LYCEE LISSANE EDDINE
BEN AL KHATIB
CPGE-LAAYOUNE

Filière : MP
AS : 2015-2016
CHAHDI

Programmation Python

RÉSOLUTION NUMÉRIQUE D'ÉQUATIONS DIFFÉRENTIELLES PAR LA MÉTHODE D'EULER

Dans ce chapitre nous allons voir comme résoudre une équation différentielle du premier ordre dont la forme mathématique est **y'= f(x,y)** à partir de la connaissance de la valeur de y = y0 pour une valeur de x = x0.

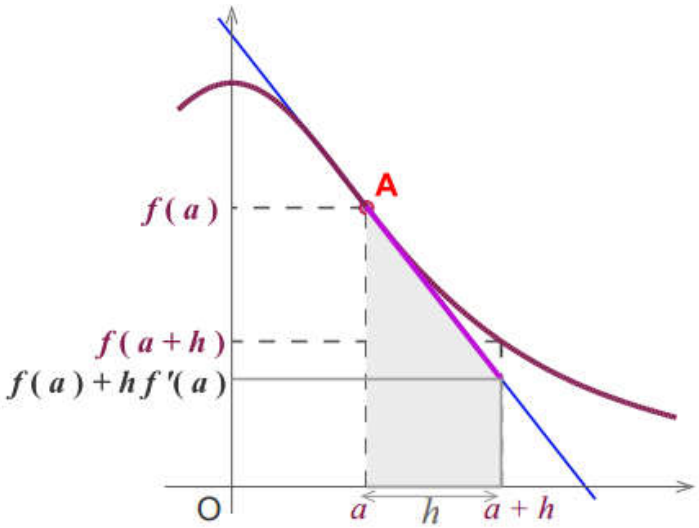
1) Méthode d'euler

Soit f est une fonction dérivable sur un intervalle I et a un réel de I. Les valeurs a, f (a) et f ' (a) nous permettent de déterminer l'équation de la tangente à la courbe représentative de la fonction au point d'abscisse a : y=f ' (a)(x a)+f(a)

Pour tout réel h non nul, l'ordonnée d'un point d'abscisse x = a + h de la tangente est y=f ' (a)h+f(a).

Quand h est proche de 0, c'est une valeur approchée de f (a + h).

Pour tout réel *h* non nul, et proche de 0 tel que *a* + *h* soit dans I on a : $f(a+h)\approx f(a)+hf'(a)$



Il s'agit de la meilleure approximation affine de la fonction au voisinage du point *a*, à condition de ne pas fixer à l'avance l'erreur admise et l'intervalle de l'approximation. Cette approximation sera d'autant meilleure que h sera petit.

2) Illustration graphique de la méthode d'Euler

a. Implémentation de la méthode d'Euler

L'implémentation de la méthode d'euler

```
#méthode d'Euler
def euler(f,y0,x0,xf,N):
    # f : fonction données
    # (y0;x0):point innitial
    # xf : point abscisse final[x0;xf]
    lesy=[y0]#liste des ordonnées yk,k=0;1;2...;N
    lesx=[x0]
    h = (xf-x0)/N #le pas
    for k in range (N):
        y0 += h*F(y0,x0)
        x0 += h
        lesy.append(y0)#stocker les solutions
        lesx.append(x0)#stocker les abscisses
    return lesx,lesy
```

b. Exemple mathématique

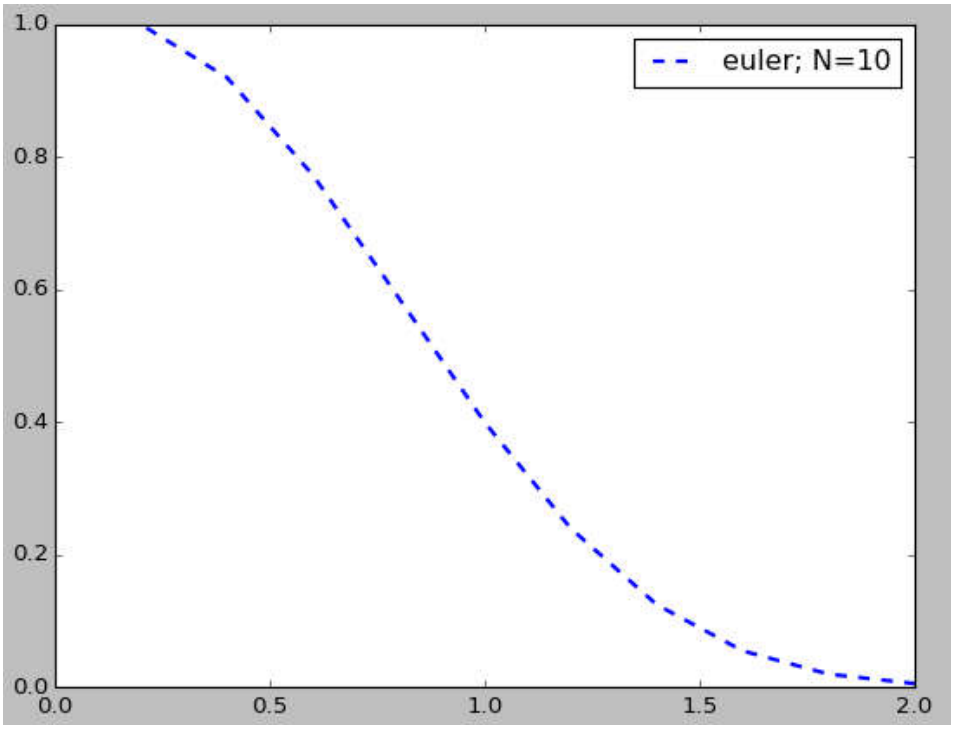
Résolution par la méthode d'Euler, dans l'intervalle [0, 2] de l'équation différentielle: **y' = -2*x*y** avec **y(0)=1** , dont on connaît la solution exacte **y=exp(x2)**

Définition de l'équation différentielle y' = F(y,x)=-2xy

```
#y' = -2xy
def F(y,x):
    return -2*x*y
```

Résolution de l'ED avec la méthode d'euler pour N=10 ,dans l'intervalle [0, 2]

```
import matplotlib.pyplot as plt

lesxy = euler(f,1,0,2,10)#euler pour N=10
lesx  = lesxy[0]
lesy  = lesxy[1]

plt.plot(lesx,lesy,"--",linewidth=2,label="euler; N=10")
plt.legend()
plt.show()
```

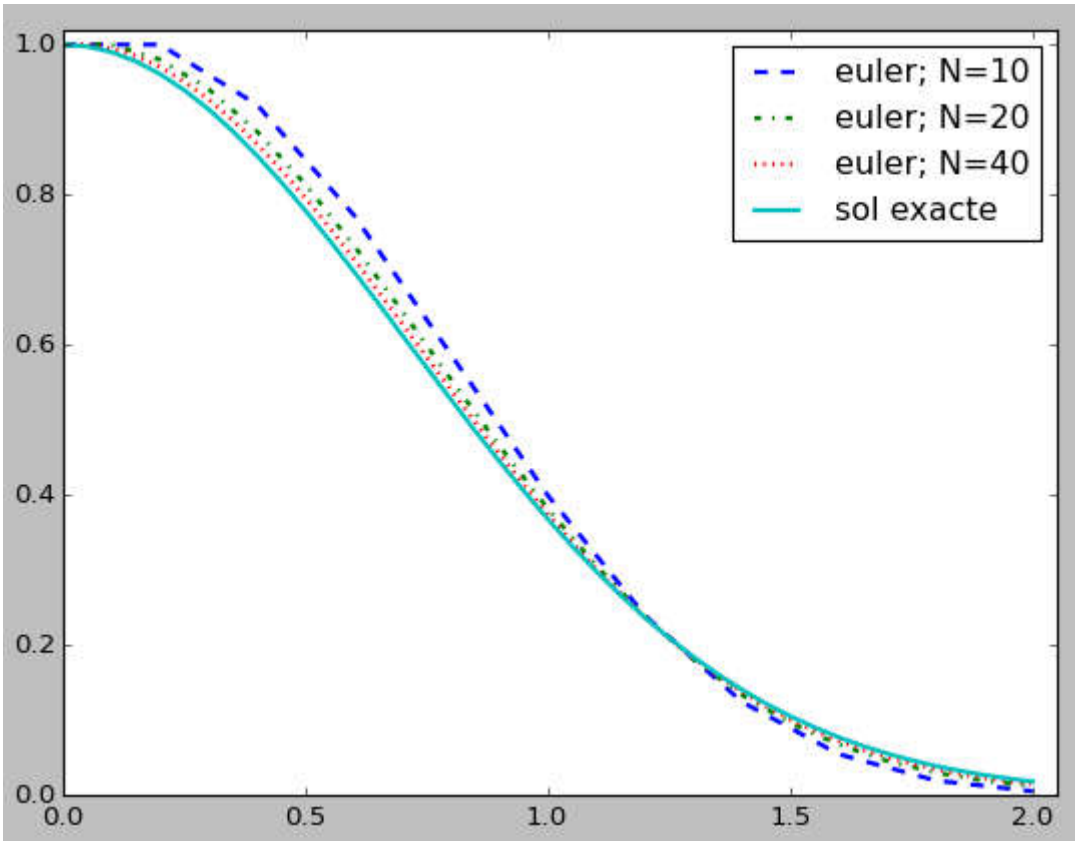


Définition de la fonction $y = \exp(-x^2)$ (la solution exacte)

```
import numpy as np
def f(x):
    return np.exp(-x*x)
```

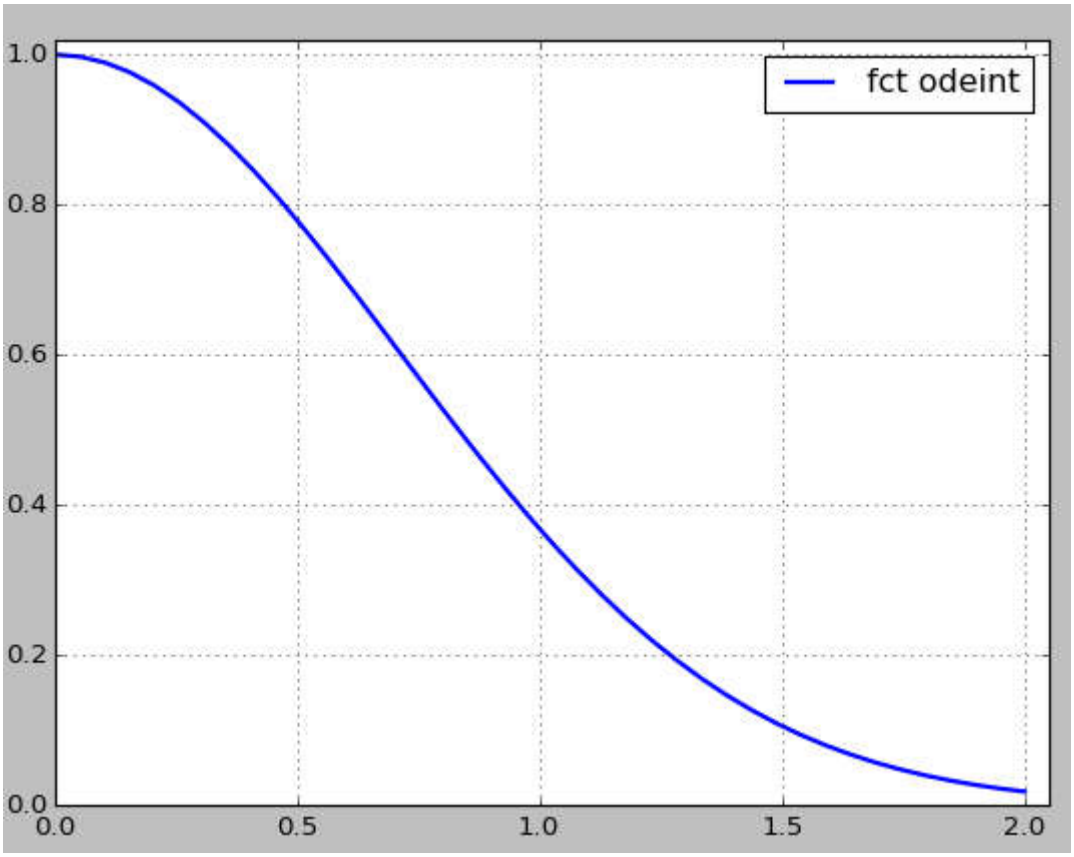
Comparaison entre la solution exacte et la méthode d'Euler

```
lesxy = euler(f,1,0,2,10)#euler pour N=10
lesx,lesy  = lesxy[0],lesxy[1]
plt.plot(lesx,lesy,"--",linewidth=2,label="euler; N=10")

lesxy = euler(f,1,0,2,20)#euler pour N=20
lesx,lesy  = lesxy[0],lesxy[1]
plt.plot(lesx,lesy,"-.",linewidth=2,label="euler; N=20")

lesxy = euler(f,1,0,2,40)#euler pour N=40
lesx,lesy  = lesxy[0],lesxy[1]
plt.plot(lesx,lesy,":",linewidth=2,label="euler; N=40")
plt.plot(lesx,f(np.array(lesx)),linewidth=2,label="sol exacte")
plt.axis([0,2.05,0,1.02])
plt.legend()
plt.show()
```



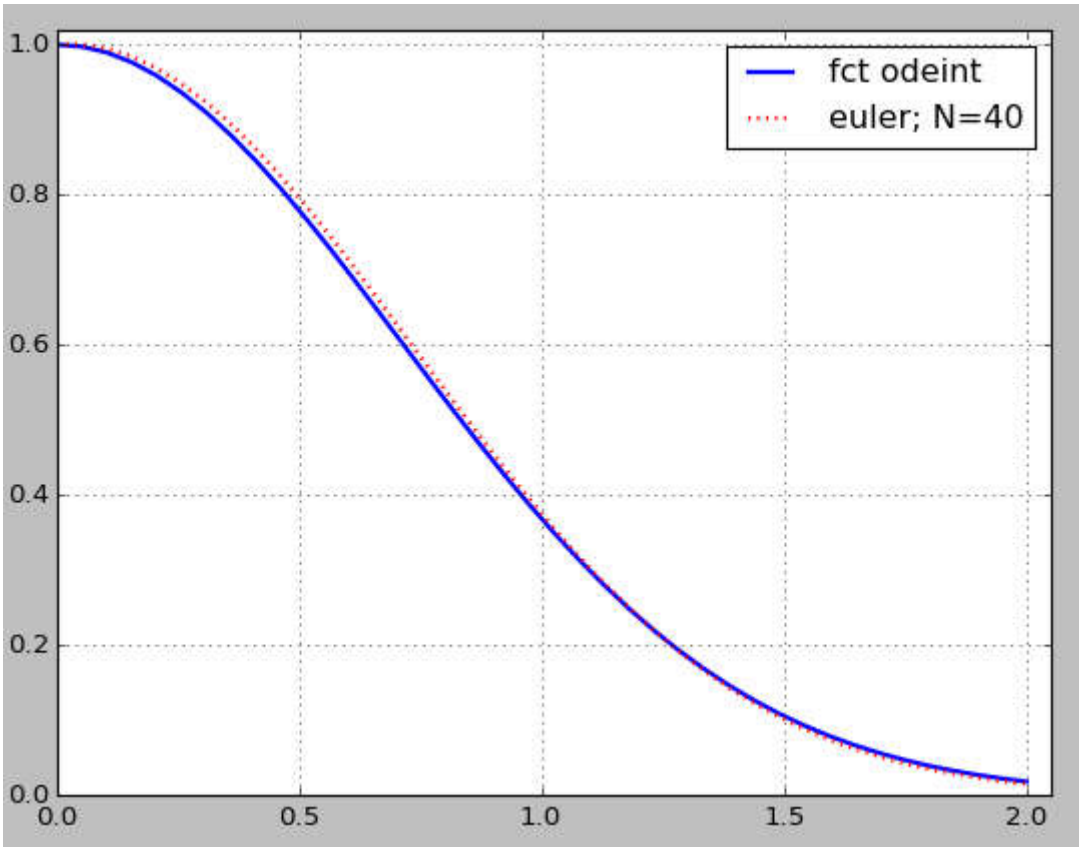
Utilisation de la fonction odeint

```
from scipy.integrate import  odeint
import numpy as np

#y' = -2xy
def F(y,x):
    return -2*x*y

lesx = np.linspace (0,2,40)
lesy = odeint(F,1,lesx)
plt.plot(lesx,lesy,linewidth=2,label="fct odeint")
plt.axis([0,2.05,0,1.02])
plt.grid( )
plt.legend()
plt.show()
```



La comparaison entre les trois méthodes d'euler et odeint

```
lesx = np.linspace (0,2,40)
lesy = odeint(F,1,lesx)
plt.plot(lesx,lesy,linewidth=2,label="fct odeint ")
lesxy = euler(f,1,0,2,40)#euler pour N=40
lesx,lesy  = lesxy[0],lesxy[1]
plt.plot(lesx,lesy,"r:",linewidth=2,label="euler; N=40")
plt.axis([0,2.05,0,1.02])
plt.grid( )
plt.legend()
plt.show()
```


fct odeint
euler; N=40
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0
1.5
2.0