Week 3

September 17, 2024

1 Week 3 Activities

1.1 1: Graphing Example/Computer Microscope

For this question I used Desmos to model Question 2 part d on page 115 of the text book. I have attached 3 images I used to find the window in which the slope of the function $y = t + 2^{-t}$ appeared constant around $t = 7$. Because this slope became constant pretty quickly, I did not try a large variety of windows.

Image 1: The whole function, the slope at the point is not necessarily visible as there may be something missing at the point. However, my estimate for $f'(7) = 1$

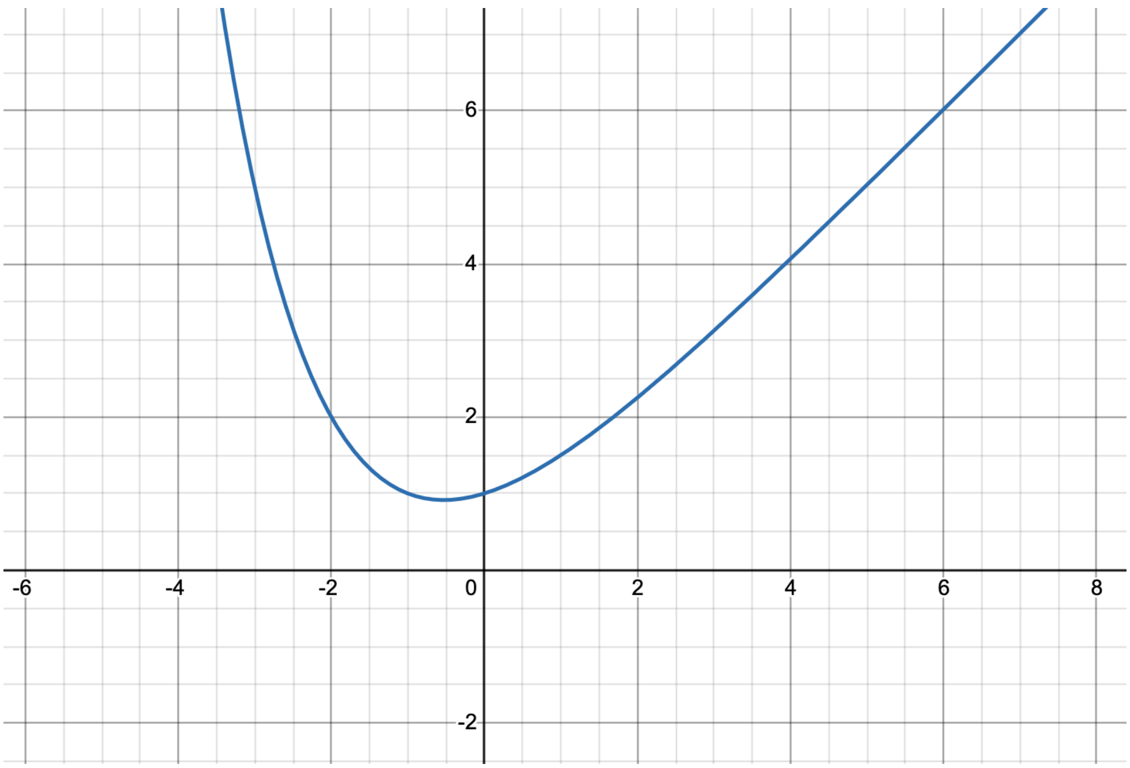


Image 2: $\Delta x = 2$ For this interval, the slope still appears to be $f'(7) = 1$, but in order to confirm I will zoom in

