Benjamin Hill
Brother Martin
MATH 411
18 November 2023

Reality Check 2: The Euler-Bernoulli Beam

This reality check project explores the displacement of a beam under various loads. The beam in question has a width (w) of 30 cm, a thickness (d) of 3 cm, and a length (L) of 2 m. Other relevant constants include $g = 9.81\frac{m}{s^2}$ (the gravitational constant), 480kg/m (the density of Douglas fir, which this beam is made of), $E = 1.3 * 10^{10}$ Pascals (the Young's modulus of this wood), and $I = wd^3/12$ (the area moment of inertia around the center of mass of the beam). The value h is a length defined as the length of the beam divided by the number of steps taken (n) to approximate the system. Let $f(x)$ be the load applied to the beam, including the weight of the beam itself. With no extra load on the beam, we have that $f(x) = -480wdg$. The vertical displacement of the beam, $y(x)$, satisfies the Euler-Bernoulli equation, $EIy'''' = f(x)$.

When subdivided into n intervals, the Euler-Bernoulli equation yields n equations in n unknowns, which can be written as a square structure matrix of the following form:

$$A = \begin{bmatrix} 16 & -9 & 8/3 & -1/4 & 0 & 0 & \cdots & 0 & 0 \\ -4 & 6 & -4 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 1 & -4 & 6 & -4 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & -4 & 6 & -4 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -4 & 6 & -4 & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 & -4 & 6 & -4 & 1 \\ 0 & 0 & \cdots & 0 & 0 & 16/17 & -60/17 & 72/17 & -28/17 \\ 0 & 0 & \cdots & 0 & 0 & -12/17 & 96/17 & -156/17 & 72/17 \end{bmatrix}$$

The vertical displacements are calculated as solutions to the matrix equation $Ay = \frac{h^4}{EI}f$, where y is the vector composed of all the discrete displacements along the beam and f is the vector of values of $f(x)$ at all the discrete values of x along the beam (note that f is a constant function to start). Now we are ready to begin exploring the displacements $y(x)$ in various situations.

Part 1

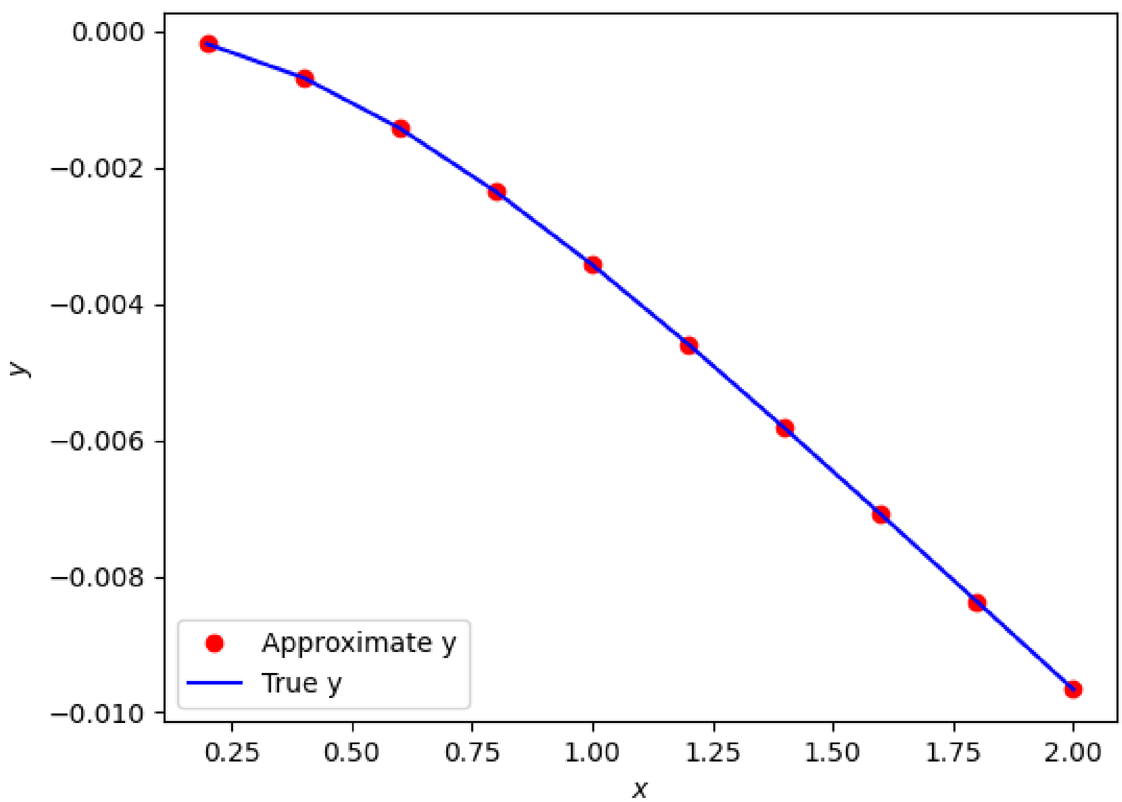
First, we create Python code to define the structure matrix above for an arbitrary size n. For our purposes, it will be most helpful to create this matrix as a sparse matrix, using the SciPy functions for creating such matrices. Then we will solve the system $Ay = \frac{h^4}{EI}f$ for y using $n = 10$, by using the scipy.sparse.linalg.spsolve method in Python. Employing these tools, we find that the solution to this system, rounded to one significant figure for convenience, is

