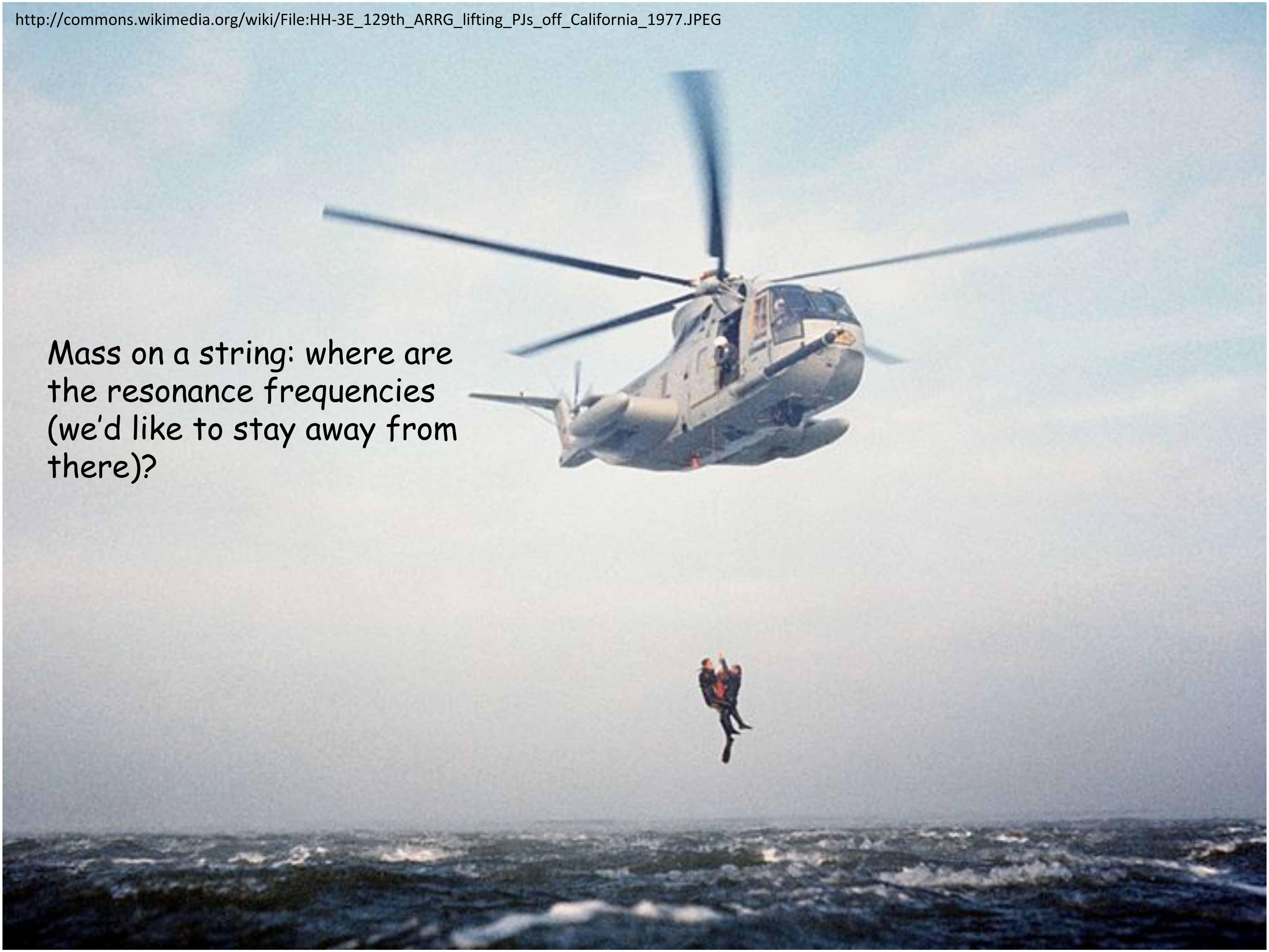
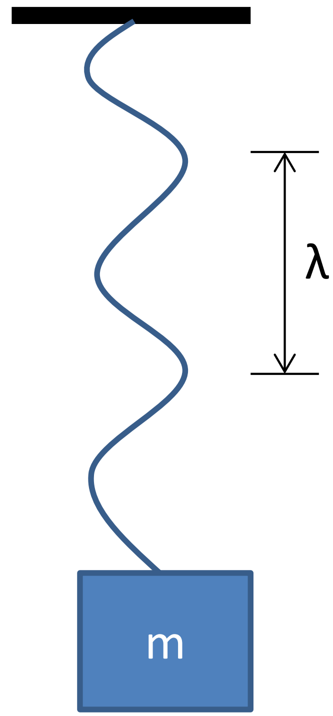
Roots Part a: Bracketing Methods

Image from Aaron Escobar


http://commons.wikimedia.org/wiki/File:HH-3E_129th_ARRG_lifting_PJs_off_California_1977.JPEG
Mass on a string: where are the resonance frequencies (we'd like to stay away from there)?

