0005
0.0010
0.0015
0.0020
0
1
2
3
4
5
6
$min(eig([Kff]))$
True value straight beam



Homework 5-4, 64 elements curved beam
Value
Load factor
0.0000
0.0002
0.0004
0.0006
0.0008
0.0010
0.0012
0
1
2
3
4
$min(eig([Kff]))$
True value straight beam