$$y = [-0.0002 \quad -0.0007 \quad -0.001 \quad -0.002 \quad -0.003 \quad -0.005 \quad -0.006 \quad -0.007 \quad -0.008 \quad -0.01]^T.$$

Considering one side of the beam is clamped and cannot move, this solution only gives the displacement of the beam starting just to the side of the clamped end, at 0.2 meters. This means that, given the solution above, the beam bends downward progressively more the farther away a portion of the beam is from the clamped end. This is exactly as expected.

Part 2

Now suppose that the "true solution" of continuous vertical displacements along the beam is given by the function $y(x) = (\frac{f}{24EI})x^2(x^2 - 4Lx + 6L^2)$, where $f(x)$ is the constant function defined above. We will check the error at the end of the beam, at $x = L$ meters, which means we will find the distance between the last entry of the solution vector from part 1 and the function $y(x)$ evaluated at $x = 2$. We compute this error to be approximately $1.6 * 10^{-15}$, which is very near machine roundoff, so this matrix method seems to give a very good approximation of the system. The displacements along the bar are shown in the following graph (generated by matplotlib.pyplot), with the approximations in red and the true solution in blue.



Part 3

Now we will rerun the calculation from Part 1 for $n = 10 * 2^k$, for $k = 1, \dots, 11$, and we will make a table of errors at $x = L$ for each n, which errors are calculated as we did in part 2. Performing all these computations, the errors at $x = L$ are approximately as follows:

k	n	Error at $x = L$	Condition number of A
1	20	$8.49 * 10^{-15}$	$5.58 * 10^5$
2	40	$1.12 * 10^{-13}$	$8.93 * 10^6$
3	80	$1.40 * 10^{-12}$	$1.43 * 10^8$
4	160	$6.97 * 10^{-12}$	$2.29 * 10^9$
5	320	$6.31 * 10^{-11}$	$3.66 * 10^{10}$
6	640	$2.36 * 10^{-9}$	$5.85 * 10^{11}$
7	1280	$6.09 * 10^{-8}$	$9.37 * 10^{12}$
8	2560	$4.44 * 10^{-7}$	$1.50 * 10^{14}$
9	5120	$6.67 * 10^{-8}$	$2.40 * 10^{15}$
10	10240	$3.31 * 10^{-5}$	$3.86 * 10^{16}$
11	20480	$2.39 * 10^{-4}$	$5.74 * 10^{17}$

The error seems to be the smallest for $n = 20$, and it increases with n. This is strange because normally we would expect that the approximation would improve with more subdivisions. In this case, the error increases so significantly because the condition number of A, also included in the above table, increases enormously with n. This explains the unexpected results.