Mass-loaded string

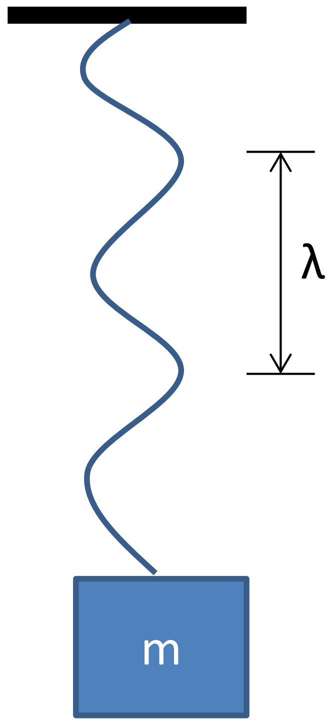


Wavelength depends on frequency of the wave on the spring:

$$\lambda = \frac{c}{f}$$

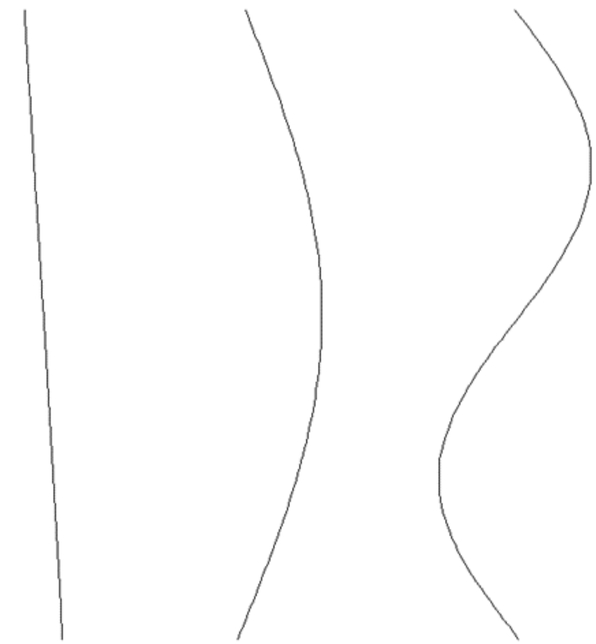
Note: the wave speed *c* is for a wave on a string under tension, like a guitar string, $C^2 = T/mu$, where *T* is the tension and *mu* the linear mass density

Mass-loaded string



This system has natural modes, which are excited at the natural frequencies:

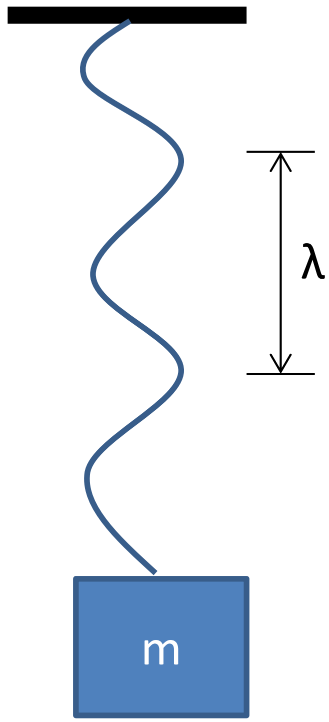
Mode 1 | Mode 2 | Mode 3

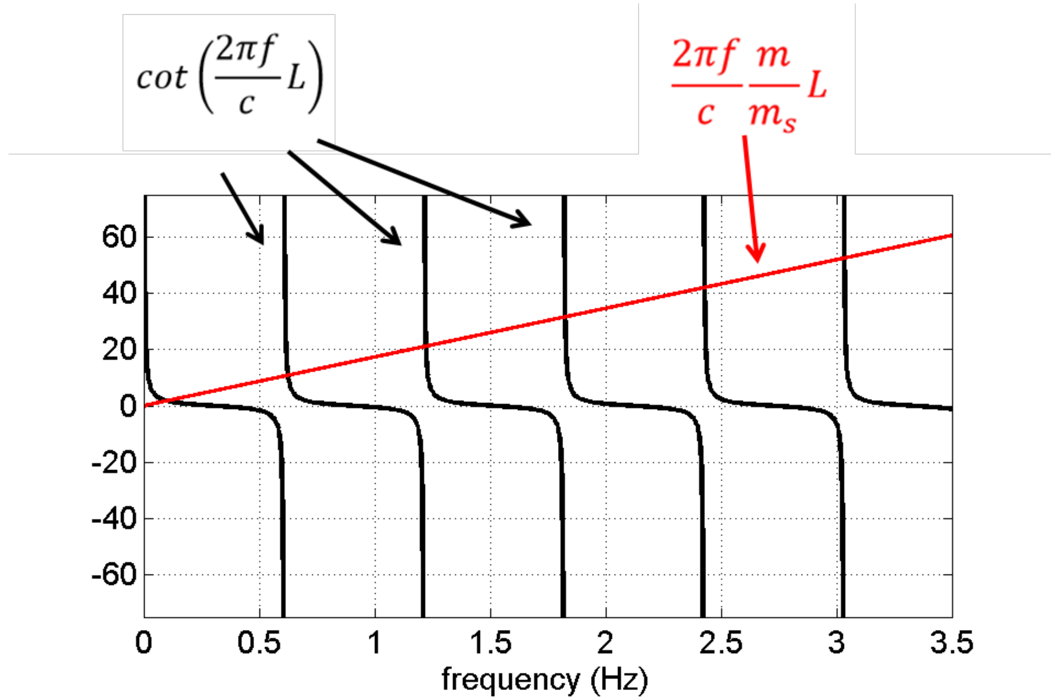


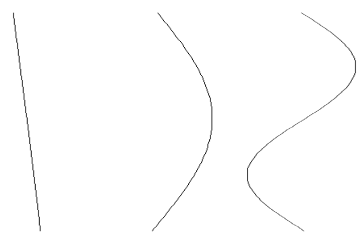
$$cot\left(\frac{2\pi f}{c}L\right) = \frac{2\pi f}{c}\frac{m}{m_s}L$$

The frequencies, f, that satisfy this equation are the natural equations. But how do we find them?

