

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 - 5

PART 3 · ≈ 50 MIN

When Newton Goes Wrong

The same loop, the same f — and a start nobody justified. Newton converges beautifully to a root that is not the answer, and flies off the map from a flat spot. Fast has a price.

3

Problem

›

Model

›

Compute

›

Analyze

›

Validate

›

Improve



PREDICT

**Same loop, same f — start $v_0 = -10$ m/s.
Nobody would choose it. A program with a
default start would.**



An AI told “just solve $f(v) = 0$ ” might. Predict, in writing: does Newton converge to 20.37? diverge? stop with an error? something else? Commit before we run it.

Hint: think about the SHAPE of $f(v) = v + v^2/14 - 50$ for negative v . What does the tangent at -10 point toward?

Problem



Model



Compute



Analyze



Validate



Improve



It converges beautifully. It is not the answer.

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

def f(v): return v + v*v/14.0 - 50.0
def fp(v): return 1.0 + v/7.0
v = -10.0          # a start nobody justified
tol = 0.0001
for n in range(1, 50):
    dv = -f(v) / fp(v)
    v = v + dv
    if abs(dv) <= tol:
        break
print('Converged in', n, 'steps:',
      'v =', round(v, 4), ' f =', round(f(v), 6))
Converged in 7 steps: v = -34.3679  f = 0.0
```

1 · PREDICT The start is -10 m/s. Suspicious? Commit before the run.

2 · RUN “Converged” in 7 steps; correction below 0.0001; $f = 0$ to six decimals. The language of success.

3 · INSPECT $f(-34.37)$ really is zero — a genuine root of the parabola. Not a speed a car can have.

4 · EXPLAIN f has TWO roots. Newton finds the one its start slides toward — the tangent at -10 pointed left. No line of code was wrong.

RUNS + CONVERGES + $f = 0 \neq$ THE ANSWER — the ladder grows its Chapter 5 rung.

The domain check catches in one line what the convergence log never will: is v^* a speed a car can have? Chapter 2’s physical-sense test, again.

Problem



Model



Compute



Analyze



Validate



Improve



Three ways the tangent lies

A FLAT SPOT

Where $f' \approx 0$ the tangent is nearly level and its zero is far away: the correction $-f/f'$ explodes. At $v = -7$ the slope is exactly zero — the step is undefined. At -6.9 the tangent lands at $+3,738$ m/s.

THE WRONG ROOT

