

Damped Pendulum

Generation Prompt & Build Log — the reference simulation for Creating AI-Generated Simulations as a Mode of Scientific Inquiry

Scientific Inquiry with AI Prompt & Build Log

Created by LLNL Summer 2026 STEM Education Research Team

Student Team: Ramina Amino, Jahanvi Chamria, Arya Ferozy, Bryanna Gonzalez, Tai Le, Zedikiah McAdams, John Navarra, Joshua Sarabia, Abdurrahman Raza

Faculty Team: Praveen Pathak, David Rakestraw, David Strubbe, Brian Utter

What This File Is

This document is the paper trail behind Damped_Pendulum_Simulation.html, the reference simulation shipped with this lesson. It shows the exact Phase-1 specification, the sequence of prompts used to generate, validate, refine, and probe the model, and the outcome of each check, so instructors and students can see a complete, honest build history for one worked example before doing the same for their own phenomenon.

Use it three ways: as the live-demo script for the in-class fail-and-validate walkthrough, as the model answer students compare their own change log against, and as the worked example of a Post-Class Extension Activity (Section 9).

1. Phase 1 Specification (Written by Hand, Before Any Generation)

- **Phenomenon:** A point mass on a rigid, massless rod of length L , swinging under gravity, losing energy to velocity-proportional (viscous) damping.
- **Driving question:** How does the swing period depend on how far back you pull the pendulum before release?
- **State variables:** Angle θ (rad), angular velocity $\omega = d\theta/dt$ (rad/s).
- **Parameters, with units:** Length L (m); gravity $g = 9.81 \text{ m/s}^2$; damping coefficient b (1/s); release angle θ_0 (rad or deg).
- **Governing relationship:** $\ddot{\theta} = -(g/L) \sin(\theta) - b\omega$ (the full nonlinear equation). The small-angle approximation replaces $\sin(\theta)$ with θ , giving simple harmonic motion, a shortcut to test for, not assume.
- **Assumptions (what is neglected and why):** Point mass; rigid massless rod (no distributed mass, no rod flex); damping proportional to speed, not the more realistic velocity-squared air drag; motion confined to one vertical plane; constant g .

Validation checks committed to in advance:

- **Check 1**, known/limiting case: with no damping and a small release angle (up to about 10 degrees), the period should equal $T = 2\pi \sqrt{L/g}$. For $L = 1$ m, T is about 2.01 s.
- **Check 2**, conserved quantity: with damping set to zero, total mechanical energy (per unit mass) should stay constant over many swings.

Prompt 0 — Specification critique (Phase 1, before any building)

Here is my model specification for a damped pendulum. Do NOT build a simulation. Critique it only: where are the assumptions incomplete or unstated? Where are units, variables, or governing relationships ambiguous? Which of my validation checks are not truly independent of the model? List what I should add or sharpen before I build.

[specification pasted here — see Section 1 above]

2. Generate

The build prompt below is the Phase-1 specification translated into a single generation request. This is the exact prompt used to produce Damped_Pendulum_Simulation.html.

Prompt 1 — Generate

Build a single self-contained HTML file (inline JS/CSS, no external libraries, runs offline in a browser) that simulates a damped pendulum.

State variables: angle θ (rad), angular velocity ω (rad/s). Parameters with units: length L (m, range 0.2-3), gravity g (m/s^2 , range 1-25, default 9.81), damping coefficient b ($1/\text{s}$, range 0-6), release angle θ_0 (deg, range 1-178). Governing relationship: $\theta_{\text{ddot}} = -(g/L)\sin(\theta) - b\omega$. Use the FULL nonlinear equation ($\sin \theta$), not the small-angle shortcut, unless I explicitly select a small-angle comparison toggle. Controls: sliders for L , θ_0 , damping, g ; a checkbox to switch to the small-angle approximation for comparison; play/pause/step/reset; a speed control. Outputs: an animation of the swinging pendulum; a plot of $\theta(t)$ and of total mechanical energy normalized to its starting value; a live numeric readout of angle, angular velocity, elapsed time, measured period, the small-angle predicted period $T = 2\pi\sqrt{L/g}$, current energy, starting energy, and percent energy drift. Include a reset button and let me set exact starting values via the sliders. Give me the full file to download, not just an in-app preview.

3. Validate Technically

Prompt 2 — Validate technically

List every control and output you implemented. For each, tell me exactly what it should do so I can test it. Do not assure me it works, give me the checklist I use to confirm it myself.

Outcome recorded for this build: all four sliders (L, theta-zero, damping, g) update the animation immediately on drag; play/pause/step/reset behave as specified; the small-angle checkbox toggles the governing equation; the theta(t)/energy plot renders on the second tab; the sweep tab runs and plots period vs. release angle. No missing controls found on this pass.

4. Validate Scientifically

Prompt 3 — Validate scientifically (drive it yourself, AI assists only in locating the number)

I am checking this simulation against a known case. With damping = 0 and theta0 = 10 degrees, for L = 1 m the period should be $T = 2\pi\sqrt{L/g}$, about 2.01 s, because at small angles the equation reduces to simple harmonic motion. Show me how to read the measured period off the simulation. Do not change the code yet, just help me locate the number.

