Simulating the free fall with the Euler Method

Date: 27/11/23; Student Nr: 2173375

Introduction

In this exercise, the free fall is simulated and analyzed under different scenarios and ultimately compared to the statistics of Baumgartner's free fall. In part a) I discuss the analytical solutions to the free fall with a constant drag factor and in part b) I employ and explore the Euler method for the same scenario. In part c) I use and explore the Euler method for the case of a varying drag factor with Baumgartner's parameters, and in part d) I investigate whether he breaks the sound barrier in the simulation.

Part a)

Method

In this part, the free fall problem was solved analytically for an object under constant gravity and with a constant drag factor, using the analytical functions given in the script for the height and the speed of the falling object as a function of time.

To implement this in my code, I iterate through the equations for height and speed for each time step, which produces values for y and v that gradually fill up the height and speed arrays until the height reaches zero or just below zero.

Results

Default parameters:

Initial height (m)	Initial velocity (m/s)	Mass (kg)	Cross-sectional area (m^2)	Drag coefficient	Air density (kg*m^-3)
1000	0	120	0.8	1.15	1.2

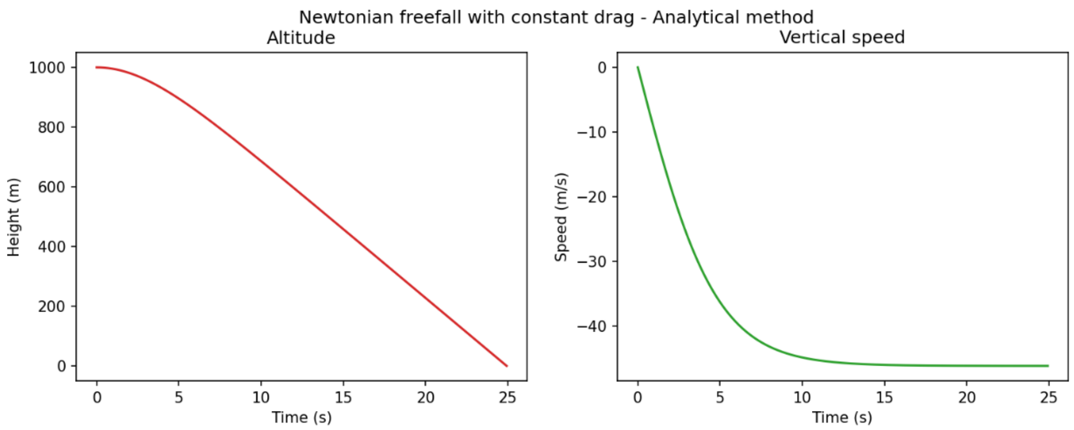


Fig 1.: (Analytical) graphs of Altitude and Speed vs Time with the default parameters and constant drag factor

Mass (kg)	Velocity at ground (m/s)	Duration of fall (s)	Time of max. velocity (s)
70	-35.2707	30.8503	30.8503
120	-46.1778	24.9203	24.9203
170	-54.9239	22.0802	22.0802

Table 1: Effect of object's mass

Area (m^2)	Velocity at ground (m/s)	Duration of fall (s)	Time of max. velocity (s)
0.2	-87.6106	17.1102	17.1102
0.8	-46.1778	24.9203	24.9203
2	-29.2069	36.3104	36.3104

Table 2: Effect of object's cross sectional area (and thereby the drag factor k)

0.5*(Cd*p0*A)/m	Velocity at ground (m/s)	Duration of fall (s)	Time of max. velocity (s)
default	-46.1778	24.9203	24.9203
Doubling mass and area	-46.1778	24.9203	24.9203

Table 3: Effect of keeping a constant ratio k/m

Discussion

In this scenario, the falling object experiences a downward gravitational force, and an air resistance (drag force) that is proportional to the square of the velocity and acts in the opposite direction. (written without vector notation for simplicity):

$$F_{net} = F_g + F_d$$

$$m\frac{dv}{dt} = mg - kv^2$$

$$\frac{dv}{dt} = g - \frac{k}{m}v^2$$

At first, as the object falls, the gravitational force dominates – the height vs time graph should thus initially resemble a concave upward parabola (as y(t) = 0.5*gt^2). The object thus picks up speed, which increases the air resistance, counteracting the gravitational force and thus decreasing the net acceleration- the slope in the velocity-time graph should thus now decrease. When the drag force equals the gravitational force, the acceleration becomes zero, and terminal (constant) velocity is reached. The height-time graph should therefore now be a straight line (constant slope), and the speed-time graph should be a horizontal line at the terminal velocity.

