

FUNDAMENTALS OF AI-AUGMENTED ENGINEERING

CHAPTER 3 / HOMEWORK 3

Engineering Homework

The Basin That Must Be Empty — A Drain-Time Compliance Decision

How Accurate Is Accurate Enough? · Converged ≠ Correct

Suggested Assessment · 20 points

target effort ≈ 65–85 minutes · type all answers and paste the required screenshot directly into this document



AI First Engineering

Figure: AI First
Engineering
(AIFE) course
logo

Taner Onsay, Ph.D. & Sebnem Onsay · © 2026 AI First Engineering ·

AI use is allowed and encouraged — as an emerging Engineering Analyst. But every prediction must be committed in writing before you compute, the screenshot must show YOUR OWN Python session running on your computer — not an AI transcript — and every AI claim you record must carry your own ACCEPT / MODIFY / REJECT verdict. Assessment rewards evidence + reasoning + verification, not polished prose.

THE ENGINEERING CONTEXT — a detention basin that must be empty

An industrial site stores stormwater in a small detention basin — plan area 24 m², filled to a depth of 1.9 m by the design storm. The basin drains by gravity through an outlet fitting in its floor. The municipal drainage permit is blunt: after the storm ends, the basin must be completely empty within 120 minutes, so that its full storage volume is available for the next storm cell. The city owns that number — not the programmer.

An engineer must certify, by numerical simulation, whether the proposed outlet drains the basin within the 120-minute limit. Draining is not steady: deep water pushes hard on the outlet, shallow water barely pushes at all — the flow starts fast and finishes slow, and no single flow rate describes the whole event.