further

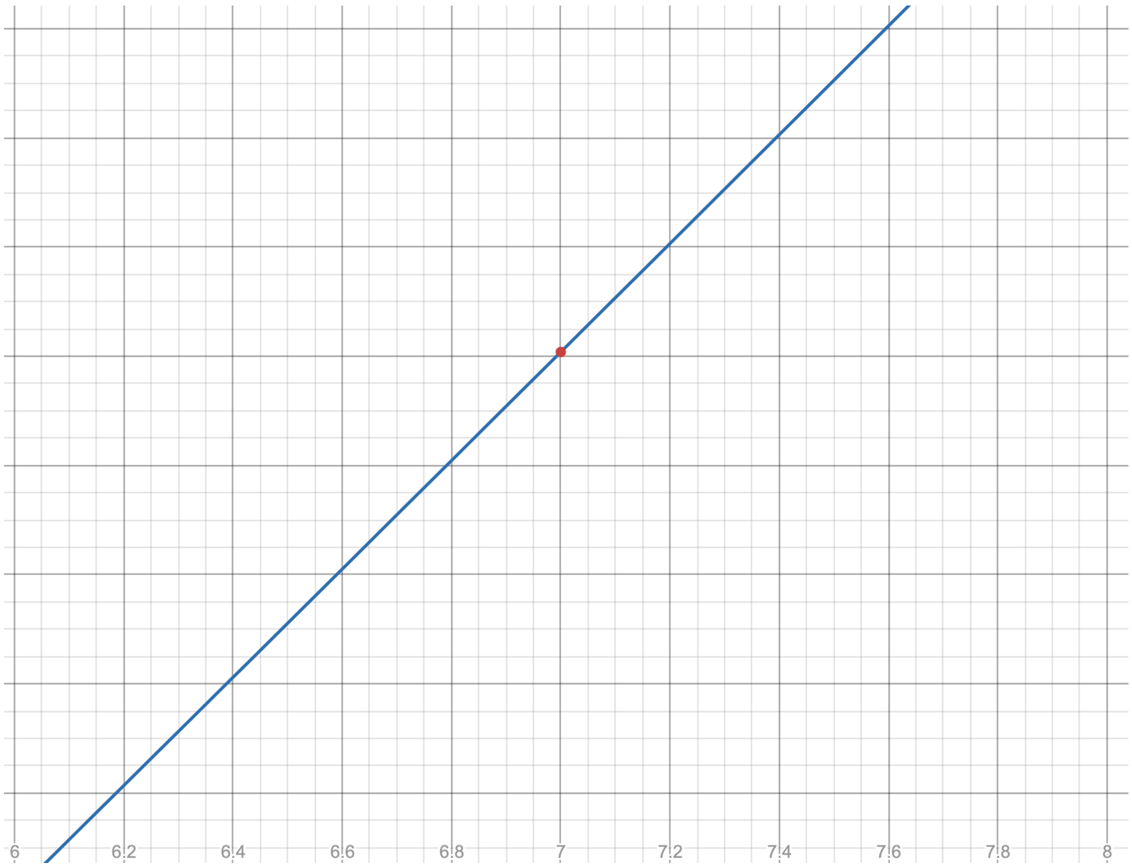


Image 3: $\Delta x = 0.02$ At this interval, it becomes clear that $f(7) \neq 7$, so the slope may be different than 1. I zoom in again.

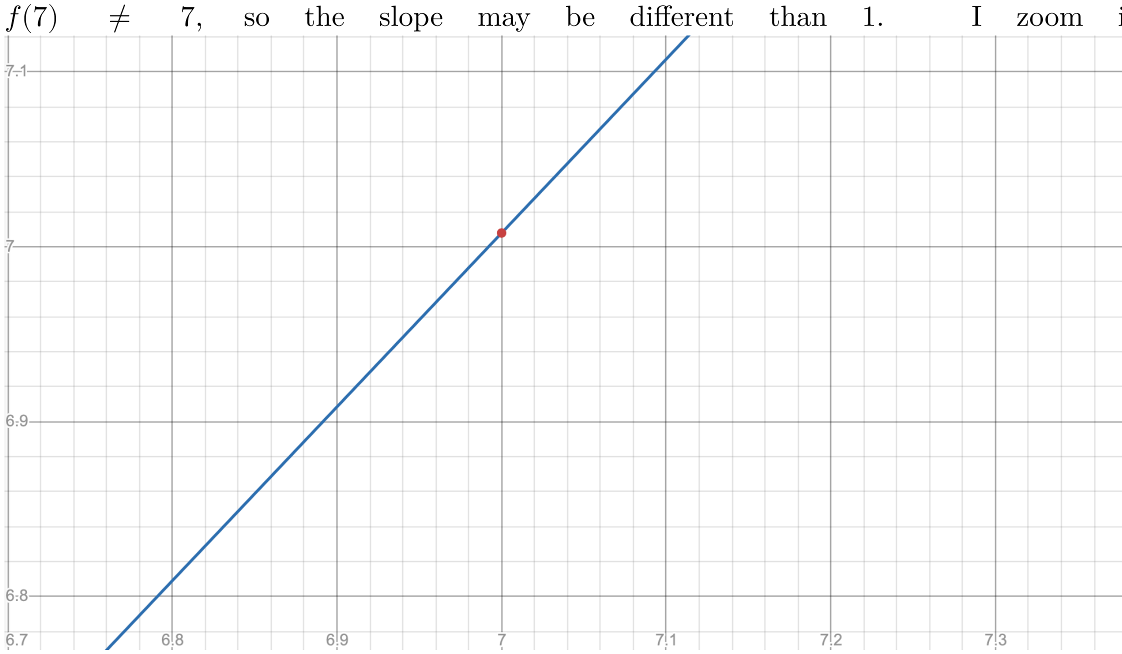
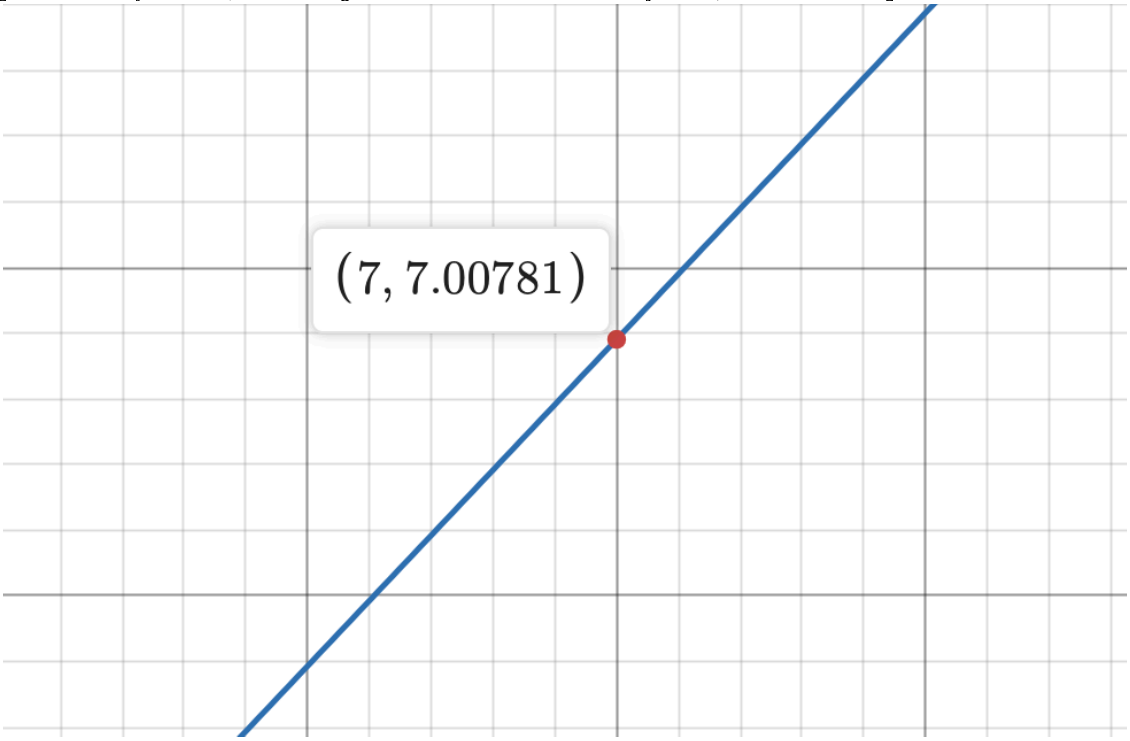
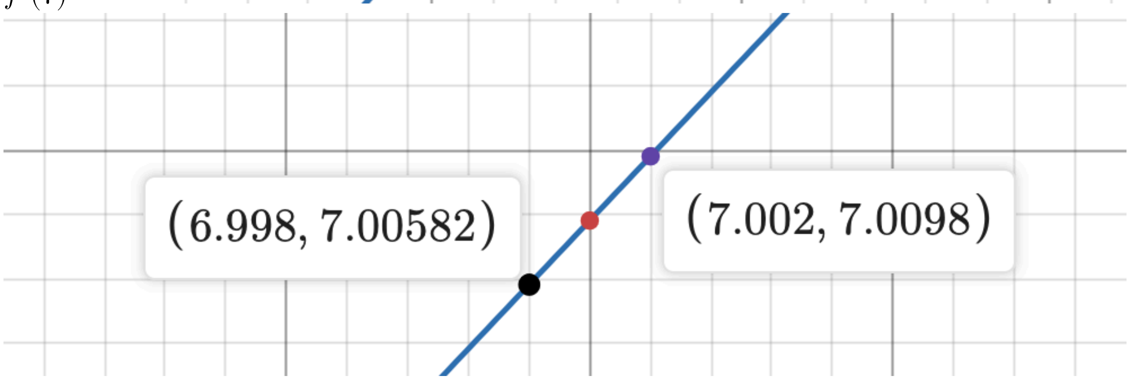


