HW04 Derivative02

September 20, 2024

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

#Activity 1

def Numerical_Point_Derivative():
    h = 1e-5
    #User inputs function and point 'a'
    func = input("Enter an algebraic function in terms of x ('x**2', 'math.
↪sin(x)', '3*x**4 - 2*x', etc.): ")
    a = float(input("Enter the point 'a' where you want to find the slope: "))

    #Convert the input into a lambda function
    f = lambda x: eval(func)

    #Calculate the derivative
    q = (f(a + h) - f(a)) / h
    #This formula worked better in the last homework, with less deviation
    #with the other method, as h decreases, it can have overshoot or more␣
↪deviation

    return q
```

[2]:
```
#Testing with function/point examples from #10
#10a
Numerical_Point_Derivative()
```

```
Enter an algebraic function in terms of x ('x**2', 'math.sin(x)', '3*x**4 -
2*x', etc.):  1/x
Enter the point 'a' where you want to find the slope:  2
```

[2]:
```
-0.24999875000708546
```

[6]:
```
#10b
Numerical_Point_Derivative()
```

```
Enter an algebraic function in terms of x ('x**2', 'math.sin(x)', '3*x**4 -
```

2*x', etc.): math.sin(7*x)
Enter the point 'a' where you want to find the slope: 3

