# Root Finding Methods: Newton-Raphson method and the Secant methods

Instructor: Ashley Gannon

Fall 2020

# Finishing up the Newton-Raphson method

# Pseudocode example

```
//Input:  a guess for x_0, x
//Output:  void, print the root to the screen

//declare/define variables for the Newton-Raphson method:
   //tolerance, error, Maxiter, iter, and x_new

loop while the number of iterations is less than the iteration cap,
iter<Maxiter, and the function value isn't close to 0, abs(f(x))>tol

   find the new value of x, x_new, using the Newton-Raphson formula

   update the counter, iter++

   compute the error, abs(x_new-x)

   if the error <tol
      break the loop
   else
      set x = new value of x for loop

print the root value to the screen
```

# Let's code it!

Take the next 20 minutes and try to write the Newton-Raphson method, `NewtonRaphson.cpp`. Recall that the Newton raphson formula is

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

If you finish writing it early, apply it to the function $f(m)$ from last lecture with an initial guess of 140.

$$f(m) = \sqrt{\frac{gm}{c_d}} \tanh\left(\sqrt{\frac{gc_d}{m}} t\right) - v(t)$$

with the derivative

$$f'(m) = \frac{1}{2}\sqrt{\frac{g}{mc_d}} \tanh\left(\sqrt{\frac{gc_d}{m}} t\right) - \frac{gt}{2m}\left(1 - \tanh^2\left(\sqrt{\frac{gc_d}{m}} t\right)\right)$$

Post your code to the discussion board **Newton Raphson method code**. Do NOT post screen shots of error messages or of your code. Copy and paste the text from your code into the discussion board.

# Newton-Raphson Method

A potential problem in implementing the Newton-Raphson method is the evaluation of the derivative. Although this is not inconvenient for polynomials and many other functions, there are certain functions whose derivatives may be difficult or inconvenient to evaluate. For these cases, the derivative can be approximated by a backward finite divided difference:

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

# Secant Methods

# Secant method

If we take the approximation for the derivative

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

and substitute it into our Newton-Raphson formula

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

we end up with

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

Equation is the formula for the *secant method*. Notice that the approach requires two initial estimates of $x$. However, because $f(x)$ is not required to change signs between the estimates, it is not classified as a bracketing method.

# Modified Secant method

Rather than using two arbitrary values to estimate the derivative, an alternative approach involves a perturbation of the independent variable to estimate $f'(x)$ by letting $x_{i-1} = x_i + \delta$

$$f'(x) \approx \frac{f(x_i + \delta) - f(x_i)}{\delta}$$

If we substitute this into our Newton-Raphson formula we end up with

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

We call this the *modified secant method*. It provides a nice means to attain the efficiency of the Newton-Raphson Method without having to compute derivatives.

# Convergence

# Convergence

convergence is the idea that if we apply enough iterations, we compute the root accurately

- it may never be exact, due to round-off error
- sometimes the method doesn't converge
- sometimes convergence is slow

For the methods that we've seen, they are known to have the following theoretical rates of convergence (for certain problems):

- Bisection method, $\alpha = 1$
- Secant method, $\alpha = 1.618$
- Newton's method, $\alpha = 2$