Panamerican Mathematical Journal
ISSN: 1064-9735
Vol 35 No. 4s (2025)

TWO-DIMENSIONAL PURSUIT PROBLEMS ON PARAMETRIC CURVES

JUAN TRIANA AND JHONNATAN RIVEROS

Abstract. In this paper, we provide a brief historical background on the pursuit curve problem, starting with the classical pursuit curve problem solved by Pierre Bouguer in the 18th century. Motivated by Hathaway's circular pursuit problem, for which no closed-form analytical solution exists, we present a generalization based on a simple pursuit-evasion problem where the evader moves along a parametric curve in two dimensions. Due to the difficulties or impossibility of finding an analytical solution, we perform a numerical approximation to solve the system of differential equations that models the problem, using the odeint function from Python. As a result, we perform computational simulations of a two-dimensional pursuit problem to generate graphical representations of the trajectories of both the pursuer and the evader, considering their initial positions and speeds, as well as various parametric curves for the evader's trajectory.

1. Introduction

The pursuit and evasion problem is often attributed to Leonardo da Vinci in the context of a cat-and-mouse chase [19]; however, there is insufficient evidence to support this claim [10]. The study of pursuit curves dates back to Pierre Bouguer's 1732 paper *Sur de nouvelles courbes auxquelles on peut donner le nom de lignes de poursuite*, in which he analyzed the case of a pirate ship traveling at constant speed v_p in pursuit of a merchant vessel moving at constant speed v_m. Bouguer's pursuit problem is often regarded as the starting point of pursuit analysis; however, the term pursuit curve was coined by George Boole in 1859. [15].

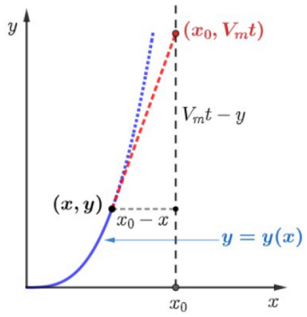

Figure 1. The geometry of Bouguer's pursuit problem.

2020 *Mathematics Subject Classification.* Primary 49N75; Secondary 34A30, 53A04.
Key words and phrases. Pursuit problem, system of differential equations, parametric curves.

Panamerican Mathematical Journal
ISSN: 1064-9735
Vol 35 No. 4s (2025)

At some time $t > 0$, the pirate ship and the merchant vessel positions are (x, y) and $(x_0, v_m t)$ respectively. Thus, the slope of the tangent line to the pursuit curve at the point (x, y) is given by

$$\frac{dy}{dx} = \frac{y - v_m t}{x - x_0} \tag{1.1}$$

From Equation (1.1) we have

$$t = \frac{y}{v_m} - \frac{x - x_0}{v_m}\frac{dy}{dx} \tag{1.2}$$

Since the pirate ship travels with constant speed v_p, we have the distance traveled given by $v_p t$. On the other hand, the distance is the arc length given by

$$v_p t = \int_0^x \sqrt{1 + \left(\frac{dy}{dz}\right)^2}\, dz \tag{1.3}$$

An analytical solution to Equation (1.3) can be found in [1]. However, in most pursuit problems we need numerical methods such as Euler, Heun, and Runge-Kutta methods and their modifications to solve systems of differential equations [3]. An approach by finite difference schemes to avoid the differential equation in the modeling process can be considered [11]. Solving for t in Equation (1.3) and by equating with Equation (1.2), we get

$$\frac{1}{v_p}\int_0^x \sqrt{1 + \left(\frac{dy}{dz}\right)^2}\, dz = \frac{y}{v_m} - \frac{x - x_0}{v_m}\frac{dy}{dx} \tag{1.4}$$

In 1877, Edouard Lucas posed a problem about the trajectories generated by three dogs, initially placed at the vertices of an equilateral triangle, each chasing the next in a cyclic manner. In 1880, Henri Brocard presented a solution, showing that the dogs would converge at a common point, currently known as the Brocard point of a triangle. This problem marked the beginning of the study of cyclic pursuit [8].

The problem of circular pursuit, also known as Hathaway's circular pursuit problem, was introduced in a 1920 issue of the American Mathematical Monthly in the context of a dog and a duck chase, where a dog is positioned at the center of a circular pond while a duck swims along its edge. A brief historical overview of this problem can be found in [10]. In addition, an interesting approach to obtaining an analytical solution of Hathaway's circular pursuit problem by Lie symmetries can be found in [2].