Image 4: $\Delta x = 0.004$ Getting closer, we can see that the slope still appears to be constant, but it is clearer that $f(7) \neq 7$ rather, it is closer to 7.0079. Because we have the y value to greater than two decimal places, I use this interval to calculate the slope. by placing two points on either side of

$t = 7$ at $t = 6.998$ and $t = 7.002$, and plotting their y values, we can find the difference in y values is also approximately 0.004, meaning that if $\Delta x = 4$ and $\Delta y = 4$, then the slope is in fact 1 over



$f'(7)$



1.2 10. (p 118/119)

True or false. If you think a statement is true, give your reason; if you think a statement is false, give a counterexample—i.e., an example that shows why it must be false. // - a) If g (t) is positive for all t, we can conclude that g(214) is positive. *False, if the slope is positive for all t but the y intercept is very negative, then the value of g(214) may be negative or 0*

- b) If g (t) is positive for all t, we can conclude that g(214) > g(17). *This is true, if the slope is positive then g(214) must be greater than g(17) because g(x) must increase as x increases*
- c) Bill and Samantha are driving separate cars in the same direction along the same road. At the start Samantha is 1 mile in front of Bill. If their speeds are the same at every moment thereafter, at the end of 20 minutes Samantha will be 1 mile in front of Bill. *True, if Bill and Samantha are driving at exactly the same speed in the same direction*

for the same amount of time, then their relative positions will remain constant and Samantha will still be 1 mile ahead 20 minutes later

- d) Bill and Samantha are driving separate cars in the same direction along the same road. They start from the same point at 10 am and arrive at the same destination at 2 pm the same afternoon. At some time during the four hours their speeds must have been exactly the same. _True. Because Bill and Samantha travel the same distance in the same amount of time, their average speeds are the same. However, whatever speeds they had during that interval could have varied greatly. For example if we assume that the average speed was v, then we know that Samantha could have started out going four times as fast as Bill $v_{S0} = 2v$, $v_{B0} = \frac{1}{2}v$, and then they could have suddenly switched halfway through, and in the end their average times were the same. But in order to accelerate to their new speeds they would have had to, at some point during the mutual acceleration, be at the same speed in order to swap.

1.3 2 (p 130/131)

