School of Mechanical & Manufacturing Engineering (SMME),

National University of Science and Technology (NUST),

Sector H-12, Islamabad

Program:BE Aerospace Engineering Section:AE-01

Session:Fall 2024 Semester:3rd

Course Title:Fundamentals of Programming(II) Python

”End Semester Project Report”

Name : Aisha Iqbal

CMS : 456928

Objective:
To simulate the flight dynamics of a rocket using Python.

Abstract:
This project is based on the fundamental concepts of computation programming and aerospace engineering to simulate a rocket's flight dynamics using Python. In order to simulate the rocket's trajectory, velocity, and acceleration over time, the simulations are taken into account for important physical characteristics such as thrust, drag coefficient, air density, fuel burn rate, and gravitational acceleration. Cross-sectional area, rocket mass, and fuel mass are taken as user inputs that help in examining various scenarios.

The behavior of the rocket under different forces, including thrust, drag, and gravitational attraction, is continuously calculated by the simulation using numerical methods. Graphs showing altitude, velocity, and acceleration as functions of time are the project outputs that provide information about the rocket's performance. To visualize this data Matplotlib is used.

Through the combination of Python and aerospace knowledge the project provides an initial understanding of rocket dynamics and an adaptable framework for further developments such as multi-stage rockets changing atmospheric circumstances. This is a first step towards more in-depth research on propulsion and aerodynamics.

Introduction:
Simulation is an important component of aerospace engineering.In rocket flight it makes it possible to analyze rocket dynamics in a virtual setting. The goal of this project is to use Python to represent important forces like lift, drag, and gravity in order to infer a rocket's flight path,velocity and acceleration..

During flight forces like thrust, air resistance and gravitational pull can change dramatically. Fuel consumption lowers the rocket's mass as it rises which affects acceleration and velocity. In practical aerospace applications it is essential to comprehend these relationships in order to maximize rocket performance, guarantee safety and also make it cost effective.

Python is a great platform for this project because it's very easy to use and it also has large libraries of data visualization tools such as Matplotlib.This project gives the basics for further study and advancement in rocket dynamics in addition to showing the implementation of theoretical concepts.

Methodology:

The methodology for this project consists of simulating the flight of a rocket using Python programming, incorporating fundamental principles of physics along with basics of aerospace engineering. The simulation shows the effects of forces like thrust, drag

and gravity on the rocket's trajectory, velocity, and acceleration over time. Below is a step-by-step explanation of the approach:

1. Input Parameters:
   User-defined inputs are collected, including:
   - Initial mass of the rocket (including fuel).
   - Fuel mass and burn rate (kg/s).
   - Thrust generated by the engine (N).
   - Drag coefficient and cross-sectional area.
   - Time step for the simulation (e.g., 0.1 seconds).
2. Initialization:
   Variables such as time, altitude, velocity, and mass are initialized. Constants like gravitational acceleration (9.81 m/s^2) and air density (1.225 kg/m^3) are predefined to simplify calculations.
3. Force Calculations:
   At each time step, forces acting on the rocket are computed:
   - Weight:Calculated as Weight=Mass×gWeight=Mass×g.
   - Drag:Determined using the formula Drag=0.5×Air Density×Velocity2×Drag Coefficient×Cross-sectional AreaDrag=0.5×Air Density×Velocity2×Drag Coefficient×Cross-sectional Area.
   - Thrust:Provided by the rocket engine and it reduces as fuel is consumed.
4. Net Force and Acceleration:
   The net force is calculated by summing the forces acting on the rocket:
   Net Force=Thrust−Drag−WeightNet Force=Thrust−Drag−Weight
   Acceleration is then computed using Newton's Second Law:
   Acceleration=Net ForceMassAcceleration=MassNet Force
5. Time:
   Acceleration is calculated and from it velocity and altitude are solved for each time step using the following equations:Velocity=Velocity +(Acceleration×Time Step)Velocity=Velocity +(Acceleration×Time Step)Altitude=Altitude +(Velocity×Time Step)Altitude=Altitude +(Velocity×Time Step)
6. Termination Condition:
   The simulation runs until the rocket's altitude drops below zero, indicating it has hit the ground.
7. Data Storage and Visualization:
   At each instant time, altitude, velocity, and acceleration values are stored in lists. These values are then plotted using Matplotlib to visualize the rocket's performance during its flight.

