TopEdu

PROJECT 2 INSTRUCTIONS

How deep does summer go?

Dig down far enough and it is winter in summer. Measure how far, using real climate data.

START HERE

Set up in ten minutes

This project comes with one short Python file, `02_soil_temperature.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 02_soil_temperature.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 02_soil_temperature.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 2 · EARTH SCIENCE AND CLIMATE

How deep does summer go?

Dig down far enough and it is winter in summer. Measure how far, using real climate data.

FILE	MATH	YOU NEED
`02_soil_temperature.py`	Algebra II, trigonometry, logarithms	A laptop and free public data

The question

Seasonal temperature swings fade and arrive later as you go underground. From real sensor data, what is your soil's thermal diffusivity, and how does it change with wetness or with the site?

Objectives

Weeks 1 to 3	**Learn to trust the model.** Run 02_soil_temperature.py until its CHECK line says PASS, and explain every line of it to your mentor. Then reproduce a known answer: The model has an exact answer: amplitude falls as e^(-z/d) and the delay grows as z/d, with d = $\sqrt{(2\kappa/\omega)}$. 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.** Drive the surface with a yearly temperature cycle, measure the amplitude and the delay at each depth, and fit the diffusivity to data from a real station. Then step away from the screen: download free soil temperatures at 5 to 100 cm from NOAA's U.S. Climate Reference Network and compare stations with different soils. 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: soil is one uniform layer here, so a mismatch at some depths is itself a finding to explain: layers or moisture may be the cause. Checkpoint: a final paper, your code and a ten-minute talk.

The model in the starter

The ground's surface warms and cools once a year. Underground the swing gets smaller and arrives later. How much smaller, and how much later, tells you the soil's thermal diffusivity (kappa).

Units: depth in metres, time in years, temperature in degrees C away from the yearly average. Run: python 02_soil_temperature.py [--show] or python 02_soil_temperature.py my_station.csv Needs: numpy and matplotlib

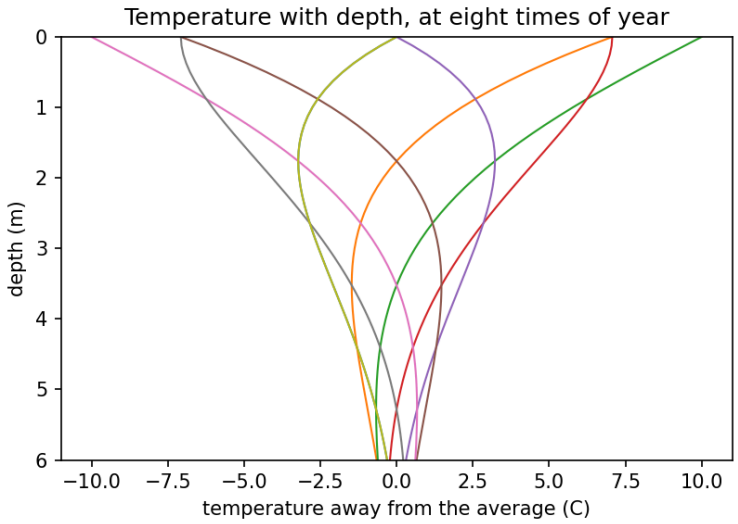
What the starter does

- Stage 1: run this file. It steps the heat equation through one year in a 20 m column of soil, driven by a yearly cycle at the surface, and compares the whole field with the exact answer. The CHECK line should say PASS.
- Stage 2: fit kappa from temperatures at several depths, first from the model, then from a real station.

What you should see

Run `python3 02_soil_temperature.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.

```
largest error over the year: 1.97e-03 degrees C (0.020% of the swing)
CHECK whole year against the exact answer -> PASS (limit 1% of the swing)
  depth 0.05 m: swing  9.78 C,   1.3 days behind the surface
  depth 0.10 m: swing  9.56 C,   2.6 days behind the surface
  depth 0.20 m: swing  9.15 C,   5.1 days behind the surface
  depth 0.50 m: swing  8.00 C,  13.0 days behind the surface
  depth 1.00 m: swing  6.40 C,  25.9 days behind the surface
kappa from shrinking: 15.81   kappa from lag: 15.73   (the model used 15.8)
damping depth d = 2.24 m: below about 3d = 6.7 m the year is almost invisible
```



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 first: if the soil were twice as diffusive (wet), how much deeper would the 1 C swing reach? Then run with KAPPA = 31.6 and check.
2. Download two years of soil temperatures from a USCRN station, put them in the simple CSV layout described in read_station, and run: python 02_soil_temperature.py my_station.csv
3. Compare two stations with different soils or rainfall. Do "kappa from shrinking" and "kappa from lag" agree? If not, why might a real soil break this one-layer model?

The data

Settings in the online lab

Setting	Range	Starts at
Thermal diffusivity	5 to 40 m²/year	15.8
Surface swing	2 to 25 °C	10

Checked on the research solver

Solver (Heat, 1D)	largest error against the exact solution over a year: 1.44e-3 °C; swing at depth 2.24 m reaches 3.58 °C over 8 frames
Known answer	amplitude 10 °C at the surface, 3.68 °C at one damping depth
Error	0.01% (limit 1.00%): passed

The real world

Download free soil temperatures at 5 to 100 cm from NOAA's U.S. Climate Reference Network and compare stations with different soils.

The honest limit

Soil is one uniform layer here, so a mismatch at some depths is itself a finding to explain: layers or moisture may be the cause.

How the full project checks itself: The model has an exact answer: amplitude falls as e^(−z/d) and the delay grows as z/d, with d = √(2κ/ω). The online lab runs the same project on the research solver (Heat, 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.