Mass-loaded string



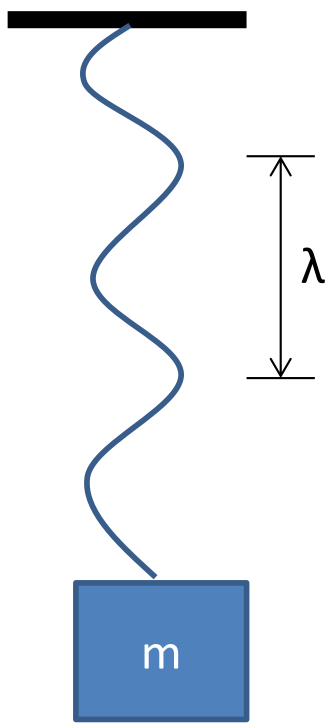




$$\cot\left(\frac{2\pi f}{c}L\right) = \frac{2\pi f}{c}\frac{m}{m_s}L$$

The frequencies, f, that satisfy this equation are the natural equations. But how do we find them?

Mass-loaded string



$$\cot\left(\frac{2\pi f}{c}L\right) = \frac{2\pi f}{c}\frac{m}{m_s}L$$

This is a transcendental equation.
How do we find *f* ???

Ans.

Option A: graphical methods

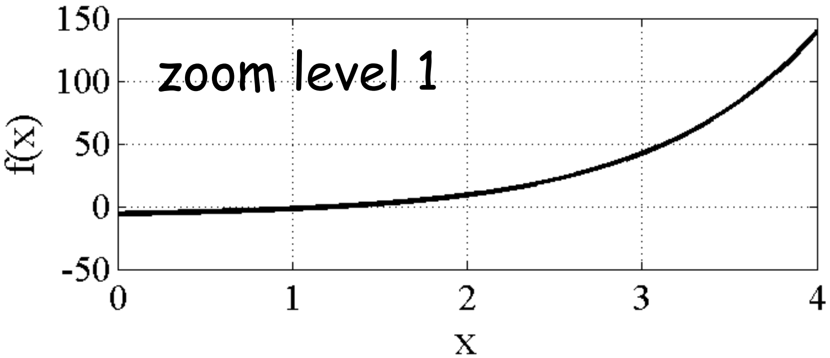
Option B: bracketing methods

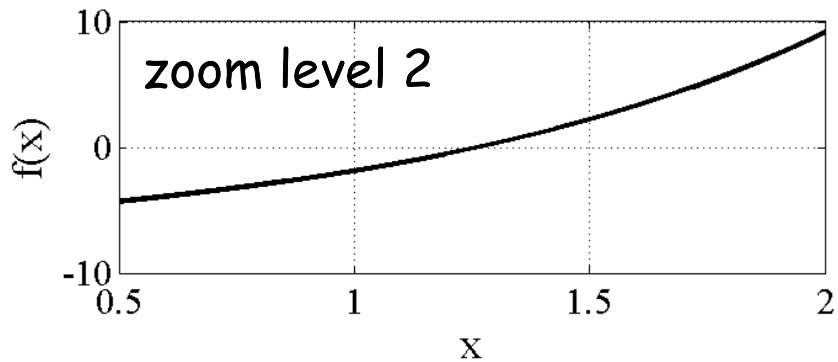
Option C: open methods

graphical methods

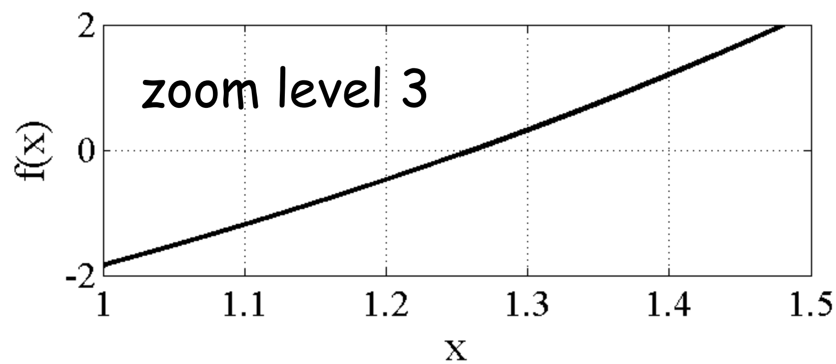
Example. Find the roots of the following equation:

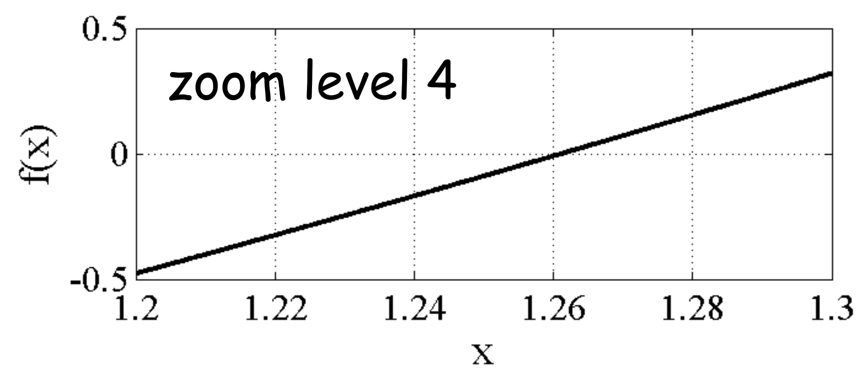
$$f(x) = 3e^x - x^2 - 9 = 0$$

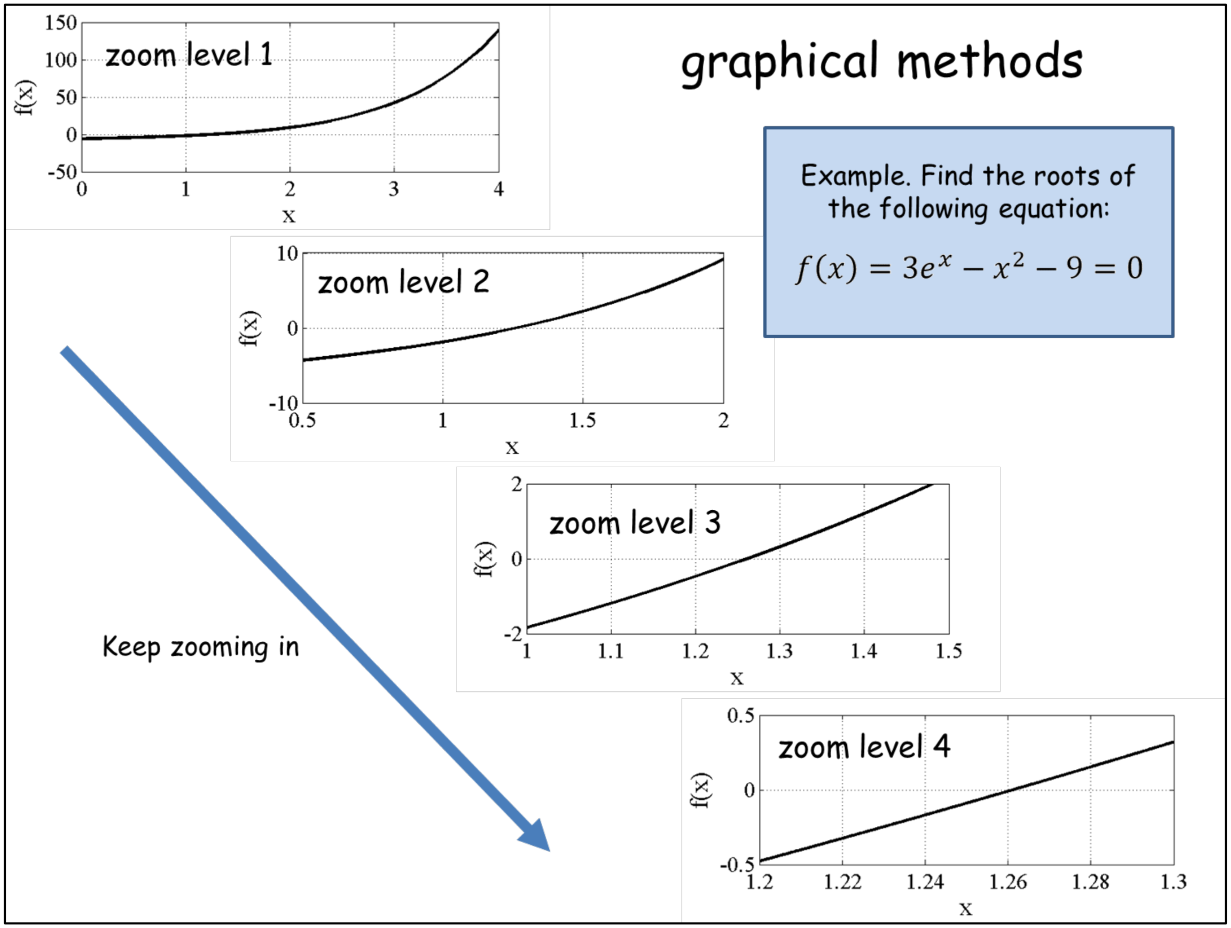




Keep zooming in







This is not practical. Can we automate this somehow?

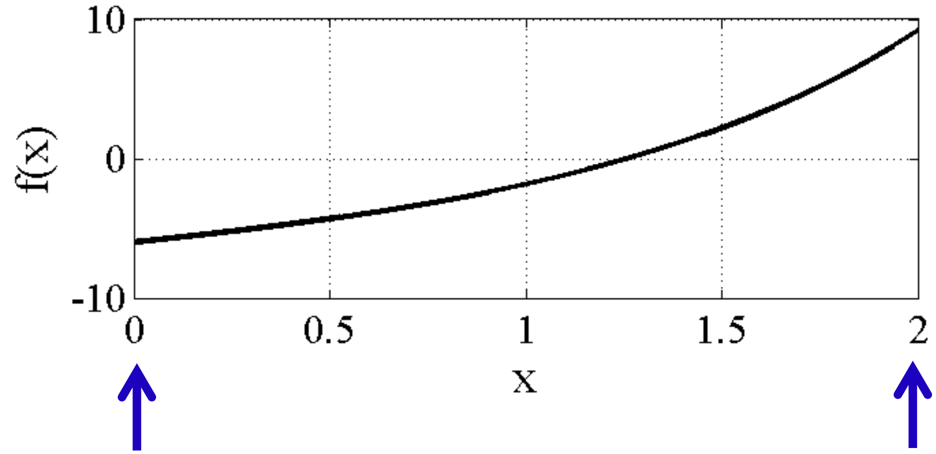
What information are we using on each sequential 'zoom'?

Bisection Method

Step 1: Make an initial guess of two values that bracket the root: x1, and x2.