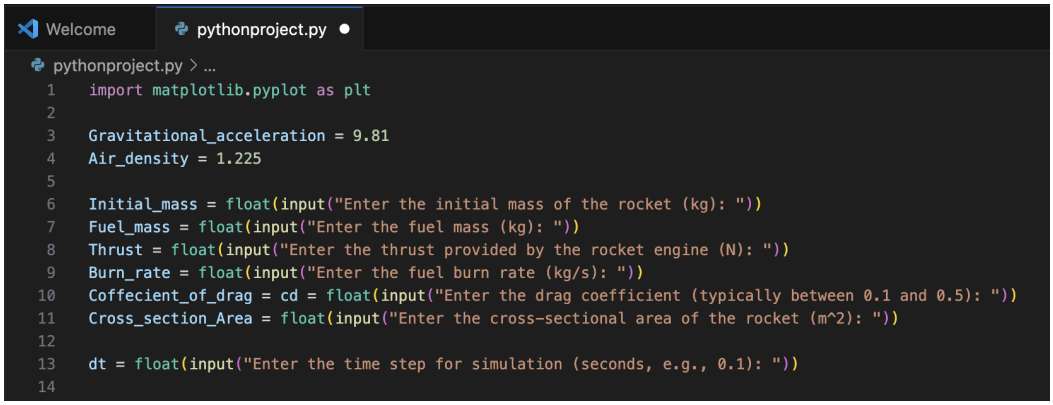
This methodology makes sure that a realistic simulation of rocket dynamics is graphed and it also gives ideas about the factors that influence its flight behavior.

Implementation:

Python programming was used to generate the rocket flight simulation, taking use of its numerical computations and data presentation capabilities. Iteratively simulating the rocket's motion, the program computes altitude, velocity, and acceleration using physics principles. A thorough description of the implementation procedure may be found below:

Code Overview:

- Input Parameters:First of all the program begins to collect data from users such as initial mass, fuel mass, thrust, burn rate, drag coefficient, and time step for simulation.
- Physics Calculations: Then forces acting on the rocket including weight, drag, and thrust, are calculated using equations.
- Simulation Loop: The program then uses a while loop to update the rocket's velocity and altitude until it returns to the ground.
- Data Storage: Time, altitude, velocity, and acceleration values are stored in lists during the simulation for visualization.
- Visualization: Matplotlib is used to plot the rocket's performance over time, displaying graphs for altitude, velocity, and acceleration.



```
    # Calculate net force
    if Fuel_mass > 0:
        net_force = Thrust - drag - weight
        Fuel_mass -= Burn_rate * dt
        mass -= Burn_rate * dt
    else:
        net_force = - drag - weight

    acceleration = net_force / mass
    velocity += acceleration * dt
    Altitude += velocity * dt

    # Store data for plotting
    time_data.append(Initial_time)
    altitude_data.append(Altitude)
    velocity_data.append(velocity)
    acceleration_data.append(acceleration)

    Initial_time += dt
```

```
    # Break if the rocket has hit the ground
    if Altitude < 0:
        break

# Plot results
plt.figure(figsize=(10, 6))

# Altitude plot
plt.subplot(3, 1, 1)
plt.plot(time_data, altitude_data)
plt.title("Rocket Flight Simulation")
plt.ylabel("Altitude (m)")
```

```

  # Velocity plot
  plt.subplot(3, 1, 2)
  plt.plot(time_data, velocity_data)
  plt.ylabel("Velocity (m/s)")

  # Acceleration plot
  plt.subplot(3, 1, 3)
  plt.plot(time_data, acceleration_data)
  plt.xlabel("Time (s)")
  plt.ylabel("Acceleration (m/s^2)")

  plt.tight_layout()
  plt.show()
```

5. Results and Discussion

The rocket's motion under defined parameters is successfully calculated and displayed by the rocket flight simulation. Graphs of altitude, velocity, and acceleration over time are among the outputs that offer information about the rocket's flight dynamics.

1. Altitude versus Time:

The rocket's rise during powered flight and its descent following fuel loss can be seen on the altitude graph. At first, the rocket rises quickly as thrust outpaces weight and drag. But as the rocket runs out of fuel, gravity and drag cause it to slow down and eventually fall back to the ground. The highest point attained before descent is represented by the peak altitude.

