ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26aug1027

# Bridging Ideal Kinematics and Computational Drag in Mechanical and Ballistic Systems

Vedaant Karthik[1]

[1]Delhi Private School, Dubai

Publication Date: 2026/09/03

**Abstract: Motion in a plane under constant gravity provides the foundation for predicting parabolic paths in mechanical and aerospace engineering. This paper, using the fundamentals of projectile motion, focuses on three main aspects of mechanical engineering: optimization of conveyor belt discharge hopper placement, dual-angle targeting for pressurized nozzles, and interceptor trajectory for defensive systems. The purpose of this research is not to create a formula or an equation that is correct in every aspect, but rather to create a middle ground between rudimentary physics and complex simulation software. By analyzing the mechanics of projectile motion in an ideal drag-free system with uniform gravity and then creating a lightweight Python framework that integrates non-linear atmospheric drag alongside baseline closed-form algebraic equations, this study examines the relationship between physical variables such as launch velocity, inclination angle, and target coordinates. Ultimately, it shows how numerical modeling can bridge theoretical mechanics and practical physics at least in the initial stages of engineering design.**

***Keywords:*** *Projectile Motion, Quadratic Equations, Planar Kinematics, Conveyor Belt Discharge, Fluid Nozzles, No-Escape Zone, Runge-Kutta Method, Air Resistance, Aerodynamic Drag.*

**How to Cite:** Vedaant Karthik (2026) Bridging Ideal Kinematics and Computational Drag in Mechanical and Ballistic Systems. *International Journal of Innovative Science and Research Technology*, 11(8), 2580-2587. https://doi.org/10.38124/ijisrt/26aug1027

## I. INTRODUCTION

Mechanics is the branch of physics focused on the behavior of physical bodies when subjected to forces or displacements [1]. Kinematics here serves as the study of motion - establishing the essential groundwork for physical design and engineering analysis. In basic planar kinematics, two-dimensional motion under a uniform gravitational field is characterized by objects following curved parabolic trajectories. These fundamentals have numerous applications in day-to-day mechanical engineering:

Continuous material handling systems use motorized belts to move bulk goods (such as grain, coal, or gravel) across processing facilities. When material reaches the end of the belt (the head pulley), it flies off into open air under gravity toward a receiving chute or hopper [2]. Predicting this parabolic trajectory is useful to prevent spillage, material degradation, and equipment malfunction or failure. This is referred to as a conveyor belt discharge system. Further, high-pressure liquid jets (nozzles) are used extensively in firefighting, agricultural sprayers, and cooling systems. Controlling or optimizing nozzle pressure and launch angle allows us to determine where the fluid stream lands.

Lastly, a point-defense system is a military shield that is used to protect important facilities or infrastructure [3]. These systems fire high-velocity interceptor projectiles or missiles to collide with and neutralize airborne targets at specific coordinates before impact. It is essential to determine launch angles and velocities for the same.

The research is motivated by the need to show how classical physics, when paired with computational tools, can be effectively applied to complex engineering problems. This study by no means seeks to replace advanced simulations; rather, it highlights the practical utility of evaluating both ideal zero-drag kinematics and non-linear quadratic drag within a Python framework. This helps to establish a middle ground between basic closed-form kinematics and heavy computational fluid dynamics (CFD) programs for rapid calculations. While the assumptions and physical constraints of this methodology are examined in detail within the Literature Review itself, this research serves as a mathematically strong starting point that can be built upon by future exploration.

## II. LITERATURE REVIEW

The study of projectile motion goes back to classical Newtonian mechanics, where objects moving under constant gravity follow simple parabolic paths. In ideal vacuum conditions, horizontal and vertical movements can be separated and solved using closed-form algebra. However, real projectiles moving through air experience aerodynamic drag. G.W. Parker studied kinematics with air resistance proportional to the square of velocity and showed that the resulting equations become coupled and nonlinear [4]. J.L. Bradshaw

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26aug1027

further examined quadratic-drag projectile motion and demonstrated the use of analytical and numerical methods to determine projectile trajectories [5].

Projectile motion has also been applied to engineering systems such as conveyor belt discharge. Hastie and Wypych reviewed methods for predicting the trajectory of material leaving conveyor belts and showed that trajectory prediction is relevant to the design of receiving and transfer systems [2]. Further research identified factors including belt velocity, pulley geometry, material properties, particle behavior and air drag as influences on the actual discharge trajectory [6]. More detailed studies have also used experimental methods and Discrete Element Modelling to investigate three-dimensional bulk-material behavior during conveyor discharge [7]. These methods can represent particle-particle collisions, material spreading, segregation, impact behavior, wear, and the interaction between the bulk material and chute surfaces. For medium- and high-speed material flows, however, basic projectile equations are still used as an initial method for plotting the discharge trajectory and estimating the location of downstream equipment [8]. Therefore, the hopper system in this paper is intentionally simplified and does not attempt to reproduce the complete physics of a real conveyor system.