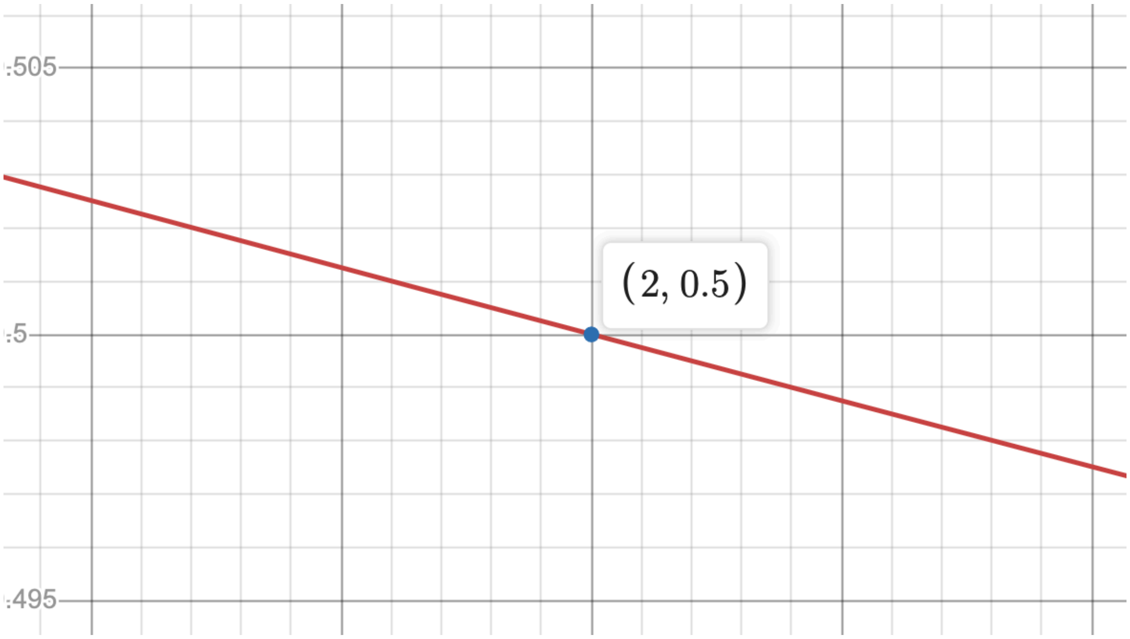
For each of the functions f below, approximate its derivative at the given value x = a in two different ways. First, use a computer microscope (i.e., a graphing program) to view the graph of f near x = a. Zoom in until the graph looks straight and find its slope. Second, use a calculator to find the value of the quotient.

$$\frac{f(a+h) - f(a-h)}{2h}$$

[2]:
```
def dev_quot(f, a, h): #find quotient
    xp = f(a+h)
    xm = f(a-h)
    return (xp-xm)/(2*h)

def mult_h(f, a):
    slope_est = list()
    for i in range(0, 6):
        h = 10**(-i)
        d = dev_quot(f, a, h)
        slope_est.append(d)

    return slope_est
```

1.3.1 (a)

$$f(x) = \frac{1}{x}$$
$$x = 2$$



The slope in this image appears to be $-\frac{1}{4}$, or -0.25. The quotient is found below, which matches this approximation as approaches 0.

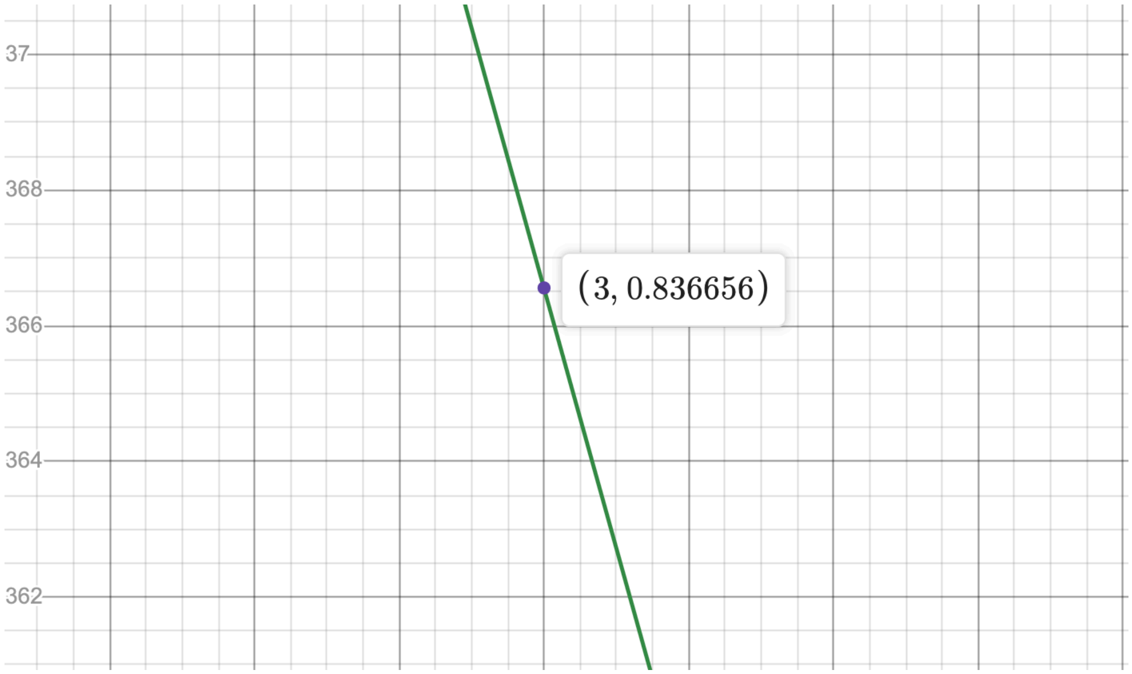
[5]:
```
def f2a(x):
    return 1/x
print(mult_h(f2a, 2))
```

```
[-0.33333333333333337, -0.2506265664160401, -0.25000625015624833,
-0.25000006249997764, -0.25000000062558314, -0.2500000000099645]
```

1.3.2 2b

$$f(x) = \sin(7x)$$

$$x = 3$$



By zooming in until the slope appears constant on the plot of $f(x)$ where (x = 7) is plotted, we see that the slope in the image appears to be $\frac{-4}{1}$ This is confirmed by the calculation below as the slope appears to approach -4 as h approaches 0.