Outcome recorded for this build: measured period at theta-zero = 10 degrees, b = 0, L = 1 m came out to 2.01 s, Check 1 passes (within 1% of the analytic prediction). With damping = 0 at any angle, the energy readout stayed within 0.3% of its starting value over 60 seconds of simulated swinging, Check 2 passes. (The simulation's built-in check panel reports both of these live as PASS/FAIL so students do not have to compute it by hand.)

5. Refine

First build passed both checks on the first generation in this instance, no refinement prompt was needed. The template below is what a real discrepancy would call for, and is the one used with students whose first attempt does not pass:

Prompt 4 — Refine (one targeted fix at a time)

My check found a discrepancy: I expected [2.01 s], the simulation shows [1.78 s]. I think the likely cause is [the integrator step size is too large / the sim silently used the small-angle approximation / g is misapplied]. Make only the change needed to address that cause, and tell me what you changed and why. I will re-run my check.

6. Instrument Mode — Sweep

Prompt 5 — Sweep (instrument mode)

Add a mode that sweeps release angle θ_0 from 5 to 175 degrees and plots the measured period against θ_0 , alongside the small-angle prediction as a flat reference line, so I can see how the true period diverges from the small-angle formula. Keep the single-run view available too.

Outcome / the payoff: sweeping θ_0 shows the true nonlinear period tracking the flat small-angle line closely below about 20 degrees, then climbing, about +0.7% at 20 degrees, roughly +7% near 1 radian (about 57 degrees), and diverging steeply as θ_0 approaches 180 degrees. This is the headline result the equations alone do not make obvious, and it is a direct test of whether a simulation quietly hard-coded $\sin(\theta)$ as θ : a sim that fails this test will show a perfectly flat line.

7. Find the Failure

Prompt 6 — Find the failure

I want to push this model until it breaks. Given my assumptions (viscous, velocity-proportional damping; rigid rod; point mass), at what input values should the model stop being valid? Let me drive the controls to those values, do not prevent it, so I can observe and document the failure.

Two documented failure modes for this build:

- **Overdamped transition.** Past a critical damping value $b = 2 \sqrt{g/L}$ (about 6.26 per second for $L = 1$ m), the pendulum stops oscillating altogether and creeps back to rest without crossing equilibrium. Students find this by sweeping the damping slider and should explain it as the point where the restoring term is overwhelmed by the damping term.
- **Large-angle divergence.** As θ_0 approaches 180 degrees, the period grows without bound (the pendulum takes longer and longer to leave the near-inverted position). A simulation that hard-coded the small-angle shortcut will show a flat, wrong period here, a teachable, catchable discrepancy.

A third, more conceptual failure worth raising with students: real air resistance is closer to velocity-squared than velocity-proportional. The viscous damping term itself is an assumption, not a law; this is where the model, not just the measurement, begins to fail, regardless of how well the code is validated.

8. Report

Prompt 7 — Report (Phase 3, student fact-checks the output)

Assemble a report from material I will paste: my specification, each validation check with its predicted and observed values, the parameter sweep, and the failure case. Use only what I give you, do not invent results or fill gaps. Mark anything you are unsure about so I can verify it.

Reminder: the review of this report against the student's own records, not the generation of it, is the graded work.

9. Post-Class Extension: Lunar Gravity

This section documents a worked Post-Class Extension Activity for the pendulum: a question that would be impractical to test experimentally in a classroom, but is a single-parameter change in a validated simulation.

- **New question.** How would the period change if this pendulum swung on the Moon instead of Earth?
- **Extension.** Set $g = 1.62 \text{ m/s squared}$ (lunar surface gravity) in place of 9.81 m/s squared , with damping = 0 and theta-zero at 10 degrees, and predict the new period before touching the slider.
- **Prediction (small-angle, before running).** $T = 2\pi \sqrt{L/g}$. For $L = 1 \text{ m}$ and $g = 1.62 \text{ m/s squared}$, T is about 4.93 s, roughly 2.45 times the Earth period, since period scales as 1 over the square root of g .
- **Result.** Setting $g = 1.62$ in the existing sim gave a measured period of about 4.95 s, within 1% of the small-angle prediction and consistent with the ratio expected from $\sqrt{9.81/1.62}$, about 2.46.
- **Validating the extension:** Set g back to 9.81 and confirm the period and energy-conservation readouts return to their originally validated values. Recovering the original validated behavior when the new setting is reversed is the check that the extension did not silently change anything else in the model.

Other workable extensions in this same spirit, changing one parameter to a regime that is impractical or unsafe to test by hand, include: compressing or stretching the simulation's time scale to observe many decay cycles in seconds instead of minutes; and testing a damping regime meant to represent an unusual environment (e.g., a viscous fluid), then confirming the model still conserves energy correctly when damping is set back to zero. In every case, the validating move is the same: return the changed parameter to its original value and confirm the previously validated behavior comes back.

Change Log for This Reference Build

v1 (initial generation): passed both validation checks on first pass; no corrective prompt needed. v1 also serves as the class's live fail-and-validate demo script; instructors can re-run Prompt 1 with a deliberately under-specified version (omit “full nonlinear,” omit units) to reproduce a realistic first failure live, per the Instructor Guide's live fail-and-validate demonstration. v1 plus the lunar-gravity extension (Section 9) together serve as the worked example referenced in the Instructor Guide's Appendix A and the Post-Class Extension Activity.