[6]: -3.834309799122248

[7]:
```
#10c
Numerical_Point_Derivative()
```

Enter an algebraic function in terms of x ('x**2', 'math.sin(x)', '3*x**4 - 2*x', etc.): x**3
Enter the point 'a' where you want to find the slope: 200

[7]: 120000.00607222317

[8]:
```
#10d
Numerical_Point_Derivative()
```

Enter an algebraic function in terms of x ('x**2', 'math.sin(x)', '3*x**4 - 2*x', etc.): 2**x
Enter the point 'a' where you want to find the slope: 5

[8]: 22.18078664952827

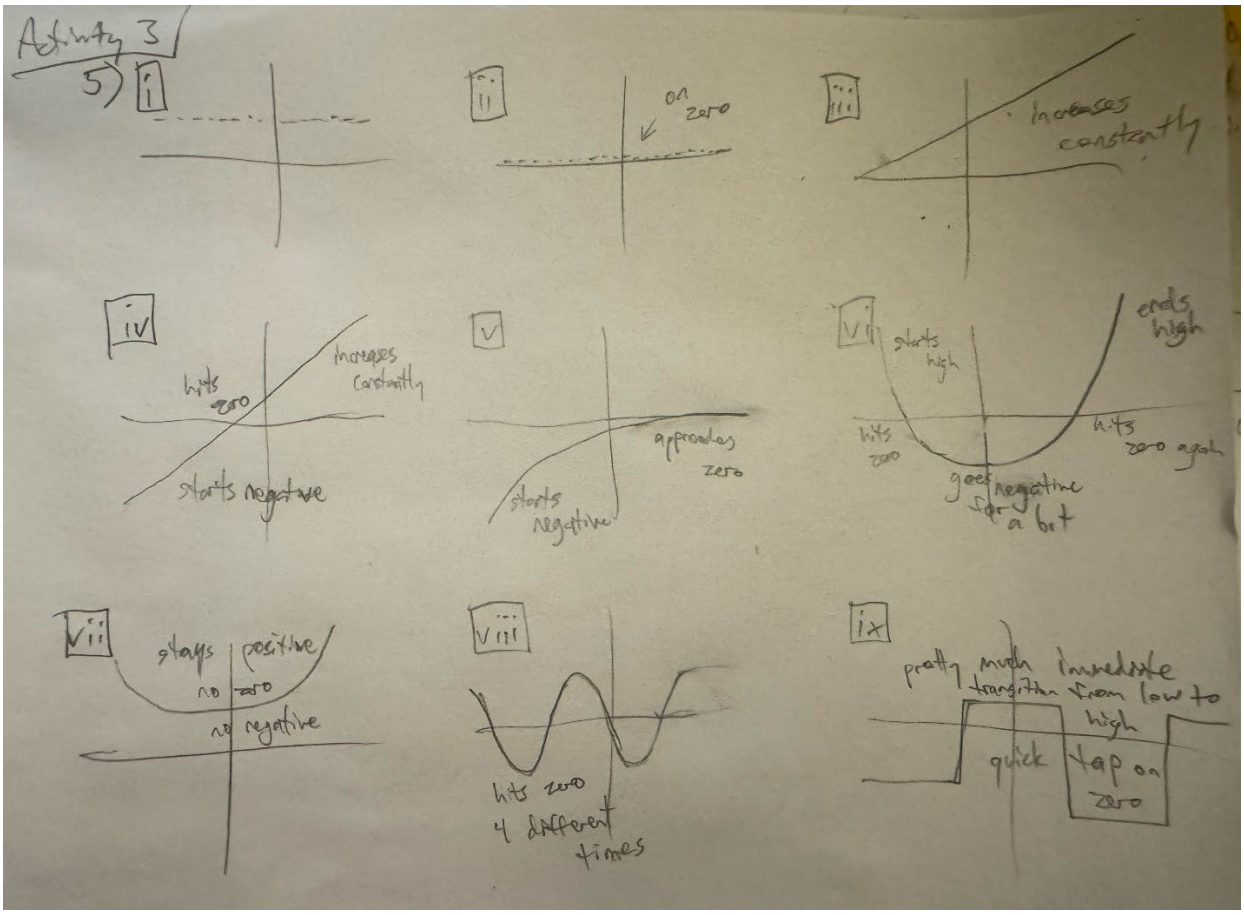
Activity 2

I used the Numerical Derivative Calculator (NDC) from Wolfram Alpha (WA) and my answers were pretty darn close. WA answers tended to round a bit, while these answers have lots of digits past the decimal point. WA also showed answers in an expression if the answer included any trigonometric function like sin, cos, etc. To make sure my answers were right, I just put the expression into this notebook and made sure the number shown was close to mine. They all were, so I think my calculator works pretty well. 4/4 solid.

Activity 3:

2) a = s, b = c(t), c = s', d = c'(t)

Activity 3 Question 5

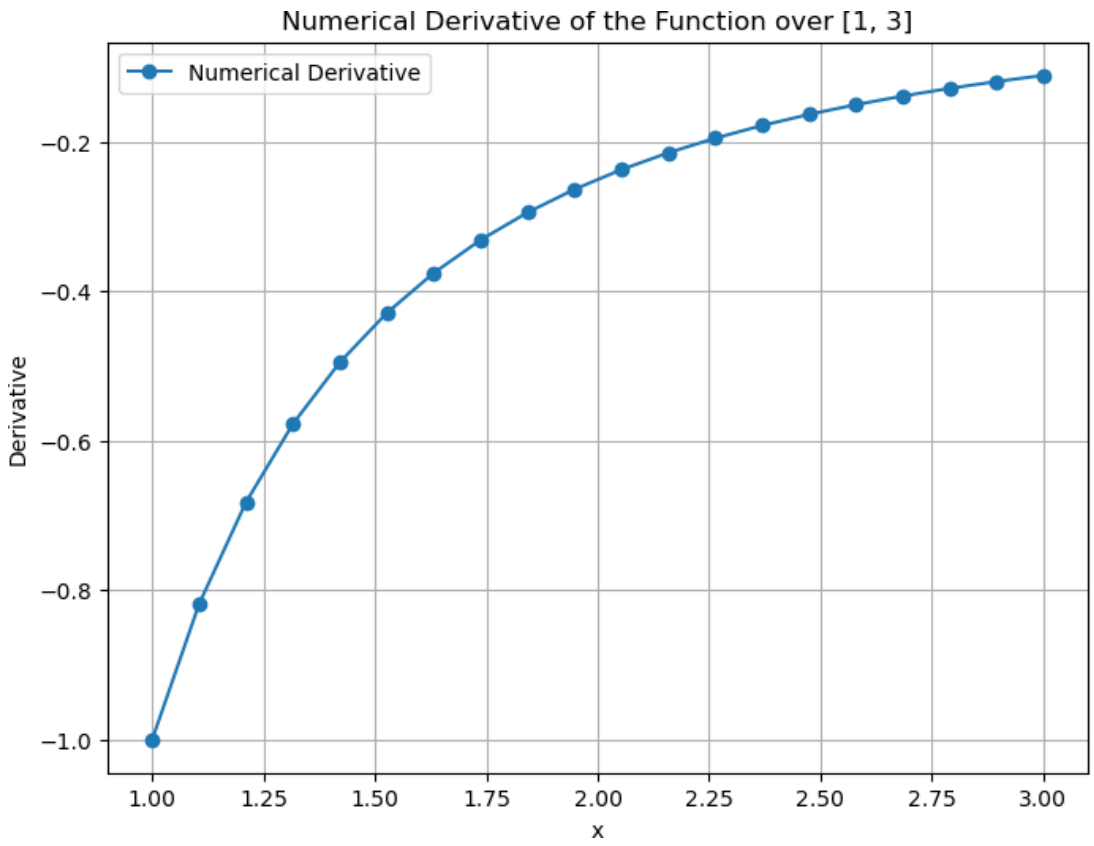


[3]:
```
#Activity 4

