HW08

Dan Holland

December 10, 2025

1 Cycling with Drag

1.1 Cycling with Air Drag

1.1.1 Problem summary

We extend the no–drag model by adding a quadratic air–drag term to the cyclist's equation of motion:

$$\frac{dv}{dt} = \frac{P}{m\,v} - \frac{1}{2}\frac{C_D\,\rho\,A}{m}v^2,$$

where the first term represents acceleration from the rider's power input and the second term is deceleration due to drag (force divided by mass). We use the same initial speed and time window as before and evolve the solution with a forward Euler step. (Code in `bicycle.py`.)

Implementation:

- Parameters: $v_0 = 4\,\mathrm{m/s}$, $m = 70\,\mathrm{kg}$, $P = 400\,\mathrm{W}$, $C_D = 0.9$, $\rho = 1.225\,\mathrm{kg/m^3}$, $A = 0.33\,\mathrm{m^2}$.
- Integrator: forward Euler with $\Delta t = 0.1\,\mathrm{s}$ on $t \in [0, 200]$ s.
- Output: velocity vs. time saved as `bicycle.png`; plot produced from the code above.

1.1.2 Results

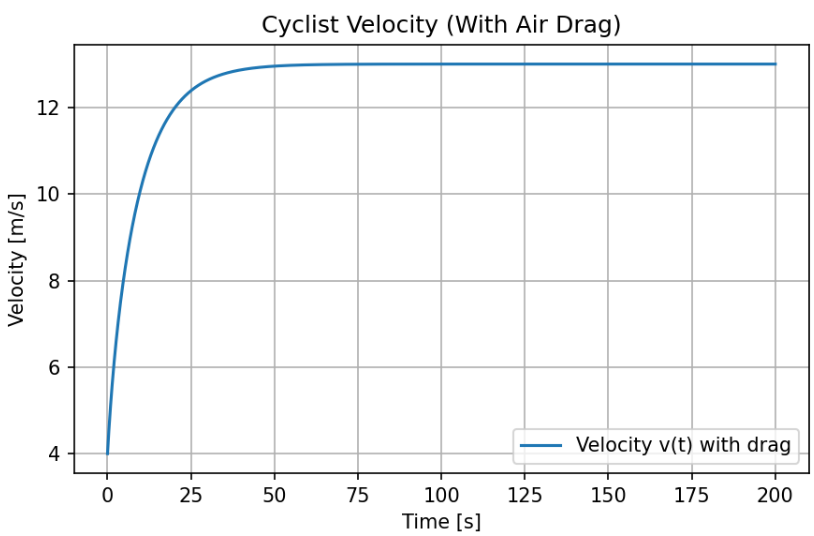

Figure 1: Cyclist velocity with quadratic air drag. Velocity rises rapidly, then approaches a constant value (terminal speed).

1.1.3 Discussion

Compared to the no–drag case (HW07), where $v(t)$ grew without bound, the drag term makes the curve bend over and level off.

- Early time: v increases quickly because drag is small at low speed.
- Approach to equilibrium: as v grows, drag $\propto v^2$ increases and increasingly cancels the power term.
- Terminal velocity: set $dv/dt \approx 0$ to balance input and losses:

$$\frac{P}{m\,v_t} = \frac{1}{2}\,\frac{C_D\,\rho\,A}{m}\,v_t^2 \quad\Rightarrow\quad v_t = \left(\frac{2P}{C_D\,\rho\,A}\right)^{1/3}.$$

  With our numbers, $v_t \approx \big(800/(0.9 \cdot 1.225 \cdot 0.33)\big)^{1/3} \approx 13\,\mathrm{m/s}$ ($\approx 29\,\mathrm{mph}$), consistent with the plateau in Fig. 1.
- Physical interpretation: at v_t the rider's power exactly offsets aerodynamic losses, so speed becomes (nearly) constant.

