IN5270
MANDATORY EERCISE 3

ERIK JOHANNES B. L. G. HUSOM

OCTOBER 17, 2019

Contents

1. Description of the problem

The goal is to compute deflection of a cable with sine functions. We have a hanging cable with tension, and the cable has a deflection $w(x)$ which is governed by:

$$Tw''(x) = l(x), \tag{1}$$

where the variables are:
- L: Length of cable - T: Tension on cable - $w(x)$: Deflection of cable - $l(x)$: Vertical load per unit length

Cable is fixed at $x = 0$ and $x = L$, and the boundary conditions are $w(0) = w(L) = 0$. Deflection is positive upwards and l is positive when it acts downwards.

Assuming $l(x) = \text{const}$, the solution is symmetric around $x = L/2$. For a function $w(x)$ that is symmetric around a point x_0, we have that

$$w(x_0 - h) = w(x_0 + h), \tag{2}$$

which means that

$$(3)lim_{h->0}(w(x_0 + h) - w(x_0 - h))/(2h) = 0. \tag{3}$$

We can therefore halve the domain, since it is symmetric. That limits the problem to find $w(x)$ in $[0, L/2]$, with boundary conditions $w(0) = 0$ and $w'(L/2) = 0$.

Scaling of variables:

$$\overline{x} = x/(L/2) \qquad \text{(setting } x = \overline{x} \text{ in code for easier notation)} \tag{4}$$
$$u = w/w_c \qquad \text{(where } w_c \text{ is a characteristic size of } w\text{)} \tag{5}$$

By putting this into the original equation we get

$$\frac{4Tw_c}{L^2}u''(\overline{x}) = l = \text{const.} \tag{6}$$

We set $|u''(\overline{x})| = 1$, and we get $w_c = 0.25lL^2/T$, and the scaled problem is

$$u'' = 1, \overline{x} \in (0,1), u(0) = 0, u'(1) = 0. \tag{7}$$

2. Exact solution of u

The exact solution of u is easily found by integrating two times, and using the boudnary conditions to find the unknown constants:

$$u'' = 1, \tag{8}$$

$$u' = x + C, \tag{9}$$

$$u'(1) = 0 \Rightarrow C = -1, \tag{10}$$

$$u = \frac{1}{2}x^2 - x + D, \tag{11}$$

$$u(0) = 0 \Rightarrow D = 0, \tag{12}$$

$$u = \frac{1}{2}x^2 - x. \tag{13}$$

3. Using two P1 elements to approximate the function

In our case, with two elements, we will have three nodes. The basis function of the three nodes are easily found by using Lagrange polynomials, which is calculated within each element Ω_i that the basis function is defined in (since we only have two elements in this case, the middle basis function φ_1 is defined in both elements, while the other two basis functions are defined in only one element). We have a constant distance between each node $h = 0.5$, and our nodes is $x_0 = 0, x_1 = 0.5, x_2 = 1$. Then we get:

$$\varphi_0 = \begin{cases} \frac{x - x_{i+1}}{x_i - x_{i+1}} = \frac{x - x_1}{-h} = -2x + 1 & x_0 \le x < x_1 \end{cases}$$

$$\varphi_1 = \begin{cases} \frac{x - x_{i-1}}{x_i - x_{i-1}} = \frac{x - x_0}{h} = 2x & x_0 \le x < x_1 \\ \frac{x - x_{i+1}}{x_i - x_{i+1}} = \frac{x - x_2}{-h} = -2x + 2 & x_1 \le x < x_2 \end{cases}$$

$$\varphi_2 = \begin{cases} \frac{x - x_{i-1}}{x_i - x_{i-1}} = \frac{x - x_1}{h} = 2x - 1 & x_1 \le x < x_2 \end{cases}$$

Our problem expressed in its Galerking formulation is:

$$(u'', v) = (f, v), \tag{14}$$

where $f = 1$. We use integration by parts to reduce the order of the derivative of u:

$$\int_0^L u''(x)v(x)dx = -\int_0^L u'(x)v'(x)dx + [vu']_0^L, \tag{15}$$

$$= -\int_0^L u'(x)v'(x)dx + u'(L)v(L) - u'(0)v(0). \tag{16}$$

We have that $u'(L) = 0$, so the middle term vanishes. Since $u(0) = 0$, we must also require $v(0) = 0$. Our variational formulation becomes

$$-(u', v') = (1, v), \tag{17}$$

$$(u', v') = -(1, v). \tag{18}$$

We can then find the element matrix for Ω_0 (φ_1^0 means φ_1 as defined in Ω_o), by the formula:

$$A_{i,j} = \int_0^{0.5} \varphi_i' \varphi_j' dx, \tag{19}$$

and the b-vector by the formula

$$b_i = -\int_0^{0.5} \varphi_i dx. \tag{20}$$

We get:

$$A^{(0)} = \begin{pmatrix} \int_0^{0.5} \varphi_0' \varphi_0' dx & \int_0^{0.5} \varphi_1'^0 \varphi_0' dx \\ \int_0^{0.5} \varphi_0' \varphi_1'^0 dx & \int_0^{0.5} \varphi_1'^0 \varphi_1'^0 dx \end{pmatrix} = \begin{pmatrix} 2 & -2 \\ -2 & 2 \end{pmatrix}$$

