

FUNDAMENTALS OF AI-AUGMENTED ENGINEERING: A CONVERSATIONAL APPROACH

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

© 2026 AI First Engineering

Problem

›

Model

›

Compute

›

Analyze

›

Validate

›

Improve

The engineering workflow you will use throughout this book

by Taner & Sebnem Onsay,

CHAPTER - 4

PART 4 · ≈ 45 MIN

Trust It — Guarantees Have Conditions

The bound is mathematically guaranteed. The engineering still isn't. What the bracket promises — and what it never can.

4

Problem



Model



Compute



Analyze



Validate



Improve



One shape, one trap, one hunt — one question left

ONE SHAPE

$f(x) = 0$ — the engineering question written so a method can hunt it. The sign of f carries the meaning.

THE TRAP

A certified sign change on a continuous f : the root cannot escape the bracket. Two evaluations buy a theorem's protection.

THE HUNT

Bisection: probe the middle, keep the half with the sign change. Report midpoint $\pm$ half-width — a bound you can point at.

THE SCHEDULE

Accuracy priced in advance: each halving doubles it. Thirteen model evaluations did what two thousand did — with a guarantee.

One question left — Chapter 3's question, aimed at a theorem: when does this guarantee deserve to be trusted? Two cases follow. Neither has an automatic answer.

Problem



Model



Compute



Analyze



Validate



Improve



It converges beautifully. It is nonsense.

```
root_finder.py - as received. It runs.

a, b = 25.0, 40.0    # the interval we were given
tol = 0.0001

# no sign-change check...
while (b - a) / 2 > tol:
    m = (a + b) / 2
    if f(a) * f(m) <= 0:
        b = m
    else:
        a = m
print("Root found:", (a+b)/2,
      "+/- 0.0001 (guaranteed)")

Root found: 39.99994277954102 +/- 0.0001 (guaranteed)
```

1 · PREDICT The interval is [25, 40]. Anything suspicious?
Commit before the run.

2 · RUN “Converged” in 17 iterations. Tolerance met. The language of a theorem.

3 · INSPECT $f(25) = +19.6$ and $f(40) = +104.3$ — NO sign change.
The loop just halved its way to an endpoint. No root was ever inside.

4 · EXPLAIN The guarantee's contract was never checked — one missing assert. And physics screamed: $d_{\text{stop}}(40) = 154$ m on 50 m of road. 144 km/h, certified “safe.”

RUNS + CONVERGES + “GUARANTEED” ≠ CORRECT — the ladder grows its Chapter 4 rung.

The known-case test catches in five seconds what the convergence log never will. Chapter 2's cheapest tool is still the sharpest.

Problem



Model



Compute



Analyze



Validate



Improve



Read the conditions before you trust the promise

① A CERTIFIED SIGN CHANGE

$f(a) \cdot f(b) < 0$ — checked in code, one assert, every time. The demo's disaster was one missing line.

② CONTINUITY OF f

A property of the model, not the program. You certify it by understanding f — the computer cannot do this one for you.

③ THE RIGHT ROOT

“At least one root” is not “the root the decision needs.” A plot or the physics tells you which crossing you trapped.

WHAT THE CONTRACT NEVER COVERS

The model. The parameters. The units. The measurements. A certified bracket on a wrong model converges — beautifully — to the wrong problem's answer. The guarantee is about the computation, never about the engineering.

Engineers read contracts — especially the ones their tools sign. Verify the conditions. Then enjoy the guarantee — for exactly what it covers.

Problem



Model



Compute



Analyze



Validate



Improve

A GUARANTEE HOLDS ONLY WHERE ITS CONDITIONS HOLD

RUNS $\neq$ CORRECT — Chapter 2. CONVERGED $\neq$ CORRECT — Chapter 3.
And now: mathematics only promises what its conditions deliver —
and checking the conditions is engineering work.

CONVERGED $\neq$ CORRECT still reigns above today's theorem. Feed a certified bracket a wrong reaction time, a wrong deceleration, km/h where it expects m/s — it converges all the same, without complaint. The bracket strengthens the numerical rungs of the validation ladder. The model rungs still demand what they always have: independent evidence.





The Chapter 4 hunt

The AI chose the method — and claims a guarantee. **Audit the contract behind the confidence.**

In Chapter 1 you hunted a wrong formula. In Chapter 2, the algorithm itself. In Chapter 3, an unsupported accuracy claim. Now the AI invokes a real theorem — and the question is whether its conditions were ever checked. This is the most convincing wrong answer yet: the mathematics is genuine; only its application may be false.

Problem



Model



Compute



Analyze



Validate



Improve



Live demonstration — the class audits the guarantee

THE DEMO — RIGHT NOW

We ask an AI, casually:

“Find the maximum safe speed for a vehicle with 50 m to stop. Use a method with a guaranteed error bound, and give me the answer to ± 0.001 m/s.”

It will produce polished code — often bisection — and a confident bound. We run it. Then the class audits the contract, not the syntax.

Did it certify the sign change — or assume the interval?

Continuity: did it ask anything about f at all?

Which crossing did it trap — the one this decision needs?

Does the claimed $\pm$ actually equal the final half-width?

Known case: does its root give $d_{\text{stop}} \approx 50$ m?

Who owns ± 0.001 — a requirement, or the AI's habit?

Verdicts: ACCEPT · MODIFY · REJECT — with evidence. A guarantee that survives your audit is one you may use — and you state exactly how you verified it. That, too, is engineering.



Three jobs, one model — choose the method for each

JOB 1 · CERTIFY ONE NUMBER

“Is 20 m/s safe, with margin, to ± 0.05 ?” One crossing, one spec, a bound for the report.

Your call — and your defense?

