

Simulation of projectile trajectory with Forward Euler method.

Tomasz Chwiej

13th October 2025

1 Introduction

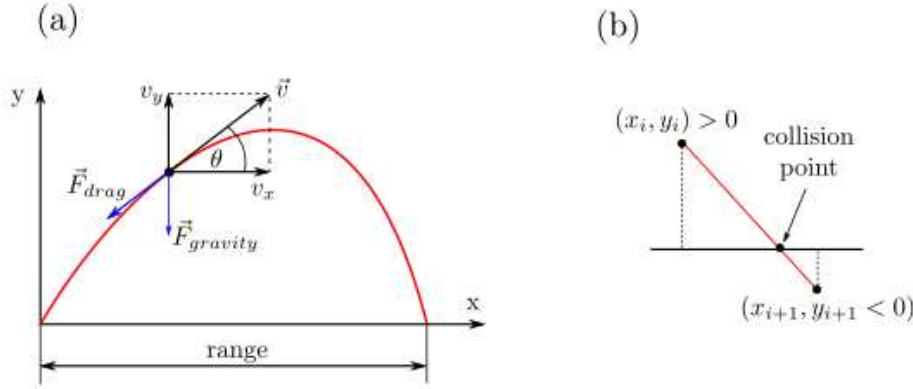


Figure 1: (a) Sketch of trajectory of inert projectile in gravitational field with drag force. (b) correction in position of object when it passes through the ground $y = 0$.

From the basic course in physics we know that the projectile launched with particular velocity v_0 with firing angle α_0 interacts with gravitational field of Earth which bends its trajectory leading to a collision of an object with the ground [Fig.1(a)]. In real simulations we must consider resistance forces exerted on object by surrounding medium, these could be the friction and drag forces, both counteracting the motion. Friction results from sticking of molecules to surface of the object and can be safely neglected for gaseous medium whereas the drag force is the consequence of throwing away the molecules from the trajectory of object what means that some part of kinetic energy of projectile is transferred to the surrounding medium. Considering the drag force in simulations of projectile flight becomes essential if one needs the reliable data. In project we will simulate the motion of inert missile with forward Euler method. We use the following expression for the gravitational force $\vec{F}_{grav}$

$$\vec{F}_{grav} = -m\vec{g}, \quad \vec{g} = [0, -g] \quad (1)$$

and for the drag force F_{drag}

$$\vec{F}_{drag} = -Dv^2, \quad v^2 = \sqrt{v_x^2 + v_y^2} \quad (2)$$

where: m is the mass, $\vec{v} = [v_x, v_y]$ is the velocity vector, D is constant drag force factor. Direction of drag force is opposite to direction of motion established by velocity

$$v_x = v \cos(\theta) \implies \cos(\theta) = \frac{v_x}{v} \quad (3)$$

$$v_y = v \sin(\theta) \implies \sin(\theta) = \frac{v_y}{v} \quad (4)$$

$$\hat{e}_v = [\cos(\theta), \sin(\theta)] \quad (5)$$

where $\theta(t)$ is the tangent angle for trajectory $\vec{r}(t)$. The components of drag forces can be written as

$$\vec{F}_{drag} = \hat{e}_v F_{drag} \quad (6)$$

$$F_{drag,x} = F_{drag} \cos(\theta) = -Dv v_x \quad (7)$$

$$F_{drag,y} = F_{drag} \sin(\theta) = -Dv v_y \quad (8)$$

The force given in Eq.8 would be a good approximation at the level of sea for e.g. the moving car, but the artillery missiles moves on the elevation of few kilometers where both the pressure and the air density decrease and following drag force also is getting smaller. Assuming the *isothermal* conditions the density should change according to $\rho(y) = \rho_0 e^{-y/y_0}$ (y_0 is the sea level) but we the temperature also decreases at higher elevation. The better approximation for density changes relies on *adiabatic* approximation

$$\rho(y) = \rho_0 \left(1 - \frac{a y}{T_0}\right)^\alpha \quad (9)$$

where $a = 6.5 \cdot 10^{-3}$ K/m and $\alpha = 2.5$ for the air, $T_0 = 293$ K temperature at the sea level, the improved drag force dependent on instantaneous altitude contains the weight function $\rho(y)/\rho_0$

$$F_{drag,x} = -Dv v_x \left(1 - \frac{a y}{T_0}\right)^\alpha \quad (10)$$

$$F_{drag,y} = -Dv v_y \left(1 - \frac{a y}{T_0}\right)^\alpha \quad (11)$$

1.1 Equations of motions

The knowledge about the forces exerted on moving object is needed for construction of *equation of motion* (EOM), this is derived from second Newton's law

$$\vec{F} = \vec{F}_{grav} + \vec{F}_{drag} \implies \frac{d^2 \vec{r}}{dt^2} = \frac{\vec{F}}{m} \quad (EOM) \implies \begin{cases} \frac{d^2 x}{dt^2} = -\frac{1}{m} Dv v_x \left(1 - \frac{a y}{T_0}\right)^\alpha \\ \frac{d^2 y}{dt^2} = -g - \frac{1}{m} Dv v_y \left(1 - \frac{a y}{T_0}\right)^\alpha \end{cases} \quad (12)$$

1.2 Solution of ODE - Forward Euler method

Now we aim at solving the system of two coupled 2-nd order ODE given in Eq.12 with one of the methods devised for solving 1-st order ODE, therefore we transform this system into another one containing four 1-st order ODE

$$\begin{cases} \frac{dx}{dt} = v_x \\ \frac{dv_x}{dt} = -\frac{1}{m} Dv v_x \left(1 - \frac{a y}{T_0}\right)^\alpha \\ \frac{dy}{dt} = v_y \\ \frac{dv_y}{dt} = -g - \frac{1}{m} Dv v_y \left(1 - \frac{a y}{T_0}\right)^\alpha \end{cases} \quad (13)$$

The problem is quite simple and not computationally demanding, thus the use of simple Forward Euler method is justified. Recall its definition

$$\frac{du}{dt} = f(t, u) \implies u_{i+1} = u_i + \Delta t f(t_i, u_i) \quad (14)$$

we apply this for a system given in Eq.13

$$\begin{cases} x_{i+1} = x_i + \Delta t V_{x,i} \\ v_{x,i+1} = v_{x,i} - \Delta t \frac{1}{m} D v v_x \left(1 - \frac{a y}{T_0}\right)^\alpha \\ y_{i+1} = y_i + \Delta t v_{y,i} \\ v_{y,i+1} = v_{y,i} - \Delta t \left(g + \frac{1}{m} D v v_y \left(1 - \frac{a y}{T_0}\right)^\alpha\right) \end{cases} \quad (15)$$

Now, notice that: (1) for $a = 0$ we recover the original drag force without elevation improvement, and (2) for $D = 0$ we get free flight without air resistance.

1.3 Numerical algorithm

Before we write the algorithm at its final form we shall consider the useability of the results. Let's assume that we are interested in launching the projectile with initial velocity v_0 at angle θ_0 with respect to the ground, then following its trajectory until it hits the ground and measuring the distance between both points. By choosing any time step $\Delta t > 0$ we make the simulation iteratively employing Eqs.15 until we get $y_{i+1} < 0$. However, we need to know the collision point exactly for $y = 0$ [see Fig.1(b)]. Because the Forward Euler shifts the object during the single time step along the straight line, the ground is reached at time t such as

$$y(t) = \frac{y_{i+1} - y_i}{\Delta t} (t - t_i) + y_i = 0 \quad (16)$$

the fraction

$$r = \frac{t - t_i}{\Delta t} \implies r = \frac{y_i}{y_i - y_{i+1}} \quad (17)$$

defines the fraction of time step after which the collision will be registered. To summarize, just after the collision ($y < 0$) is detected the position of the object must be corrected. In following pseudocode there is used the *do-while loop* as we do not know how many time steps should be taken to get the collision.

```

initialization: t=0, Δt, v0, θ0,
    x = y = 0
    vx = v0 cos(θ0)
    vy = v0 sin(θ0)
    RUN = true
do{
    save for collision correction:
        xold ← x
        yold ← y
        told ← t

    calculate new position:
        Fx = Eq.15
        Fy = Eq.15
        x ← x + Δt vx
        vx ← vx + Δt  $\frac{F_x}{m}$ 
        y ← y + Δt vy
        vy ← vy + Δt  $\frac{F_y}{m}$ 

```

```

    detect the collision + correct position:
        if (y<0){
             $r = \frac{y_{old}}{y_{old}-y}$ 
             $x \leftarrow x_{old} + (x - x_{old})r$ 
             $y \leftarrow y_{old} + (y - y_{old})r$  //should give y=0
             $t \leftarrow t - (1 - r)\Delta t$ 

            RUN=false //stop simulation
        }

    write to file: t,x,y
} while (RUN)
}

```

2 Practical part

Tasks to do

1. Implement algorithm given in Sec.1.3 on computer, take the following as input parameters: Δt , $x = y = 0$, v_0 , θ_0 , D , a , g , m , α , T_0 . Because it will be executed several times it'd be better to put it in separate function. To this end assume that all parameters, but angle, are expressed using SI units: $x, y = [m]$, $v = [m/s]$, $m = [kg]$, $t = [s]$, $a = [K/m]$, $D = [Ns^2/m^2]$, $T = [K]$. In all calculations use fixed values of $g = 9.81$ and $T_0 = 293$.
2. Check the correctness of algorithm simulating the trajectory without the drag force, set $D = 0$, then derive analytical formulae for the range x_{max} and the time of flight t_{max} for given v_0 and θ_0 . Perform simulations for the following input parameters: $m = 1$, $v_0 = 100$, $\theta = 45^\circ$ and $\Delta t = t_{max}/n$, $n = 10, 20, 50, 100, 200, 500$. Make the plot containing these trajectories. For each trajectory calculate the global error at the collision point $E_{glob} = x_{max}^{exact} - x_{max}^{num}$. Make a plot of global error in function of time step Δt , use double logarithmic scale.
3. Enable the drag force but turn off the altitude amendment assuming $a = 0$. Based on previous results choose some small Δt and find trajectories for $D = 0; 1 \cdot 10^{-4}; 2 \cdot 10^{-4}; 5 \cdot 10^{-4}; 10^{-3}$ and $v_0 = 100$, $\theta = 45^\circ$, $m = 1$. Draw trajectories in one plot. Next for $D = 0; 1 \cdot 10^{-3}; 2 \cdot 10^{-3}$ find the range of projectile for $v_0 = 100$ and $m = 1$ scanning the firing angle $\theta_0 = 15^\circ, 16^\circ, \dots, 64^\circ, 65^\circ$. Draw the range x_{max} as function of initial firing angle for each D in separate figures.
4. Drag force with altitude correction. Assume parameters: $m = 20$, $v_0 = 700$ (firing conditions close to some artillery cannons) and $D = 1 \cdot 10^{-3}$, then turn on altitude correction taking $a = 6.5 \cdot 10^{-3}$. Simulate trajectories of inert projectile for: $\alpha = 35^\circ$ and $\alpha = 45^\circ$. Repeat simulations for $a = 0$. Compare the four trajectories in one plot.
5. At home prepare the report. Analyze the global error in Forward Euler method (does it behave as we expect?), the influence of drag force, without and with altitude improvement, on the trajectory shape and the range of projectile.

3 Example results

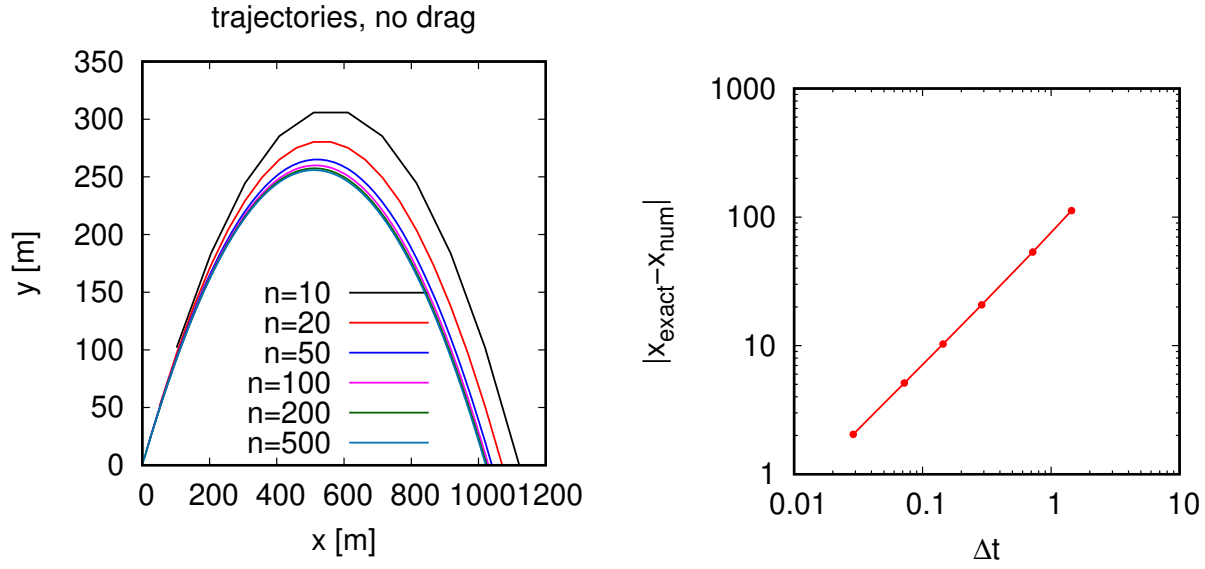


Figure 2: LEFT: Example trajectories of projectile calculated with Forward Euler method for different number of nodes $\Delta t = t_{\max}/n$ for fixed $t_{\max}$. RIGHT: Global error for $x(t_{\max})$ in function of Δt in log-log scale. Results (left & right) obtained without drag force.

