Modelling Free-fall of a Skydiver using the Euler Method.

Angus Brewster

November 2023

Task

Use python to simulate the free-fall of skydiver, Felix Baumgartner, using the Euler method to solve the equations of motion considering both constant and altitude dependent drag factors.

1 History

On the 14^{th} October 2012 Felix Baumgartner set the current world record for both highest altitude skydive as well as highest speed skydive. He achieved this with the Red Bull Stratos project, where the Austrian Skydiver was dropped from a high altitude helium balloon in the stratosphere above New Mexico, United States. In Baumgartner's descent he reached a speed of 1,357km/h and deployed his parachute earlier than planned at 4 minutes and 19 seconds after dropping. Whilst the drop is famous for breaking the 1960 record for highest-altitude jump (set by Joseph Kittinger, Baumgartners mentor), he also broke the unofficial record for the highest manned balloon flight of 37,640m.

2 Theory

For an object travelling through air, the air resistance is proportional to the square of the velocity and acts in the opposite direction:

$$\mathbf{F} = -\mathbf{k}\mathbf{v^2}\hat{\mathbf{v}} = -\mathbf{k}|\mathbf{v}|\mathbf{v} = -\mathbf{k}\mathbf{v}\mathbf{v}. \tag{1}$$

The constant k can be calculated using:

$$k = \frac{C_d \rho_0 A}{2} \tag{2}$$

Where C_d is the drag coefficient, A is the cross sectional area of the projectile and ρ_0 is the air density. In order to be a realistic simulation of projectile motion, the acceleration must vary, so Newton's equations of motion produce second order ordinary diferential equations:

$$\frac{dy}{dt} = f(y,t), \tag{3}$$

$$m\frac{dv_y}{dt} = -mg - kv_y v_y. \tag{4}$$

Euler's method states that $y_{n+1} = y_n + h \cdot f(t_n, y_n)$. From this we can find equations for $v_{y,n+1}$, $y_n + 1$ and $t_n + 1$:

$$v_{y,n+1} = v_{y,n} - \delta t(g + \frac{k}{m}|v_{y,n}|v_{y,n}), \tag{5}$$

$$y_{n+1} = y_n + \delta t \cdot v_{y,n}, \tag{6}$$

$$t_{n+1} = t_n + \delta t. \tag{7}$$

3 Part A

The equations (3) and (4) can be solved to give analytical solutions for y and v_y:

$$y = y_0 - \frac{m}{k}\log_e[\cosh(\sqrt{\frac{kg}{m}} \cdot t)], \tag{8}$$

$$v_y = -\sqrt{\frac{mg}{k}}\tanh(\sqrt{\frac{kg}{m}} \cdot t). \tag{9}$$

Equations (8) and (9) were modeled in python using a for loop to calculate y values for a set number of points, with an additional statement preventing a negative height error. When plotted the analytical solution to the equations of motion gives:

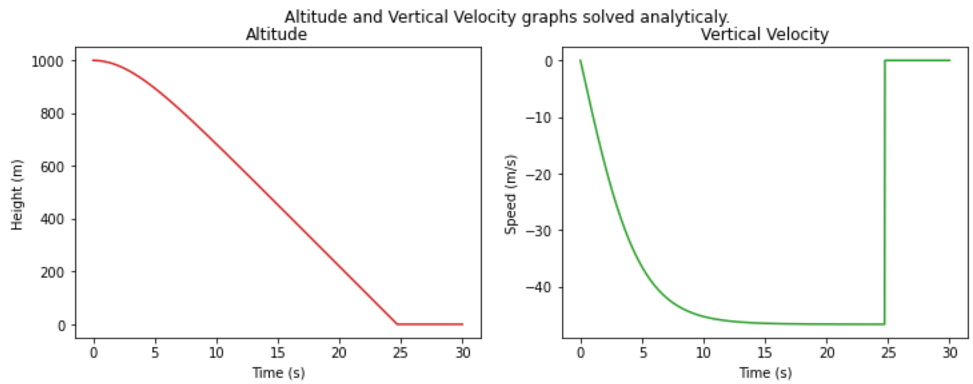

Figure 1: Altitude and Velocity graphs solved analytically for part A.

4 Part B

Equations (5)(6)(7) were then used as solutions to the free-fall problem using the Euler method. This was again modeled with python using a for loop and non zero height argument.