A similar simplification can be made for fluid nozzles. A liquid stream leaving a nozzle can be approximated as a projectile when its initial velocity and direction are considered and gravitational motion is isolated. However, real liquid jets are affected by additional factors such as nozzle geometry, pressure, fluid properties, turbulence, velocity distribution and air entrainment [9]. Studies of liquid jets show that nozzle shape and flow conditions influence both jet trajectory and breakup characteristics. [10]. The closed form equation analysis, along with the computational drag python model do not account fluid continuity, jet breakup, droplet-size distribution, viscosity, surface tension, turbulence, nozzle geometry, air crossflow, evaporation, or spray dispersion. Consequently, the calculated landing point is not intended to represent the complete shape or intensity distribution of a real liquid jet. It is instead a kinematic estimate that demonstrates how launch speed and angle affect reachability.

In the case of defensive interceptors, aerodynamic drag becomes increasingly significant as velocity increases because quadratic drag depends on the square of velocity [11]. More complete atmospheric models can also consider changing air density, changing aerodynamic coefficients, three-dimensional motion, crosswinds and other aerodynamic effects. Before fire, missile types, operational delay, and firepower are also evaluated [12]. These factors are acknowledged in this study but are not included in the model. The term “interceptor” in this paper therefore refers only to a hypothetical projectile.

Thus, the three engineering cases in this research are not intended to be complete models of conveyor hoppers, fluid nozzles, or interceptors. In every case, the analysis is deliberately restricted to two models: ideal projectile motion without drags and non-ideal projectile motion with quadratic drag.

## III. METHODOLOGY

➢ *Classical Kinematics (Baseline)*

For this half of the methodology, we will be considering ideal (classical) mechanics, disregarding aerodynamic drag.

Consider a particle launched at time $t = 0$ from origin (0, 0) with initial speed $u$ at an angle $\theta$ relative to the ground. Assuming negligible air resistance, the initial velocity vector can be expressed in terms of horizontal and vertical components:

$$u_x = u\cos\theta$$

$$u_y = u\sin\theta$$

With zero horizontal acceleration ($a_x = 0$), the horizontal distance over a time ‘t’ is:

$$x(t) = (u\cos\theta) \quad (1)$$

Along the vertical axis, gravity pulls downward at $a_y = -g$. The vertical position, therefore, is:

$$y(t) = (u\sin\theta)t - \frac{1}{2}gt^2 \quad (2)$$

Solving for $t$ from (1) and substituting in (2), we get:

$$y = (u\sin\theta)\left(\frac{x}{u\cos\theta}\right) - \frac{1}{2}g\left(\frac{x}{u\cos\theta}\right)^2$$

$$y = x\tan\theta - \frac{g}{2u^2\cos^2\theta}x^2 \quad (3)$$

This equation describes the trajectory of a particle in projectile motion. Rearranging, and taking height ‘$h$’ = $y$ where $h$ represents the height at which the projectile strikes another surface, we have a quadratic equation of the form $Ax^2 + Bx + C = 0$.

$$\left(\frac{g}{2u^2\cos^2\theta}\right)x^2 - (\tan\theta)x + h = 0$$

We can now use the quadratic formula to compute the landing distance of the given projectile.

$$x = \frac{\tan\theta \pm \sqrt{\tan^2\theta - \frac{2gh}{u^2\cos^2\theta}}}{\frac{g}{u^2\cos^2\theta}} \quad (4)$$

Note that the “landing” distance here refers to the $x$ coordinate of the projectile upon reaching the target height. If $h = 0$ , as in, the projectile strikes the ground, the equation yields the standard range formula $x = \frac{u^2\sin(2\theta)}{g}$. For $h > 0$ and $h < 0$, we get multiple values of $x$ which will be discussed in depth in further sections.

