Tecnología e Innovación

| Artículo de revisión | Review papel | Artigo de revisão |

Dynamic analysis of rounded projectiles: Software solution development

Fecha de recibido: 28 de julio 2023	Fecha de aprobado: 17 de abril 2024
Reception date: July 28, 2023	Approval date: April 17, 2024
Data de recebimento: 28 de julho de 2023	Data de aprovação: 17 de abril de 2024

Juan David Roa Camargo
https://orcid.org/0009-0004-0756-464X
roa_juan@javeriana.edu.co
Estudiante de Ingeniería Mecánica
Investigador – Pontificia Universidad Javeriana, Colombia
Rol del investigador: teórico y escritura
Mechanical Engineering Student
Rresearcher – Pontificia Universidad Javeriana, Colombia
Researcher's role: theoretical and writing
Engenharia Mecânica Estudiante
Investigador – Pontificia Universidad Javeriana, Colômbia
Papel do investigador: teórico e escritor

Juan Pablo Ruiz Puente
https://orcid.org/0009-0005-0995-5836
juanruizp@javeriana.edu.co
Estudiante de Ingeniería Mecánica
Investigador – Pontificia Universidad Javeriana, Colombia
Rol del investigador: teórico y escritura
Mechanical Engineering Student
Rresearcher – Pontificia Universidad Javeriana, Colombia
Researcher's role: theoretical and writing
Engenharia Mecânica Estudiante
Investigador – Pontificia Universidad Javeriana, Colômbia
Função do investigador: Teórico e escritor

Camilo Andrés Bayona Roa
https://orcid.org/0000-0001-9508-1779
ca_bayonar@javeriana.edu.co
Doctor en Ingeniería
Docente e investigador – Pontificia Universidad Javeriana, Colombia
Rol del investigador: teórico y escritura
Grupo de Investigación: BASPI
Dh. D. in Engineering
Professor and researcher – Pontificia Universidad Javeriana, Colombia
Researcher's role: theorist and writer
Research Group: BASPI
Doutoramento em Engenharia
Docente e investigador – Pontificia Universidad Javeriana, Colômbia
Papel do investigador: teórico e escritor
Grupo de investigação: BASPI

Cómo citar este artículo: Roa Camargo, J. D., Ruiz Puente, J. P. & Bayona Roa, C. A. (2024). Dynamic analysis of rounded projectiles: Software solution development. *Ciencia y Poder Aéreo*, *19*(2), 46-57. https://doi.org/10.18667/cienciaypoderaereo.822

Dynamic analysis of rounded projectiles: Software solution development

Abstract: The study of object motion has intrigued scholars for centuries, yet grasping the underlying physics and mathematics has remained a challenge. However, recent advancements in computational methods and the utilization of mathematical models from the 18th century have enabled a profound understanding of motion and accurate approximations of real-life object movements. By harnessing tools such as MATLAB and engineering analytical methods, we can create applications that simulate the motion of spherical projectiles. This provides valuable insights into real-world object motion and the associated forces and physics. Dynamic analysis, encompassing both kinematics and kinetics, allows for a detailed exploration of motion dynamics. Inspired by the concept of an 'Olympic goal,' (Clayfield, n. d.), our developed application allows users to visualize the impact of aerodynamic forces on objects. It demonstrates concepts like drag, lift, and the 'Magnus effect,' offering initial insights into object motion in fluid environments (Mody, 2015). This understanding acts as a foundation for modeling more intricate systems, including airplanes, rockets, and aerospace components. Notably, the application's graphical representations of essential modeling elements provide a significant advantage. Furthermore, by emphasizing the motivations behind these phenomena, the application fosters curiosity and encourages users to delve deeper into these captivating events.

Keywords: Drag; kinematics; kinetics; lift; Magnus effect; mathematical modelling.

Análisis dinámico de proyectiles redondeados: desarrollo de soluciones con *software*

Resumen: El estudio del movimiento de los objetos ha intrigado a los académicos durante siglos, pero comprender la física y las matemáticas subyacentes ha sido un desafío. Los avances recientes en métodos computacionales y el uso de modelos matemáticos del siglo XVIII permitieron comprender el movimiento y aproximaciones precisas de los movimientos reales de los objetos. Al aprovechar herramientas como MATLAB y métodos analíticos de ingeniería, podemos crear aplicaciones que simulen el movimiento de proyectiles esféricos. Esto brinda valiosos conocimientos sobre el movimiento real de los objetos y las fuerzas y la física asociadas. El análisis dinámico, que abarca tanto la cinemática como la cinética, permite una exploración detallada de la dinámica del movimiento. Inspirada en el concepto de un 'gol olímpico', nuestra aplicación desarrollada permite a los usuarios visualizar el impacto de las fuerzas aerodinámicas en los objetos. Demuestra conceptos como la resistencia, la sustentación y el 'efecto Magnus', ofreciendo ideas iniciales sobre el movimiento de los objetos en entornos fluidos. Esta comprensión sienta las bases para modelar sistemas más complejos, como aviones, cohetes y componentes aeroespaciales. Es importante destacar que las representaciones gráficas de los elementos esenciales de modelado de la aplicación ofrecen una ventaja significativa. Además, al enfatizar las motivaciones detrás de estos fenómenos, la aplicación despierta la curiosidad y anima a los usuarios a profundizar en estos eventos cautivadores.

Palabras clave: arrastre; cinemática; cinética; sustentación; efecto Magnus; modelización matemática.

Análise dinâmica de projéteis arredondados: Desenvolvimento de solução de *software*

Resumo: O estudo do movimento de objetos tem intrigado estudiosos há séculos, no entanto, compreender a física e matemática subjacentes tem sido um desafio. No entanto, os avanços recentes em métodos computacionais e a utilização de modelos matemáticos do século XVIII permitiram uma compreensão profunda do movimento e aproximações precisas dos movimentos reais dos objetos. Ao utilizar ferramentas como MATLAB e métodos analíticos da engenharia, podemos criar aplicações que simulam o movimento de projéteis esféricos. Isso proporciona insights valiosos sobre o movimento de objetos no mundo real e as forças e a física associadas. A análise dinâmica, que abrange tanto a cinemática quanto a cinética, permite uma exploração detalhada da dinâmica do movimento. Inspirada pelo conceito de um 'gol olímpico', nossa aplicação desenvolvida permite que os usuários visualizem o impacto das forças aerodinâmicas nos objetos. Ela demonstra conceitos como arrasto, sustentação e o 'efeito Magnus', oferecendo insights iniciais sobre o movimento de objetos em ambientes fluidos. Essa compreensão serve como base para modelar sistemas mais complexos, incluindo aviões, foguetes e componentes aeroespaciais. É importante destacar que as representações gráficas dos elementos essenciais de modelagem da aplicação proporcionam uma vantagem significativa. Além disso, ao enfatizar as motivações por trás desses fenômenos, a aplicação desperta a curiosidade e incentiva os usuários a se aprofundarem nesses eventos cativantes.

Palavras-chave: Arrasto; cinemática; cinética; elevação; efeito Magnus; modelagem matemática.

Introduction

Projectiles have played a significant role throughout human history, serving purposes ranging from hunting and defense to space exploration. The study of projectile motion is crucial in achieving these objectives, with its analysis dating back several centuries. Beginning with Aristotle in the classical period of Greece during the 4th century, the study of motion evolved into more practical methodologies, notably exemplified by Isaac Newton in the 17th century (Dixit *et al*., 2017). This branch of study, known as dynamics, encompasses two main areas: kinematics, which focuses on movement without considering forces, and kinetics, which investigates the forces responsible for object motion. Understanding these phenomena has been of great interest, particularly due to the inherent challenges in modeling such systems. Complex interactions with fluids, such as lift and drag forces, significantly influence an object's motion through the air, enabling the use of airplanes to overcome gravity.

Gaining a comprehensive understanding of how objects respond to these forces allows us to decipher their motion. Beginning with the modeling of a sphere or rounded projectile, which may appear simple at first glance but progressively becomes more challenging as we consider variables and equations that change over time, we encounter additional elements that influence their behavior. For instance, rotational effects, such as the 'Magnus effect,' explain why rotating objects in fluids alter their direction due to a force orthogonal to both the object's velocity and rotation vectors (Kray *et al*., 2013). Parameters like the Reynolds number indicate the level of fluid turbulence (Blevins, 1985), while the drag and lift coefficients vary over time based on changes in velocity. Given the complex nature of these forces, which are also instrumental in understanding accelerations, we employ computational tools such as MATLAB.

Utilizing MATLAB's app designing tool, MATLAB App Designer, we can develop a user-friendly and intuitive software that predicts the movement of rounded projectiles using computational and numerical methods. This interactive software enables users to manipulate variables and observe the progression of forces, accelerations, velocities, positions, and other essential data required to model the projectile's trajectory. The resulting trajectory is visualized in a three-dimensional plane, providing a comprehensive and vivid understanding of the projectile's motion, enriching the user's conceptual grasp.

The primary objective of this work extends beyond the mere development of this tool; it aims to communicate the underlying computational approach and its implementation to the academic community. By doing so, we seek to not only showcase the versatility and power of MATLAB in solving complex mechanical problems, but also to offer this software as a valuable educational resource. We envision its application in academic settings to facilitate a deeper understanding of projectile motion, offering a practical teaching-learning tool that bridges theoretical knowledge with real-world application. Furthermore, this initiative opens the door for its adaptation and utilization in a broader range of mechanical problems, encouraging exploration, innovation, and a hands-on approach to learning in the fields of physics and engineering. Through this dissemination, we aim to inspire further research, collaboration, and development of educational tools that leverage computational methods for enhanced understanding and innovation in various scientific domains.

Objectives

Main objective: To construct a sophisticated yet user-friendly software that accurately simulates the trajectory of a spherical projectile. The core aim is to utilize vector mechanics to create a computational model that not only reflects real-life movements but also demonstrates the application of fundamental physical laws in a way that resonates with real-world observations.

Secondary objectives:

- demonstrate the applicability of the computational model by simulating a real-life scenario, integrating authentic values and measurements derived from actual events.

- To design an intuitive interface within the software that enables users to effortlessly explore and manipulate key variables influencing projectile motion, such as forces, accelerations, and velocities, thereby fostering a deeper understanding of the dynamics at play.
- To communicate the computational approach to the academic community, such that it can be exploited in other mechanic problems and teaching-learning opportunities.

Methodology

We will formulate first the model in its most illustrative expression, later on we will continue with the development of the mathematical modeling, finally our application or solution will be held by the explanation of how the code works, and with this we will give a final answer to our problem, which — as said before — it is based on the understanding of how motion works on projectiles with a software solution approach.

Understanding the Problem

Grasping the principles of mathematics and physics in practice is notably challenging. This challenge intensifies when we attempt to map these theoretical concepts into real-world scenarios. A key issue we face is the translation of these theories into applications that are intuitive and relatable. In our context, the movement of spherical projectiles offers a prime example of this problem. We will examine the 'Olympic Goal' in soccer to highlight the complexities involved. This scenario encapsulates the pivotal characteristics of projectile motion, demonstrating the intricate interplay of forces that govern the flight of a rounded projectile. The problem lies in distilling these multifaceted principles into a format that is easily understood and visualized, providing a clear model that mirrors the real behavior of round projectiles in a comprehensible manner.

Physical and Mathematical Modeling

Part I – Space

Before delving into further detail, it is essential to establish a frame of reference. In physics, the choice of reference frame is arbitrary and largely dependent on the observer's perspective; there is no singular 'ideal' frame of reference. For our purposes, we have selected a reference system that provides the most clarity and relevance for interpreting our specific problem, we will be using the following:

- z will be used for height
- y will be used for depth
- x will be used for length

In our soccer field, this can be visualized as given in Figure 1.

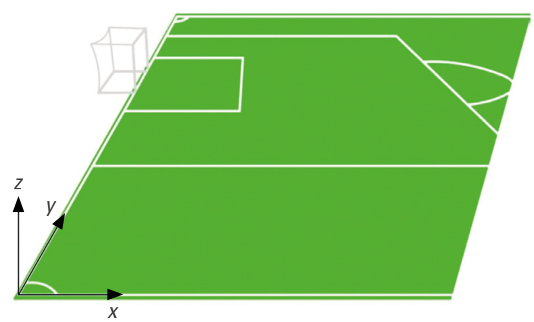

Figure 1. Illustration of soccer field with system of reference
Source: Own elaboration.

Part II – Dynamics Approach

Firstly, to describe the dynamics of a spherical projectile, we used the Newton's Dynamics Principle (Meriam & Kraige, 2012), which stablishes that all the forces acting in the object (projectile) of mass m are proportional to its acceleration a. This is:

$$\sum \boldsymbol{F} = m\boldsymbol{a} \tag{1}$$

For convenience, bold notation is used for vectors and tensors in the present work. For a spherical projectile, if the external disturbances caused by its

Escuela de Postgrados de la **Fuerza Aérea Colombiana** movement through a fluid are considered, then the following forces act on the sphere: its weight $\boldsymbol{W}$, the buoyant force $\boldsymbol{F_B}$, the Magnus force $\boldsymbol{F_M}$ (being a sort of lift) and the drag force $\boldsymbol{F_D}$. Hence:

$$\boldsymbol{F_D} + \boldsymbol{F_M} + \boldsymbol{F_B} + \boldsymbol{W} = m\boldsymbol{a} \quad (2)$$

Furthermore, to properly describe how these forces act on the sphere, we must compute the magnitude of each force. For this reason, a Cartesian reference system is used as follows:

Table 1.
General description of the dynamic equations

On x-axis

$$F_{Dx} + F_{Mx} = ma_x \quad (3)$$

On y-axis

$$F_{Dy} + F_{My} = ma_y \quad (4)$$

On z-axis

$$F_{Dz} + F_{Mz} + F_B + W = ma_y \quad (5)$$

Once our dynamic equations have been defined, we identify the parameters and variables of the problem. In this case, the mass of the projectile is known, as well as the forces at each instant of the flight. Our only unknown is the acceleration, which describes the cinematics of the projectile (Cook, 2007). The magnitude of the drag force is defined for each -axis as follows:

$$F_{Di} = -C_D A \frac{\rho V_i^{\,2}}{2} [N] \quad (6)$$

where:

C_D: Drag coefficient
A: Cross-sectional area $[m^2]$
ρ: Density of the fluid $\left[\frac{kg}{m^3}\right]$
V_i: Magnitude of the projectile's relative velocity to the fluid in the -axis $\left[\frac{m}{s^2}\right]$

Here, the drag force is *negative* as it opposes to the velocity of the object (contrary to the direction of movement) and hence, it varies for each axis with the respective velocity (Sarafian, 2015). Also, the drag coefficient is given by the following non-linear approximated function (Figure 2) of the Reynolds number $\Re$, which is defined as follows:

$$\Re = \frac{VL}{v} \quad (7)$$

where:

V: Magnitude of the projectile's velocity $\left[\frac{m}{s}\right]$
L: Characteristic length of the object (in this case, the diameter) $[m]$
v: Kinematic viscosity of the fluid $\left[\frac{m^2}{s}\right]$

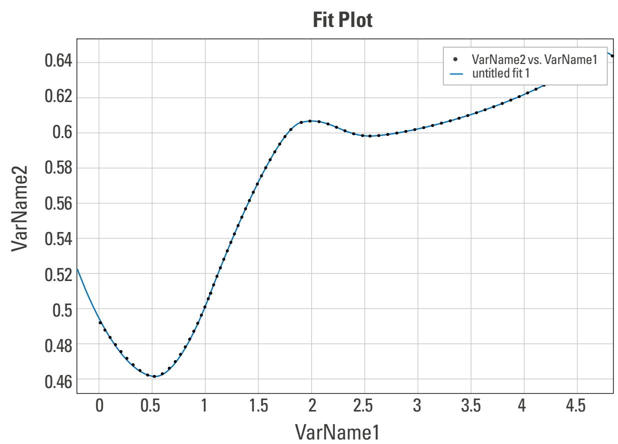

Figure 2. Non-linear approximation of the drag coefficient for a sphere in terms of the Reynolds Number[1]. The resulting approximation is: $C_D = 0.5195 * sin(2.718e - 6\Re + 0.1875) + 0.2265 * sin$
Source: Own elaboration.

Meanwhile, the Magnus force on each axis depends on the lift coefficient, the angular velocity $\boldsymbol{\omega}$ and the velocity of the projectile $\boldsymbol{v}$. Its direction is given by $\boldsymbol{\omega} \times \boldsymbol{v}$ (Robinson y Robinson, 2013), so that for each axis the magnitude is defined as:

$$F_{Mx} = C_{Lx} A \frac{\rho v_y^{\,2}}{2}, F_{My} = C_{Ly} A \frac{\rho v_z^{\,2}}{2}, F_{Mz} = C_{Lz} A \frac{\rho v_x^{\,2}}{2} [N] \quad (8)$$

1 This was taken from Cengel & Cimbala (2010) and numerically interpolated with https://apps.automeris.io/wpd/.

Here, the lift coefficient, similarly to the drag coefficient, is also given by a non-linear approximated function (Figure 3), but in this case it depends on a *rotation ratio* defined for each axis as:

$$Ratiox = \frac{\omega_k L}{2v_y}, Ratioy = \frac{\omega_y L}{2v_x}, Ratioz = \frac{\omega_x L}{2v_z} \quad (9)$$

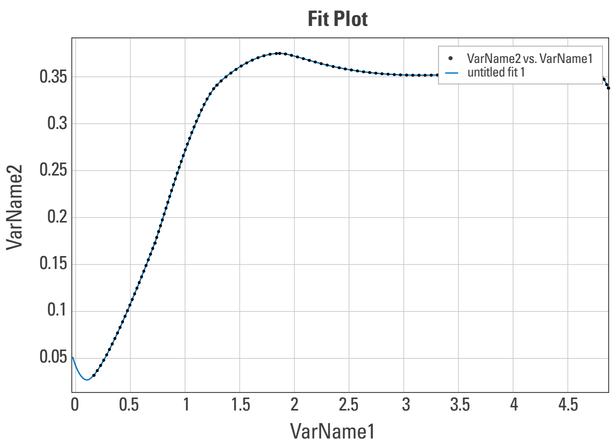

Figure 3. Non-linear approximation of the lift coefficient for a sphere in terms of the rotation ratio[2]. For example, the C_{Lx} is $C_{Lx} = 0.2944 + (-0.1077 * cos(Ratiox * w1)) + (-0.03208 * sin(Ratiox * w1)) + \cdots$

Source: Own elaboration.

The buoyant force is defined by the Archimedes Principle as follows:

$$\boldsymbol{F_B} = \rho \boldsymbol{g} V \quad (10)$$

where:

ρ: Density of the fluid $\left[\frac{kg}{m^3}\right]$
g: Gravitational acceleration $\left[\frac{m}{s^2}\right]$
V: Displaced (submerged) volume of the object $[m^3]$

It is important to mention that the buoyant force can be either a negative, positive, or neutral buoyancy. Specifically, it's negative if its effect is negligible and other forces like the weight have a far-reaching impact. If the buoyancy is positive, then it's the opposite: the biggest force is the buoyant. So, if its neutral, that means that the buoyant force is in equilibrium with other forces along its axis of action. And the weight if the projectile is $\boldsymbol{W} = m\boldsymbol{g}$. Variables such as air density, kinematic viscosity and gravity change with altitude, yet the change is very subtle, and it is considerable once we have high altitudes such as 1000 m and above.

Rewriting out the forces, we have the complete dynamic equations from where we can solve for each component of the acceleration. This is:

$$\boldsymbol{a} = a_x \boldsymbol{i} + a_y \boldsymbol{j} + a_z \boldsymbol{k} \quad (11)$$

$$a_x = \frac{A\rho}{2m}(-C_D v_x^2 + C_{Lx} v_y^2) \quad (12)$$

$$a_y = \frac{A\rho}{2m}(-C_D v_y^2 + C_{Ly} v_z^2) \quad (13)$$

$$a_z = \frac{A\rho}{2m}(-C_D v_z^2 + C_{Lz} v_x^2) + \frac{g}{m}(\rho V - m) \quad (14)$$

With $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ being the unit vectors of the Cartesian reference system.

Part III – Kinematics

Now, as the lift coefficient vary through time (depending on the velocity), the use of numerical methods is necessary to integrate the previous equation for each time instant of the flight. Several possibilities can be adopted here, such as the family of high-order Runge-Kuta schemes. For simplicity and clearness of our numerical formulation, we use the Euler's method to integrate the numerical solution for velocities and positions of the projectile at each time instant. A thorough analysis of the numerical accuracy and stability is presented at the results section to clarify the numerical integrator choice.

We know that if an object moves with a varying acceleration, then its velocity and position will greatly change. The acceleration is integrated in time, obtaining the velocity of the object defined as:

2 This was taken from Cengel & Cimbala (2010) and numerically interpolated with https://apps.automeris.io/wpd/.

$$\boldsymbol{v}(t+\Delta t)=\boldsymbol{v}(t)+\mathbf{a}(t)\Delta t \tag{15}$$

$$\boldsymbol{v}(t+\Delta t)=v_x(t+\Delta t)\boldsymbol{i}+v_y(t+\Delta t)\boldsymbol{j}+v_z(t+\Delta t)\boldsymbol{k} \tag{16}$$

where:
t: Time instant $[s]$
Δt: Time increment $[s]$
$v(t)$: Projectile velocity at (t) $\left[\frac{m}{s}\right]$
$v(t+\Delta t)$: Projectile velocity at $(t+\Delta t)$ $\left[\frac{m}{s}\right]$

If the acceleration is integrated twice, we obtain the position of the object defined as:

$$\boldsymbol{r}(t+\Delta t)=\frac{1}{2}\boldsymbol{a}(t)\Delta t^2+\boldsymbol{v}(t)\Delta t+\boldsymbol{r}(t) \tag{17}$$

$$\boldsymbol{r}(t+\Delta t)=x(t+\Delta t)\boldsymbol{i}+y(t+\Delta t)\boldsymbol{j}+z(t+\Delta t)\boldsymbol{k} \tag{18}$$

where:
$\boldsymbol{r}(t)$: Projectile position at (t) $[m]$
$\boldsymbol{r}(t+\Delta t)$:Projectile position at $(t+\Delta t)$ $[m]$

It is important to mention that both the final position and the complete flight time are known parameters of the problem. Finally, using equations 11-18 we can simulate the complete kinematics of the spherical projectile.

Coding process

We developed our main code by using all our variables and equations, and we integrated them by using the Euler method, which is a way to solve such problems involving differential equations. The use of this method is applied through a loop that resembles that of an iteration step by step, choosing our minimum value of time that is related to the computational power we have available. The coded algorithm follows the steps to be explained next:

1. **Initialization:** The code first sets up the necessary physical constants, such as gravity, drag, and Magnus coefficients. It also initializes arrays to store the projectile's positions, velocities, and accelerations.
2. **Defining forces:** The forces acting on the projectile are defined as functions to be used during the Euler's method loop. This includes gravitational force, drag force (which depends on the velocity and drag coefficient), and Magnus force (which depends on the rotation velocity of the projectile).
3. **Euler's method loop**:
   - **Forces calculation:** Forces are computed as functions of the actual velocities.
   - **Acceleration calculation:** Forces are divided by mass to obtain acceleration.
   - **Velocity update:** The current acceleration is added to the current velocity, multiplied by the time step, to obtain the new velocity.
   - **Position update:** The current velocity is added to the current position, multiplied by the time step, to obtain the new position.
   - **Terminal velocity check:** If the projectile reaches terminal velocity, it sets the velocity to the terminal velocity.
   - **Stop condition:** The loop checks whether the projectile has hit the ground, signaling that the simulation should stop.
4. **Storage of parameters:** Throughout the simulation, various parameters such as drag force, lift coefficients, and Reynolds number are stored at each time step.
5. **Termination:** The simulation ends when the stop condition is met.

We used MATLAB App Designer to build up an interface where we could change the projectiles and fluids parameters and where we could analyze the position, velocities, and accelerations of the projectile. But also, the software interface is designed to report the nondimensional coefficients and calculated forces along the projectile flight.

Simulation and results

It's important to mention that the Eulers method was used to compute the accelerations, as it provides a minimum error by calculating the values with an iterating differential, we developed an UI (user interface) that allows the user to easily use all the important variables that could get measured with utensils in real life. In this regard, we use several sources of data for real life measurements of a flying soccer ball and its properties, between them we can find that we have:

- Velocities: less than 211 km/h or less than 58.6 m/s. (Clayfield, n. d.).
- Mass of a ball: 450g aprox. (TheFA, n. d.).
- Diameter of a ball: 22 cm or 0.22 m (RSS).
- Air density (at 273 kelvin and 101.325 kPa): 1.293 kg/m^3 (Earth Data Open Access for Open Science, n. d.).
- Air kinematic viscosity: 1.48 × 10^{-5} m^2/s (Cadence CFD Solutions, n. d.)
- Gravity acceleration: 9.803 m/s^2 (Meyers, 2001).

The following screen (Figure 4) shows the initial conditions used for the software interface.

Under those conditions, we run the simulation and obtain a trajectory that is visualizable in three dimensions, a trajectory that is constructed by using the Euler's method point by point, and resolves into an actual parabola like trajectory, as seen in Figure 5.

Reports are also available for the trajectory visualization in two-dimensional planes, together with calculated forces and drag and lift coefficients through the projectile's flight. This has been designed to report that most of the accelerations dynamically depend on the calculated forces. This is observed in a separated graph: forces and accelerations actually change not being completely linear nor completely parabolic, showing how in real life we approximate these complex results as a parabola. Our position, velocities and accelerations results are shown in Figure 6.

As we can see, there are components that change over time, and by clicking on the interface graph, the exact value is deployed (red for positions, green for velocities and blue for accelerations). The drag forces (which are the forces of air resistance due to the friction and pressure of air), the Magnus force, and/or the gravity are depicted. On the z-axis or vertical axis, the acceleration tends to be the common gravity acceleration measurement when the ball is in its maximum height, where the velocities are small and only the gravity is considerable at all (in such axis at least).

We consider that the distance between the center of the Earth and the ball is considerable enough to even see a change of 1% of the gravity value. Yet if we want to consider such changes, we can make a difference

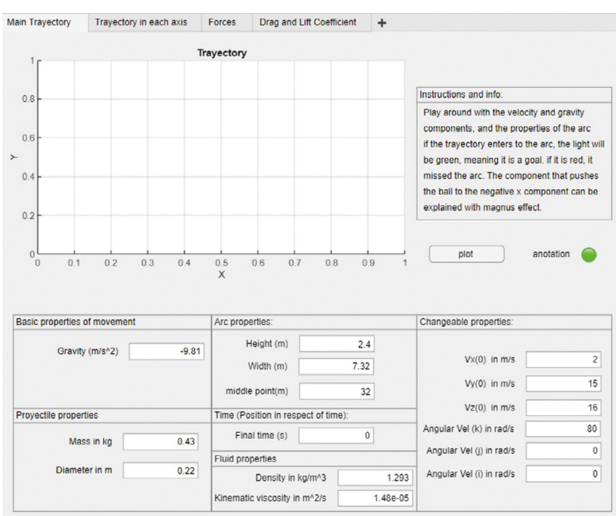

Figure 4. User interface: Initial conditions and parameters

Source: Own elaboration.

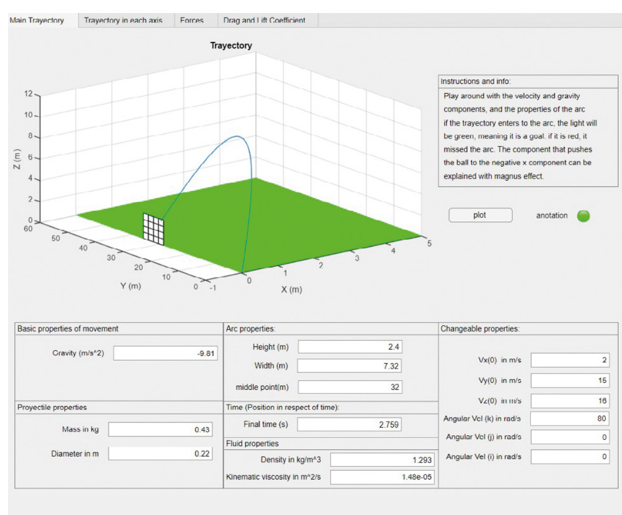

Figure 5. Trajectory result of the simulation

Source: Own elaboration.

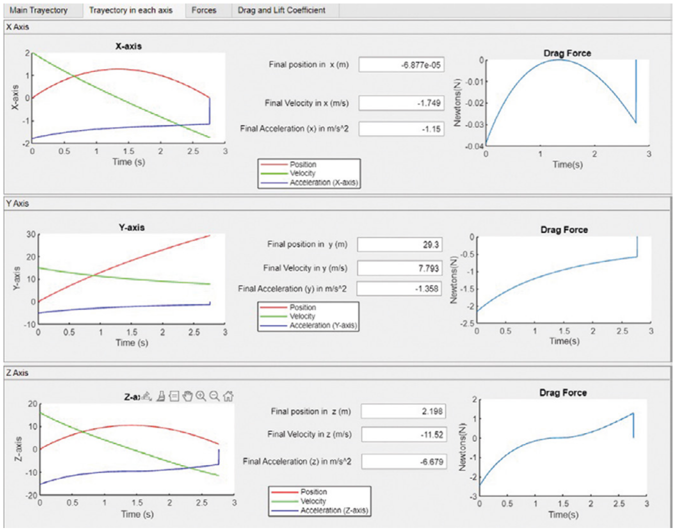

Figure 6. Developed trajectory and forces in each direction
Source: Own elaboration.

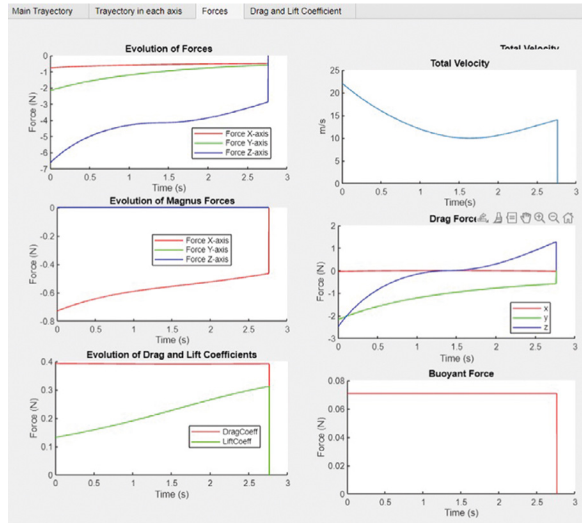

Figure 7. Developed forces in each direction
Source: Own elaboration.

between our local gravity, air density and air viscosity, and the ones considered through formulas. Remember that such considerations are not only negligible by chance, but also because it requires more computing processing power, which is one of the reasons we use numerical methods that do not resemble less than 1 millisecond (to generate more precision), yet the one we used is enough to particular problems and more refined methods would actually show a similar behavior (such as Runge-Kutta of 4th order).

We can see the development of the Magnus force, the drag coefficient and lift coefficient, and other properties in Figure 7. We can observe that the buoyant force (which is considered if the ball would be weighted in a vacuum) is constant and really minimal, showing how little of an impact it actually has in a projectile (unless it is submerged in a denser liquid like water).

The Reynolds number and the lift and drag coefficients are depicted in Figure 8. On the drag coefficient, we can stablish that we reach a plateau, where no noticeable variation is seen, this is due to the surface of the sphere (which is considered as a smooth ball). This is also the cause of the Figure 7 drag force constant behavior.

The Reynolds number result validates our model, since most of literature refer to the drag coefficient of a ball a number around 0.4, which is what we observed in our simulation. Also notice that the lift coefficient changes over time, which also explains how the air resistance has an impact on how the projectile moves with the Magnus effect, and as we can see, the lift force actually decreases (because it is imparting less negative acceleration), and thus we see that the acceleration on the x-axis actually decreases. Remember that also the ratio between the velocity of rotation in rad/s and the velocity of an axis affect how the forces are developed, consider we also just have rotation in an axis that allows force on the x-axis, but the user can add different rotation directions.

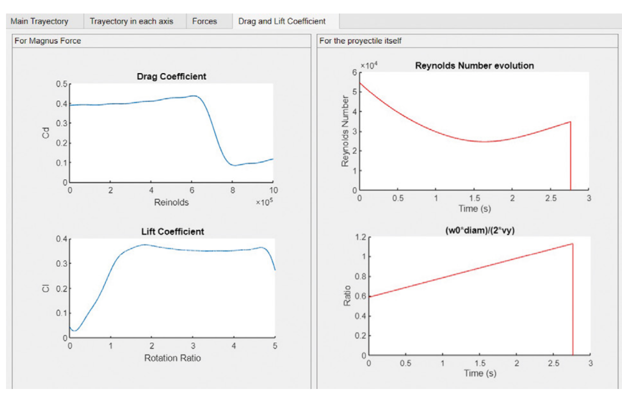

Figure 8. Developed non-dimensional coefficients
Source: Own elaboration.

A further investigation is performed over the time step used in the Euler method. This is important in order to validate the use of the present software. In that regard, results for several time steps (0.0001, 0.001, 0.01, 0.1, 0.5, 1) is used to create comparative graphs that show the sensitivity of the simulation results to different time step sizes. These graphs shown in Figure 9 are important in numerical methods, like Euler's method, because they can show how a smaller time step can lead to more accurate results, although at the cost of increased computation time.

In the provided log-log plots of the Euler method's results, we observe a consistent trend towards stabilization of solutions with decreasing time step sizes. This indicates convergence of the method, which is a desired attribute in numerical simulations. The plots reveal a distinct proportional relationship between the logarithm of the time step size and the logarithm of the solution's differences, suggesting a power-law behavior characteristic of convergent processes.

Euler's method, being a first-order technique, should exhibit linear convergence. This is substantiated by the log-log plots where the relationship between the log of the time step size and the log of the error approximates a straight line with a slope of -1 for the position and -1 for the velocity. This slope indicates that the Euler method's error is directly proportional to the time step size, confirming its linear order of convergence. Such linear convergence is typical for Euler's method and serves as a validation of its implementation.

The plots do not show any indication of numerical instability within the computed time scales. Numerical instability typically manifests as divergent solutions when the time step size is too large, which is not observed here. The absence of instability within these human-scale computational times indicates that the chosen time step sizes are within the method's stability limits. This is a positive indication of the robustness of Euler's method for the simulated scenarios.

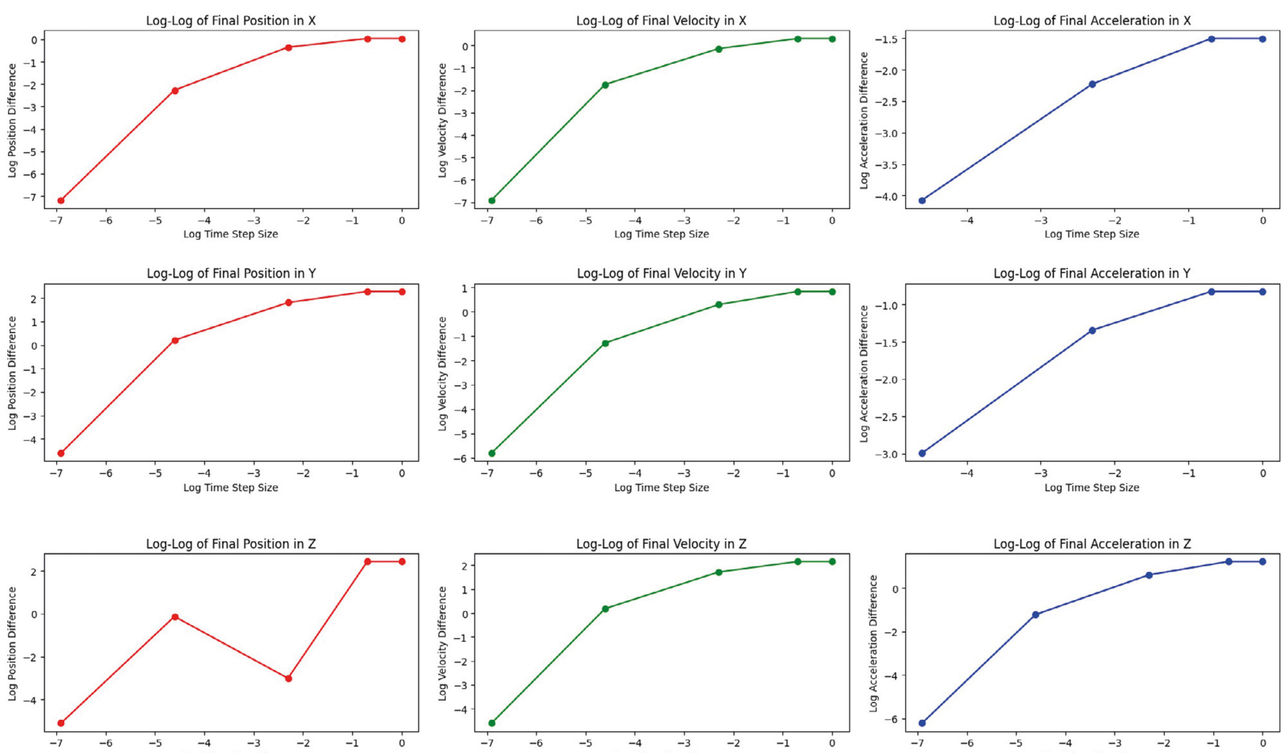

Figure 9. Convergence analysis of the Euler's method against the time step size. Log-log scales in all variables

Source: Own elaboration.

Escuela de Postgrados de la **Fuerza Aérea Colombiana**

While Euler's method has shown stable convergence, higher-order methods like Runge-Kutta offer the potential for more accurate solutions with fewer computational steps. The Runge-Kutta method, for example, can provide superior accuracy and stability, particularly for stiff equations or more complex dynamics. This could be a consideration for future work, where computational overhead allows for the use of more sophisticated algorithms to achieve better precision without proportionately increasing the computational cost.

Conclusions

- We have a versatile and user-friendly application that allows us to understand in a more enjoyable way how forces exerted by a fluid on an immersed solid object work. This method can be applied to any form of projectile, yet we have encountered that it is the best example of the underlying mechanical concepts. For example, by approximating the complex fluid flow solution of the Navier-Stokes, by using solely the Newton's 2nd law and nondimensional coefficients.
- We also understand that its development is genuine and realistic as it can complete the trajectory using real data. Furthermore, the investigation here exposed is a demonstration of basic fluid simulation in an integral manner, which is the most comprehensive and functional in the experimental field. Besides, the only 'limit' of the simulation is under extreme situations, where in some cases thermodynamics would need to be applied to account for changing viscosity and density of the fluid based on altitude and temperature reached by the projectile in motion. However, it works as an immediate analysis, and we can confidently say that it could be applied in real life in different fields giving an appropriated and complete description of motion, such as scoring an Olympic goal in a soccer game.
- Other numerical integration methods, such as the Runge-Kutta of 4th order, may be a future line of research, improving the accuracy of the velocity and position of the projectile in our software. The conducted sensitivity analysis underscores the reliability of Euler's method for the range of time steps considered. However, for applications demanding higher precision or those involving more complex dynamics, the implementation of higher-order methods such as Runge-Kutta could be beneficial. The choice of the numerical method must balance the computational cost with the required accuracy, and our analysis provides a solid foundation for making an informed decision in this regard.

References

Anderson, J. D. (2010). *Fundamentals of Aerodynamics* (5th ed.). McGrawHill.

Ang, D. G. (2013). Shape and Size Matter for Projectile Drag. *The Journal of Advanced Undergraduate Physics Laboratory Investigations, 2*. https://tinyurl.com/4k9uckwp

Blevins, R. D. (1985). *Applied Fluid Dynamics Handbook*. Van Nostrand Reinhold.

Cadence CFD Solutions (n. d.). *The Relationship Between the Kinematic Viscosity of Air and Temperature*. Cadence System Analysis. https://tinyurl.com/mtkpttn6

Cengel, Y. A. & Cimbala, J. M. (2010). *Fluid Mechanics Fundamentals and Applications*. McGrawHill.

Clayfield, B. (n. d.). *How Fast is a Soccer Ball Kicked?* [online]. https://tinyurl.com/3jm8mwmu

Cook, M. J. (2007). *Flight Dynamics Principles: A Linear Systems Approach to Aircraft Stability and Control*. Elsevier.

Dixit, U. S., Hazarika, M. & Davim, J. P. (2017). *A Brief History of Mechanical Engineering (Materials Forming, Machining and Tribology)*. Springer.

Earth Data Open Access for Open Science. (n. d.). *Air Mass/Density*. NASA. https://tinyurl.com/mph3fhb8

Goldstein, S. (1938). *Modern Developments in Fluid Dynamics: An Account of Theory and Experiment Relating to Boundary Layers Turbulent Motion and Wakes*. Oxford University Press.

Kray, T., Franke, J. & Frank, W. (2014). Magnus Effect on a Rotating Soccer Ball at High Reynolds Numbers. *Journal of Wind Engineering and Industrial Aerodynamics, 124*, 46-53. https://doi.org/10.1016/j.jweia.2013.10.010

Meriam, J. L. & Kraige, L. G. (2012). *Engineering Mechanics: Dynamics* (7th ed.). John Wiley & Sons.

Meyers, R. A. (Ed.). (2001). *Encyclopedia of Physical Science and Technology* (3rd ed.). Academic Press.

Mody, V. (2015). *High School Physics: Projectile Motion*. CreateSpace Independent Publishing Platform.

Robinson, G. & Robinson, I. K. (2013). The Motion of an Arbitrarily Rotating Spherical Projectile and its Application to Ball Games. *Physica Scripta, 88*(1). https://doi.org/10.1088/0031-8949/88/01/018101

TheFA. (n. d.). *Law 2: The Ball. IFAB Laws of the Game 2023-24* [online]. https://tinyurl.com/4kut4smy

Said, A. A., Mshewa, M. M., Mwakipunda, G. C., Ngata, M. R. & Mohamed, E. A. (2023). Computational Solution to the Problems of Projectile Motion under Significant Linear Drag Effect. *Open Journal of Applied Sciences, 13*(4), 508-528. https://doi.org/10.4236/ojapps.2023.134041

Sarafian, H. (2015). Impact of the Drag Force and the Magnus Effect on the Trajectory of a Baseball. *World Journal of Mechanics, 5*(4), 49-58. https://doi.org/10.4236/wjm.2015.54006