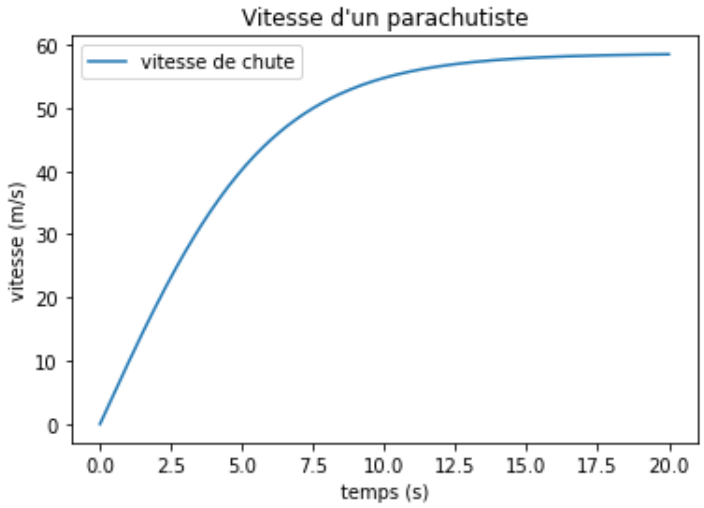
TP Euler - CORRECTION

Exercice 0.1. *Résolution numérique d'une équation différentielle non-linéaire d'ordre 1 : chute avec frottements*

```
import matplotlib.pyplot as pl
import time
import math as mt
import numpy as np


# Constantes physiques
k = .2
m = 70.
g = 9.81
t0 = 0.
v0 = 0.
tmax = 20.

# Fonction f
def f(y,t):
    return -k/m*y**2.+g

# Methode d'Euler d'ordre 1
def euler1(f,t0,y0,tmax,h):
    # Initialisations
    t=[t0]
    y=[y0]
    while t[-1]<= tmax:
        y.append(y[-1] + h*f(y[-1],t[-1]))
        t.append(t[-1] + h)

    return (y,t)
# Resolution de l'ED et affichage de la courbe
v,t = euler1(f,t0,v0,tmax,.1)

pl.plot(t,v,label='vitesse de chute')
pl.title('Vitesse d\'un parachutiste')
pl.xlabel('temps (s)')
pl.ylabel('vitesse (m/s)')
pl.legend()

# Complexite en temps de calcul
for h in [.1,.01,.001]:
    tic=time.time()
    euler1(f,t0,v0,tmax,h)
    tac=time.time()
    print('Le temps de calcul pour h=',h,'est de',tac-tic,'s')
```

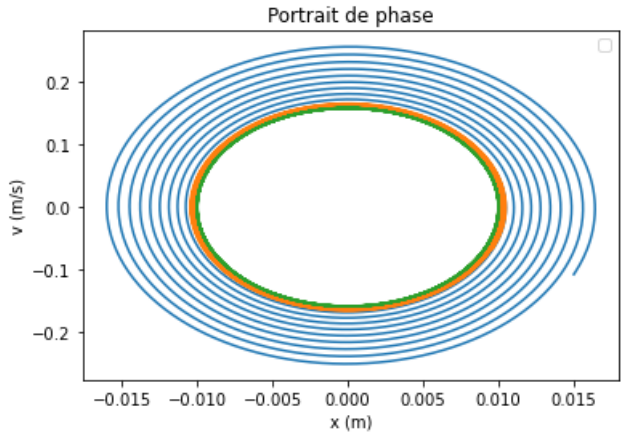
```
# R : tps de calcul inversement proportionnel a h. C'est donc une methode d'ordre 1 (c
```

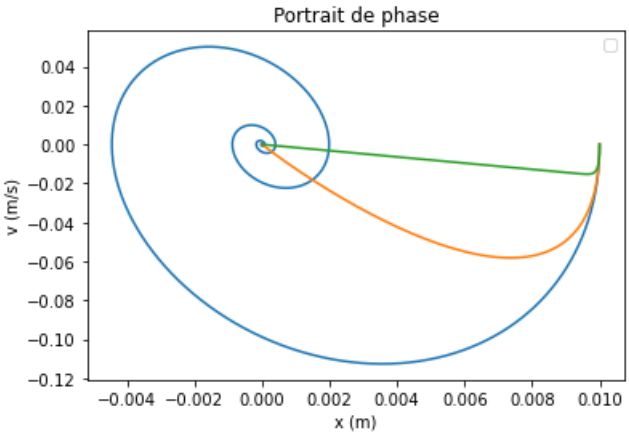


Exercice 0.2. *Résolution numérique d'une équation différentielle d'ordre 2 : oscillateur amorti*

```
# Constantes physiques
k = 25.
m = .1
w0 = mt.sqrt(k/m)
g = 9.81
t0 = 0.
x0 = .01
v0 = 0.
tmax = 4.


# Fonction f
def f(x,v,t):
    return (v,-w0**2*x)

# Methode d'Euler d'ordre 1
def euler2(f,t0,x0,v0,tmax,h):
    # Initialisations
    t=[t0]
    x=[x0]
    v=[v0]
    while t[-1]<= tmax:
        xn,vn = tuple(np.array([x[-1],v[-1]]) +h*np.array(f(x[-1],v[-1],t[-1])))
        t.append(t[-1] + h)
        x.append(xn)
```

```
        v.append(vn)

    return (x,v,t)

# Resolution de l'ED et affichage de la courbe
for h in [.001,.0001,0.00001]:
    x,v,t=euler2(f,t0,x0,v0,tmax,h)


    pl.plot(x,v)
    pl.title('Portrait de phase')
    pl.xlabel('x (m)')
    pl.ylabel('v (m/s)')
    pl.legend()
pl.show()

# R :  diverge pour h trop grand !!

pl.close()


def f(x,v,t):
    return (v,-w0/Q*v-w0**2*x)

# Resolution de l'ED et affichage de la courbe
for Q in [2.,.5,.1]:
    x,v,t=euler2(f,t0,x0,v0,tmax,.0001)


    pl.plot(x,v)
    pl.title('Portrait de phase')
    pl.xlabel('x (m)')
    pl.ylabel('v (m/s)')
    pl.legend()

pl.show()
```



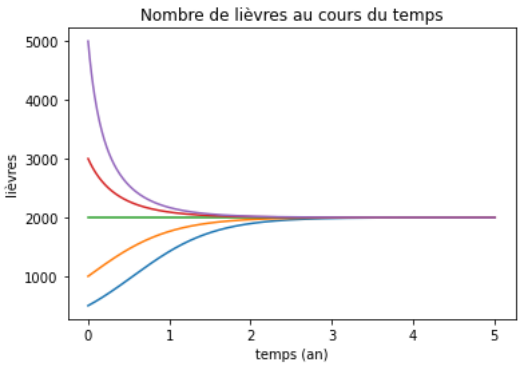


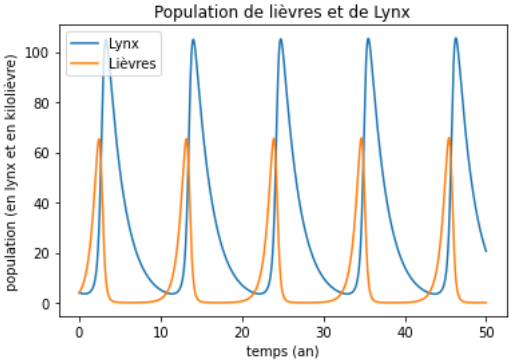
Exercice 0.3. *Résolution numérique d'un système d'équation différentielle d'ordre 1 : modèle de Lotka-Volterra d'évolution des populations*

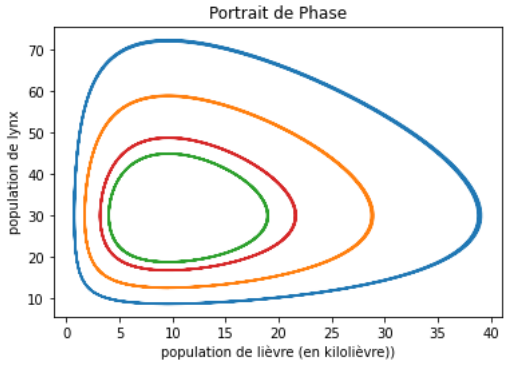
```
def euler1(f,t0,y0,tmax,h):
    # Initialisations
    t=[t0]
    y=[y0]
    while t[-1]<= tmax:
        y.append(y[-1] + h*f(y[-1],t[-1]))
        t.append(t[-1] + h)

    return (y,t)

a = 2.
b = 0.001
t0 = 0.
tmax = 5.

# Fonction f
```

```
def f1(x,t):
    return (a-b*x)*x

for y0 in [500,1000,2000,3000,5000]:
    y,t = euler1(f1,t0,y0,tmax,.01)

    pl.plot(t,y,label=str(y0)+' lievres')
    pl.title('Nombre de lievres au cours du temps')
    pl.xlabel('temps (an)')
    pl.ylabel('lievres')
pl.show()


pl.close()



def euler2(f,t0,x0,v0,tmax,h):
    # Initialisations
    t=[t0]
    x=[x0]
    v=[v0]
    while t[-1]<= tmax:
        xn,vn = tuple(np.array([x[-1],v[-1]]) +h*np.array(f(x[-1],v[-1],t[-1])))
        t.append(t[-1] + h)
        x.append(xn)
        v.append(vn)

    return (x,v,t)


# Constantes physiques
a = 2.
b = 0.001
t0 = 0.
tmax = 5.

# Fonction f
def f1(x,t):
    return (a-b*x)*x



# Constantes physiques
a = 1.5
b = 0.05
c = 0.48
d = 0.05
t0 = 0.
```

```
y0=4.
l0=4.
tmax = 50.
h = 0.0005

# Fonction f
def f2(l,y,t):
    return ( a*l-b*l*y , -c*y + d*l*y )

l,y,t = euler2(f2,t0,l0,y0,tmax,h)

pl.plot(t,y,label='Lynx')
pl.plot(t,l,label='Lievres')
pl.legend()
pl.title('Population de lievres et de Lynx')
pl.xlabel('temps (an)')
pl.ylabel('population (en lynx et en kilolievre)')
pl.show()
pl.close()

for y0 in [10,15,30,40]:

    l,y,t = euler2(f2,t0,l0,y0,tmax,h)

    pl.plot(l,y)
    pl.title('Portrait de Phase')
    pl.ylabel('population de lynx')
    pl.xlabel('population de lievre (en kilolievre))')
pl.show()
```




Population de lièvres et de Lynx
Lynx
Lièvres
population (en lynx et en kilolièvre)
temps (an)
0
20
40
60
80
100
0
10
20
30
40
50


Portrait de Phase
population de lynx
population de lièvre (en kilolièvre))
10
20
30
40
50
60
70
0
5
10
15
20
25
30
35
40