Taylor Approximation

March 20, 2019

```
In [4]: import numpy as np
        import matplotlib.pyplot as plt
        %matplotlib inline
```

1 Define a differentiable function that maps from real number to real number.

Let's suppose a differentiable function is f(x) = sin(x)

```
In [7]: def f(x):
            return np.sin(x)
```

2 Define a domain of the function.

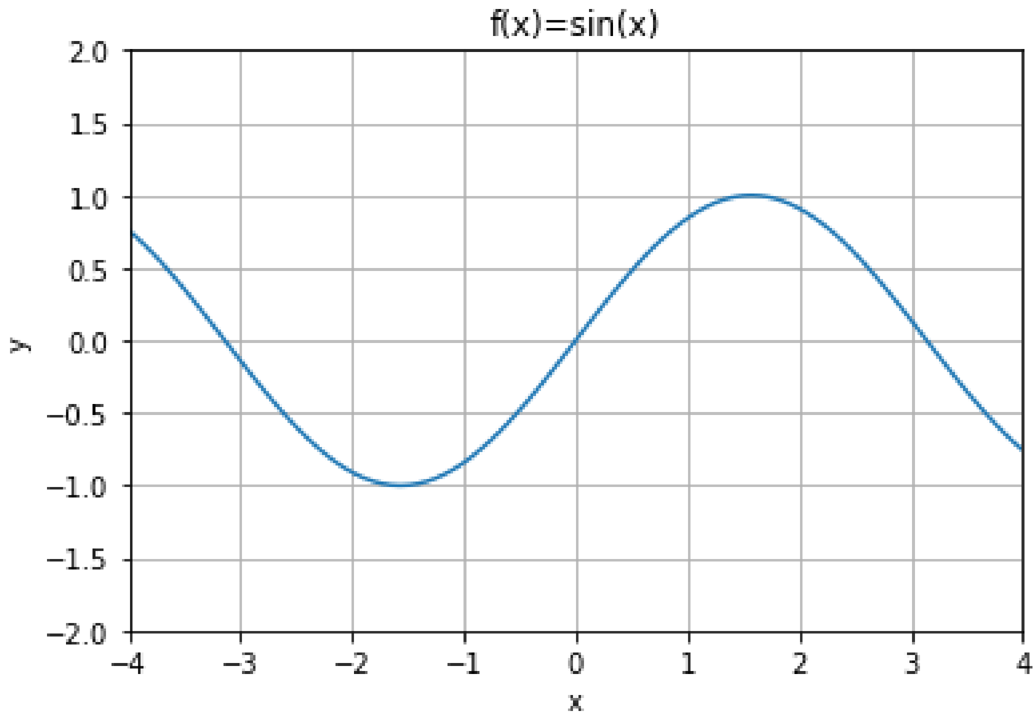
A domain of the function is from -4 to 4. x=[-4,4]

```
In [8]: x=np.linspace(-4,4,100)
```

3 Plot the function

Let's plot the sin(x).

```
In [9]: plt.xlim(-4,4)
        plt.ylim(-2,2)
        plt.plot(x,f(x))
        plt.xlabel('x')
        plt.ylabel('y')
        plt.grid(True)
        plt.title('f(x)=sin(x)')
        plt.show()
```



4 Select a point within the domain

The point that i choose is (0,0)

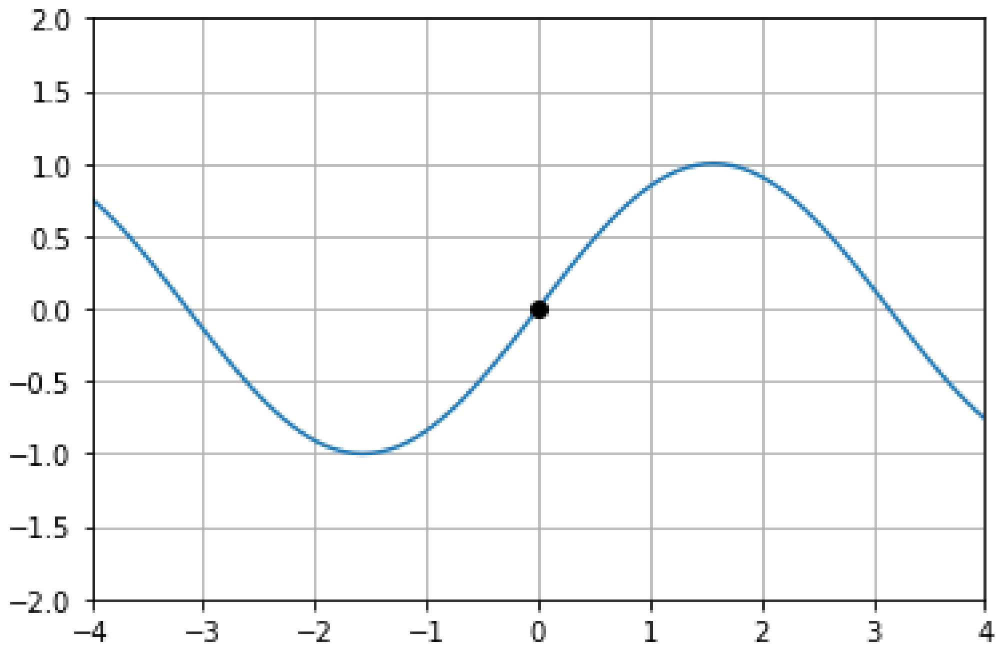
```
In [10]: w0 = 0
         w1 = 0
         print("Point= (",w0,",",w1,")")
```

```
Point= ( 0 , 0 )
```

5 Mark the selected point on the function

I marked the selected point on the f(x)=sin(x).

```
In [11]: plt.xlim(-4,4)
         plt.ylim(-2,2)
         plt.plot(x,f(x))
         plt.grid(True)
         plt.plot(0,0, marker='o', linestyle= 'None', markeredgecolor='black',color='black')
         plt.show()
```



6 Define the first-order Taylor approximation at the selected point.

This is a code to define the first-order Taylor approximation of sin(x) at x=0.

```
In [12]: def factorial(n):
             if n<=0:
                 return 1
             else:
                 return n*factorial(n-1)

         def taylor(x,n):
             i=0
             p=-1
             while i<=n:
                 p += 1/(factorial(i))*(x)**i
                 i+=1
             return p
```

First-order Taylor series of f(x)=sin(x) => y=x

7 Plot the Taylor approximation with the same domain of the original function

In [13]:
```
def plot():
    x_lims = [-5,5]
    x1 = np.linspace(x_lims[0],x_lims[1],800)
    y1 = []

    func = taylor(x,1)
    print("")

    plt.plot(x,taylor(x,1),label='$y=x$')
    plt.xlim(x_lims)
    plt.ylim([-2.5,2.5])
    plt.xlabel('x')
    plt.ylabel('y')
    plt.grid(True)
    plt.plot(x,f(x))
    plt.title('First-order Taylor series approximation of f(x)=sin(x)')
    plt.show()

plot()
```