The graphs produced by the program (Fig.1) exhibit the above-described characteristics. However, according to the program, the maximum velocity is only reached when the object hits the ground, so even though the velocity is approximately constant from 15 seconds onwards (as seen by the horizontal line), it is still increasing very minimally until the end. The velocity at the ground is thus, strictly speaking, not the terminal velocity here.

A heavier object experiences a greater gravitational force, and thus accelerates more, reaching a higher terminal velocity more quickly. This can also be observed here (the maximum velocity approximating the terminal velocity), as seen in table 1.

A larger cross-sectional area (and therefore larger drag factor k) increases the drag force, which leads to a lower net force and thus decreases the acceleration on the object, leading to a slower fall and a lower terminal velocity, which table 2 shows.

With a constant ratio k/m, the motion is the same in all cases, as $v_t = \sqrt{\frac{mg}{k}}$, and the acceleration is determined by Eq. 6. This is confirmed in table 3.

Part b)

Method

Here the free fall problem with a constant drag factor was solved using the Euler method. In addition to height and speed, I also calculated the acceleration to visualize the solutions better. The same default parameters as in part a) were used, with the mass, cross-sectional area and drag coefficient adapted to a sky-diver and with the air density at ambient temperature and pressure.

The Euler method produces following equations for height, speed and acceleration, whereby the step size dt is specified beforehand:

$$a_n = g - \frac{k_0}{m} * {v_{n-1}}^2$$

$$v_n = v_{n-1} - dt * a_{n-1}$$

$$y_n = y_{n-1} + dt * v_{n-1}$$

To code it, I defined a function where arrays for height, speed and acceleration are iterated through and gradually filled with each time step, breaking the loop when the height reaches zero or just below zero.

To explore the effect of the step size on the motion and the error between the two methods, I plotted the RMSE of the height and speed values of the two methods, and the time it took for the program to complete the Euler method against the step size.

Lastly, I also implemented the modified Euler method, which adds an additional estimation step at the midpoint of the interval.

Results

With the default parameters, an initial height of 1000m, zero initial velocity and a step size of 0.1:

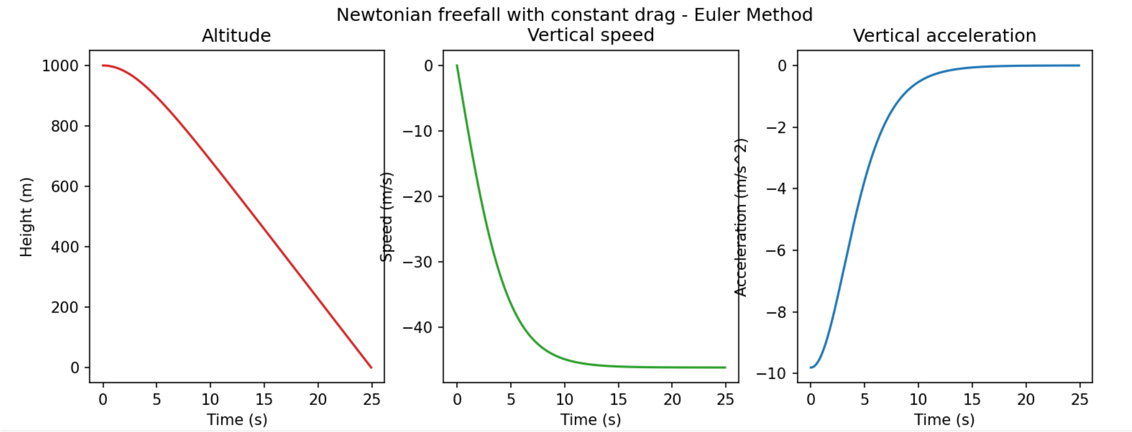


Fig 2.: Euler graph of Altitude, Speed and Acceleration vs Time with the default parameters, constant drag factor, and dt 0.1

Method	Velocity at ground (m/s)	Duration of fall (s)
Euler	-46.1789	24.9025
Analytical	-46.1779	25.0025
Difference	0.001016	0.100010