Pursuit curve problems arise in various contexts, allowing different movement strategies for both the pursuer and the evader [16]. These problems can even be generalized to scenarios involving multiple interacting objects [7]. In addition, pursuit problems have pedagogical value, as they introduce undergraduate students to modeling real-life situations using differential equations [6]. Even when considering the simplest two-dimensional case, where the pursued moves in a straight line and both the pursuer and the pursued have constant velocities, pursuit problems can serve as valuable teaching tools [9]. In practical applications, a capture may be considered to occur when the pursuer is within a short distance of the evader [18].

In this paper, motivated by Hathaway's circular pursuit problem, we present computational simulations of a two-dimensional pursuit problem in which the pursued object follows a parametric curve. Due to the complexity of obtaining an analytical solution for the resulting system of differential equations, we generate numerical approximations using *odeint* function from Python's SciPy library.

2. Material and methods

The position vector of the pursuer and the evader at some arbitrary time $t \geq 0$ are denoted $P(t)$ and $E(t)$ respectively. Let $r(t)$ be the separation vector between $P(t)$ and $E(t)$, thus $r(t) = E(t) - P(t)$. Here we use an Argand diagram to represent the vectors [4], thus a vector $w(t)$ is rewritten in the form $w(t) = x_w(t) + iy_w(t)$. Since we are interested in a pursuit problem, we have that the pursuer's velocity is a vector pointing at the evader for any $t \geq 0$, therefore

$$\frac{dP}{dt}(t) = \left|\frac{dP}{dt}(t)\right| \hat{r}(t) = \left|\frac{dP}{dt}(t)\right| \frac{E(t) - P(t)}{|E(t) - P(t)|} \tag{2.1}$$

On the other hand, the evader's velocity is given by $\frac{dE}{dt}(t) = \frac{dx_E}{dt} + i\frac{dy_E}{dt}$. Without loss of generality, we assume that the pursuer is n times faster than the evader, thus the speed of the evader is

$$\left|\frac{dP}{dt}(t)\right| = n\sqrt{\left(\frac{dx_E}{dt}\right)^2 + \left(\frac{dy_E}{dt}\right)^2} \tag{2.2}$$

By replacing Equation (2.2) in Equation (2.1) we have

$$\frac{dP}{dt}(t) = n\sqrt{\left(\frac{dx_E}{dt}\right)^2 + \left(\frac{dy_E}{dt}\right)^2} \frac{E(t) - P(t)}{|E(t) - P(t)|} \tag{2.3}$$

Since $E(t) = x_E(t) + iy_E(t)$ and $P(t) = x_P(t) + iy_P(t)$ we get the following system of differential equations

$$\frac{dx_P}{dt} = n\sqrt{\left(\frac{dx_E}{dt}\right)^2 + \left(\frac{dy_E}{dt}\right)^2} \frac{x_E(t) - x_P(t)}{\sqrt{(x_E(t) - x_P(t))^2 + (y_E(t) - y_P(t))^2}}$$

$$\frac{dy_P}{dt} = n\sqrt{\left(\frac{dx_E}{dt}\right)^2 + \left(\frac{dy_E}{dt}\right)^2} \frac{y_E(t) - y_P(t)}{\sqrt{(x_E(t) - x_P(t))^2 + (y_E(t) - y_P(t))^2}}$$

Numerical methods for solving first-order differential equations with an initial value can be adapted to solve systems of first-order differential equations [14]. In [13], a comparison between the RK4 method and Python's *odeint* function for solving Bouguer's problem was presented. Here we present a Python code to solve a pursuit problem.

```
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt
from math import *
```

Panamerican Mathematical Journal
ISSN: 1064-9735
Vol 35 No. 4s (2025)

We begin our Python code by importing the necessary modules, including *odeint* from *scipy.integrate* package, to solve the system of differential equations. The following Python code represents the system of differential equations for a pursuit problem in which the evader is moving along the edge of the ellipse $x = 4cos(t)$, $y = 2sin(t)$.

```
def systemODE(w, t):
    x, y = w
    n=1.03
    xe=4*cos(t)
    ye=2*sin(t)
    l=sqrt((-4*sin(t))**2 + (2*cos(t))**2)
    dxdt = n*l*((xe-x)/sqrt((xe-x)**2 + (ye-y)**2))
    dydt = n*l*((ye-y)/sqrt((xe-x)**2 + (ye-y)**2))
    return [dxdt, dydt]
```

