TopEdu

PROJECT 5 INSTRUCTIONS

The missing notes of a closed pipe

Tap a pipe and it plays a note. Predict that note, then find what the textbook gets wrong.

START HERE

Set up in ten minutes

This project comes with one short Python file, `05_pipe_resonance.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 05_pipe_resonance.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 05_pipe_resonance.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 5 · ACOUSTICS AND MUSIC

The missing notes of a closed pipe

Tap a pipe and it plays a note. Predict that note, then find what the textbook gets wrong.

FILE	MATH	YOU NEED
`05_pipe_resonance.py`	Algebra II, trigonometry	A phone and about $10 of PVC pipe

The question

A pipe closed at one end should play only odd harmonics. How closely does a real pipe follow the model, and what is the size of the 'end correction' the ideal model leaves out as the pipe gets wider?

Objectives

Weeks 1 to 3	**Learn to trust the model.** Run 05_pipe_resonance.py until its CHECK line says PASS, and explain every line of it to your mentor. Then reproduce a known answer: Pipe modes are $f_n = (2n - 1)c / 4L$, with no even harmonics. 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 pulse in the pipe, extract its frequencies, and compare them with recordings of PVC pipes of several lengths and diameters. Then step away from the screen: tap or blow across PVC pipes and record them with a free phone spectrum app. 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 model has a perfect open end. The gap between the model and your recording is the discovery, not a failure. Checkpoint: a final paper, your code and a ten-minute talk.

The model in the starter

Sound in a pipe closed at one end and open at the other obeys the wave equation. The closed end reflects pressure as it is; the open end (ideally) forces pressure to zero. So the pipe only fits odd quarter-waves, and its notes are f_n = (2n - 1) c / 4L: 1, 3, 5, … times the lowest note, with no even harmonics.

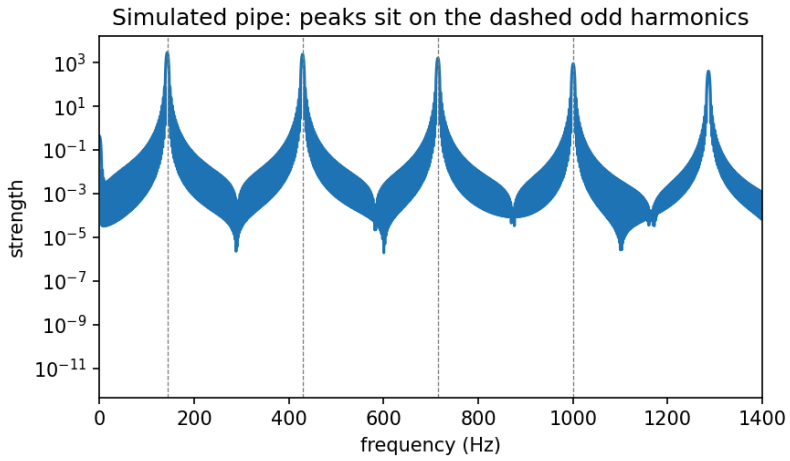
What the starter does

- Stage 1: run this file. It starts a pressure pulse in a 0.6 m pipe, listens at the closed end, finds the notes with a Fourier transform, and checks the first two against the formula. The CHECK line should say PASS.
- Stage 2: record real PVC pipes, find their notes with the same code, and measure what the ideal model leaves out.

What you should see

Run `python3 05_pipe_resonance.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.

```
note 1: simulated  142.92 Hz, formula  142.92 Hz, error 0.000%
note 2: simulated  428.75 Hz, formula  428.75 Hz, error 0.001%
ratio of note 2 to note 1: 3.000 (the ideal pipe says 3: no even harmonic)
CHECK first two notes against (2n - 1) c / 4L -> PASS (limit 1%)
```



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. Predict, then record: a 0.6 m PVC pipe closed at one end (a palm or a cap). Is its lowest note above or below the formula? Run: python 05_pipe_resonance.py my_pipe.wav
2. End correction: a real open end acts as if the pipe were longer, by about delta. From each recording, delta = c / (4 f_1) - L. Measure it for pipes of three or more widths. How does delta depend on the radius? (Textbooks quote about 0.6 times the radius. Do your pipes agree?)
3. Look for even harmonics in your recordings. If any appear, what could make them?
4. The speed of sound changes with temperature: c is about 331 + 0.6 T(C) m/s. Note the room temperature.

The data

Settings in the online lab

Setting	Range	Starts at
Pipe length	0.2 to 2 m	0.6
Speed of sound	320 to 360 m/s	343

Checked on the research solver

Solver (Wave, 1D)	mode 1: 142.90 Hz; mode 2: 428.43 Hz
Known answer	142.92 Hz; 428.75 Hz
Error	0.07% (limit 1.00%): passed
A first finding	the second mode sits at 2.998 times the first: 3, with no even harmonic

The real world

Tap or blow across PVC pipes and record them with a free phone spectrum app.

The honest limit

The model has a perfect open end. The gap between the model and your recording is the discovery, not a failure.

How the full project checks itself: Pipe modes are $f_n = (2n - 1)c / 4L$, with no even harmonics. The online lab runs the same project on the research solver (Wave, 1D).

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.