# Satellite Orbit Visualization Project: A Technical Overview

## 1. Introduction

This project involves visualizing satellite orbits in 3D space, applying real textures to both Earth and Moon, and showcasing the orbit path of selected satellites. Using Two-Line Element (TLE) data, we parsed satellite information, predicted positions over time, and visualized the trajectory along with the Earth and Moon models.

## 2. Project Objectives

The primary objectives of this project were to:

- Parse TLE data for selected satellites.
- Predict satellite positions over a given time range.
- Visualize the satellite orbit with the Earth and Moon in 3D.
- Apply real textures to both the Earth and Moon for realistic visualization.
- Enable the user to select different satellites for visualization.

## 3. Data and Satellite Selection

### 3.1 TLE Data

We used up-to-date Two-Line Element (TLE) data provided by CelesTrak to gather satellite orbital parameters. TLE data consists of two lines for each satellite:

- **Line 1**: Contains satellite identification, orbital parameters, and epoch date.
- **Line 2**: Contains the satellite's inclination, longitude of the ascending node, eccentricity, argument of perigee, mean anomaly, and mean motion.

### 3.2 Available Satellites

For this project, we considered several well-known satellites:

- **ISS** (International Space Station)
- **Hubble**
- **Landsat 8**
- **StarLink-32637**

These satellites were chosen to provide variety in the types of orbits (e.g., low Earth orbit for ISS, geostationary orbit for Starlink).

### 3.3 Satellite Selection Process

The user can select the satellite of interest by specifying the satellite name (e.g., "ISS", "Hubble"). The program allows the user to input the satellite name, fetch its TLE data, and predict its orbit.

## 4. Technologies and Methodologies

### 4.1 Technologies Used

- **Python**: The primary programming language for the project.
- **Matplotlib**: Used for 3D plotting and data visualization of the satellite's orbit and the Earth-Moon system.
- **NumPy**: Used for numerical computations, particularly for generating spherical coordinates and matrix operations for orbital path predictions.
- **Pyephem / Skyfield / sgp4**: Libraries for orbit prediction based on TLE data. We used these libraries to calculate the satellite's position over time.
- **PIL / Matplotlib (image processing)**: Used for loading and applying Earth and Moon textures to the 3D models.

### 4.2 Methodology

#### 4.2.1 Step 1: Parse TLE Data

- **Input**: The program accepts TLE data as input, which could be provided in a file or directly through the code.
- **TLE Parsing**: The `sgp4` or `pyephem` library is used to parse TLE data, creating a satellite object that encapsulates its orbital parameters.

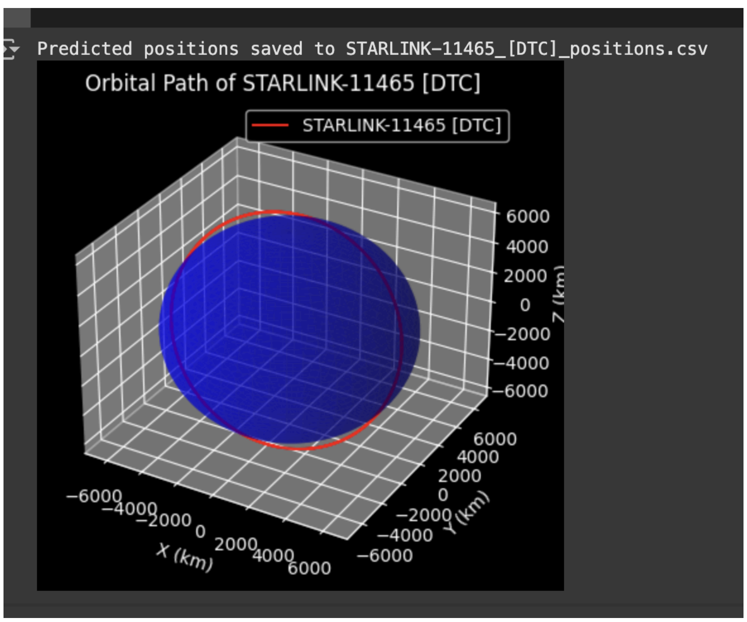
#### 4.2.2 Step 2: Satellite Position Prediction

- **Orbit Calculation**: The satellite's position is predicted using the TLE data over a specified time range.
  - The `sgp4` library is used to predict the satellite's position based on its orbital elements.
  - For each time step, the satellite's position (latitude, longitude, altitude) is calculated in a 3D coordinate system (X, Y, Z).

