# Computer simulations in physics

Project 1

Semi-realistic projectile motion simulation

Dávid Pesznyák Eötvös Loránd University Physics MSc

## Contents

1	Introduction						
<b>2</b>	Bas	ic concepts and ideas	<b>2</b>				
	2.1	The equation of motion $\ldots \ldots \ldots$	2				
	2.2	Drag effects	3				
	2.3	Barometric formula	4				
	2.4	Curvature of Earth	4				
3	Nur	nerical methods	5				
4	Con	nparison of analytical and numerical results	6				
	4.1	Results without air resistance	6				
	4.2	Results with Stokes drag	10				
5	Nur	nerical analysis of semi-realistic effects	12				
	5.1	Comparison of Stokes and Newton drag	12				
	5.2	Newton drag with and without the barometric formula	14				
	5.3	Comparison of Newton's and uniform gravity	15				
6	The	optimal throwing angle	15				
7	$\mathbf{Sun}$	nmary of results	18				
A	$\mathbf{Stol}$	kes drag: solving the equation of motion	19				
в	But	cher tables	22				
Re	References						

## 1 Introduction

Simple projectile motion is one of the first physical phenomena that have been examined thoroughly in the history of physics. Galileo Galilei himself conducted several measurements and showed that generally the trajectory of thrown projectile is parabolic [1]. However, this holds for such projectiles that are affected by downward acting gravity only.

The main purpose of my project is to investigate and analyse projectile trajectories under different circumstances through numerical simulations. The aim of my work is to numerically reproduce Galileo's parabolic trajectory of the simplest case and then move further and introduce certain types of drag, height dependent air density or the curvature of Earth to the simulated system making it more and more realistic (hence the name of the project: "semi-realistic").

Personally, I have been fascinated by the elegance of the use of differential equations in physics since I first met these concepts. I have somewhat similar feelings towards most kinds of computer simulations so numerical differential equation solvers truly impress and interest me. Furthermore, I find projectile motion a topic that is a prime example of how different effects can be added to a model to make it more and more realistic.

## 2 Basic concepts and ideas

#### 2.1 The equation of motion

The problem of projectile motion can be generally formulated as an initial value problem regarding a second order ordinary differential equation (ODE). This ODE is none other than the Newton equation, which can be solved analytically for simple downward acting gravity. Taking into account the aforementioned effects finding the ODE's solution could become cumbersome (if even possible). That is when numerical methods become necessary.

The simplest Newton equation containing only the downwards gravitational term is the following:

$$m\ddot{\mathbf{r}}(t) = m\mathbf{g} \tag{1}$$

where m is the mass of the (now possibly point-like) projectile,  $\mathbf{r}(t)$  is the time dependent position vector and  $\mathbf{g} = (0, 0, -g)$  is the gravitational acceleration vector. Let the dot indicate the time derivative (hence double dot means second time derivative). The general solution of (1) can be obtained easily and can be written as

$$\mathbf{r}(t) = \mathbf{r}_0 + \mathbf{v}_0 t + \frac{1}{2} \mathbf{g} t^2 , \qquad (2)$$

where  $\mathbf{r}_0$  and  $\mathbf{v}_0$  constants are fixed by the initial value problem (meaning  $\mathbf{r}(0) = \mathbf{r}_0$  and  $\dot{\mathbf{r}}(0) = \mathbf{v}_0$ ). We can see that (2) gives the parabolic solution.

#### 2.2 Drag effects

For most cases air resistance is neglected. We will consider only symmetric projectiles because in such a case the force originating from air resistance always acts in the opposite direction of motion, meaning that velocity and the drag force vectors are parallel. At lower speeds this drag force is proportional to  $|\mathbf{v}| = v$  (Stokes drag) and at larger speed values to  $v^2$  (Newton drag).

Stokes drag is mostly true for spherical objects in a viscous fluid with small Reynolds number [2]. The formula for Stokes drag can be written as

$$\mathbf{F}_{\mathrm{S}} = -6\pi\eta R \mathbf{v} , \qquad (3)$$

where  $\eta$  is the dynamic viscosity and R is the radius of the spherical object. We can see that Stokes drag is quite a special case and several assumptions must be made when using it. However, the equations of motion can be solved for Stokes drag, thus the equation of motion is (rotating into the coordinate system with no motion along the y-axis)

$$\ddot{\mathbf{r}} = \mathbf{g} + \frac{1}{m} \mathbf{F}_{\mathrm{S}} \qquad \Longrightarrow \qquad \begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix} = \begin{pmatrix} -\zeta v_x \\ 0 \\ -g - \zeta v_z \end{pmatrix} , \qquad (4)$$

where  $\zeta = 6\pi\eta Rm^{-1}$ . We can see that these equations are not coupled so they can be solved individually (for detailed derivation see Appendix A.). The time dependency of the horizontal and vertical positions are

$$x(t) = \frac{v_{x,0}}{\zeta} \left( 1 - e^{-\zeta t} \right) + x_0 , \qquad (5)$$

$$z(t) = \frac{1}{\zeta} \left[ \left( v_{z,0} + \frac{g}{\zeta} \right) \left( 1 - e^{-\zeta t} \right) + gt \right] + z_0 .$$
(6)

For projectile motions in air on Earth the Newton drag is the more realistic, although more complicated situation, since it ruins the linearity of the equation. The formula of the Newton drag force is

$$\mathbf{F}_N = -\frac{1}{2}C\varrho A v \mathbf{v} , \qquad (7)$$

where C is the drag coefficient,  $\rho$  is the air density and A is the cross sectional area of the projectile [3]. The general case of the projectile motion with Newton drag cannot be solved analytically so suitable numerical integration methods must be used.

#### 2.3 Barometric formula

In the above expression (7)  $\rho$  density is mostly considered constant but some z height dependency can be added to our model. Using the hydrostatic equation (of motion) and the ideal gas model one can obtain the so-called barometric formula which is the following:

$$\varrho(z) = \varrho_0 \exp\left(-\frac{Mg}{RT}z\right),\tag{8}$$

where  $\rho_0$  is the air density at sea level, M is the molar mass of air, R is the universal gas constant and T is the temperature. Generally temperature can also be considered a height dependent quantity but we will neglect this effect.

#### 2.4 Curvature of Earth

In (1) we consider a flat Earth with uniform gravity but this is only an approximation. In our simulations we can use Newton's law of gravity to take the curvature of our planet into account:

$$\mathbf{F}_G = -\gamma \frac{mM_{\mathbf{\delta}}}{r^3} \mathbf{r} \,, \tag{9}$$

where  $\gamma$  is the gravitational constant,  $M_{\delta}$  is the mass of Earth and r is distance from the centre of Earth. During the comparison of uniform and Newtonian gravity I will use the approximate literary values of these constants:  $\gamma = 6.67408 \cdot 10^{-11} \text{ m}^3 \text{ kg}^{-1} \text{ s}^{-2}$ ,  $M_{\delta} = 5.9722 \cdot 10^{24} \text{ kg}$  and the radius of Earth will be  $R_{\delta} = 6.371 \cdot 10^6 \text{ m}.$ 

## 3 Numerical methods

One can find several different methods and techniques to solve ODEs numerically in various programming languages [4–6]. The basic idea behind numerical ODE solvers is to discretize the given continuous equation to finite steps with a fixed or adaptive step size while keeping numerical errors reasonably low. One such discretization is the Euler method with fixed step size. Our first order equation for y(t) can be written generally as

$$\dot{y}(t) = f(t, y(t)) \longrightarrow y_{n+1} = y_n + hf(t_n, y_n) \text{ and } t_{n+1} = t_n + h.$$
 (10)

where n indicates the number of the current step and h is the fixed step size. I am going to use the more trustworthy fourth order Runge–Kutta (RK4) and adaptive Cash–Karp (CK) methods (which is also part of the RK-family) and compare their accuracy and performance.

Both the RK4 and CK methods use a certain number of evaluations at different points of the f(t, y(t)) function. Generally an RK method with arbitrary order can be written in the following form:

$$y_{n+1} = y_n + \sum_{i=1}^{s} c_i k_i + \mathcal{O}(h^{\text{order}+1}) , \qquad (11)$$

where  $s \ge$  order and the  $k_i$  values are generated iteratively:

$$k_{1} = hf(t_{n}, y_{n}), \qquad (12)$$

$$k_{2} = hf(t_{n} + a_{2}h, y_{n} + b_{21}k_{1}), \qquad (12)$$

$$k_{3} = hf(t_{n} + a_{3}h, y_{n} + b_{31}k_{1} + b_{32}k_{2}), \qquad \vdots$$

$$k_{s} = hf(t_{n} + a_{s}h, y_{n} + b_{s,1}k_{1} + b_{s,2}k_{2} + \dots + b_{s,s-1}k_{s-1}).$$

The adaptive CK method uses a fifth order Runge–Kutta method with a fourth order embedded method to estimate error, such as

$$y_{n+1}^* = y_n + \sum_{i=1}^6 c_i^* k_i , \qquad (13)$$

so the error estimate is

$$\Delta = y_{n+1} - y_{n+1}^* = \sum_{i=1}^6 (c_i - c_i^*) k_i .$$
(14)

By setting some desired accuracy  $\Delta_0$  we can decide to keep the step or to throw it away and try again with an in some way modified step size (hence the name adaptive). The step size modification happens according to the following formula:

$$h' = \begin{cases} Sh \left| \frac{\Delta_0}{\Delta} \right|^{0.20} &, \text{ if } \Delta_0 < \Delta \\ Sh \left| \frac{\Delta_0}{\Delta} \right|^{0.25} &, \text{ if } \Delta_0 > \Delta \end{cases}$$

$$(15)$$

where S is some safety factor few percent less than unity. In Appendix B. the Butcher tables of RK4 and adaptive CK can be found. The implemented integration methods and the simulations themselves – written with the help of mainly [6] – can be found in [7] GitHub repository.

## 4 Comparison of analytical and numerical results

As mentioned above, one can solve analytically the simplest cases of neglected air resistance and Stokes drag. During the simulations the gravitational acceleration was taken to be  $g = 9.80665 \text{ ms}^{-2}$  and for the Stokes drag  $\zeta = 10 \text{ s}^{-1}$ . The value of  $\zeta$  is overestimated in order to have a visually more detectable effect for the drag; in reality  $\zeta$  has a magnitude of about  $10^{-4}$ .

#### 4.1 Results without air resistance

The simulated RK4 trajectories without air resistance plotted together with the analytical result are shown in Fig. 1. From the plots one can assume that all the used fixed h step sizes are feasible, though naturally the smaller the step size the smoother the result. Taking the difference of the analytical and numerical results into account one can find that it will be approximately  $\propto 10^{-5}$ . This is a fortunate case and partially a consequence of the symmetry of the parabola and the choice of the step size.

In the case of neglected air resistance the total energy  $E = m(v_x^2 + v_z^2)/2 - mgz$  of the projectile is a conserved quantity and this shall be true for the simulation data as well. The calculated energies for different step sizes are shown in Fig. 2. We can notice that the energy is indeed conserved with an acceptable error: it fluctuates close to the analytical result.



Figure 1: Simulated trajectories via RK4 method with different fixed h step sizes. Initial numerical values were  $x_0 = 0, z_0 = 0, v_{x,0} = 10 \text{ ms}^{-1}, v_{z,0} = 10 \text{ ms}^{-1}$ .



Figure 2: Calculated total energy values from the RK4 simulation results with different fixed h step sizes. Initial numerical values were  $x_0 = 0, z_0 = 0, v_{x,0} = 10 \text{ ms}^{-1}, v_{z,0} = 10 \text{ ms}^{-1}$ .

I have repeated the simulations with the adaptive CK method using different  $\varepsilon$  error parameters; the results are shown in Fig. 3. It is clear that with a higher allowed error rate the quality of the resulting trajectory can deteriorate drastically over time.



Figure 3: Simulated trajectories via adaptive CK method with different  $\varepsilon$  error parameters. Initial numerical values were  $x_0 = 0, z_0 = 0, v_{x,0} = 10 \text{ ms}^{-1}, v_{z,0} = 10 \text{ ms}^{-1}$  and the starting step size was h = 0.01 m.

The changes in the step size during the runs are displayed in Fig. 4. Looking at the plots we can certify that the implemented adaptive method works convincingly, since the step size shortens when the simulation reaches the top of the parabola – here the function "curves" more, hence the function changes faster and shorter steps are indeed necessary. One can also notice that the smaller the error parameter the smaller the step size (just as we need it to be).

Naturally, the real curve determined through the simulation is not the trajectory, but the time dependency of the spatial coordinates x(t) and z(t). This does not contradict the argument above about the connection of the "curving" of the parabola and the changing step size, since it is known that z(t) is a quadratic function of t and x(t) is directly proportional to t.



Figure 4: The changes of the step size during the simulations using the adaptive CK method with different  $\varepsilon$  error parameters. Initial numerical values were  $x_0 = 0, z_0 = 0, v_{x,0} = 10 \text{ ms}^{-1}, v_{z,0} = 10 \text{ ms}^{-1}$  and the starting step size was h = 0.01 m.

Energy conservation can be tested in the case of adaptive CK method as well (see results in Fig. 5.) and it is not that unequivocal, but with a sufficiently small error parameter one can deem it conserved. Taking the analysis of RK4 and adaptive CK into account one might conclude that RK4 still has the upper hand, but we shall see that the situation is not that simple at all.



Figure 5: Calculated total energy values from the simulation results via CK method with different  $\varepsilon$  error parameters. Initial numerical values were  $x_0 = 0, z_0 = 0, v_{x,0} = 10 \text{ ms}^{-1}, v_{z,0} = 10 \text{ ms}^{-1}$  and the starting step size was h = 0.01 m.

#### 4.2 Results with Stokes drag

I repeated the simulations including Stokes drag from expression (3); the plotted trajectories created with the RK4 method alongside with the analytical results are shown in Fig. 6. The drag effect is unmistakable and we can see that the RK4 method loses accuracy, especially in the  $h = 10^{-1}$  m case: the maximum difference from the analytical solution is nearly  $\propto 10^{-2}$  which is several orders higher than it was in the case of the symmetrical parabola. The energy of the system is not a good indicator of accuracy in this scenario since in a dissipative system the total energy is not conserved.



Figure 6: Simulated trajectories with Stokes drag via RK4 method with different fixed h step sizes. Initial numerical values were  $x_0 = 0, z_0 = 0, v_{x,0} = 10 \text{ ms}^{-1}, v_{z,0} = 10 \text{ ms}^{-1}$ .

I performed the simulations again with the adaptive CK method. The obtained trajectories for different error parameters are presented in Fig. 7. and we can notice that these results are one step ahead of the RK4 method with  $h = 10^{-1}$ . The determined z(t) and x(t) functions are not a parabola and a simple linear expression this time but something much more complicated, as we saw it in (5), and as it is shown in Fig. 8. If we also take a look at Fig. 9. showing the time evolution of the step lengths, we can conclude that the adaptive correction of the step lengths managed to overcome the sharper curves of the z(t) function.



Figure 7: Simulated trajectories with Stokes drag via adaptive CK method with different error parameters. Initial numerical values were  $x_0 = 0, z_0 = 0, v_{x,0} = 10 \text{ ms}^{-1}, v_{z,0} = 10 \text{ ms}^{-1}$  and the starting step size was h = 0.01 m.



Figure 8: The analytical z(t) and x(t) functions with uniform gravity and Stokes drag included.



Figure 9: The changes of the step size during the simulations including Stokes drag using the adaptive CK method with different  $\varepsilon$  error parameters. Initial numerical values were  $x_0 = 0, z_0 = 0$  $v_{x,0} = 10 \text{ ms}^{-1}, v_{z,0} = 10 \text{ ms}^{-1}$  and the starting step size was h = 0.01 m.

According to the analysis of RK4 and the adaptive CK methods one can state that both methods are suitable for most cases. With solutions that contain slowly changing functions we shall use RK4 with a proper fixed step size, since it is still one of the most reliable differential equation solver methods and its run time is much more predictable than that of the adaptive integrators. However, as we have seen, the adaptive CK method has its advantages as well, regarding solutions that contain both slowly and fast changing sections<sup>1</sup>. We can expect more of such solution functions in the following, hence I will use the adaptive CK method with a small enough  $\varepsilon$  error parameter to handle these possible problems.

### 5 Numerical analysis of semi-realistic effects

#### 5.1 Comparison of Stokes and Newton drag

Through a number of simulations I tried to qualitatively determine the main characteristic differences between applying Stokes or Newton drag in the analysed system. We have already defined  $\zeta = 6\pi\eta Rm^{-1}$ for Stokes drag, thus we can do the same for Newton drag as well:  $\zeta' = \frac{1}{2}C\varrho Am^{-1}$ .

<sup>&</sup>lt;sup>1</sup>Remark: In fact we could have come to this statement without applying Stokes drag. The same argument holds for sharper parabolas as well.

This way the equations of motion including Newton drag will be

$$\ddot{\mathbf{r}} = -\zeta' |\dot{\mathbf{r}}| \dot{\mathbf{r}} \implies \begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix} = \begin{pmatrix} -\zeta' v_x \sqrt{v_x^2 + v_z^2} \\ 0 \\ -g - \zeta' v_z \sqrt{v_x^2 + v_z^2} \end{pmatrix}.$$
(16)

It is a critical question how to estimate the ratio of  $\zeta$  and  $\zeta'$ . The exact and rigorous calculation is out of the scope of this project, thus I will apply the following *back of the envelope* estimation. Let us consider a spherical projectile in air at standard temperature with mass m = 1 kg. The cross sectional area will be  $A = R^2 \pi$ . The  $\eta$  dynamical viscosity of air is  $\eta \approx 1.803 \cdot 10^{-5}$  kg (ms)<sup>-1</sup> [8], the drag coefficient of a solid sphere is  $C \approx 0.5$  [9] and the density of air can be taken as  $\rho \approx 1.2$  kg m<sup>-3</sup>. Taking all these values into account we can determine a case (which is just a bit unrealistic) when the numerical values of  $\zeta$  and  $\zeta'$  can be considered equal (such a case would need a radius of  $R \approx 0.1$  mm) and we will implement this case into the simulations. The simulation results for the trajectories are shown in Fig. 10.



Figure 10: Different trajectories with either Stokes or Newton drag at different  $v_{x,0} = v_{z,0} = v_0$  initial velocities. The starting step size was h = 0.01 m, the error parameter was  $\varepsilon = 10^{-4}$  and the numerical value of  $\zeta$  and  $\zeta'$  was 1 in both scenarios.

One can easily notice that for lower speeds the effect of Newton drag lowers as well compared to the Stokes drag case. As the speed increases the Newton drag also has more impact on the trajectories and will overcome the Stokes drag in strength. The effect of raising and lowering the magnitude of  $\zeta$  and  $\zeta'$  is not in our field of interest now – possibly the drag effect would simply intensify or weaken – hence I did not perform such simulations.

#### 5.2 Newton drag with and without the barometric formula

I incorporated the (8) barometric formula into the Newton drag expression in order to investigate the case of height dependent air density. The coefficient in the exponent of the barometric formula is  $\lambda = -Mg(RT)^{-1} \approx -0.00012 \text{ m}^{-1}$ , which is a very small quantity, hence I used a few order of magnitudes higher  $\lambda$  (= 0.01 m<sup>-1</sup>) value to determine the quantitative shape of the trajectory. The  $\rho_0$  density parameter was 1.2250 kg m<sup>-3</sup>. The results of the simulations are presented in Fig. 11.



Figure 11: Different trajectories with and without height dependent density at different  $v_0 = v_{x,0} = v_{y,0}$ initial velocities. The starting step size was h = 0.01 m, the error parameter was  $\varepsilon = 10^{-5}$ .

We can see that with the exponentially decreasing density the maximum z and x coordinates increase as well. With further simulations we can determine the velocity dependence of this increase shown in Fig. 12. The behaviour of the curves seems to be logarithmic (at least in leading order) in the horizontal case, though a more precise study of the problem would be needed to obtain a more detailed result.



Figure 12: The initial velocity dependence of differences of x coordinates between the cases of constant and barometric air density.

#### 5.3 Comparison of Newton's and uniform gravity

Up to now we have only considered a flat Earth with uniform gravity and as it was mentioned previously we can use Newton's law of gravity from equation (9) instead of the rather simple F = mg gravitational force. During the performed simulations I compared the realised trajectories in both cases with different initial velocities, the results are shown in Fig. 13.



Figure 13: Trajectory results using Newtonian and uniform gravitational fields with different  $v_0 = v_{x,0} = v_{z,0}$  initial velocities. The starting step size was h = 0.01 m and the error parameter  $\varepsilon = 10^{-4}$ .

We can see the quadratic and elliptical trajectories side by side and one can notice that the higher the initial velocity the greater the difference between them. This way we could demonstrate that the uniform gravitational field is actually an appropriate approximation for shorter length scales. Naturally, for greater lengths the Newtonian theory is required and the problem of projectile motion will branch out to the well-known Kepler problem of planetary motion.

## 6 The optimal throwing angle

Finally, I have tested and verified the well known  $45^{\circ}$  optimal throwing angle for maximal horizontal displacement in the simple case of uniform gravity. In order to set the correct  $v_{x,0}$  and  $v_{z,0}$  initial velocities I used the analogy of the known formulas of polar coordinates

$$v_{x,0} = v_0 \cos \varphi \quad \text{and} \quad v_{z,0} = v_0 \sin \varphi ,$$

$$\tag{17}$$

where  $v_0$  will be the magnitude of the velocity and  $\varphi$  the throwing angle throughout the simulations. For the simplest of cases I performed numerous simulations at fixed  $v_0$  and different  $\varphi$  values and determined the endpoint of the trajectories. The results are presented in Fig. 14. It is self-evident that the optimal throwing angle is still at  $\varphi = 45^{\circ}$ .



Figure 14: Horizontal trajectory endpoints at different  $\varphi$  throwing angles. The curve is scaled with the maximum of obtained x coordinates and converted to percentage values. The magnitude of the initial velocity was  $v_0 = 1 \text{ ms}^{-1}$ , the starting step size was h = 0.01 and the error parameter was  $\varepsilon = 10^{-5}$ .

Performing further simulations we can determine the optimal throwing angles for uniform gravity paired with either Stokes or Newton drag. The drag effects are characterised by the previously used  $\zeta$  and  $\zeta'$ parameters, hence I ran simulations at different angles and with different  $\zeta, \zeta'$  values while keeping the magnitude of the initial velocity fixed. Because of this, I demonstrate the results using heat maps showing the horizontal components of endpoints of trajectories at certain ( $\varphi, \zeta$  or  $\zeta'$ ) values. Furthermore, I determined the optimal throwing angle at each  $\zeta$  or  $\zeta'$ ; these angles are highlighted by a white curve on the heat maps.

The heat map of Stokes drag is shown in Fig. 15. and the map calculated for Newton drag can be viewed in Fig. 16. One can notice that in both cases the stronger the drag effect (meaning the greater the value of  $\zeta$  or  $\zeta'$ ) the more acute throwing angle will be optimal to achieve the maximal displacement. Taking the value of the parameters to be zero we get back to the simple case of uniform gravity with a  $45^{\circ}$  optimal angle.



Figure 15: Heat map of trajectory horizontal endpoints at different  $\varphi$  angles and  $\zeta$  Stokes drag parameters. The values are scaled with the maximum of obtained x coordinates and converted to percentage values. The magnitude of the initial velocity was  $v_0 = 1 \text{ ms}^{-1}$ , the starting step size was h = 0.01 and the error parameter was  $\varepsilon = 10^{-5}$ .



Figure 16: Heat map of trajectory horizontal endpoints at different  $\varphi$  angles and  $\zeta'$  Newton drag parameters. The values are scaled with the maximum of obtained x coordinates and converted to percentage values. The magnitude of the initial velocity was  $v_0 = 1 \text{ ms}^{-1}$ , the starting step size was h = 0.01 and the error parameter was  $\varepsilon = 10^{-5}$ .

## 7 Summary of results

During the analysis of semi-realistic projectile motion I investigated and compared some of the wellknown complicating factors of the problem. Through the comparison of Stokes and Newton drag I managed to demonstrate that these effects dominate different velocity ranges. After that I performed the comparison of Newton drag effects with and without height dependent air density. The obtained results imply that the implementation of the barometric formula results in longer trajectories. I also analysed the transition between the approximation of uniform gravity and the more general Newton's law of gravity and concluded that for most of the practical problems on Earth the application of uniform gravity is suitable.

Finally, I performed numerous simulations to numerically check the 45° optimal throwing angle for the case of F = mg. Hereupon I introduced Stokes and Newton drag into the system again and determined the optimal throwing angle for different  $\zeta$  and  $\zeta'$  characteristic parameters and made the general qualitative conclusion that for Stokes and Newton drag the greater the  $\zeta$  or  $\zeta'$  parameters the more acute throwing angle is required to achieve maximal horizontal range.

## A Stokes drag: solving the equation of motion

As we have mentioned, the equations of motion in (3) can be solved analytically (and by hand quite simply). Let us start with the simpler one to calculate x(t); the equation is a second order homogeneous ordinary differential equation:

$$\ddot{x} = -\zeta \dot{x} . \tag{18}$$

Let us introduce a new function  $v(t) = \dot{x}$ , hence

$$\dot{v} = -\zeta v \tag{19}$$

and from now on we only have to deal with a first order equation. The solution can be obtained by the separation of variables:

$$\dot{v} = \frac{dv}{dt} = -\zeta v \qquad \Longrightarrow \qquad \int dv \ \frac{1}{v} = -\zeta \int dt \ .$$
 (20)

The solution of the integrals are simple, thus

$$v(t) = c e^{-\zeta t} , \qquad (21)$$

where c is some integration constant. Now we integrate again to obtain x(t):

$$\dot{x} = \frac{dx}{dt} = ce^{-\zeta t} \qquad \Longrightarrow \qquad \int dx = c \int dt \ e^{-\zeta t} \ .$$
 (22)

With  $c_1 = -c$  and adding  $c_2$  to the expression as another integration constant we get

$$x(t) = \frac{c_1 e^{-\zeta t}}{\zeta} + c_2 .$$
(23)

We want  $x(0) = x_0$  and  $\dot{x}(0) = v_{x,0}$  initial conditions to be true, thus

$$\dot{x}(0) = -c_1 = v_{x,0} \tag{24}$$

and from this

$$x(0) = -\frac{v_{x,0}}{\zeta} + c_2 = x_0 \implies c_2 = x_0 + \frac{v_{x,0}}{\zeta}.$$
 (25)

Finally we get the first line of (5):

$$x(t) = \frac{v_{x,0}}{\zeta} \left( 1 - e^{-\zeta t} \right) + x_0 .$$
(26)

The second equation which is for z(t) is slightly more complicated. It is a second order ordinary inhomogeneous differential equation:

$$\ddot{z} = -\zeta \dot{z} - g . \tag{27}$$

Since it is an inhomogeneous equation its complementary solution can be given as the sum of the complementary solution of the homogeneous part and a particular solution of the inhomogeneous equation:

$$z(t) = z_h(t) + z_p(t)$$
 (28)

One can notice that the solution of the homogeneous part is already determined since it is the same as x(t) (without fitting to initial values, of course). The inhomogeneous part is just the constant g so our *ansatz* will be  $z_p(t) = At$  and from here we continue with the method of undetermined coefficients. Let us substitute  $z_p(t)$  back into the original equation:

$$\ddot{z}_p = -\zeta \dot{z}_p - g \implies A = -\frac{g}{\zeta}$$
 (29)

From this the complementary solution is

$$z(t) = \frac{c_1 e^{-\zeta t}}{\zeta} - \frac{gt}{\zeta} + c_2 .$$
 (30)

Naturally, we have initial conditions now as well:  $z(0) = z_0$  and  $\dot{z}(0) = v_{z,0}$ , thus

$$\dot{z}(0) = -c_1 - \frac{g}{\zeta} = v_{z,0} \qquad \Longrightarrow \qquad c_1 = -\left(v_{z,0} + \frac{g}{\zeta}\right). \tag{31}$$

Furthermore,

$$z(0) = -\frac{1}{\zeta} \left( v_{z,0} + \frac{g}{\zeta} \right) + c_2 = z_0 \qquad \Longrightarrow \qquad c_2 = z_0 + \frac{1}{\zeta} \left( v_{z,0} + \frac{g}{\zeta} \right). \tag{32}$$

By collecting and simplifying the terms one can get the second line of (5):

$$z(t) = \frac{1}{\zeta} \left[ \left( v_{z,0} + \frac{g}{\zeta} \right) \left( 1 - e^{-\zeta t} \right) + gt \right] + z_0 .$$

$$(33)$$

This way we derived both of the expressions for x(t) and z(t).

## **B** Butcher tables

i	$a_i$		$c_i$			
1	0					$\frac{1}{6}$
2	$\frac{1}{2}$	$\frac{1}{2}$				$\frac{1}{3}$
3	$\frac{1}{2}$	0	$\frac{1}{2}$			$\frac{1}{3}$
4	1	0	0	1		$\frac{1}{6}$
	j	1	2	3	4	

Table 1: Butcher table of RK4 method.

Table 2: Butcher table of adaptive CK method.

i	$a_i$			$b_{ij}$			$c_i$	$c_i^*$
1	0						$\frac{37}{378}$	$\frac{2825}{27648}$
2	$\frac{1}{5}$	$\frac{1}{5}$					0	0
3	$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$				$\frac{250}{621}$	$\frac{18575}{48384}$
4	$\frac{3}{5}$	$\frac{3}{10}$	$-\frac{9}{10}$	$\frac{6}{5}$			$\frac{125}{594}$	$\frac{13525}{55296}$
5	1	$-\frac{11}{54}$	$\frac{5}{2}$	$-\frac{70}{26}$	$\frac{35}{27}$		0	$\frac{277}{14336}$
6	$\frac{7}{8}$	$\tfrac{1631}{55296}$	$\tfrac{175}{512}$	$\tfrac{575}{13824}$	$\frac{44275}{110592}$	$\tfrac{253}{4096}$	$\frac{512}{1771}$	$\frac{1}{4}$
j		1	2	3	4	5		

## References

- [1] https://en.wikipedia.org/wiki/Projectile\_motion
- [2] https://en.wikipedia.org/wiki/Stokes%27\_law
- [3] https://en.wikipedia.org/wiki/Drag\_equation
- [4] R. H. Landau, M. J. Paez, C. Bordeianu. A Survey of Computational Physics introductory computational science. Princeton University Press., 2008.
- [5] https://stegerjozsef.web.elte.hu/teaching/szamszim/index.php
- [6] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery. Numerical recipes, 3rd Edition. Cambridge University Press., 2007.
- [7] https://github.com/pd0220/ComputerSimulationsInPhysicsELTE2020
- [8] https://www.engineeringtoolbox.com/air-absolute-kinematic-viscosity-d\_601.html
- [9] https://www.engineeringtoolbox.com/drag-coefficient-d\_627.html