Part 4

Now we will add a sinusoidal pile to the beam, which means that we will add to $f(x)$ a function of the form $s(x) = -pg \sin\frac{\pi}{L}x$, for some value of p. Now, given the "true solution"

$$y(x) = \frac{f}{24EI}x^2(x^2 - 4Lx + 6L^2) - \frac{pgL}{EI\pi}\left(\frac{L^3}{\pi^3}\sin\frac{\pi}{L}x - \frac{x^3}{6} + \frac{L}{2}x^2 - \frac{L^2}{\pi^2}\right),$$

We can confirm that y does satisfy the Euler-Bernoulli equation. It is also the case that $y(0) = y'(0) = y''(L) = y'''(L) = 0$, which means this solution satisfies the clamped-free boundary conditions, as desired. Thus we conclude that $y(x)$ is indeed a solution to the system under a sinusoidal load. For more details on how these computations were done, see the accompanying Python code.

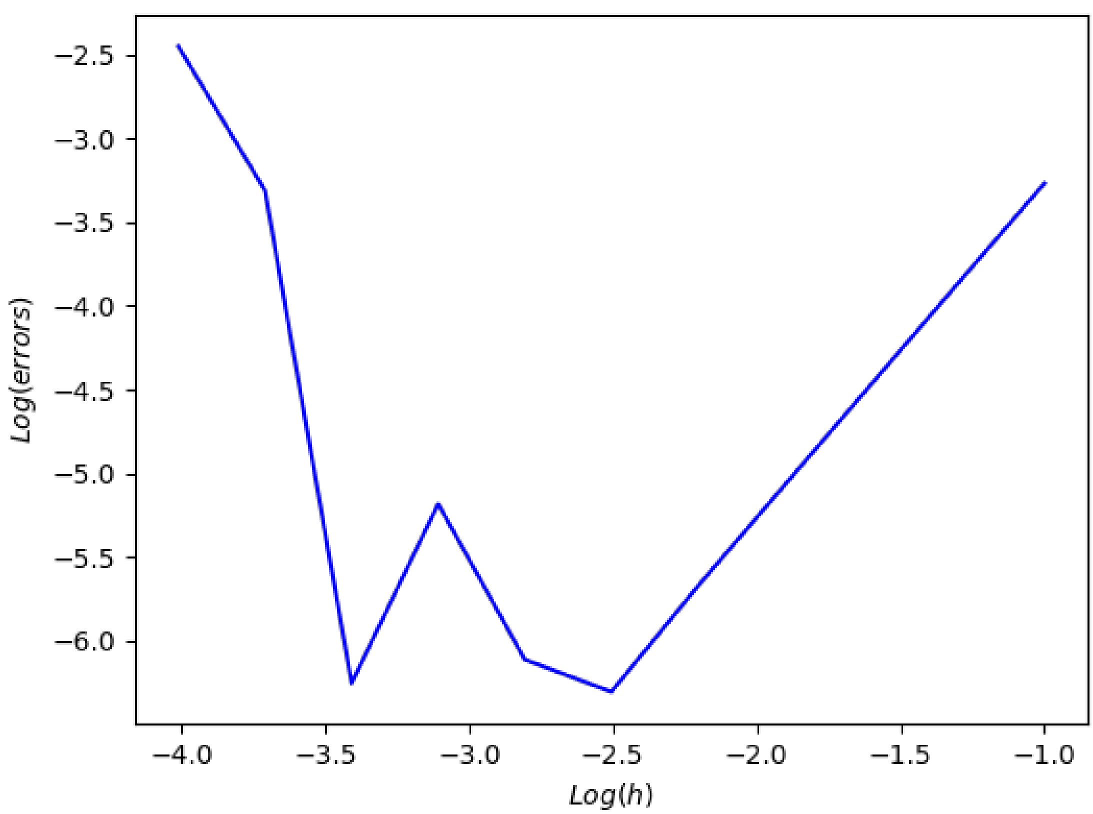
Part 5

Now we will rerun the calculations from Part 3 for the sinusoidal load, with $p = 100\frac{kg}{m}$, and compare our approximations with the true solution above. The table of errors for this system is as follows:

k	n	Error at $x = L$	Condition number of A
1	20	$5.38 * 10^{-4}$	$5.58 * 10^5$
2	40	$1.35 * 10^{-4}$	$8.93 * 10^6$
3	80	$3.39 * 10^{-5}$	$1.43 * 10^8$

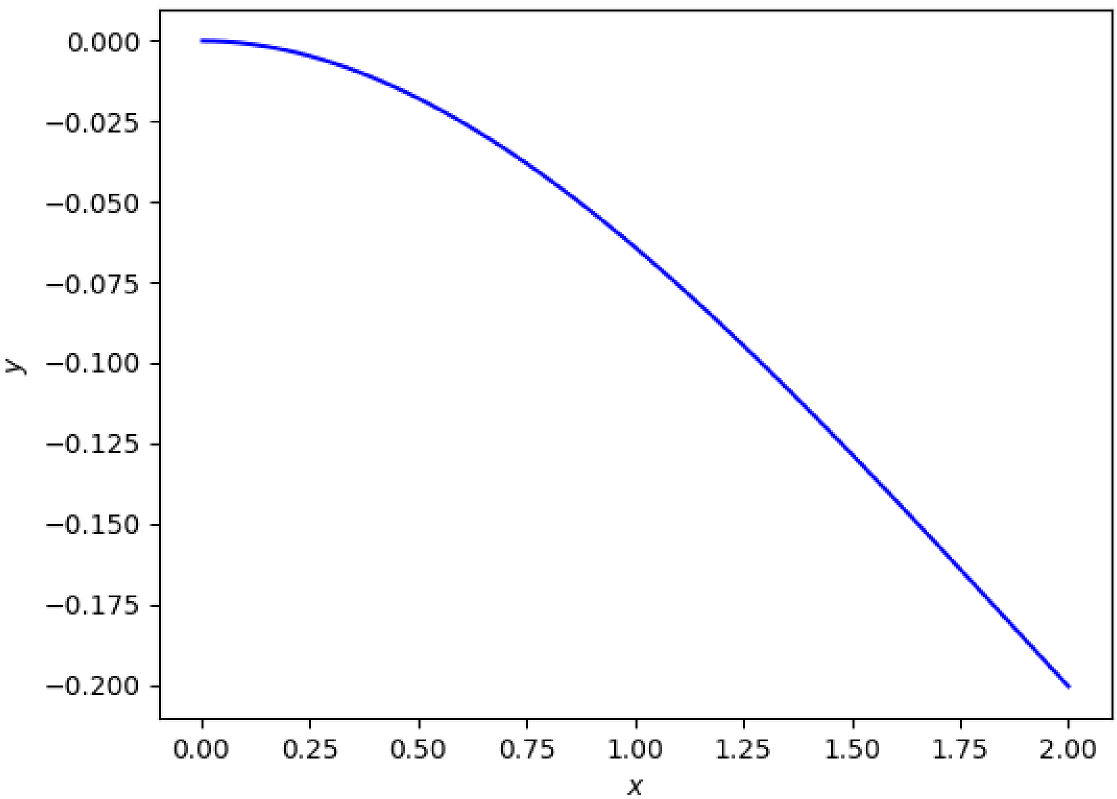
4	160	$8.49 * 10^{-6}$	$2.29 * 10^{9}$
5	320	$2.12 * 10^{-6}$	$3.66 * 10^{10}$
6	640	$4.96 * 10^{-7}$	$5.85 * 10^{11}$
7	1280	$7.72 * 10^{-7}$	$9.37 * 10^{12}$
8	2560	$6.56 * 10^{-6}$	$1.50 * 10^{14}$
9	5120	$5.58 * 10^{-7}$	$2.40 * 10^{15}$
10	10240	$4.87 * 10^{-4}$	$3.86 * 10^{16}$
11	20480	$3.54 * 10^{-3}$	$5.74 * 10^{17}$