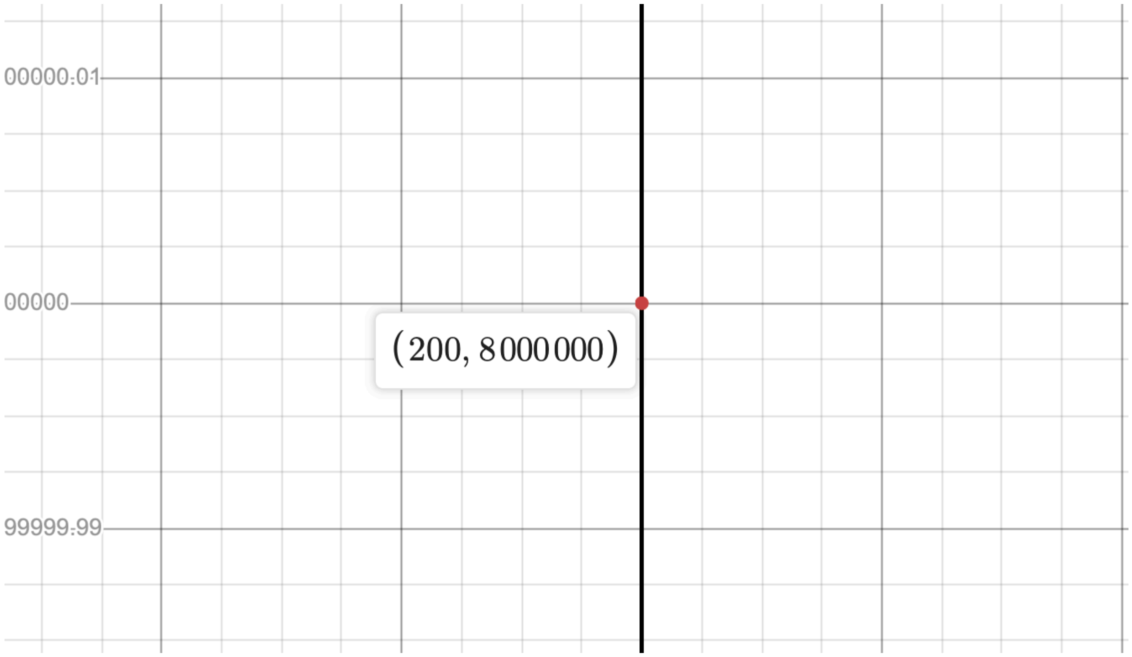
[6]:
```
import numpy as np
def f2b(x):
    return np.sin(7*x)

print(mult_h(f2b, 3))
```

```
[-0.3598507836935006, -3.528568772540893, -3.830974403016585,
-3.834073509789371, -3.8341045084616665, -3.8341048184842292]
```

1.3.3 2c

$$f(x) = x^3$$

$$x = 200$$



By zooming in on the graph of this function until the slope appears constant at $x = 200$, we can see that the slope appears to be very, very large or undefined at this point, as the change in x is very small compared to the change in y. Entering the point and function into the quotient gives an expected large number:

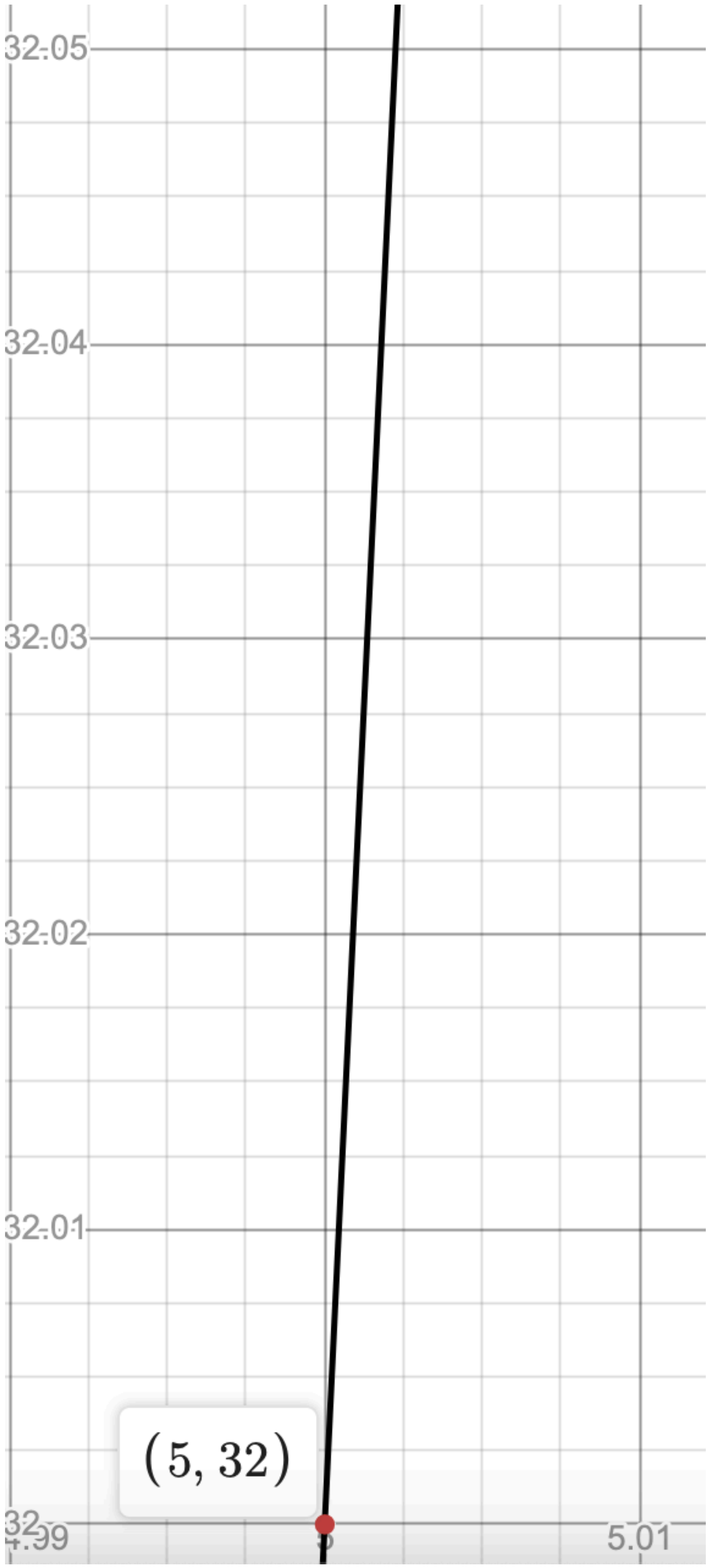
[7]:
```
def f2c(x):
    return x**3
print(mult_h(f2c, 200))
```

```
[120001.0, 120000.00999999233, 120000.00009988435, 120000.00000186265,
120000.00000465661, 120000.00006519257]
```