#### 4.2.3 Step 3: 3D Visualization

- **Create Spherical Models**: We used spherical coordinates to create the 3D surfaces for Earth and Moon. These surfaces were represented by matrices of coordinates (`X`, `Y`, `Z`), which were generated using the following formula:
    - X=R⋅cos(u)⋅sin(v)X = R \cdot \cos(u) \cdot \sin(v)
    - Y=R⋅sin(u)⋅sin(v)Y = R \cdot \sin(u) \cdot \sin(v)
    - Z=R⋅cos(v)Z = R \cdot \cos(v) where `R` is the radius (Earth or Moon radius), and `u` and `v` are angular parameters.
- **Apply Textures**: Earth and Moon textures were applied to the spherical surfaces using the `plot_surface()` function in Matplotlib. We normalized texture image data to fit within the expected range of `[0, 1]` to prevent errors during rendering.

#### 4.2.4 Step 4: Satellite Orbit Rendering

- **Orbit Path**: The satellite's predicted orbit was plotted in 3D space, showing the satellite's trajectory over time. The orbit was shown as a red line on the 3D plot.
- **Earth and Moon**: Both the Earth and Moon were displayed in the same plot with textures applied, giving a realistic view of the Earth-Moon system with the satellite orbit overlaid.
- 


#### 4.2.5 Step 5: Output

- **Image Generation**: The resulting 3D plot, including the satellite's orbit, Earth, and Moon with textures, was saved as a high-quality PNG image.
- **File Management**: The output images were saved in a folder named "output_images" for easy access and further analysis.

# 5. Results and Visualizations

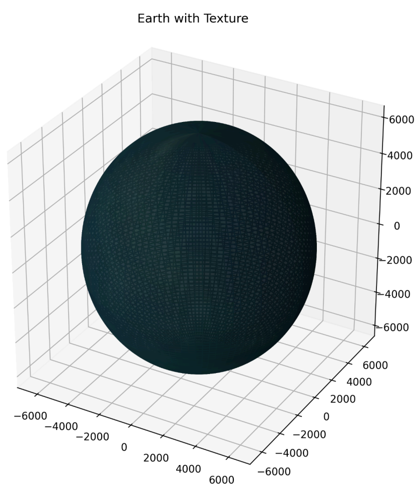
## 5.1 Visualization of the Orbit

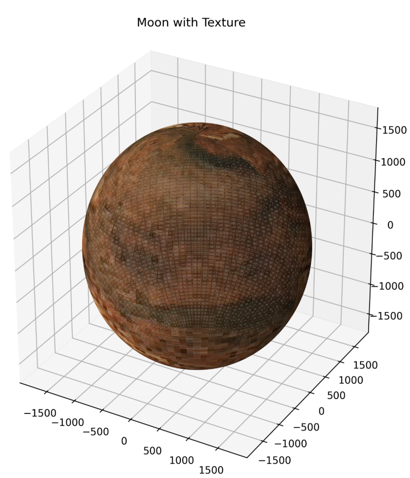
The final result is a 3D plot showing:

- The Earth with a realistic texture.
- The Moon with a simple gray texture (or optionally, with its own texture).
- The satellite's orbit path in space over a specified time range.

## 5.2 Sample Output

Below is an example of the output file generated for the ISS satellite:




-

## 6. Conclusion

The project successfully implemented a system to predict satellite positions using TLE data and visualize them in 3D. By incorporating real textures for both the Earth and Moon, the project provided an immersive and accurate visualization of satellite orbits. The technology stack, including Python, Matplotlib, and libraries like `sgp4`, enabled effective parsing, prediction, and visualization of the orbital data.

## 7. Future Work and Improvements

- **Interactive Visualizations**: Implementing interactive 3D plots where users can rotate and zoom in on the Earth, Moon, and satellite orbit.
- **Real-Time Tracking**: Integrating real-time satellite tracking data to predict current positions dynamically.
- **More Textures**: Applying more detailed textures for the Moon and Earth, such as maps for the Moon's surface or high-resolution Earth imagery.

Link to file : isro_task_2.ipynb
Satellite output csv: satellite_otuput
TLE txt file: last30.txt
Source of TLE data: celeswebsite_last30days(22dec24)
Source of moon texture :https://visibleearth.nasa.gov/?page=13
Source of earth texture
:https://visibleearth.nasa.gov/images/153701/desert-peninsula-in-turkmenistan