assignment02

September 27, 2018

1 This script demonstrates the first order Taylor expansion of a given function

2 Name : Ji-Su Lee

3 Student ID : 20141718

4 import packages for plotting graphs and manipulating data:

```
In [3]: import numpy as np
        import matplotlib.pyplot as plt
```

5 define my function: $f(x) = \arcsin(x)$

```
In [4]: def myFunction(x):
            f = np.arcsin(x)
            return f
```

6 define the derivative of my function:

$f'(x) = \frac{1}{\sqrt{1-x^2}}$

```
In [5]: def myDerivativeFunction(x):
            Df = np.divide(1, np.sqrt(1-np.power(x, 2)))
            return Df
```

7 define the domain of the function:

$x = [-1 : 1 : 0.1]$

```
In [6]: x = np.arange(-1, 1, 0.1)
```

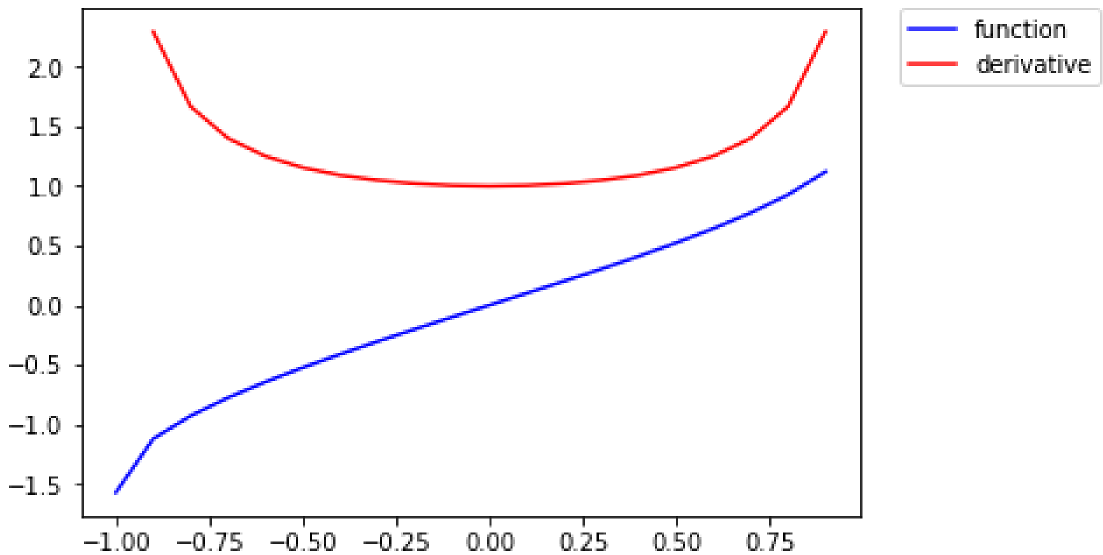
8 compute the graph

```
In [7]: f = myFunction(x)
        Df = myDerivativeFunction(x)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:2: RuntimeWarning: divide by z
```

9 plot the graphs for the function and its derivative

```
In [8]: plt.figure(1)
        plt.plot(x, f, 'b', label="function")
        plt.plot(x, Df, 'r', label="derivative")
        plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
        plt.show()
```



10 define Taylor approximation:

$\hat{f(x)} = f(x_0) + f'(x_0)(x - x_0)$

```
In [18]: def myTaylorApprox(x, p):
             TA = myFunction(p) + myDerivativeFunction(x) * (x - p)
             return TA
```

11 pick three points:

p1 = -0.5
p2 = 0
p3 = 0.5

```
In [24]: p1 = -0.5
         p2 = 0
         p3 = 0.5
```

12 define the domain of each approximations:

x1 = [-1 : 0.1 : 0]
x2 = [-0.5 : 0.1 : 0.5]
x3 = [0 : 0.1 : 1]

```
In [25]: x1 = np.arange(-1, 0, 0.1)
         x2 = np.arange(-0.5, 0.5, 0.1)
         x3 = np.arange(0, 1, 0.1)
```

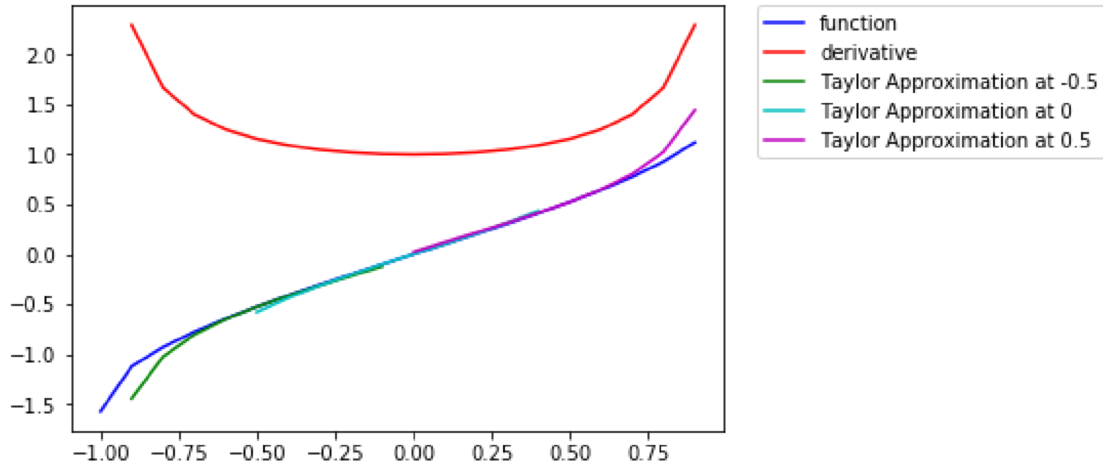
13 compute Taylor Approximations

```
In [26]: TA1 = myTaylorApprox(x1, p1)
         TA2 = myTaylorApprox(x2, p2)
         TA3 = myTaylorApprox(x3, p3)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:2: RuntimeWarning: divide by z
```

14 plot the graphs for the function, its derivative, and its Taylor expansions

```
In [27]: plt.figure(1)
         plt.plot(x, f, 'b', label="function")
         plt.plot(x, Df, 'r', label="derivative")
         plt.plot(x1, TA1, 'g', label="Taylor Approximation at -0.5")
         plt.plot(x2, TA2, 'c', label="Taylor Approximation at 0")
         plt.plot(x3, TA3, 'm', label="Taylor Approximation at 0.5")
         plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
         plt.show()
```


function
derivative
Taylor Approximation at -0.5
Taylor Approximation at 0
Taylor Approximation at 0.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75