TopEdu

PROJECT 4 INSTRUCTIONS

# How thick can lab-grown tissue be?

Lab-grown organs have a size limit, and it comes from physics. Work out what it is.

START HERE

# Set up in ten minutes

This project comes with one short Python file, `04_tissue_oxygen.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 04_tissue_oxygen.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 04_tissue_oxygen.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 4 · BIOPHYSICS AND MEDICINE

# How thick can lab-grown tissue be?

*Lab-grown organs have a size limit, and it comes from physics. Work out what it is.*

| FILE | MATH | YOU NEED |
|---|---|---|
| `04_tissue_oxygen.py` | Algebra II, exponentials; cosh is introduced in the project | A laptop and published research |

## The question

Oxygen diffuses into tissue while cells consume it, which limits how large a tissue grown without blood vessels can be. What is the largest cube whose centre still gets enough oxygen, and do shape and surface area matter?

## Objectives

| | |
|---|---|
| **Weeks 1 to 3** | **Learn to trust the model.** Run 04_tissue_oxygen.py until its CHECK line says PASS, and explain every line of it to your mentor. Then reproduce a known answer: A slab has the exact answer c = cosh((x-a)/L) / cosh(a/L), with penetration length L = √(D/k).<br>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.<br>Checkpoint: a one-page plan with your question and prediction. |
| **Weeks 6 to 9** | **Run it, then measure the world.** Find the side length at which the centre falls to a chosen oxygen threshold; repeat for flat slabs and long bars of the same volume. Then step away from the screen: compare the penetration length with published oxygen diffusion limits and with the sizes of real organoids and spheroids.<br>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: consumption here is proportional to oxygen. Real cells keep consuming at a nearly fixed rate until oxygen is low, so say how that would change your estimate.<br>Checkpoint: a final paper, your code and a ten-minute talk. |

## The model in the starter

Oxygen soaks into tissue from its surface while the cells use it up. With no blood vessels, the centre of a large piece starves. In scaled units the oxygen c obeys -laplacian(c) + c = 0 inside, with c = 1 on the surface. Lengths are measured in "penetration lengths" L = sqrt(D / k): D is how fast oxygen spreads, k how fast it is used.

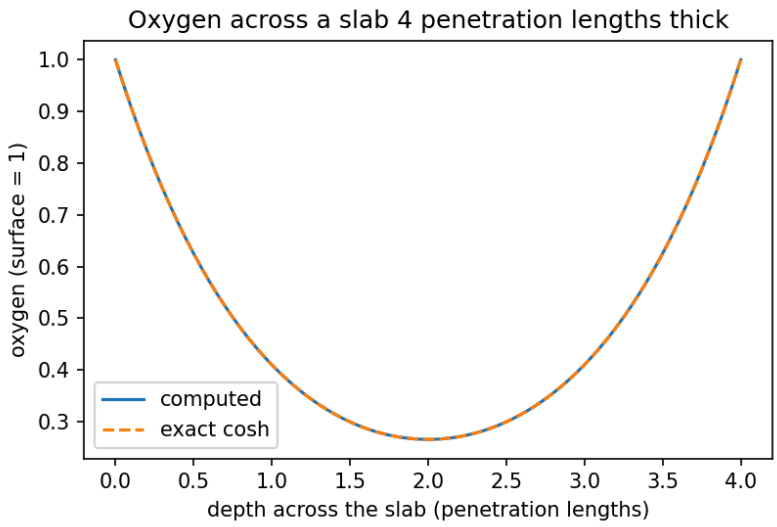
## What the starter does

- Stage 1: run this file. It solves a flat slab (exact answer: cosh) and a cube (exact answer: a series), and checks both. The CHECK line should say PASS.
- Stage 2: find the largest cube, slab and bar whose centre stays above an oxygen threshold you choose.

## What you should see

Run `python3 04_tissue_oxygen.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.

```
slab 4 thick: centre 0.2659, exact 0.2658, worst error 5.3e-05
cube side 1: centre oxygen 0.9462, series 0.9459, error 0.03%
cube side 2: centre oxygen 0.8073, series 0.8061, error 0.15%
cube side 4: centre oxygen 0.4630, series 0.4596, error 0.73%
CHECK slab and cubes against exact answers -> 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. Pick a threshold, say the centre must keep 20% of the surface oxygen. Predict the largest cube side, then find it by trying sides between 2 and 8 with box_centre_series (fast) and checking one with box().
2. Same volume, other shapes: a slab 0.5 thick and a long square bar (make one side 10). Which keeps its centre alive best? Is it the one with the most surface?
3. Turn units back into micrometres: look up D for oxygen in tissue and a cell consumption rate, compute L = sqrt(D / k), and compare your limit with published organoid sizes (often a few hundred micrometres).

4. Honest limit: real cells use oxygen at a nearly fixed rate until it runs low. Would that make your limit larger or smaller? Argue it before you test it.

## The data

### *Settings in the online lab*

| Setting | Range | Starts at |
|---|---|---|
| Cube side | 0.5 to 8 penetration lengths | 2 |

### *Checked on the research solver*

| | |
|---|---|
| **Solver (Reaction-diffusion, 3D)** | side 1: centre 0.9462; side 2: centre 0.8072; side 4: centre 0.4632 |
| **Known answer** | 0.9459; 0.8061; 0.4596 |
| **Error** | 0.78% (limit 1.00%): passed |

## The real world

Compare the penetration length with published oxygen diffusion limits and with the sizes of real organoids and spheroids.

## The honest limit

Consumption here is proportional to oxygen. Real cells keep consuming at a nearly fixed rate until oxygen is low, so say how that would change your estimate.

How the full project checks itself: A slab has the exact answer c = cosh((x-a)/L) / cosh(a/L), with penetration length L = √(D/k). The online lab runs the same project on the research solver (Reaction-diffusion, 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.