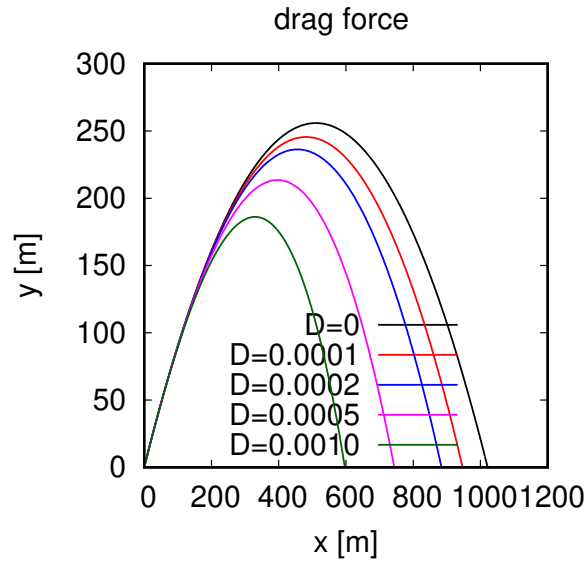


Figure 3: Example trajectories of projectile with drag force taken into account, simulations were made for the same initial parameters: $\Delta t, x_0, y_0, v_{x0}, v_{y0}$.

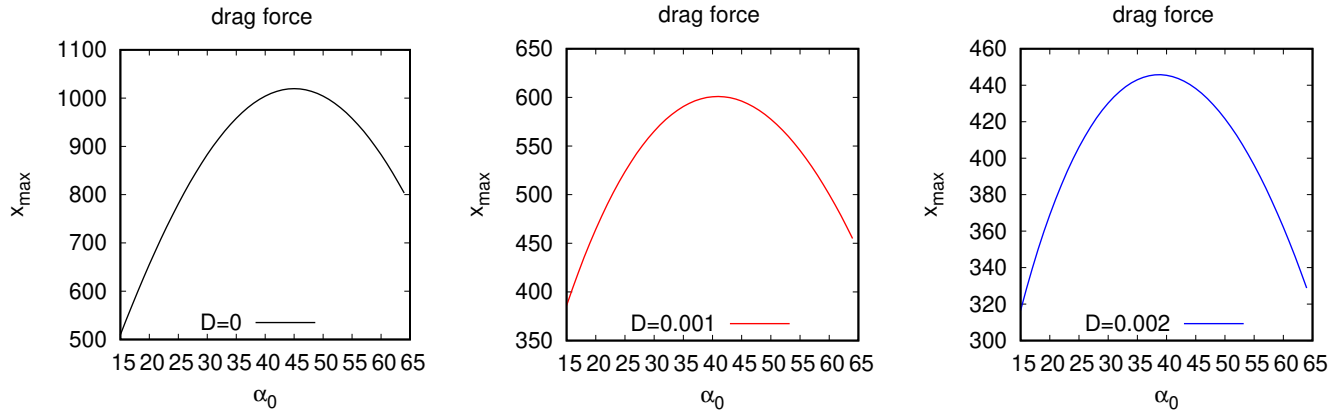


Figure 4: The range ($x_{\max}$) of projectile as function of firing angle α_0 for three drag force parameters D and the same initial parameters $\Delta t, x_0, y_0, v_0$. Maximal range shift towards the smaller angles for stronger air resistance.

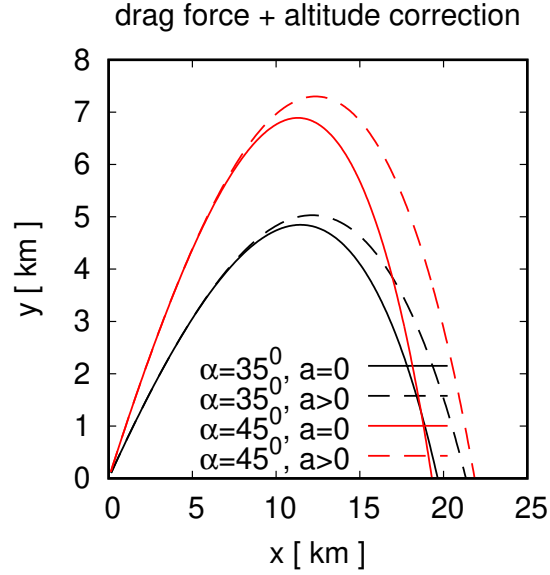


Figure 5: Example trajectories of projectile with (dashed) and without (solid) an altitude correction in the drag force taken into account.