Notice the condition numbers here are the same as in Part 3, which means the condition number enormously increases with n, as it did before. This causes the error at $x = L$ to increase after $n = 640$. For $n < 640$, the error is proportional to h^2, which can be seen below on the log-log plot of the errors and h-values. Note that n gets larger as h gets smaller, so n increases from right to left on this graph. The value $n = 640$ is at $\log(h) = -2.5$. We see here that $n = 640$ is a kind of optimal value, at which the error is low and the size of the matrix (and thus the length of the computation) is also low.



Part 6

Now we will remove the sinusoidal load and add a $70kg$ diver to the beam, balancing on the last $20cm$ of the beam. This means that we must add to our original constant function $f(x_i)$ a force per unit length of $-\frac{70kg}{0.2m}$ to all $1.8 \leq x \leq 2$. Then we will approximate the solution again with the optimal value of $n = 640$ which we found in the previous part, where the error at $x = L$ is minimized. The solution is plotted below, and we find the deflection at the free end of the diving board to be $y(L) = -0.2005$, which means the free end of the board is about 20 centimeters lower than the clamped end.



Part 7

Finally, we will fix the end of the beam, making a "clamped-clamped" beam system, obeying identical boundary condition at each end: $y(0) = y'(0) = y(L) = y'(L) = 0$. This version is used to model sag in a structure, like a bridge. The new system of n equation of n unknowns can be represented by the matrix

$$A' = \begin{bmatrix} 16 & -9 & 8/3 & -1/4 & 0 & 0 & \cdots & 0 & 0 \\ -4 & 6 & -4 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 1 & -4 & 6 & -4 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & -4 & 6 & -4 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -4 & 6 & -4 & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 & -4 & 6 & -4 & 1 \\ 0 & 0 & \cdots & 0 & 0 & 1 & -4 & 6 & -4 \\ 0 & 0 & \cdots & 0 & 0 & -1/4 & 8/3 & -9 & 16 \end{bmatrix}.$$

