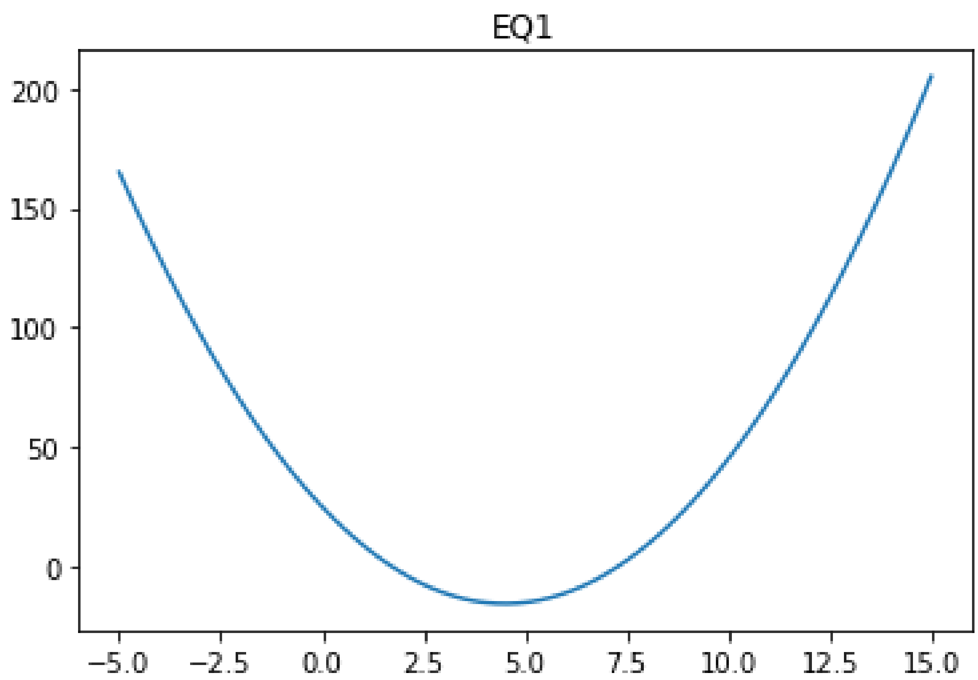
Finding the numerical value of a derivative

May 6, 2022

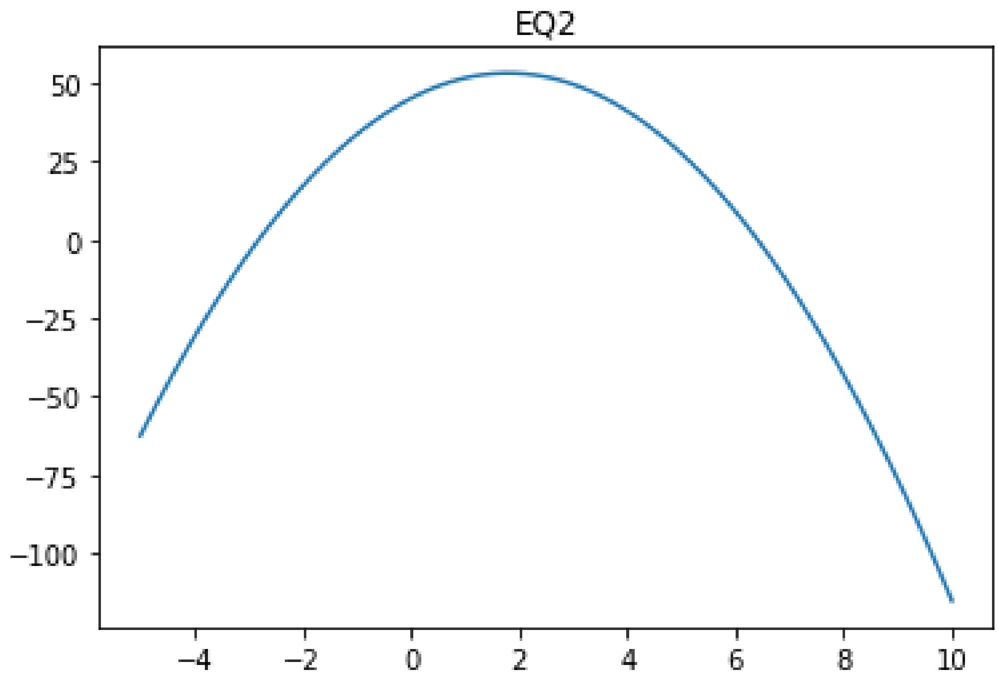
[1]:
```python
# importing required packages
import numpy as np
import matplotlib.pyplot as plt
```

[17]:
```python
# Equation 1 and its plotting
x1 = np.linspace(-5, 15, num=300)
eq1 =2*(x1**2)-(18*x1)+25
plt.plot(x1,eq1)
plt.title("EQ1")
plt.show()
```



[18]:
```python
# Equation 2 and its plotting
x2 = np.linspace(-5, 10, num=300)
eq2 =-2.5*(x2**2)+ (9*x2)+45
plt.plot(x2,eq2)
```

```
plt.title("EQ2")
plt.show()
```



[4]:
```
# definition of equation 1
def _eq1(x):
    return 2*(x**2) - (18*x) + 25
```

[5]:
```
# definition of equation 2
def _eq2(x):
    return -2.5*(x**2) + (9*x) + 45
```

[6]:
```
# it returns f of x
def fnc(f,x):
    y = f(x)
    return y
```

[7]:
```
# controlling the function
print(fnc(_eq1,0))
print(fnc(_eq2,0))
```

```
25
45.0
```

If x = 0 at equations 1 and 2, the f of x must be 25 and 45 respectively.

```
[8]: # definition of method-related function

#-------------------descriptions begin
# a is the function input value that use to find the derivative of the function␣
↪at point a.
# eq is function and h is step size
#-------------------descriptions end

def central_dif_2(eq,a,h):
    Jp1 = a+h
    Jm1 = a-h
    d1_fx = ( fnc(eq,Jp1)-fnc(eq,Jm1) ) / (2*h) # +O()
    return d1_fx
```

For more general knowledge, just search “Numerical Differentiation”

For knowledge of “central_dif_2”, just search like “Central-difference formulas”

```
[13]: # step size
h = 0.0001

# dif point
a = 1.9

# dif around a
d1_fx = central_dif_2(_eq2,a,h)
print("The value of the first derivative of the equation at point " + str(a) + "␣
↪is: " + str(d1_fx))
```

```
The value of the first derivative of the equation at point 1.9 is:
-0.49999999998107114
```

```
[14]: # the first derivative of the equation 2 = -5*x +9

# x = 5
# d1_eq2 = -5*x + 9

d1_eq2 = -5*a + 9
print("Exact solution is: " + str(d1_eq2))
```

```
Exact solution is: -0.5
```

```
[15]: # step size
h = 0.001

# dif point
a = 5

# dif around a
```

```
d1_fx = central_dif_2(_eq1,a,h)
print(d1_fx)
```

2.000000000005997

[16]:
```
# the first derivative of the equation 1 = 4*x - 18

# x = 5
# d1_eq1 = 4*x - 18

d1_eq1 = 4*a - 18
print("Exact solution is: " + str(d1_eq1))
```

Exact solution is: 2

[ ]:
```

```