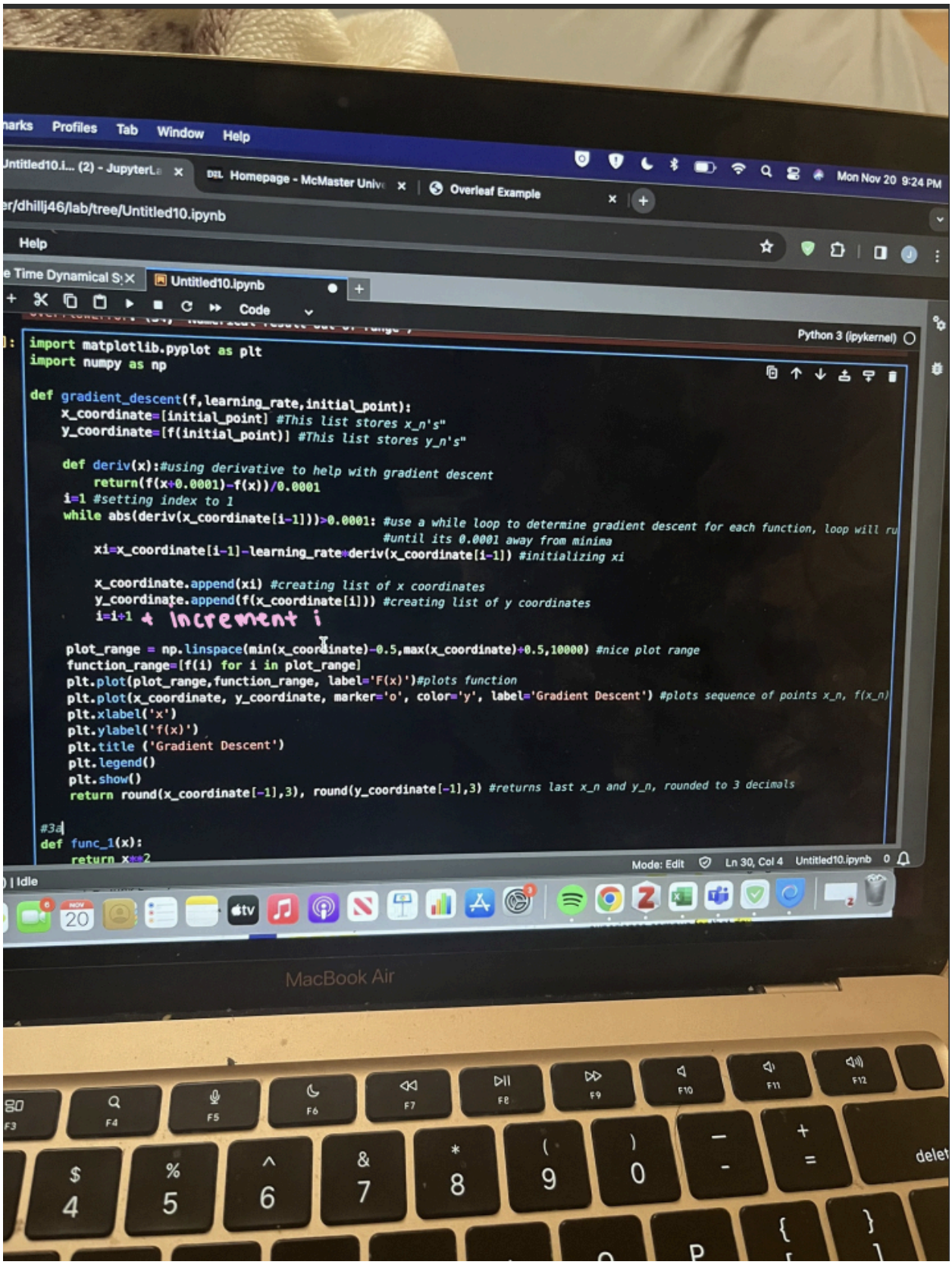
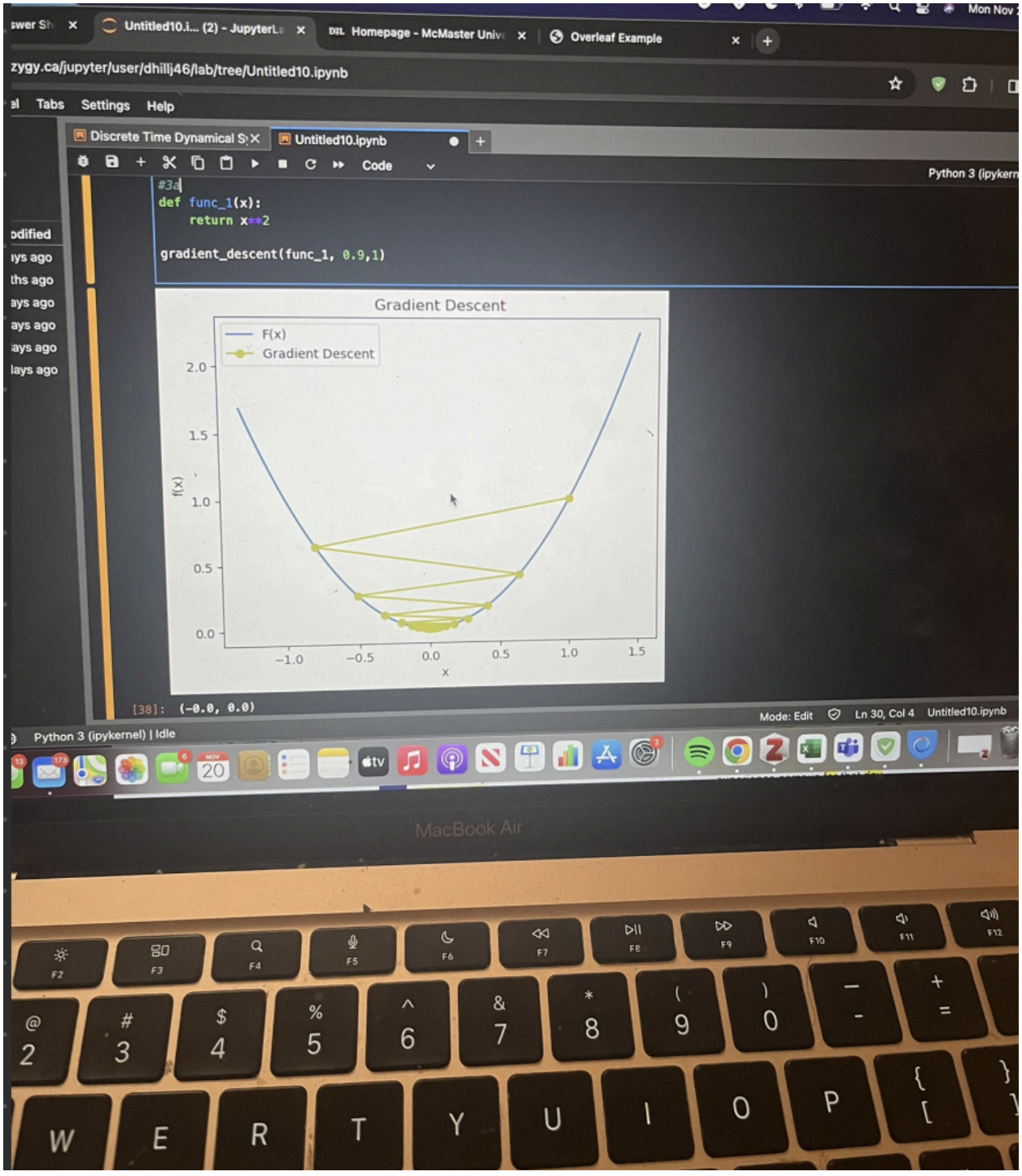
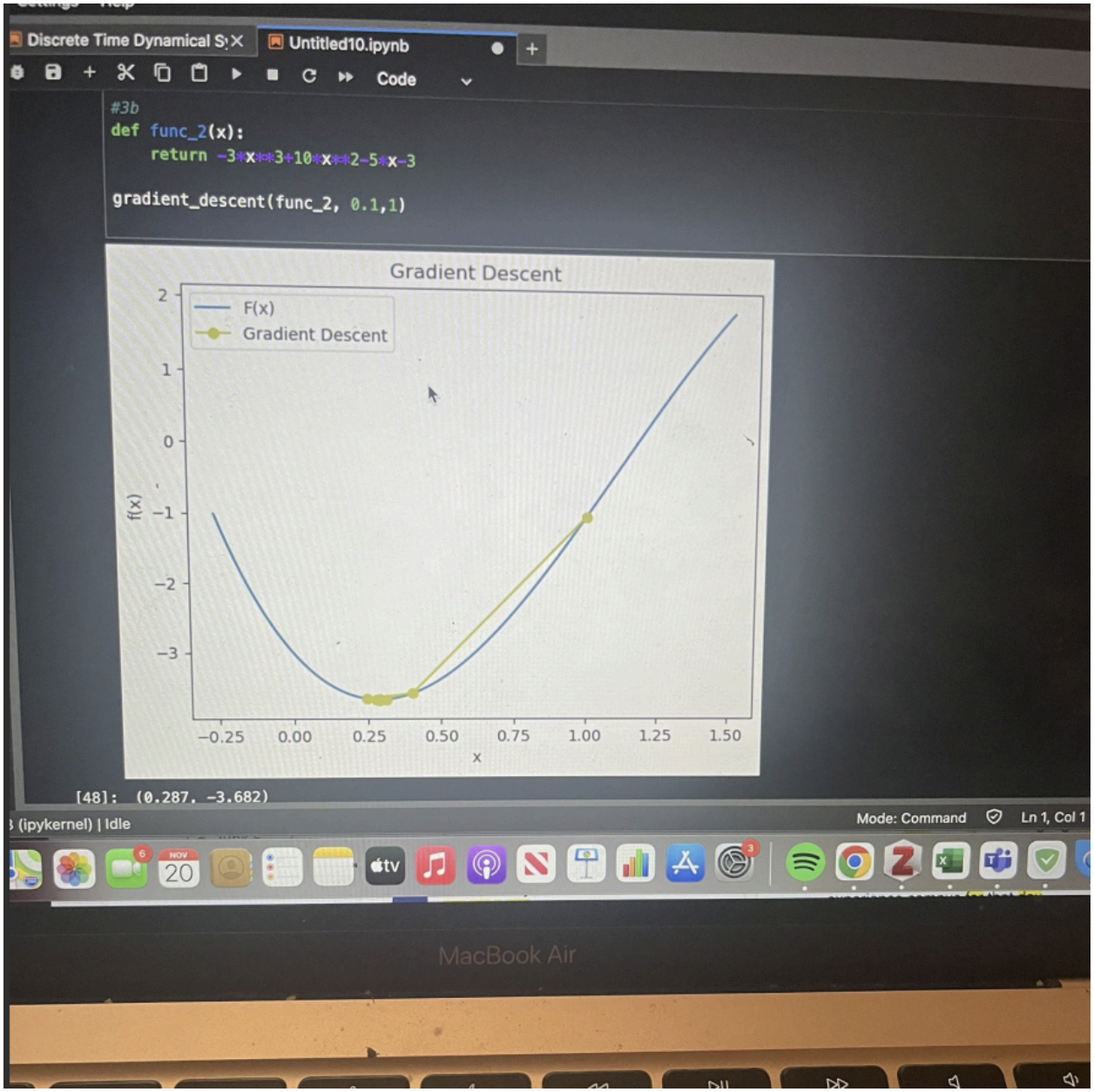
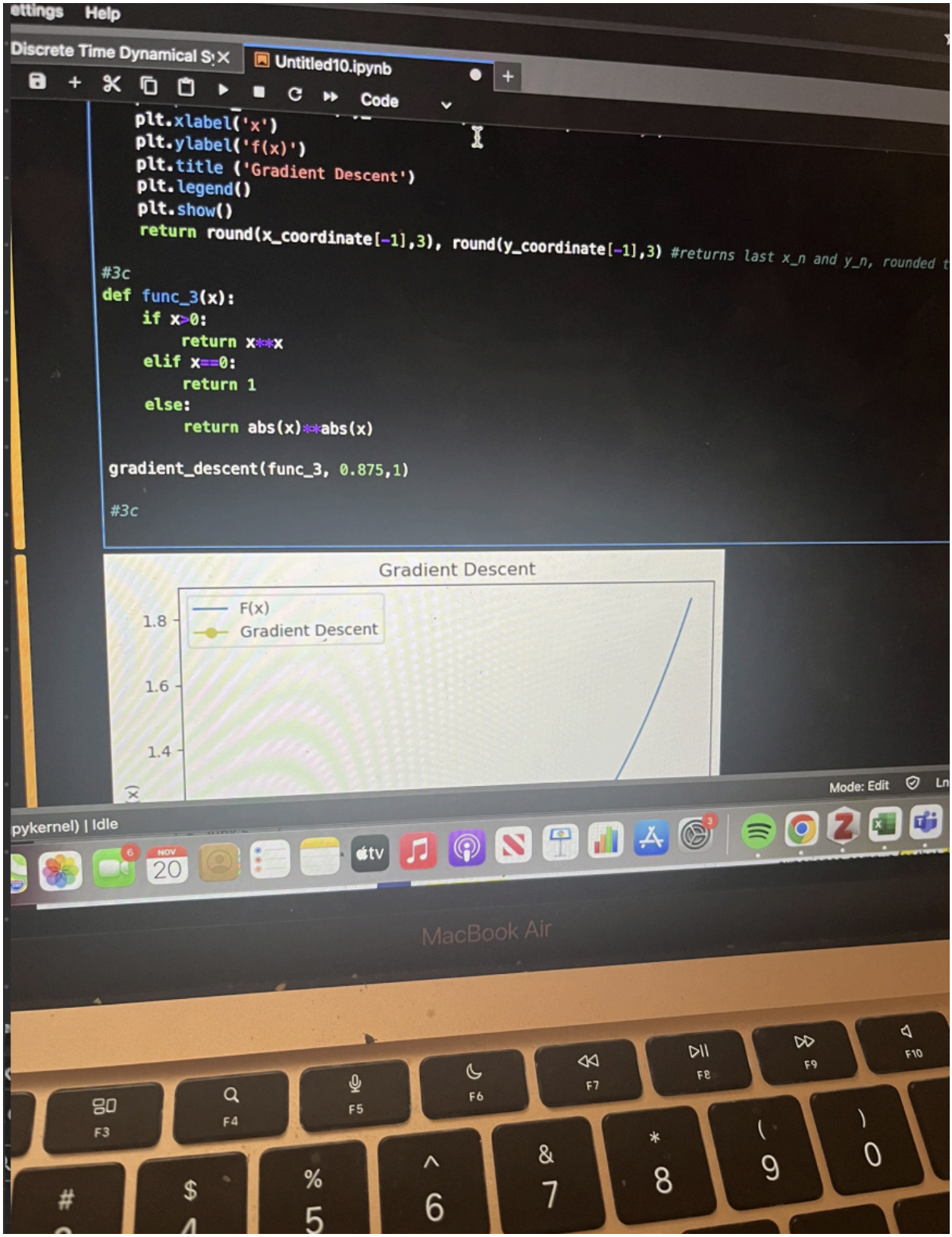
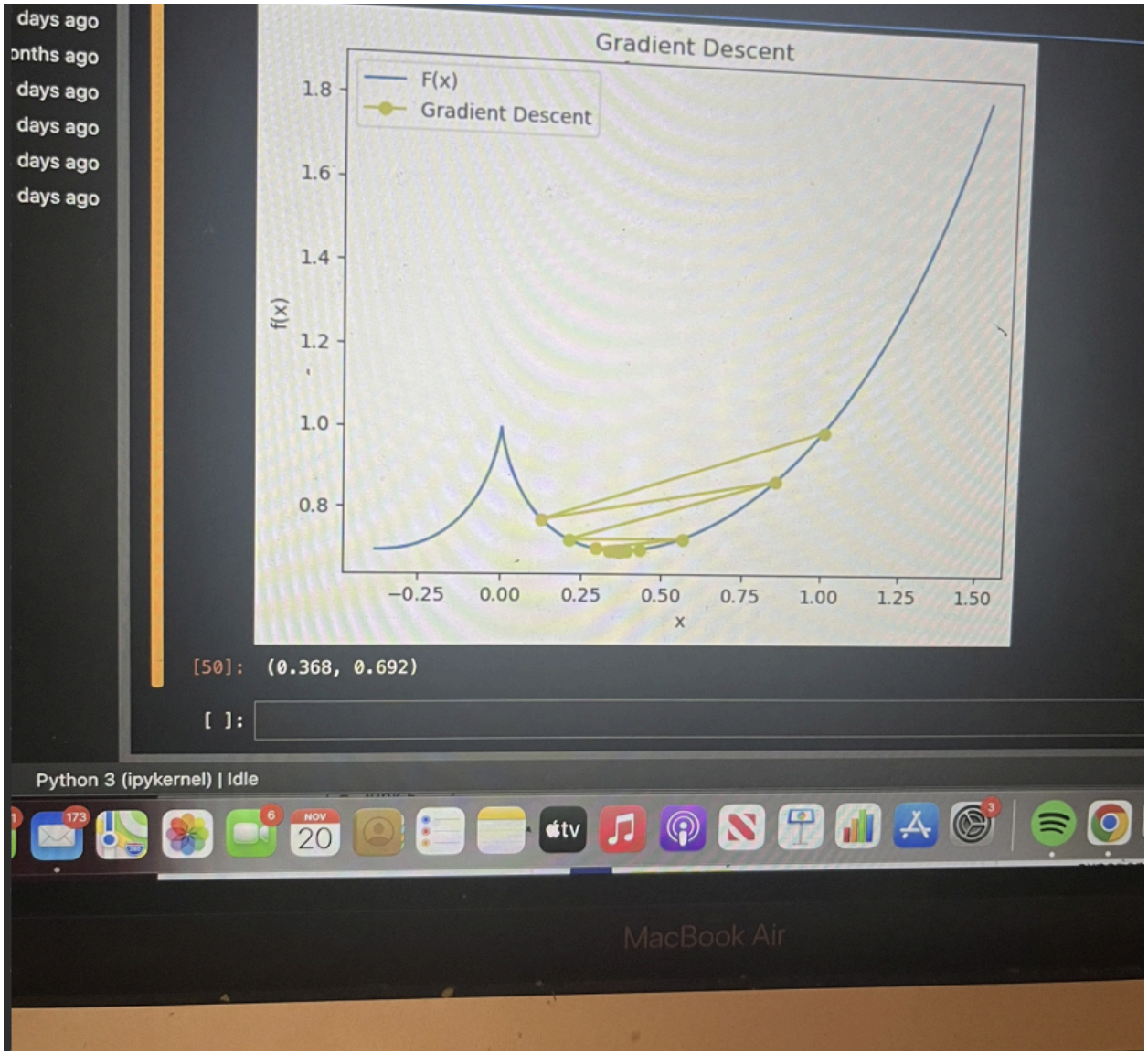
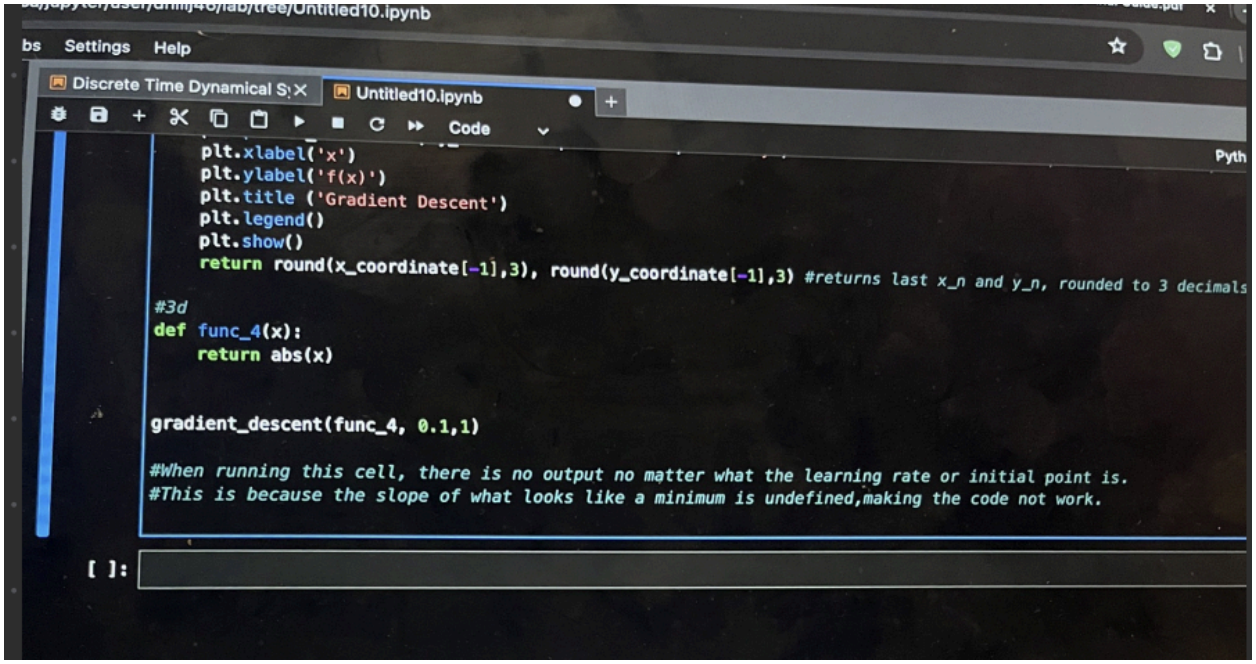

Profiles Tab Window Help
Mon Nov 20 9:24 PM
Homepage - McMaster Univ
Overleaf Example
/dhillj46/lab/tree/Untitled10.ipynb
Help
Time Dynamical S
Untitled10.ipynb
Code
Python 3 (ipykernel)
import matplotlib.pyplot as plt
import numpy as np
def gradient_descent(f,learning_rate,initial_point):
