

FUNDAMENTALS OF AI-AUGMENTED ENGINEERING: A CONVERSATIONAL APPROACH

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

© 2026 AI First Engineering



The engineering workflow you will use throughout this book
by Taner & Sebnem Onsay,

CHAPTER - 3

PART 3 · ≈ 45 MIN

Step Size, Resolution & the Cost of Accuracy

One line of your own Chapter 2 code decides how good the answer can possibly be — and what it costs. Time to turn that knob on purpose.

3

Problem

›

Model

›

Compute

›

Analyze

›

Validate

›

Improve



The knob you already own: step

```
●●● max_safe_speed.py - Chapter 2, verbatim

t_r, a, d_avail = 1.0, 7.0, 50.0

v = 0.0          # start slow (m/s)
step = 0.5       # <-- the knob

while stopping_distance(v + step, t_r, a) \
    <= d_avail:
    v = v + step

print("Max safe speed:", v, "m/s")

Max safe speed: 20.0 m/s
```

READ IT LIKE AN ENGINEER

Chapter 2 told you: “step = 0.5 sets the resolution of the answer. Hold that thought.” **This Part is the held thought.**

Nothing about this program is new. What is new is the question we ask it.

One number changes between every case today: step. Everything you are about to observe traces back to this one line.

An investigation, not a programming lesson. You built this machine in Chapter 2. Today you study how it behaves.

Problem



Model



Compute



Analyze



Validate



Improve



PREDICT



Cut the step from 0.5 to 0.1 m/s. What happens to the reported $v_{\max}$?

Does the error shrink by exactly 5×? And — can a smaller step ever leave the answer unchanged? Write all three predictions down.

You built this machine. You should be able to predict it. (Your Part 2 calculation already whispered one of the answers.)





Excel: the step study, one row per experiment

step (m/s)	reported v_max	E_abs (m/s)	E_%	model evaluations
1.0	20.0	0.368	1.81 %	21
0.5	20.0	0.368	1.81 %	41
0.1	20.3	0.068	0.33 %	204
0.05	20.35	0.018	0.088 %	408
0.01	20.36	0.0079	0.039 %	2,037

READ THE TABLE

Each row is a complete numerical experiment — same model, same road, one knob changed.

v_max climbs toward 20.37 and stalls. Error falls. # evaluations explode.

Now look at the first two rows.

$v_{ref} = 20.368 \text{ m/s}$ · $t_r = 1.0 \text{ s}$ · $a = 7 \text{ m/s}^2$ · $d_{avail} = 50 \text{ m}$ — every column serves the question

Step 1.0 and step 0.5 report the SAME answer. Refining 2× bought nothing. Why?

Problem



Model



Compute



Analyze



Validate



Improve



The search only sees its own grid

STEP = 0.5 — THE GRID THE SEARCH CAN SEE



STEP = 0.25 — A FINER GRID SEES MORE



■ $v_{ref} = 20.368$ — the crossing lives between grid points, invisible to both coarse grids

WHY 1.0 AND 0.5 TIED

20.0 sits on both grids, and the crossing lies within one step above it in both. Same last-safe point → same report.

THE BOUND — FOR THIS SEARCH

The crossing sits between the last safe and first unsafe grid point, so this incremental search's error is bounded by about one step. That bound belongs to THIS algorithm — not to every numerical method.

WHY NOT EXACTLY 5×

Where the crossing falls inside the grid sets the actual error. The bound shrinks with the step; the error, only roughly.

Error is bounded by the step — not proportional to it. “Halve the step, halve the error” is dead. What survives: the smaller the step, the finer the grid — and the tighter the bound.



Resolution: the finest distinction the computation can make

RESOLUTION

The finest distinction a procedure can make or report. For the incremental search: one step — the answer is only known to within one grid interval.