Having derived Equation (4), we know that

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26aug1027

$$\text{DISCRIMINANT } = D_1 = \tan^2\theta - \frac{2gh}{u^2\cos^2\theta}$$

If $D > 0$, the equation has two real roots, implying that the object crosses height $h$ at two $x$ coordinates. $D = 0$ means that that parabolic path touches height $h$ exactly once, whereas $D < 0$ gives complex roots, suggesting that the initial velocity $u$ is too low to reach the target height $h$. To calculate the required launch angle $\theta$ to hit a given coordinate $(x_0, y_0)$, we substitute $\frac{1}{\cos^2\theta} = 1 + \tan^2\theta$ into Equation (3):

$$y_0 = x_0\tan\theta - \frac{gx_0^2}{2u^2}(1 + \tan^2\theta)$$

Substituting $\tan\theta$ as $T$,

$$\left(\frac{gx_0^2}{2u^2}\right)T^2 - x_0T + \left(y_0 + \frac{gx_0^2}{2u^2}\right) = 0 \qquad (5)$$

Thus, we have a quadratic equation in terms of $\tan\theta$ that can be used to compute the launch angle. Solving this equation for $T$ yields two roots ($T_1$ and $T_2$), providing two distinct launch angles ($\theta_1 = \arctan T_1$ *and* $\theta_2 = \arctan T_2$). This proves that two different parabolic angles can reach the exact same target point. It may also be noted that setting the discriminant of this equation to 0 gives the maximum height that can be reached by the projectile for a given initial velocity $u$ and range $x$. This maximum height will be termed as the "No-Escape Zone".

$$D_2 = x_0^2 - 4\left(\frac{gx_0^2}{2u^2}\right)\left(h + \frac{gx_0^2}{2u^2}\right) \qquad (6)$$

Equating to 0:

$$y_{\max} = \frac{u^2}{2g} - \frac{g}{2u^2}x^2$$

➢ *Case Study 1: Conveyor Belt Discharge (Hopper System)*

When handling bulky amounts of materials, loose material (coal, grains, limestone, etc.) leaves a conveyor belt at speed $u$ and angle $\theta$ , following a projectile path, toward a hopper located below the discharge pulley ($h < 0$) [2]. Because $h$ is negative, $h = -|h|$, the discriminant of Equation (4) becomes:

$$D_1 = \tan^2\theta + \frac{2g|h|}{u^2\cos^2\theta} > 0$$

Thus Equation 4 yields two roots, of which the negative root is discarded.

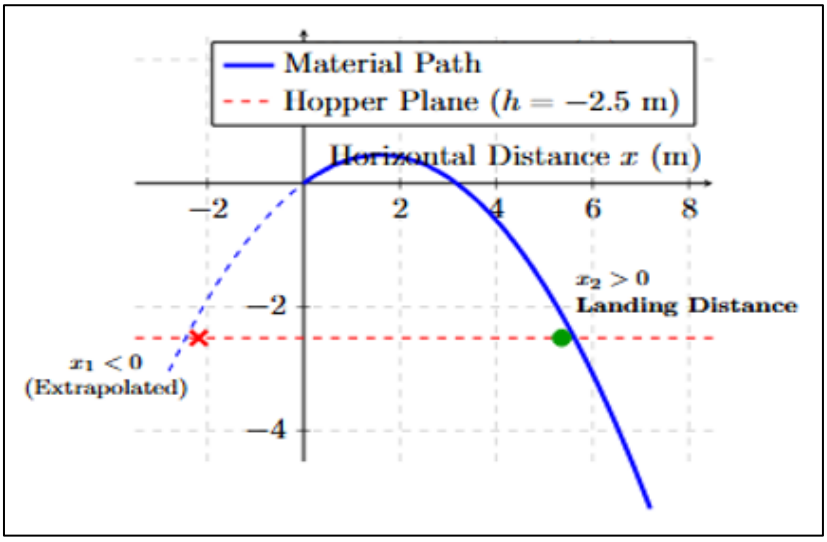


Fig 1 Trajectory Graph Showing the Physical Path ($x \geq 0$) and the Non-Real Root. ($x_1 < 0$).

- *Sample Industrial Calculations:*

Consider a conveyor belt operating with belt speed $u = 10.0$ m/s, angle $\theta = 30.0°$, target hopper drop height $h = -1.50$ m, and $g = 9.80665$ m/s$^2$ .

These values fall well within CEMA (Conveyor Equipment Manufacturers Association) standards [13].

Substituting into Equation 4:

$$-1.50 = x\tan(30°) - \frac{9.80665}{2(10.0)^2\cos^2(30°)}x^2$$

$$0.06538x^2 - 0.57735x - 1.50 = 0$$

Discarding the negative root leaves $x = 10.93m$, thus providing the exact hopper placement.

