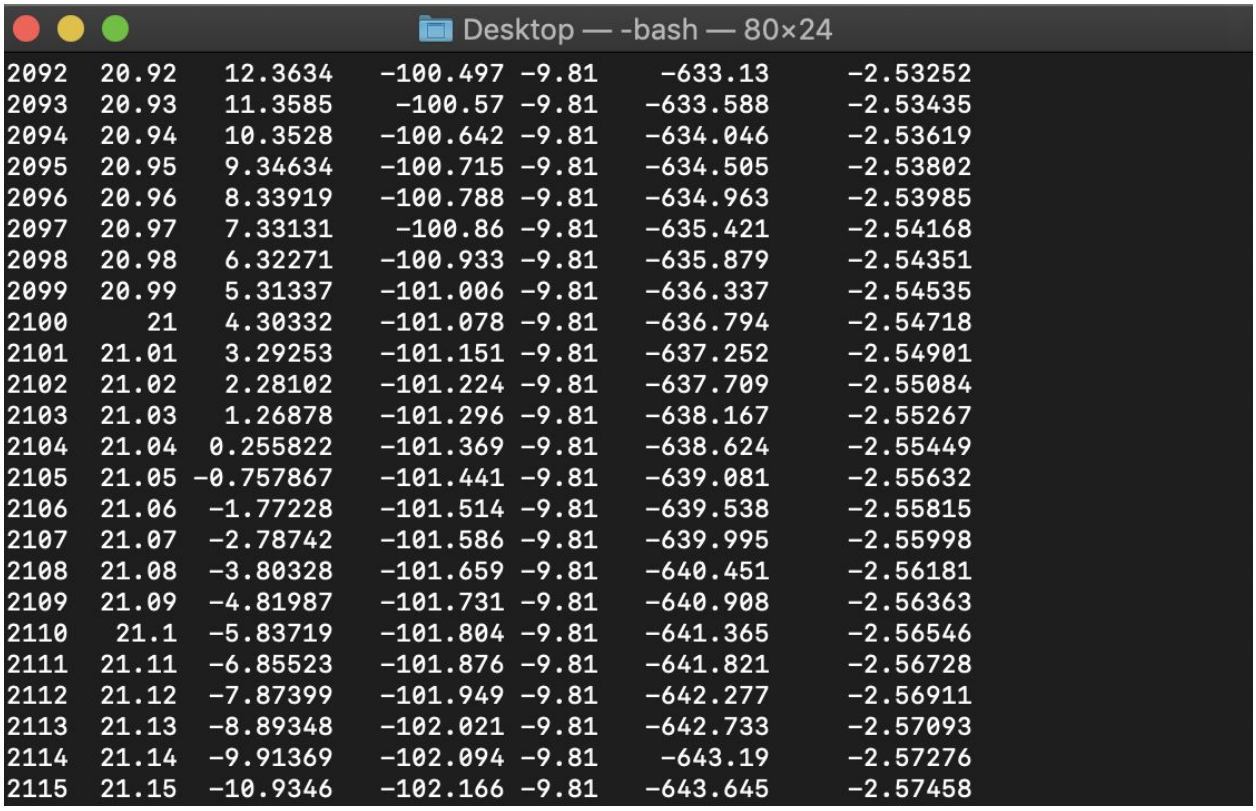
Math Project
Tristan Shah
Professor Folk

