

FUNDAMENTALS OF AI-AUGMENTED ENGINEERING

CHAPTER 4/LAB-4: Engineering Lab (3 Hours)



Fill to the Mark

Brackets, sign changes & bisection — how do we find the crossing without walking the whole road?

Taner Onsay, Ph.D. & Sebnem Onsay
Authors
August 2026 · Revision 1.0

© 2026 AI First Engineering

Problem

›

Model

›

Compute

›

Analyze

›

Validate

›

Improve

Today a red cup and two syringes perform bisection — then a model certifies it

Fill to the Mark

What is this lab about? Bisection with water — +60 mL to the first crossing, trap the target between two states, halve every move (30, 15, 7.5 ...) — then certify the same target as a computed root $f(h) = 0$:

Search → Trap → Hunt → resolution wall → Write → Trap → Hunt → Trust → Decide (DO › OBSERVE › RECORD › TALK TO AI › EVALUATE › DECIDE ...)

LEARNING OBJECTIVES By the end of this lab, you will be able to:

- Perform bisection physically — crossing, bracket, halved move, sign — and log it
- Say what the water activity shares with bisection — and where the apparatus limits it
- Choose the instrument from the required resolution — and decide when to stop
- Write a target as $f(h) = 0$, certify a bracket, argue continuity — on the model
- Bisect by hand, in Excel, and in Python — tolerance from the requirement
- Validate the computed root against the real cup — state what the guarantee omits

TOOLS YOU WILL USE

- Red cup with one target line + two syringes — bisection in volume
- Ruler + calibrated target and tolerance — geometry and requirement
- Excel + Python / Jupyter — bisection in height, provable bound
- AI assistant — may propose the method, never signs for it
- Your engineering judgment — certify, own the tolerance, decide





What you will do (step-by-step):

- 1 Readiness quiz — then verify Excel, Python, AI workspace
- 2 Meet the kit · the rules · safety briefing · roles
- 3 Commit predictions P1–P4 — before any water moves
- 4 SEARCH IT · TRAP IT · HUNT IT with water — +60 mL to the first crossing · halve the move · log every operation
- 5 Debrief — DID WE JUST DO BISECTION WITH WATER? · the resolution wall · the calibrated target
- 6 WRITE IT — measure the cup · $V(h) \cdot f(h) = V(h) - V_{\text{target}}$ · the sign
- 7 TRAP IT — bracket from physics · certify the signs · three halvings by hand
- 8 HUNT IT — requirement → tolerance → schedule · Excel table · Python loop
- 9 TRUST IT — measure the real line · the contract · BREAK THE AI
- 10 Final certified statement · record → report · human review, then sign

Would you sign a root because the code printed “guaranteed”? You will sign one today — after you have checked the conditions yourself.





WRITE › TRAP › HUNT › TRUST · GUARANTEE = CONTRACT

In Chapter 4 the AI role deepens — the analyst may PROPOSE a method — while you remain the lead engineer: the bracket evidence, the tolerance, the conditions behind every guarantee, and the decision stay yours.

THE RECURRING LOOP — INSIDE PROBLEM › MODEL › COMPUTE › ANALYZE › VALIDATE › IMPROVE

DO

›

OBSERVE

›

RECORD

›

TALK TO AI

›

EVALUATE

›

DECIDE

... and back to DO — the conversation evolves as the experiment evolves.



You, not AI: run and log the physical search, write $f(x) = 0$ and own the sign, certify the sign change, set the tolerance from the requirement, verdict every AI claim — ACCEPT · MODIFY · REJECT — sign.



AI, for you: method proposals with reasons, candidate brackets for YOUR certification, the bisection you specified in Excel / Python, checks of your hand halvings — on the evidence you provide.

AI assists. Humans verify. Engineers decide — and read the contract.

Never ask AI to invent what you did. A recommendation arrives with its contract attached — or it goes back.



AI as engineering analyst — who does what in Chapter 4



HUMAN TEAM — LEAD ENGINEER

- Own the search record, the model, and the certified root
- Perform every physical operation and measurement — honestly
- Write the question as $f(x) = 0$ — and own what f means
- Certify the sign change before trusting any guarantee
- Set the tolerance from the engineering requirement
- Confirm the conditions behind every claimed bound · validate against the cup
- Choose the method — defend the choice and the decision — sign



AI — EMERGING ENGINEERING ANALYST

- Implement the bisection you specified — Excel or Python
- Suggest candidate brackets — for YOUR certification
- Check your hand halvings against its own
- Run both a sweep and bisection and tabulate the comparison
- Propose a method for the job — with reasons
- Explain any method on demand — analyst and tutor at once
- Draft report text from your evidence — never the certificate

A deeper analyst, the same constitution. A recommendation arrives with its contract attached — or it goes back.



Create your AI workspace — the engineering record

ONE WORKSPACE · “LAB-4 — FILL TO THE MARK”

- Lab instructions + report template
- Operation log · every move, the first crossing, every bracket update
- Predictions P1–P4 (committed) · debrief answers
- Cup geometry · calibrated target + tolerance · V(h) hand check
- Bracket certificate · hand halvings · Excel / Python log
- Root $\pm$ bound · validation vs the measured line
- BREAK THE AI findings · your verdicts · the decision

The record documents what happened.

The report communicates what matters.

Feed the record as you go — don't reconstruct it at 2:45.

STARTER PROMPTS

PROMPT-1 “Here are the corresponding lecture notes, lab instructions and the lab report template. Just read them for later reference and wait for my command.”
(drop or attach PDF versions of all 3 documents right into this chat)

PROMPT-2 “You are our AI engineering analyst for the LAB-4 Fill to the Mark root-finding laboratory. We are the lead engineers. You may propose methods and candidate brackets, implement the bisection we specify, check our arithmetic, and explain any method on demand — but every recommendation must arrive with its evidence: the bracket signs, the origin of any error bound, and who owns the tolerance. Never change our measured values, our requirement, or our units without flagging the change explicitly. Never invent measurements or calibration values, and never declare a root ‘guaranteed’ or ‘validated’ — checking the conditions and validating against the cup is our job. This workspace is the engineering record for our final report.”

Any capable AI platform works — a project/workspace feature (e.g., ChatGPT Projects) is convenient, but one persistent conversation is enough.



Search first — then certify the crossing

0:00–0:10		Phase 0 · Readiness quiz + verify Individual quiz — concepts, not calculation · Excel, Python, AI workspace proven working.
0:10–0:20		Phase 1 · Kit · rules · safety · the question One target line, two syringes, reservoir, roles · +60 mL to the first crossing, then halve — add or remove.
0:20–0:30		Phase 2 · Commit predictions P1–P4 +60 steps to the crossing · halvings before the wall · the syringe switch · total moves — unedited.
0:30–0:55		Phase 3 · BISECTION WITH WATER +60 to the first crossing · bracket · 30 → 15 → 7.5 ... · every move logged · correct + in tolerance + valid log.
0:55–1:10		Phase 4 · Debrief — BISECTION WITH WATER? Yes, in logic — limited by the apparatus · the resolution wall · the calibrated target revealed.
1:10–1:25		Phase 5 · WRITE IT Measure the cup · the frustum model $V(h) \cdot f(h) = V(h) - V_{\text{target}}$ · the sign gains meaning.
1:25–1:45		Phase 6 · TRAP IT Bracket from physics · certify $f(0) < 0 < f(H)$ · continuity · three halvings by hand.
1:45–2:15		Phase 7 · HUNT IT Requirement → tolerance → predicted n · Excel table, IF decision exposed · Python loop · root ± bound.
2:15–2:30		Phase 8 · TRUST IT Measure the real line · the contract's three conditions · what the guarantee never covers.
2:30–2:45		Phase 9 · BREAK THE AI Audit a “guaranteed” root · attack the contract, not the syntax · rewrite the claim.
2:45–3:00		Phase 10 · Certify · record · sign Final certified statement · evidence complete · human review — then sign.



Stations & logistics — eight parallel searches

EACH STATION'S KIT — NOMINALLY IDENTICAL



Red 16-oz soup cup with ONE target line

The line is the nominal target — its calibrated volume is revealed AFTER the activity.



One 60-mL syringe · one 10-mL syringe

Two resolutions — the required move decides which one can deliver or REMOVE it reliably.



Central reservoir + a station RETURN CUP

WALK, never run · draw only from the reservoir · removed water goes into the return cup — never the floor or reservoir.



Operation log sheet + ruler + team laptops

Printed log, one per team · ruler for Phase 5 · Excel · Python/Jupyter · AI assistant.

TEAM RULES

- Rotate roles: runner · syringe operator · recorder · inspector (calls BELOW / ABOVE / WITHIN).
- Everyone runs at least once; everyone records at least once.
- One engineering record per team — team number, members, date, station at the top NOW.
- The cup stays on the station table — water travels, the cup never does.

EXPERIMENTAL DESIGN: eight stations bisect toward the SAME kind of target with the SAME kit — one class comparison.



Safety & conduct — water on a floor is a slip hazard



SAFETY — WATER, WALKING, WIPING

- WALK — never run between reservoir and station; a running team is disqualified
- Water is for the cup — no drinking, no squirting; syringes stay below shoulder height
- Floor spills are wiped IMMEDIATELY — the operator stops and wipes before continuing
- Laptops and cables stay clear of the wet zone; the cup stays on the table
- REMOVED water goes into the station return cup — never onto the floor, never back into the shared reservoir



HONEST DATA

- Start the log row BEFORE the water moves — move, direction, syringe; then the observation
- The first crossing is recorded as a CROSSING — it is the most valuable row in the log
- A move with no plunger reading is a gap marked “—”, never an invention
- Decisions and reasons belong in the log — today, the algorithm is data
- Rejected AI advice belongs IN the record, with your verdict

Fast is nothing without correct logic, in tolerance, logged, and safe. The record is the deliverable.



Readiness check — concepts, not calculation



READINESS QUIZ

- ☐ 5 questions · 5 minutes
- ☐ Individual · closed notes · no AI
- ☐ Collected before any water moves



EXCEL + PYTHON

- ☐ Excel opens and computes
- ☐ Notebook runs a test cell
- ☐ You can interrupt the kernel



AI WORKSPACE

- ☐ “LAB-4” workspace created
- ☐ Starter prompts pasted
- ☐ Instructions + template uploaded



If something is broken

- Ask your instructor/lab facilitator for help — don't burn 40 minutes alone
- Paste the exact error into AI and ask what it means (often faster than you'd think)
- Keep your team moving on one working machine meanwhile

*All nine green? CHECKPOINT ① — the TA reviews quiz answers before the challenge starts.
Experienced with another IDE? Fine — as long as you can run and modify a notebook and support yourself.*



The rules of the activity — identical at every station

- 1 EMPTY cup on the table — ONE target line, visible to the inspector
- 2 SEARCH IT: +60 mL per trip, walking, until the level FIRST crosses the line
- 3 Log row BEFORE water moves: move · direction · syringe — observe AFTER
- 4 After every move the inspector calls: BELOW · ABOVE · WITHIN TOLERANCE
- 5 FIRST CROSSING = stop marching: previous BELOW + current ABOVE = bracket
- 6 HUNT IT: next move = HALF the last interval — 30 mL — direction as observed
- 7 ADD if BELOW · REMOVE if ABOVE — removed water into the station return cup
- 8 Halve every move: $30 \rightarrow 15 \rightarrow 7.5 \rightarrow 3.75 \dots$ pick the syringe that resolves it
- 9 STOP successfully when WITHIN tolerance. If the syringe cannot resolve the next halving first, stop and report the resolution limit.
- 10 Success = correct logic + in tolerance + valid log + safe — not the fastest



Why so strict? The rules ARE the algorithm: coarse search, first crossing, maintained bracket, halved move, observed sign, stopping decision. Change one rule and you are no longer bisecting.



A red cup, one line — and a target you can cross, then trap



FIND: the level AT the target line — within tolerance, with a valid bisection log

...+60-mL steps to the first crossing, then halved moves — ADD or REMOVE — decided by what you see. The line is the nominal target; the tolerance is the acceptance region.

The parallel: coarse sweep → first crossing → bracket → halve

Chapter 2 walked until it crossed. Chapter 4: the crossing is INFORMATION — use the bracket, probe halfway.

Assumptions go in the record — including this one:

TALK TO YOUR AI TEAMMATE “Log our setup: one target line, 60-mL and 10-mL syringes, reservoir across the room, +60 mL to the first crossing, then halved moves add-or-remove, stop within the posted tolerance.”

ENGINEER DECIDES *estimating the cup from “16 oz” is a MODELING choice — state it; the calibrated target arrives in Phase 4.*



Predict — in writing, right now

1

P1 — +60-mL additions before the first crossing

How many 60-mL steps until the level first passes the line? Wrong is fine; uncommitted is not.

2

P2 — Halvings before the resolution wall

After the first crossing: how many physical halvings (30, 15, 7.5 ...) can you perform before instrument resolution matters? Commit a number.

3

P3 — When does the 10-mL syringe take over?

At roughly what requested move size (mL) will the 60-mL syringe stop being adequate? One number plus one sentence.

4

P4 — Total physical operations to the tolerance

Count every ADD and REMOVE: how many moves until the state is WITHIN the posted tolerance? Commit a number.



TALK TO YOUR AI TEAMMATE “Here are our committed predictions P1–P4 — log them. Do not evaluate them yet.”

WHY PREDICT FIRST?

A prediction made after the activity isn't a prediction — it's a story. Committing first is what makes the Phase 4 comparison honest — and it is how real engineering validation works.

Predictions go in the record BEFORE the first move — and are never edited afterward.



ENGINEER DECIDES the numbers you commit are yours — AI logs them, you own them.

Problem



Model



Compute



Analyze



Validate



Improve



Bisection with water — search, trap, hunt

1

SEARCH IT — +60 mL until the first crossing

Empty cup. Add exactly 60 mL per trip and observe after each: BELOW ... BELOW ... then ABOVE. The moment the level passes the line, STOP marching. Log that row as the crossing.

2

TRAP IT — the first crossing is a bracket

Previous state BELOW + current state ABOVE: the target is trapped inside the last 60-mL interval. The PAIR of states — not the 60 mL — is your bracket. Write both cumulative volumes down.

3

HUNT IT — halve the move, read the sign

Last interval 60 mL → move 30 mL the other way (REMOVE). BELOW → ADD 15; ABOVE → REMOVE 15. Then 7.5, 3.75 ... keep the two states that straddle the line. STOP successfully when WITHIN tolerance. If the syringe cannot resolve the next halving first, stop and report the resolution limit.



TALK TO YOUR AI TEAMMATE “Log our plan: +60 mL per trip until the first crossing, then 30 → 15 → 7.5 ... in the direction the observation demands. Do not choose our stopping point.”

THE WINNING CONDITION

CORRECT LOGIC + WITHIN TOLERANCE + A VALID BISECTION LOG + SAFE. Speed is not scored: reaching the line by guessing wins nothing; a log showing the first crossing, both bracket states, every halved move with its direction and sign, and a defended stop is bisection with water.

A crossing is information. Do not keep marching. Use the bracket.



ENGINEER DECIDES the halving is the algorithm; the direction is the observation; the syringe is your decision.

Problem ›

Model ›

Compute ›

Analyze ›

Validate ›

Improve



Every move is a row — the crossing, the bracket, the halving

Step	Cumulative (mL)	Move (mL)	ADD / REMOVE	Syringe	Observed: BELOW / ABOVE / WITHIN	Bracket [low, high] (mL)	Next half-step · decision
1 · ex.	60	60	ADD	60 mL	BELOW	—	+60 again

One row per move — write the move, direction and syringe BEFORE water moves; the cumulative volume, the observed state and the bracket AFTER. In order, the log must show: +60 ... +60 → FIRST CROSSING → -30 → ±15 → ±7.5 → ... → STOP.

Before the first crossing the bracket column is “—”; from the crossing on it holds the two cumulative volumes that straddle the line · the direction after -30 is never pre-filled — it comes from the observation · a move with no plunger reading is a gap marked “—”, never an invention.

Record EVERY move — including the one that crossed the line. The crossing is not a mistake; it is the row that creates the bracket.

 **TALK TO YOUR AI TEAMMATE** “Here is our operation log so far. Check that every move after the crossing is half the previous one and that the bracket column always holds two states on opposite sides — flag slips; do not choose our next move.”

 **ENGINEER DECIDES** the operator moves the water, the recorder writes, the inspector calls the sign — nobody moves water before the row is started.

CHECKPOINT ② — first crossing logged with both bracket states · halving under way · roles rotating.





Did you just do bisection with water? YES — with limits.



Two states on opposite sides — the first crossing

BELOW then ABOVE: the target trapped between two known volumes



Probe halfway — keep the half
that still straddles the line



The observation supplied the sign — BELOW, ABOVE, WITHIN



Every move halved — a schedule, not an instinct

60 → 30 → 15 → 7.5 → 3.75 — until the instrument said stop

WHERE THE PHYSICAL GUARANTEE LEAKS

- Syringe scale resolution, delivery / removal error, retained droplets, transfer losses.
- Meniscus and visual judgment — and the thickness of the line itself.
- The cup's real geometry, and the calibration's own uncertainty.
- A mathematical f on a continuous model has a THEOREM behind its bound; water has an INSTRUMENT behind its bound.

Same logic, two spaces: this afternoon the variable was VOLUME (mL); in Phase 5 it is HEIGHT (mm) — same algorithm, different units, a provable bound.

The LOGIC was bisection. The CERTIFICATE is limited by the apparatus.

Problem



Model



Compute



Analyze



Validate



Improve



Two syringes, one line — the Chapter 3 toolkit, in your hands



Two syringes = two resolutions

RESOLUTION

The algorithm says HALVE AGAIN; the instrument asks CAN I RESOLVE THAT? The 10-mL answers where the 60-mL cannot.



Every move = an evaluation you paid for

COST

Why not the 10-mL from the start? Count the trips to the first crossing. Why not the 60-mL to the end? It cannot deliver 3.75 mL reliably.



The line = the nominal target · the tolerance = the acceptance region

TOLERANCE

The target is a point. How close is close enough is a requirement — posted, not chosen.



Halve only while the halving means something

SUFFICIENCY

Mathematical resolution halves forever; physical resolution hits a wall. Stopping there is engineering, not surrender.



Reading a plunger has uncertainty too

MEASUREMENT

Near the line, ± 1 mL on a move is the difference between WITHIN and ABOVE — confidence shrinks as moves shrink.



One cup is an exercise — 100,000 cups is a factory

MASS PRODUCTION

A wasted move per cup is real money at scale. Chapter 4 adds: the bracket, the schedule — and the conditions behind the guarantee.

Chapter 3 gave you resolution, convergence, tolerance, sufficiency and cost. Chapter 4 adds the bracket, the sign change, the halving schedule — and the contract behind its guarantee. Today you met the resolution wall with a syringe in your hand.



The line, in millilitres — revealed only now

1

Read the posted calibration

TARGET LINE V_{target} _____ mL · ACCEPTANCE TOLERANCE $\pm$ _____ mL — measured and set by the instructor before class. Copy them into the record, with units.

2

Compare — your bracket vs the calibration

Your last two straddling states [_____, _____] mL and final volume vs V_{target} : calibration inside your bracket? final state within tolerance? Where could a difference live — plunger reading, drips, meniscus, calibration?

3

Revisit P1–P4 — honestly

+60 steps to the crossing, halvings before the wall, the syringe switch, total moves: what did the evidence say about each prediction? Answer in the record; never edit the originals.

CALIBRATION NOTES

The line was calibrated by pouring measured water — a measurement with its own uncertainty. Note the scale step of each syringe: 60-mL _____ mL, 10-mL _____ mL. A calibrated value is a REFERENCE, not the truth — Chapter 3's honest naming, again.

Is the calibrated V_{target} inside your final bracket? If yes, log and calibration agree. If not — name the sources; maybe nobody is wrong.

TALK TO YOUR AI TEAMMATE *"Here are the calibrated target and tolerance and our final bracket — check whether the calibration lies inside our bracket and list where any difference could come from."*

ENGINEER DECIDES *the calibrated V_{target} is the reference for Phase 5 — copy it exactly, units and all.*

Problem



Model



Compute



Analyze



Validate



Improve



One geometric model stands in for the real cup



The model: $V(h) = (\pi \cdot h / 3) \cdot (r_b^2 + r_b \cdot r_h + r_h^2)$

$r_h = r_b + (r_t - r_b) \cdot h/H$ · straight tapered walls · $\text{mm}^3 \div 1000 = \text{mL}$



Measure YOUR cup — inner diameters at bottom and rim, inner depth H — with a ruler, to the millimetre



Hand-check ONE point — V at half depth, on paper



Then write it: $f(h) = V(h) - V_{\text{target}} \rightarrow$ the line is the ROOT

V_{target} = the calibrated target-line volume — the sign of f carries the meaning

$r_b = 35, r_t = 45, H = 100 \text{ mm}: V(50) \approx 221 \text{ mL}$ (illustrative — use YOURS)

WHAT THE MODEL LEAVES OUT

- Curved or ribbed walls · the lid seat · a rounded bottom — real cups are not frusta.
- Your ruler: $\pm 0.5 \text{ mm}$ on a diameter, and V goes like r^2 .
- The meniscus — what “level with the line” means to the eye.
- Real geometry, represented by three numbers. Useful $\neq$ complete.

A model this simple can still be **USEFUL** — if validated for its purpose. Phase 8 tests it against the real line.

Problem



Model



Compute



Analyze



Validate



Improve



Python — the model gets a name, then f gets its sign

WORKING METHOD — YOUR MEASUREMENTS DRIVE THE CODE

- AI may coach the code; the GEOMETRY stays yours — measured, in mm
- `cup_volume()` must reproduce your hand-checked point
- Units: mm³ inside, mL out — one ÷ 1000, in one place
- $f(h) = \text{cup_volume}(h) - V_{\text{target}}$ — V_{target} is the CALIBRATED target volume
- Evaluate f at two heights and say what each sign MEANS
- Same signature everywhere — the Phase 7 loop calls this f
- Unexplainable code does not enter your program

Two small functions is right-sized.

A model and the question — both testable.

If the hand check and `cup_volume()` disagree, the code is wrong or the arithmetic is — find out which before anything is built on it.

THE CODE

```
import math
def cup_volume(h, rb, rt, H):          # mm in, mL out
    r = rb + (rt - rb) * h / H         # radius at height h
    return math.pi*h/3 * (rb*rb + rb*r + r*r) / 1000.0
def f(h):                              # f(h) = 0 at the target
    line
    return cup_volume(h, rb, rt, H) - V_target
# YOUR measurements (illustrative values shown):
rb, rt, H = 35.0, 45.0, 100.0          # mm
V_target = 400.0                      # mL — calibrated target
line
print(cup_volume(50.0, rb, rt, H))     # 221.2 — matches the hand
check?
print(f(0.0), f(H))                   # -400.0 105.3 -> a sign
change
```

Same signature everywhere — call $f(h)$ exactly that way inside the Phase 7 loop too.



Three signs, three physical situations — before any search

THE SHAPE OF THE QUESTION

$f(h) = V(h) - V_{\text{target}} \quad [\text{mL}]$
 $f(h) < 0 \rightarrow$ BELOW the line: at height h the cup holds less than the target
 $f(h) = 0 \rightarrow$ THE LINE: the root the decision cares about
 $f(h) > 0 \rightarrow$ ABOVE the line: more than the target — this afternoon, REMOVE

The classic gap: reporting $V(h)$ instead of $f(h)$ — and losing the sign that carries the meaning.

Zero is not algebraic decoration; it is the edge the decision cares about — the inspector's BELOW / ABOVE, now in millilitres of a model.

YOUR f STATEMENT MUST CONTAIN:

- ☐ The function, with units on both sides
- ☐ What each sign means for THIS cup
- ☐ Where V_{target} came from — the calibration
- ☐ Two evaluated values of f — heights and signs
- ☐ Which crossing the decision needs (there is one)
- ☐ One sentence: why the SIGN, not the value, drives the search
- ☐ ...all written BEFORE any bracket is chosen

WRITE IT before you TRAP IT — a thousand engineering questions, one mathematical shape: $f(x) = 0$.



Certify the bracket — the guarantee begins here

1

Choose the cage from physics

$a = 0$ (an empty cup holds nothing — below the target) and $b = H$ (a brim-full cup holds more than the target volume — above). No computing needed to know the signs. Generous is fine.

2

Certify — evaluate both ends

$f(0) = -V_{\text{target}} < 0$. $f(H) = V_{\text{full}} - V_{\text{target}} > 0$. Two model runs; both signs written in the record. In code: `assert f(a) * f(b) < 0` — Principle No. 2, one line.

3

Read the fine print

Continuity: $V(h)$ is a polynomial in h — continuous. One crossing: more height never means less water, so f climbs the whole way. Which root the decision needs: the only one.



TALK TO YOUR AI TEAMMATE “Here is our $f(h)$ and our bracket $[0, H]$ with both signs. Check the certification and say what the guarantee will and will not cover.”

THE CONTRACT

A sign change on a continuous f traps at least one root: a continuous curve cannot get from below zero to above zero without crossing it. The bracket is the guarantee — no sign change, no promise. Continuity is a property of the MODEL: you certify it by understanding $V(h)$, not by running code.

Two evaluations bought a theorem's protection. Your two straddling volumes this afternoon bought the same LOGIC — with an instrument's uncertainty



ENGINEER DECIDES the assert certifies the trap in code — the continuity argument is yours, on paper.

Problem ›

Model ›

Compute ›

Analyze ›

Validate ›

Improve

No AI, no Python, no Excel — the exam rep

n	midpoint m (mm)	V(m) (mL)	sign of f(m)	bracket kept (mm)	
1	H / 2	fill in	- / +	fill in	
2	fill in	fill in	fill in	fill in	
3	fill in	fill in	fill in	fill in	

Row 1: $m = H/2$ — one model run $V(m)$, compare with $V_target \rightarrow$ the sign of $f(m) \rightarrow$ keep the half whose ends still differ in sign. Rows 2 and 3 the same. Every number is your geometry; the only judgment per row is WHICH HALF SURVIVES.

PREDICT first: after three halvings, how wide is the bracket — and what $\pm$ can you already guarantee? (Illustrative: $H = 100\text{ mm} \rightarrow 12.5\text{ mm wide, } \pm 6.25\text{ mm}$ — before a single line of code.)

The arithmetic is Phase 5. The judgment — which half survives — is Chapter 4. That judgment is the exam question.



TALK TO YOUR AI TEAMMATE “Here are our three hand halvings. Check each $V(m)$ and each surviving bracket against your own arithmetic — flag disagreements only.”



ENGINEER DECIDES if AI and your row disagree, one of you evaluated $V(m)$ wrong — find out which before Phase 7.

Three model runs. A guaranteed $\pm$ you can point at — in millimetres, on a model, with a theorem behind it. Same logic as your $30 \rightarrow 15 \rightarrow 7.5\text{ mL}$; a different variable, and a different kind of certificate.

Problem



Model



Compute



Analyze



Validate



Improve

The requirement chooses the tolerance — and n

Step	What you compute	Your value	Who owns it
① Requirement (GIVEN)	the posted acceptance tolerance on the target volume: $\Delta V_{\text{allowed}} = \pm \text{ ______ }$ mL (from the Phase 4 calibration)	 ______ mL	the specification
② mL per mm at the line	$A \approx \pi \cdot r_h^2 / 1000$, r_h at your hand-halving bracket	 ______ mL/mm	your geometry
③ Height tolerance	$\text{tol}_h = \Delta V_{\text{allowed}} / A$	 ______ mm	derived — never chosen
④ Halvings needed	smallest n with $W_0 / 2^{n+1} \leq \text{tol}_h$, $W_0 = H$	$n = \text{ ______ }$	the schedule — BEFORE running
⑤ Evaluations, total	n midpoint runs + 2 endpoint checks	 ______ 	the price list

Illustrative ($r_b = 35$, $r_t = 45$, $H = 100$ mm, posted tolerance ± 2.5 mL): $A \approx 5.9$ mL/mm; $\text{tol}_h \approx 0.42$ mm; $100/2^{n+1} \leq 0.42 \rightarrow n = 7$; 9 evaluations in total. Yours will differ — the METHOD is the point.

Record: “Our requirement is $\pm \text{ ______ }$ mL, i.e. $\pm \text{ ______ }$ mm at the line; from a ______ mm cage, bisection needs ______ halvings — predicted before running.”

Tolerance is a requirement, not a Python setting. Here the requirement is a VOLUME — converting it into a height tolerance is engineering, done before the loop runs.

 **TALK TO YOUR AI TEAMMATE** “Here is our requirement in mL and our cross-section at the line. Check our height tolerance and our halving count from $W_0/2^{n+1} \leq \text{tol}$. Do not pick a tolerance for us.”

 **ENGINEER DECIDES** a teammate proposes $\text{tol} = 0.001$ mm “to be safe”? Ask which requirement asked for it — and what it costs.

For the first time you can promise accuracy in advance — and price it in evaluations. Accuracy becomes a schedule, not an experiment.