x_coordinate=[initial_point] #This list stores x_n's"
y_coordinate=[f(initial_point)] #This list stores y_n's"
def deriv(x):#using derivative to help with gradient descent
return(f(x+0.0001)-f(x))/0.0001
i=1 #setting index to 1
while abs(deriv(x_coordinate[i-1]))>0.0001: #use a while loop to determine gradient descent for each function, loop will ru
#until its 0.0001 away from minima
xi=x_coordinate[i-1]-learning_rate*deriv(x_coordinate[i-1]) #initializing xi
x_coordinate.append(xi) #creating list of x coordinates
y_coordinate.append(f(x_coordinate[i])) #creating list of y coordinates
i=i+1 # increment i
plot_range = np.linspace(min(x_coordinate)-0.5,max(x_coordinate)+0.5,10000) #nice plot range
function_range=[f(i) for i in plot_range]
plt.plot(plot_range,function_range, label='F(x)')#plots function
plt.plot(x_coordinate, y_coordinate, marker='o', color='y', label='Gradient Descent') #plots sequence of points x_n, f(x_n)
plt.xlabel('x')
plt.ylabel('f(x)')
plt.title ('Gradient Descent')
plt.legend()
plt.show()
return round(x_coordinate[-1],3), round(y_coordinate[-1],3) #returns last x_n and y_n, rounded to 3 decimals
#3a
def func_1(x):
return x**2
Mode: Edit
Ln 30, Col 4
Untitled10.ipynb
Idle
MacBook Air


Untitled10.i... (2) - JupyterLa
Homepage - McMaster Univ
Overleaf Example
zygy.ca/jupyter/user/dhillj46/lab/tree/Untitled10.ipynb
Tabs
Settings
Help
Discrete Time Dynamical S
Untitled10.ipynb
Code
Python 3 (ipykern
#3a
def func_1(x):
return x**2
gradient_descent(func_1, 0.9,1)
Gradient Descent
F(x)
Gradient Descent
f(x)
x
[38]: (-0.0, 0.0)
Mode: Edit
Ln 30, Col 4
Untitled10.ipynb
