CSEN1131P - SOFTWARE ENGINEERING LAB EXPERIEMENT-1

Implement weather modeling using the quadratic solution in stages: hardcoding variables keyboard input, read from a file, for a single set of input, multiple sets of inputs. save all versions, debug, fix problems, create a Github account.

Your Github Link:

Programs link from your account:

Name of the Student: G Nithin kumar

Complete Roll No:vu21csen0100244

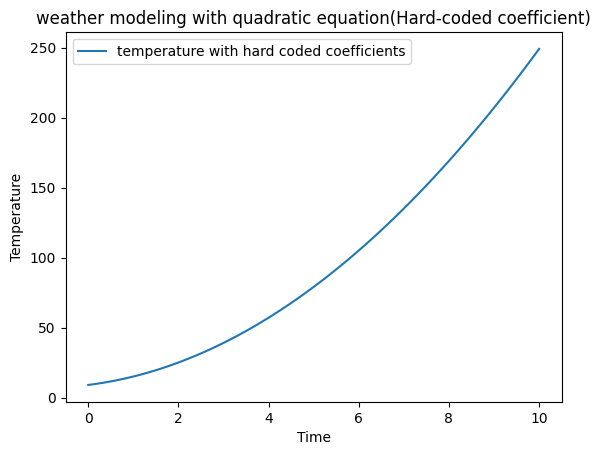
G Nithin kumar

Vu21csen0100244

Implement weather modeling\* using the quadratic solution in stages: hard-coding variables keyboard input, read from a file, for a single set of input, multiple sets of inputs.

a ) hard coding variables

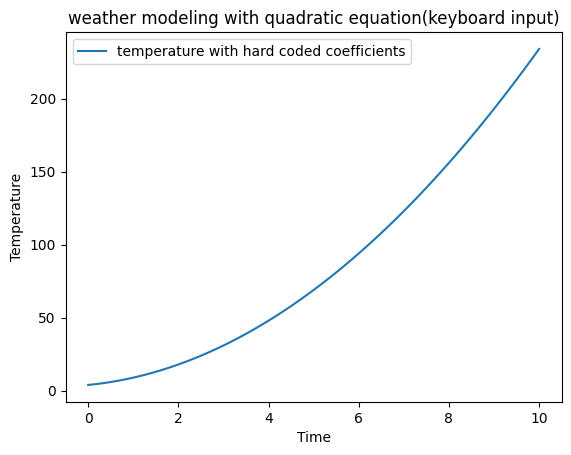
import matplotlib.pyplot as plt  
import numpy as np  
# hard code variables  
def quadratic\_model(time):  
 a=2  
 b=4  
 c=9  
 temperature = a\*time\*time + b\*time + c  
 return temperature  
  
time\_values=np.linspace(0,10,100)  
temperature\_hardcoded = quadratic\_model(time\_values)  
  
plt.plot(time\_values,temperature\_hardcoded, label='temperature with hard coded coefficients')  
plt.xlabel('Time')  
plt.ylabel('Temperature')  
plt.legend()  
plt.title('weather modeling with quadratic equation(Hard-coded coefficient)')  
plt.show()



b)keyboard input

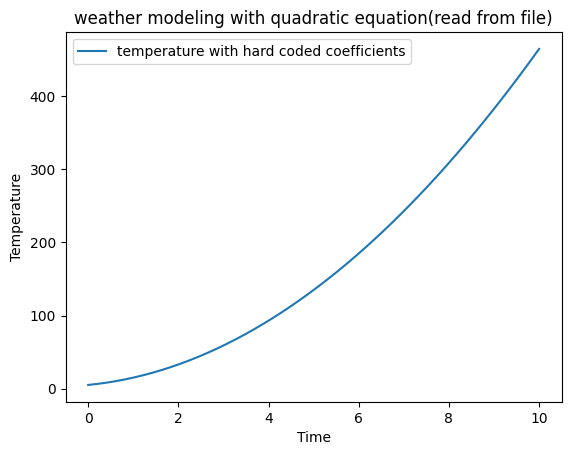
import matplotlib.pyplot as plt  
import numpy as np  
# hard code variables  
def quadratic\_model(a,b,c,time):  
 temperature = a\*time\*time + b\*time + c  
 return temperature  
  
