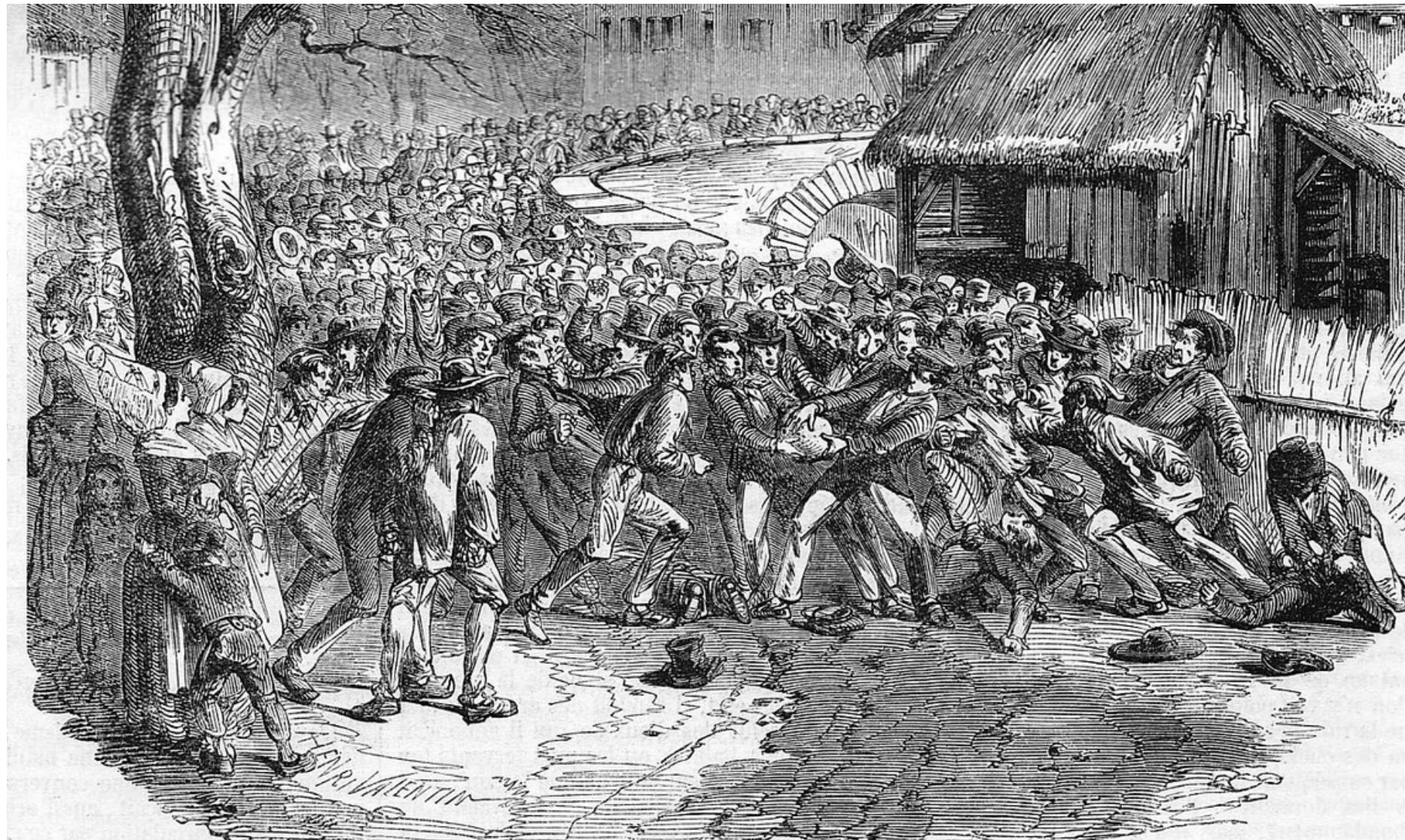


# Qu'est-ce que le soccer?



- Remonte à l'antiquité
- Chuju et la choule
- Football moderne apparaît 1863

# Coupe Du Monde 2014

3.572 milliards de  
téléspectateurs  
en moyenne



sky SPORTS



sky SPORTS



# Comment est-ce possible?

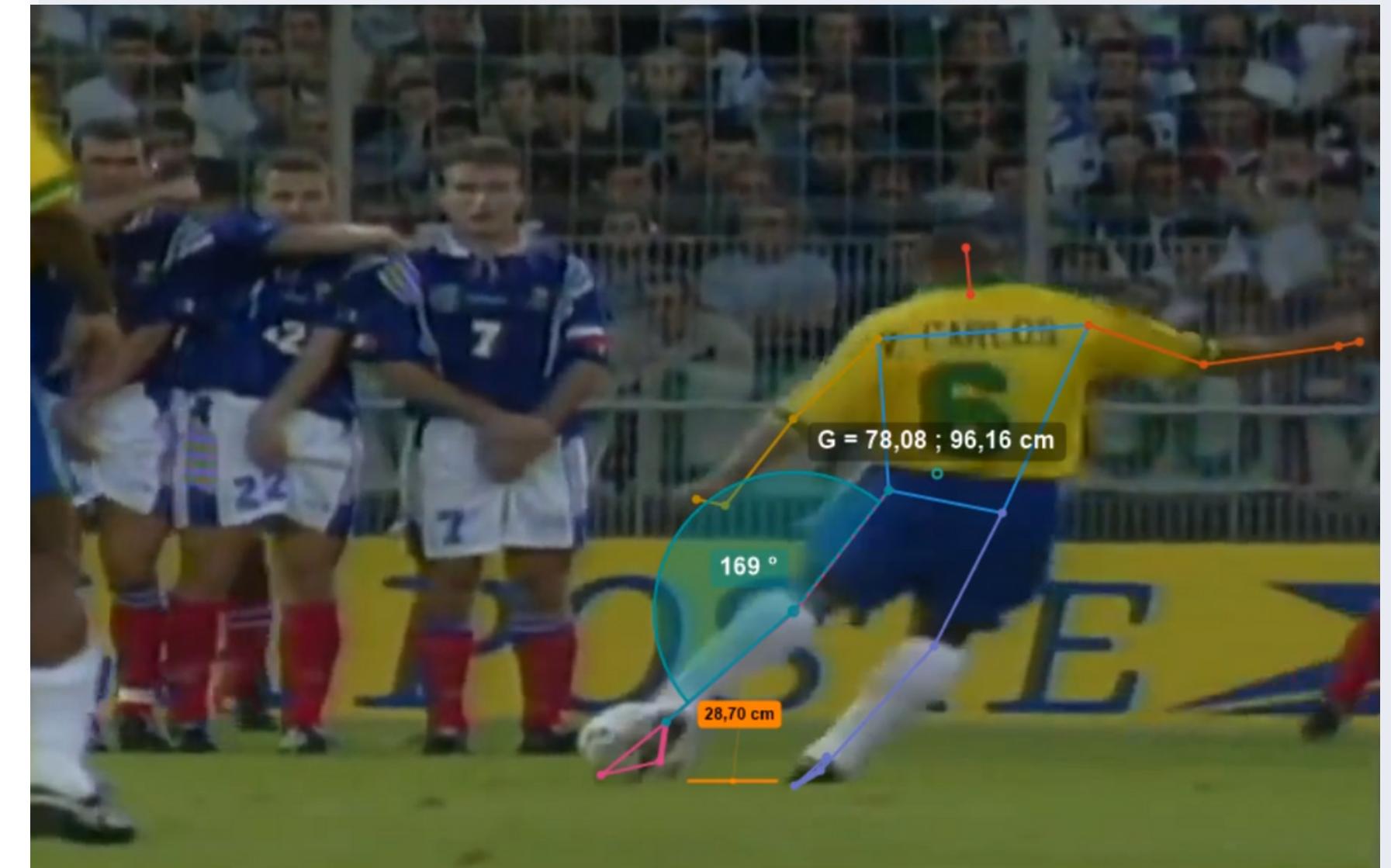
Roberto Carlos coup franc 1997 vs France

# Plan de présentation

- Sujet
- Méthodes
- Biomécanique
- Cinématique
- Résultats
- Conclusion

# Méthodes

- Analyse de la trajectoire
- Analyse de la biomécanique



# Biomécanique

## Kinovea

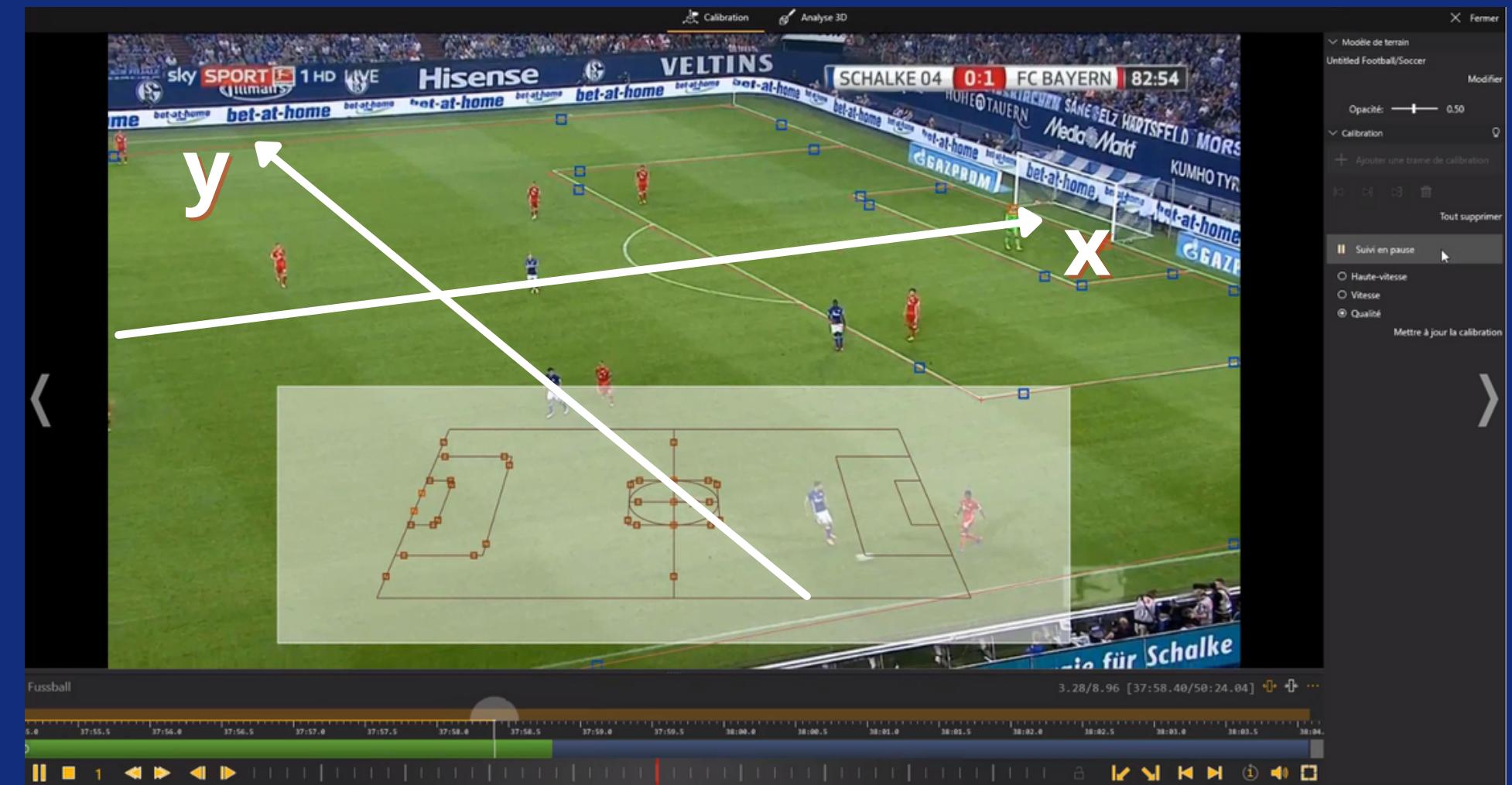
- Vecteur biomécanique
- Point de stabilité



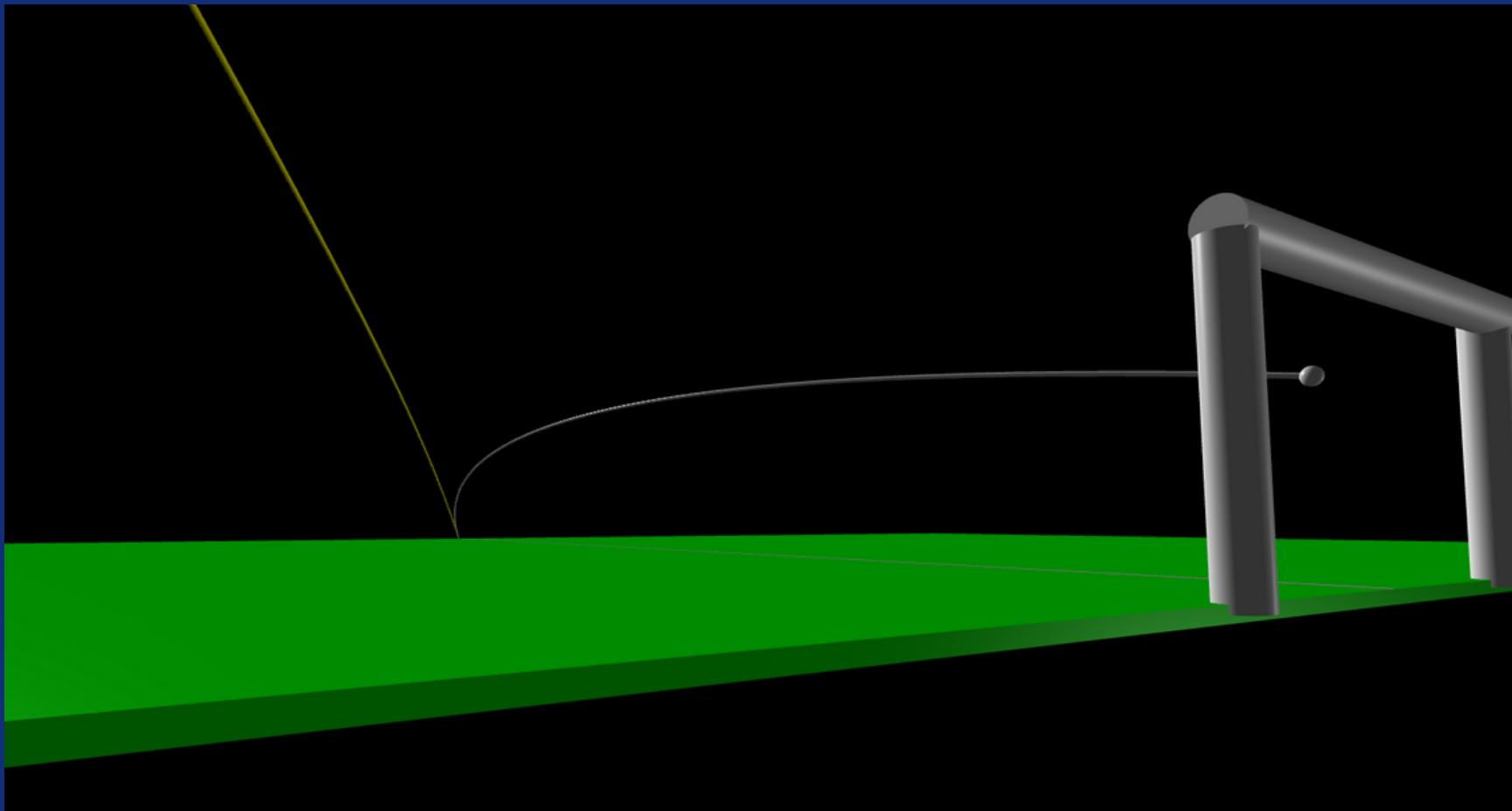
# Trajectoire

## Dartfish

- Modélisation 3D
- Traqueur de ballon automatique



# Trajectoire

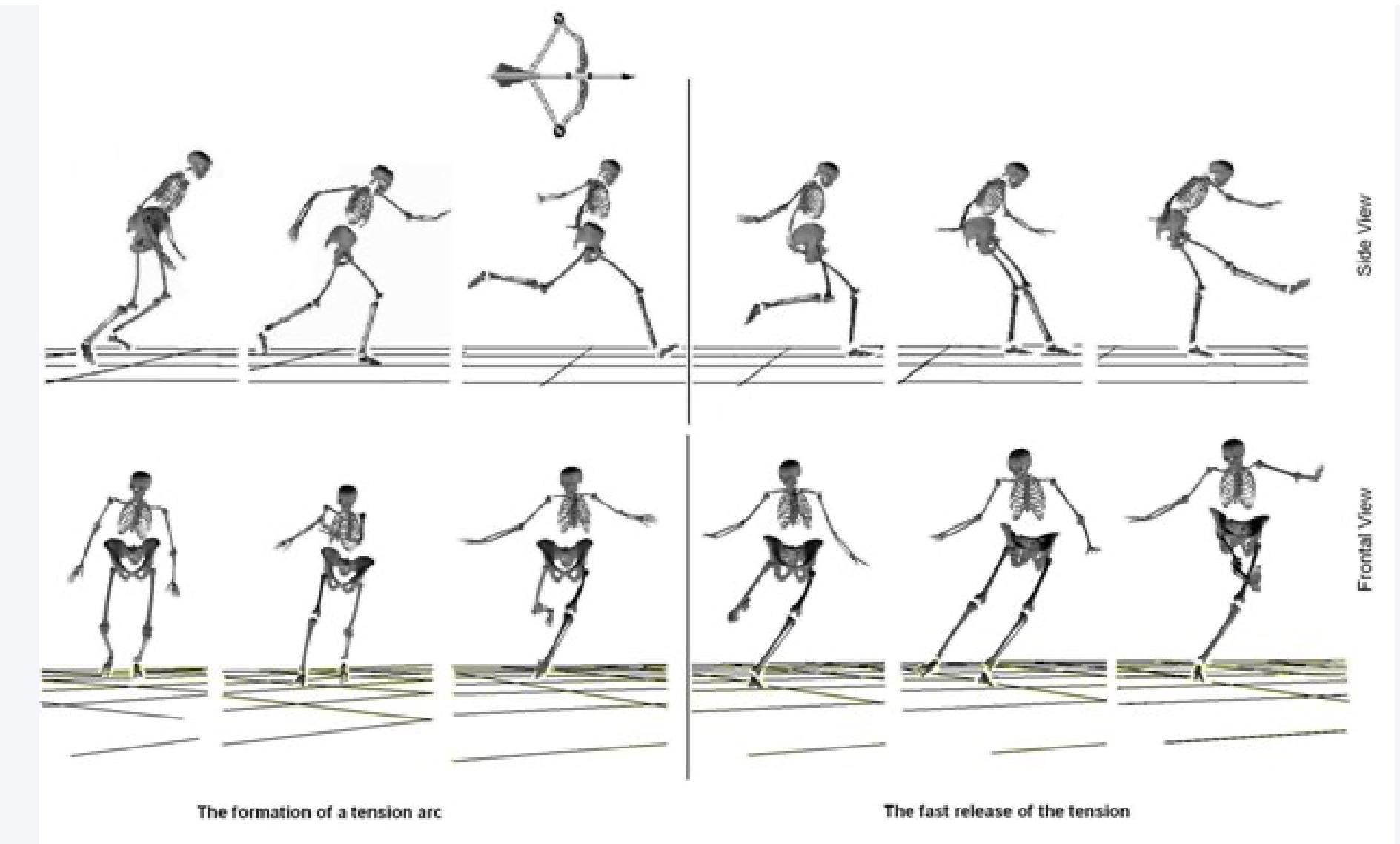


## VPython

- Calcul numérique
- Graphiques
- Simulation 3D

# Concepts biomécaniques

- L'approche
- L'ajustement
- L'armement
- Le contact du pied
- Le suivi



# L'approche

- Angle d'approche
- Distance du ballon
- Impulsions talon-pointe des pieds
- Dernier pas



# L'ajustement



- Distance pied-ballon
- Position antérieur-postérieur du pied
- Genou fléchi
- Rotation du bassin

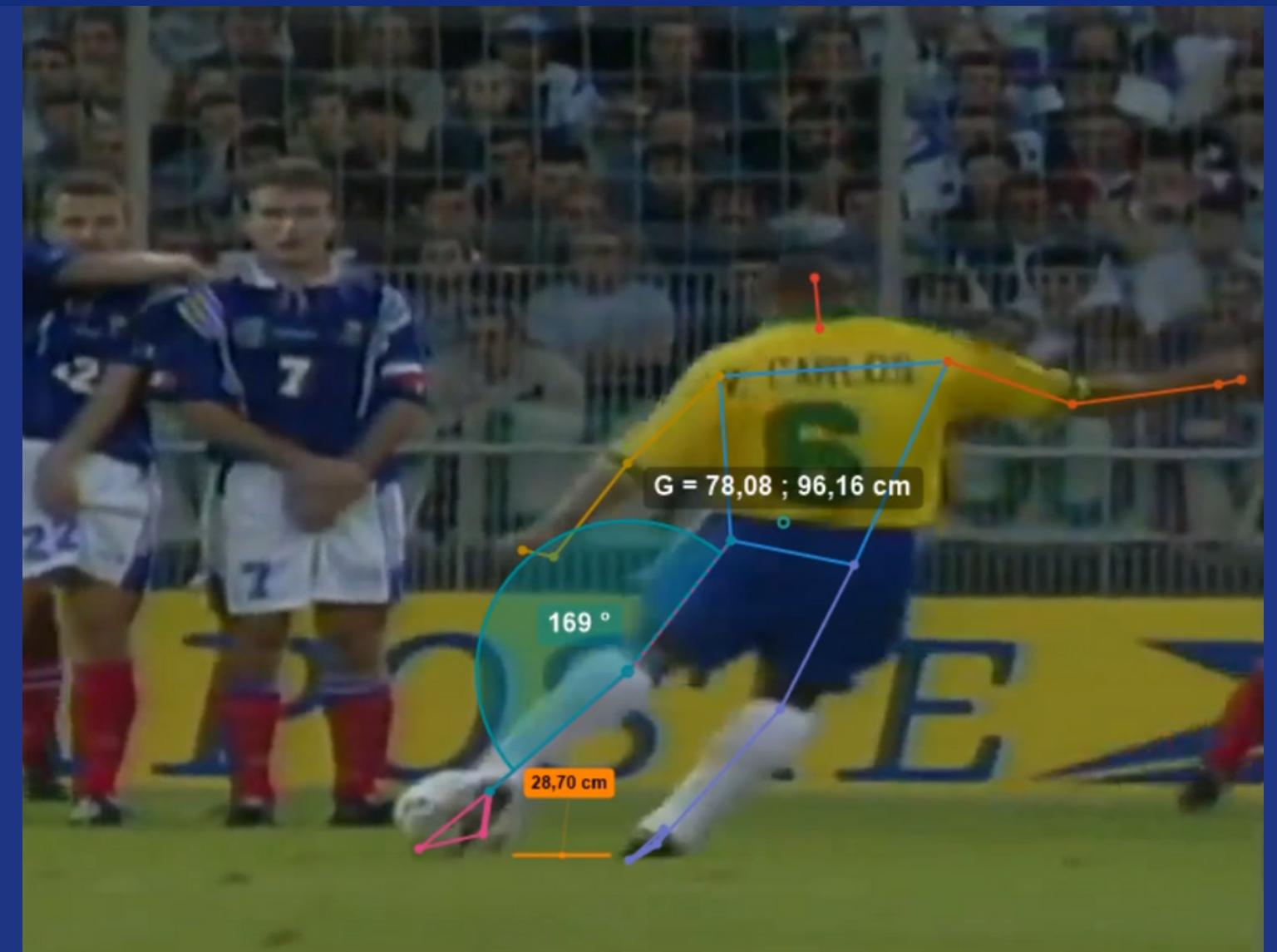
# L'armement

- Placement de la tête
- Flexion du genou
- Basculement de la jambe



# Contact du pied

- Zone de frappe
- Force appliquée et énergie transmise
- Muscles recrutés



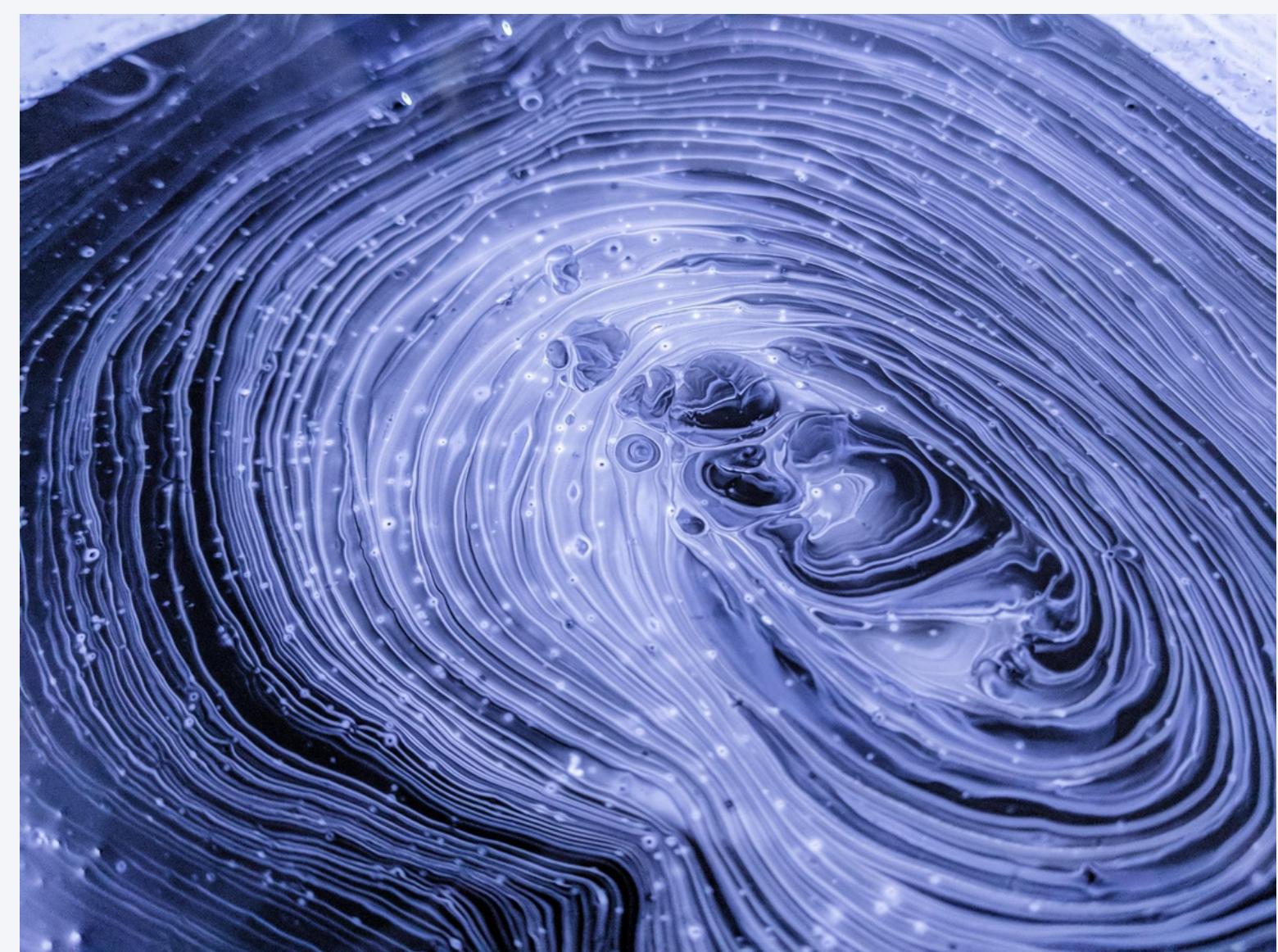
# Suivi



- Déplacement de poids
- Suivi de la jambe
- Détermine la trajectoire

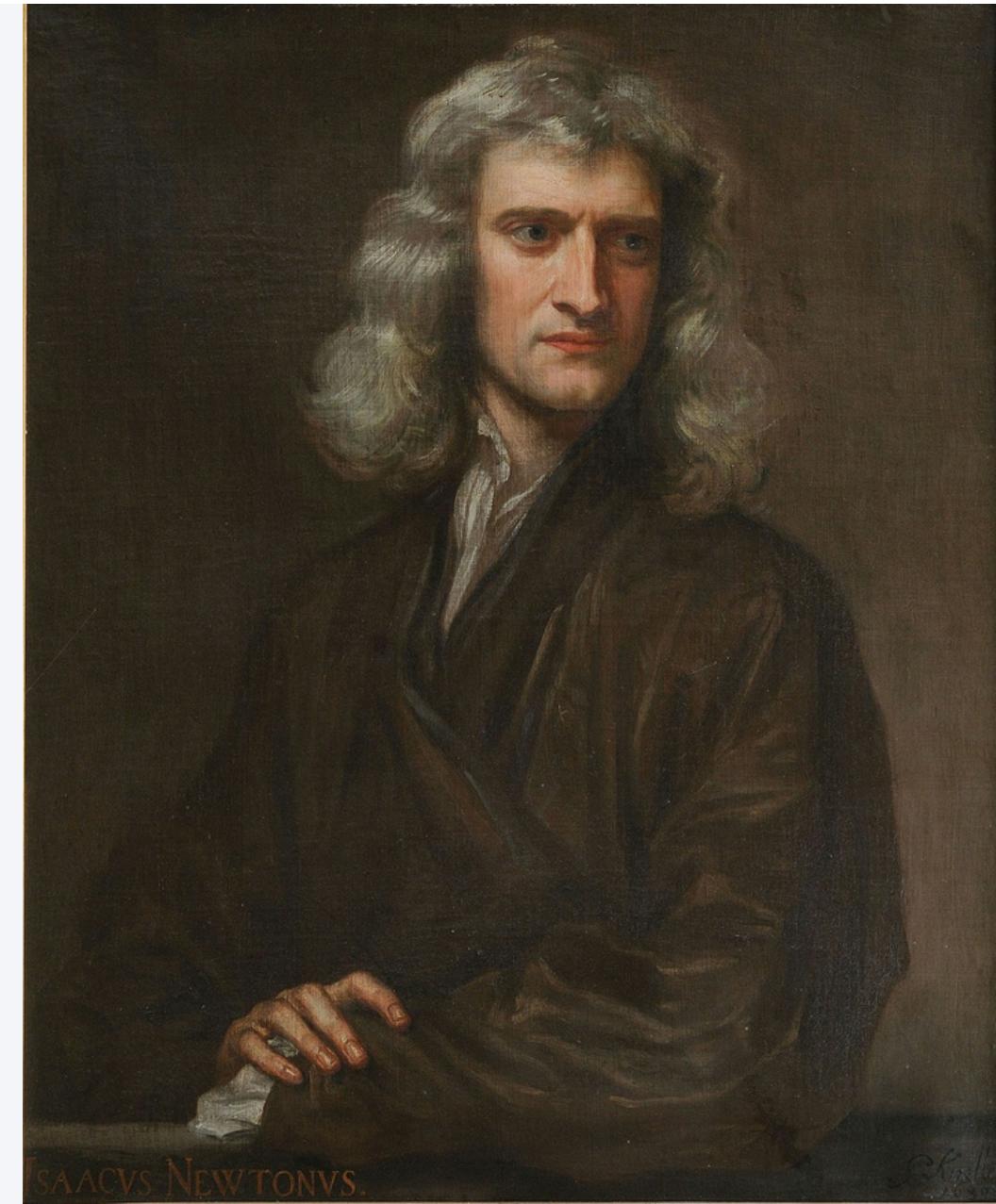
# Cinématique

- Lois de Newton
- Types de mouvements
- Trainée
- Effet Magnus

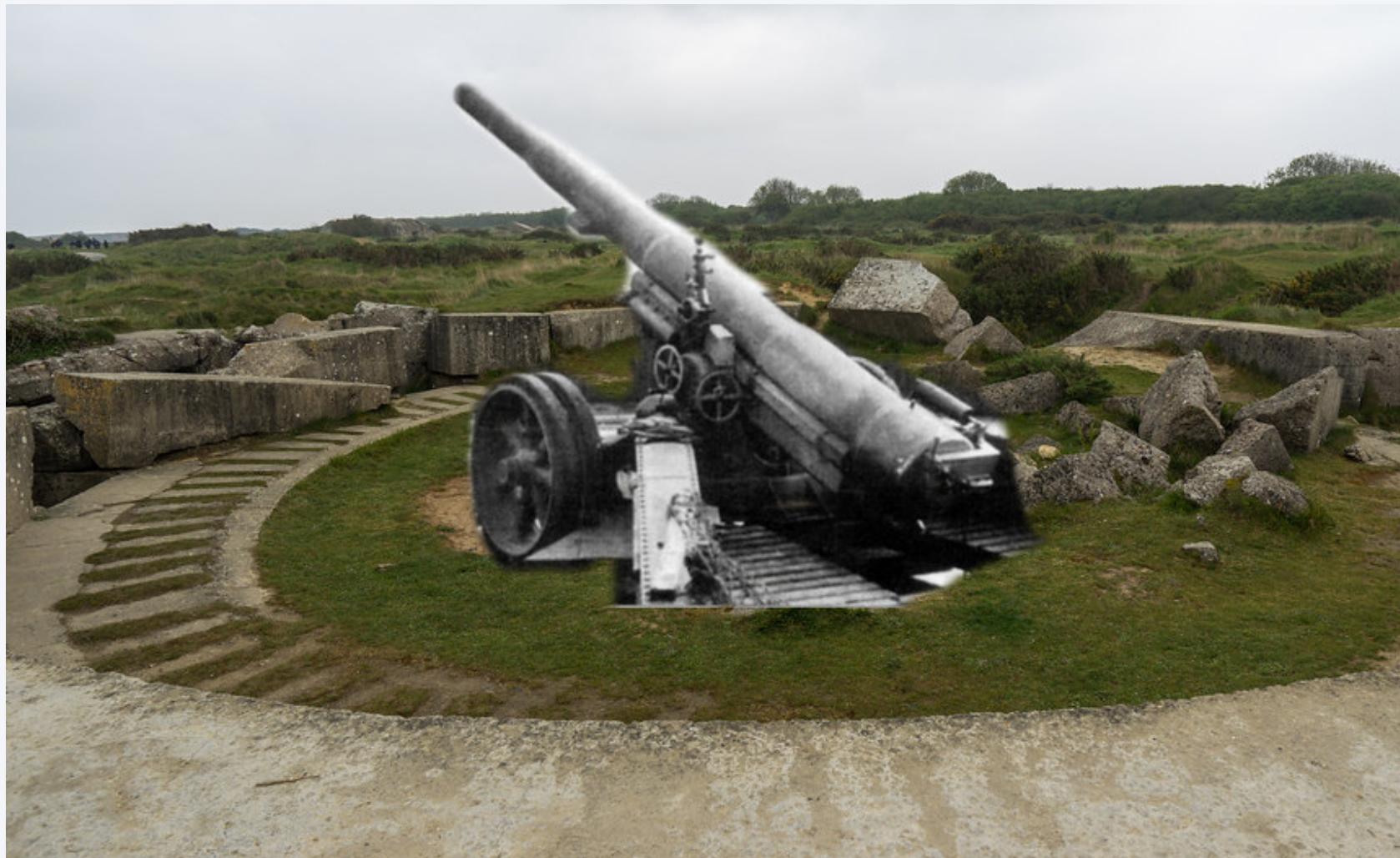


# Lois de Newton

- Inertie
- $F = ma$
- Action-réaction



# Cinématique



- MRU : Pas d'accélération
- MRUA : Accélération constante
- Projectile : MRU horizontal  
MRUA vertical

# Cinématique

$$a = \frac{v_f - v_i}{\Delta t} \quad v_f = v_i + a\Delta t$$

---

$$\Delta x = v_i \Delta t + \frac{a \Delta t^2}{2}$$

---

$$\Delta x = \frac{v_f^2 - v_i^2}{2a} \quad v_f^2 = v_i^2 + 2a\Delta x$$

---

$$\Delta x = \frac{(v_f + v_i)\Delta t}{2}$$

---

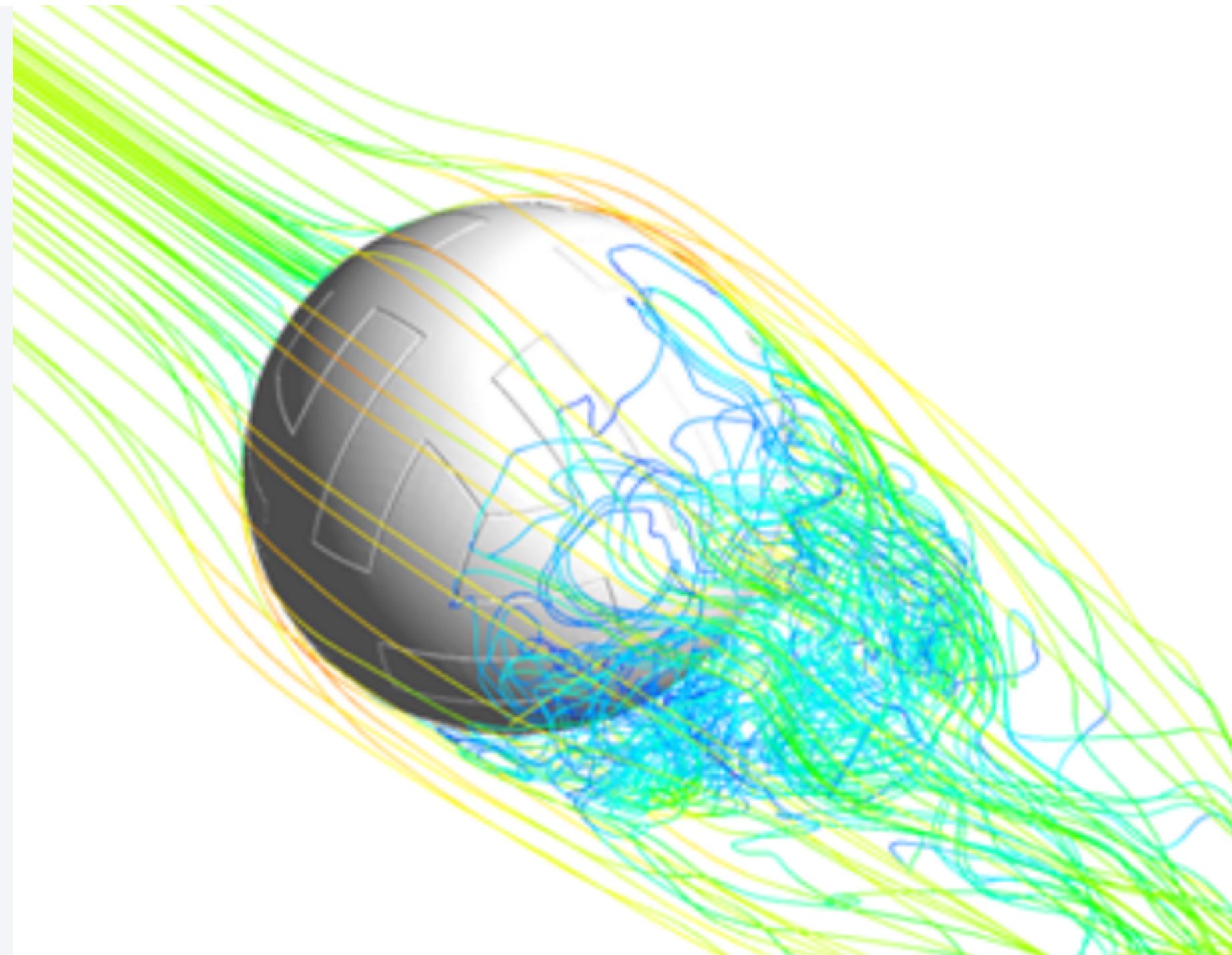
## Équations du MRUA

où

- a : accélération
- vi : vitesse initiale
- vf : vitesse finale
- Δx : déplacement
- Δt : le délai de temps

# La trainée

- S'oppose au mouvement
- Se produit entre un fluide et un objet
- Dépendante de la vitesse



# La trainée

$$F_P = \frac{1}{2} C_L \rho A v^2$$

## Équation

où

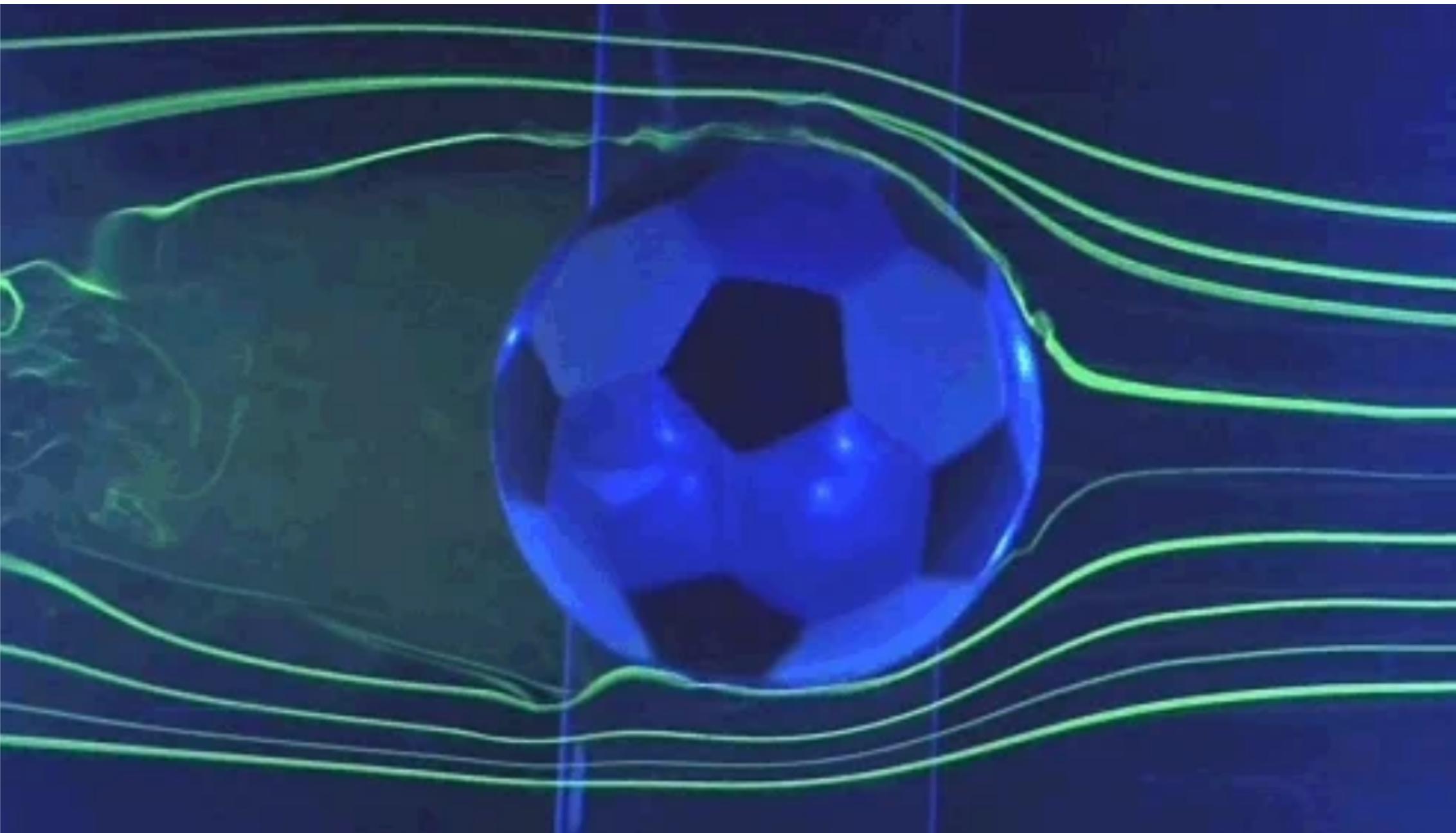
- CL : coefficient de trainée
- p : densité du fluide
- A : section transversale de l'objet
- v : vitesse de l'objet

# Effet Magnus

- Rotation dans un fluide
- Changements de pression
- Déviation de trajectoire



# Effet Magnus



# Effet Magnus

$$\overrightarrow{F_M} = s(\hat{w} \times \hat{v})$$

## Équation

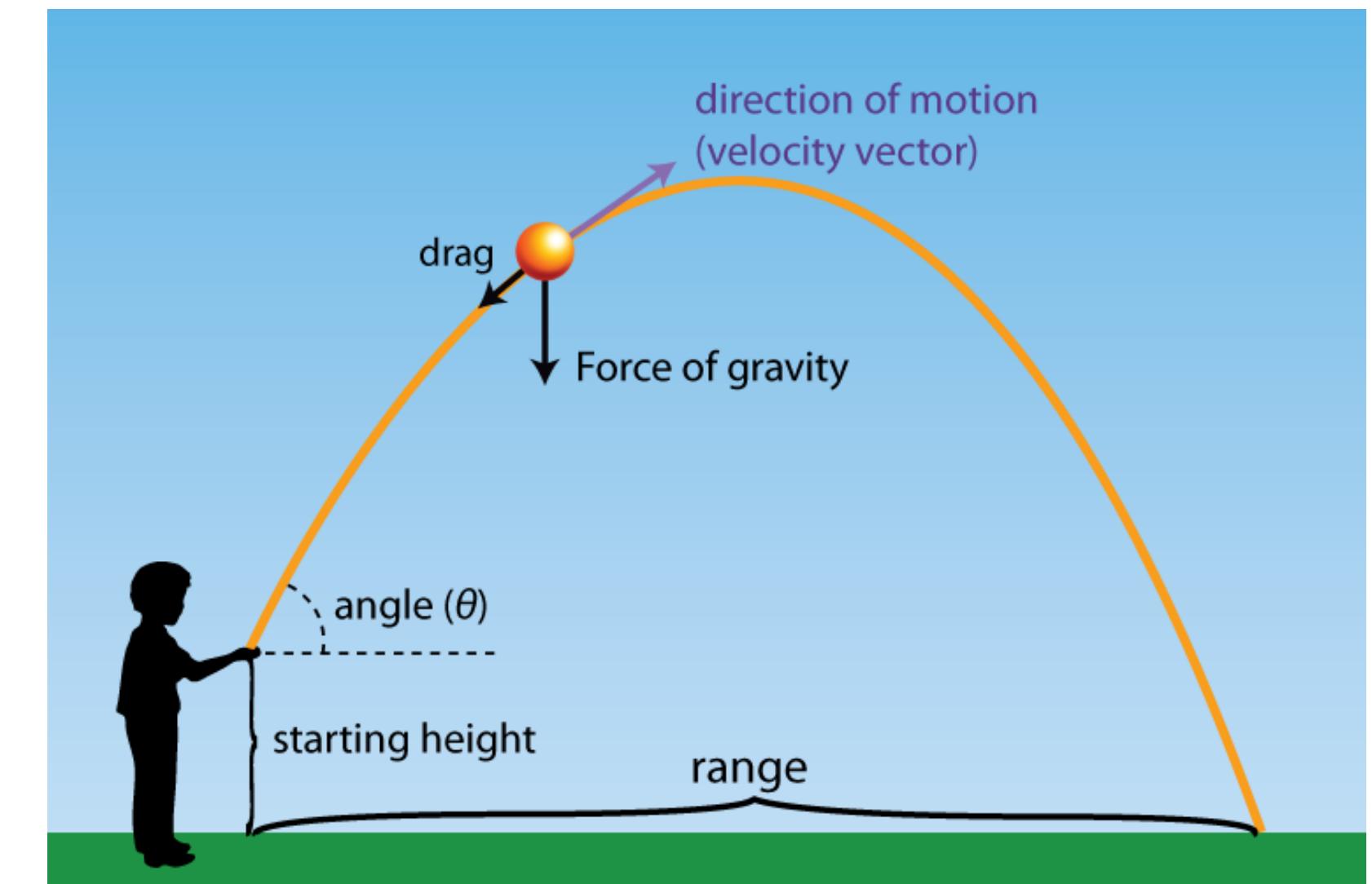
où

- $s$  : Coefficient de résistance de l'air
- $w$  : vitesse angulaire
- $v$  : vitesse

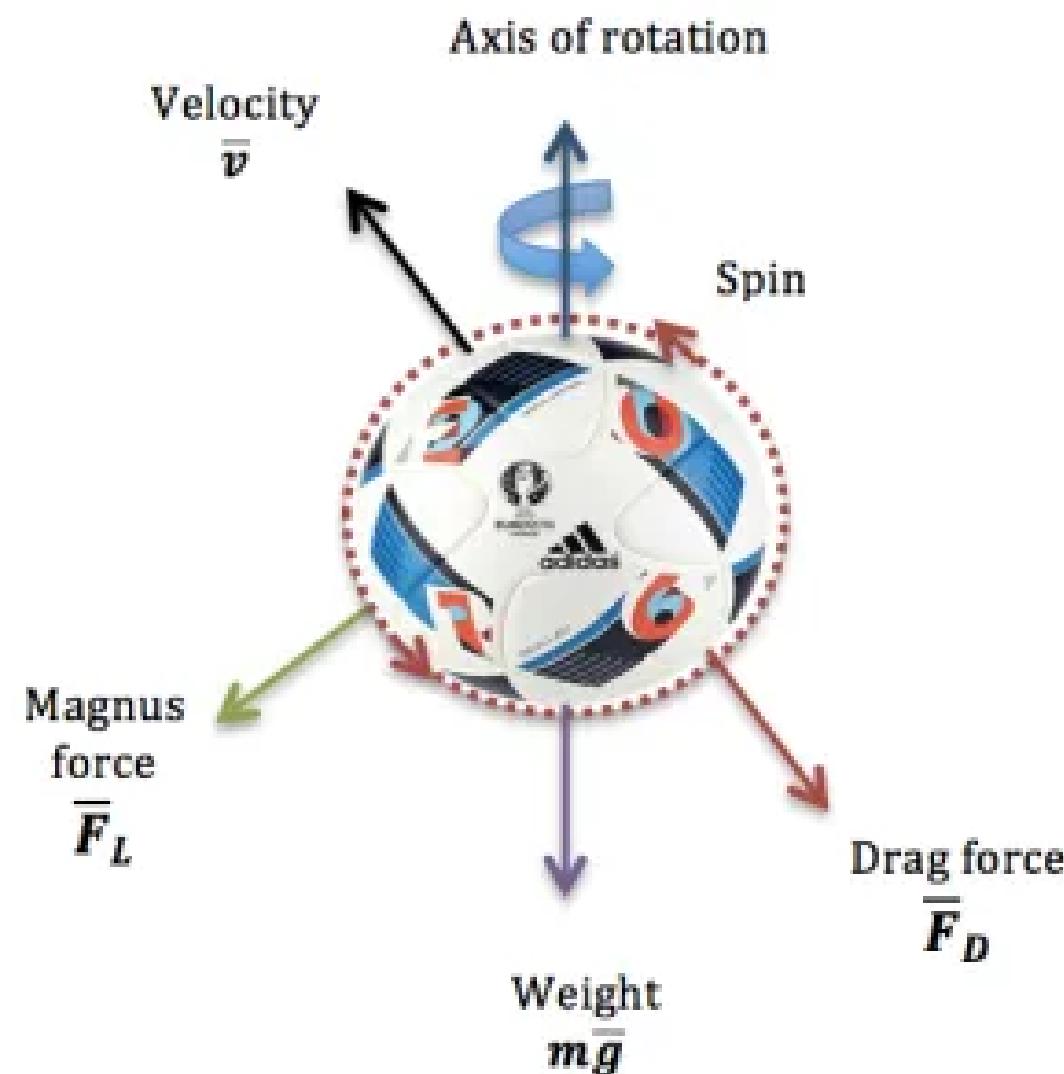
# Analyse dynamique

Balle dans l'air sans rotation de la balle :

- Frottement de l'air
- Présence de gravité
- Trajectoire parabolique constante



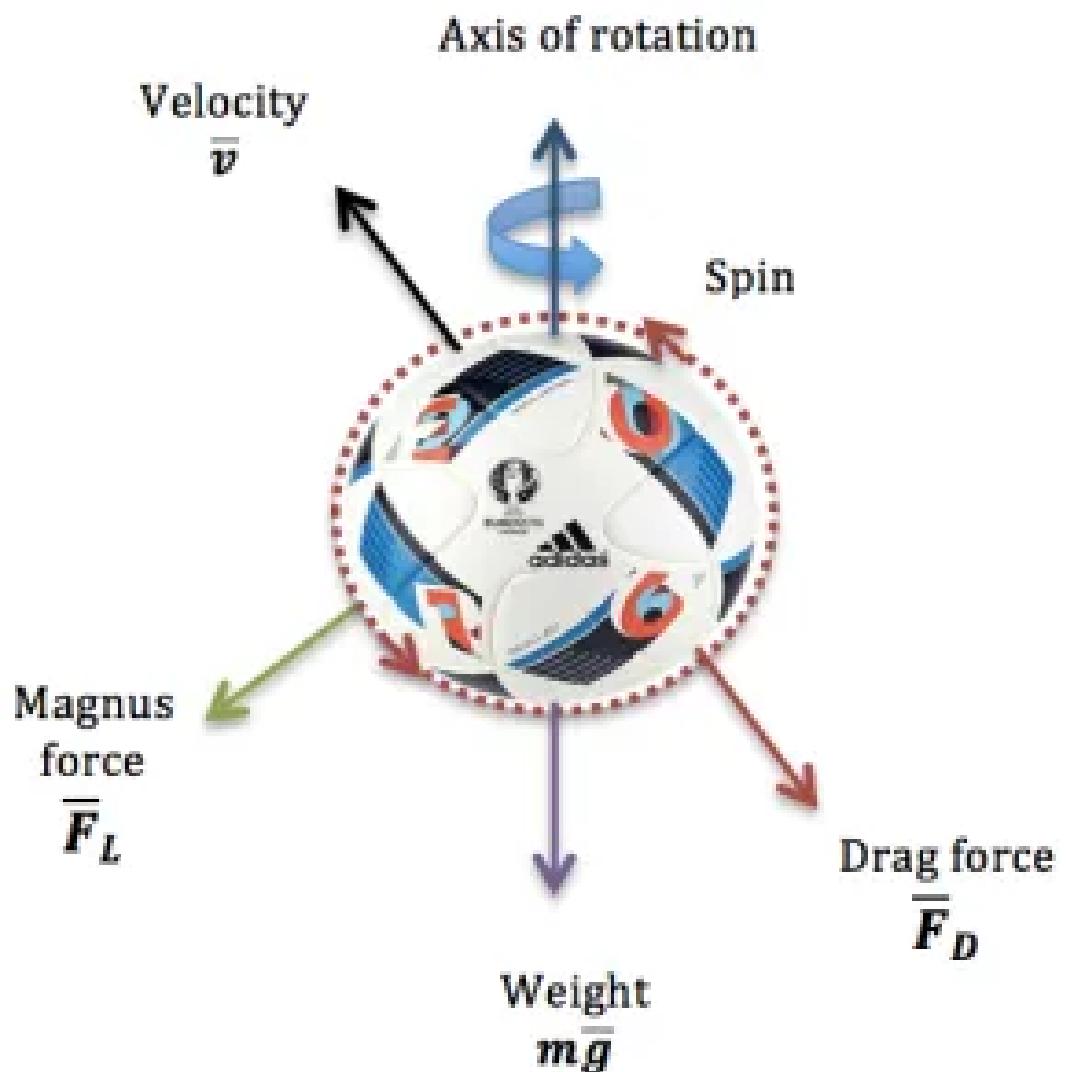
# Analyse dynamique



Balle dans l'air avec effet de rotation de la balle :

- Force Magnus
- Force de trainée
- Force gravitationnelle

# Analyse dynamique



$$\overrightarrow{ma} = -\overrightarrow{mg} + \overrightarrow{FP} + \overrightarrow{FM}$$

# Simulation 3D VPython

- Calcul numérique
- Éviter les systèmes d'équations complexes
- Mettre en pratique la théorie
- Visualiser trajectoire

```
36 #mass of the soccer ball
37 ball.m= .43
38
39 #Angular velocity of ball
40 ball.omega=vector(-25,88,0)
41
42 ball2=sphere(pos=ball.pos, radius=ball.radius, color=color.yellow, make_trail=True)
43
44 #launch speed in m/s
45 v0=44
46 #launch angle
47 theta = 15*pi/180
48
49 #initial velocity vector
50 ball.v=v0*vector(.40,sin(theta),-cos(theta))
51 #initial momentum vector
52 ball.p=ball.m*ball.v
53
54 ball2.p=ball.p
55 ball2.m=ball.m
56
57 rho=1.02 #density of air
58 C=.15 #the drag coefficient for a sphere
59 A = pi*ball.radius**2
60 s=.0033 #this is a magnus force constant
61
62 t=0
63 dt=0.001
64
65 while ball.pos.z>=-18.5:
66     rate(1000)
67     #scene.camera.follow(ball)
68     #calculate the velocity- it makes it easier to calc air drag
69     ball.v=ball.p/ball.m
70     #calculate the force
71     #note that to square velocity, must first find magnitude
72     #in order to make it a vector, I multiply by unit vector for v
73     F=ball.m*g-.5*rho*C*norm(ball.v)*mag(ball.v)**2+s*cross(ball.omega,ball.v)
74     F2=ball2.m*g
75     #update the momentum
76     ball.p=ball.p+F*dt
77     ball2.p=ball2.p+F2*dt
78     #update the position
79     ball.pos=ball.pos+ball.p*dt/ball.m
80     ball2.pos=ball2.pos+ball2.p*dt/ball2.m
81     #update the time
82     t=t+dt
83     fA.plot(t, ball.pos.z)
84     fB.plot(t, ball.pos.x)
85     fc.plot(t, ball.pos.y)
86     gV1.plot(t, mag(ball.v))
87     gVx.plot(t, abs(ball.v.z))
88     gVy.plot(t, ball.v.y)
89     gVz.plot(t, ball.v.x)
90     nos1.plot(ball.nos.z, ball.nos.v)
```

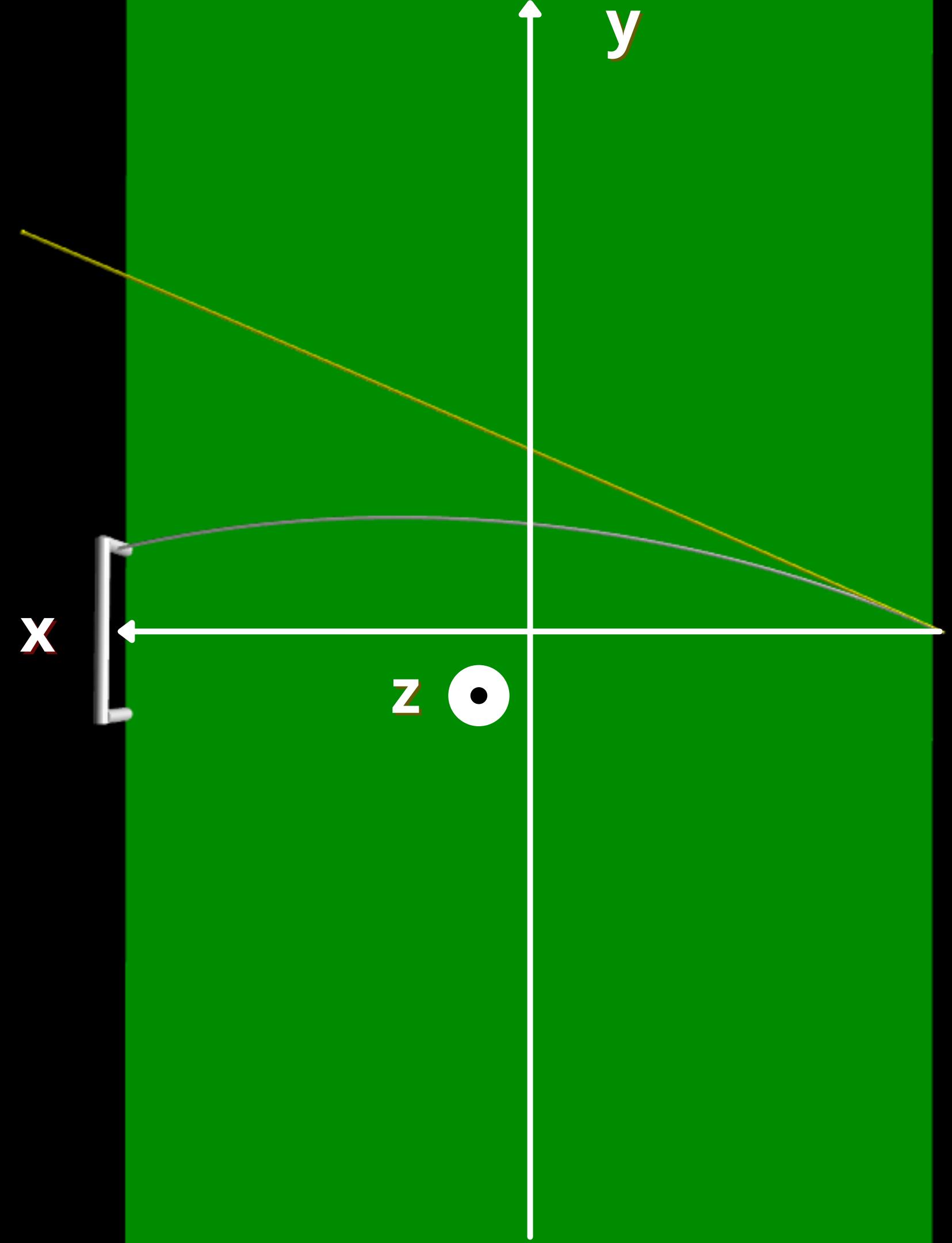
# **calcule la force résultante sur ballon**

**actualise la  
quantité de  
mouvement**

**passe au  
prochain  
temps**

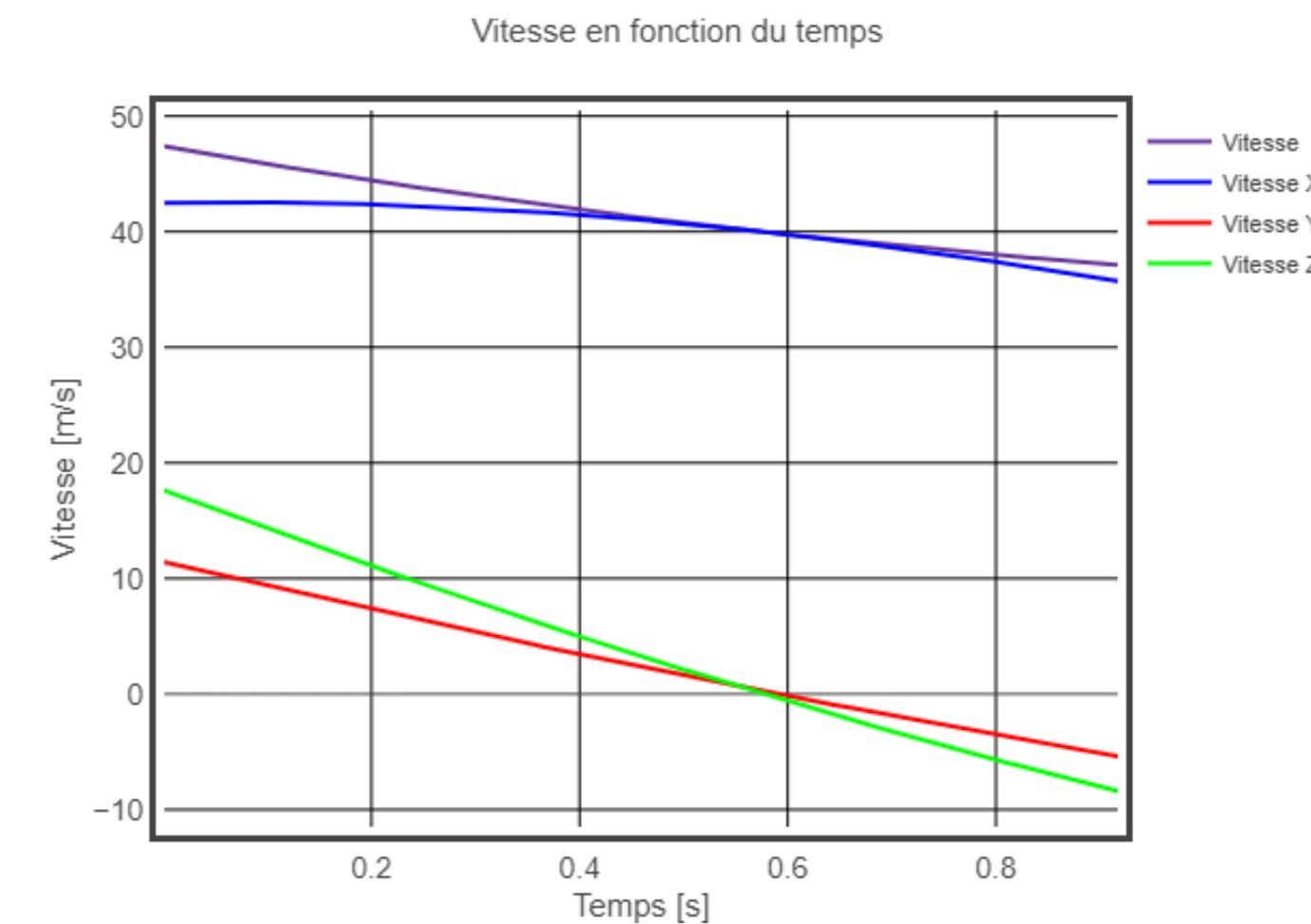
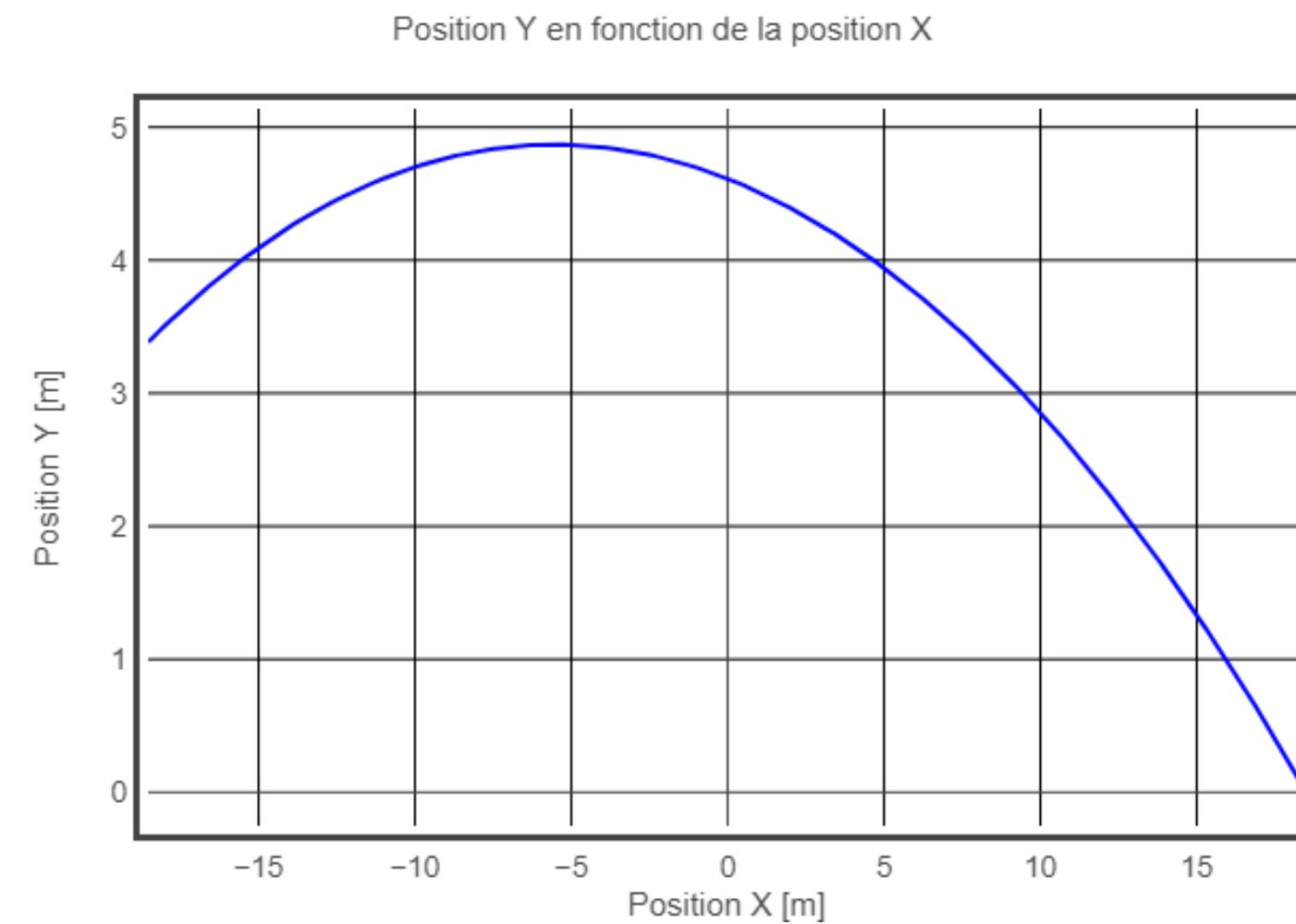
```
F=ball.m*g-.5*rho*A*C*norm(ball.v)*mag(ball.v)**2+s*cross(ball.omega,ball.v)
F2=ball2.m*g
#update the momentum
ball.p=ball.p+F*dt
ball2.p=ball2.p+F2*dt
#update the position
ball.pos=ball.pos+ball.p*dt/ball.m
ball2.pos=ball2.pos+ball2.p*dt/ball2.m
#update the time
t=t+dt
```

**actualise la  
position**





# Résultats



# Résultats en vidéo





# Conclusion

- Mouvement biomécanique important
- Force de trainée, force de Magnus primordiale
- Chaque étape constitue l'ensemble



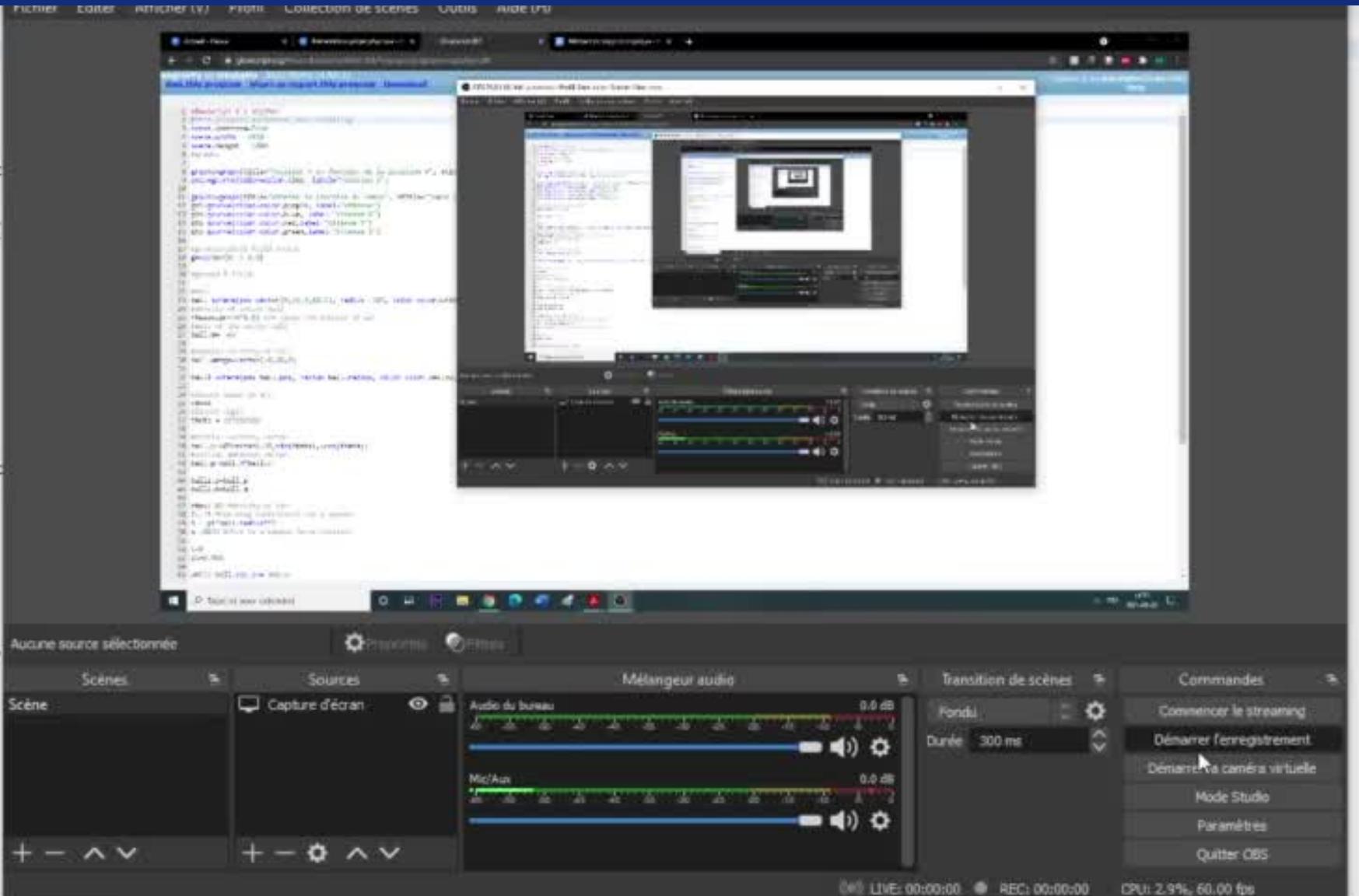
# Que se passerait-il sans obstacles?

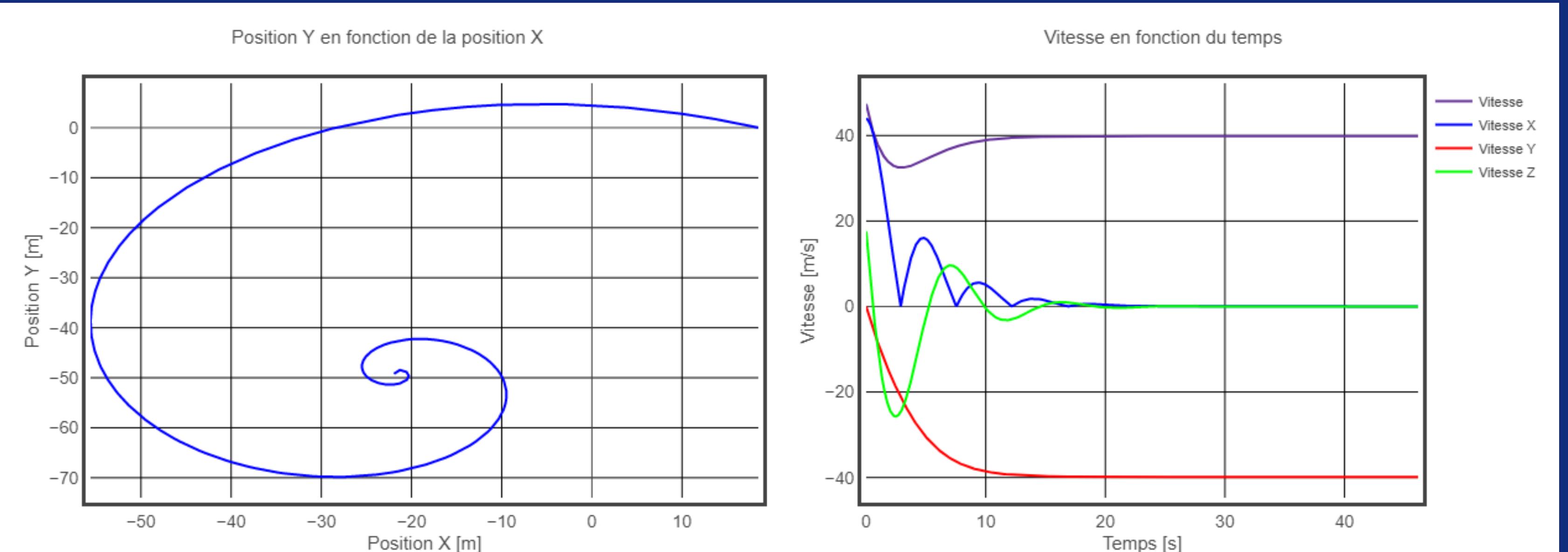
Trajectoire en spirale?

```

1 GlowScript 3.1 VPython
2 #this prevents accidental user scrolling
3 scene.userzoom=True
4 scene.width = 1920
5 scene.height = 1080
6 #graphs
7
8 graph2=graph(title="Position Y en fonction de la position X", xtitle="Position X")
9 pos1=gcurve(color=color.blue, label="Position X")
10
11 graph3=graph(title="Vitesse en fonction du temps", xtitle="Temps [s]")
12 gV1=gcurve(color=color.purple, label="Vitesse")
13 gVx=gcurve(color=color.blue, label="Vitesse X")
14 gVy=gcurve(color=color.red,label="Vitesse Y")
15 gVz=gcurve(color=color.green,label="Vitesse Z")
16
17 #gravitational field vector
18 g=vector(0,-9.8,0)
19
20 #ground & field
21
22 #ball
23 ball=sphere(pos=vector(0,-1.4,18.5), radius=.105, color=color.white)
24 #density of soccer ball
25 rhoSoccer=74*1.02 #74 times the density of air
26 #mass of the soccer ball
27 ball.m=.43
28
29 #Angular velocity of ball
30 ball.omega=vector(-0,88,0)
31
32 ball2=sphere(pos=ball.pos, radius=ball.radius, color=color.yellow, alpha=0.5)
33
34 #launch speed in m/s
35 v0=44
36 #launch angle
37 theta = 15*pi/180
38
39 #initial velocity vector
40 ball.v=v0*vector(.40,sin(theta),-cos(theta))
41 #initial momentum vector
42 ball.p=ball.m*ball.v
43
44 ball2.p=ball.p
45 ball2.m=ball.m
46
47 rho=1.02 #density of air
48 C=.15 #the drag coefficient for a sphere
49 A = pi*ball.radius**2
50 s=.0033 #this is a magnus force constant
51
52 t=0
53 dt=.001
54
55 while ball.pos.z>=-300.5:

```





# Merci

Rayane Sahi et Adam Bekri

