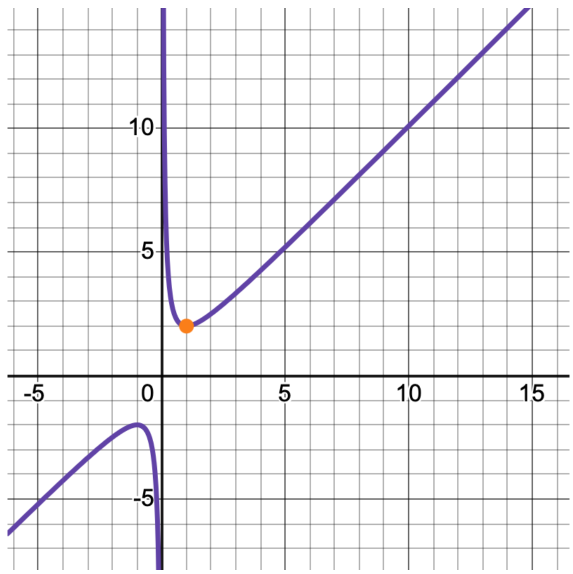
In-class Exercise 5.1
Unconstrained optimization (derivative-free methods)
AMATH 301
University of Washington
Jakob Kotas

1. Consider the function

$$f(x) = x + \frac{1}{x}, \qquad x > 0$$

which is unimodal and has a global minimum at $x = 1$, where $f(1) = 2$. It's easy to show this with calculus, but we won't use calculus in this problem.



(a) Beginning with the initial interval $[0.5, 5]$, perform trisection iteration to approximate the minimum. That is, if the initial interval is $[a, b]$ then we set $x_1 = \frac{2}{3}a + \frac{1}{3}b$ and $x_2 = \frac{1}{3}a + \frac{2}{3}b$. If $f(x_1) < f(x_2)$, then keep the interval $[a, x_2]$ and relabel it as $[a, b]$, then repeat. If $f(x_1) > f(x_2)$, then keep the interval $[x_1, b]$ and relabel it as $[a, b]$, then repeat. Terminate as soon as the interval width $b - a < 10^{-6}$.

(b) Beginning with the initial interval $[0.5, 5]$, perform golden section iteration to approximate the minimum. That is, if the initial interval is $[a, b]$ then we set $x_1 = ca + (1-c)b$ and $x_2 = (1-c)a + cb$ where $c = (\sqrt{5} - 1)/2 = 1/\varphi$ where φ is known as the golden ratio. Otherwise, the same rules are followed as in trisection iteration above.

Visualization of trisection and golden section iterations

(c) (Conceptual) Why does the golden ratio come into play here?

(d) (Conceptual) What is the size of the interval after each iteration, compared the size of the interval previously? Under this metric, which is more efficient: trisection or golden section iteration?

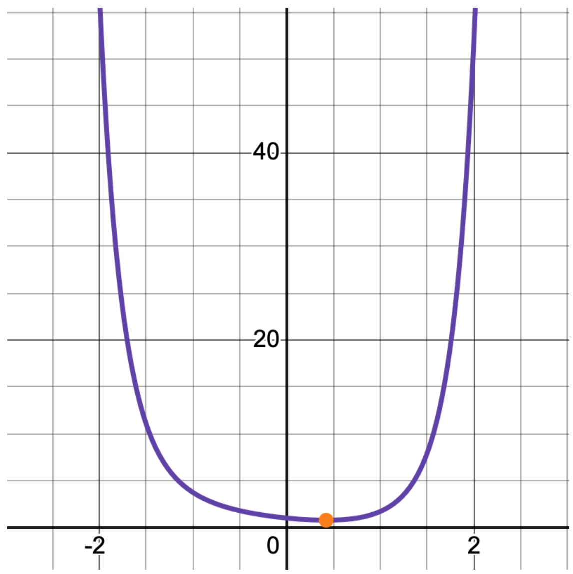
(e) (Conceptual) How many function evaluations (calculating $f(x)$ for some x) must be performed at each iteration for trisection vs. golden section iteration? Under this metric, which method is more efficient?

(f) (Conceptual) Why does bisection not work for finding a local minimum of a function?

2. Consider the function

$$g(x) = e^{x^2} - x$$

which is unimodal and has a global minimum at $x \approx 0.419365$, where $g(x) \approx 0.772914$.



Beginning with initial points $x_1 = -1$, $x_2 = 0$, $x_3 = 2$, perform successive parabolic interpolation to approximate the minimum. Use the formula formula for x_0, the minimum of the parabolic interpolation through x_0, x_1, and x_2, either from Kutz's video or the version shown below:

$$x_0 = \frac{g(x_1)(x_3^2 - x_2^2) + g(x_2)(x_1^2 - x_3^2) + g(x_3)(x_2^2 - x_1^2)}{2[g(x_1)(x_3 - x_2) + g(x_2)(x_1 - x_3) + g(x_3)(x_2 - x_1)]}.$$

If $x_0 < x_2$, then discard x_3 and relabel the remaining three points as x_1, x_2, x_3 in increasing order. If $x_0 > x_2$, then discard x_1 and relabel the remaining three points as x_1, x_2, x_3 in increasing order. Repeat and terminate as soon as either $x_3 - x_2 < 10^{-6}$ or $x_2 - x_1 < 10^{-6}$.

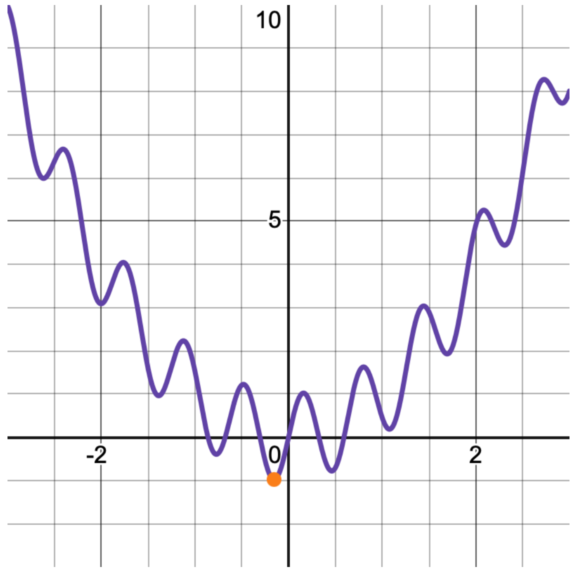
Visualization of successive parabolic interpolation

3. Use the built-in function `fminbound` from the `scipy.optimize` library to approximate the solutions for problems (1) and (2).

4. Consider the non-unimodal function:

$$h(x) = x^2 + \sin(10x)$$

which has a unique global minimum at $x \approx -0.153999$, where $h(x) \approx -0.975810$. Use golden section iteration with different starting intervals to see what happens with this problem.



Visualization of non-unimodal function