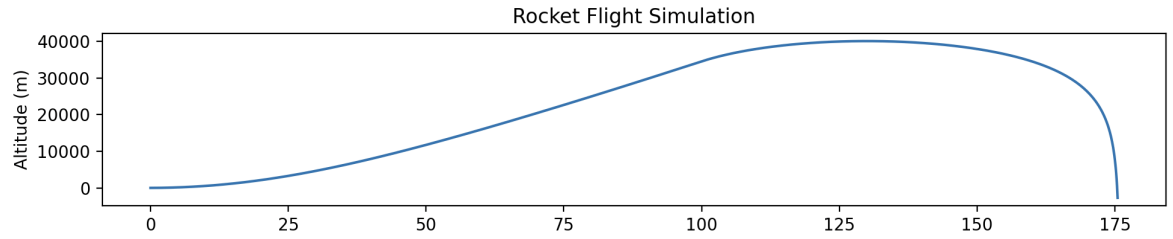


Figure 1: Time vs Altitude simulations

2. Velocity and Time:

The rocket's changing speed during the flight can be seen by the velocity graph. The velocity increases during rise until weight and drag overcome the upward motion, causing the net force to begin to decrease. The velocity decreases with fuel depletion and eventually turns negative during descent.

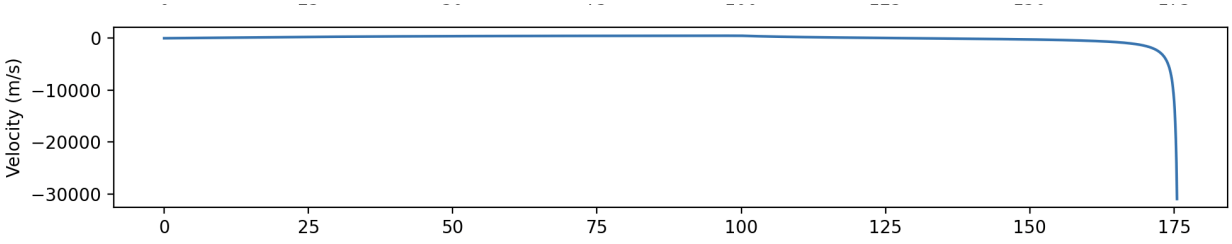


Figure 2: Time vs Velocity

3. Time versus Acceleration:

The forces exerting on the rocket can be seen in the acceleration graph. Because of the push there is a lot of acceleration during the first phase. Acceleration varies slightly as a result of the mass decreasing as fuel burns. Acceleration turns negative as fuel runs out with drag resistance and gravitational deceleration taking over.

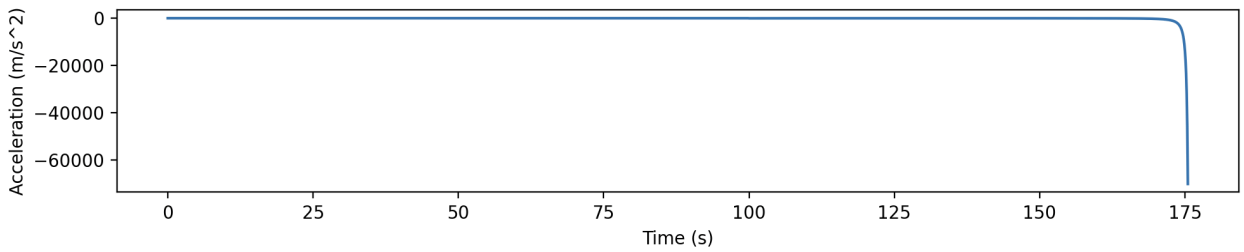
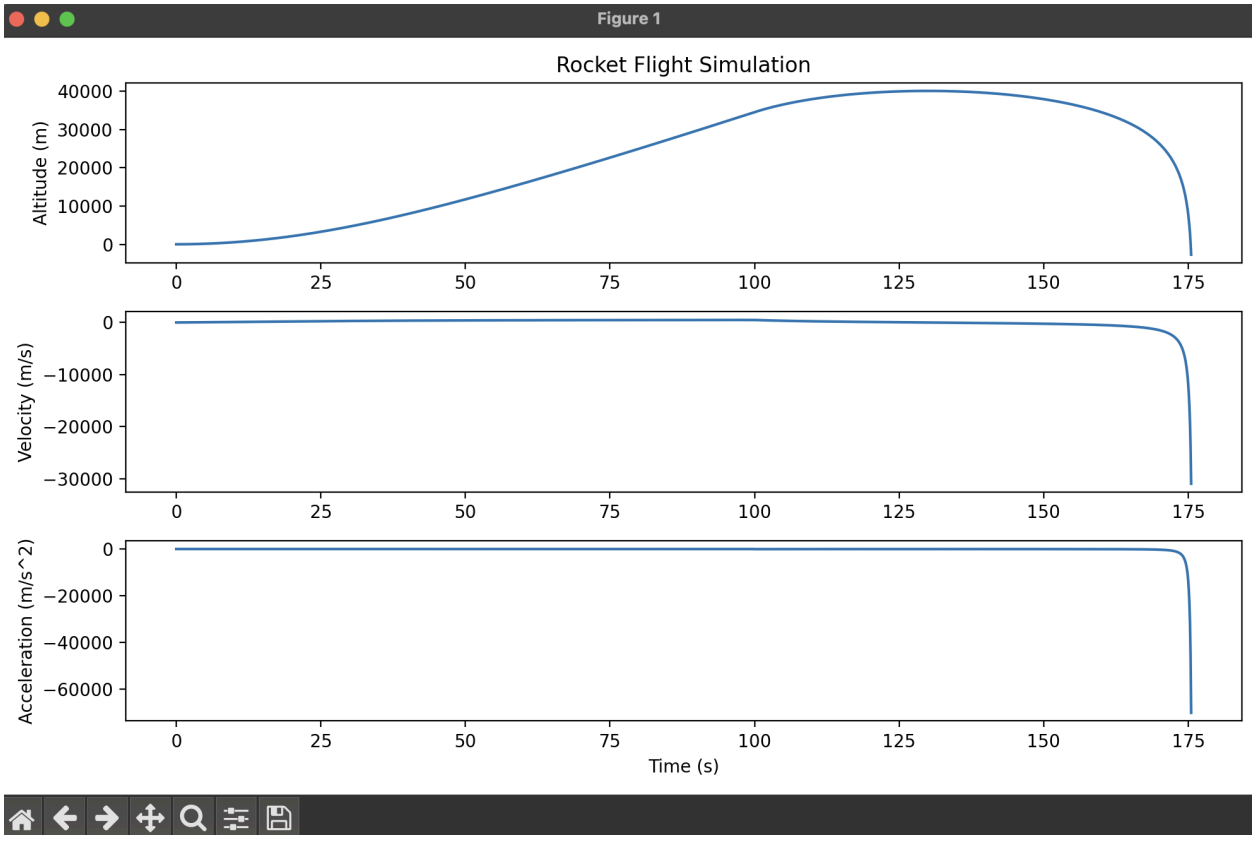