➢ *Case Study 2: Fluid Nozzle*

In the case of fluid nozzles like sprinklers and fire-hose nozzles, when $D_1 = \tan^2\theta - \frac{2gh}{u^2\cos^2\theta} > 0$, the fluid stream crosses the target elevation at two points, $x_1$ during ascent and $x_2$ during descent.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26aug1027

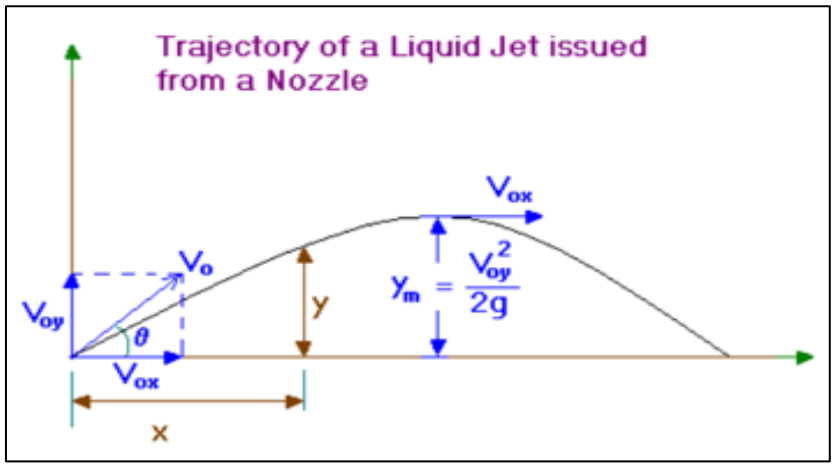


Fig 2 Trajectory of a Liquid Jet Issued from a Nozzle

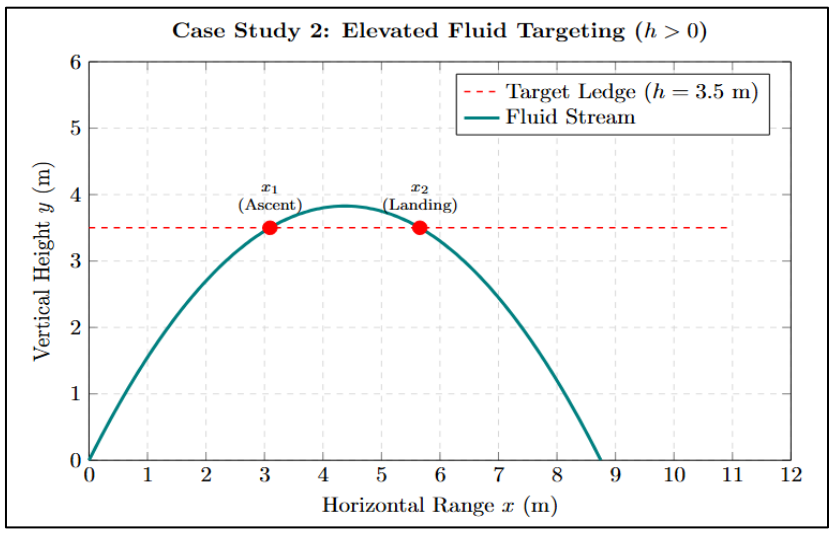


Fig 3 Liquid Crossing Target Height (3.5 Meters) at Two Different Points, $x_1 = 3.09m$ and $x_2 = 5.66m$

- *Sample Industrial Calculations:*

For a pressurized water nozzle targeting a coordinate ($x_0 = 20.0$ m, $y_0 = 5.0$ m) with a launch speed $u = 18.0$ m/sunder gravity ($g = 9.80665$ m/s$^2$):

$$\left(\frac{9.80665 \times 20.0^2}{2 \times 18.0^2}\right)T^2 - 20.0T + \left(5.0 + \frac{9.80665 \times 20.0^2}{2 \times 18.0^2}\right) = 0$$

Solving for $T = \tan\theta$ gives $T_1 \approx 0.7028$, $T_2 \approx 2.6008$ . Finally, we get:

$$\theta_1 \approx 35.10°$$

$$\theta_2 \approx 68.97°$$

➢ *Case Study 3: Defensive Interception*

As discussed earlier, for point-defense interceptors engaging threats at target altitude $h$, the discriminant (Equation 6) dictates the physical feasibility of the interceptor.

$$D_2 = x_0^2 - 4\left(\frac{gx_0^2}{2u^2}\right)\left(h + \frac{gx_0^2}{2u^2}\right)$$

Equating it to zero yields the "no-escape zone" – the height beyond which the missile cannot engage the threat:

$$y_{\max} = \frac{u^2}{2g} - \frac{g}{2u^2}x^2$$

The engagement of the defensive missile with the incoming threat can be divided into three distinct cases:

- Sub-Critical (D<0): The target altitude lies above the system's kinetic reachability envelope. No real launch angle exists ($\theta \notin \mathbb{R}$).
- Apex Intercept (D=0): The target coordinate lies exactly on the boundary ($y_{max}$). Exactly one unique launch angle ($\theta = arctanT$) hits the target at the peak.
- Super-Critical (D>0): The target sits strictly inside the reachability zone. This produces two distinct launch angles ($\theta 1 = arctanT1$ and $\theta 2 = arctanT2$), - one with a low trajectory and one with a high trajectory – both of which intersect the target coordinate.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26aug1027