By using the formula for the element vector b, we obtain

$$b^{(0)} = \begin{pmatrix} [-x^2 + x]_0^{0.5} \\ [x^2]_0^{0.5} \end{pmatrix} = \begin{pmatrix} -0.25 \\ -0.25 \end{pmatrix}$$

We do the similar thing for element Ω_1:

$$A^{(1)} = \begin{pmatrix} 2 & -2 \\ -2 & 2 \end{pmatrix}$$

$$b^{(1)} = \begin{pmatrix} [-x^2 + 2x]_{0.5}^1 \\ [x^2 - x]_{0.5}^1 \end{pmatrix} = \begin{pmatrix} -0.25 \\ -0.25 \end{pmatrix}$$

We gather the element matrices and element vectors:

$$A = \begin{pmatrix} 2 & -2 & 0 \\ -2 & 4 & -2 \\ 0 & -2 & 2 \end{pmatrix}, \qquad b = \begin{pmatrix} -0.25 \\ -0.5 \\ -0.25 \end{pmatrix}$$

This will now be used to solve the linear system $Ac = b$. We have a boundary condition $u(0) = 0$, and we can solve this in two ways.

The first method is to exclude the unknown at $x = 0$, which do by just removing the top row (and also the first column). The second method is that we modify the linear system with $c_0 = 0$, and therefore we set $b_0 = 0$ and the first row of A to 0 (except the first element, which is set to 1). This way, we ensure that $u(0) = 0$. The result is the same for both methods, and is shown in figure 3.1. My code is in the file `cable_2P1.py`, which is included in the same GitHub repository as this report, and clearly show how I have implemented both methods for dealing with the boundary conditions. To solve the linear system, I have used the `numpy.linalg.solve` function.

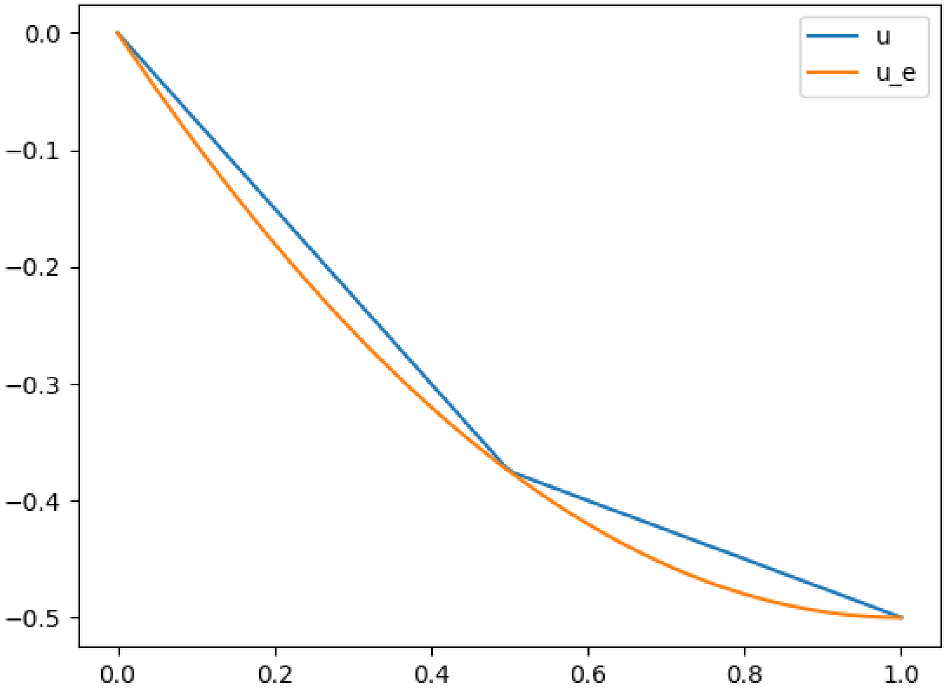

FIGURE 3.1. Approximating u_e by using P1 elements.