1.2 The Stokes Term

1.2.1 Problem summary

We add a small linear (viscous) drag term to the air–drag model:

$$\frac{dv}{dt} = \frac{P}{m\,v} - \frac{1}{2}\,\frac{C_D\rho A}{m}\,v^2 - \frac{\eta A}{mh}\,v.$$

The new term is proportional to v and is expected to matter only at very low speeds. (Code used: `bicycle.py`.)

Implementation:

- Parameters: $v_0 = 4\,\mathrm{m/s}$, $m = 70\,\mathrm{kg}$, $P = 400\,\mathrm{W}$, $C_D = 0.9$, $\rho = 1.225\,\mathrm{kg/m^3}$, $A = 0.33\,\mathrm{m^2}$, $\eta = SI2 \times 10^{-5} Pa\,s$, $h = 2\,\mathrm{m}$.
- Integrator: forward Euler, $\Delta t = 0.1\,\mathrm{s}$, $t \in [0, 200]$ s.
- Output: velocity vs. time saved as `bicycle2.png`.

1.2.2 Results

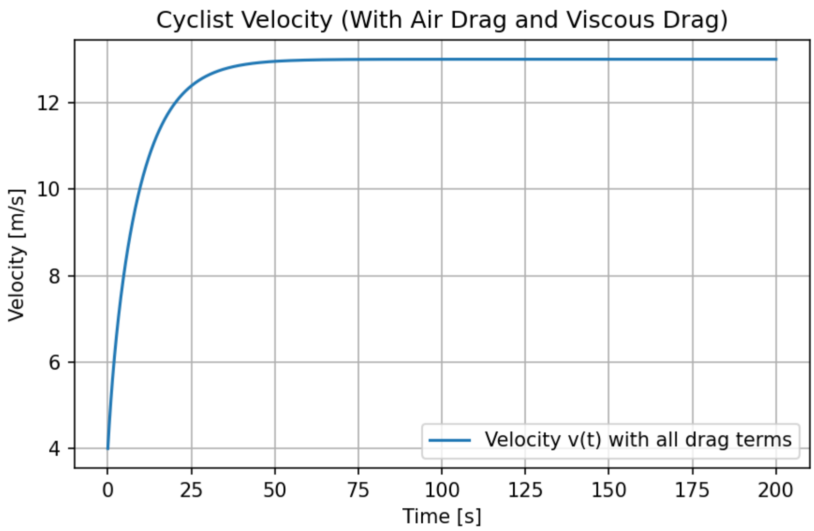

Figure 2: Cyclist velocity with quadratic air drag *and* viscous drag. The curve is essentially the same as Task 1a and approaches a similar terminal speed.

1.2.3 Discussion

Compared to Task 1a (quadratic drag only), adding the viscous term barely changes the curve:

- Terminal velocity is almost unchanged. The viscous contribution is tiny at normal cycling speeds.
- Why negligible? Magnitude check at $v = 10\,\mathrm{m/s}$.

$$\left|\frac{1}{2}\frac{C_D \rho A}{m} v^2\right| \approx \frac{0.5 \cdot 0.9 \cdot 1.225 \cdot 0.33 \cdot 10^2}{70} \approx 2.6 \times 10^{-1}\,\mathrm{m/s^2},$$

  while

$$\left|\frac{\eta A}{mh} v\right| \approx \frac{(2 \times 10^{-5}) \cdot 0.33 \cdot 10}{70 \cdot 2} \approx 4.7 \times 10^{-7}\,\mathrm{m/s^2}.$$

  The quadratic term is $\sim 5 \times 10^5$ times larger.

- Where would viscous = quadratic? Solve $\frac{\eta A}{mh} v = \frac{1}{2}\frac{C_D \rho A}{m} v^2 \Rightarrow v_* = \dfrac{2\eta}{C_D \rho h}$. With our numbers: $v_* = \dfrac{2(2 \times 10^{-5})}{0.9 \cdot 1.225 \cdot 2} \approx 1.8 \times 10^{-5}\,\mathrm{m/s}$, effectively zero. This confirms the statement that the viscous term is negligible except at the very smallest velocities.

