TopEdu

PROJECT 1 INSTRUCTIONS

Where should the hot chip go?

Your phone throttles when it gets hot. Find out why where the chip sits matters so much.

START HERE

Set up in ten minutes

This project comes with one short Python file, `01_chip_cooling.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 01_chip_cooling.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 01_chip_cooling.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 1 · HEAT AND ENGINEERING

Where should the hot chip go?

Your phone throttles when it gets hot. Find out why where the chip sits matters so much.

FILE	MATH	YOU NEED
`01_chip_cooling.py`	Algebra II, exponentials; Fourier series are met along the way	A laptop. Optional: a small metal plate and low-cost thermistors, with an adult

The question

A hot chip sits on a metal plate cooled along one edge. How much does moving the chip, or spreading its heat, change the peak temperature, and can you predict the best layout for two chips?

Objectives

Weeks 1 to 3	**Learn to trust the model.** Run 01_chip_cooling.py until its CHECK line says PASS, and explain every line of it to your mentor. Then reproduce a known answer: Compare the first steady-state runs with the series solution you derive for the plate. 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.** Sweep the chip position and size, record the peak temperature, then add a second heat source and search for the coolest layout. Then step away from the screen: heat a small aluminium plate with a power resistor, cool one edge in ice water, and read temperatures with low-cost thermistors. 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: material properties are uniform, so the model cannot represent a plate made of two metals. Checkpoint: a final paper, your code and a ten-minute talk.

The model in the starter

A chip heats a square metal plate. The bottom edge (y = 0) is held cold, the other three edges are insulated. How hot does the plate get, and how much does the chip's position change that?

Units are scaled: the plate is 1 by 1, and temperature is "degrees above the cold edge". Run: python 01_chip_cooling.py (add --show to open the plots in a window) Needs: Python 3 with numpy and matplotlib (pip install numpy matplotlib)

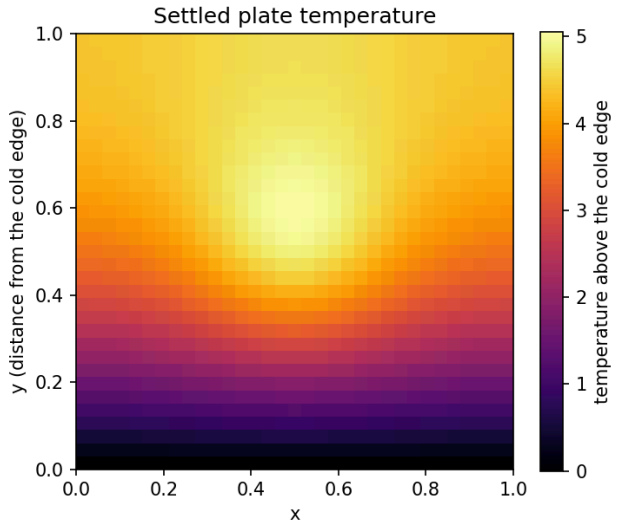
What the starter does

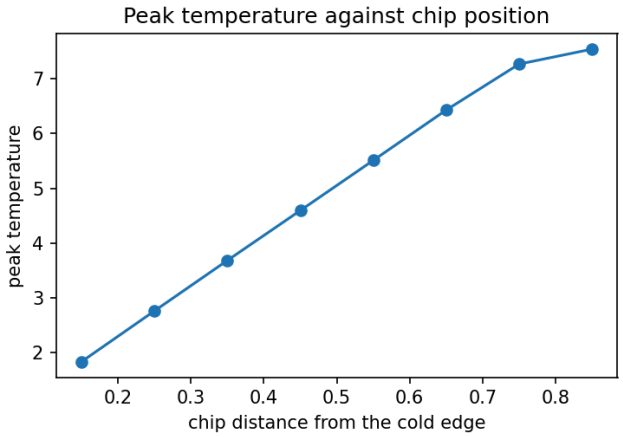
- Stage 1 (weeks 1 to 3): run this file. It solves the heat equation step by step until the plate settles, then checks the peak temperature against an exact series answer. You should see a CHECK line that says PASS.
- Stage 2 (weeks 4 to 9): change the chip, then answer your own question in the YOUR TURN section at the bottom.

What you should see

Run `python3 01_chip_cooling.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.

```
settled after 32248 steps; peak 5.0544 at x = 0.50, y = 0.59
series answer at that point: 5.0495
CHECK peak temperature against the series: error 0.10% -> PASS (limit 1%)
  chip at distance 0.15 from the cold edge: peak 1.837
  chip at distance 0.25 from the cold edge: peak 2.763
  chip at distance 0.35 from the cold edge: peak 3.684
  chip at distance 0.45 from the cold edge: peak 4.595
  chip at distance 0.55 from the cold edge: peak 5.510
  chip at distance 0.65 from the cold edge: peak 6.428
  chip at distance 0.75 from the cold edge: peak 7.267
  chip at distance 0.85 from the cold edge: peak 7.540
```





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. Write your prediction first: if the chip size doubles with the same total power, does the peak halve? (Hint: to keep the total power fixed, scale POWER by (old size / new size)^2.) Then run it.
2. Add a second chip: heat_source(x, y, 0.3, 0.4) + heat_source(x, y, 0.7, 0.6). Search for the two positions with the lowest peak. Is the best layout symmetric? Is it what you guessed?

3. Grid check: run with N = 17, 33 and 65. How much does the peak change? Report that in your paper.
4. Real world: heat an aluminium plate with a power resistor and cool one edge in ice water. Which of your predictions survive? The gap is the interesting part.

The data

Settings in the online lab

Setting	Range	Starts at
Chip position, x	0.15 to 0.85 plate widths	0.5
Distance from the cooled edge	0.15 to 0.85 plate widths	0.5
Chip size	0.06 to 0.25 plate widths	0.12

Checked on the research solver

Solver (Heat, 2D)	peak temperature 7.3284 at (0.25, 0.8125)
Known answer	series solution 7.3271
Error	0.02% (limit 1.00%): passed
A first finding	moving the chip from 0.7 to 0.2 plate widths from the cooled edge changes the peak from 7.33 to 2.44

The real world

Heat a small aluminium plate with a power resistor, cool one edge in ice water, and read temperatures with low-cost thermistors.

The honest limit

Material properties are uniform, so the model cannot represent a plate made of two metals.

How the full project checks itself: Compare the first steady-state runs with the series solution you derive for the plate. The online lab runs the same project on the research solver (Heat, 2D).

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.