time\_values=np.linspace(0,10,100)  
a=int(input("Enter value of a : "))  
b=int(input("Enter value of b : "))  
c=int(input("Enter value of c : "))  
temperature\_hardcoded = quadratic\_model(a,b,c,time\_values)  
  
plt.plot(time\_values,temperature\_hardcoded, label='temperature with hard coded coefficients')  
plt.xlabel('Time')  
plt.ylabel('Temperature')  
plt.legend()  
plt.title('weather modeling with quadratic equation(keyboard input)')  
plt.show()

Enter value of a : 2  
Enter value of b : 3  
Enter value of c : 4



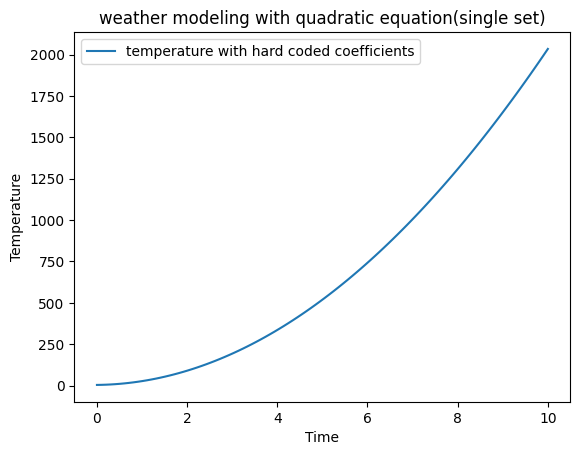
c) read from a file

f = open("VALUES.txt", "r")  
  
import matplotlib.pyplot as plt  
import numpy as np  
# hard code variables  
def quadratic\_model(f,time):  
 a=int(f.readline())  
 b=int(f.readline())  
 c=int(f.readline())  
 temperature = a\*time\*time + b\*time + c  
 return temperature  
  
time\_values=np.linspace(0,10,50)  
temperature\_hardcoded = quadratic\_model(f,time\_values)  
  
plt.plot(time\_values,temperature\_hardcoded, label='temperature with hard coded coefficients')  
plt.xlabel('Time')  
plt.ylabel('Temperature')  
plt.legend()  
plt.title('weather modeling with quadratic equation(read from file)')  
plt.show()



d)for a single set of input

import matplotlib.pyplot as plt  
import numpy as np  
# hard code variables  
def quadratic\_model(time):  
 a,b,c=[20,3,4]  
 temperature = a\*time\*time + b\*time + c  
 return temperature  
  
time\_values=np.linspace(0,10,100)  
temperature\_hardcoded = quadratic\_model(time\_values)  
  
plt.plot(time\_values,temperature\_hardcoded, label='temperature with hard coded coefficients')  
plt.xlabel('Time')  
plt.ylabel('Temperature')  
plt.legend()  
plt.title('weather modeling with quadratic equation(single set)')  
plt.show()



e) multiple sets of inputs.

import matplotlib.pyplot as plt  
import numpy as np  
# hard code variables  
def quadratic\_model(time,a,b,c):  
 temperature = a\*time\*time + b\*time + c  
 return temperature  
list = [(20,3,4),(4,3,1),(2,5,1)]  
time\_values=np.arange(0,51,1)  
  
for i,(a,b,c) in enumerate(list):  
 temperature\_values = quadratic\_model(time\_values,a,b,c)  
 label=f'Set{i+1}: a={a},b={b},c={c}'  
 plt.plot(time\_values,temperature\_values, label=label)  
  
  
plt.xlabel('Time')  
plt.ylabel('Temperature')  
plt.legend()  
plt.grid(True)  
plt.title('weather modeling with quadratic equation(multiple set)')  
plt.show()