Step size	0.1
RMSE height	1.3868
RMSE speed	0.2526
Euler computation time(s)	0.0010

Tables 4 and 5: Results with dt 0.1

Method	Velocity at ground (m/s)	Duration of fall (s)
Euler	-46.1778	24.9172
Analytical	-46.1778	24.9172
Difference	0	0.000001

Step size	0.0001
RMSE height	0.0014
RMSE speed	0.0003
Euler computation time(s)	0.4997

Table 6 and 7: Results with dt 0.0001

Ratio k/m (Euler-0.1)	Velocity at ground (m/s)	Duration of fall (s)
k/m	-46.1789	24.9025
k/m*4	-23.0901	44.9045
k/m *1/4	-87.9404	17.1017

Table 8: Effect of varying the ratio k/m

Step size	0.1
RMSE height	0.0501
RMSE speed	0.0036
Euler computation time(s)	0.0065

Table 9: Using the modified Euler method

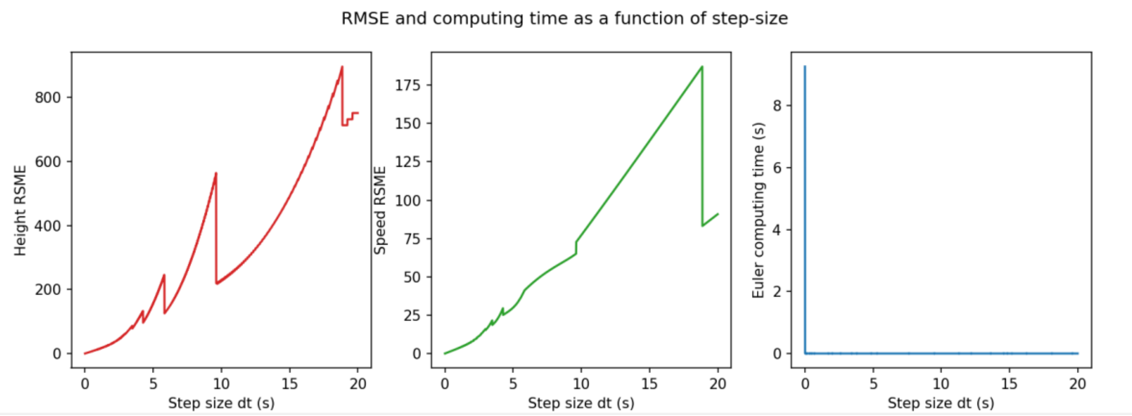
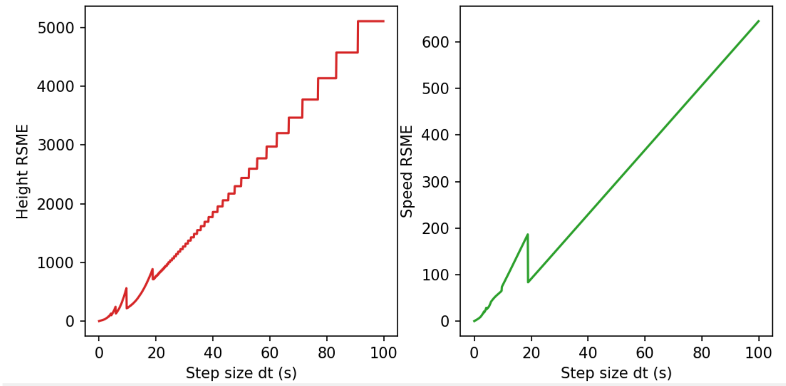


Fig 3.: Height and Speed RMSE, and Euler computing time as a function of step size



Step size	10000
RMSE height	inf
RMSE speed	69334
Euler comp. time(s)	0

Fig 4 and Table 10.: Effect of a large step size

Discussion

As can be seen in Tables 4-7, the difference between the outputted values and the RSME between the height and speed values obtained from both methods is very minimal, and the height and speed graphs are the same, confirming that former was implemented correctly. Here, the graph for acceleration also visualizes how terminal velocity is reached when acceleration is zero.

Tables 5 and 7 show how a smaller step size produces more accurate results, significantly reducing the RSME. Yet they also show how the computation time for the Euler method is increased, which is still relatively small for a step size of 0.0001, but quickly becomes a lot bigger upon further reduction. The bigger dt, the more inaccurate the results are, as the Euler method cannot capture the dynamics very well anymore.