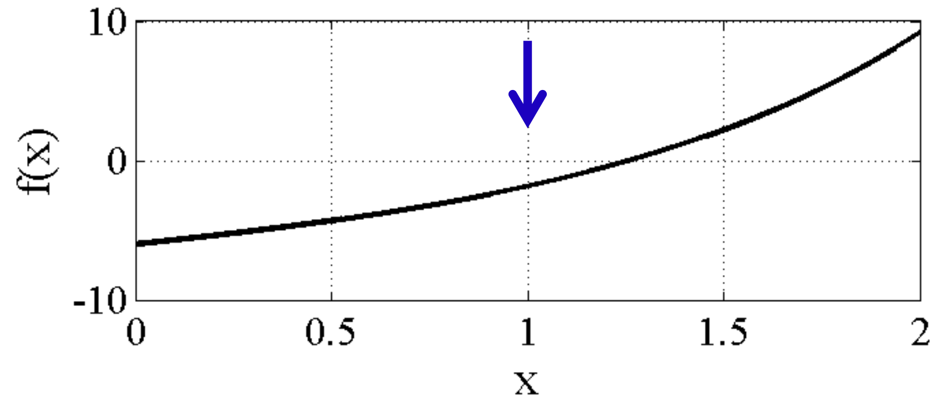
x1 = 0.0
x2 = 2.0

(the root is bracketed by x1 and x2)



Step 2: Estimate the root as half-way between x1 and x2: xr = (x1 + x2)/2

xr = 1.0



Bisection Method

<u>Step 3</u>: Evaluate our answer. If it is good enough, we're done. If not, keep going to step 4.

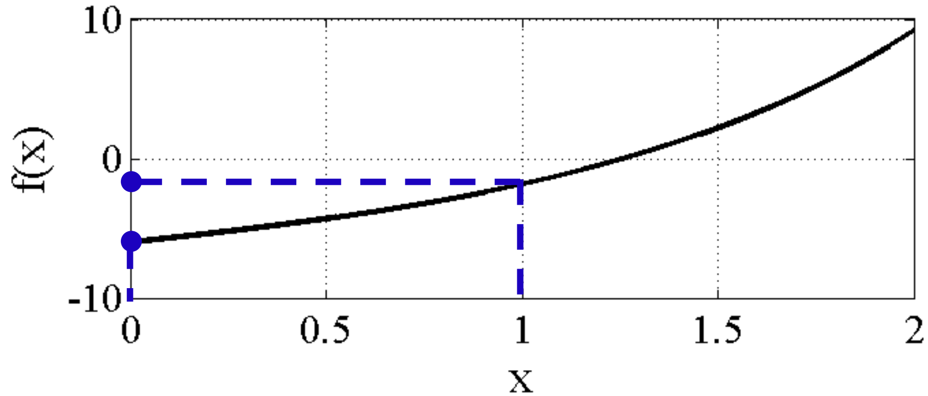
How do we evaluate whether we've done 'good enough'?

is $\frac{x_r - x_{old}}{x_r} \times 100$ < N% ?

Some prescribed precision in our answer

Bisection Method

Step 4: Compute the product of the function at the lower bound, and at the midpoint: s = f(x1)*f(xr)



s = (-6.00)*(-1.85) = 11.1

Step 5: Examine the sign of s. There are three options:

1. s < 0 → the midpoint lies between x1 and xr, so set x2 equal to xr. Return to step 2.
2. s > 0 → the midpoint lies between xr and x2, so set x1 equal to xr. Return to step 2.
3. s = 0 → this is the root, and you are done (unlikely).

What happens if there is no root between x1 and x2?

s > 0

x1 = 1.0
x2 = 2.0

Step 6: Return to step 2.

Bisection Method

What do we need to know at the beginning (inputs):

- f(x) -- the function we are trying to find the roots of
- Initial guesses for x1 and x2
- We need to know how good is 'good enough' (an acceptable error)
- A maximum number of iterations in case we get stuck!

What do we want to get out of the program?

- The x-location of the root -- xr
- The error associated with this estimate of xr
- The number of iterations we used

What are some of the things our bisection will need to do

- Calculate f(x) for different values of x, many times
- Calculate s and move forward in different ways, depending on the sign
- Assess the error in our estimate of x, and stop when its acceptable to do so

Bisection Method

Here is what our program should look like:

1. Make initial guesses for x1 and x2

2. Assign xr to be equal to x1 (why? See step 5a)

3. Define an acceptable error in the estimate of xr: Estop = 0.1

4. *Initialize* our error estimate to something larger than Estop: E = Estop*2

5. While our error estimate is greater than our acceptable error
   A. Assign xr to be the 'old' root estimate (xrOld = xr;)
   B. Calculate a new xr (midpoint between x1 and x2)
   C. Estimate the error in xr: E = abs(xr-xold)/xr
   D. Calculate s
   E. If s < 0, set x2 = xr; elseif s > 0, set x1 = xr; else stop this iteration

When 5 is complete, xr is our estimate of the midpoint, E is our estimate of the error in xr. If we like, we can keep track of the number of iterations we needed.

