

Numerical Integration

Remember learning how to integrate in Calculus? Remember how most integrals do not have closed form solution?

Useless!

Instead, we will learn **numerical integration** and use it to solve integrals you couldn't solve in your calculus class, in a way that is easier than your calculus class was.

We will be looking at integrals of the form:

$$\int_b^a f(x) dx$$

Remann Sum

The **Remann Sum** is a numerical integration technique that approximates the area under a curve using infinitesimal rectangles.

Essentially, we evaluate:

$$\int_b^a f(x) dx = \sum_{n=0}^{n-1} f(x) \Delta x$$

Spreadsheet

As usual, the first approach is using a spreadsheet in Microsoft Excel. In this particular case, this tool works very poorly. The number of cells you need become unweildly.

22 March 2018

As we decrease the size of dx in our numerical integration, the accuracy of our numerical solution approaches the exact solution. If dx goes to zero, the solution becomes exact. Now we will create a program to do this.

```
def func(x):  
    ''' An abstract function we want to integrate. '''  
  
    return 0.5 * x**3 - 1.5 * x**2 + 12 * x - 13  
  
def integrate(func, a, b, dx=0.1):
```

```
''' A Remann integrating function. '''

x = np.arange(a, b, dx)
y = func(x)

return sum(y * dx)
```