

Group work: optimization

One-dimensional optimization

An object subject to linear drag due to air resistance is projected upward at a specified velocity. Its altitude z is modeled as a function of time t as

$$z = z(t) = z_0 + \frac{m}{c} \left(v_0 + \frac{mg}{c} \right) \left(1 - e^{-\frac{c}{m}t} \right) - \frac{mg}{c}t$$

where

- z = altitude (m) above Earth's surface (which has $z = 0$)
- z_0 = initial altitude (m)
- m = mass (kg)
- c = a linear drag coefficient ($\frac{\text{kg}}{\text{s}}$)
- v_0 = initial velocity ($\frac{\text{m}}{\text{s}}$), where positive velocity is up
- t = time (s)

Given $g = 9.81 \frac{\text{m}}{\text{s}^2}$, $z_0 = 100$ m, $v_0 = 55 \frac{\text{m}}{\text{s}}$, $m = 80$ kg, and $c = 15 \frac{\text{kg}}{\text{s}}$, use `optimize()` to:

- Graph the object's altitude vs. time.
- Find the time at which the object strikes the ground.
- Find the object's maximum height.
- Find the time at which the object reaches its maximum height.

(I got these answers:

- The object hits the ground at $t \approx 11.611$ s.
- The object's maximum height is ≈ 192.861 m.
- The object reaches its maximum height at $t \approx 3.832$ s.

)

Multi-dimensional optimization

The two-dimensional distribution of pollutant concentration in a channel can be described by

$$z = \text{concentration}(x, y) = 7.9 + 0.13x + 0.21y - 0.05x^2 - 0.016y^2 - 0.007xy$$

- Graph the concentration in the region defined by $-10 \leq x \leq 10$ and $0 \leq y \leq 20$.
- Use `gradient.descent()` to find the location of the peak concentration given that it lies in the given region. Notice that using *descent* to find a peak requires a little cleverness. Check your work by adding the peak concentration point $(x, y, z = \text{concentration}(x, y))$ to your graph.
- Use `optim()` to solve the same problem two more times:
 - Try `method="Nelder-Mead"` and provide `fn` but not `gr`
 - Try `method="BFGS"`, which approximates Newton's method, providing `fn` and `gr`

How many function calls to `fn` and/or `gr` did `optim()` make in each case? (Hint: check the "Value" section of `?optim`.) Which method would you expect to be faster? (We'll learn how to time computations soon.)

(I got $(x, y) \approx (0.853, 6.376)$. Your solution may be different, depending on your choices of `gamma`, `epsilon`, and `n`, but it should be near the peak of your graph.)

What to turn in

Please submit one "optimization.Rmd" file that includes comments near the top containing the names and email addresses ("`...@wisc.edu`") of your group members, one per line.