def Numerical_Function_Derivative():
    h = 1e-5
    #User inputs function
    func = input("Enter an algebraic function in terms of x ('x**2', 'math.
↪sin(x)', '3*x**4 - 2*x', etc.): ")


    #Convert the input into a lambda function
    f = lambda x: eval(func)
```

```
    #generate 20 points from 1 to 3
    Xvals = np.linspace(1, 3, 20)
    DeriVals = []
    #Calculate the derivative
    for a in Xvals:
        q = (f(a + h) - f(a)) / h
        DeriVals.append(q)

    #plot results using matplotlib
    plt.figure(figsize=(8, 6))
    plt.plot(Xvals, DeriVals, marker='o', label='Numerical Derivative')
    plt.title("Numerical Derivative of the Function over [1, 3]")
    plt.xlabel("x")
    plt.ylabel("Derivative")
    plt.grid(True)
    plt.legend()
    plt.show()
```

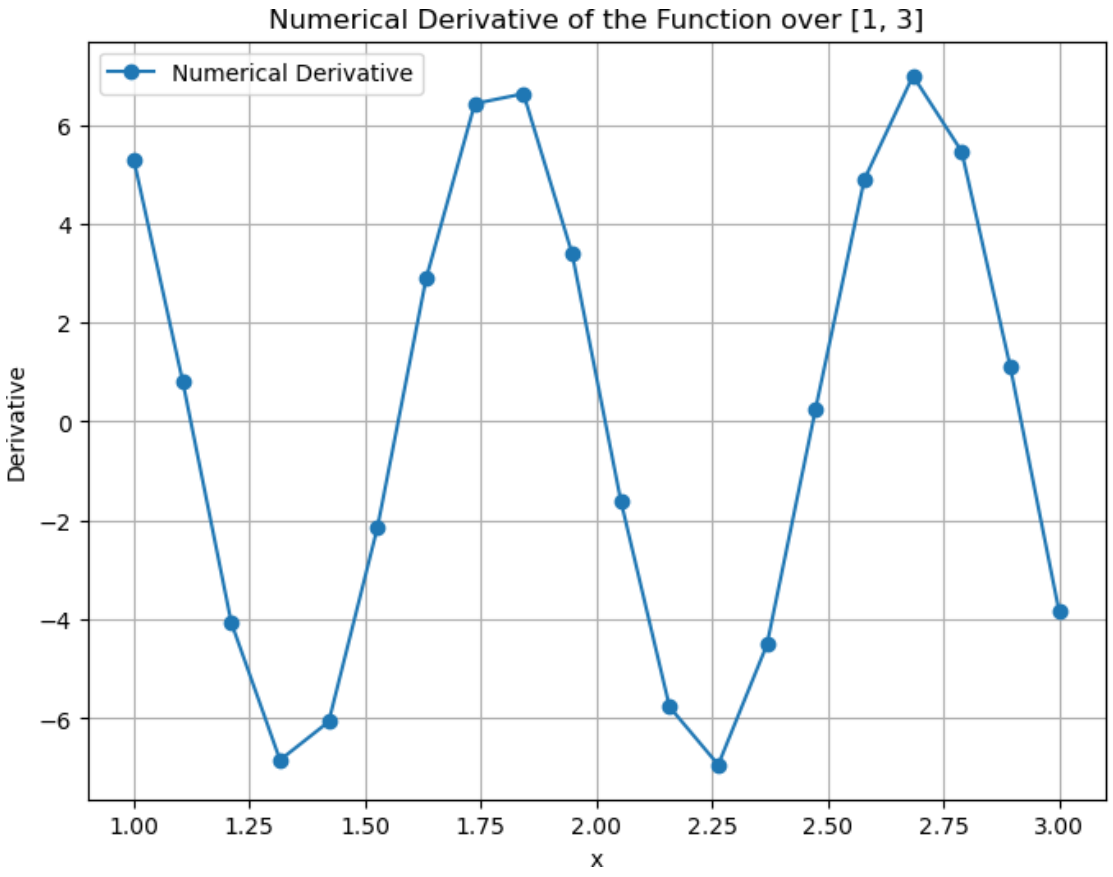
[15]:
```
Numerical_Function_Derivative()
```