The dependence of the RSME's and the computation on the step size is visualized in figure 2 and 3. In figure 2, the graphs are parabolic, which is expected as the Euler method is a first order method and the local error is thus proportional to the square of the step size. However as the step size becomes very large, figure 3 shows how the dependence has now become linear- suggesting that the quadratic term becomes negligible compared to some other error source.

With a dt of 10000, the program outputs an infinite RSME for height (table 10). It wasn't possible to make dt smaller than 0.00001, as the program gives a memory error, but one would expect the accuracy to decrease again, as the numerical precision is lost.

Just as in part a), the terminal velocity increases and the duration of the fall decreases with a decreasing ratio k/m (increasing mass).

Finally, I also observed how using the modified Euler method instead increases the accuracy (Table 9 compared to Table 5), as it uses a two-step process to update the solution.

Part c)

Method

In this part, the Euler method is used to solve the free fall problem with a *varying* drag factor, by using an exponentially varying air density.

To code this, I adapted my Euler method function to allow for the option of using a varying drag factor, with the first two equations now being:

$$k_n = k_0 * e^{\frac{y_{n-1}}{h}}$$

$$a_n = g - \frac{k_n}{m} * {v_{n-1}}^2$$

Results

Using the initial height of Baumgartner's jump (39000 m), the usual default parameters, a scale height of the atmosphere of 7640 m, and a step size of 0.01, following results are produced:

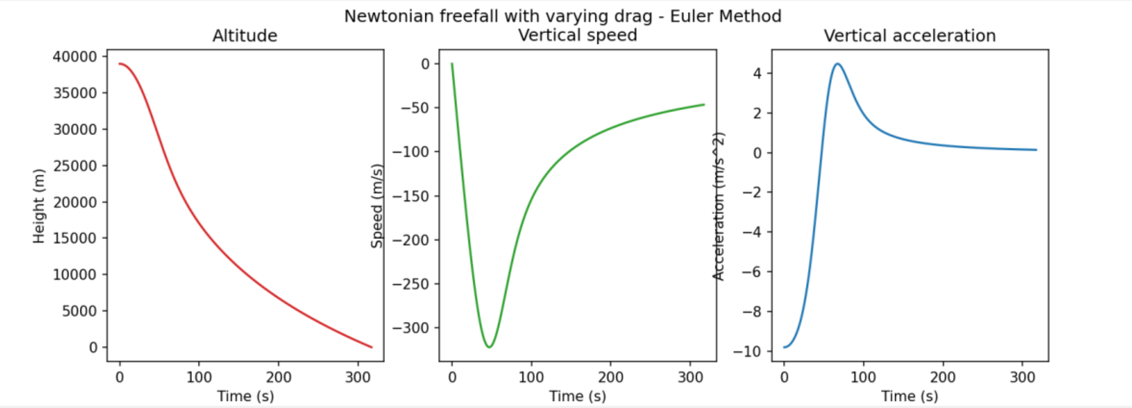


Fig 5: Euler method with varying drag factor and step size of 0.01

Method	Velocity at ground (m/s)	Duration of fall (s)	Maximum velocity (m/s)	Time of max. velocity (s)
Euler- dt 0.01	-46.5172	317.0632	-322.2335	46.9204