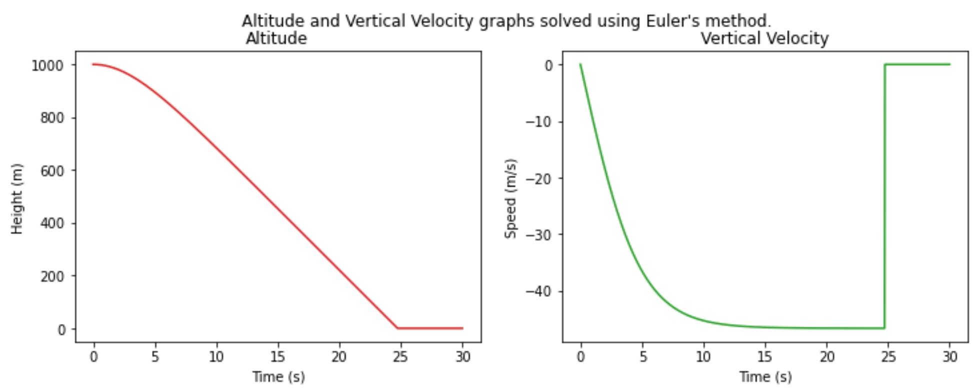

Figure 2: Altitude and Vertical Velocity graphs solved using Euler's method.

A starting height(y_0) of 1km and initial velocity(vy_0) of 0 were used in the calculation. The C_d used was 1 as it is within the values of 1.0-1.3 given in the exercise and was also used in other sources on the web[2] [1]. The value used for A was $0.6m^2$, and m of $80kg$ [2]. The simulation is calculated over 30 seconds so that it can be observed over its whole parabola and what happens when the skydiver reaches the ground (if he weren't using a parachute).
You can see that this graph is very similar to the analytical method in Part A. The Number of points (recorded as 'resolution') was then varied as this results in a varying δt:

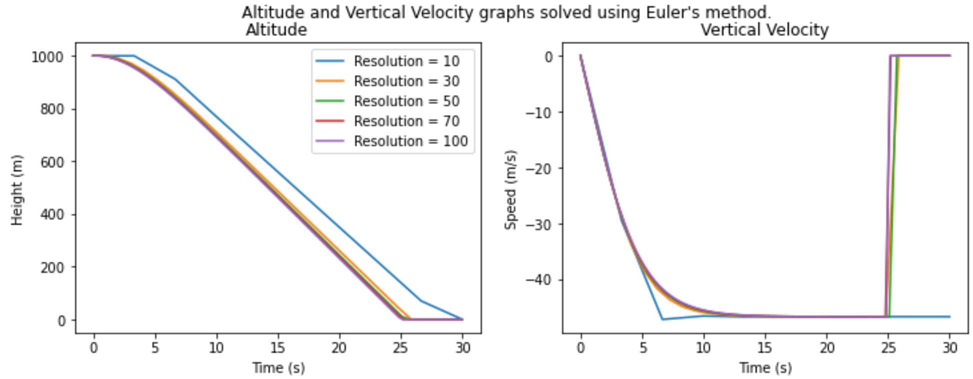

Figure 3: Altitude and Vertical Velocity graphs solved using Euler's method, with varying δt

With lower resolutions, the height is over estimated per value t and the speed under estimated but as δt approaches infinity, the model tends to approximate the analytical prediction. This is because as δt is decreased the Euler approximation better approximates an integral rather than a sum.

5 Part C

The problem was now modelled in a more realistic manner. At the height Baumgartner actually jumped the air density was lower than at sea level so the drag factor k is much lower initially. To account for this the k is replaced with $k(y)$: k as a function of height. $k(y)$ was found using

$$\rho y = \rho_0 e^{(}\frac{-y}{h}). \tag{10}$$

and (2). The value of h was given to be $7.64km^2$. In the code this was included by including a k values numpy array that fills up as the for loop iterates through i. The Euler Method solution presented:

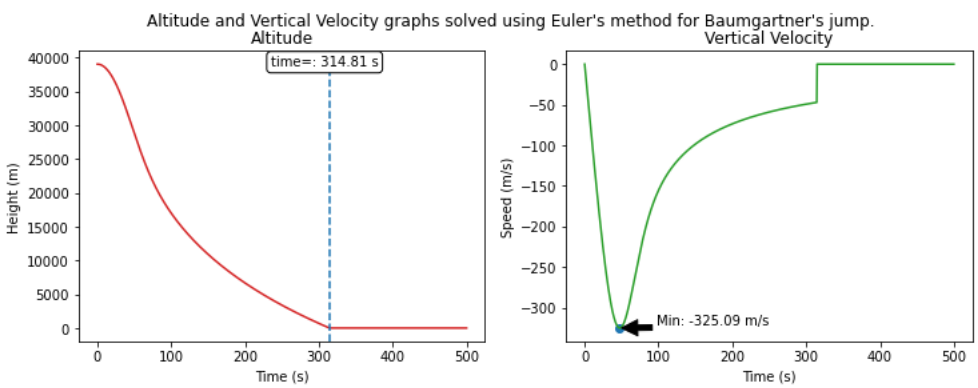

Figure 4: Altitude and Vertical Velocity graphs solved using Euler's method for Baumgartner's jump.

Terminal Velocity for a given ρ can be calculated using the follow equation:

$$V_t = \sqrt{\frac{2mg}{\rho A C_d}}. \tag{11}$$

The expected terminal velocity for a ρ of 1.2 is ≈ 46.7 however as seen in Figure4 the actual velocity was significantly higher at 325.09m/s. This is because in this simulation ρ is modelled as variable as well as g.

Varying the Jump parameters gives some interesting in site into the equations modelling the jump. First of all varying the mass, and thus the k constant results in:

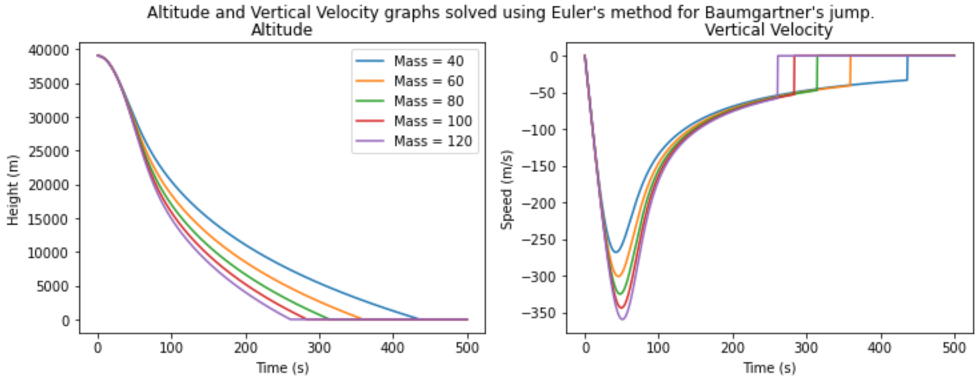

Figure 5: Altitude and Vertical Velocity graphs solved using Euler's method for Baumgartner's jump, varied over values of m.

This means that as m increases and k decreases the terminal velocity increases and the time to reach the ground decreases.

Varying the initial height of the drop also has an impact on the terminal velocity and air time:

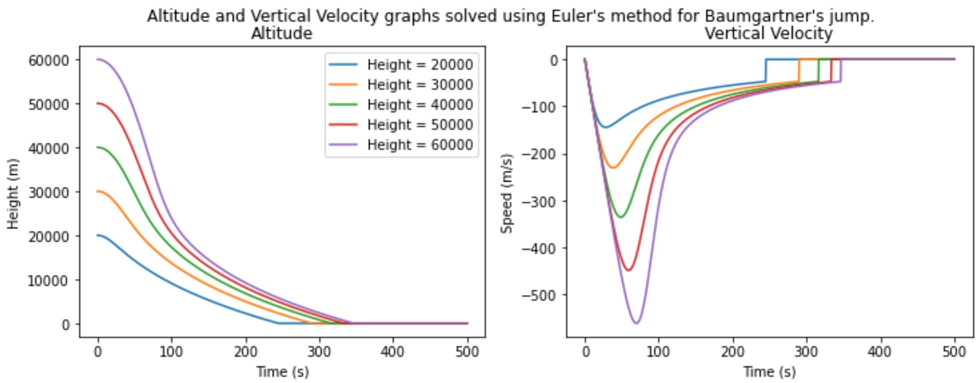

Figure 6: Altitude and Vertical Velocity graphs solved using Euler's method for Baumgartner's jump, varied over values of y0.

As the initial height of the drop increases so does both the air time and the terminal velocity.

6 discussion

Euler's method, although intuitive, has downsides. When each step is computed it is possible that part of the function is missed if it has intricate details smaller than the step size, this results in an error. As steps are summed on one another the errors accumulate resulting in a significant final error in the data. This consistent truncation error means it is not possible for the Euler method to give a correct value but just close to it. Higher order accuracy methods such as Runge-Kutta Method (RK4) are more accurate as they calculate 4 times per time step. Whilst this may sound demanding, they are also generally more efficient than Euler's as although more calculations are required per time step they are less time step sensitive so often come out being computationally favourable regardless.

References

[1] Libre texts physics, 8.3: Body orientation during a skydive. Accessed: 27/11/2023.

[2] F R Greening. Baumgartner's jump and the physics of freefall. *Physics Education*, 48(2):139, mar 2013.