The bisection loop — your f, one WHILE, one assert

THE HUNT — AUTOMATED

```

a, b = 0.0, H           # physical cage (mm)
fa, fb = f(a), f(b)     # 2 endpoint evaluations
assert fa * fb < 0      # certify the sign change
tol = tol_h             # from the REQUIREMENT (mm)
n = 0
while (b - a) / 2 > tol:
    m = (a + b) / 2
    fm = f(m); n = n + 1 # ONE new evaluation per halving
    print(n, a, b, m, fm, (b - a) / 2)

```

If the assert fails: your cage does not trap a sign change — is V_{target} above the brim-full model volume?

Re-read the calibration and your geometry before touching the loop; the guarantee rests on the certified signs.

BEFORE YOU RUN IT — WRITE DOWN:

- ☐ Your predicted n — from the Phase 7 schedule
- ☐ The first midpoint — $H/2$ — and its sign (your hand row 1)
- ☐ What stops the loop: the half-width, never a count
- ☐ What the assert would do if the cage had no sign change
- ☐ Then run — compare the printed rows with your three hand rows
- ☐ Excel first, if you prefer: same rows, the IF decision exposed
- ☐ Any disagreement is evidence — chase it before Phase 8

Seventeen printed digits against a ± 0.4 mm bound? You know better: report the midpoint to the digits the bound justifies.



Inspect the table — don't just fill it

n	a (mm)	b (mm)	m (mm)			
1	0	H	H / 2			
2	fill in	fill in	fill in			
3	fill in	fill in	fill in			
...	...	...	...			
n_final	fill in	fill in	fill in			

Excel: a and b carried from the previous row, $m = (a + b)/2$, sign test $f(a) \cdot f(m) \leq 0 \rightarrow$ IF decides new a and new b ; copy the row down. Python prints the same rows. Same algorithm, two platforms.

Report: $h^* = \text{_____ mm} \pm \text{_____ mm}$ (midpoint $\pm$ final half-width) — digits justified by the bound, not by the printout.

Every row preserves the evidence: the bracket is visible data, the IF decision is exposed, and the half-width shows the schedule being kept.

TALK TO YOUR AI TEAMMATE “Here is our iteration log. Verify each row keeps the half with the sign change and that the half-width halves every row. Do not report a root for us.”

ENGINEER DECIDES does the root estimate have to move monotonically toward the line? Look at your m column before answering.

PREDICT $\rightarrow$ RUN $\rightarrow$ INSPECT — the printed n must match the schedule. If it doesn't, the code or the tolerance is wrong.

Problem



Model



Compute



Analyze



Validate



Improve



Measure the real line — then sort the disagreement



Numerical — the bisection bound

NUMERICAL

Certified: $\pm$ the final half-width. If the gap is larger than that, the bound is NOT the culprit.



Model shape — straight walls assumed

MODEL

Curved or ribbed walls, a rounded bottom, the lid seat: the frustum is a stand-in.



Geometry measurement — diameters, depth

MEASUREMENT

$V \propto r^2$: a 1-mm diameter error moves $V(h)$ by several mL near the top.



Calibration — the posted V_{target}

PARAMETER

The reference has its own uncertainty; a millilitre either way moves the root.



Units — mm^3 , mL, cm

UNITS

A missing ± 1000 converges beautifully to nonsense. Physics catches it: is h^* inside $[0, H]$?



The right root — the crossing the decision needs

RIGHT ROOT

$V(h)$ climbs the whole way: one crossing. Say so from the physics — don't assume it.

Measure the real height of the target line with your ruler: $h_{\text{measured}} = \underline{\hspace{1cm}} \text{ mm}$. Compare with $h^* \pm \text{bound}$. The guarantee covers the computation only — the gap belongs to the other five rows.



Read the conditions before you trust the promise



① SIGN CHANGE

$f(a) \cdot f(b) < 0$ — checked in code, one assert, every time. Show both values in the record. Without it the loop still “converges” — to an endpoint.

Certified: YES / NO — with the two signs.



② CONTINUITY

A property of the MODEL, not the program. $V(h)$ is a polynomial — continuous. A lookup table with a gap or a hard IF could fake a sign change with no root at all.

Certified by understanding f — not by running it.



③ THE RIGHT ROOT

“At least one root” is not “the root the decision needs.” Here f climbs the whole way: exactly one crossing. Say it from the physics — more water, never less.

Which crossing — and why it is the only one.

WHAT THE CONTRACT NEVER COVERS: the model, the geometry measurements, the calibration, the units. A certified bracket on a wrong cup model converges — beautifully — to the wrong cup’s line.

RUNS $\neq$ CORRECT · CONVERGED $\neq$ CORRECT · and now: a guarantee holds only where its conditions hold — checking them is engineering work.



Phase 9 · 2:30–2:45 · Audit the guarantee — not the syntax



Interrogate the contract — pick at least three

Sign change certified — or assumed? · continuity — did it ask about f ? · which crossing? · does ± 0.01 equal the final half-width (10 halvings of 100 mm $\rightarrow \pm 0.049$)? · whose cup is 100 mm deep? · who owns ± 0.01 mm? · known case: $V(h^*) \approx V_{\text{target}}$?



Re-read the AI conversation

Did AI silently assume a cup depth, a tolerance, an iteration count, or a bracket you never certified? Verdict what you find ›



Document every attack

What you asked, what it claimed, what you concluded — “the guarantee survived our audit” is a finding too, if you show HOW you verified each condition.

VERDICT WHAT YOU FIND:



ACCEPT — with your reason



MODIFY — then retest



REJECT — with evidence



VERIFIED — also a result

BREAK THE AI: “I ran bisection on $[0, 100 \text{ mm}]$ for 10 iterations and found the line at $h = 82.71 \text{ mm} \pm 0.01 \text{ mm}$, guaranteed by the bisection theorem — the line can be placed with certainty.” Audit the **CONTRACT** behind the confidence, then rewrite the claim into one you would sign.



Rebuild the claim — from the contract, not the confidence



1 · EVIDENCE

List what a certified root needs:
bracket + both signs, continuity, the
crossing the decision needs, the final
half-width, the requirement behind
the tolerance, a known-case check.

Present? Missing?

**“Guaranteed by the theorem” —
which conditions did it check?**



2 · INTERROGATE

Put your evidence against the claim:
your certified signs, your schedule
 $W_0/2^{n+1}$, your measured H , your
requirement-derived tol_h , your
measured line. Where exactly does it
overreach?

**± 0.01 mm from 10 halvings of
100 mm? Compute $100 / 2^{11}$.**



3 · REWRITE → DEFEND

“Bracket $[0, _]$ certified ($f = _, _$);
 $V(h)$ continuous, one crossing; after
 $_$ halvings $h^* = _ \pm _$ mm,
meeting $\pm _$ mm; $V(h^*) = _ \text{ mL} \approx$
 V_target ; measured line $_ \text{ mm}$ —
the gap is model / measurement”

**AI may draft the rewrite — it
becomes YOURS when you can
defend every condition.**

Never grade tone — grade the CONTRACT. A confident root with no certified signs, no final width, and no requirement is not a guarantee; it is typography.

A guarantee that survives your audit is one you may use — and you state exactly how you verified it. That, too, is engineering.



Make — and defend — the certified root

“For our cup ($r_b = \underline{\hspace{1cm}}$, $r_t = \underline{\hspace{1cm}}$, $H = \underline{\hspace{1cm}}$ mm) and the calibrated target $V_{\text{target}} = \underline{\hspace{1cm}}$ mL, we certified the bracket $[0, \underline{\hspace{1cm}}]$ ($f = \underline{\hspace{1cm}}$, $\underline{\hspace{1cm}}$); $V(h)$ is continuous with one crossing. The posted $\pm \underline{\hspace{1cm}}$ mL tolerance gives $\text{tol}_h = \underline{\hspace{1cm}}$ mm; bisection needed $\underline{\hspace{1cm}}$ halvings (predicted $\underline{\hspace{1cm}}$). Result: $h^* = \underline{\hspace{1cm}} \pm \underline{\hspace{1cm}}$ mm. Known case: $V(h^*) = \underline{\hspace{1cm}}$ mL. Measured line: $\underline{\hspace{1cm}}$ mm; the gap is attributed to $\underline{\hspace{1cm}}$. Decision: the model IS / IS NOT adequate for placing the line.”

YOUR DEFENSE — one line of evidence on each rule:

The four lines: ① certified sign change + continuity ② hand halvings agree with the log ③ predicted $n = \text{run } n$; root $\pm$ half-width ④ known case + measured line + stated limitations.

An engineering answer is a ROOT + ITS CERTIFICATE + ITS VALIDATION + A DECISION. A bare root is a guess wearing a theorem.

Problem



Model



Compute



Analyze



Validate



Improve



Leave with the record complete — the report follows



EVIDENCE OF DOING

Operation log — every move, the crossing, every bracket

Predictions P1–P4 — unedited

Calibrated target + tolerance, with units

Cup geometry + $V(h)$ hand check

Measured line height



EVIDENCE OF THINKING

BISECTION WITH WATER? — answered, with limits

Bracket certificate + hand halvings

Requirement $\rightarrow$ tol_h $\rightarrow$ predicted n

Excel / Python log + BREAK THE AI



DECISIONS RECORDED

Root $\pm$ bound, justified digits

Contract conditions checked

Validation verdict + named sources

Model adequate: YES / NO + why



BEFORE YOU LEAVE

Missing report items listed

Assembly assigned — who, by when

Station reset — cup and return cup emptied into the waste bucket & dried, syringes emptied, floor dry

CHECKPOINT ③ — TA sign-off

Final formatting may be finished after lab — a complete, honest record tonight beats a polished report missing evidence.

Problem



Model



Compute



Analyze



Validate



Improve



Final report · human review & approval

“Before submission, we reviewed the completed report against our original engineering record — measurements, calculations, observations, code, plots, tests, validation evidence, and conclusions. We accept responsibility for the submitted engineering work.”

Human reviewed and approved.

EVERY TEAM MEMBER SIGNS — BEFORE
SUBMISSION (digital signature: create
once, copy-paste)

The recurring question, now literal: ***Would you confidently put your name on this result?***

AI assists. Humans observe. Humans verify. Humans decide. Humans approve.

Problem ›

Model ›

Compute ›

Analyze ›

Validate ›

Improve



One concise report — built from your record



FRAME IT

- 1 Problem / objective
- 2 The activity — rules & roles
- 3 Assumptions
- 4 Apparatus · safety · photo
- 5 Predictions P1–P4



GROUND IT

- 6 Operation log — every move
- 7 Debrief — bisection with water?
- 8 Calibrated target · bracket vs calibration



BUILD IT

- 9 Geometry + $V(h)$ + hand check
- 10 WRITE IT — $f(h)$ and the sign
- 11 TRAP IT — certified bracket
- 12 Three halvings by hand
- 13 Requirement $\rightarrow \text{tol}_h \rightarrow n$
- 14 Excel / Python log + screenshot



OWN IT

- 15 TRUST IT — measured line + contract
- 16 AI analysis + BREAK THE AI
- 17 Final certified statement
- 18 Method judgment + limitations
- 19 Human review & approval — signed

A few honest sentences per section. The log, the tables, and the screenshot carry the evidence — prose carries the certificate.

Problem



Model



Compute



Analyze



Validate



Improve

WHAT JUST HAPPENED

You searched with water until you crossed a line,
trapped the target between two states and halved your
way in,
wrote the same target as $f(h) = 0$, trapped it, and hunted it
on a schedule,
**checked the contract behind the guarantee — and
validated the root against the real cup.**

SEARCH → TRAP → HUNT — first with water, then with a model. The syringes met a resolution wall; the model's bisection closed in with a bound you could prove before running. And that bound covered only the computation: the cup, the ruler, and the calibration still needed you. Chapter 5 asks the question this laboratory earned: bisection threw the VALUES of f away and kept only the signs — what happens when we let the values steer?

Problem

›

Model

›

Compute

›

Analyze

›

Validate

›

Improve

RUBRIC for the Lab Report

© 2026 AI First Engineering · Fundamentals of AI-Augmented Engineering · CHAPTER 4 / LAB-4

FUNDAMENTALS OF AI-AUGMENTED ENGINEERING

CHAPTER 4 / LAB-4

Engineering Report Rubric

Fill to the Mark — Main LAB-4 Team Report

Suggested Assessment · 40 points

Used by the instructor and, where applicable, by an AI grading agent. Assessment rewards the honest bisection record, the certified bracket, requirement-owned tolerance, bound-based reporting, validation against the physical cup, and human responsibility.

Criterion	What full credit looks like	Pts
Problem, rules, assumptions & conditions	Clear objective; the physical rules (+60 mL to the first crossing; then halved moves, add or remove, decided by observation) and roles stated; assumptions listed; the calibrated target volume and acceptance tolerance recorded with units as posted by the instructor — never invented.	3
Committed predictions	P1–P4 (steps to the crossing · halvings before the resolution wall · the syringe switch · total moves) recorded before the first move and left unedited; each revisited against the evidence afterward. Prediction correctness is not graded — commitment and reasoning are.	3
Operation log — the bisection record	Every move logged in order: cumulative volume, move, direction, syringe, observed state, current bracket; the first crossing marked; every move after it half the previous one with the direction consistent with the observed sign; the bracket column always holds two states on opposite sides; a defended stopping decision. The algorithm can be reconstructed from the log alone.	6
Debrief — BISECTION WITH WATER?	Shows from the team's own log the first crossing, the initial bracket, the first half-step, the halving, an add/remove decision, a bracket update, the syringe-resolution decision and the stop; distinguishes the bisection LOGIC (yes) from the physical certificate (limited by syringe resolution, transfer losses, meniscus and line judgment, geometry, calibration); keeps volume-space and height-space bisection distinct. Chapter 3 connection: the instrument switch is driven by required resolution; the resolution wall decides the stop.	4
Cup geometry, model & WRITE IT	Inner diameters and depth measured with units; the frustum model applied with one hand-checked $V(h)$ point that the Python function reproduces; $f(h) = V(h) - V_target$ written and each sign given its physical meaning (below the line · at the line · above the line).	4



RUBRIC for the Lab Report

© 2026 AI First Engineering · Fundamentals of AI-Augmented Engineering · Chapter 4 / LAB-4

Engineering Report Rubric — continued

Suggested Assessment · 40 points

Criterion	What full credit looks like	Pts
TRAP IT — the certified bracket	Bracket $[0, H]$ defended from physics; $f(0)$ and $f(H)$ evaluated and their signs shown; continuity argued from the model form (a polynomial in h); single crossing argued (more height never means less water); the assert present in code.	4
Hand halvings, requirement → tolerance → schedule	Three halvings by hand with the surviving bracket and half-width each row; the height tolerance derived from the posted acceptance tolerance through the cross-section at the line $(\Delta V_{\text{allowed}} / A)$; the halving count predicted from $W_0/2^{n+1} \leq \text{tol}$ BEFORE running, and compared with the run.	5
Excel / Python bisection + iteration log	Excel rows or the Python log show n , a , b , m , $f(a)$, $f(m)$, the sign decision and the half-width; the loop stops on width, not on a fixed count; the root reported as midpoint $\pm$ half-width with digits justified by the bound; screenshot of the team's own session included.	5
TRUST IT — validation & the contract	Predicted line height compared with the measured line on the real cup; the difference attributed to named sources (model shape, diameter measurement, calibration, units) — never to the bisection bound; the three contract conditions checked and what the guarantee does NOT cover stated explicitly.	3
BREAK THE AI, final statement & approval	The planted guarantee claim attacked with at least three specific, checkable points and rewritten into a statement the team would sign; the final certified statement complete (root $\pm$ bound, tolerance and its owner, conditions checked, validation result, decision); human approval signed.	3
TOTAL		40



CHAPTER - 4

Lab 4 · ≈ 30 MIN

Lab Extension Challenge:

(Individual Assignment)

Students can do it at their convenience (individually) but remind them it is part of the Lab-4 assessment.

Problem



Model



Compute



Analyze



Validate



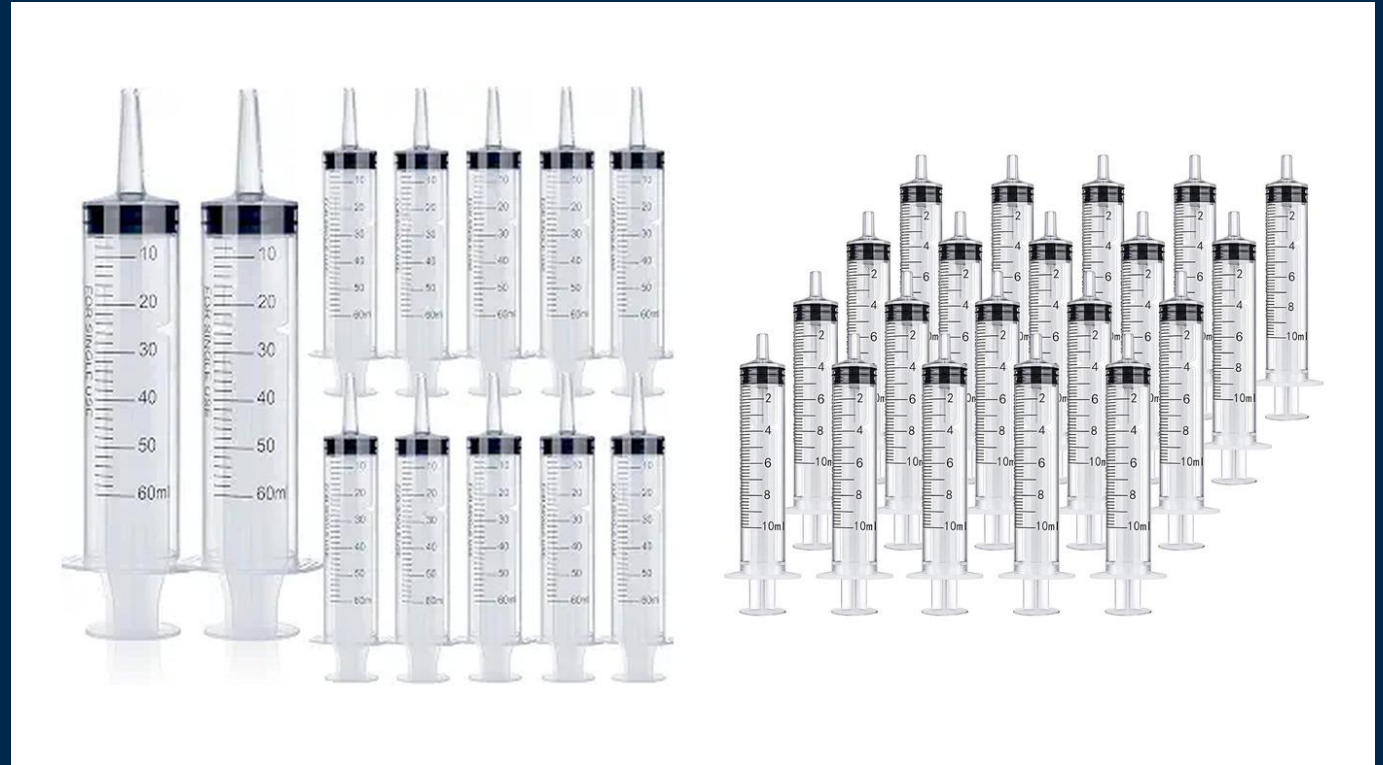
Improve



One Cup, Five Targets — Choose the Method

Bisection certifies one number. A sweep draws the landscape. Which do you buy for five targets — and what does an AI’s “guaranteed” interpolation actually guarantee? Predict first — then compute.

An extension of the Lab-4 root-finding study. You already own the cup model, a certified bracket, and a working bisection — now a product family of FIVE fill targets puts METHOD CHOICE itself to the test, and an AI shortcut claims a guarantee it may not have.



Problem

>

Model

>

Compute

>

Analyze

>

Validate

>

Improve



An assessment mini-challenge — not another lab report

1

Designed as part of the Lab-4 assessment

The rubric is on the last slide of this section.

2

Separate from the 3-hour lab

An individual follow-on, completed at your convenience — no water, no cup: your Lab-4 geometry, calibrated target, and functions are all you need.

3

Submit a separate short report

Download and use Chp4-Lab4-LabExtension-Challenge-OneCupFiveTargets-Report-Template.docx

THE ENGINEERING QUESTION

A product line needs fill marks for FIVE target volumes on the same cup.

Bisection five times? One sweep of $V(h)$?

An AI proposes a shortcut with a “guaranteed” bound. Which method — and which claims — deserve your **Signature?** flow as the whole lab:

Predict → Compute → Compare → Audit
AI → Validate → Decide



ENGINEER DECIDES

Engineers decide.

the question is not which method converges — it is which method certifies what THIS job needs, at what cost. AI assists. Humans verify.

Problem

>

Model

>

Compute

>

Analyze

>

Validate

>

Improve



Commit to a prediction

1

The halving count

Five targets, each bisected from $[0, H]$ to your Lab-4 tol_h : how many model evaluations in total? Commit a number.

2

The sweep

One sweep of $V(h)$ every 1 mm from 0 to H brackets ALL five targets at once. How many evaluations — and what $\pm$ does a 1-mm

3

Cheaper — and enough?

Which costs less for five targets? Does the cheaper one meet your tol_h ? Circle: bisection $\times 5$ · sweep · neither alone.

4

The shortcut

An AI will propose interpolating between two sweep points, claiming a guaranteed ± 0.05 mm. Before seeing it: can any theorem you know guarantee that? YES / NO



ENGINEER DECIDES record your predictions NOW — no revising after you compute (correctness is not graded).

THE MODEL

$$f_i(h) = V(h) - V_i, \quad i = 1 \dots 5$$

same cup · same geometry · same tol_h — held fixed

Look closely: five roots, one continuous, climbing $V(h)$.

What does a single sweep learn that five bisections never see?

Problem

**Model**

Compute



Analyze



Validate



Improve



Five targets — two methods on trial

Target V _i (mL)	Sweep bracket (mm)	Bisection: n · h* ± bound (mm)	Evaluations	Notes
200	___ - ___	n = ___ · ___ ± ___	___	
250	___ - ___	n = ___ · ___ ± ___	___	
300	___ - ___	n = ___ · ___ ± ___	___	
350	___ - ___	n = ___ · ___ ± ___	___	
Lab V _{target}	___ - ___	n = ___ · ___ ± ___	___	known case: measured mark mm

HELD FIXED

same cupsame
tol_h

Your Lab-4 r_b, r_t, H ·
V_{target}

Targets: 200, 250, 300, 350
mL + lab target

...all below the brim-full V(H) —
check first

THEN — ONE PICTURE

Plot V(h) from the sweep (h on the horizontal axis, mL on the vertical) with the five target lines and the five bisection roots marked. Before reading on: does your committed prediction of the cheaper method survive contact with the totals?



ENGINEER DECIDES reuse your Lab-4 f and loop — one extra loop over the five targets; the sweep is your Chapter 2 search at a big step. If you are rewriting bisection, stop and ask why.





Interrogate the comparison — don't assume it

THE COST CHECK

Bisection = $5 \times (n + 2)$ evaluations; the sweep = $H/1 \text{ mm} + 1$ evaluations, once, for all five. Which is cheaper — and by how much?
Write the actual totals next to your committed prediction.

THE BOUND CHECK

The sweep certifies $\pm 0.5 \text{ mm}$ per root (half a step). Does that meet your `tol_h`? What step WOULD — at what cost?
Read it from your table — the data, not your expectation.

MULTIPLICATIVE vs ADDITIVE

Halve the sweep step: cost $\times 2$ for one more halving of the bound. One more bisection halving: cost $+1$ for the same gain.
This asymmetry is the entire story of numerical methods.

THE LANDSCAPE

The sweep drew $V(h)$ and showed ONE crossing per target. Five bisections proved five bounds — and drew nothing.
Job 1 certifies a number. Job 2 draws the landscape.



ENGINEER DECIDES *use your computed evidence — not what you expected to see. The totals decide.*

Problem



Model



Compute



Analyze

**Validate**

Improve



Ask AI for the analysis — then grade it

SUGGESTED PROMPT

“Here is the cup model $V(h) = (\pi \cdot h/3)(r_b^2 + r_b \cdot r_h + r_h^2)$ with my measured r_b , r_t , H , and a sweep of $V(h)$ every 1 mm. For target $V = \underline{\hspace{1cm}}$ mL the sweep brackets the root between $h = \underline{\hspace{1cm}}$ and $\underline{\hspace{1cm}}$ mm. Propose the fastest way to locate the line to ± 0.05 mm from these two sweep points, and state the guarantee behind your bound.”

Any capable AI assistant (your engineering analyst). Record its method and its claimed bound concisely. Do NOT copy the full answer.

NOW AUDIT THE AI RESPONSE

- 1 Which theorem backs the claimed bound — a sign change, or an assumption about the curve?
- 2 Check it: bisect from the SAME sweep bracket to ± 0.05 mm — how many halvings, and do the two roots agree?
- 3 Did it invent a geometry value, a bound, or a “nearly linear” assumption you did not certify?
- 4 Verdict: ACCEPT · MODIFY · REJECT — with the contract as evidence.
- 5 What ONE change would make its answer certifiable?



THE KEY DISTINCTION

the curve’s shape.

an interpolated root may be accurate; it is not certified. Bisection’s bound needs only a sign change — interpolation’s error depends on



ENGINEER DECIDES

AI proposes; engineers check the contract behind every bound before signing.

Problem



Model



Compute



Analyze



Validate



Improve



Validate the roots — then choose the method

1

INDEPENDENT CHECK

Verify without repeating the analysis: plug each bisection root back into $V(h)$ — does $V(h^*)$ sit within $\Delta V_{\text{allowed}}$ of its target? For the lab target, compare h^* with the line you MEASURED in Lab 4.

One line of Python or a calculator. Does the gap to the measured line exceed the bound? Then it is model or measurement — name it.

2

ENGINEERING JUDGMENT

State a defensible conclusion: for placing five marks on this product line, which method do you recommend — bisection per target, the sweep, or sweep-then-bisect — and at what cost?

Say what each method certifies and what it does not. “Best method” is a property of the method AND the question AND the budget.

You are NOT asked to derive interpolation-error theory — identifying the missing contract is the deliverable.

A GUARANTEE HOLDS ONLY WHERE ITS CONDITIONS HOLD — and no method validates the cup model. Validation does.

Problem

›

Model

›

Compute

›

Analyze

›

Validate

›

Improve



One compact report — 10 points

LAB EXTENSION CHALLENGE REPORT

separate from your main Lab-4 report · ½–1 page

- 1 **Committed predictions — halvings, sweep cost, the shortcut**
- 2 **Inputs — geometry, V_{target} , tol_h ; targets below $V(H)$**
- 3 **Five-target table — brackets · n , $h^* \pm \text{bound}$ · totals**
- 4 **$V(h)$ plot with the five targets and roots**
- 5 **Cost & bound comparison — multiplicative vs additive**
- 6 **AI shortcut — claimed bound, your audit + verdict**
- 7 **Known-case validation — $V(h^*)$; lab target vs measured line**
- 8 **Final method judgment — what each certifies; recommendation**

No formal Intro / Methods / Results / Discussion structure required.

Recommended Challenge Rubric	Points
Committed predictions (before computing)	1
Five-target bisection table with certified bounds	3
Sweep comparison — cost and what each method buys	2
AI shortcut audit — ACCEPT / MODIFY / REJECT with evidence	2
Known-case validation	1
Final method judgment	1
TOTAL :	10

Serious enough to take seriously — small enough that Lab-4 still belongs to the main lab.

Problem

>

Model

>

Compute

>

Analyze

>

Validate

>

Improve

RUBRIC for the Lab Extension Report

FUNDAMENTALS OF AI-AUGMENTED ENGINEERING

CHAPTER 4 / LAB-4

Lab Extension Challenge Rubric

One Cup, Five Targets — Choose the Method, Audit the Shortcut · Individual Assessment

Suggested Assessment · 10 points

Used by the instructor and, where applicable, by an AI grading agent. Assessment rewards method judgment, certified bounds, honest cost accounting, and evidence-based auditing of an AI shortcut.

Criterion	What full credit looks like	Pts
Committed predictions (before computing)	Halving-count and sweep-cost predictions committed and left unedited.	1
Five-target bisection table with certified bounds	One target per row: bracket, halvings, root ± half-width, all at the lab's requirement-derived tolerance; evaluation total counted honestly (endpoint checks included).	3
Sweep comparison — cost and what each method buys	The 1-mm sweep run once for all targets; its per-target brackets and ±0.5 mm resolution compared with bisection's bounds and cost; the multiplicative-vs-additive asymmetry named.	2
AI shortcut audit — ACCEPT / MODIFY / REJECT	The interpolation claim audited against the contract: which theorem, if any, backs its “guaranteed” bound; a verdict with evidence, and the MODIFY that would make it certifiable.	2
Known-case validation	The lab target's computed height compared with the measured mark from the lab; the gap explained as model/measurement, not numerical.	1
Final method judgment	Bisection certifies one number; the sweep draws the landscape; the choice defended for THIS job — with the caveat that no method validates the model.	1
TOTAL		10