HW07 - Newton's Method

October 27, 2024

[9]:
```
pip install -U matplotlib
```

```
Requirement already satisfied: matplotlib in /opt/conda/lib/python3.10/site-
packages (3.6.2)
Collecting matplotlib
  Downloading
matplotlib-3.9.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (8.3
MB)
                           8.3/8.3 MB
88.4 MB/s eta 0:00:00:00:0100:01
Requirement already satisfied: contourpy>=1.0.1 in
/opt/conda/lib/python3.10/site-packages (from matplotlib) (1.0.6)
Requirement already satisfied: fonttools>=4.22.0 in
/opt/conda/lib/python3.10/site-packages (from matplotlib) (4.38.0)
Requirement already satisfied: packaging>=20.0 in
/opt/conda/lib/python3.10/site-packages (from matplotlib) (22.0)
Requirement already satisfied: kiwisolver>=1.3.1 in
/opt/conda/lib/python3.10/site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.10/site-
packages (from matplotlib) (0.11.0)
Requirement already satisfied: numpy>=1.23 in /opt/conda/lib/python3.10/site-
packages (from matplotlib) (1.23.5)
Requirement already satisfied: python-dateutil>=2.7 in
/opt/conda/lib/python3.10/site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: pyparsing>=2.3.1 in
/opt/conda/lib/python3.10/site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: pillow>=8 in /opt/conda/lib/python3.10/site-
packages (from matplotlib) (9.2.0)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.10/site-
packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
Installing collected packages: matplotlib
  Attempting uninstall: matplotlib
    Found existing installation: matplotlib 3.6.2
    Uninstalling matplotlib-3.6.2:
      Successfully uninstalled matplotlib-3.6.2
Successfully installed matplotlib-3.9.2
Note: you may need to restart the kernel to use updated packages.
```

[6]:
```python
import matplotlib
print(matplotlib.__version__)
```

3.9.2

[7]:
```python
#Activity 5

def f(x):
    return x**2 / 4 + x/4 - 5

def fp(x, h=0.0001):
    return (f(x+h) - f(x)) / h

def newton(start, steps):
    x = start
    for n in range(steps+1):
        print(n, x)
        fx = f(x)
        fpx = fp(x)
        if fpx == 0:
            print(f"Error at step {n}: derivative is zero at x = {x}. Newton's␣
↪method cannot continue.")
            return
        x = x - fx/fpx
```

[8]:
```python
#Activity 6

for start in range(8):
    newton(start, 8)
```

```
0 0
1 19.998000199861156
2 10.242974519545106
3 5.813984277807521
4 4.260587780625367
5 4.007134763587546
6 4.000005726227173
7 4.000000000067267
8 4.000000000000001
0 1
1 6.99980000666388
2 4.599952002495357
3 4.035294339131196
4 4.000137720502819
5 4.000000003637513
6 4.000000000000041
7 4.0
```

```
8 4.0
0 2
1 4.799944001111923
2 4.060376521103884
3 4.0004003311898995
4 4.0000000222531265
5 4.000000000000248
6 4.0
7 4.0
8 4.0
0 3
1 4.142840816558176
2 4.0021988222168625
3 4.000000561352948
4 4.000000000006272
5 4.0
6 4.0
7 4.0
8 4.0
0 4
1 4.0
2 4.0
3 4.0
4 4.0
5 4.0
6 4.0
7 4.0
8 4.0
0 5
1 4.0909173552953355
2 4.000901232418823
3 4.000000100239153
4 4.000000000001115
5 4.0
6 4.0
7 4.0
8 4.0
0 6
1 4.307705325336159
2 4.009850058017163
3 4.00001086594466
4 4.000000000133849
5 4.000000000000002
6 4.0
7 4.0
8 4.0
0 7
1 4.600015999894749
```

```
2 4.035301425680015
3 4.000137775519423
4 4.0000000036398085
5 4.000000000000041
6 4.0
7 4.0
8 4.0
```

[9]:

```
#Activity 6 continued

for start in range(-8, 8):
    newton(start, 8)
```

```
0 -8
1 -5.5999839998921
2 -5.035286809761855
3 -5.000136887037603
4 -5.000000000561029
5 -4.999999999999994
6 -5.0
7 -5.0
8 -5.0
0 -7
1 -5.307679289831045
2 -5.009842249863545
3 -5.000010630828618
4 -4.999999999894436
5 -5.000000000000002
6 -5.0
7 -5.0
8 -5.0
0 -6
1 -5.090900826368347
2 -5.000898947761684
3 -5.000000079786312
4 -4.9999999999991145
5 -5.0
6 -5.0
7 -5.0
8 -5.0
0 -5
1 -5.0
2 -5.0
3 -5.0
4 -5.0
5 -5.0
6 -5.0
7 -5.0
```

```
8 -5.0
0 -4
1 -5.142873469619209
2 -5.0021967818833515
3 -5.000000511552891
4 -4.999999999994345
5 -5.0
6 -5.0
7 -5.0
8 -5.0
0 -3
1 -5.800056001111286
2 -5.060378195718848
3 -5.0003990380232795
4 -5.000000013257587
5 -4.999999999999853
6 -5.0
7 -5.0
8 -5.0
0 -2
1 -8.000200006697423
2 -5.600048002492011
3 -5.03529389622655
4 -5.000136941887556
5 -5.000000000562087
6 -4.999999999999994
7 -5.0
8 -5.0
0 -1
1 -21.002000200232136
2 -11.24483055760317
3 -6.814708267018109
4 -5.260741316802101
5 -5.007137614647464
6 -5.000005572532187
7 -4.999999999941533
8 -5.000000000000001
0 0
1 19.998000199861156
2 10.242974519545106
3 5.813984277807521
4 4.260587780625367
5 4.007134763587546
6 4.000005726227173
7 4.000000000067267
8 4.000000000000001
0 1
1 6.99980000666388
```

```
2 4.599952002495357
3 4.035294339131196
4 4.000137720502819
5 4.000000003637513
6 4.000000000000041
7 4.0
8 4.0
0 2
1 4.799944001111923
2 4.060376521103884
3 4.0004003311898995
4 4.0000000222531265
5 4.000000000000248
6 4.0
7 4.0
8 4.0
0 3
1 4.142840816558176
2 4.0021988222168625
3 4.000000561352948
4 4.000000000006272
5 4.0
6 4.0
7 4.0
8 4.0
0 4
1 4.0
2 4.0
3 4.0
4 4.0
5 4.0
6 4.0
7 4.0
8 4.0
0 5
1 4.0909173552953355
2 4.000901232418823
3 4.000000100239153
4 4.000000000001115
5 4.0
6 4.0
7 4.0
8 4.0
0 6
1 4.307705325336159
2 4.009850058017163
3 4.00001086594466
4 4.000000000133849
```

```
5 4.000000000000002
6 4.0
7 4.0
8 4.0
0 7
1 4.600015999894749
2 4.035301425680015
3 4.000137775519423
4 4.0000000036398085
5 4.000000000000041
6 4.0
7 4.0
8 4.0
```