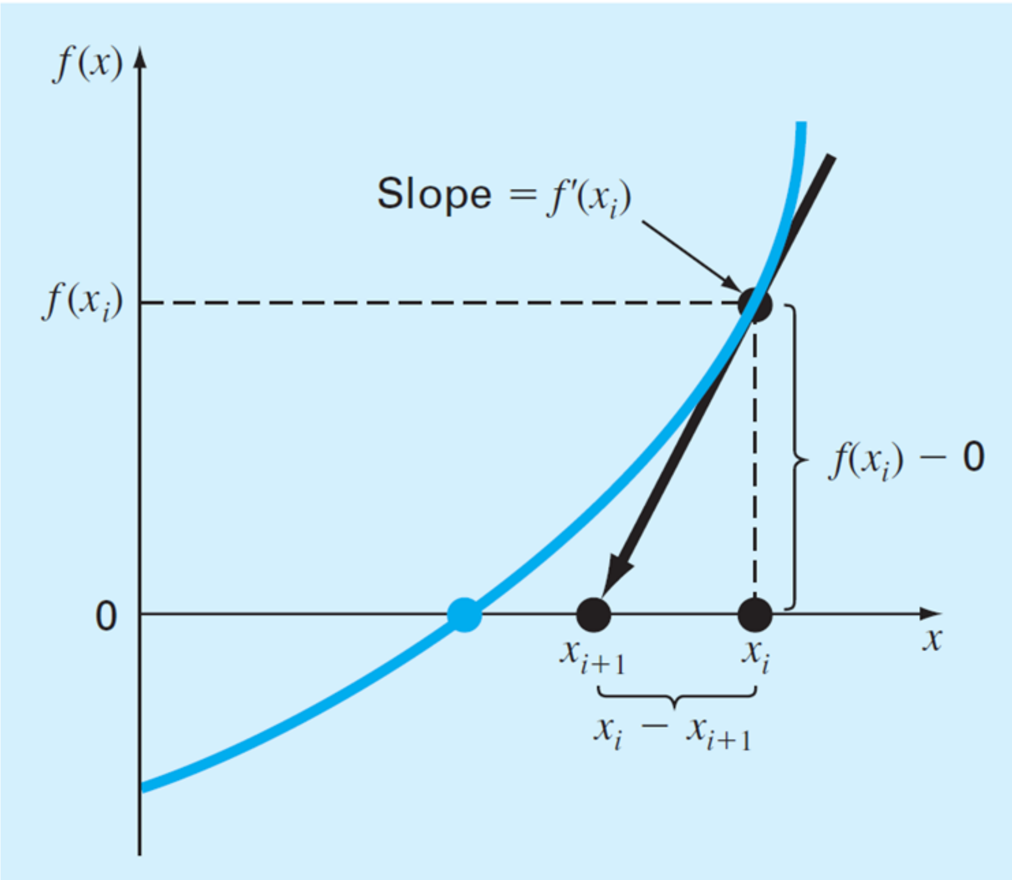
Bisection Method

This is a pretty straight forward procedure. What are the downfalls?

- We need two starting guesses – these have to bracket the solution to start with or we'll be in trouble.

- Might not be the most efficient way to find a root
  - Is the midpoint the best one to choose?

Newton-Raphson Method

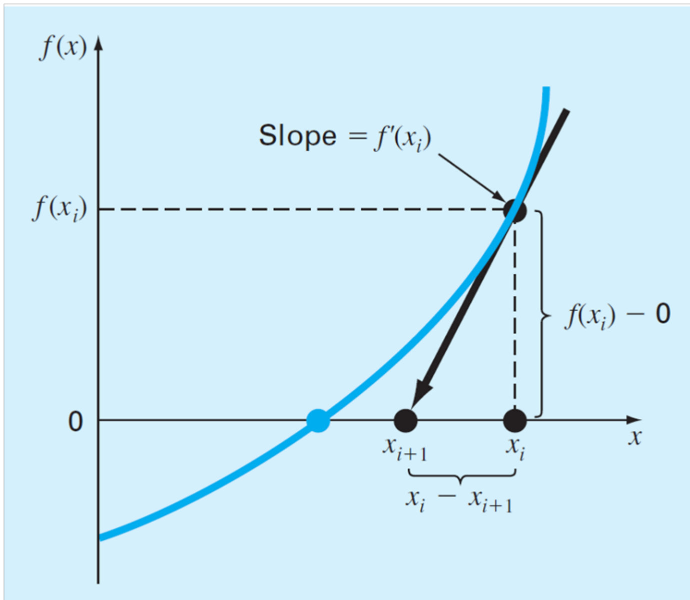


Chapra, Figure 6.4

Key Points:

- Starts with only a single point (as opposed to two points for bracketing methods)
- Uses information about the slope of the function (a tangent line) to 'guess' at the true root.

Newton-Raphson Method



Chapra, Figure 6.4

Recall Taylor series:

$$f(x_{i+1}) \cong f(x_i) + f'(x_i)(x_{i+1} - x_i)$$

If x_{i+1} is the root, than $f(x_{i+1})=0$. So we can rearrange the equation above as follows:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

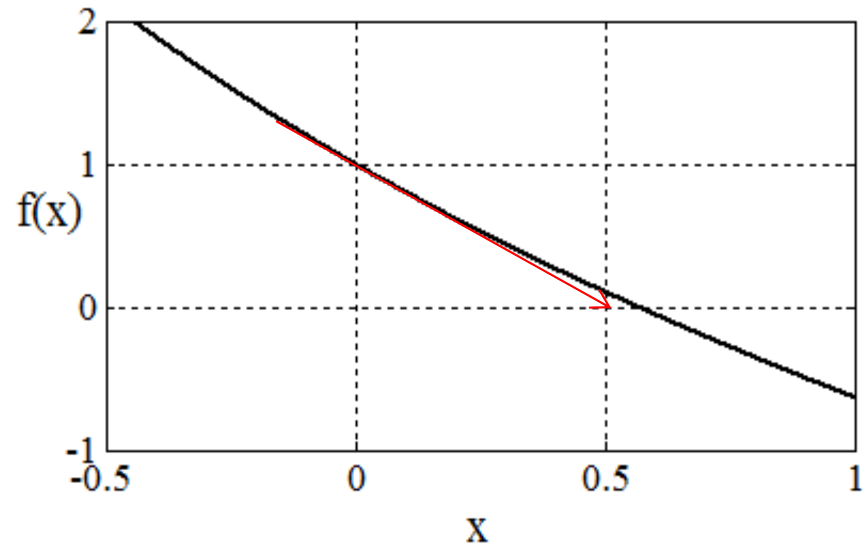
This is the Newton-Raphson formula.

Important: To use the Newton-Raphson formula we need to know $f(x_i)$ and $f'(x_i)$ exactly. (this works for functions we know and can take a derivative of).

Newton-Raphson Method Example

$$f(x) = e^{-x} - x$$

$$f'(x) = -e^{-x} - 1$$



Newton-Raphson formula.

$$x_{i+1} = x_i - \frac{e^{-x_i} - x_i}{-e^{-x_i} - 1}$$

New 'guess'.

Previous 'guess'.

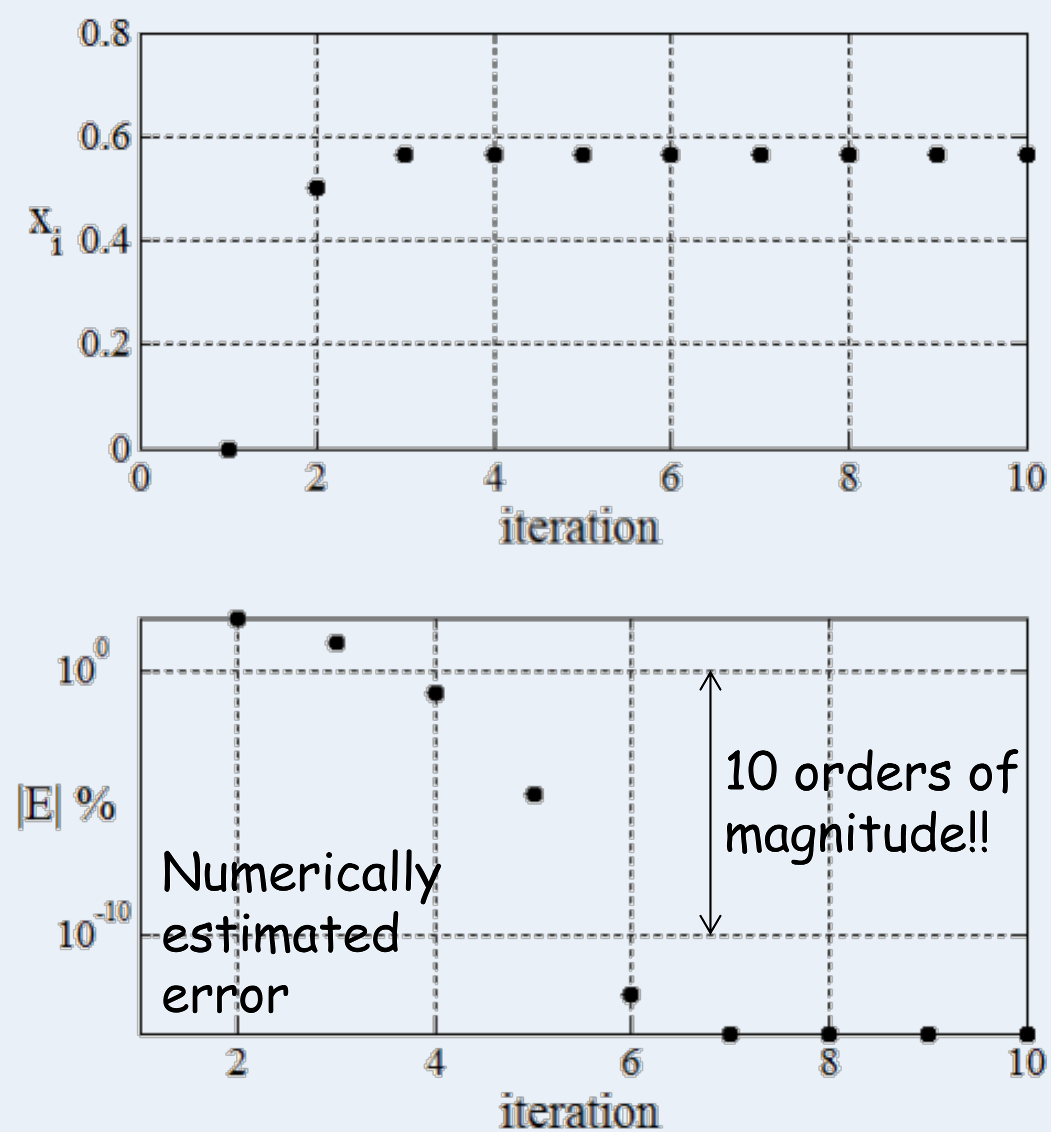
Iteration	x_i	|E| %
0	0	-
1	0.500000	11.8
2	0.566311	0.147
.	.	.
.	.	.
.	.	.

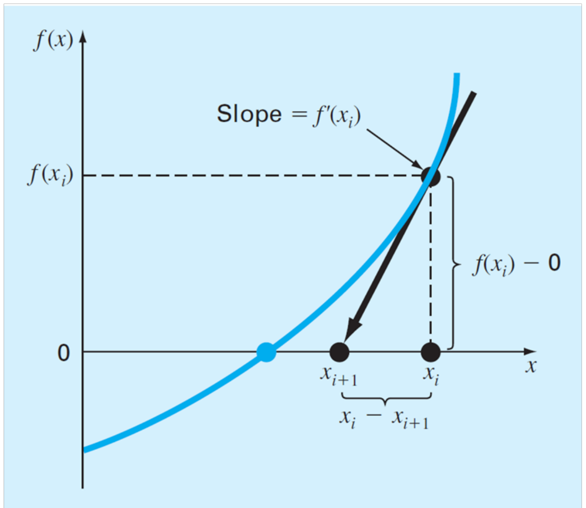
Newton-Raphson Method Error

$$E_{i+1} = -\frac{f''(x_r)}{2f'(x_r)} E_i^2$$

Error in the new estimate (i+1) is proportional to the square of the previous error → quadratic convergence!

Return to our example: $f(x) = e^{-x} - x$





Chapra, Figure 6.4

Secant Method

Newton-Raphson formula.

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

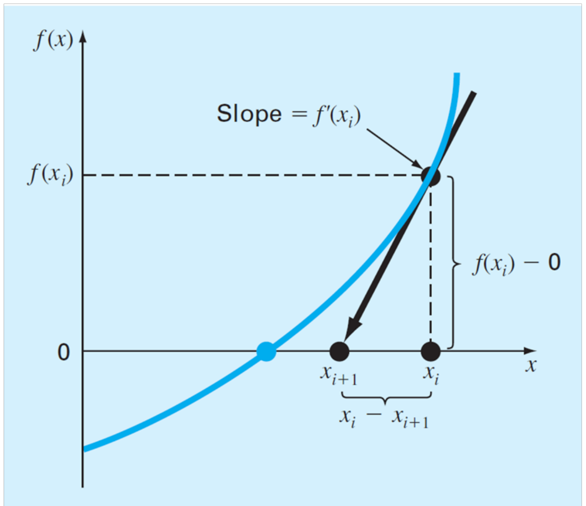
Question:. What if we don't know the derivative?

Answer: Approximate the derivative using a finite backward difference.

$$f'(x) \cong \frac{f(x_{i-1}) - f(x_i)}{x_{i-1} - x_i}$$

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \cong x_i - \frac{f(x_i)(x_{i-1} - x_i)}{(f(x_{i-1}) - f(x_i))}$$

Secant method



Chapra, Figure 6.4

Secant Method

Secant method

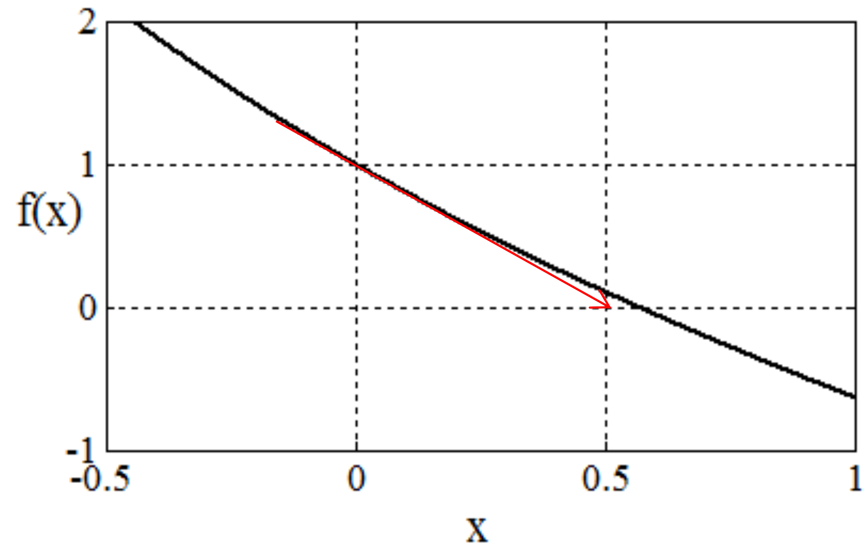
$$x_{i+1} = x_i - \frac{f(x_i)(x_{i-1} - x_i)}{\left(f(x_{i-1}) - f(x_i)\right)}$$

Downside: we now need to points instead of one (although they don't need to bracket the roots)

Suggested Approach: Make $x_{i+1} - x_i$ *small*. (but not so small that round-off error begins to hurt)

Secant Method Example

$$f(x) = e^{-x} - x$$



$$x_{i+1} = x_i - \frac{f(x_i)(x_{i-1} - x_i)}{(f(x_{i-1}) - f(x_i))}$$

$-(x_{i+1} - x_i) = dx = 1 \times 10^{-6}$

Iteration	x_i	|E| %
0	0	-
1	0.500000	11.8
2	0.566311	0.147
.	.	.
.	.	.
.	.	.

Similar results to Newton-Raphson, but we didn't need to find the derivative.

Take-home Messages

- Newton-Raphson and Secant methods are useful ways to find roots
  - Don't need to bracket the root
  - Fast convergence
  - If there is a root nearby N-R will converge to that root, but not necessarily the one you want
- Relatively easy to understand
- Function must be differentiable!
- It's always a good idea to try to find a good guess, either by plotting the function, or by looking at a simplified problem first, or from experiment, …