```
Enter an algebraic function in terms of x ('x**2', 'math.sin(x)', '3*x**4 - 2*x', etc.):  1/x
```



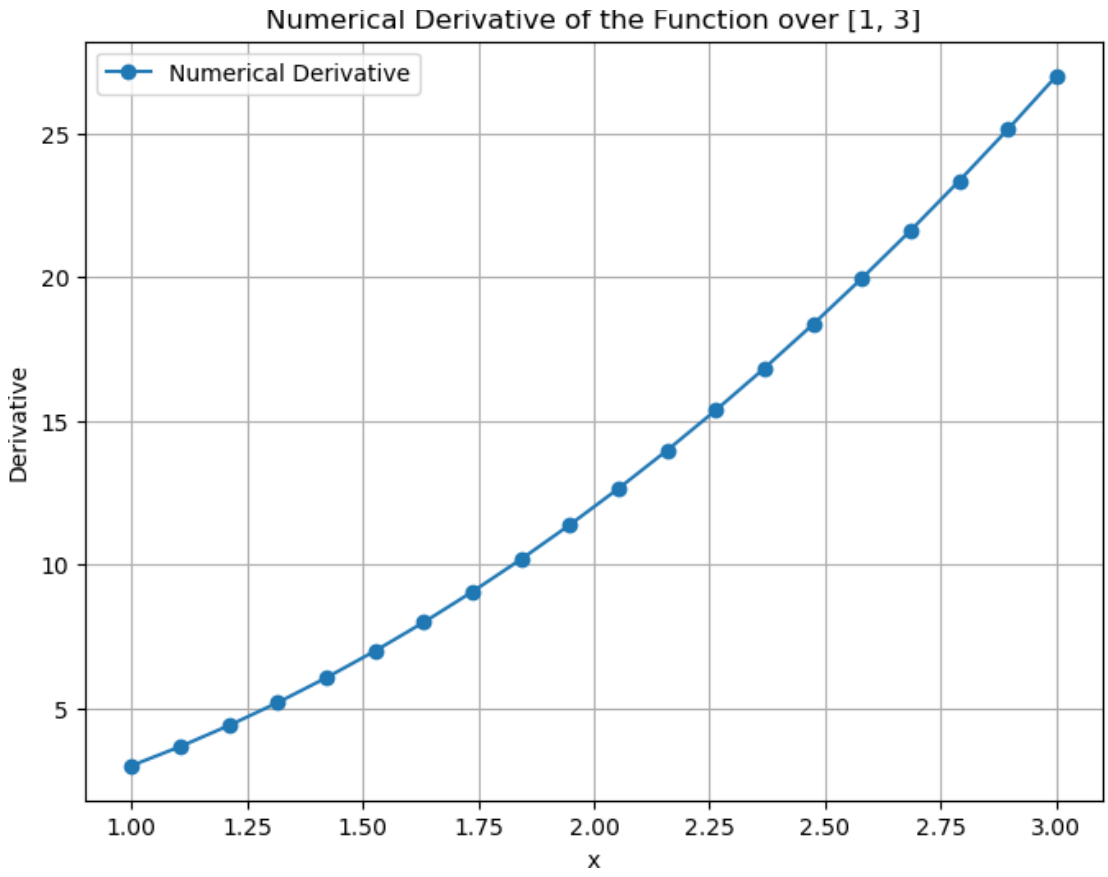
[16]: Numerical_Function_Derivative()

Enter an algebraic function in terms of x ('x**2', 