Activity 7 For this we have to set f'(x) equal to zero

We know f(x) = $\frac{x^2}{4} + \frac{x}{4} - 5$

So f'(x) = $\frac{x}{2} + \frac{1}{4}$

Setting f'(x) to zero: $\frac{x}{2} + \frac{1}{4} = 0$

$\frac{x}{2} = -\frac{1}{4}$

$x = -\frac{1}{2}$

[10]:
```
#Activity 7 continued

newton(-0.5, 25)
```

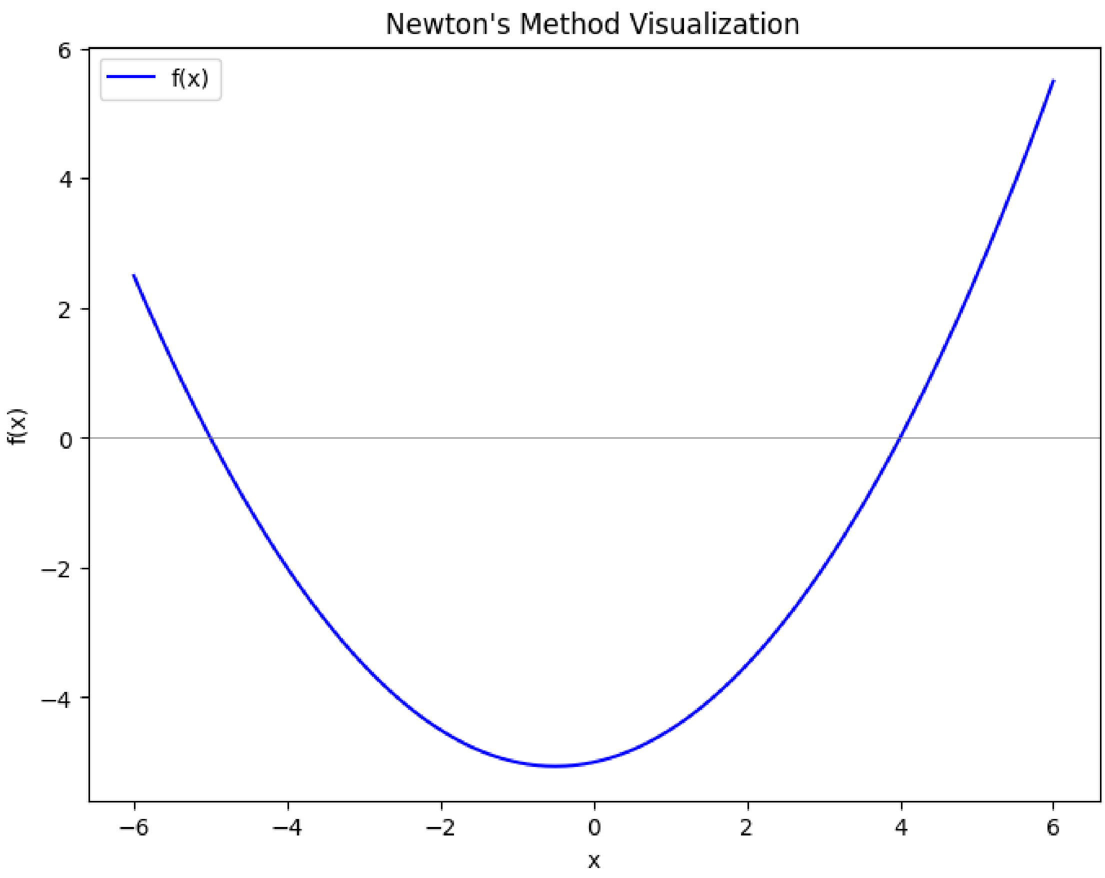
```
0 -0.5
1 202499.48324507626
2 101249.50004999932
3 50624.509636741364
4 25312.003776278594
5 12655.752486162497
6 6327.626992092744
7 3163.5651244543137
8 1581.535786445191
9 790.52431719914
10 395.0249834628322
11 197.28811557639105
12 98.44527393439235
13 49.07499120924026
14 24.491756442212875
15 12.401036000005227
16 6.735360650087943
17 4.517073030397821
18 4.0266503556348585
19 4.000078744608138
```

```
20 4.000000001563863
21 4.000000000000018
22 4.0
23 4.0
24 4.0
25 4.0
```

[11]:

```python
#Activity 8
import numpy as np
from matplotlib import pyplot as plt
from matplotlib.animation import FuncAnimation

start = -4
steps = 8
x_vals = [start]
for n in range(steps+1):
    x = start
    fx = f(x)
    fpx = fp(x)
    if fpx == 0:
        print(f"Error at step {n}: derivative is zero at x = {x}.")
    x_vals.append(x)
    x = x - fx / fpx

#setting up the plot parameters
x_range = np.linspace(-6, 6, 400)
fig, ax = plt.subplots(figsize=(8, 6))
ax.plot(x_range, f(x_range), label="f(x)", color="blue")  #plotting f(x)
ax.axhline(0, color="gray", linewidth=0.5)  #x-axis

#initializing points and lines for animation
point, = ax.plot([], [], 'ro')  #point for current x
line, = ax.plot([], [], 'r-', linewidth=0.75)  #tangent line

#animating initializing function
def init():
    point.set_data([], [])
    line.set_data([], [])
    return point, line

#animating updating function
def update(n):
    x = x_vals[n]
    y = f(x)

    #updating point for current x
    point.set_data(x, y)
```

```
    #calculating tangent line at point
    slope = fp(x)
    tang_x = np.array([x - 1, x + 1])  #tangent line range around x
    tang_y = f(x) + slope * (tang_x - x)
    line.set_data(tang_x, tang_y)

    return point, line

#running animation
ani = FuncAnimation(fig, update, frames=len(x_vals), init_func=init, blit=True,␣
 ↪repeat=False)

#displaying animation
plt.title("Newton's Method Visualization")
plt.xlabel("x")
plt.ylabel("f(x)")
plt.legend()
plt.show()
```



```
[ ]:
```

```
[ ]:
```