Since $n > 1$ we will have that the pursuer captures the evader. We consider pursuer starts from $(0, 0)$, for the simulation we use 500 nodes.

```
x0 = 0
y0 = 0
w0 = [x0, y0]
t = np.linspace(0,pi,500)
xe=4*np.cos(t)
ye=2*np.sin(t)
sol = odeint(systemODE, w0, t)
plt.figure()
plt.plot(sol[:,0], sol[:, 1], label='Pursuer')
plt.plot(xe, ye, label='Evader',linestyle='--')
plt.legend()
plt.show()
```

It is easy to verify that we get Hathaway's circular pursuit problem when we consider $x = acos(t)$, $y = asin(t)$ with $a > 0$. The Following Figure shows the result of the pursuit problem described by the code above.

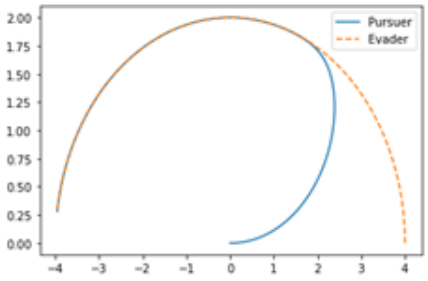

FIGURE 2. Simulation with $n = 1.03$.

The code presented in this section will be useful for generating simulations of pursuits on several trajectories, but we need to compute derivatives. Therefore, we may use the *sympy* library of Python to compute these derivatives through

Panamerican Mathematical Journal
ISSN: 1064-9735
Vol 35 No. 4s (2025)

symbolic variables. However, it should be noted that using symbolic calculations increases the computational cost.

3. Results and discussion

The following figure shows the simulations of the pursuit problem with the evader moving along the edge of the ellipse defined by $x = 4cos(t)$, $y = 2sin(t)$ for several values of $0 < n < 1$. Since the pursuer does not capture the evader we consider a longer time interval for the simulation. Consequently, we can observe the trajectory described by the pursuer (t = np.linspace(0, 30, 500)).

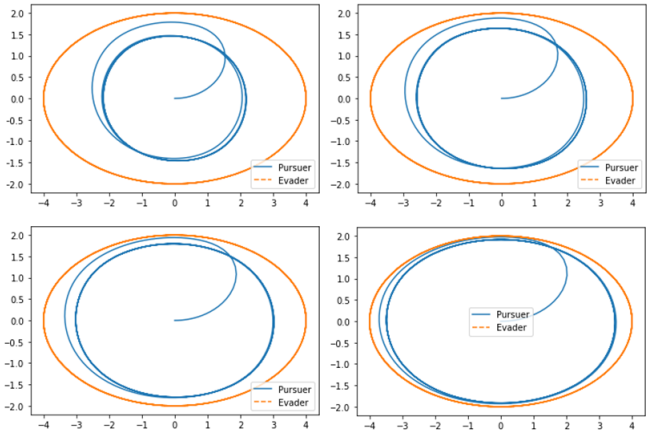

Figure 3. Simulation with $n = 0.6, 0.7, 0.8$ and 0.9 respectively with final time $t_f = 30$.

We define a capture as occurring when the distance between the evader and the pursuer is 0.01 or less. To analyze the sensitivity of the calculations with respect to the capture time and the number of nodes considered, we perform the simulation with the evader moving along $x = 4cos(t)$, $y = 2sin(t)$ and the pursuer initially positioned at the origin, for several values of n using 500 and 5000 nodes.

Table 1. Simulation of the capture time for several tolerance values

n	Capture time with 500 nodes	Capture time with 5000 nodes
1.05	2.07761	2.06758
1.04	2.40499	2.39311
1.03	3.08493	3.08566
1.02	4.15521	4.14522
1.01	6.44057	6.43087

We conclude that the model is highly sensitive to the value of n, but not to the number of nodes. The following figure shows the simulation of the pursuit problem with the evader moving along the edge of the cicloid $x = t - sin(t)$, $y = -cos(t)$ with $n = 1.05$ (t = np.linspace(0, 6.8, 500)) and $n = 0.9$ (t = np.linspace(0, 12, 500)) and the pursuer starting at the point $(-1, 0)$.

Panamerican Mathematical Journal
ISSN: 1064-9735
Vol 35 No. 4s (2025)

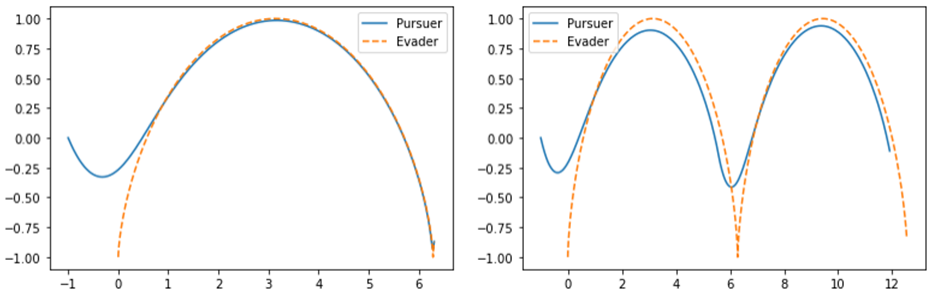

FIGURE 4. Simulation with $n = 1.05$ and (left), simulation with $n = 0.9$ (right)

The following figure shows the simulation of the pursuit problem with the evader moving along the edge of the astroid such that $x = sin^3(t)$, $y = cos^3(t)$ with $n = 0.9$. The pursuer is located at the origin.

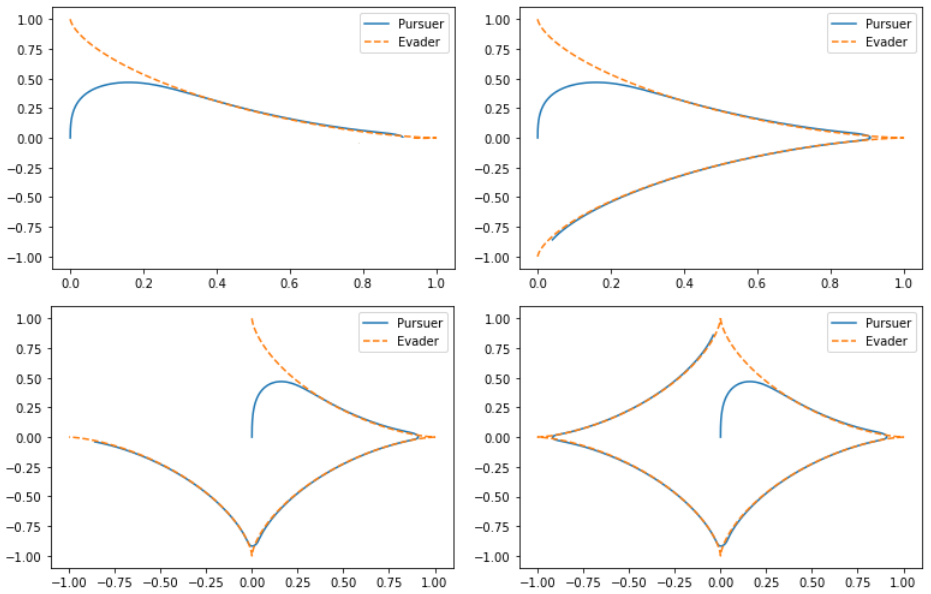

FIGURE 5. Simulation with n=0.9 on the edge of the astroid.

4. Conclusion

Reviewing applied mathematical methods for solving problems is an effective approach in problem-based learning (PBL) [12], since pursuit curves originate from real-life contexts. Therefore, pursuit problems can be incorporated into PBL activities [5]. To motivate students during these activities, it is useful to introduce a problem framed in a historical context [17]. We suggest using Bouguer's problem to assess students' conceptual understanding of modeling with differential equations.

Panamerican Mathematical Journal
ISSN: 1064-9735
Vol 35 No. 4s (2025)

A comparison between solutions obtained through numerical methods, such as the Runge-Kutta methods implemented by students in Python and those using the *odeint* function, can serve as a numerical methods challenge. This activity can be replicated using other software tools, such as Matlab. Students may also be interested in exploring alternative pursuit strategies as part of a PBL activity. However, teacher guidance is essential, particularly in topics such as differential equations, parametric curves, numerical methods, and programming languages.

As a programming challenge, we could propose implementing a dynamic simulation of pursuit curves. It is recommended to use the Python library *matplotlib* and import various functions to generate animations in different output formats, such as HTML5 and JSHTML formats.

To prepare students for calculating and plotting parametric equations, symbolic libraries can be used. As a programming challenge, we propose to perform 2D pursuit simulations using symbolic mathematics; the user input will consist of the parametric curve entered via the keyboard, along with the initial position and time data, the output of the code will be a graphical simulation of the pursuit problem.

References

1. T. Azevedo, A. Pelluso. Space pirates: a pursuit curve problem involving retarded time. American journal of physics, vol. 90, 2022, pp. 730-735. https://doi.org/10.1119/5.0069298.
2. C. Basquerotto, A. Ruiz, S. Da Silva, H. Weber. An illustrative application of the Lie symmetries in the context of first-order mechanical systems: Hathaway's circular pursuit problem. Acta mechanica, vol. 233, 2022, 1031-1039. https://doi.org/10.1007/s00707-022-03150-5.
3. R. Chapra, S. Canale. Numerical methods for engineers (7th ed.). Mc Graw Hill, 2015.
4. L. Ferro, J. Triana, S. Mendoza, A geometric interpretation of the multiplication of complex numbers. Journal of physics: Conference series, vol. 1674, 2020, 012005. doi:10.1088/1742-6596/1674/1/012005.
5. D. Fisher, Y. Kusumah, J. Dahlan, J. Project-based learning in mathematics: A literature review. Journal of physics: Conference series, vol. 1657, 2020, 012032. doi:10.1088/1742-6596/1657/1/012032.
6. R. Gutoiu, T. Cheche. The Dog and rabbit chase problem. Romanian reports in physics, vol. 74, 2022, 903.
7. S. Kumkov, S. Le Ménec, V., Patko. Solvability sets in pursuit problem with two pursuers and one evader. IFAC Proceeding, Vol. 47, no. 3, 2014, 1543-1549. https://doi.org/10.3182/20140824-6-ZA-1003.01853.
8. J. Marshall, M. Broucke, B. Francis. Pursuit formations of unicycles. Automatica, vol. 42, no. 1, 2006, 3-12. https://doi.org/10.1016/j.automatica.2005.08.001.
9. I. Mancev, L. Kostic. Various methods for solving the pursuit problem with two objects on a plane. European journal of physics, vol. 44, no. 3, 2023, 035002. https://doi.org/10.1088/1361-6404/acc0e7.
10. Nahin, P. Chases and escapes: The mathematics of pursuit and evasion. Princeton University Press, 2012.
11. V. Ochkov, I. Vasileva. Application of difference schemes to the decision the pursuit problem. Spiiras proceedings, vol. 18, no. 6, 2019. 1407-1433.https://doi.org/10.15622/sp.2019.18.6.1407-1433.
12. S. Rézio, M. Andrade, M. Teodoro. Problem-based learning and applied mathematics. Mathematics, vol. 10, no. 16, 2022, 2862. https://doi.org/10.3390/math10162862.
13. J. Riveros, J. Triana. Modelado y simulación de curvas de persecución para el problema de Bouguer mediante Python. Editorial Universidad Católica de Colombia, 48-60. https://doi.org/10.14718/EncuentroCienc.Basicas.2023.7.5.
14. T. Sauer. Numerical analysis (3rd ed.) Pearson, 2018.

Panamerican Mathematical Journal
ISSN: 1064-9735
Vol 35 No. 4s (2025)

15. K. Toman, N. Voulgarakis. Stochastic pursuit-evasion curves for foraging dynamics. Physica A: Statistical mechanics and its applications, vol. 597, 2022, 127324. https://doi.org/10.1016/j.physa.2022.127324
16. J. Triana, D. Roldán. Una técnica de persecución utilizando mimetismo no convencional. Revista UDCA actualidad y divulgación científica vol. 16, no. 2, 2013, 519-523. https://doi.org/10.31910/rudca.v16.n2.2013.926.
17. J. Triana, J. Ruiz. El arma secreta de la primera guerra mundial. Revista cubana de física, vol. 35, no. 1, 2018, 60-66.
18. A. Von Moll, M. Patcher, Z. Fuch. Pure pursuit with an effector. Dynamic games and applications, vol. 13, 2023, 961-979. https://doi.org/10.1007/s13235-022-00481-9
19. S. Yoshihara, T. Ohira. Pursuit and evasion: from singles to groups. Journal of physics: Conference series, vol. 2207, 2020, 012014. doi:10.1088/1742-6596/2207/1/012014

Department of Basic Science, Universitaria Agustiniana, Bogotá, Colombia
Email address: juang.triana@uniagustiniana.edu.co

IED El Porvenir, Bogotá, Colombia
Email address: jhonnatan.riveros284@educacion.bogota.edu.co