Several crossings, one answer. Newton finds the root its start slides toward — and reports it with the same confidence. Which crossing the decision needs is engineering, not mathematics.

OUT OF THE WORLD

An iterate outside the physical domain — a negative speed, a load past the end of a beam — is mathematically generated and physically inadmissible. A method has no idea where the world ends. You do.

YOUR $f(v)$, CHECKED — NOT ASSUMED

For $v > 0$ the slope $1 + v/7$ never drops below 1 and f climbs the whole way: from ANY physical start Newton converges to 20.37 ($0 \rightarrow 5$ steps, $40 \rightarrow 4$). The traps live exactly where the physics never sent you — which is where a default, a typo, or an AI might.

CHAPTER 5 · PRINCIPLE No. 2

NEWTON CONVERGES TO A ROOT — YOU DECIDE WHETHER IT IS THE ANSWER.

Problem



Model



Compute



Analyze



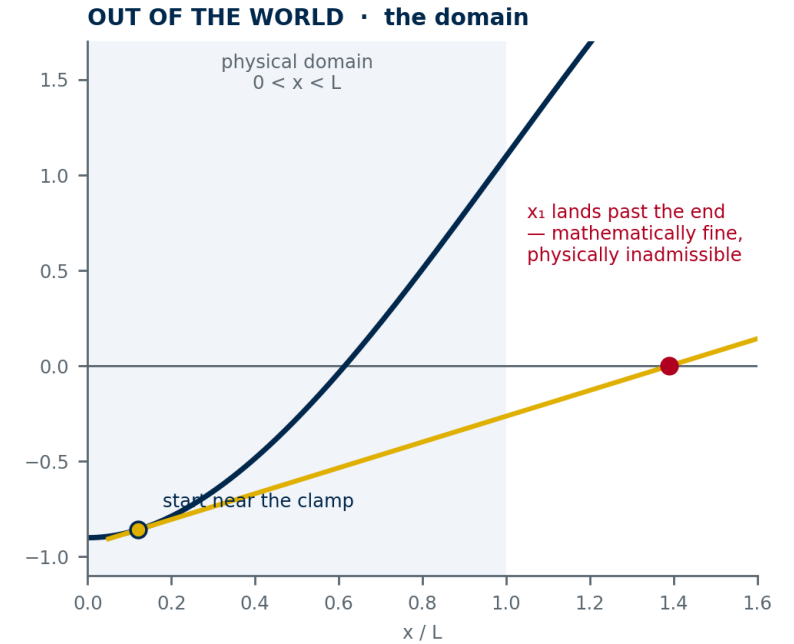
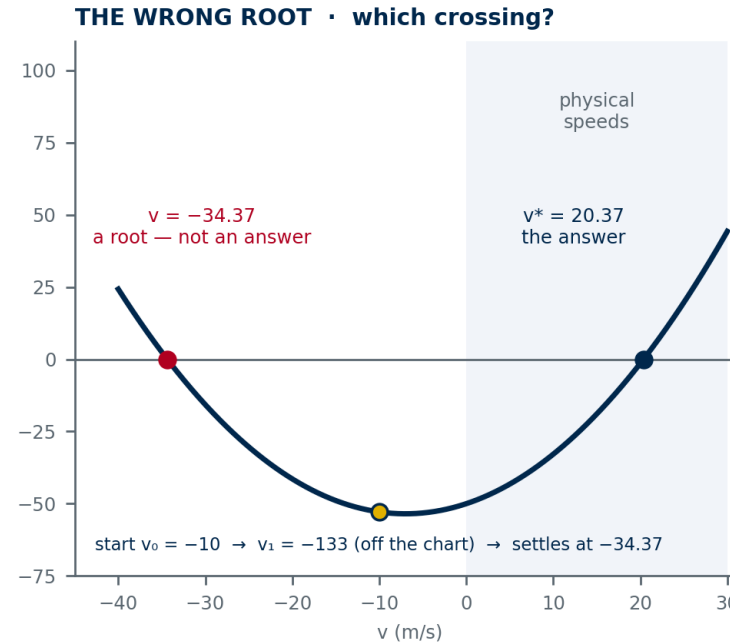
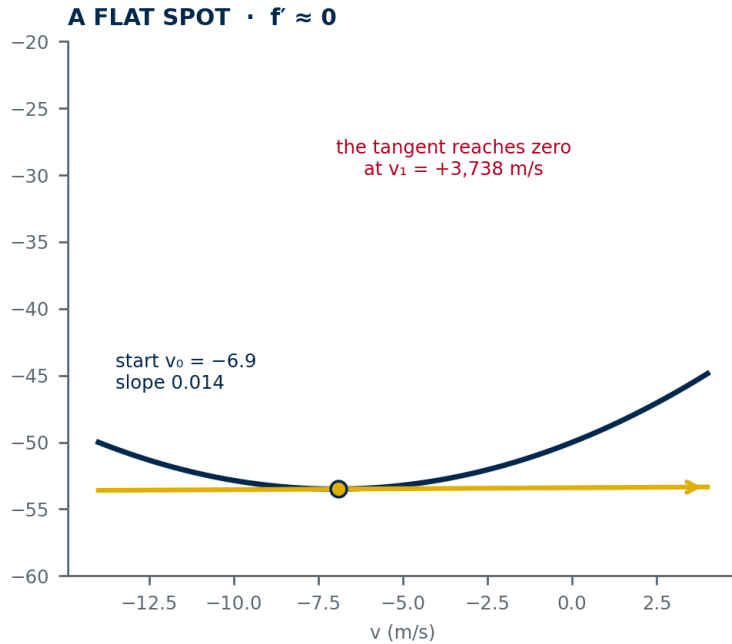
Validate



Improve



Where the tangent stops telling the truth



Same loop, three landscapes. Left: a flat slope throws the step off the chart. Middle: two genuine roots, and the start picks one. Right: the step lands where the world ends.



From $v_0 = -6.9$: off the map in one step, home in twelve

n	v_n (m/s)	$f(v_n)$ (m)	$f'(v_n)$	correction (m/s)	v_{n+1} (m/s)
0	-6.9	-53.50	0.014	+3,745	3,738
1	3,738.1	1,001,761	535.007	-1,872	1,866
2	1,865.6	250,427	267.518	-936	930
3	929.5	62,593	133.787	-468	462
12 steps to $ \text{correction} < 0.01$ — it converged to 20.368 in the end. Was that Newton's virtue, or the parabola's mercy?					
11	20.368	0.0000	3.910	-0.0000	20.370

READ ROW 0

Slope 0.014, correction +3,745 m/s: the tangent reached zero eleven times the speed of sound away. Every later row halves the iterate back.

READ THE RECOVERY

It came home because a parabola has nowhere else to go on the right. A curve with a hump, or a bounded domain, would not be so kind. Twelve steps, no guarantee.

A guard is not a fix. A domain check would have stopped this at row 0 — and told you the start was the problem. That is what guards are for: refusing, loudly, so the engineer thinks.

Problem



Model



Compute



Analyze



Validate



Improve



Newton with guards: the same loop, three refusals

● ● ● guarded newton — PSEUDOCODE