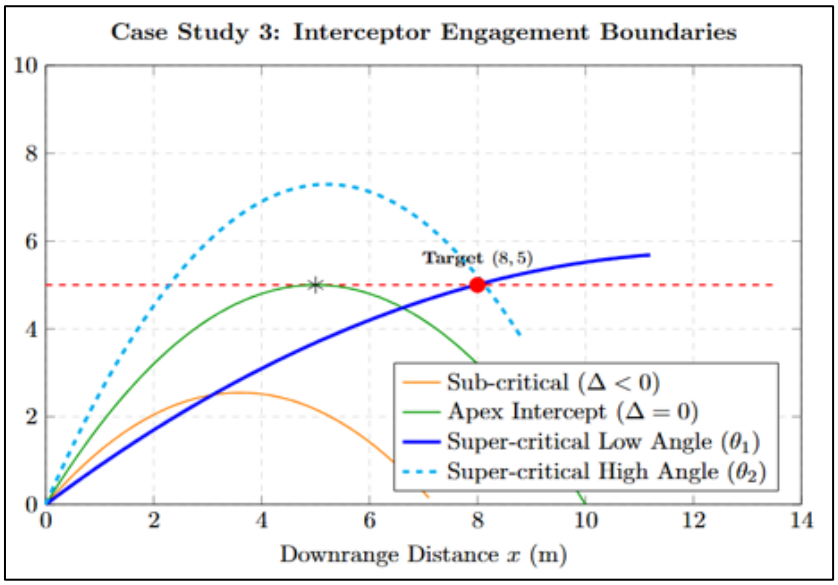


Fig 4 Graph Showing Sub-Critical Reachability ($D < 0$), Single Apex Contact ($D = 0$) – Target Taken as (5,5), and Two Different Launch Angles ($\theta_1, \theta_2$) Intersecting the Target Coordinate (8,5) in Super-Critical Conditions ($D > 0$).

- *Sample Industrial Calculations:*

For a defense battery firing an interceptor at $u = 250$ m/s to hit a target at $x_0 = 3000$ m, $y_0 = 1200$ m:

$$706.08T^2 - 3000T + 1906.08 = 0$$

Solving this equation gives $T_1 \approx 0.7788 \Longrightarrow \theta_1 \approx 37.91°$and $T_2 \approx 3.4693 \Longrightarrow \theta_2 \approx 73.93°$. The maximum vertical ceiling directly overhead ($x = 0$), from Equation (6):

$$y_{\max} = \frac{250^2}{2 \times 9.80665} \approx 3186.6 \text{ m}$$

Therefore, at this launch velocity, the defense system can intercept artillery at a maximum of 3186.6 m – provided that it is directly overhead. This is the no escape zone.

➢ *Computational Analysis and Further Physics:*

The above case studies computed under ideal kinematics assume zero horizontal drag and constant vertical acceleration. However, in non-ideal environments, aerodynamic resistance and drag significantly alters trajectory behavior. This section of the paper will analyze the true path of the projectiles implementing a Python-based numerical solver to integrate the non-linear differential equations of quadratic drag and measure how real trajectories diverge from ideal parabolic models through a comparative graph. It may be noted that the following derivation of the Ordinary Differential Equations (ODE) is not novel and is from existing literature. Projectiles experience quadratic aerodynamic drag given by:

$$\vec{F}_d = -\frac{1}{2}\rho C_d A \|\vec{v}\|\vec{v}$$

Where $\rho$ represents fluid density, $C_d$ denotes the drag coefficient, $A$ is cross-sectional area, and $\vec{v}$ is instantaneous velocity. Let the velocity vector be defined:

$$\vec{v} = \begin{pmatrix} v_x \\ v_y \end{pmatrix}$$

Applying Newton's second law, and taking the upwards direction as positive:

$$m\frac{d\vec{v}}{dt} = \vec{F}_g + \vec{F}_d$$

$$\vec{F}_g = \begin{pmatrix} 0 \\ -mg \end{pmatrix}$$

We then substitute the gravitational and aerodynamic forces and divide by mass $m$ throughout.

$$\frac{d\vec{v}}{dt} = \begin{pmatrix} 0 \\ -g \end{pmatrix} - \frac{\rho C_d A}{2m}\sqrt{v_x^2 + v_y^2}\begin{pmatrix} v_x \\ v_y \end{pmatrix}$$