1.3.4 2d

$$f(x) = 2^x$$

$$x = 5$$


32.05
32.04
32.03
32.02
32.01
32
4.99
5
5.01
(5, 32)

The slope of this line around $x = 5$ is also steep, but I estimated it at about $f'(x) = 21$, and used the code below to confirm that I was a little bit off, as the slope seems to get closer to 22.18 as h approaches 0:

[8]:
```
def f2d(x):
    return 2**x

print(mult_h(f2d, 5))
```

```
[24.0, 22.198475359917644, 22.1808873914922, 22.180711554058874,
22.180709795645015, 22.180709777153137]
```

1.4 3 (p 130/ 131)

In a later section we will establish that the derivative of f (x) = x3 at x = 1 is exactly 3: f (1) = 3. This question concerns the freedom we have to choose points in a window to estimate f (1) (see page 121). Its purpose is to compare two quotients, to see which gets closer to the exact value of f (1) for a fixed "field of view" x. The two quotients are

$$Q_1 = \frac{f(a+h) - f(a-h)}{2h}$$
$$Q_2 = \frac{f(a+h) - f(a)}{h}$$

where $a = 1$

1.4.1 (a): Getting a table for Q1, Q2

Code is written below:

[34]:
```
from tabulate import tabulate
def Q1(f, a, h):
    xp = f(a+h)
    xm = f(a-h)
    return (xp-xm)/(2*h)

def Q2(f, a, h):
    xp = f(a+h)
    xa = f(a)
    return (xp-xa)/h

def f3a(x):
    return x**3

a = 1
tab = [['h', 'Q1', 'Q2']]
g_tab = []
for k in range(0, 8):
    h = 0.5**k
```

```
    q1 = Q1(f3a, a, h)
    q2 = Q2(f3a, a, h)
    tab.append([h, q1, q2])
    g_tab.append([h, q1, q2])

print(tabulate(tab))
```

```
---------  ----------------  ----------------
h          Q1                Q2
1.0        4.0               7.0
0.5        3.25              4.75
0.25       3.0625            3.8125
0.125      3.015625          3.390625
0.0625     3.00390625        3.19140625
0.03125    3.0009765625      3.0947265625
0.015625   3.000244140625    3.047119140625
0.0078125  3.00006103515625  3.02349853515625
---------  ----------------  ----------------
```

1.4.2 3(b)

How many digits of Q1 stabilize in this table? How many digits of Q2? *As seen in the table above, 5 digits stabilize in Q1, while only 2 stabilize in Q2.*

1.4.3 3(c)

Which is a better estimator—Q1 or Q2? To indicate how much better, give the value of h for which the better estimator provides an estimate that is as close as the best estimate provided by the poorer estimator.

The best estimator is Q1. This can be seen because the best estimate given by Q2 where $h = 0.0078125$ *is* ≈ 3.0235 *This value is surpassed by Q1 when* $h = 0.125$.