1.3 Cycling with Grade (Uphill/Downhill)

1.3.1 Problem summary

We include the component of gravity along the road. With grade given in percent, we set

$$\theta = \arctan\!\big(\mathrm{grade}/100\big), \qquad F_{\mathrm{g},x} = mg\sin\theta,$$

so

$$\frac{dv}{dt} = \frac{P}{m\,v} - \frac{1}{2}\frac{C_D \rho A}{m} v^2 - \frac{\eta A}{mh} v - g\sin\theta.$$

For small slopes we use $\sin\theta \approx \tan\theta \approx \mathrm{grade}/100$.

Implementation:

- Parameters as before: $v_0 = 4\,\mathrm{m/s}$, $m = 70\,\mathrm{kg}$, $P = 400\,\mathrm{W}$, $C_D = 0.9$, $\rho = 1.225\,\mathrm{kg/m^3}$, $A = 0.33\,\mathrm{m^2}$, $\eta = SI2 \times 10^{-5} Pa\,s$, $h = 2\,\mathrm{m}$, $g = 9.81\,\mathrm{m/s^2}$.
- Integrator: forward Euler, $\Delta t = 0.1\,\mathrm{s}$, $t \in [0, 200]$ s.
- Grades explored: 0% (flat), +10% (uphill), −10% (downhill), +20% (steep uphill).

1.3.2 Results

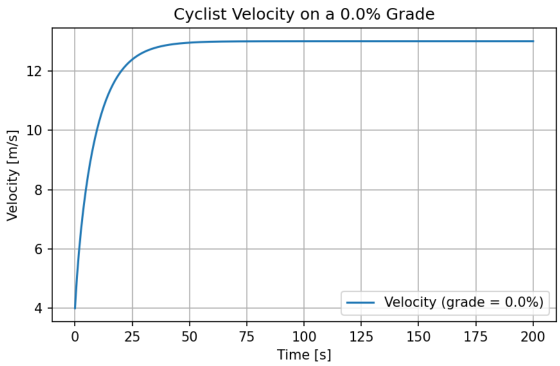

Figure 3: Grade 0% (flat).

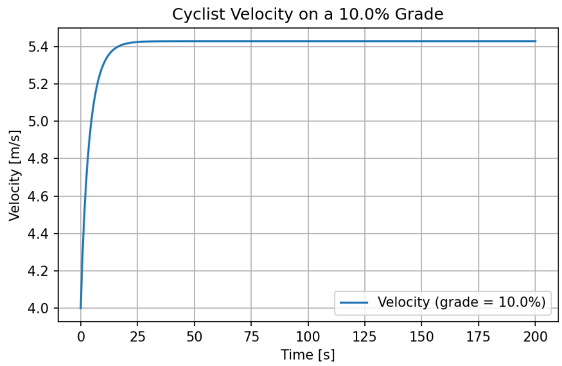

Figure 4: Grade +10% (uphill).

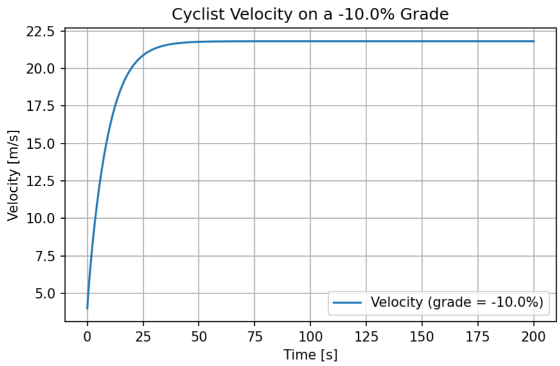

Figure 5: Grade −10% (downhill).