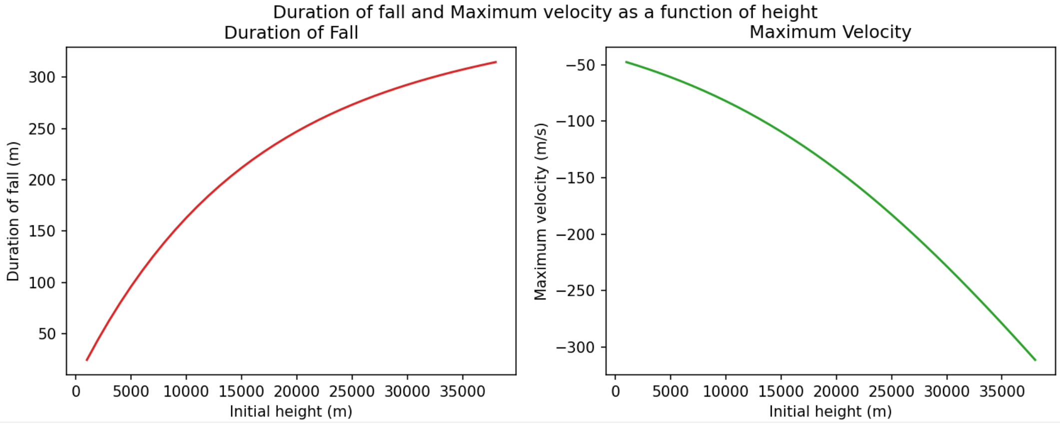


Fig 6: Effect of initial height on the duration of the fall and the maximum velocity achieved.

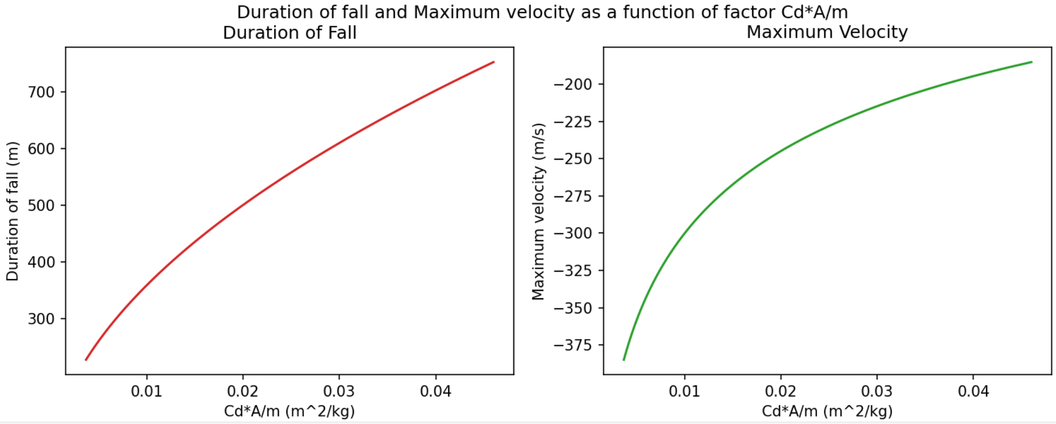


Fig 7: Effect of the factor Cd*A/m on the duration of the fall and the maximum velocity achieved.

Discussion

From the graphs for altitude, speed and acceleration following conclusions can be made: In the first part, the behaviour is roughly the same with a constant drag factor- This is as expected as at first, the gravitational force dominates, accelerating the object, which increases its velocity and thus the air resistance, decreasing the net acceleration.

The exponentially varying air density means that decreasing the height exponentially increases the drag factor and thus the drag force. The drag force therefore keeps exponentially increasing, reaching a point where the acceleration is zero and the velocity reaches its maximum value, as seen in the graph. Then the drag force dominates, leading to a net upwards acceleration that act against the direction of motion, which decreases the velocity of the fall. As the velocity decreases, the drag force decreases, causing a decrease in net acceleration until it becomes zero and the gravitational force is in equilibrium with the reduced drag force. This can also be observed in the graphs.

The maximum velocity reached is thus much greater than the velocity at ground (which, as seen by the non-existent plateau in the velocity graph is not the terminal velocity).

The obtained values are similar to Baumgartner's statistics. In the simulation, the fall takes 5 minutes and 17 seconds and a maximum velocity of -322.23 m/s is reached, while Baumgartner took 4 minutes and 19 seconds with a maximum speed of 373 m/s. This is likely due to some inaccuracies in the Euler method or/and a difference in the free fall parameters.

When duration of fall and maximum velocity reached are plotted against initial height, the graph is a concave upwards- and concave downward parabola, respectively. This is because with large initial heights, even though the object gains a larger maximum speed due to the gravitational force accelerating it over more time, the drag force also becomes exponentially more significant with a height increase, making the increase in duration less pronounced.

By increasing the factor Cd*A/m, the drag force is made more significant, which decreases the maximum velocity and increases the duration of the fall.

