Python Programming Exercise

Prof. Rodrigo da Silva Guerra

25 de fevereiro de 2019

1 Object Oriented Programming Exercise: High-Altitude Balloon

For this exercise you will write a program to simulate, in a simplified way, the release of a high-altitude balloon. This balloon will be released from Da'an Park and it will raise, under the influence of wind, up to an altitude of 60 thousand meters. When it reaches 60 thousand meters, it will burst and it will start to fall. It will fall, also under the influence of the wind, until it crashes on the ground. Your program should be able to compute the whole trajectory of the balloon, from its release up to is crashing site.

1.1 Problem Details

We will consider an spherical balloon of radius 0.5 m, weigh of 10 kg, filled up with a low density gas that will provide a buoyancy force of 150 N – for simplification, we will consider the gravity to be always constant, equal to $9,8$ m/s^2 (weight force of 98 N pulling down).

The wind will have a velocity of $5,14$ m/s (aproximately 10 knots), and its direction will be given by the equations

$$
\begin{aligned}
v_x &= 5,14\sin(\sqrt{(x-c_x)^2+(y-c_y)^2}) \\
v_y &= 5,14\cos(\sqrt{(x-c_x)^2+(y-c_y)^2}) \\
v_z &= 0,0
\end{aligned}
$$

where x, y are respectively the longitude e latitude, in degrees, and v_x, v_y and v_z are the velocity components in that location, in m/s. The variables c_x and c_y are the longitude and latitude of the location where the balloon will be released, in Da'an Park..

The Figure 1 shows how the wind direction changes according to the location of the balloon. In this model the wind is independent of the altitude, and its vertical component will be null..

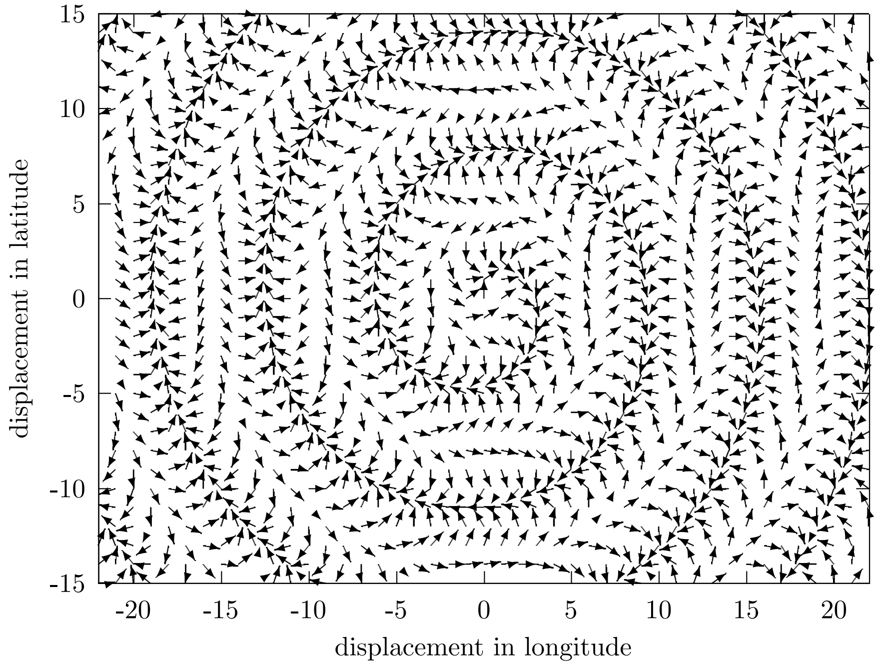

Figura 1: Direction of the wind depending on the location.

This wind will cause a drag force[1] to be computed by

$$
\begin{aligned}
f_x &= (v_x - \dot{x})\pi r^2 C_1 \\
f_y &= (v_x - \dot{y})\pi r^2 C_1 \\
f_z &= -\dot{z} * \pi r^2 C_1
\end{aligned}
$$

where $\dot{x}$, $\dot{y}$ and $\dot{z}$ are the velocity components, in m/s, r is the radius of the balloon, in m, and C_1 is a constant equal to $0,1$.

Subject to gravity, buoyancy, and wind drag, the balloon will raise until it surpasses 60 thousand meters of altitude. Then it will burst, loosing all its buoyancy and reducing its radius to $0,05$ m. Then the balloon will fall. During the fall the balloon will keep the same mass, and it will be subject to the same wind conditions, but due to the reduced radius, the drag force will change.

[1]this is just a simplification, not very realistic aerodynamics

1.2 Dynamic Simulation

To simulate the balloon you will need to sum the weight force, the buoyancy and the drag components. The acceleration will be computed by the second law of Newton

$$
\begin{aligned}
\ddot{x} &= \bar{f}_x/m \\
\ddot{y} &= \bar{f}_y/m \\
\ddot{z} &= \bar{f}_z/m
\end{aligned}
$$

where $\ddot{x}$, $\ddot{y}$ and $\ddot{z}$ are the acceleration components, and m is the balloon mass.

After this, using Euler method, this acceleration will be integrated to compute velocity, and integrated again to compute position

$$
\begin{aligned}
\dot{x}_{k+1} &= \dot{x}_k + T\ddot{x}_k & \qquad x_{k+1} &= x_k + \frac{T\dot{x}_k}{111.111\cos\left(y_k \frac{\pi}{180^\circ}\right)} \\
\dot{y}_{k+1} &= \dot{y}_k + T\ddot{y}_k & \qquad y_{k+1} &= y_k + \frac{T\dot{y}_k}{111.111} \\
\dot{z}_{k+1} &= \dot{z}_k + T\ddot{z}_k & \qquad z_{k+1} &= z_k + T\dot{z}_k
\end{aligned}
$$

where k and $k+1$ indicate time kT, and $(k+1)T$, where T is the sampling period. Notice that the increments in meters are converted[2] toi degrees in latitude, divided by 111.111, and in longitude, divided by $111.111\cos\left(y_k \frac{\pi}{180^\circ}\right)$.

1.3 Implementation

Please create an object oriented code, including classes and methods. The Figura 2 shows the expected result

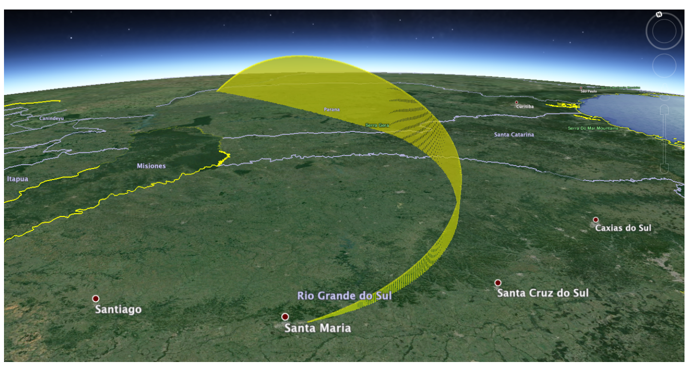

Figura 2: Expected Result

[2]This is an approximation, and the cos function in the computer requires argument in radians.