Practice 1: Introduction to Linux and Python

Nanometric System Simulation - Nanoscience and Nanotechnology UAB 2022/23

Xabier Oianguren Asua 1456628

Question 1 : List the most important Linux commands used today.

Some of the most relevant commands employed in Linux today are among others:

- `pwd`: This prints the working directory or current path of the terminal's interpreter.
- `ls`: This allows to list the directories and files in the current path.
- `cd <path>`: This allows to change the working directory to the one given by `<path>`, which can be a relative or an absolute path.
- `mkdir <name_dir>`: This allows to create a directory in the current path, with the name of the directory being `<name_dir>`.
- `python3`: This allows to open a Python3 interpreter if it is installed.
- `rm <filename>`: This allows to remove the file `<filename>`.

Question 2 : Modify one of the python programs used today.

We choose the Python script named `parabolic_motion.py` of the teaching repository [1]. Our modified version can be found in Ref. [2] with the name `Lab_01.py`. The improvement/modification we incorporated is that we add a dimension to the modeling, allowing to have a parabolic motion in 3D. For this, we set an additional input for the user, such that the script asks now for both the azimuth and polar angles for the initial velocity vector. Then, the plotting part of the script is also modified to allow a 3D plot of the desired trajectory. An example of the output can be found in Figure 1, while the main routine (that was modified) can be found in Listing 1.

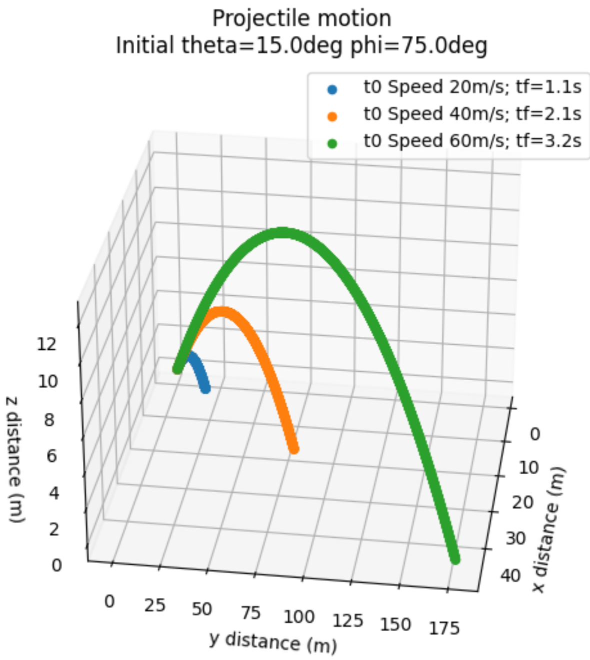

Figure 1: Example output plot after inputting 15 deg as polar angle and 75 deg as azimuth.

Listing 1: Main routine of the script that was modified following the explanation in the text.

```python
def draw_trajectory(us, theta_deg, phi_deg):
    #convert angle in degrees to rad
    theta = np.radians(theta_deg)
    phi = np.radians(phi_deg)
    #gravity acceleration in m/s2
    g = 9.8
    # Initialize figure
    fig = plt.figure(figsize=(10,10))
    ax = fig.add_subplot(111, projection='3d')

    for u in us:
        # Time of flight
        t_flight = 2*u*np.sin(theta)/g
        # find time intervals
        intervals = np.arange(0, t_flight, 0.001)
        # create an empty list of x,y and z coordinates
        x = []
        y = []
        z = []
        #Do a loop over time calculating the coordinates
        for t in intervals:
            x.append(u*np.cos(theta)*np.cos(phi)*t)
            y.append(u*np.cos(theta)*np.sin(phi)*t)
            z.append(u*np.sin(theta)*t - 0.5*g*t*t)
        #Plot the results
        cmap = ax.scatter3D(x,y,z, label=f"t0 Speed {u}m/s; tf={t_flight:.2}s") # c=intervals, cmap='winter',
    #fig.colorbar(cmap, ax=ax)
    ax.set_xlabel("x distance (m)")
    ax.set_ylabel("y distance (m)")
    ax.set_zlabel("z distance (m)")
    ax.set_title(f"Projectile motion\nInitial theta={theta_deg:.4}deg phi={phi_deg:.4}deg")
    ax.legend()
    plt.show()
```

References

[1] "Github repository with the python scripts of the professor employed for the lab practice." https://github.com/jfaraudo/Python_examples.

[2] "Github repository with the python script generated for the report." https://github.com/Oiangu9/_Miscellaneous/tree/main/SSN.