'math.sin(x)', '3*x**4 - 2*x', etc.): math.sin(7*x)



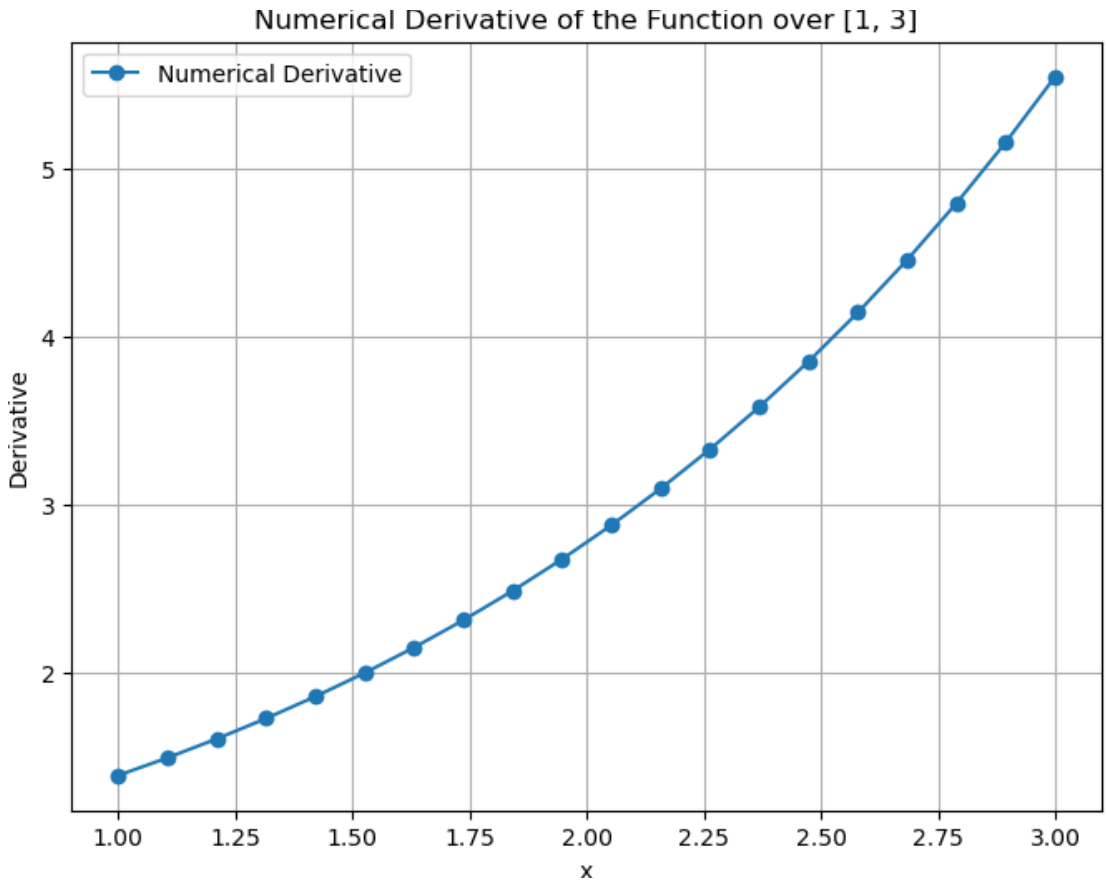
[17]: Numerical_Function_Derivative()

Enter an algebraic function in terms of x ('x**2', 'math.sin(x)', '3*x**4 - 2*x', etc.): x**3



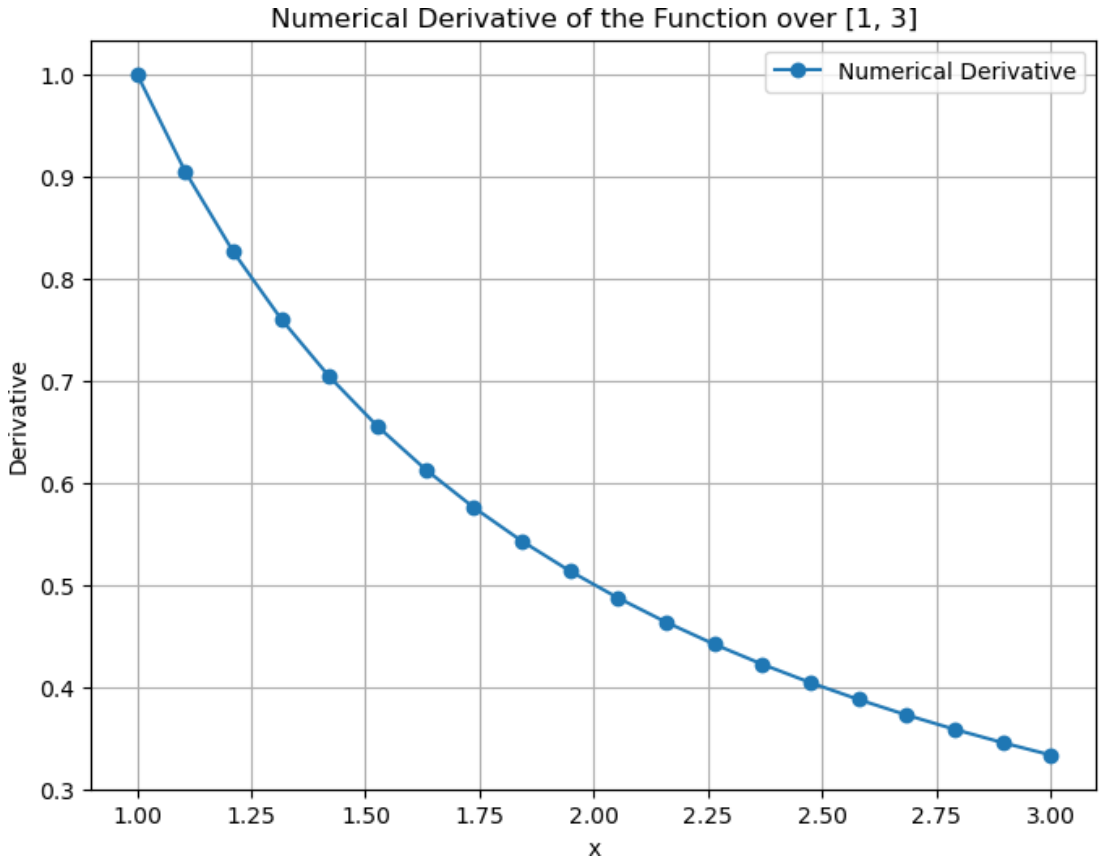
[18]:
```
Numerical_Function_Derivative()
```

Enter an algebraic function in terms of x ('x**2', 'math.sin(x)', '3*x**4 - 2*x', etc.): 2**x



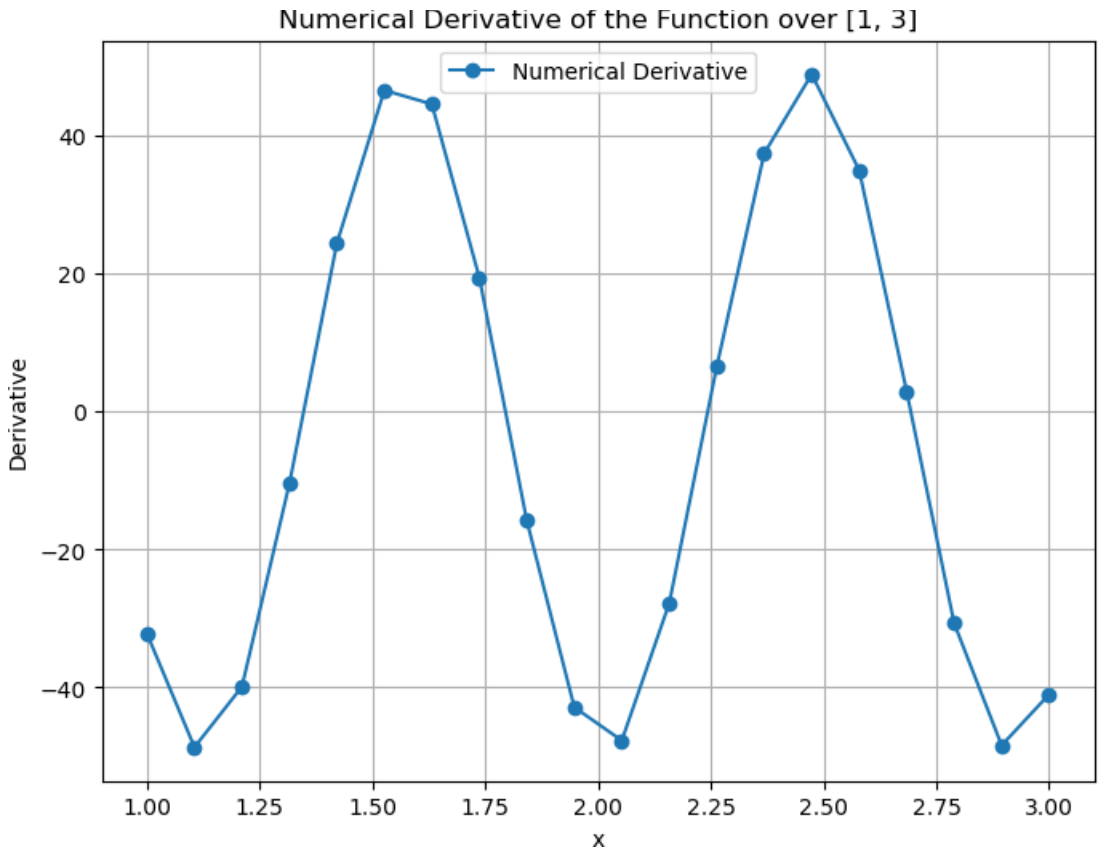
[4]:
```
