Class Prep 12 | 6.1.2 - 6.1.3

Riley Primeau

2022-11-28

Section 6.1.2 Netown-Raphson Method

```
newton = function(f, fp, x, tol = 1e-3, m = 100)
{
  iter = 0

  oldx = x
  x = oldx + 10 * tol

  while(abs(x-oldx) > tol)
  {
    iter = iter + 1
    if(iter > m)
      stop("No solution found")
    oldx = x
    x = x - f(x)/fp(x)
  }
  return(x)
}

f = function(x) {x^2 - 1}
fp = function(x) {2*x}
newton(f, fp, 1.25, tol = 1e-3)

## [1] 1

newton(f, fp, -1100, tol = 1e-6)

## [1] -1

newton(f, fp, 1e6, tol = 1e-9)

## [1] 1

f = function(x) {x^2 - 2*x + 1}
fp = function(x) {2*x - 2}
newton(f, fp, 1.25, tol = 1e-3)

## [1] 1.000508

newton(f, fp, -1100, tol = 1e-6)

## [1] 0.9999995
```

```
newton(f, fp, 1e6, tol = 1e-9)

## [1] 1

newton(f, fp, 0, tol = 1e-3)

## [1] 0.9990332

newton(sin, cos, 2, tol = 1e-6)

## [1] 3.141593

newton(sin, cos, pi, tol = 1e-6)

## [1] 3.141593

newton(sin, cos, pi/2, tol = 1e-6)

## [1] 99978.04

cos(pi/2)

## [1] 6.123032e-17
```

Section 6.1.3 Secant Method

```
secant = function(f, x, tol = 1e-3, m = 100)
{
  i = 0

  oldx = x
  oldfx = f(x)
  x = oldx + 10*tol

  while(abs(x - oldx) > tol)
  {
    i = i+1
    if(i>m)
      stop("No solution found")
    fx = f(x)
    newx = x - fx * ((x - oldx)/(fx - oldfx))
    oldx = x
    oldfx = fx
    x = newx
  }
  return(x)
}

f = function(x) {x^2 - 1}
secant(f, 1.25, tol = 1e-3)

## [1] 1

secant(f, -1100, tol = 1e-6)

## [1] -1

secant(f, 1e6, tol = 1e-9)

## [1] 1

secant(sin, 2, tol = 1e-6)

## [1] 3.141593

secant(sin, pi, tol = 1e-6)

## [1] 3.141593

#secant(sin, pi/2, tol = 1e-6) <- interesting that it throws an error now...
```