JOB 2 · DRAW THE LANDSCAPE

A design study: how does $v_{\max}$ move as the available road varies from 30 to 70 m? You need the curve, not a point.

Which tool draws landscapes?

JOB 3 · A BRAND-NEW f

A function nobody on the team understands yet. No bracket, no plot, no intuition — yet.

What comes first — and second?

Reveal — after the argument: bisection certifies (Job 1) · the sweep draws landscapes (Job 2) · reconnaissance then bisection (Job 3). “Best method” is a property of the method AND the question AND the budget.

CHAPTER 3's SUFFICIENCY LESSON, PROMOTED

Then it judged answers; now it judges algorithms. The march was never the villain — unexamined use of any method is.

Problem



Model



Compute



Analyze



Validate



Improve



CAN YOU DO BISECTION WITH WATER?

Cross it. Trap it. Halve it. Trust it. You have seen bisection on a screen — now you will hold it in your hands.

THE MOMENT · FIRST CROSSING

BEFORE BELOW AFTER ABOVE

THE TARGET IS TRAPPED.

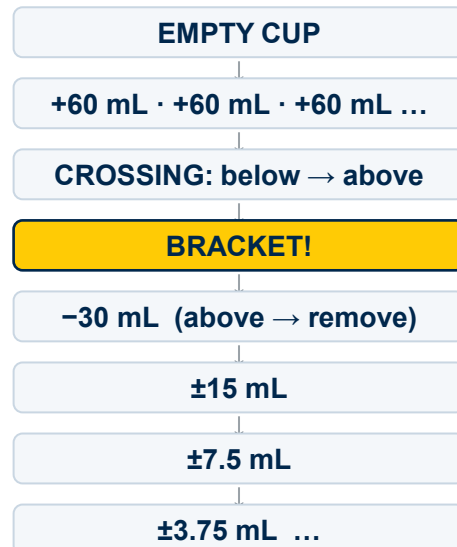
A crossing is information. Use it.

TWO SYRINGES

60 mL 10 mL

ALGORITHM: halve again.
INSTRUMENT: can I resolve that?
Resolution — not a rule — decides the switch.

PHYSICAL · VOLUME (mL)



*How many +60s? Unknown — that is the search.
After -30, the observed sign picks ADD or REMOVE.*

CHAPTER 4, IN A CUP

- WRITE IT** $f(h) = V(h) - V_{\text{target}}$
- TRAP IT** Below + Above → bracket
- HUNT IT** Halve → observe → keep the correct half
- TRUST IT** What exactly can you guarantee?

PHYSICAL mL: 60 → 30 → 15 → 7.5 → ...

COMPUTATIONAL h [mm]: $[a, b] \rightarrow \text{midpoint} \rightarrow \text{keep half} \rightarrow \text{repeat}$

Same logic. Different variable. Different certificate.

TOLERANCE MET → SUCCESS
INSTRUMENT LIMIT FIRST → REPORT THE LIMITATION
Not the same outcome. Only one of them is the contract kept.

IN LAB 4 YOU WILL ① physically bracket and halve · ② model the same cup · ③ compute the root · ④ certify the numerical bound · ⑤ validate against the real cup · ⑥ decide what you actually trust

The mathematical interval can keep shrinking. The physical experiment eventually meets a resolution limit. **So what can you actually guarantee?**
A guarantee is a contract — and Lab 4 asks whether yours survives contact with the real world.

Problem



Model



Compute



Analyze



Validate



Improve

What must be true before the laboratory

A THOUSAND QUESTIONS, ONE SHAPE:
 $f(x) = 0$

THE BRACKET IS THE GUARANTEE

ACCURACY BY DESIGN — NOT BY EXHAUSTION

The engineer writes the equation, certifies the trap, schedules the accuracy,
reads the contract behind every guarantee —
and still validates the root against reality before deciding.

Ahead: bisection is safe because it is humble — it uses only the SIGN of f and throws the value away. But look once more: $f(20.0) = -1.43$ and $f(20.5) = +0.52$. The right wall is almost three times closer to zero — the values are pointing at the root. Chapter 5 asks: what happens when we let the values steer? How much faster can the hunt get — and what new ways to fail arrive with the speed? The same crossing, one more time: **faster, cleverer, riskier.**

Problem

›

Model

›

Compute

›

Analyze

›

Validate

›

Improve

AI = Engineering Analyst: “Audit the guarantee with me — do not just hand me a root.”



What we established in this Chapter Root Finding & the Certified Answer:

Engineering questions become $f(x) = 0$ — one shape, many problems.

A sign change on a continuous f traps a root: the bracket is the guarantee.

Bisection halves the bracket: $\text{error} \leq \text{half-width}$, on a schedule.

Guarantees have conditions — engineers check the contract.

ACCURACY BY DESIGN, NOT BY EXHAUSTION · CONVERGED $\neq$ CORRECT still reigns.



AI COMPANION PROMPT: Certified-Root Prompt:

“Here is my $f(x) = 0$, my proposed bracket, and the accuracy my decision requires. Before any code: check my sign-change evidence, tell me exactly what bisection's guarantee will and will not cover here, and work out how many iterations my requirement implies. Then we run it — and I will verify the result against a known case.”



ENGINEER'S CHECK BOX:

Claiming a root? Show the bracket and both signs.

Claiming a bound? Show the final width it comes from.

Chose a tolerance? Name the requirement behind it.

Guarantee satisfied? The model still needs validating.



AI COMPANION PROMPT: Go One Step Deeper Prompt:

“Construct a case where bisection runs happily and returns a wrong or useless answer. Do not tell me the flaw — let me find the violated condition myself, then question me until I can state the guarantee's exact contract from memory.”