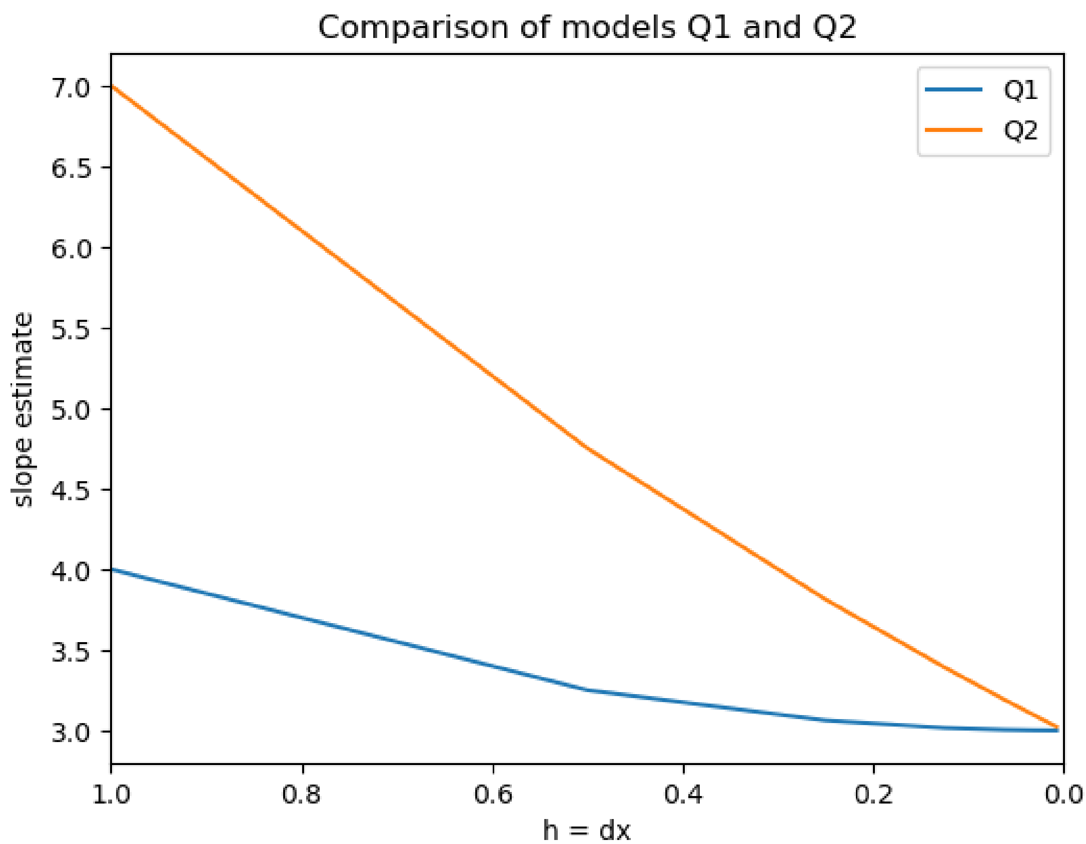
1.4.4 3(d)

Plots shown below:

Although both models converge to 0, Q2 gives a closer initial estimate with $h = 1$, and changes less as h approaches 0.

[43]:
```
h_lst = np.array(g_tab)[:, 0]
q1_lst = np.array(g_tab)[:, 1]
q2_lst = np.array(g_tab)[:, 2]

import matplotlib.pyplot as plt

fig, ax = plt.subplots()
```

```python
ax.plot(h_lst, q1_lst, label = "Q1")
ax.plot(h_lst, q2_lst, label = "Q2")
ax.set_ylabel("slope estimate")
ax.set_xlabel("h = dx")
ax.set_xlim(1, 0)
ax.set_title("Comparison of models Q1 and Q2")
ax.legend()
plt.show()
```



1.5 4 (p. 131)

Repeat all the steps of the last question for the function f (x) = $\sqrt{x}$ at $x = 9$. The exact value of f (9) is 1/6.

1.5.1 4(a)

[44]:
```
def f4a(x):
    return x**0.5

a4 = 9
tab4 = [['h', 'Q1', 'Q2']]
g_tab4 = []
for k in range(0, 8):
    h4 = 0.5**k
    q14 = Q1(f4a, a4, h4)
    q24 = Q2(f4a, a4, h4)
    tab4.append([h4, q14, q24])
    g_tab4.append([h4, q14, q24])
print('1/6 as a decimal is', 1/6)
print(tabulate(tab4))
```

```
1/6 as a decimal is 0.16666666666666666
---------  -------------------  -------------------
h          Q1                   Q2
1.0        0.16692526771109462  0.16227766016837952
0.5        0.16673105406183764  0.16441400296897601
0.25       0.1666827471986032   0.16552506059643868
0.125      0.16667068578158784  0.16609194718914466
0.0625     0.16666767138179495  0.1663783151691831
0.03125    0.16666691784147503  0.16652224137045835
0.015625   0.1666667294601183   0.16659439142901533
0.0078125  0.16666668236501891  0.16663051337502566
---------  -------------------  -------------------
```

1.5.2 4(b):

Q1 is still a much better estimate of the actual value of the slope at $x = 9$. 8 digits stabilize using Q1 as compared with 5 for Q2.

1.5.3 4(c)

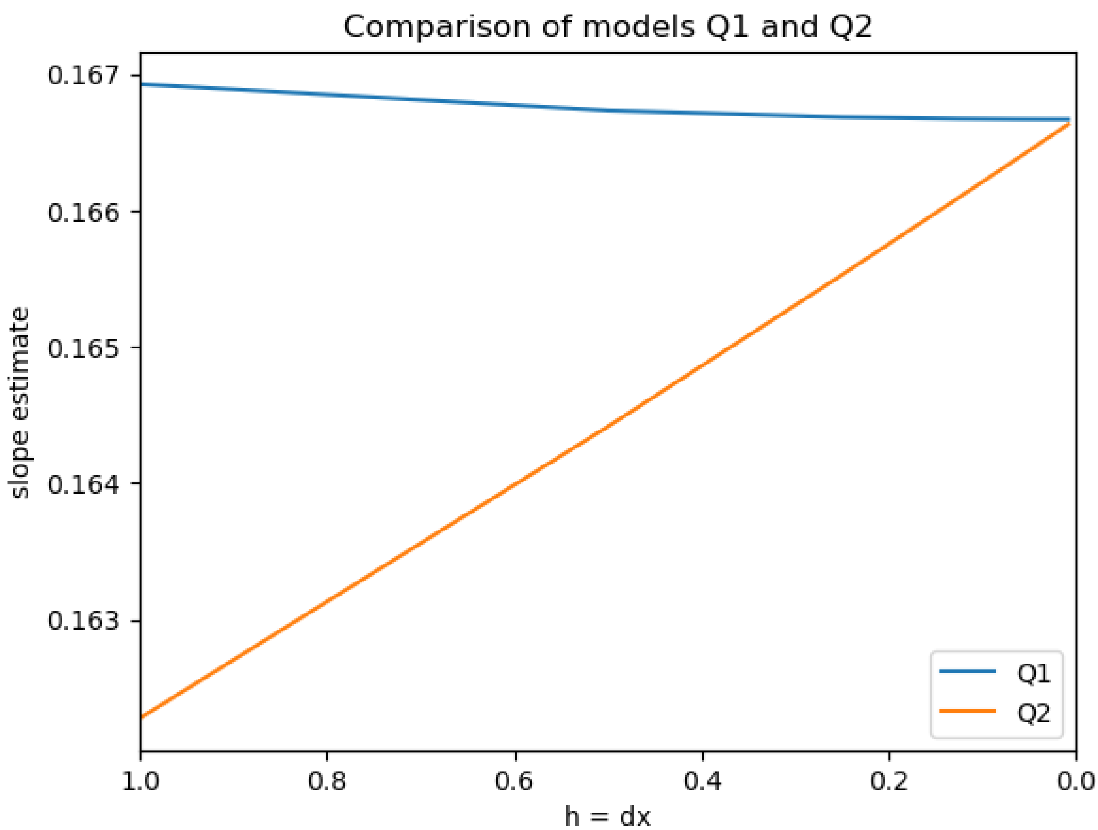
Q1 is a better estimator. 5 digits are stabilized in Q1 at $h = 0.25$, a much larger interval than the $h = 0.0078125$ where q2 stabilizes 5 digits.

1.5.4 4(d)

Plots shown below:

[45]:
```
h4_lst = np.array(g_tab4)[:, 0]
q14_lst = np.array(g_tab4)[:, 1]
q24_lst = np.array(g_tab4)[:, 2]

import matplotlib.pyplot as plt
```

```
fig, ax = plt.subplots()

ax.plot(h4_lst, q14_lst, label = "Q1")
ax.plot(h4_lst, q24_lst, label = "Q2")
ax.set_ylabel("slope estimate")
ax.set_xlabel("h = dx")
ax.set_xlim(1, 0)
ax.set_title("Comparison of models Q1 and Q2")
ax.legend()
plt.show()
```



1.6 6 (p. 132)

Use one of the methods of problem 2 to estimate the value of the derivative of each of the following functions at x = 0: y = 2x, y = 3x, y = 10x, and y = (1/2)x. These are called exponential functions, because the input variable x appears in the exponent. How many decimal places accuracy do your approximations to the derivatives have?

The answers below use $h = 0.025$ and are found to the 16th decimal place

[49]:
```python
# derivative of y=2^x at x = 0
def f6a(x):
    return 2**x
def f6b(x):
    return 3**x
def f6c(x):
    return 10**x
def f6d(x):
    return 0.5**x
h6 = 0.025
a6 = Q1(f6a, 0, h6)
b6 = Q1(f6b, 0, h6)
c6 = Q1(f6c, 0, h6)
d6 = Q1(f6d, 0, h6)

print('the derivative of y=2^x at x = 0 is ', a6)
print('the derivative of y=3^x at x = 0 is ', b6)
print('the derivative of y=10^x at x = 0 is ', c6)
print('the derivative of y=0.5^x at x = 0 is ', d6)
```

```
the derivative of y=2^x at x = 0 is  0.6931818711487048
the derivative of y=3^x at x = 0 is  1.098750415644445
the derivative of y=10^x at x = 0 is  2.30385697782731
the derivative of y=0.5^x at x = 0 is  -0.6931818711487048
```

1.7 15 (p. 134)

If f (a) = b, f (a) = −3 and if k is small, which of the following is the best estimate for f (a + k)? $a + 3k$, $b + 3k$, $a + 3b$, $b - 3k$, $a - 3k$, $3a - b$, $a2 - 3b$, $f'(a + k)$

If f(a)=b and we know that the slope of f where $x = a$ is $f'(a) = -3$, then the best way to estimate the next interval of $f(a+k)$ is to first multiply the slope by the change in x to get the change in y: $dy = -3 * k$, and then add that to b, the value at $f(a)$. So the best estimate from the list above is

$$b - 3k$$

1.8 16 (p.134)

If f is differentiable at a, which of the following, for small values of h, are reasonable estimates of f (a)?

The important aspect of these estimates of the derivative is that they are averages over the interval specified in the numerator. In this case the correct estimates are:

$$f'(a) = \frac{f(a+h) - f(a-h)}{2h}$$

The preceding equation in the book with only h in the denominator is not a true average, the slope will end up being 2 times what the actual estimate should be. In this case, the range of the derivative is 2h, h on either side of a, so the average must be over $2h$

$$f'(a) = \frac{f(a+h) - f(h)}{h}$$

is absolutely not correct, because h is a constant, this will pretty much just give $f(a+h)-1$ rather than the slope. If the second term in the numerator was $f(a)$, then this would be a more reasonable estimate

$$f'(a) = \frac{f(a+2h) - f(a-h)}{3h}$$

This one is interesting because it is an average that will find the average slope over the interval that includes a, but the interval is centered on $a + 0.5h$ rather than a. I would argue that if h is very small this could be an OK estimate, but I wouldn't use this equation for the purpose of finding $f'(a)$.

```
[ ]:
```