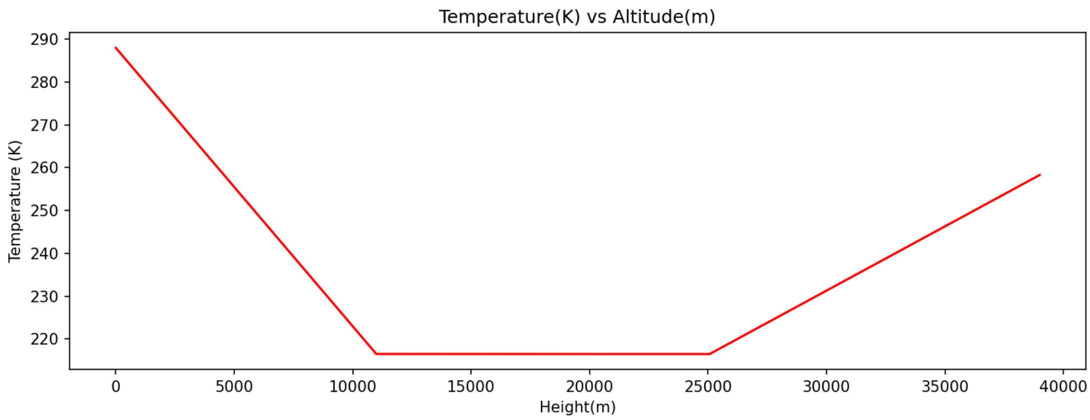
Part d)

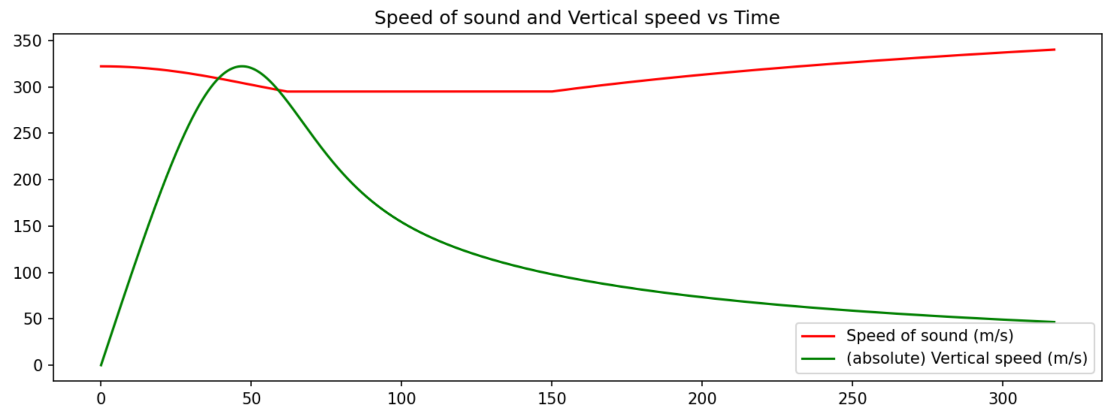
Here I calculated Baumgartner’s maximum Mach number during his fall and investigated whether he managed to break the sound barrier at any point.

This was done by calculating the speed of sound as a function of atmospheric temperature, which depends on the altitude. The Temperature-Altitude dependence is visualized in the Results section (both equations given in the script).

To find the period at which the sound barrier is broken, the difference between Baumgartner’s absolute speed and the speed of sound is calculated- if his absolute speed is bigger than the speed of sound, this will be a positive value. I thus find the first and the last time point at which this is the case. I then plotted his speed and the speed of sound as a function of time to visualise this sound barrier-breakage.

Results and Discussion





Baumgartner's maximum Mach number is 1.059103.

The sound barrier is broken between seconds 39.0504 and 58.8806, corresponding to respective heights of 32181.5288 m and 25932.5622 m.

For Baumgartner to break the sound barrier in this simulation, he has to weigh a minimum of 99kg (with a cross sectional area of 0.8m)- anything below and the barrier is not broken, or have a maximum cross sectional area of about 0.97 m^2 (with a mass of 120kg).

Conclusion

In conclusion, the Euler method managed to track the analytical solution very closely for the case of a constant drag factor, with minimal errors when an appropriate step size was used. The modified Euler Method increased this accuracy even more. For the (realistic) case of the free fall with a varying drag factor, the Euler method produced very similar values as Baumgartner's statistics for his fall. The accuracy could be improved, however, by using other methods to solve ODEs such as the Runge-Kutta methods.