CHAPTER 1 RETURNS — SAME WORD, NEW HOME

Your timer could not split the four ball-drops — its resolution hid the difference. **Numerical resolution is the same idea, with the instrument's tick replaced by the algorithm's grid.**

RESOLUTION IS NOT ACCURACY

Finer resolution sharpens what the computation can claim. Whether the claim is TRUE still depends on the model, the parameters, the units, the measurements. **Hold that — Parts 4 and 5.**

Problem ›

Model ›

Compute ›

Analyze ›

Validate ›

Improve



The sweep: your Chapter 2 pieces, one FOR loop

```
step_study.py

def stopping_distance(v, t_r, a):      # Chapter 2 — unchanged
    return v*t_r + v**2 / (2*a)

def find_v_max(step, t_r=1.0, a=7.0, d_avail=50.0):
    v = 0.0                          # the Chapter 2 search,
    while stopping_distance(v + step, t_r, a) <= d_avail:
        v = v + step                 # now a named task
    return v

for step in [1.0, 0.5, 0.1, 0.05, 0.01]:
    print(f"step {step:<5} v_max {find_v_max(step):.2f} m/s")

step 1.0    v_max 20.00 m/s
step 0.5    v_max 20.00 m/s
step 0.1    v_max 20.30 m/s
step 0.05   v_max 20.35 m/s
step 0.01   v_max 20.36 m/s
```

EVERY LINE IS A CHAPTER 2 IDEA

FOR — a list you can count (the crates).

WHILE — a condition decides (the search).

A function — a named, testable task.

PREDICT before you run: what will the 0.05 row print? Your Excel table already knows.

The print format shows two decimals — matched to this study's error evidence, not to a rule. Precision is a claim, in code too.

You are not learning a program. You are investigating one you already own.

Problem



Model



Compute



Analyze



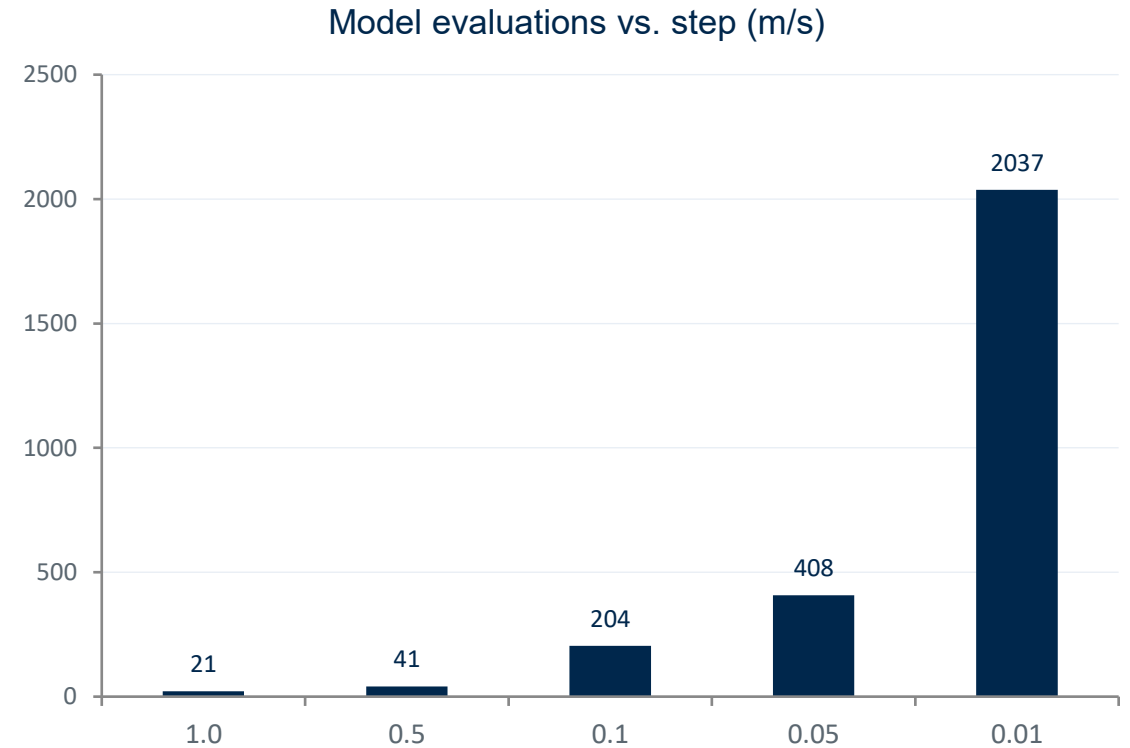
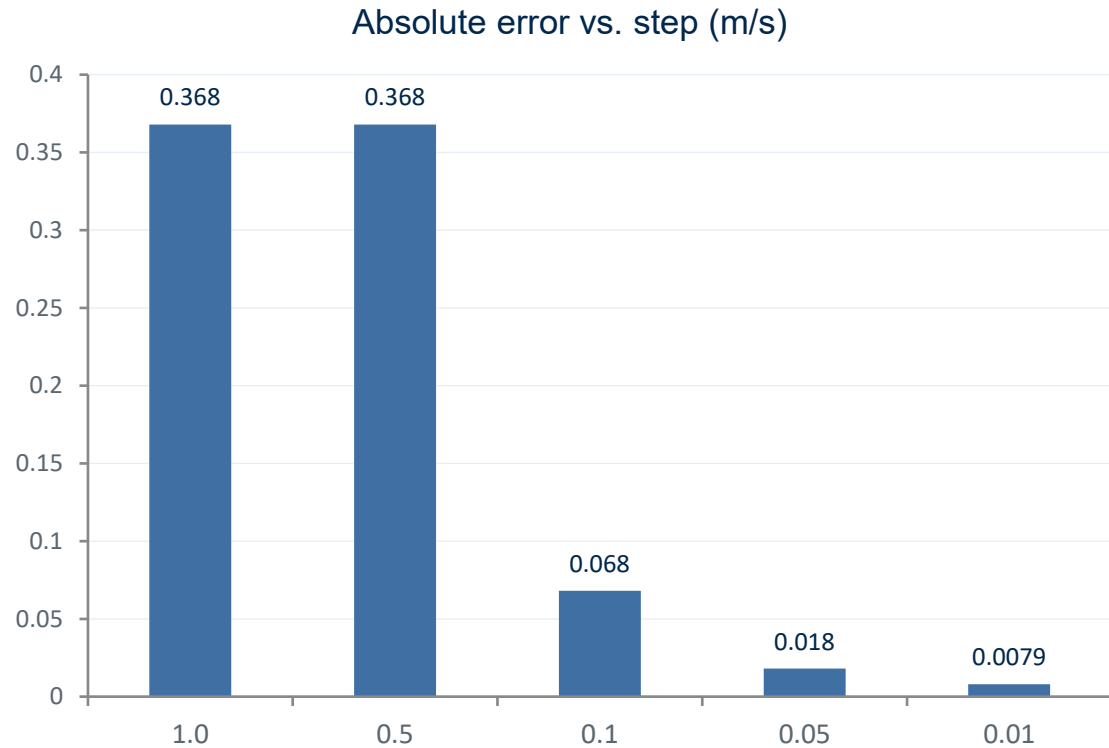
Validate



Improve



For this search, finer resolution costs more computation



Here, reducing the step improves the resolution — and multiplies the evaluations. The broader principle: more accuracy often has a cost — so ask whether the decision needs it.

Problem



Model



Compute

**Analyze**

Validate



Improve



Is the next decimal worth ten times the computation?



FOR THIS SEARCH: 10× FINER, $\approx$ 10× THE WORK

For this incremental search over the same range, a 10× smaller step needs $\approx$ 10× the model evaluations — smaller steps, same road. Milliseconds here; in a crash simulation or a CFD run, a similar refinement trade is **hours of compute and real money**.

THE ENGINEERING QUESTION

Not “can we refine?” — we always can. But: **“does the DECISION need what the refinement buys?”** Answering that needs a requirement — exactly where Part 4 begins.





AI begins to analyze — you still own the judgment



AI MAY...

automate the sweep you defined
organize results into a table
suggest an informative plot
describe the pattern it sees
propose explanations for a surprise (the 1.0 / 0.5 tie)
challenge your accuracy claim — when you ask it to



THE ENGINEER MUST...

define the problem and the parameter study
predict the trend before computing
understand the algorithm being swept
verify any number AI produces
interpret the convergence evidence
choose and **justify** the tolerance — and **decide** whether the result is **good enough**

The transition is beginning, not complete. An analyst's report is input to your judgment — never a substitute for it. **AI assists. Human verifies. Engineer decides.**

Problem



Model



Compute



Analyze



Validate



Improve