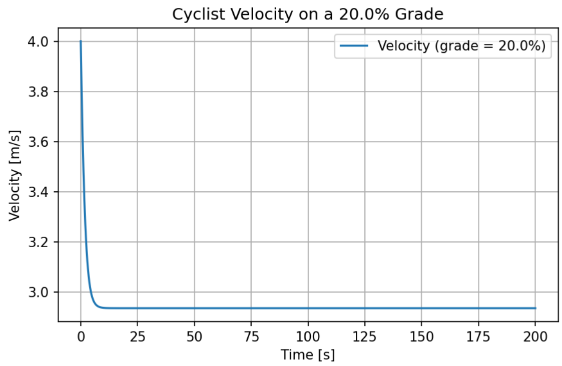

Figure 6: Grade +20% (steep uphill).

1.3.3 Discussion

Intuition matches the physics: uphill reduces terminal speed; downhill increases it.

- How grade changes terminal velocity. At equilibrium $dv/dt \simeq 0$,

$$\frac{P}{mv_t} = \frac{1}{2}\frac{C_D \rho A}{m} v_t^2 + \frac{\eta A}{mh} v_t + g \sin\theta,$$

  a general balance between power input and losses. Increasing $\sin\theta$ (uphill) shifts the balance to a lower v_t; negative $\sin\theta$ (downhill) to a higher v_t.

- Viscous vs quadratic drag. The linear term is negligible at cycling speeds; e.g. at $v = 10\,\mathrm{m/s}$, the quadratic term contributes $\approx 2.6 \times 10^{-1}\,\mathrm{m/s^2}$ while the viscous term contributes $\approx 4.7 \times 10^{-7}\,\mathrm{m/s^2}$ (over 5×10^5 times smaller).

- When does the rider slow down immediately? At $t = 0$ with $v_0 = 4\,\mathrm{m/s}$,

$$\left.\frac{dv}{dt}\right|_{t=0} = \frac{P}{mv_0} - \frac{1}{2}\frac{C_D\rho A}{m}v_0^2 - \frac{\eta A}{mh}v_0 - g\sin\theta.$$

  Using the parameters above, the first three terms give $\approx 1.387\,\mathrm{m/s^2}$. Thus the motion turns negative when $g\sin\theta > 1.387\,\mathrm{m/s^2} \Rightarrow \sin\theta \gtrsim 0.141$, i.e. a grade $\gtrsim 14\%$. This matches the strong slowdown seen at +20% (Fig. 6) and explains why +10% (Fig. 4) still reaches a modest v_t.

- Is there a grade where the rider cannot maintain $v_0 = 4\,\mathrm{m/s}$? Yes, any grade above $\sim 14\%$ makes $dv/dt < 0$ at start, so the cyclist slows below v_0.

- Downhill behavior. With $\sin\theta < 0$ (Fig. 5), gravity aids motion and v_t increases; it still levels off because quadratic drag grows as v^2.

2 Random Walk (Single Walker)

2.1 Problem summary

We simulate a one–dimensional random walk of n steps. The walker starts at $x_0 = 0$ and takes unit steps of ± 1 with equal probability at each time step. The goal is to generate and plot position versus step number for two independent walks with $n = 100$.

Implementation:

- Initialize $x = 0$; store the starting point so the curve includes step 0.
- For each step $1 \ldots n$: choose $+1$ or -1 uniformly at random and update x.
- Create the step index $\{0, \ldots, n\}$ and plot position vs. step number.
- Run the program twice to produce two independent realizations (`random1.png`, `random2.png`).

