TopEdu

PROJECT 9 INSTRUCTIONS

When the textbook beam formula breaks

Every engineering student learns this formula. Find out exactly when it stops being true.

START HERE

Set up in ten minutes

This project comes with one short Python file, `09_cantilever_beam.py`. It solves the project's equation with simple methods you can read line by line, checks itself against an answer that is known exactly, and ends with a list of first experiments. The research is what you do after that.

1. **Install Python 3** from python.org (on a Mac it may already be there: type `python3 --version` in Terminal).
2. **Download the starter** from the project's page, or all ten as a zip from topedu.club/topedu/research/starter-code/.
3. **Install the two libraries** in Terminal (Mac) or PowerShell (Windows):

   ```
   python3 -m pip install numpy matplotlib
   ```

   On Windows, type `py` where this guide says `python3`.
4. **Run it:**

   ```
   python3 09_cantilever_beam.py --show
   ```

   The CHECK line must say PASS. `--show` opens the plots; without it they are saved in the `output` folder.

No computer of your own? Google Colab works in a browser for free: open a new notebook, upload the file with the folder icon on the left, and run `!python 09_cantilever_beam.py` in a cell.

The rules that make it research

- **Predict before you run.** Write your guess, and why, in your log first. A wrong prediction you can explain is worth more than a right one you cannot.
- **Keep a log.** Date, what you changed, what you expected, what happened. Your paper is written from it.
- **Check the grid.** Run your key result again with twice the points. If the answer moves, report by how much.
- **Never edit the CHECK to make it pass.** A FAIL after your change is information: undo, confirm PASS, then change one thing at a time.
- **Own every line.** Your mentor explains; you write and run. Name everyone and everything that helped, including this starter.

The full guide, with all ten projects and five numerical ideas the starters use, is at topedu.club/topedu/research/starter-code/.

PROJECT 9 · CIVIL AND MECHANICAL ENGINEERING

When the textbook beam formula breaks

Every engineering student learns this formula. Find out exactly when it stops being true.

FILE	MATH	YOU NEED
`09_cantilever_beam.py`	Algebra II	Rulers, coins and a clamp

The question

The classic beam formula is taught as fact, but it assumes a long, slender beam. How short can a beam be before the formula is wrong by 5%, and does the material's Poisson ratio change that?

Objectives

Weeks 1 to 3	**Learn to trust the model.** Run 09_cantilever_beam.py until its CHECK line says PASS, and explain every line of it to your mentor. Then reproduce a known answer: A slender beam's tip sag should approach $3\rho g L^4 / (2Eh^2)$. Our test runs give 0.996 of that at length/depth 14 and 1.073 at 3. Checkpoint: you reproduce a known answer and report how close you got.
Weeks 4 and 5	**Ask your own question.** Narrow the question above to one sentence that could turn out to be wrong, and write your prediction before you run anything. The first experiments below are a place to start. Checkpoint: a one-page plan with your question and prediction.
Weeks 6 to 9	**Run it, then measure the world.** Simulate a square-section beam fixed at one end, sagging under its own weight, while shortening it step by step, and map the error of the formula. Then step away from the screen: clamp rulers or strips at different lengths to a desk, load them with coins, and measure the sag. Checkpoint: one chart comparing your model with reality.
Weeks 10 to 12	**Explain the gap, and share it.** Work out why your model and your measurements disagree, test at least one explanation, and write it up yourself. Say which numbers came from the starter and which from the lab. Keep the honest limit in view: the clamped end has a stress singularity, so results near it depend on the grid. Report grid sensitivity; this is a study and not a design tool. Checkpoint: a final paper, your code and a ten-minute talk.

The model in the starter

A beam fixed at one end sags under its own weight. The textbook (Euler-Bernoulli) answer for the tip sag is w = q L^4 / (8 E I) with q = weight per length, E the stiffness of the material, I = h^4 / 12 for a square. It assumes a long, slender beam. A short beam also bends by shear, which the formula leaves out. Timoshenko's theory adds it: w = q L^4 / (8 E I) + q L^2 / (2 kappa G A), with G = E / (2 (1 + nu)) and kappa = 5/6.

The full 3D answer needs the online lab (elasticity in 3D is too heavy for a starter). This file does the parts you can do on a laptop:

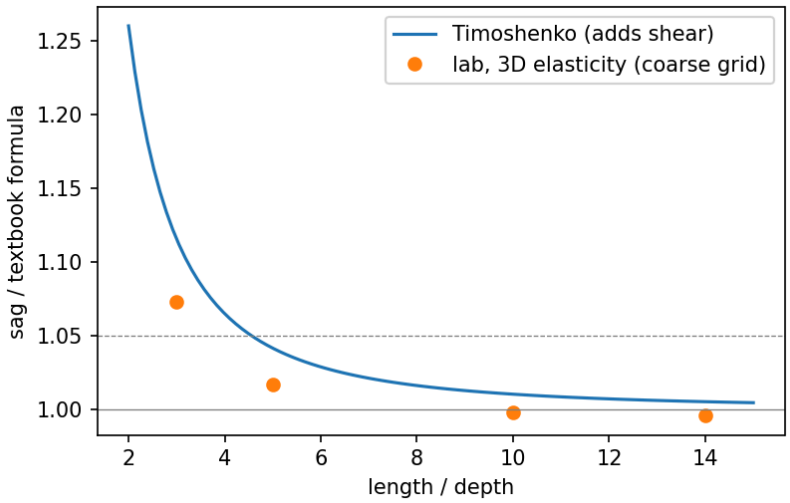
What the starter does

- Stage 1: solve the beam equation E I w"" = q numerically and check it against the formula. CHECK should say PASS.
- Then it compares both theories with the lab's 3D results that we measured when we tested this project.
- Stage 2: measure real rulers, fit the L^4 law, and find where the formula stops working.

What you should see

Run `python3 09_cantilever_beam.py`. This is the output from the run that built this guide. The numbers that depend on your computer's speed may differ; the CHECK line must say PASS.

```
beam equation: tip sag 0.12500, formula 0.12500, error 0.003%
CHECK numerical beam against q L^4 / (8 E I) -> PASS (limit 1%)
tip sag / textbook formula, by length / depth:
  L/h   Timoshenko   lab 3D (our test)
   14   1.005        0.996
   10   1.010        0.998
    5   1.042        1.017
    3   1.116        1.073
Timoshenko says the textbook formula is 5% low once L/h < 4.6; does the 3D solver agree?
```



Your first experiments

From the YOUR TURN list at the bottom of the file. Write your prediction in your log *before* you run each one.

1. Rulers: clamp a ruler to a desk with 10, 15, 20, 25 and 30 cm hanging over, measure the tip sag, and save a CSV with columns length_cm,sag_mm. Run: python 09_cantilever_beam.py my_rulers.csv Predict the power first. Coins at the tip change it: why?
2. From your fit, work out E for the ruler (weigh it to get q). Compare with a published value for the material.

3. In the online lab, rerun the 3D beam at L/h = 3 and 5 with a finer grid. How much do the numbers above move? The clamped end has a stress singularity, so report how your answer depends on the grid.
4. Change nu to 0.05 and 0.4. Does the 5% threshold move? Check one value in the lab.

The data

Settings in the online lab

Setting	Range	Starts at
Length / depth	2 to 14 ratio	10
Poisson ratio	0.05 to 0.4	0.3

Checked on the research solver

Solver (Linear elasticity, 3D)	L/h 14: sag/formula 0.996 (1066 iterations, 47.2 s); L/h 10: sag/formula 0.998 (737 iterations, 22.7 s); L/h 5: sag/formula 1.017 (433 iterations, 8.8 s); L/h 3: sag/formula 1.073 (313 iterations, 5.5 s)
Known answer	with shear 1.005; with shear 1.010; with shear 1.042; with shear 1.116
Error	3.77% (limit 8.00%): passed
A first finding	the formula is close for long beams and increasingly wrong for short ones, which is the project's question

The real world

Clamp rulers or strips at different lengths to a desk, load them with coins, and measure the sag.

The honest limit

The clamped end has a stress singularity, so results near it depend on the grid. Report grid sensitivity; this is a study and not a design tool.

How the full project checks itself: A slender beam's tip sag should approach $3\rho g L^4 / (2Eh^2)$. Our test runs give 0.996 of that at length/depth 14 and 1.073 at 3. The online lab runs the same project on the research solver (Linear elasticity, 3D).

REFERENCE

When something goes wrong

You see	Try this
`No module named numpy`	Run the install line again with the same Python you use to run the file: `python3 -m pip install -r requirements.txt`
Numbers become nan or huge	A time step is past its speed limit, usually after you made the grid finer. Each solver works out its step from the grid; check you did not fix it by hand.
`did not settle`	An iterative solver needs more sweeps for a finer grid. Raise its limit, or keep the grid smaller while you explore.
A CHECK line says FAIL after your change	Good: your change moved the model away from the case with a known answer. Undo it, confirm PASS, then change one thing at a time.
No window opens	Add `--show`, or open the PNG files in the `output` folder.
A recording gives strange notes	Record in a quiet room, save as .wav (not .m4a), trim silence at the start, and set the note you expect in the file.

How the starters relate to the online lab

The online lab, for members, runs each project on the research solver (AutoSBPSAT): high-order SBP-SAT methods, the full 3D elasticity for project 9, and frames of a time series. The starters use plain second-order methods instead, so every line is short enough to read. Both are checked against the same kind of exact answers. Use the starter to understand and to explore quickly; use the lab for results you will quote, and say in your paper which one produced each number.

How these starters were checked

Every number in "What you should see" came from running the starter while this guide was built, on Python 3.9.23, numpy 1.24.4, matplotlib 3.5.2. `check_all.py` also feeds each optional data reader (the soil-station CSV, the ruler CSV and two .wav recordings) a made-up file with a known answer, and confirms the answer comes back. They have not yet been used by a student. The twelve-week plan and the time estimates in it are a design, to be revised after the first group: tell your mentor what was unclear.