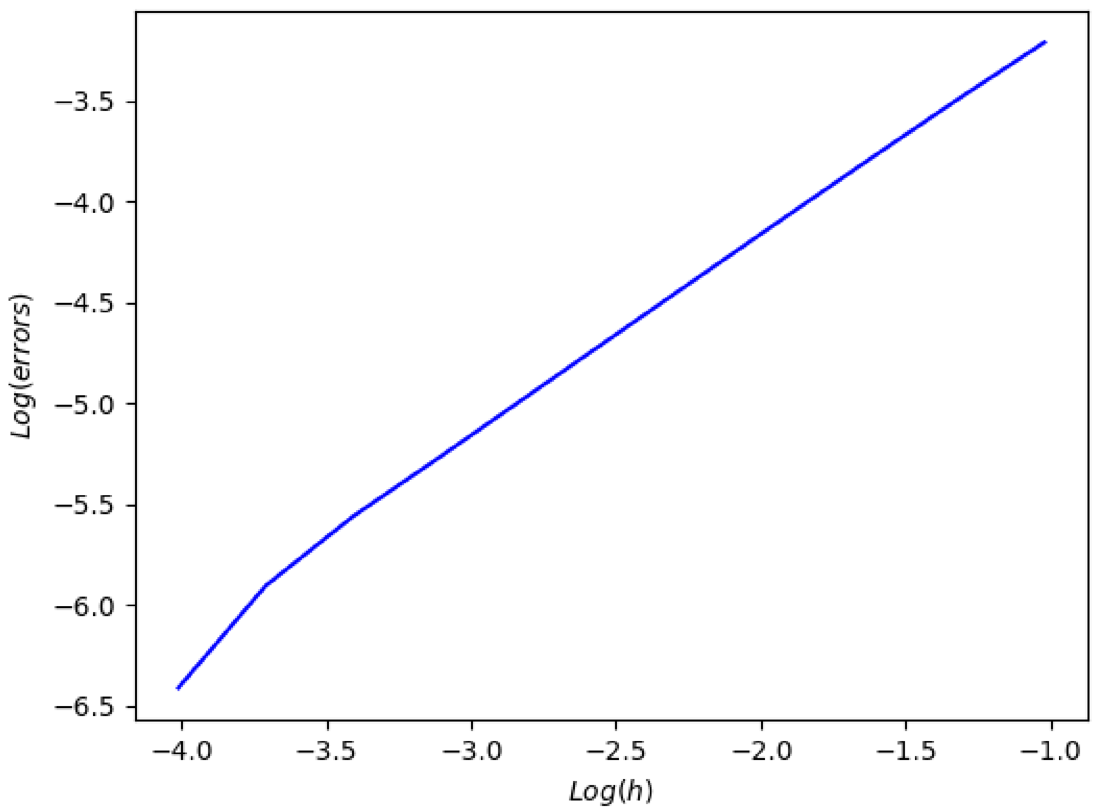
Notice that the bottom-right corner of the matrix is a reflection of the top-left entries, which is what models both ends of the beam being clamped. Also note that the size of the matrix is now $n{+}{=}\ 1$ because both endpoints are included, so each solution will be approximated with one more point than before. We will introduce a sinusoidal load to the beam as before, and we will plot our computed approximations against the true solution, which is given to be

$$y(x) = \frac{f}{24EI}x^2(L-x)^2 - \frac{pgL^2}{EI\pi^4}\left(L^2 \sin\frac{\pi}{L}x + \pi x(x-L)\right).$$

Here is a table of the errors at $x = L/2$ (the middle of the beam) for each n:

k	n	Error at $x = L/2$	Condition number of A
1	21	$6.15 * 10^{-4}$	$1.70 * 10^{4}$
2	41	$3.27 * 10^{-4}$	$2.26 * 10^{5}$
3	81	$1.69 * 10^{-4}$	$3.29 * 10^{6}$
4	161	$8.55 * 10^{-5}$	$5.01 * 10^{7}$
5	321	$4.31 * 10^{-5}$	$7.82 * 10^{8}$
6	641	$2.16 * 10^{-5}$	$1.24 * 10^{10}$
7	1281	$1.08 * 10^{-5}$	$1.96 * 10^{11}$
8	2561	$5.42 * 10^{-6}$	$3.13 * 10^{12}$
9	5121	$2.74 * 10^{-6}$	$5.00 * 10^{13}$
10	10241	$1.24 * 10^{-6}$	$8.00 * 10^{14}$
11	20481	$3.88 * 10^{-7}$	$1.28 * 10^{16}$

The errors in the approximation at $x = L/2$ decrease with n in this case, even though the condition number of A still rises enormously with n. This situation seems to make the errors proportional to h instead of h^2. See the log-log plot of the errors and h-values below. The approximation only improves with larger n, which is finally what we would expect. The high condition numbers of A do not have so much effect in this case.


−3.5
−4.0
−4.5
−5.0
−5.5
−6.0
−6.5
Log(errors)
−4.0
−3.5
−3.0
−2.5
−2.0
−1.5
−1.0
Log(h)