#Activity 5
Numerical_Function_Derivative()
```

Enter an algebraic function in terms of x ('x**2', 'math.sin(x)', '3*x**4 - 2*x', etc.): math.log(x)



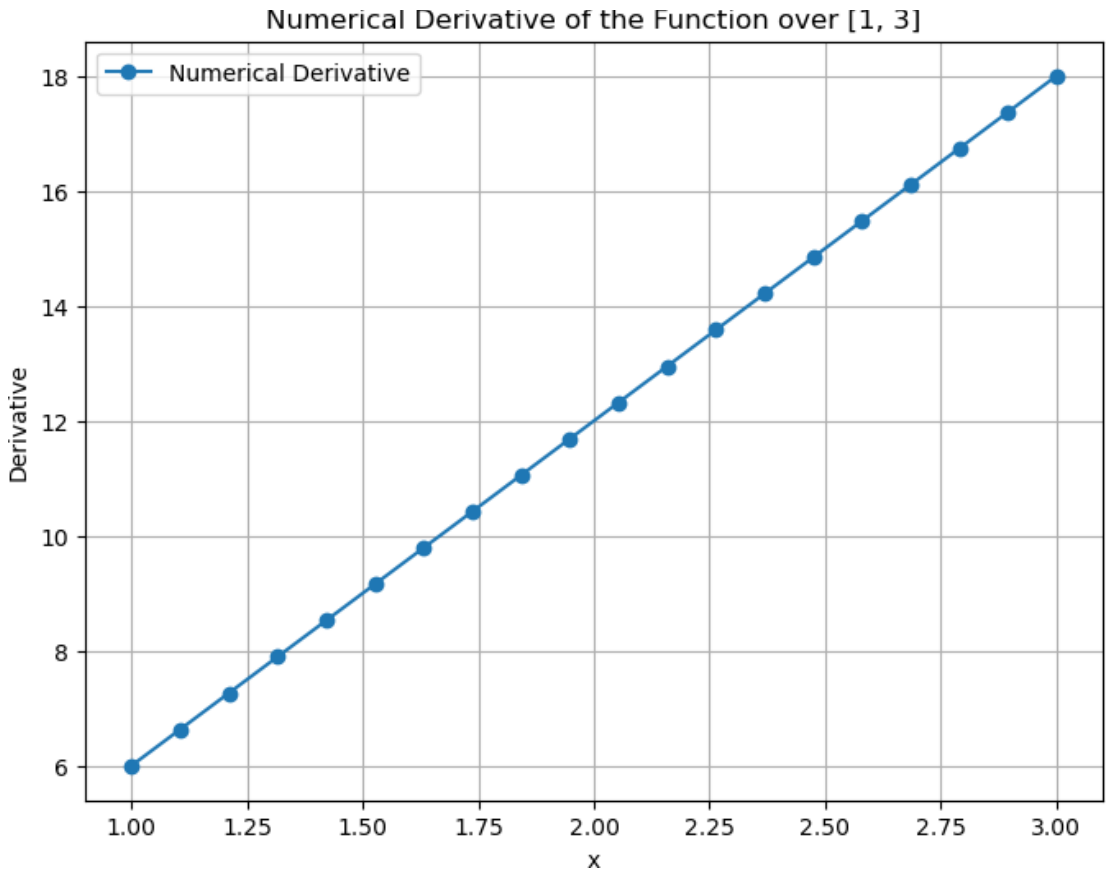
[5]:
```
Numerical_Function_Derivative()
```