Figure 3: Acceleration vs Time

4. Output and Simulations:

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

(base) aishaiqbal@Aishas-MacBook-Air Python % /Applications/miniconda3/bin/python /Users/aishaiqbal/Python/pythonproject.py
Enter the initial mass of the rocket (kg): 1000
Enter the fuel mass (kg): 500
Enter the thrust provided by the rocket engine (N): 20000
Enter the fuel burn rate (kg/s): 5
Enter the drag coefficient (typically between 0.1 and 0.5): 0.2
Enter the cross-sectional area of the rocket (m^2): 0.5
Enter the time step for simulation (seconds, e.g., 0.1): 0.1
2024-11-19 15:43:31.474 python[5835:342930] +[IMKClient subclass]: chose IMKClient_Legacy
2024-11-19 15:43:31.474 python[5835:342930] +[IMKInputSession subclass]: chose IMKInputSession_Legacy
(base) aishaiqbal@Aishas-MacBook-Air Python %
```



Conclusion:

The mechanics of vertical rocket motion are modeled by the rocket flight simulation, which captures the effects of drag,thrust and gravity. The project demonstrates how complicated systems in aircraft engineering can be simulated using computational tools like Python. The outcomes which are displayed as graphs can show the rocket's path from the start to end and give every detail about its behavior.

One important conclusion is that as the acceleration and velocity change over time and fuel consumption and mass reduction play a crucial influence in determining performance. The simulation provides a solid basis for comprehending fundamental rocket dynamics and fits in with theoretical predictions. This study serves as a basis for upcoming developments in rocket design and highlights the significance of using computational techniques with aerospace concepts.

Applications:

1.Educational Tool: Helps students understand the physics of rocket dynamics and the interplay of forces during flight.
2.Preliminary Design Analysis: Provides a basic framework for evaluating rocket performance before prototyping.
3.Aerospace Engineering Training: Offers insights into the role of mass reduction, drag, and thrust in real-world applications.
4.Simulation-Based Learning: Encourages the use of programming for solving complex aerospace problems.
5.Advanced Research: Serves as a model for exploring multi-stage rockets, variable thrust or atmospheric effects.

7. References:

1.Anderson, J. D. (2010). *Fundamentals of Aerodynamics*. McGraw-Hill Education.

- Referenced for understanding the principles of drag, lift, and other aerodynamic forces.

2.NASA Glenn Research Center. *Rocket Principles*. Retrieved from https://www.nasa.gov/glenn/

- Used as a resource for understanding the physical principles of rocket motion.

3.Python Software Foundation. *Python Documentation*.Retrieved from https://docs.python.org/3/