Now let

```
# 3. ODE and introducing the function
def trajectory_ode(t, state, k_factor):
    x, y, vx, vy = state
    v = np.hypot(vx, vy)
    return [vx, vy, -k_factor * v * vx, -g - (k_factor * v * vy)]
```

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26aug1027

$$k = \frac{\rho C_d A}{2m} \quad (7)$$

The acceleration vector can then be written as

$$\frac{d\vec{v}}{dt} = \begin{pmatrix} -kv_x\sqrt{v_x^2 + v_y^2} \\ -g - kv_y\sqrt{v_x^2 + v_y^2} \end{pmatrix}.$$

Finally, we resolve the acceleration vector into its horizontal and vertical components and get the coupled system of ordinary differential equations:

$$\boxed{\frac{dv_x}{dt} = -kv_x\sqrt{v_x^2 + v_y^2}}$$

$$\boxed{\frac{dv_y}{dt} = -g - kv_y\sqrt{v_x^2 + v_y^2}}.$$

Because coupled ordinary differential equations do not have a simple solution, we use a Runge-Kutta (RK) – an iterative numerical alogirthm – order 5(4) integrator on Python (*solve_ivp*). :

We begin by assigning the $\theta$ and $u$ values from cases 2 and 3. For the conveyor belt discharge system, drag is not as prominent as in the nozzle and interceptors, and hence for the compactness of this section it has not been included. The value of constant $k$ as defined in equation (7) has been taken, in each of the cases, using empirical values for $C_d, A, \rho, and\ m$.

```
import numpy as np
from scipy.integrate import solve_ivp
import matplotlib.pyplot as plt
g = 9.80665
# 1. Our Case study 2
u_jet = 18.0
theta_jet_deg = 35.10
k_jet = 0.5 * 1.225 * 0.47 * (np.pi * 0.003**2) / 0.005
# 2. Our Case study 3
u_missile = 250.0
theta_missile_deg = 45
k_missile = 0.5 * 1.225 * 0.20 * 0.0177 / 15.0
```

We define the function (*trajectory_ode)* to model how the projectile moves through the air while considering both gravity and air resistance. The function takes three inputs: continuous time $t$, a state vector (*state* ) containing the projectile's current horizontal position $x$, vertical position $y$, horizontal velocity $v_x$, and vertical velocity $v_y$, and the lumped drag factor $k$. The line *(x, y, vx, vy = state)* unpacks these four values so they each can be used individually in calculations, after which the total speed (accounting for horizontal and vertical speed) is calculated. Finally, the function returns four rates of change, which are repeatedly solved by *(solve_ivp)* to calculate the exact position and velocity of the projectile at every instant of flight:

```
# Case Study 2
rad_jet = np.radians(theta_jet_deg)
init_jet = [0.0, 0.0, u_jet * np.cos(rad_jet), u_jet * np.sin(rad_jet)]
t_eval_jet = np.linspace(0, 3.0, 300)
sol_jet = solve_ivp(trajectory_ode, (0, 3.0), init_jet, args=(k_jet,), t_eval=t_eval_jet)
# Case Study 3
rad_missile = np.radians(theta_missile_deg)
init_missile = [0.0, 0.0, u_missile * np.cos(rad_missile), u_missile * np.sin(rad_missile)]
t_eval_missile = np.linspace(0, 25.0, 500)
sol_missile = solve_ivp(trajectory_ode, (0, 25.0), init_missile, args=(k_missile,), t_eval=t_eval_missile)
```

Next, we convert the launch angles from degrees to radians using (*np.radians*) and construct the initial state vector for both case studies. We define time arrays using (*np.linspace*) to specify the evaluation window. The *(solve_ivp)* function then integrates the differential equations over time and stores the output inside solution objects. By comparing these simulated $x$ and $y$ coordinates against the theoretical zero-drag trajectory equation $y_{\text{ideal}}(x) = x\tan\theta - \frac{gx^2}{2u^2\cos^2\theta}$, we can extract numerical metrics to calculate the precise altitude loss, range reduction, and percentage errors caused by air drag. This is done from plotting graphs generated by *matplotlib* using data from the above code, illustrated graphically under the "Results and Discussion" section.

## IV. RESULTS AND DISCUSSIONS

For the ideal model (without air resistance), we used standard trajectory formulas and solved quadratic equations to analyze three setup cases: positioning a conveyor hopper, aiming a fluid nozzle, and intercepting a target. By checking

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26aug1027

the discriminant, we determined whether a trajectory could hit a specific target point. This highlighted the numerous practical applications of classical physics in mechanical engineering. To model real-world conditions, we used Python (`scipy.integrate.solve_ivp`) with a 5th-order Runge-Kutta algorithm to solve the differential equations with air resistance and plot the trajectories. Extracting these results into Table 1 shows that air drag increases rapidly as speed goes up ($F_d \propto v^2$). For the low-speed fluid jet ($u = 18.0$ m/s), drag makes only a small difference, reducing peak height by 2.01% and range by 3.96%. However, for the high-speed interceptor missile ( $u = 250.0$ m/s), drag causes a massive drop, peak altitude falls by 23.48% (1202.9 m $to$ 920.5 m) and total range collapses by 37.88% (6178.7 m to 3838.2 m). Because the drag-damped peak only reaches 920.5 m, the missile fails to hit the 1200.0 m target entirely.A question then arises: What anglewould the missile have to be fired at, with the same velocity to hit the same target height, in a real-world environment with air resistance? What is the difference between this angle and the angle obtained in ideal conditions (Case Study 3)? Intuitively, the presence of drag would necessitate a larger launch angle. Simple trial and error into the Python framework built during the methodology – reveals that an angle of 45 degrees results in the interceptor reaching a height of 1200m, albeit at a much lower x coordinate. An empirical formula or observation for the same may be arrived at through future research and derivation.

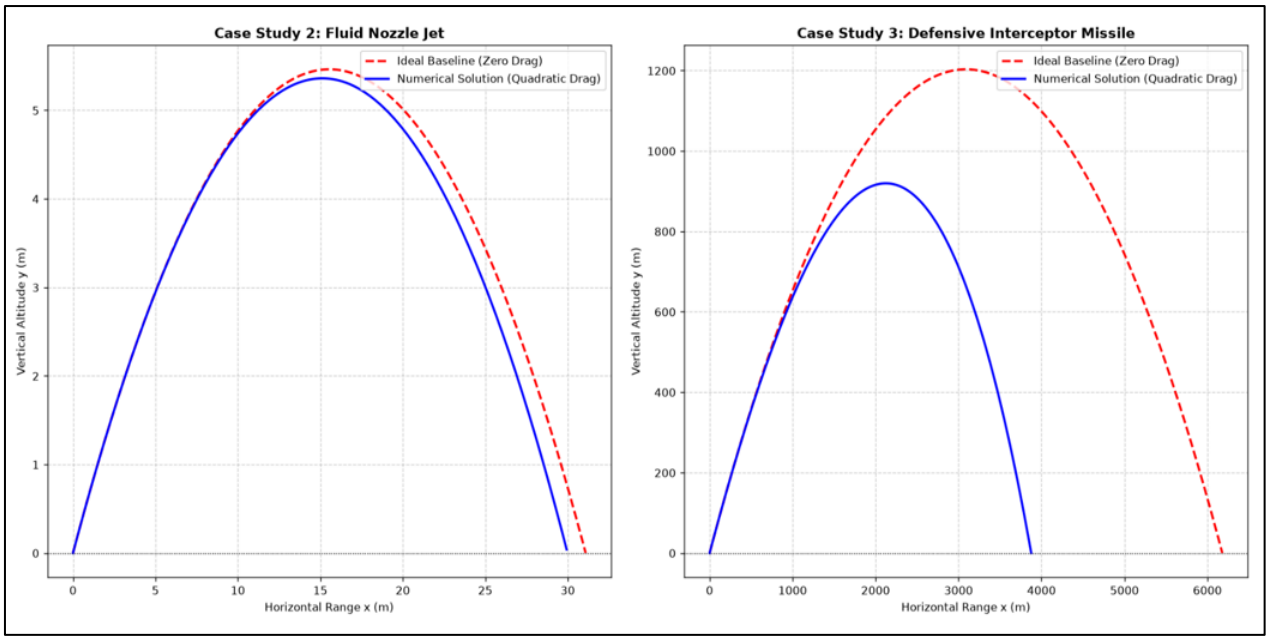


Fig 5 Graphs Created by *matplotlib* Showing Trajectory of Projectile Under Ideal and Real-Life Conditions (Quadratic Drag).

Table 1 Ideal vs Drag Integration

| System Scenario | Metric | Closed-Form Ideal | Numerical Drag (ODE) | Relative Error (%) |
|---|---|---|---|---|
| **Case Study 2 (Fluid Jet)** | Target Intercept ($x$ at $y = 5.0$ m) | 20.00 m | 18.45 m | -7.75% |
| $u = 18.0$ m/s, $\theta = 35.10°$ | Apex Altitude ($y_{max}$) | 5.46 m | 5.35 m | -2.01% |
| | Max Range ($R$) | 31.08 m | 29.85 m | -3.96% |
| **Case Study 3 (Interceptor)** | Target Intercept ($x$ at $y = 1200.0$ m) | 3000.0 m | Unreachable ($y_{max} < 1200$ m) | – |
| $u = 250.0$ m/s, $\theta = 37.91°$ | Apex Altitude ($y_{max}$) | 1202.9 m | 920.5 m | -23.48% |
| | Max Range ($R$) | 6178.7 m | 3838.2 m | -37.88% |

## V. CONCLUSION

This paper evaluated the limits of classical kinematic theory by contrasting closed-form parabolic solutions against a custom numerical Python solver across three engineering applications. The closed-form analytical equations serve as a crucial theoretical starting point, establishing the upper bound (maximum possible performance) of any ballistic trajectory before environmental losses are introduced. However, incorporating quadratic air resistance ($F_d \propto v^2$) via Python's `scipy.integrate.solve_ivp` highlights the drastic performance drop experienced by high-velocity systems. While high-fidelity Computational Fluid Dynamics (CFD) packages—such as ANSYS (Analysis System) Fluent, OpenFOAM, or COMSOL Multiphysics—provide exact 3D aerodynamic flow profiles, they demand significant computational resources, complex CAD (Computer-Aided Design ) geometry inputs, and long render times. This custom Python framework offers a fast, accessible, and lightweight numerical alternative. It bridges the gap between oversimplified formulas and heavy industry simulations - enabling effective initial-stage trajectory screening and control adjustments during early engineering design. Thefeore, despite its limitations, this analysis and model serves as a mathematiacally rigorous starting point for future research.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26aug1027

## REFERENCES


[1]. P. Dourmashkin, "Chapter 1: Introduction to classical mechanics," *MIT OpenCourseWare: 8.01SC Classical Mechanics*, Fall 2016. [Online]. Available: https://ocw.mit.edu/courses/8-01sc-classical-mechanics-fall-2016/mit8_01scs22_chapter1.pdf

[2]. D.B. Hastie and P.W. Wypych, "The prediction of conveyor trajectories," in *Proc. Int. Materials Handling Conf. (Beltcon 14)*, Johannesburg, South Africa, Aug. 2007, Paper B14-02.

[3]. Forumul Securității Maritime, "Point-defense missile systems," *Forumul Securității Maritime*. [Online]. Available:https://www.forumulsecuritatiimaritime.ro/point-defense-missile-systems/

[4]. G.W. Parker, "Projectile motion with air resistance quadratic in the speed," *Amer. J. Phys.*, vol. 45, no. 7, pp. 606–610, July 1977, doi: 10.1119/1.10812.

[5]. J. L. Bradshaw, "Projectile motion with quadratic drag," American Journal of Physics, vol. 91, no. 4, pp. 258–263, 2023, doi: 10.1119/5.0095643.

[6]. D. B. Hastie, P. W. Wypych and P. C. Arnold, "Influences on the prediction of conveyor trajectory profiles," *Particulate Science and Technology*, vol. 28, no. 2, pp. 132–145, 2010.

[7]. D. Ilic and C. Wheeler, "Transverse bulk solid behaviour during discharge from troughed belt conveyors," *Advanced Powder Technology*, vol. 28, no. 9, pp. 2410–2430, 2017.

[8]. D. J. Kruse, "Chute designs and trajectories using the discrete element method," in *Proc. Int. Materials Handling Conf. (Beltcon 15)*, Boksburg, South Africa, Sep. 2009, Paper B15-15. [Online]. Available: https://www.beltcon.org.za/wp-content/uploads/2024/12/B15-15-Kruse-Chute-Designs-and-Trajectories-using-DEM.pdf

[9]. Y. Dong, L. Bai, C. Liu, and L. Wang, "Modeling water jet trajectory based on three step profile velocity assumption and fluid governing equations along the centerline of the jet," *J. Eng. Res.*, 2026, doi: 10.1016/j.jer.2026.06.034.

[10]. E. Pulat, B. Sözen, and E. Erim, "Numerical and experimental investigation of projectile trajectory with aerodynamic drag," *Journal of Thermal Engineering*, vol. 10, no. 5, pp. 1285–1296, Sept. 2024.

[11]. Owen, J. P., & Ryu, W. S. (2005). The effects of linear and quadratic drag on falling spheres: An undergraduate laboratory. *European Journal of Physics*, *26*(6), 1085–1091. https://doi.org/10.1088/0143-0807/26/6/016

[12]. Cintron, L. (2010). Embedded systems – missile detection/interception. *Undergraduate Journal of Mathematical Modeling: One + Two*, *2*(2), Article 2. https://doi.org/10.5038/2326-3652.2.2.2

[13]. Conveyor Equipment Manufacturers Association. (2004). *Bulk material belt conveyor troughing and return idlers selection and dimensions* (CEMA Standard No. 502-2004).