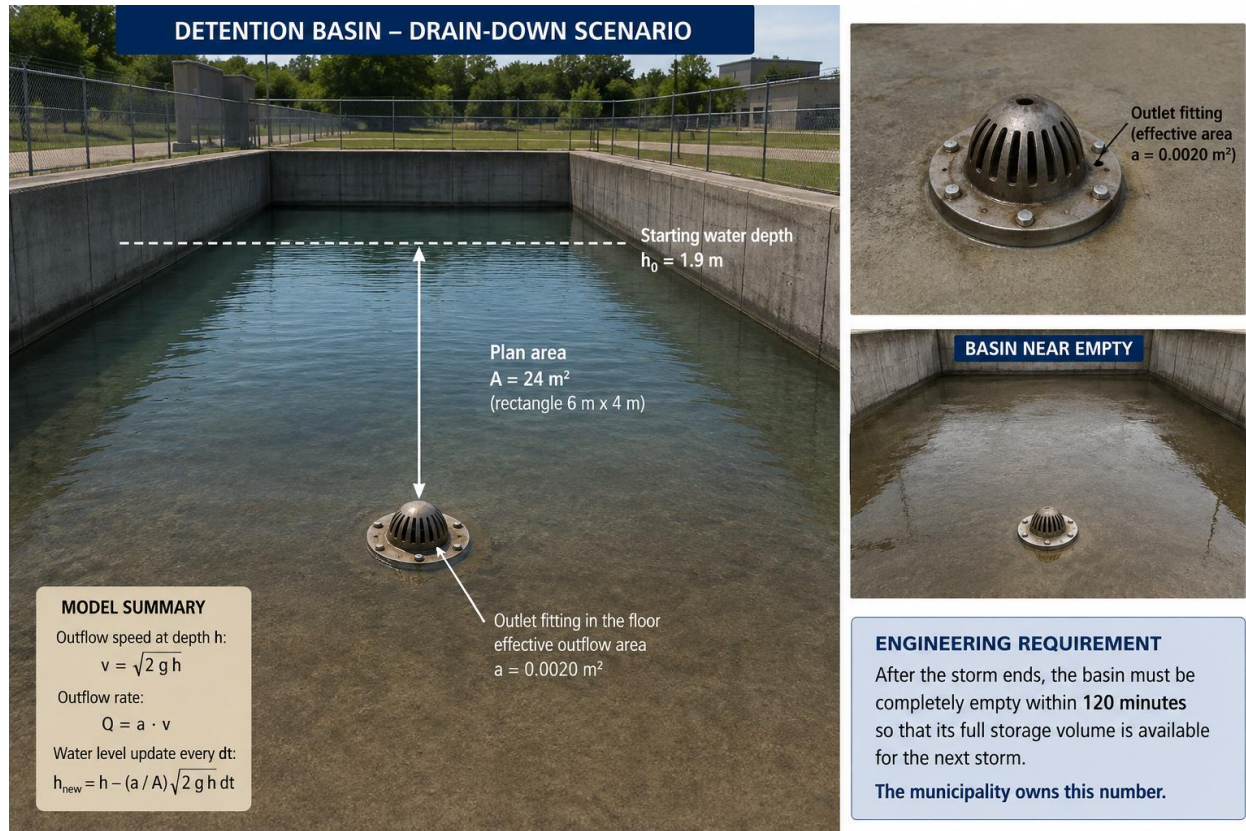


Figure: Detention-basin drain-down scenario for this homework — the stored water at $h_0 = 1.9$ m, the floor outlet, and the municipal 120-minute drain-down requirement. The model box and parameter table below remain the authoritative values for computation.

THE DRAIN-DOWN MODEL — provided; no fluid-mechanics course required

Water leaves the outlet at **the free-fall speed you met in Chapter 1**: at depth h , the outflow speed is $v = \sqrt{2 \cdot g \cdot h}$ — the speed an object would reach falling freely from height h . Deeper water → faster outflow. As the basin empties, the level falls and the outflow slows.

During each short time interval dt , the volume $Q \cdot dt$ leaves through the outlet, where $Q = a \cdot v$ is the outflow rate at the current depth. Spread over the plan area A , the level falls by $(Q / A) \cdot dt$:

$$h_{\text{new}} = h - (a / A) \cdot \sqrt{2 \cdot g \cdot h} \cdot dt$$

h = water depth (m) · A = basin plan area (m^2) · a = effective outflow area (m^2) · $g = 9.81 \text{ m/s}^2$
 dt = time step (s)

The simulation holds the outflow rate constant across each time step, while the real outflow falls continuously — so the computed drain time depends on the dt you choose. a is the outlet assembly's rated effective discharge area (real outlets pass less flow than their raw opening size; the rating accounts for it). The values below are instructional design parameters for this assignment. Everything you compute is a MODEL PREDICTION — not yet field evidence about a physical basin.

Quantity	Symbol	Value	Units
Basin plan area	A	24	m^2

Quantity	Symbol	Value	Units
Effective outflow area of the outlet	a	0.0020	m ²
Starting water depth (design storm)	h0	1.9	m
Gravitational acceleration	g	9.81	m/s ²
Baseline time step	dt	180	s
Compliance drain-time limit (municipal permit)	T_limit	120	min

Time guide: Part A ≈ 10 min · Part B ≈ 8 min · Part C ≈ 10 min · Part D ≈ 15 min · Part E ≈ 10 min · Part F ≈ 15 min · Part G ≈ 8 min · Part H ≈ 6 min. If you are far past 90 minutes, submit what you have with honest notes — do not expand the assignment.

PART A · The Problem & Committed Predictions (≈ 10 min)

Study the figure together with the context, model, and parameter boxes above — observe the physical system before computing anything about it. Then commit — in writing, before any computation. A prediction made after seeing the answer is not a prediction. Wrong is fine; uncommitted is not. Do not edit these later.

A1 — The Problem, in Your Own Words. In one or two sentences: what exactly are you asking the computer to calculate, and what engineering decision depends on the result?

ANSWER A1:

A2 — Prediction P1 · The Two Halves. Which takes LONGER to drain: the top half of the stored water (1.9 m down to 0.95 m) or the bottom half (0.95 m down to empty)? Commit, then give one sentence of physical reasoning.

P1: top half ☐ bottom half ☐ about the same ☐

REASONING P1:

A3 — Prediction P2 · Which Way Does the Simulation Err? The simulation freezes the outflow rate at its start-of-interval value for the whole interval — but the real outflow falls throughout the interval. Will a coarse (large) dt therefore tend to predict a drain time that is too SHORT, too LONG — or is it impossible to say? One sentence of reasoning.

P2: too short ☐ too long ☐ impossible to say ☐

REASONING P2:

A4 — Prediction P3 · Refinement. When dt is made smaller and smaller, do you expect the predicted drain time to increase, decrease, or stay about the same? Commit now — Part D will hold you to it.

P3: increase ☐ decrease ☐ stay about the same ☐

PART B · Own the Model — One Known Case (≈ 8 min)

The program in Part C is provided nearly complete: unlike the last homework, you are not asked to build the algorithm again. This homework is about interrogating the number the program produces. But a provided program is not a black box — before running anything, show that you own the model.

B1 — Quantities and Roles. Complete the Role column: is each quantity an input parameter (set once, never changes), a state variable (changes every pass of the computation), or an output (reported at the end)? Some quantities may serve more than one role — if so, state both.

Quantity	Symbol	Units	Role — your answer
Basin plan area	A	m^2	
Effective outflow area of the outlet	a	m^2	
Gravitational acceleration	g	m/s^2	
Time step	dt	s	
Water depth during draining	h	m	
Elapsed drain time	t	s	

B2 — Why the Draining Slows. In one or two sentences of physics — no equations: why is the outflow fast when the basin is full and slow when it is nearly empty?

ANSWER B2:

B3 — One Update, by Hand. Using the model box and the parameter table, hand-calculate the FIRST time step at $dt = 180 \text{ s}$: the outflow rate at full depth, and the water depth after the first step. Show the arithmetic. Keep both numbers — they are your known case for testing the program in Part C.

ANSWER B3: Q at full depth = _____ m^3/s . Depth after the first 180 s = _____ m. Arithmetic:

PART C · Run the Baseline Simulation (≈ 10 min)

The program below is complete except for the two marked lines. Copy it into your Python environment (IDLE, Spyder, Jupyter, VS Code — any is fine), complete ① and ② using the model box, and run it at the baseline $dt = 180 \text{ s}$. Your B3 hand values tell you exactly what the first pass of the loop must do.

```
# basin_drain.py — Chapter 3: detention-basin drain time
```

```

import math

A, a = 24.0, 0.0020      # plan area (m^2), effective outflow area (m^2)
g, h0 = 9.81, 1.9        # gravity (m/s^2), starting depth (m)
dt = 180.0                # time step (s) <-- the resolution knob

h = h0                    # water depth (m) – the state variable
t = 0.0                    # elapsed time (s)
steps = 0

while _____ :        # COMPLETE ①: repeat while water remains
    Q = a * math.sqrt( _____ ) # COMPLETE ②: outflow rate – model box
    h = h - (Q / A) * dt           # the level falls as volume Q*dt leaves
    t = t + dt
    steps = steps + 1

print("Drain time:", t, "s =", t/60.0, "min  (", steps, "steps )")

```

C1 — Evidence. Paste ONE screenshot showing your completed program together with its printed output, with no unresolved error. This is the only screenshot this homework requires.

PASTE PYTHON CODE / OUTPUT SCREENSHOT HERE

(click here and paste your screenshot — Ctrl+V / Cmd+V)

C2 — Result & Known-Case Check. Report the baseline result and test it against your own B3 evidence. (A temporary `print(h)` as the loop runs is a legitimate instrument — remove it afterward.)

Baseline drain time (dt = 180 s): _____ min **steps:** _____

First pass agrees with your B3 hand values? YES ☐ NO ☐

If NO — what you checked (one sentence):

C3 — The Verdict at This Resolution. Compare with the 120-minute permit limit and record the verdict. Part D returns to it.

At dt = 180 s the basin PASSES ☐ **/ FAILS** ☐ **Margin:** _____ min.

PART D · The Resolution Study (≈ 15 min)

One line of the program — `dt` — decides how good the answer can possibly be. Rerun the program at each time step below (change `dt`, run again; nothing else changes) and complete the table. Then interrogate the numbers BEFORE any reference answer exists — exactly as an engineer must on a real problem, where nobody hands out the true value.

dt (s)	predicted drain time (min)	verdict vs 120 min (PASS / FAIL)	change from previous (min)
600			—
180 (baseline)			
60			
20			
5			

D1 — Read the Trend. Which direction does the predicted drain time move as dt shrinks — and are the changes between successive rows getting smaller? Is this evidence consistent with convergence? Why is it evidence rather than proof?

ANSWER D1:

D2 — The Flip. Did the engineering verdict change inside your table? Between which two rows? If you had stopped after your dt = 180 s baseline — a program that ran perfectly — what would you have certified to the city?

ANSWER D2:

D3 — Reconcile Your Predictions. Return to P2 and P3 and reconcile them with the table. Explain the direction physically: the simulation freezes each interval's outflow at its starting — largest — value, so what does the simulated basin do that the real one cannot? And for THIS decision, is an error that pushes toward PASS the safe side or the unsafe side?

ANSWER D3:

D4 — Resolution, Literally. Your dt = 600 s result is a whole multiple of 10 minutes — it has to be. Why? What does that alone say about how finely that run can possibly know the drain time?

ANSWER D4:

PART E · Error, the Reference & When to Stop (≈ 10 min)

You have interrogated the numerical evidence on its own terms. Only now does an independent reference enter — the order matters, because on most real engineering problems no reference exists, and the convergence evidence you just gathered is all you get.

INDEPENDENT REFERENCE SOLUTION — revealed only NOW

The idealized drain-down model has an exact total drain time. You are NOT expected to derive it; it exists here purely as independent evidence against your simulation:

$$T_{\text{drain}} = (A/a) \cdot \sqrt{(2 \cdot h_0 / g)}$$

One line of Python: `import math; print( (24.0/0.0020) * math.sqrt(2*1.9/9.81) / 60.0 )`

Look inside it: $\sqrt{(2 \cdot h_0 / g)}$ is exactly Chapter 1's free-fall time from the starting depth — about 0.62 s from 1.9 m — and $A/a = 12,000$. The basin's drain time is twelve thousand Chapter-1 ball-drops. An early model has returned as validation evidence.

E1 — Reference drain time = _____ min. Verdict at the reference: PASSES ☐ FAILS ☐

E2 — Measure the Error. Compute a small number of purposeful errors — in minutes, because the requirement itself is stated in minutes:

E_abs at dt = 180 s = _____ min

E_abs at dt = 5 s = _____ min

E_% at dt = 180 s = _____ %

E3 — The Comparison That Matters. Compare the numerical error at dt = 180 s with the margin between the reference answer and the 120-minute limit. Which is larger? Could you responsibly certify PASS or FAIL from the baseline run alone? This comparison — error against decision margin — matters more here than any percent.

ANSWER E3:

E4 — When Is Refinement Sufficient? Who owns the 120-minute requirement — and what does it demand of your numerical accuracy? Using only your change column (imagine the reference did not exist), state the dt at which YOU would have been willing to certify, and defend it in one or two sentences. Guide: the change between successive refinements should be comfortably smaller than the margin separating the predicted drain time from the compliance limit.

ANSWER E4:

What did refinement cost? Your steps column runs from 11 steps at dt = 600 s to about 1,500 at dt = 5 s. For this small program that is milliseconds either way — refinement here is essentially free, and the honest reason to stop is sufficiency, not cost. The principle still matters: in a crash simulation or a CFD model, the same 100× refinement is hours of computing and real money.

PART F · AI Analyst & BREAK THE AI (≈ 15 min)

In earlier homework AI helped you build. Its role now advances: AI analyzes numerical evidence — YOUR evidence — and you judge the analysis. The rule has not changed: AI assists. Human verifies. Engineer decides.

F1 — Hand AI Your Table. Give any capable AI assistant your completed Part D table (retype or paste it), with a prompt close to this:

“Here is my resolution study for a detention-basin drain-time simulation [your Part D table]. Do not tell me whether the basin complies. Describe the trend as dt decreases, propose a physical explanation for why the predicted drain time keeps rising, state whether this evidence supports convergence, and list what additional evidence you would want before certifying compliance against a 120-minute limit.”

Record ONE meaningful claim from the AI analysis, verbatim (one or two lines — not the whole conversation).

ANSWER F1:

F2 — Your Verdict on the Analysis. Judge that claim as the engineer — the verdict now applies to AI analysis, not AI code. Do not manufacture disagreement: ACCEPT is a fully legitimate verdict when independently justified.

ACCEPT ☐ MODIFY ☐ REJECT ☐

WHY (1–2 sentences):

BREAK THE AI — The Compliance Memo. A different AI assistant was asked directly whether the basin complies, and produced the memo below. Every number in it is real — rerun those two time steps yourself if you doubt it. The code is correct. The model is correct. Something else is being sold.

AI COMPLIANCE MEMO — verbatim

“To confirm numerical accuracy, I ran the drain-down simulation at two different time steps. At $dt = 300$ s the basin empties in 115.0 minutes; at $dt = 240$ s it empties in 116.0 minutes. The two runs agree to within one minute, so the numerical solution has converged. Both results sit comfortably below the 120-minute requirement, with four to five minutes of margin. Conclusion: the proposed basin complies with the municipal drainage regulation.”

F3 — Attack the Claim, Not the Code. Using your own Part D and Part E evidence, give at least THREE specific reasons the memo’s engineering claim does not stand. Aim at the accuracy claim: what makes “agree within one minute” weak evidence here? Which direction is this method’s error known to push the verdict? What happened when YOU actually refined?

ANSWER F3:

F4 — Verdict on the Memo. One check, one sentence.

ACCEPT ☐ **MODIFY** ☐ **REJECT** ☐

WHY (one sentence):

PART G · Independent Validation (≈ 8 min)

Convergence evidence and a reference formula are still not the whole validation ladder. Two quick physical checks close it — both reuse numbers you already own.

G1 — A Physical Bound. The basin cannot possibly drain faster than if its initial — maximum — outflow rate continued for the whole event. Hand-compute that fastest-possible time from the starting volume $V_0 = A \cdot h_0$ and your B3 outflow rate: $T_{\text{bound}} = V_0 / Q$. This is a LOWER bound on the drain time: any predicted total drain time below it would violate the physical bound.

$T_{\text{bound}} =$ _____ min. **All simulated results respect this bound:**
YES ☐ **NO** ☐

G2 — The Two Halves, Measured. Make one temporary change: set the loop condition to `while h > h0 / 2:` and rerun at $dt = 5$ s. (Restore it afterward.) This measures the top half of the drain directly.

Top half (1.9 → 0.95 m): _____ min

Bottom half ($dt = 5$ s total – top half): _____ min

Which did P1 predict — and why, physically, is the bottom half so much slower?

G3 — Physical Reasonableness — and the Final Step. From your runs (a temporary print of h inside the loop is a legitimate instrument): does the simulated depth decrease monotonically, most rapidly near the beginning? Now inspect the very last step: does the update overshoot slightly below $h = 0$? If so, explain why that small negative value is a numerical-resolution artifact — it can never exceed one step's worth of outflow, so its size is controlled by dt — rather than a physical prediction about the water.

ANSWER G3:

PART H · CONVERGED ≠ CORRECT — Certify (≈ 6 min)

H1 — What Convergence Does NOT Establish. Your refined simulation and the reference now agree to within seconds. State TWO things this beautiful numerical agreement still does not establish about the real basin. (Think: the measured plan area, the outlet's actual effective area in the field, debris, the idealized model itself.)

ANSWER H1:

H2 — The Engineering Certification. Complete the statement. Every blank matters — and report the drain time to the digits your error evidence justifies, not every digit Python printed.

“Under the stated drain-down model and parameters ($A = 24 \text{ m}^2$, $a = 0.0020 \text{ m}^2$, $h_0 = 1.9 \text{ m}$), my refined simulation predicts a drain time of approximately _____ minutes (at $dt =$ _____ s, with a change of _____ minutes from the previous refinement), against the municipal 120-minute drain-down requirement.

The proposed design PASSES ☐ / FAILS ☐ the requirement. My coarse baseline run supported the opposite verdict; the difference is numerical resolution, not physics. I recommend:

This is a model-based computational conclusion under the stated assumptions — it is not experimental proof about a physical basin, and field verification is required before final acceptance.”

Evidence behind this certification (one line each): ① hand-checked known case ② convergence evidence — the change column ③ physical bound respected ④ independent reference agreement.

EVIDENCE ①:

EVIDENCE ②:

EVIDENCE ③:

EVIDENCE ④:

THE NUMBER WAS NEVER THE PROBLEM.

Once correctly implemented, the same program produced a PASS or a FAIL depending only on its numerical resolution. The code can be correct, the model can be the intended model — and the engineering conclusion can still be premature. Carry the principle into every simulation you will ever sign: a numerical answer is incomplete until you know its accuracy — and “accurate enough” is defined by the engineering decision, not by the computer.

SUBMIT · Evidence Checklist & Human Review (≈ 3 min)

Submit ONE completed Word document — this file — containing all of the following:

- ☐ A1 problem statement · committed predictions P1, P2, P3 with reasoning — unedited
- ☐ B1 roles table · B2 physical explanation · B3 hand-computed known case with arithmetic
- ☐ C1 code + output screenshot · C2 baseline result and known-case check · C3 recorded baseline verdict
- ☐ Part D resolution table (all five rows, PASS/FAIL and change columns) · D1–D4 interpretations

- ☐ E1 reference value and verdict · E2 error values · E3 error-vs-margin comparison · E4 defended stopping point
- ☐ F1 recorded AI claim · F2 verdict on the analysis · F3 three-point attack on the memo · F4 memo verdict
- ☐ G1 hand-computed bound · G2 two-halves measurement · G3 physical-reasonableness & final-step check
- ☐ H1 converged-≠-correct statement · H2 completed certification with all four evidence lines
- ☐ Human review statement signed and dated (below)

SUBMIT ONE COMPLETED HOMEWORK DOCUMENT

Keep all required evidence in this document rather than as separate Python or image files. A separate .py file is welcome as an optional attachment, but the document must stand on its own. If this homework is being used as part of a course, follow your instructor's submission and file-naming requirements.

HUMAN REVIEW

"I reviewed the predictions, calculations, code, screenshot, resolution study, and written responses in this submission and confirm that they represent the work I performed, that I can explain every line of the program, and that I can defend every number in my certification."

Human reviewed and approved: _____

Student name: _____ Date: _____

Suggested Assessment Rubric — 20 points

Used by the instructor and, where applicable, by an AI grading agent. All required evidence must appear inside this document — the grader will not open separate Python or image files. A submission whose program runs, but whose certification is not supported by resolution evidence, cannot earn a high score.

Criterion	What full credit looks like	Pts
Committed predictions & problem framing (A1–A4)	A1 names both the computed quantity and the decision that depends on it. P1–P3 committed, with the requested reasoning sentences. Prediction correctness is NOT graded — commitment and reasoning are.	2
Model understanding & known case (B1–B3)	Roles column correct (which quantities are parameters, which are state, what is reported). B2 gives the physical reason the draining slows. B3 hand values correct with the arithmetic shown — they are checked against the instructor key and against your own Part C run.	2

Criterion	What full credit looks like	Pts
Baseline simulation (C1–C3)	Both marked lines completed correctly; the screenshot shows the completed code and clean output; the reported drain time and step count are consistent with the key; first-pass agreement with B3 confirmed; the verdict at this resolution recorded with its margin.	2
Resolution study & convergence interpretation (D1–D4)	Table complete and internally consistent: five rows, verdicts, and a correctly computed change column. D1 reads the trend and the shrinking changes, and states why this is evidence rather than proof. D2 identifies the row pair where the verdict changes and names what stopping early would have certified. D3 explains the direction of the numerical error physically and calls its safety implication for this decision. D4 explains the whole-multiple-of-dt resolution.	4
Error, margin & sufficiency (E1–E4)	Reference computed and its verdict recorded; absolute errors in minutes and the percent error consistent with the key. E3 makes the error-versus-decision-margin comparison quantitatively and answers the certification question from it. E4 names who owns the requirement and defends a stopping dt whose change is comfortably smaller than the margin.	3
AI analyst & BREAK THE AI (F1–F4)	One verbatim AI claim recorded and judged with independent justification — the verdict now applies to analysis, not code. The memo is attacked with at least three specific, evidence-based reasons drawn from the student's own Part D and Part E results. Verdicts are graded on justification, never on which box is checked.	3
Independent validation (G1–G3)	The physical bound hand-computed and checked against every simulated result. The two halves measured and tied back to P1 with the physical reason. Monotonic, front-loaded draining confirmed from evidence, and the tiny final-step overshoot below $h = 0$ explained as a resolution artifact, not a bug.	2
CONVERGED $\neq$ CORRECT & the certification (H1–H2)	Two legitimate not-established items. Certification complete and internally consistent: a value with justified digits plus its dt and change stated, a verdict consistent with the evidence, a real recommendation, the model-based caveat intact, and all four evidence lines filled.	2
	TOTAL	20