```
Enter an algebraic function in terms of x ('x**2', 'math.sin(x)', '3*x**4 -
2*x', etc.):     7*math.cos(7*x)
```



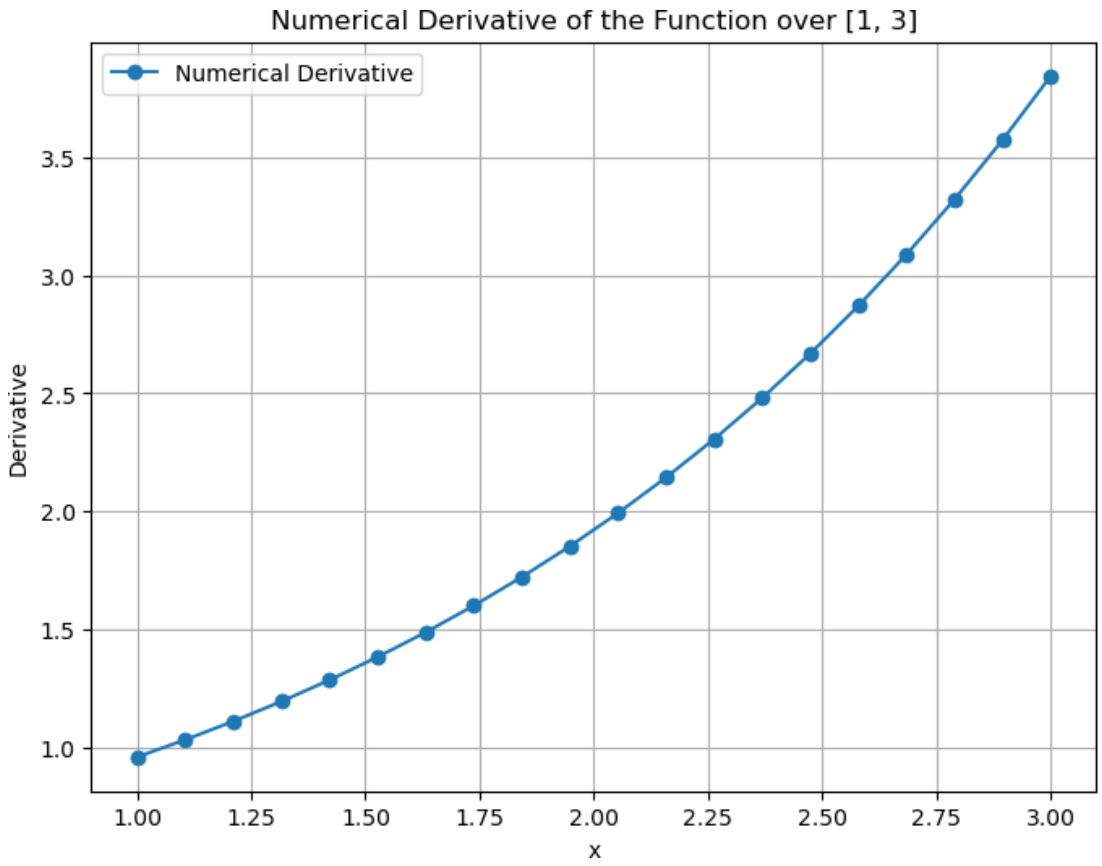
[6]:
```
Numerical_Function_Derivative()
```

```
Enter an algebraic function in terms of x ('x**2', 'math.sin(x)', '3*x**4 -
2*x', etc.):     3*x**2
```



[7]:
```
Numerical_Function_Derivative()
```

```
Enter an algebraic function in terms of x ('x**2', 'math.sin(x)', '3*x**4 -
2*x', etc.):     math.log(2) * 2**x
```



Activity 5 continued

I used Wolfram Alpha again and compared the graphs that I have here to the ones WA creates and they are pretty much the exact same, so I'm kinda proud of myself with these activities.

[ ]: