

Chutes avec frottements quadratiques

1 Chute verticale avec frottements

Étude cinématique: $O\vec{M} = z\vec{e}_z$ donc $\vec{v} = \dot{z}\vec{e}_z$ et $\vec{a} = \ddot{z}\vec{e}_z$

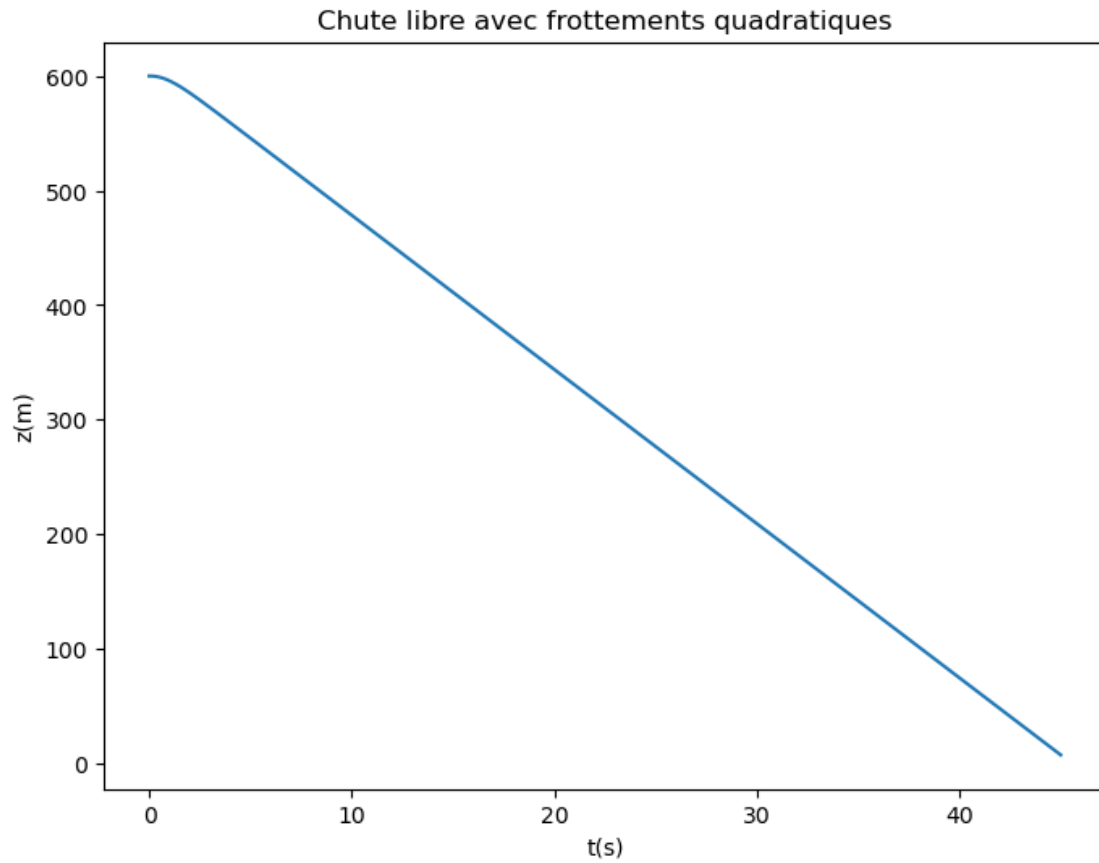
Forces:

- poids $-mg\vec{e}_z$
- frottements $-\lambda \cdot \|\vec{v}\| \cdot \vec{v}$ avec $\vec{v} = \dot{z}\vec{e}_z$ et $\|\vec{v}\| = \sqrt{\dot{z}^2} = |\dot{z}|$

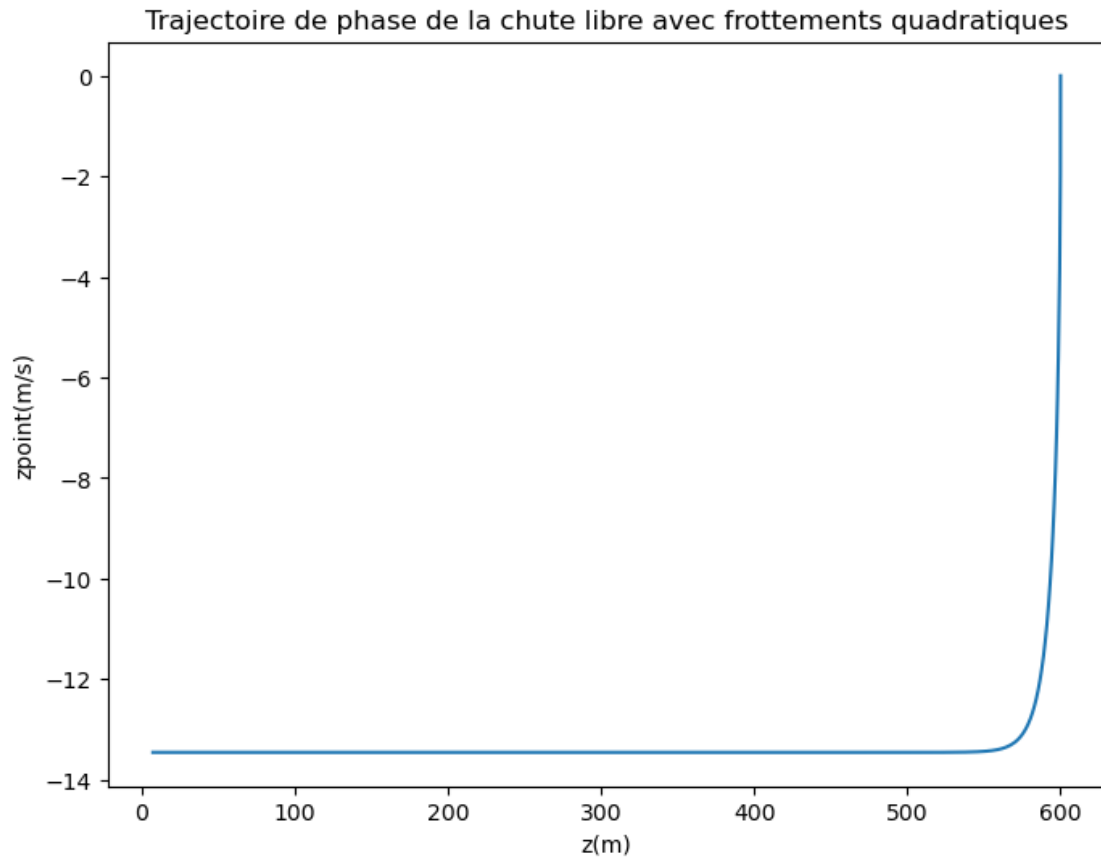
Le PFD: $m\ddot{z}\vec{e}_z = -mg\vec{e}_z - \lambda \cdot |\dot{z}| \cdot \dot{z}\vec{e}_z$ à projeter sur $\vec{e}_z$.

```
[11]: import numpy as np
m,lam,g=17e-3,9.2e-4,9.8
def acceleration(z,dzdt,t):
    return -g-lam*abs(dzdt)*dzdt/m
h=0.1
tmax=45
N=int(tmax/h)
tab_t=np.zeros(N+1)
tab_z=np.zeros(N+1)
tab_dzdt=np.zeros(N+1)
tab_t[0],tab_z[0],tab_dzdt[0]=0,600,0
for n in range(N):
    tab_t[n+1]=tab_t[n]+h
    tab_z[n+1]=tab_z[n]+h*tab_dzdt[n]
    tab_dzdt[n+1]=tab_dzdt[n]+h*acceleration(tab_z[n],tab_dzdt[n],tab_t[n])
```

```
[12]: import matplotlib.pyplot as plt
plt.figure(figsize=(8,6))
plt.plot(tab_t,tab_z)
plt.xlabel("t(s)")
plt.ylabel("z(m)")
plt.title("Chute libre avec frottements quadratiques")
plt.show()
```



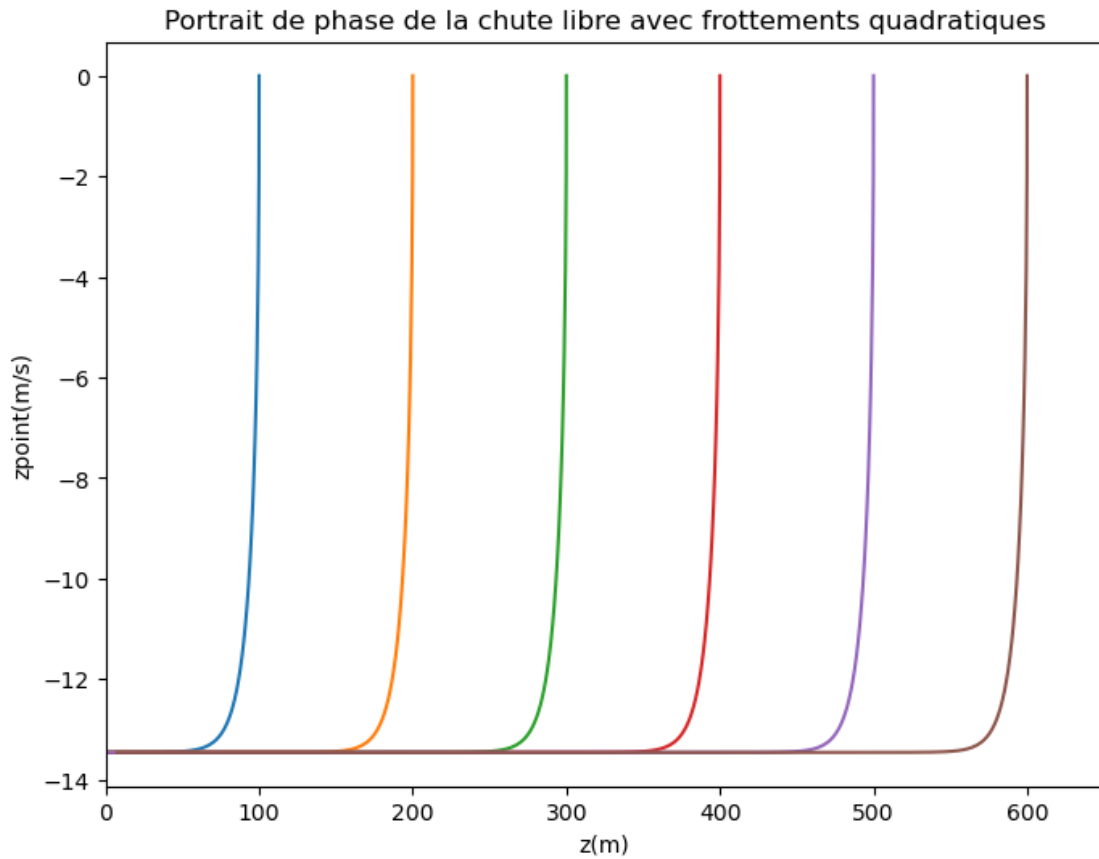
```
[13]: plt.figure(figsize=(8,6))
plt.plot(tab_z,tab_dzdt)
plt.xlabel("z(m)")
plt.ylabel("zpoint(m/s)")
plt.title("Trajectoire de phase de la chute libre avec frottements_
↪quadratiques")
plt.show()
```



Vitesse limite: $13,4 \text{ m/s} = 48 \text{ km/h}$

```
[14]: def trace_tph(h0,v0):
    h=0.1
    tmax=45
    N=int(tmax/h)
    tab_t=np.zeros(N+1)
    tab_z=np.zeros(N+1)
    tab_dzdt=np.zeros(N+1)
    tab_t[0],tab_z[0],tab_dzdt[0]=0,h0,v0
    for n in range(N):
        tab_t[n+1]=tab_t[n]+h
        tab_z[n+1]=tab_z[n]+h*tab_dzdt[n]
        tab_dzdt[n+1]=tab_dzdt[n]+h*acceleration(tab_z[n],tab_dzdt[n],tab_t[n])
    plt.plot(tab_z,tab_dzdt)
plt.figure(figsize=(8,6))
for h0 in range(100,601,100):
    trace_tph(h0,0)
plt.xlabel("z(m)")
plt.ylabel("zpoint(m/s)")
```

```
plt.xlim([0,650])
plt.title("Portrait de phase de la chute libre avec frottements quadratiques")
plt.show()
```



2 Tir balistique avec frottements

Étude cinématique: $\vec{OM} = x\vec{e}_x + z\vec{e}_z$ donc $\vec{v} = \dot{x}\vec{e}_x + \dot{z}\vec{e}_z$ et $\vec{a} = \ddot{x}\vec{e}_x + \ddot{z}\vec{e}_z$

Forces:

- poids $-mg\vec{e}_z$
- frottements $-\lambda \cdot \|\vec{v}\| \cdot \vec{v}$ avec $\vec{v} = \dot{x}\vec{e}_x + \dot{z}\vec{e}_z$ et $\|\vec{v}\| = \sqrt{\dot{x}^2 + \dot{z}^2}$

Le PFD: $m\ddot{x}\vec{e}_x + m\ddot{z}\vec{e}_z = -mg\vec{e}_z - \lambda \cdot \sqrt{\dot{x}^2 + \dot{z}^2} \cdot (\dot{x}\vec{e}_x + \dot{z}\vec{e}_z)$ à projeter sur les deux axes.

```
[6]: import numpy as np
m,lam,g=7,3e-3,9.8
v0,alpha=1500,40*np.pi/180
def acceleration(x,z,dxdt,dzdt,t):
    return -lam*np.sqrt(dxdt**2+dzdt**2)*dxdt/m
```

```

def accelerationz(x,z,dxdt,dzdt,t):
    return -g-lam*np.sqrt(dxdt**2+dzdt**2)*dzdt/m
h=0.01
tmax=48
N=int(tmax/h)
tab_t=np.zeros(N+1)
tab_x=np.zeros(N+1)
tab_z=np.zeros(N+1)
tab_dxdt=np.zeros(N+1)
tab_dzdt=np.zeros(N+1)
tab_dxdt[0],tab_dzdt[0]=v0*np.cos(alpha),v0*np.sin(alpha)
for n in range(N):
    tab_t[n+1]=tab_t[n]+h
    tab_x[n+1]=tab_x[n]+h*tab_dxdt[n]
    tab_z[n+1]=tab_z[n]+h*tab_dzdt[n]

    ↪ tab_dxdt[n+1]=tab_dxdt[n]+h*accelerationx(tab_x[n],tab_z[n],tab_dxdt[n],tab_dzdt[n],tab_t[n])

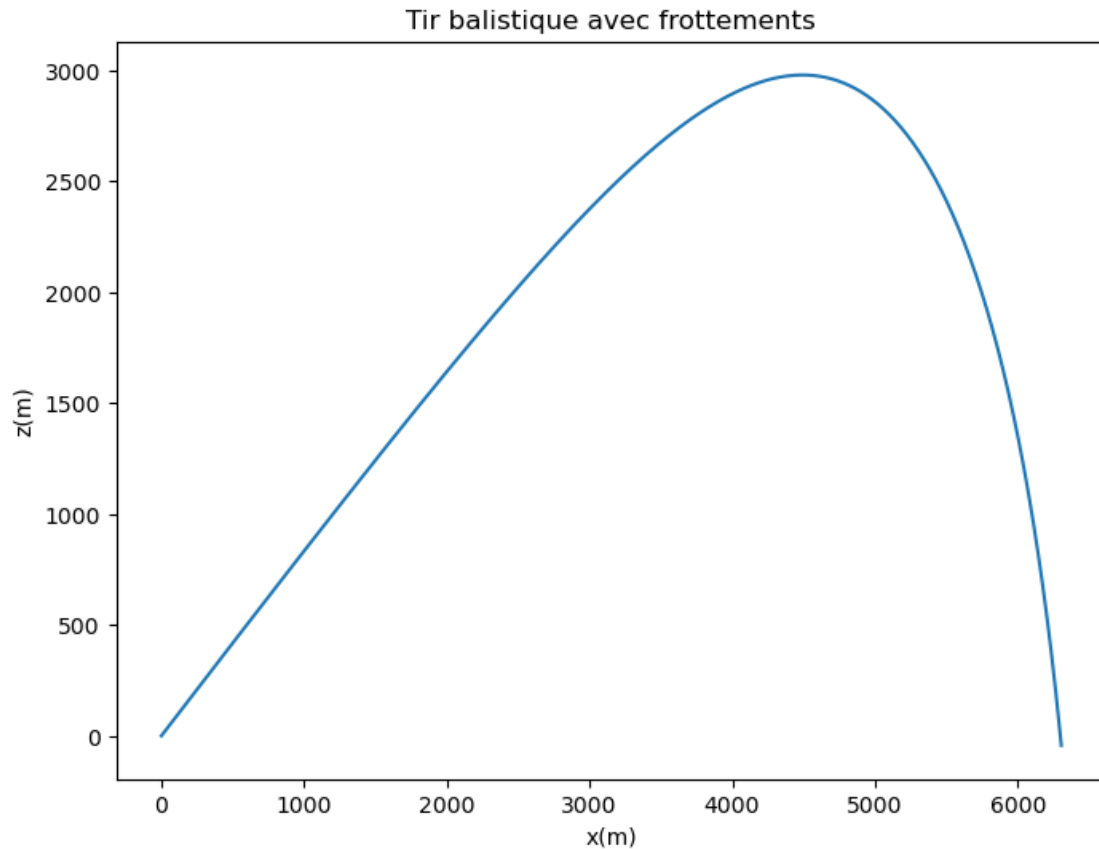
    ↪ tab_dzdt[n+1]=tab_dzdt[n]+h*accelerationz(tab_x[n],tab_z[n],tab_dxdt[n],tab_dzdt[n],tab_t[n])

```

```

[7]: import matplotlib.pyplot as plt
plt.figure(figsize=(8,6))
plt.plot(tab_x,tab_z)
plt.xlabel("x(m)")
plt.ylabel("z(m)")
plt.title("Tir balistique avec frottements")
plt.show()

```



```
[8]: def portee(v0,alpha):
    m,lam,g=7,3e-3,9.8
    def accelerationx(x,z,dxdt,dzdt,t):
        return -lam*np.sqrt(dxdt**2+dzdt**2)*dxdt/m
    def accelerationz(x,z,dxdt,dzdt,t):
        return -g-lam*np.sqrt(dxdt**2+dzdt**2)*dzdt/m
    h=0.01
    tmax=100
    N=int(tmax/h)
    tab_t=np.zeros(N+1)
    tab_x=np.zeros(N+1)
    tab_z=np.zeros(N+1)
    tab_dxdt=np.zeros(N+1)
    tab_dzdt=np.zeros(N+1)
    tab_dxdt[0],tab_dzdt[0]=v0*np.cos(alpha),v0*np.sin(alpha)
    for n in range(N):
        tab_t[n+1]=tab_t[n]+h
        tab_x[n+1]=tab_x[n]+h*tab_dxdt[n]
        tab_z[n+1]=tab_z[n]+h*tab_dzdt[n]
```

```

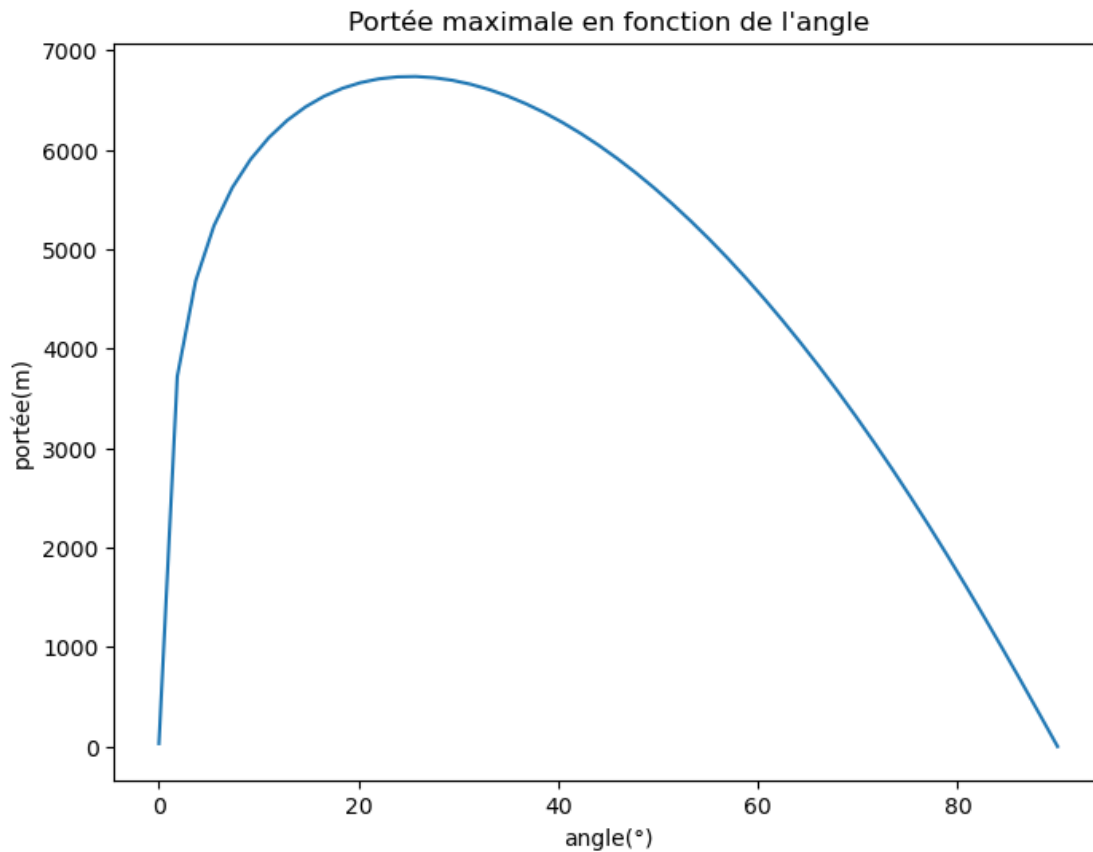
    ↪ tab_dxdt[n+1]=tab_dxdt[n]+h*accelerationx(tab_x[n],tab_z[n],tab_dxdt[n],tab_dzdt[n],tab_t[n])
    ↪ tab_dzdt[n+1]=tab_dzdt[n]+h*accelerationz(tab_x[n],tab_z[n],tab_dxdt[n],tab_dzdt[n],tab_t[n])
    i=0
    while tab_z[i]>=0:
        i=i+1
    return tab_x[i]

```

```

[9]: tab_alphas=np.linspace(0,np.pi/2,50)
    tab_portees=np.zeros_like(tab_alphas)
    for i in range(len(tab_alphas)):
        alpha=tab_alphas[i]
        tab_portees[i]=portee(1500,alpha)
    plt.figure(figsize=(8,6))
    plt.plot(tab_alphas*180/np.pi,tab_portees)
    plt.xlabel("angle(°)")
    plt.ylabel("portée(m)")
    plt.title("Portée maximale en fonction de l'angle")
    plt.show()

```



Portée maximale: 6750m pour $\alpha = 25^\circ$.

[]: