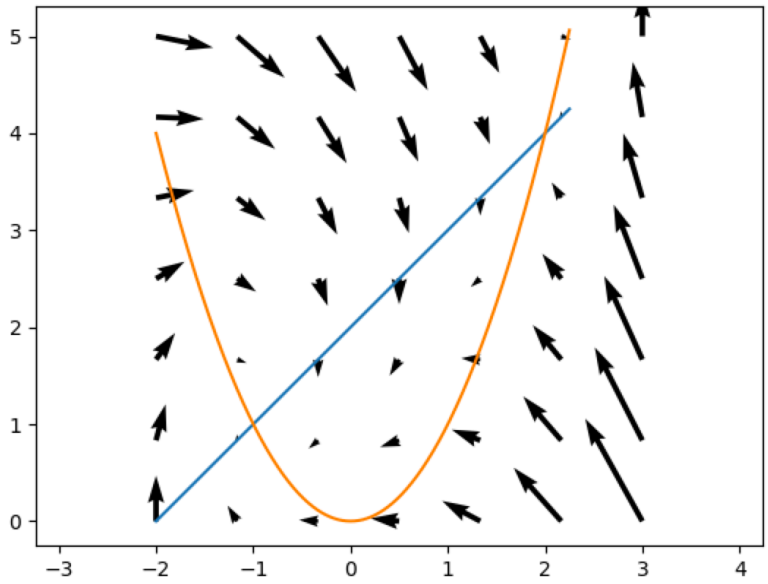
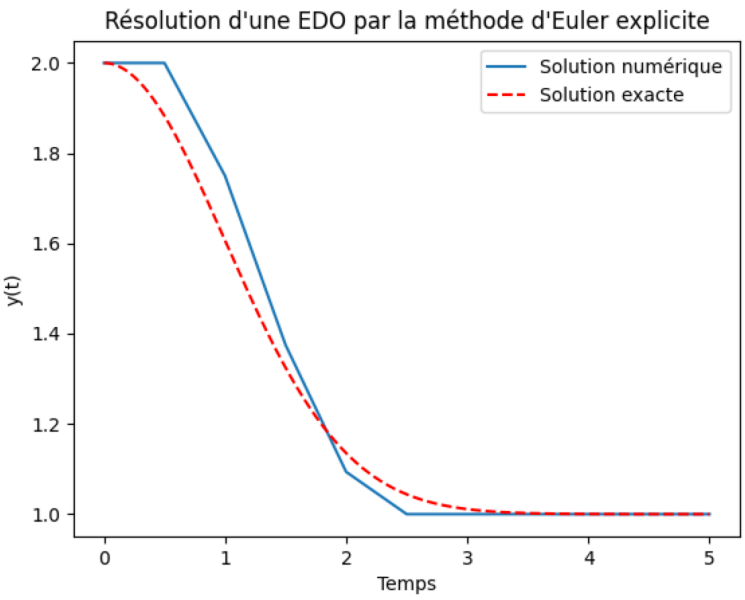
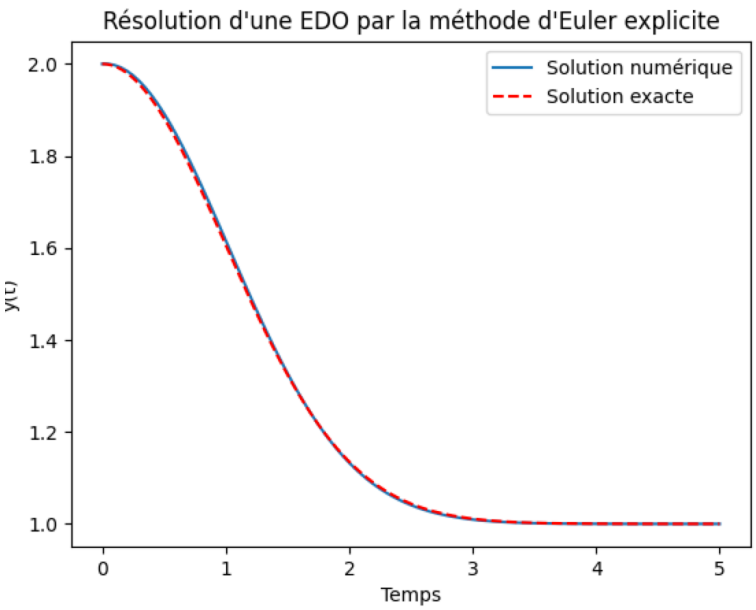
```
from numpy import *
from matplotlib.pyplot import *
from scipy.integrate import *

def f(x,t):
    return [x[1]-x[0]-2,x[0]**2-x[1]]

T=linspace(-2,2.25,1000)
plot(T,T+2)
plot(T,T**2)

X = np.linspace(-2, 3, 7)
Y = np.linspace(0, 5, 7)
U, V = meshgrid(X, Y)

Q1,Q2=f((U,V),0)

q = quiver(X,Y,Q1,Q2)
axis('equal')
show()
```



```
from matplotlib.pyplot import *
from numpy import *

N=100
T=5
y_0=2

def f(t,y):
    return t-t*y

def solution(t):
    return 1+exp(-t**2/2)

def Euler(T,f,y_0,N):
    h=T/N
    Y=[y_0]
    t=0
    y=y_0
    for k in range(1,N+1):
        y=y+h*f(t,y)
        t+=h
        Y.append(y)
    return Y

Y=Euler(T,f,y_0,N)
liste_T=linspace(0,T,N+1)
plot(liste_T,Y,label='Solution numérique')
plot(linspace(0,T,5000),solution(linspace(0,T,5000)),'--r',label='Solution exacte')
xlabel('Temps')
ylabel('y(t)')
legend()
title("Résolution d'une EDO par la méthode d'Euler explicite")
show()
```





```
liste_N=range(10,2010,10)
Err=[]
for N in liste_N:
    liste_T=linspace(0,T,N+1)
    Erreur=array(Euler(T,f,y_0,N))-(solution(liste_T))
    Err.append(max(abs(Erreur)))
plot(log(liste_N),log(Err))
xlabel('Nombre de points N, en échelle logarithmique')
ylabel('Erreur en norme infinie, en échelle logarithmique')
title("Illustration de l'ordre de convergence de la méthode d'Euler explicite")
show()
(a,b)=polyfit(log(liste_N),log(Err),1)
print(a)
```