```
GET  $x_0$  inside the domain    # justified
GET tol, max_steps
REPEAT
  IF  $|f'(x_n)|$  tiny  $\rightarrow$  STOP "flat slope"
  SET  $x_{n+1} = x_n - f(x_n) / f'(x_n)$ 
  IF  $x_{n+1}$  outside domain  $\rightarrow$  STOP "left the world"
  IF steps > max_steps  $\rightarrow$  STOP "not converging"
UNTIL  $|x_{n+1} - x_n| \leq \text{tol}$ 
CHECK  $f(x_{n+1}) \approx 0$  · known case · physics
REPORT  $x_{n+1}$  — as evidence, not as a bound
```

WHAT A GUARD DOES

It refuses to continue — and says why. A flat slope, an iterate off the map, a step count exhausted: each one is information about your START or your MODEL, delivered before a wrong answer can look like a right one.

WHAT A GUARD NEVER DOES

It never certifies. A run that trips no guard has proved only that no guard tripped. The wrong-root run on slide 4 passes every guard here — because -34 m/s is a root, and the domain check is YOUR line to write.

Chapter 4's assert refused to hunt without a bracket. Newton's guards refuse to keep walking off the map. Neither replaces the engineer's validation.

Problem



Model



Compute



Analyze



Validate



Improve

CONVERGED FAST IS NOT CONVERGED RIGHT

RUNS $\neq$ CORRECT — Chapter 2. CONVERGED $\neq$ CORRECT — Chapter 3.

A guarantee holds only where its conditions hold — Chapter 4.

And now: Newton's convergence comes with no guarantee at all — the start, the slope and the domain decide which root, and whether.

A collapsing correction is evidence that the iterates have settled — on some root of some model. It says nothing about which root the decision needs, nothing about the physical domain, and nothing about whether the model is right. Validation is the only certificate Newton ever gets: $f(x^*) \approx 0$ in engineering units, x^* inside the world, a known case, the physics. The bracket strengthened the numerical rungs of the ladder; the sprint strengthens none of them — it only arrives sooner.

Problem

›

Model

›

Compute

›

Analyze

›

Validate

›

Improve



The Chapter 5 hunt

The AI chose the method, the start and the stop — and reports a root to twelve digits. Audit the convergence behind the confidence.

In Chapter 1 you hunted a wrong formula. In Chapter 2, the algorithm. In Chapter 3, an unsupported accuracy claim. In Chapter 4, a real theorem applied outside its conditions. Now the AI's mathematics may be flawless AND its root may be genuine — and the question is whether the root is the answer, whether the start was justified, and what a twelve-digit residual actually certifies. The most convincing wrong answer yet: everything checks, except the engineering.

Problem



Model



Compute



Analyze



Validate



Improve



Live demonstration — the class audits a “converged” root

THE DEMO — RIGHT NOW

We ask an AI, casually:

“Use Newton’s method to find the maximum safe speed for a vehicle with 50 m to stop. Give me the exact answer.”

It will produce polished code, a start, a derivative and a root with many digits. We run it. Then the class audits the convergence, not the syntax.

Which start — and who justified it?

Was f' derived from OUR model — check it at one point.

Did every iterate stay inside the physical domain?

Did it stop on a tolerance the requirement owns — or a count?

Known case: does $d_{\text{stop}}(v^*) \approx 50$ m? Is v^* a speed a car can have?

“Exact”? What does a 10^{-12} residual actually certify?

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



Four questions for any converged Newton run

① IS IT A ROOT?

Plug it back: $|f(x^*)|$ in engineering units — metres of road, millimetres of deflection. A small correction is not a small residual; check both.

② IS IT THE ROOT?

Inside the physical domain? The crossing the decision needs? A plot, the physics, or a bracket tells you — the loop cannot.

③ ANOTHER START?

Run it again from a second justified start. Agreement is evidence; disagreement is a finding — several roots, or a flat spot between them.

WHAT CONVERGENCE NEVER COVERS

The model. The parameters. The units. The derivative's correctness. The measurements. A Newton run on a wrong model converges — faster than ever — to the wrong problem's answer. Convergence is about the computation; correctness is about the engineering.

Fast is not certified. Ask the four questions, then enjoy the speed — for exactly what it delivered.

Problem



Model



Compute



Analyze



Validate



Improve