Problem 1: Newton's Method

Try out Newton's method with the following functions and see what happens:

- Notice that the following function has roots for $x \in (-1, 1)$:

$$f(x) = \frac{1}{1+x^2} - \frac{1}{2}$$

- Notice that the following function has a root when $x = 0$

$$f(x) = \begin{cases} \sqrt{x}, & x \geq \\ -\sqrt{-x}, & x < 0 \end{cases}$$

- Notice that this function has a root at $x = 0$

$$f(x) = x^{1/3}$$

Try plotting these functions and "how Newton's method looks like". Why does this happen?

Problem 2: Round off error

Derivatives

The formal definition of derivative is the following:

$$f'(a) = \lim_{\varepsilon \to 0} \frac{f(a+\varepsilon) - f(a)}{\varepsilon}.$$

Then if might be expected that a "good" approximation for this derivative is the following:

$$\lim_{h \to 0} \frac{f(a+h) - f(a)}{h}.$$

So we might expect that every time we set h to be smaller we get a better approximation to the derivative, right? Try it out. Why does this fails? Plot your solution.

Math vs computational math

As seen in lecture, the order in which we tell the computer to perform operations is quite important. "Real" math operations might be different from computational ones. Try if yourself.

- We know that $4.9 - 4.845 = 0.055$. So if we ask Python if this is $4.9 - 4.845 = 0.055$ then it should output a True. What happens when you set $4.9 - 4.845 == 0.055$. Why?
- Notice that:

$$1 = 1 + \frac{1}{3} - \frac{1}{3} + \frac{1}{3} - \frac{1}{3} + \frac{1}{3} - \frac{1}{3} + ...$$

  What happens when you code this "a lot of times"?
- Try to "fix" the last problem.
- Define two functions in Python

$$f(a,b) = \frac{100000a}{10b}$$
$$g(a,b) = \frac{a}{b}\frac{100000}{10}$$

  They're mathematically equivalent, but what happens when being coded?