TopEdu

PROJECT 3 INSTRUCTIONS

How small can a nature reserve be?

Why do some species vanish from parks that look big enough? The answer fits in one equation.

START HERE

Set up in ten minutes

This project comes with one short Python file, `03_critical_patch.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 03_critical_patch.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 03_critical_patch.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 3 · MATHEMATICAL BIOLOGY

How small can a nature reserve be?

Why do some species vanish from parks that look big enough? The answer fits in one equation.

FILE	MATH	YOU NEED
`03_critical_patch.py`	Algebra II, exponentials, one guided substitution	A laptop and published field studies

The question

A population grows inside a habitat and is lost when it wanders out. For a fixed area, which rectangle keeps a species alive, and how does the minimum size depend on how far the species roams?

Objectives

Weeks 1 to 3	**Learn to trust the model.** Run 03_critical_patch.py until its CHECK line says PASS, and explain every line of it to your mentor. Then reproduce a known answer: For an a by b rectangle the decay rate must match $D\pi^2(1/a^2 + 1/b^2)$, which gives the classic critical patch size. 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.** Write density as u = e^(rt) · v, so v obeys the heat equation. Measure how fast v decays in each shape: the species survives when r beats that decay rate. Then step away from the screen: put measured dispersal and growth rates from published field studies into the model, then compare with the size of a real reserve. 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: growth is exponential with no crowding, so this predicts whether a species survives, not how large its population becomes. Checkpoint: a final paper, your code and a ten-minute talk.

The model in the starter

Animals in a rectangular reserve multiply at rate r and wander at rate D. Any that cross the edge are lost. Write the density as u = e^(r t) v. Then v obeys the plain heat equation dv/dt = D laplacian(v) with v = 0 on the edge, and v dies away at some rate lambda. The species survives exactly when r is bigger than lambda.

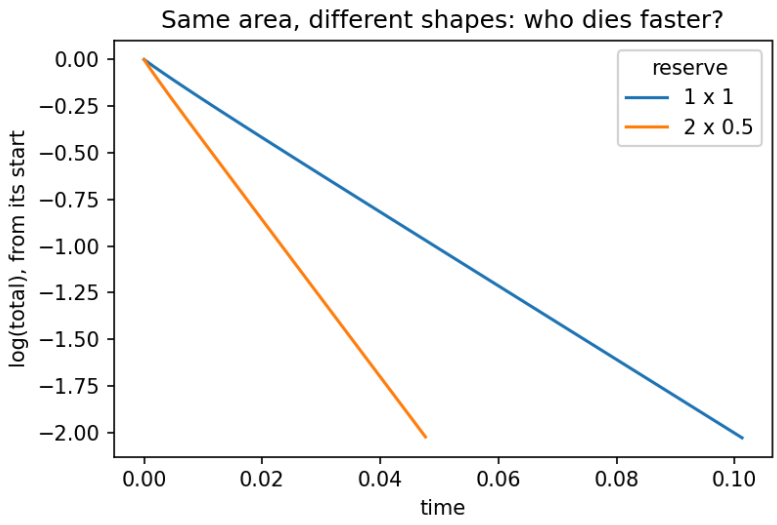
What the starter does

- Stage 1: run this file. It measures lambda for a 1 x 1 and a 2 x 0.5 reserve (the same area) and checks both against the exact rate D pi^2 (1/a^2 + 1/b^2). The CHECK line should say PASS.
- Stage 2: use real dispersal and growth rates and ask which shapes of a real reserve would keep a species.

What you should see

Run `python3 03_critical_patch.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.

```
1 x 1 reserve: measured decay rate 19.749, exact 19.739, error 0.05%
   so the species needs a growth rate r > 19.75 to survive here
2 x 0.5 reserve: measured decay rate 42.084, exact 41.946, error 0.33%
   so the species needs a growth rate r > 42.08 to survive here
CHECK decay rates against D pi^2 (1/a^2 + 1/b^2) -> PASS (limit 1%)
  area 1, length/width 1: needs r > 19.74  (1.00 times the square)
  area 1, length/width 1.5: needs r > 21.38  (1.08 times the square)
  area 1, length/width 2: needs r > 24.67  (1.25 times the square)
  area 1, length/width 3: needs r > 32.90  (1.67 times the square)
  area 1, length/width 4: needs r > 41.95  (2.12 times the square)
  area 1, length/width 6: needs r > 60.86  (3.08 times the square)
```



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. The list above uses the formula. Check two of its shapes with decay_rate() before you trust it.
2. A real species: find published values of D (km^2 per year) and r (per year) for one animal or plant. What is the smallest square reserve that keeps it? Compare with the size of a real reserve.
3. Predict, then test: does an L-shaped reserve do better or worse than a square of the same area? (Hint: set v to zero on a corner block every step to cut the L out of a rectangle.)

4. Say what the model leaves out: here nothing limits growth, so it tells you "survives or not", not "how many".

The data

Settings in the online lab

Setting	Range	Starts at
Reserve width	0.5 to 4 length units	1
Reserve height	0.5 to 4 length units	1
Roaming rate D	0.2 to 4 area per time	1

Checked on the research solver

Solver (Heat, 2D (with a substitution))	1 x 1: decay rate 19.762; 2 x 0.5: decay rate 41.717
Known answer	19.739; 41.946
Error	0.54% (limit 1.00%): passed
A first finding	same area, but the long thin reserve needs a growth rate 2.1 times higher to survive

The real world

Put measured dispersal and growth rates from published field studies into the model, then compare with the size of a real reserve.

The honest limit

Growth is exponential with no crowding, so this predicts whether a species survives, not how large its population becomes.

How the full project checks itself: For an a by b rectangle the decay rate must match $D\pi^2(1/a^2 + 1/b^2)$, which gives the classic critical patch size. The online lab runs the same project on the research solver (Heat, 2D (with a substitution)).

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.