2.2 Results

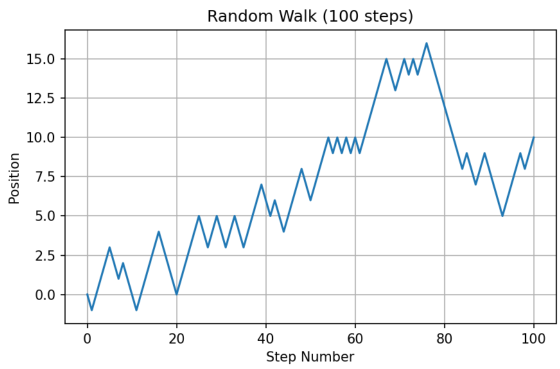

Figure 7: Random walk A ($n = 100$).

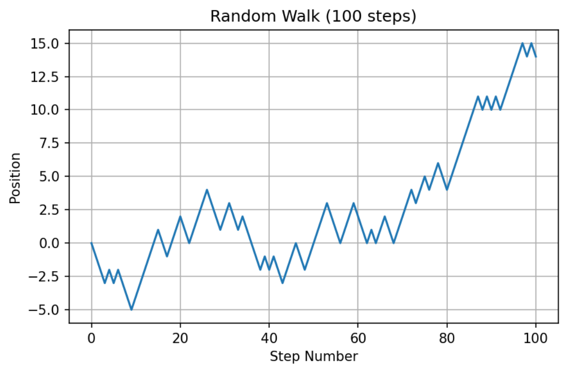

Figure 8: Random walk B ($n = 100$).

2.3 Discussion

Both trajectories begin at 0 and then wander due to the sequence of ± 1 steps. Although each step has zero mean, individual paths are irregular and can drift positive or negative for many steps in a row. Over many iterations, we expect the average position $\langle x_n \rangle$ to stay near zero, while the typical spread (e.g., $\sqrt{\langle x_n^2 \rangle}$) grows with $\sqrt{n}$; the two example paths in Figs. 7–8 show how different outcomes can be, even with identical rules.

3 Group of Random Walkers

3.1 Problem summary

We simulate a group of $N_{\text{walkers}} = 500$ independent one–dimensional random walkers, each starting at $x_0 = 0$ and taking unit steps of ± 1 with equal probability for $n = 100$ steps. At each step n we record the group statistics $\langle x_n \rangle$ (mean displacement) and $\langle x_n^2 \rangle$ (mean squared displacement). Code: `random.py`.

Implementation:

- For each walker: build a list of positions $[x_0, \ldots, x_n]$ by adding ± 1.
- For each step index n: collect all walkers' positions at that step and compute $\langle x_n \rangle$ and $\langle x_n^2 \rangle$.
- Plot $\langle x_n^2 \rangle$ vs. step number n (saved as `random3.png`).

3.2 Results

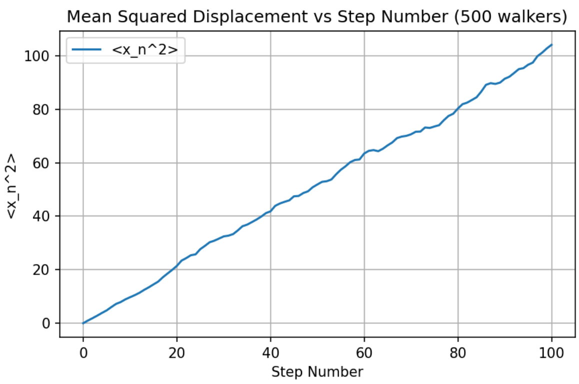

Figure 9: Mean squared displacement $\langle x_n^2 \rangle$ as a function of step number n for 500 walkers. The curve is approximately linear in n.

3.3 Discussion

- Why $\langle x_n \rangle \approx 0$: Symmetry. Each step is equally likely to be $+1$ or -1, so left and right displacements cancel across the group.
- Why $\langle x_n^2 \rangle$ increases: Even with zero mean, walkers spread out from the origin; squaring removes sign and measures the spread.
- Relationship: For a simple unbiased walk with unit steps, $\langle x_n^2 \rangle \approx n$. The plot in Fig. 9 shows an approximately straight line with slope near 1, matching expectations.