Desktop — -bash — 80×24

893	8.93	636.706	1.39172	-9.81	8.76783	0.0350713
894	8.94	636.719	1.29327	-9.81	8.14759	0.0325904
895	8.95	636.732	1.19484	-9.81	7.52751	0.03011
896	8.96	636.744	1.09644	-9.81	6.90758	0.0276303
897	8.97	636.755	0.998066	-9.81	6.28781	0.0251513
898	8.98	636.765	0.899714	-9.81	5.6682	0.0226728
899	8.99	636.774	0.801387	-9.81	5.04874	0.020195
900	9	636.782	0.703085	-9.81	4.42944	0.0177178
901	9.01	636.789	0.604808	-9.81	3.81029	0.0152412
902	9.02	636.795	0.506556	-9.81	3.1913	0.0127652
903	9.03	636.8	0.408328	-9.81	2.57247	0.0102899
904	9.04	636.805	0.310125	-9.81	1.95379	0.00781516
905	9.05	636.808	0.211947	-9.81	1.33527	0.00534107
906	9.06	636.81	0.113794	-9.81	0.7169	0.0028676
907	9.07	636.811	0.0156651	-9.81	0.0986898	0.000394759
908	9.08	636.811	-0.0824389	-9.81	-0.519365	-0.00207746
909	9.09	636.81	-0.180518	-9.81	-1.13726	-0.00454906
910	9.1	636.808	-0.278573	-9.81	-1.75501	-0.00702003
911	9.11	636.806	-0.376602	-9.81	-2.3726	-0.00949038
912	9.12	636.802	-0.474608	-9.81	-2.99003	-0.0119601
913	9.13	636.797	-0.572588	-9.81	-3.6073	-0.0144292
914	9.14	636.791	-0.670544	-9.81	-4.22442	-0.0168977
915	9.15	636.785	-0.768475	-9.81	-4.84139	-0.0193656
916	9.16	636.777	-0.866381	-9.81	-5.4582	-0.0218328

Columns: “Index”, “Timestamp”, “Position (m)”, “Velocity m/s”, “Gravity (m/s/s)”, “ARF kg*(m/s^2)”, “d(t)”

1.) According to the table the maximum height that the object reaches is 636.808 meters at 9.05 seconds. (I used an interval of 0.01 seconds but technically I could have gone as accurate as I wanted by changing the “interval” variable in the code.



2.) According to the table the object hits the ground at 21.04 seconds at a velocity of -101.369 m/s.

modeling.py

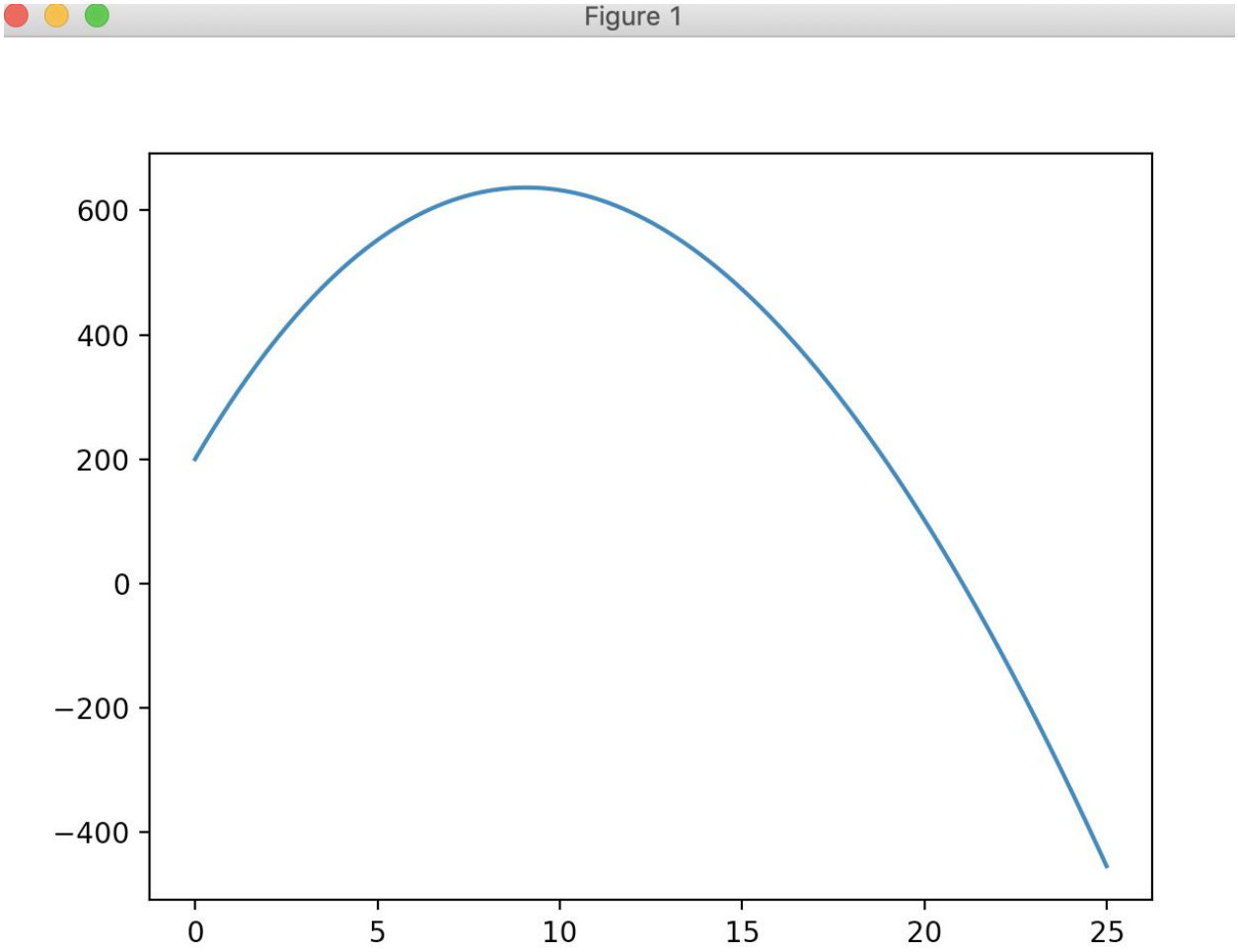
```python
import math
import pandas as pd
import matplotlib.pyplot as plt

pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
pd.set_option('display.width', None)

thing = pd.DataFrame(columns = ['t','st','vt','at','ARF','dt'])
thing.loc[0,'t'] = 0
thing.loc[0,'st'] = 200
thing.loc[0,'vt'] = 100
thing.loc[0,'at'] = -9.81
thing.loc[0,'ARF'] = 0
thing.loc[0,'dt'] = 0
interval = 0.01
for i in range(2500):
    thing.loc[i,'t'] = i * interval
    thing.loc[i,'at'] = -9.81
    thing.loc[i,'ARF'] = 6.3 * thing.loc[i,'vt']
    thing.loc[i,'dt'] = thing.loc[i,'ARF'] / 250
    thing.loc[i+1,'vt'] = -thing.loc[i,'dt']*interval + thing.loc[i,'at']*interval + thing.loc
    thing.loc[i+1,'st'] = thing.loc[i,'st'] + thing.loc[i,'vt'] * interval

thing = thing[['t','st','vt','at','ARF','dt']]
print(max(thing['st']))
print(thing)
plt.figure(1)
```

-Code for project



-Graph of Position vs Time over 25 seconds (sampling every 0.01 seconds)

3.)

a.) The procedure I used to calculate the data for each row is as follows:

- Calculate ARF for current row: multiply 6.3 (constant of air resistance) to the velocity of the current row.
- Calculate d(t) for the current row: divide ARF (current row) by the weight of the object (250 kg).
- Calculate the velocity of the NEXT row: d(t) * time interval + a(t) * time interval + previous velocity.
- Calculate position of the next row: current position + current velocity * time interval.

b.) The advantage of solving this problem numerically is that it is easier to do and requires less thinking than solving it algebraically.

c.) The disadvantage of solving this problem numerically is that no matter how small you set your time interval you will never get a perfect answer. However if you were able to solve this problem algebraically you would be able to arrive at an exact answer.

d.) Making a table with a smaller time interval (longer table) would increase accuracy and mostly solve the disadvantage of solving this numerically. However a longer table requires more computing power. There are no disadvantages that could not be solved by making a longer table.

e.) The way you would go about creating a table accounting for the decrease in air resistance and force of gravity is by changing each of their equations. EX:

The equation for force of gravity is:

$a(t) = -9.81$

It is a constant since we are assuming that it is not changing as height increases. In order to model the change you would have to use an equation that calculation that predicts the force of gravity with respect to position.

The equation would look more like this:

$a(p) = -9.81/p$

As position increases the force of gravity decreases. You would incorporate this into the table by adding one extra step: Creating a new column with the instantaneous acceleration due to gravity & instantaneous acceleration due to air resistance.

Note: This program can be used as a calculator for problems like this. As long as you change your starting velocity, position, weight, and time interval you can generate a graph of the position vs time of the object. Further, the more you decrease the time interval the more accurate your guess, therefore the maximum accuracy of your answer is proportional to the computing power you have at your disposal.

Full code for this project is on github: