

La vache et le paysan

Année 2025-2026

Alice Puech, Arthur Jacques, Amandine Pons, Noa Boucher : élèves de première et terminale
Honoré Sallard, Elise Nurit, Méline Calmejane : élèves de seconde

Établissement(s) : Lycée Émile Duclaux, Aurillac

Enseignant-e(s) : Rocq Alexandre, Lallemand Fabrice

Chercheur-Chercheuse(s) : Manchon Dominique, CNRS

Table des matières

1	Introduction	1
1.1	Présentation du sujet	1
1.2	Résultats	2
2	Définition précise de la configuration	2
3	Résolution approchée par la « force brute »	2
4	Étude du problème à l'aide du théorème de Pythagore	4
5	Infiniment petit et trigonométrie	6
5.1	Première conjecture	6
5.2	Utilisation de l'infiniment petit	6
5.3	Lien avec notre première conjecture	7
6	Conclusion	8
7	Annexe	8
	Notes d'édition	10

1. Introduction

1.1. Présentation du sujet

En début d'année, nous avons choisi de travailler sur le problème suivant : *La vache et le paysan* dont l'objectif était de trouver le chemin le plus rapide que devait emprunter le paysan pour rejoindre sa vache. Nous allons donc vous présenter les différentes approches qui nous ont permis de déterminer le chemin le plus court.

On se place dans un repère orthonormé $(O; \vec{i}, \vec{j})$ du plan. Un paysan part du point de départ $P(-10,0)$ et souhaite rejoindre sa vache au point $V(5,12)$ en prenant le moins de temps possible. Or,

son chemin est divisé en deux : une partie herbeuse dans laquelle il se déplace à 2 m/s et une partie boueuse pendant laquelle sa vitesse diminue à 1 m/s.

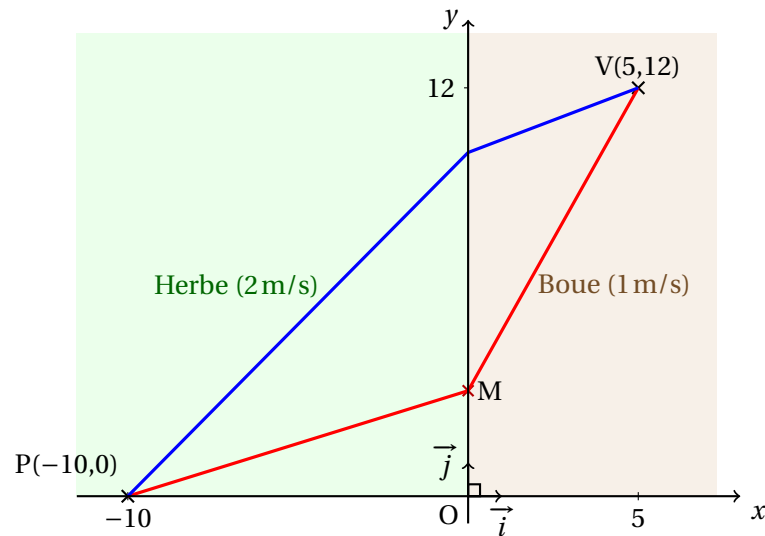


FIGURE 1 – Schéma du problème dans un repère orthonormé avec deux trajectoires possibles.

1.2. Résultats

- Nous avons obtenu une solution approchée en réalisant une simulation numérique avec Python.
- Nous avons ensuite étudié le problème d'un point de vue géométrique en utilisant le théorème de Pythagore et la trigonométrie.

2. Définition précise de la configuration

Nous avons d'abord choisi des valeurs par défaut. Elles correspondent à la configuration de la figure 1. P est la position du paysan et V celle de la vache. M est un point mobile sur l'axe des ordonnées, donc de coordonnées $(0, a)$ permettant de trouver le chemin le plus court. Le problème consiste à déterminer la position du point M, donc la valeur de a , sur l'axe des ordonnées qui minimise le temps de trajet. On cherche a dans l'intervalle $[0, 12]$ car M doit être situé entre P et V.

Sur la figure 1, nous avons tracé le chemin bleu symbolisant le chemin le plus rapide.

3. Résolution approchée par la « force brute »

Nous avons d'abord choisi de représenter tous les chemins possibles pour le problème et d'en déduire le meilleur grâce à un programme de force brute en langage Python.

Ce dernier va calculer, en fonction des coordonnées des points de départ et pour chaque valeur de a allant de 0 jusqu'à 12 (avec un pas de 0,1), le temps pris pour rejoindre le point V. Ce calcul permet d'en déduire un tracé sur un graphique grâce au module turtle : nous représentons la trajectoire avec une couleur qui varie en fonction du temps de trajet, plus le temps est court, plus la couleur est bleue, et inversement, plus le temps est long, plus la couleur est rouge.

Les variations de rouge, vert et bleu sont calculées en fonction du temps : on calcule d'abord un coefficient qui est égal à :

$$\text{coefficient} = \frac{\text{temps}_{\text{parcours}} - \text{temps}_{\text{min}}}{\text{temps}_{\text{max}} - \text{temps}_{\text{min}}}$$

Ensuite, nous appliquons ce coefficient à trois fonctions R, G, B :

$$\Rightarrow G(\text{coef}) = -255 \times \text{coef} + 255$$

$$\Rightarrow B(\text{coef}) = 255/255^{\text{coef}}$$

$$\Rightarrow R(\text{coef}) = 255 \times \text{coef}$$

$$\Rightarrow \text{Si } \text{coef} = 0 \Leftrightarrow \text{Meilleur chemin}$$

Le programme permet donc de visualiser chaque chemin et de déterminer, selon les coordonnées du Paysan et de la Vache, une solution approchée du meilleur chemin.

Voici le code de la fonction qui calcule le temps de parcours pour chaque valeur de a et qui retourne un dictionnaire associant à chaque valeur de a le temps de parcours correspondant. Le code de la fonction de tracé du graphique est placé en annexe. (1)

```
from math import sqrt

def calcule_temps(xP,yP,xV,yV,v1,v2):
    """ recherche du temps minimum pour atteindre
        le point V à partir du point P
    """

    xM=0
    yM=0
    temps={}

    # boucle pour déterminer la série des temps de parcours
    while yM<=12:
        #calcul des distances
        PM=sqrt((xM-xP)**2+(yM-yP)**2)
        MV=sqrt((xV-xM)**2+(yV-yM)**2)

        #calcul du temps total
        temp1=PM/v1
        temp2=MV/v2
        temp_total=temp1+temp2
        temps[yM]=temp_total
        yM= round(yM+0.1,1)

    cles=[k for k in temps.keys()]
    value=[i for i in temps.values()]
    mini=100
    indice=0

    # boucle pour trouver le temps minimum
    while indice<len(value):
        if value[indice]<=mini:
            mini=value[indice]
            kmini=indice
        indice+=1

    print("le temps minimum est",mini,"s avec a =",cles[kmini])

    return temps
```

```
# Exécution de la fonction
calculer_temps(-10,0,5,12,2,1)
```

Pour une exécution avec les points : P(-10,0), S(5,12)

Nous obtenons le graphique ci-dessous :

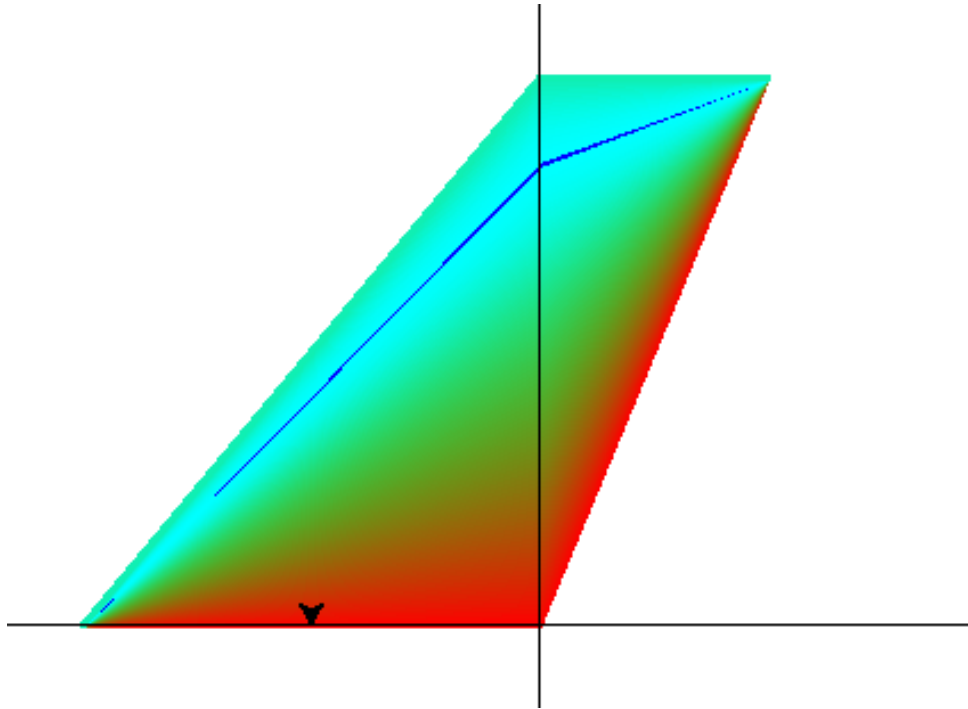


FIGURE 2 – Tracé des trajectoires possibles

Avec cette simulation, nous obtenons une valeur approchée de $a \approx 10,1$ qui correspond à la position du point M pour laquelle le temps de parcours est minimal, soit environ 12,46 sec.

4. Étude du problème à l'aide du théorème de Pythagore

Nous nous appuyons sur la configuration de la figure 1 pour trouver une expression du temps de trajet en fonction de a .

Tout d'abord, on peut calculer la distance PM grâce au théorème de Pythagore appliqué dans le triangle POM rectangle en O :

$$PM = \sqrt{100 + a^2}$$

De même, on peut calculer la distance MV :

$$MV = \sqrt{25 + (12 - a)^2}$$

Le temps de parcours dans l'herbe, à la vitesse de 2 m/s est alors :

$$t_{herbe} = \frac{PM}{2} = \frac{\sqrt{100 + a^2}}{2}$$

et le temps de parcours dans la boue, à la vitesse de 1 m/s, vaut :

$$t_{boue} = MV = \sqrt{25 + (12 - a)^2}$$

Nous pouvons maintenant donner une expression de la fonction f en a , $f(a)$ qui correspond au temps total de parcours en fonction de a :

$$f(a) = \frac{\sqrt{100 + a^2}}{2} + \sqrt{25 + (12 - a)^2}$$

Nous avons cherché à trouver le minimum de cette fonction pour déterminer la valeur de a qui correspond au chemin le plus rapide.

Pour trouver le minimum, il faut calculer la dérivée, on sait que f est de la forme $u + v$ avec u définie par :

$$u(a) = \frac{\sqrt{100 + a^2}}{2}$$

et v est définie par :

$$v(a) = \sqrt{25 + (12 - a)^2}$$

On en déduit que $f' = u' + v'$:

$$\begin{aligned} u'(a) &= \frac{2a}{4\sqrt{100 + a^2}} = \frac{a}{2\sqrt{100 + a^2}} \\ v'(a) &= \frac{-(12 - a)}{\sqrt{25 + (12 - a)^2}} = \frac{a - 12}{\sqrt{25 + (12 - a)^2}} \\ f'(a) &= \frac{a}{2\sqrt{100 + a^2}} + \frac{a - 12}{\sqrt{25 + (12 - a)^2}} \end{aligned}$$

Nous n'avons pas réussi à étudier le signe de la dérivée pour trouver le minimum de la fonction $f(a)$. La simplification de la dérivée donne des polynômes de degré 5 que nous ne savons pas résoudre. Nous avons donc utilisé une approche graphique à l'aide de la calculatrice.

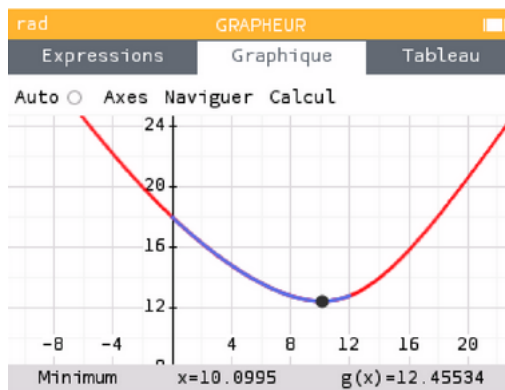


FIGURE 3 – Tracé de la courbe de f pour trouver le minimum

Nous avons rentré la formule dans le grapheur de la calculatrice, pour calculer le temps le plus court mis par le paysan pour rejoindre sa vache : $f(a) = \frac{\sqrt{100 + a^2}}{2} + \sqrt{25 + (12 - a)^2}$.

Nous avons obtenu $a \approx 10,0995$ qui est la valeur minimale de a pour que PV soit le plus court possible (en temps de parcours) et $f(a) \approx 12,45534$.

Ces résultats sont cohérents avec ceux obtenus par la simulation numérique, ce qui nous a permis de valider notre approche numérique.

5. Infiniment petit et trigonométrie

5.1. Première conjecture

En réalisant des figures à l'échelle de la trajectoire optimale, sur papier, puis avec Geogebra, nous avons mesuré les angles α et β définis sur la figure 4 et nous avons observé que α était environ deux fois plus grand que β lorsque la vitesse est deux fois plus élevée.

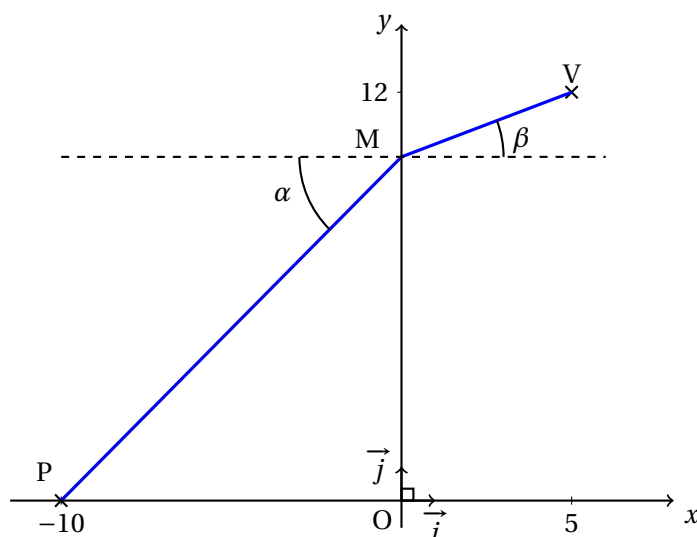


FIGURE 4 – Définition des angles α et β pour la conjecture

Nous avons alors émis la conjecture suivante :

$$\text{Si } n = \frac{v_1}{v_2}, \text{ alors } \alpha = n \times \beta$$

Nous avons voulu vérifier cette conjecture en faisant de nouvelles figures avec une vitesse de 3 m/s dans la partie herbe et 1 m/s dans la partie boue pour s'assurer que α était bien 3 fois plus grand que β . Cela a confirmé notre conjecture.

5.2. Utilisation de l'infiniment petit

Nous avons, cependant, voulu vérifier si notre conjecture fonctionnait avec des valeurs extrêmes. Nos tests avec par exemple un point P très proche du point O, nous ont montré que la conjecture ne fonctionnait pas dans tous les cas. Dans ce cas, l'angle α est proche de 90° et nous n'avons pas pu trouver de relation simple entre les angles α et β .

Nous avons ensuite cherché à comprendre pourquoi cela ne fonctionnait pas.

Notre professeur a alors proposé d'augmenter de δ l'ordonnée du point M en position optimale pour définir un point M'

Nous obtenons alors la figure 5.

Comme la variation δ est très petite, nous pouvons considérer que la trajectoire PM'V est encore optimale et donc que la petite longueur d_1 ajoutée à la trajectoire dans la partie herbe compense exactement la petite longueur d_2 retirée dans la partie boue.

En zoomant autour des points M et M', nous obtenons la figure 6 (2).

Nous considérons donc que le temps mis pour parcourir d_1 est égal au temps mis pour parcourir d_2 , soit $\frac{d_1}{2} = d_2$ et donc $d_1 = 2d_2$.

En utilisant les angles complémentaires, nous pouvons établir les égalités d'angles indiquées sur la figure 6, ce qui nous donne :

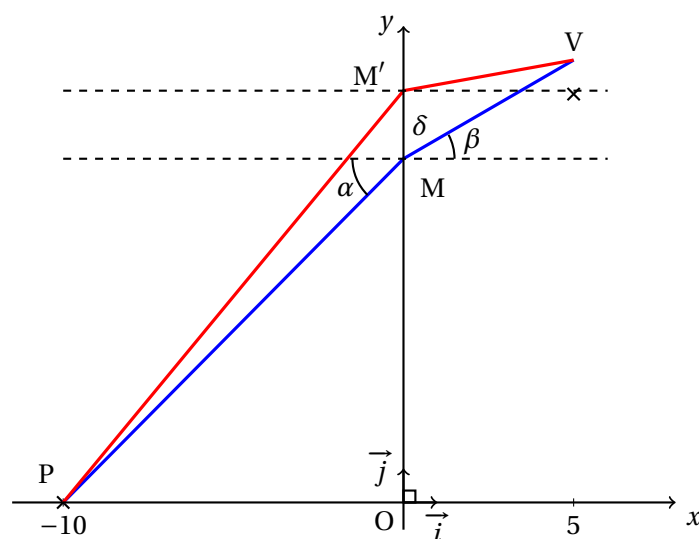


FIGURE 5 – Définition des angles α et β pour la conjecture

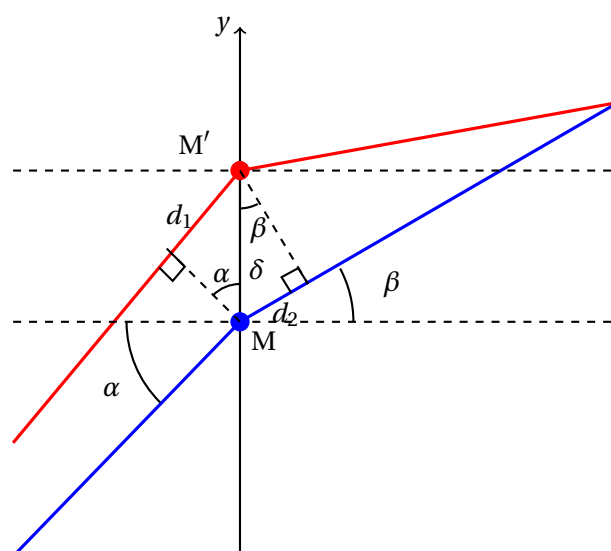


FIGURE 6 – Zoom local autour de M et M'

$$\sin(\alpha) = \frac{d_1}{\delta} \quad \text{et} \quad \sin(\beta) = \frac{d_2}{\delta}$$

Sachant que $d_1 = 2d_2$, nous en déduisons que :

$$\boxed{\sin(\alpha) = 2 \sin(\beta)}$$

Nous avons donc démontré que la position optimale du point M vérifie la relation $\sin(\alpha) = n \sin(\beta)$ avec $n = \frac{v_1}{v_2}$, ce qui correspond à la loi de la réfraction de Snell-Descartes.

5.3. Lien avec notre première conjecture

Pour comprendre pourquoi nous avons trouvé la conjecture $\alpha = n \times \beta$ au lieu de $\sin(\alpha) = n \sin(\beta)$ lors de nos premiers essais, nous avons tracé à la calculatrice les courbes des fonctions $\sin(x)$ et x (figure 7).

Nous constatons que pour des angles proches de 0° , les fonctions $\sin(x)$ et la fonction identité (définie par $f(x) = x$) sont très proches, ce qui explique que nous avons trouvé la relation $\alpha = n \times \beta$ pour

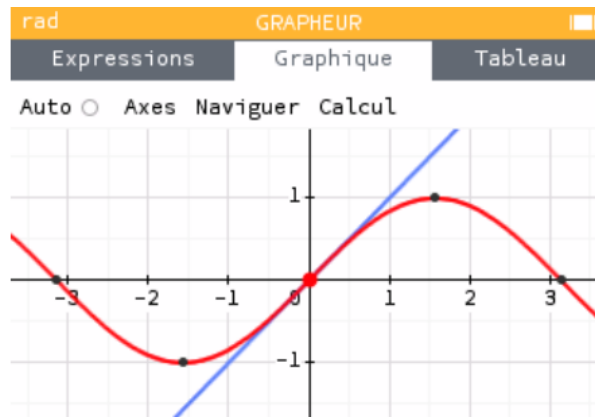


FIGURE 7 – Tracé des fonctions $\sin(x)$ et x

des angles relativement petits. Cependant, lorsque les angles deviennent plus grands, la différence entre les deux fonctions devient plus grande, ce qui explique pourquoi notre première conjecture ne fonctionnait pas dans tous les cas.

6. Conclusion

Ce problème de recherche de chemin optimal entre deux milieux de vitesses différentes nous a amenés à modéliser la situation numériquement puis mathématiquement. En utilisant la trigonométrie et l'étude d'une variation infiniment petite, nous avons pu retrouver la loi de la réfraction de Snell-Descartes appliquée ici à un problème de trajectoire.

7. Annexe

Code de tracé du graphique

```
from random import *
import turtle as tt

def graph(xP,yP,xS,yS,yR,v1,v2):

    xR=0
    temp1=0
    temp2=0
    temps={}
    liste=[]

    #scale permet un affichage sur une échelle plus grande du graphique
    scale=20
    y=0
    rep=0
    tt.penup()
    tt.color((0,0,0))
    tt.speed(0)
    tt.setup(800,600)
    tt.title("graphe de trajectoire optimale")
    tt.goto(-300,0)
    tt.setheading(0)
    tt.pendown()
    tt.forward(600)
    tt.penup()
```



```

tt.goto(0,300)
tt.setheading(-90)
tt.pendown()
tt.forward(600)
tt.penup()
tt.goto(-100,0)

a=calculer_temps(xP,yP,xS,yS,v1,v2)

for k,v in a.items():
    liste.append((k,v))

maxi=liste[0][1]
for k in range(len(liste)):
    if liste[k][1]> maxi:
        maxi=liste[k][1]

mini=liste[0][1]
for k in range(len(liste)):
    if liste[k][1]<mini:
        mini=liste[k][1]
        x_mini = liste[k][0]
print("le temps minimum est",mini,"s avec x =",x_mini)

while rep<=((yR)/0.1):
    coef=(liste[rep][1]-mini)/(maxi-mini)

    #choisis le couleur du stylo en fonction du temps total obtenu
    # /\: le temps peut varier en fonction des valeurs dans les paramètres c

    """if coef>=0.8:
        tt.color("red")
    elif coef >= 0.6:
        tt.color("orange")
    elif coef>=0.4:
        tt.color("yellow")
    elif coef >= 0.1:
        tt.color("green")
    elif coef>=0.01:
        tt.color("blue")
    elif coef==0:
        tt.color("cyan")"""

    R = round(coef*255)
    if coef != 0.0:
        G = round(-255*coef+255)
        B = round(255/255**coef)
        couleur = (R,G,B)
    else :
        couleur = (0,0,255)

    #print(couleur, coef)

    tt.colormode(255)
    tt.color(couleur)

    tt.pensize(3)

    # affichage du graphique

```

```

        tt.speed(0)
        tt.penup()
        tt.goto(xP * scale, yP * scale)

        tt.pendown()
        tt.goto(xR * scale, liste[rep][0] * scale)
        tt.penup()
        tt.goto(xR * scale, liste[rep][0] * scale)

        tt.pendown()
        tt.goto(xS * scale, yS * scale)
        rep+=1
        # trace un graphique non gradué avec les axes x et y
    tt.penup()
    tt.pensize(1)
    tt.color((0,0,0))
    tt.speed(0)
    tt.setup(800,600)
    tt.title("graphe de trajectoire optimale")
    tt.goto(-300,0)
    tt.setheading(0)
    tt.pendown()
    tt.forward(600)
    tt.penup()
    tt.goto(0,300)
    tt.setheading(-90)
    tt.pendown()
    tt.forward(600)
    tt.penup()
    tt.goto(-100,0)

    return tt.done()

xP=-10
yP=0
xS=5
yS=12
yR=yS
v1=2
v2=1
print(xP,yP,xS,yS,yR,v1,v2)
graph(xP,yP,xS,yS,yR,v1,v2)

```

Notes d'édition

(1) Dans le programme, v_1 (respectivement v_2) est la vitesse dans le milieu herbeux (respectivement boueux).

(2) À propos de la figure 6, les deux angles codés « droits » ne sont en réalité pas droits (H' , le projeté orthogonal de M' sur (MV) et H'' sur (MV) tel que $M'VH''$ est isocèle en V ne peuvent être confondus), mais δ étant petit, ces angles sont proches d'être droits et considérés comme tels.