Python 3 (ipykernel) | Idle
MacBook Air


Discrete Time Dynamical S
Untitled10.ipynb
Code
#3b
def func_2(x):
return -3*x**3+10*x**2-5*x-3
gradient_descent(func_2, 0.1,1)
Gradient Descent
F(x)
Gradient Descent
f(x)
x
[48]: (0.287, -3.682)
(ipykernel) | Idle
Mode: Command
Ln 1, Col 1
MacBook Air


ettings Help
Discrete Time Dynamical S
Untitled10.ipynb
Code
plt.xlabel('x')
plt.ylabel('f(x)')
plt.title ('Gradient Descent')
plt.legend()
plt.show()
return round(x_coordinate[-1],3), round(y_coordinate[-1],3) #returns last x_n and y_n, rounded t
#3c
def func_3(x):
if x>0:
return x**x
elif x==0:
return 1
else:
return abs(x)**abs(x)
gradient_descent(func_3, 0.875,1)
#3c
Gradient Descent
F(x)
Gradient Descent
1.8
1.6
1.4
Mode: Edit
pykernel) | Idle
MacBook Air
F3
F4
F5
F6
F7
F8
F9
F10
#
$
%
^
&
*
(
)
4
5
6
7
8
9
0


days ago
onths ago
days ago
days ago
days ago
days ago
Gradient Descent
F(x)
Gradient Descent
f(x)
1.8
1.6
1.4
1.2
1.0
0.8
−0.25
0.00
0.25
0.50
0.75
1.00
1.25
1.50
x
[50]: (0.368, 0.692)
[ ]:
Python 3 (ipykernel) | Idle
MacBook Air


Untitled10.ipynb
bs Settings Help
Discrete Time Dynamical S
Untitled10.ipynb
Code
Pyth
plt.xlabel('x')
plt.ylabel('f(x)')
plt.title ('Gradient Descent')
plt.legend()
plt.show()
return round(x_coordinate[-1],3), round(y_coordinate[-1],3) #returns last x_n and y_n, rounded to 3 decimals
#3d
def func_4(x):
return abs(x)
gradient_descent(func_4, 0.1,1)
#When running this cell, there is no output no matter what the learning rate or initial point is.
#This is because the slope of what looks like a minimum is undefined,making the code not work.
[ ]: