

# TP 1: Equations différentielles

DRAME Mariam et BLEUSE Chloé

November 23, 2017

## 1 Exemple

Après avoir calculer  $u$  et l'avoir dessiné sur papier, nous pouvons écrire le programme suivant pour la représenter:

```
def u(x):  
    y = -exp(-x)  
    return y
```

On définit alors  $f$  par la fonction suivante:

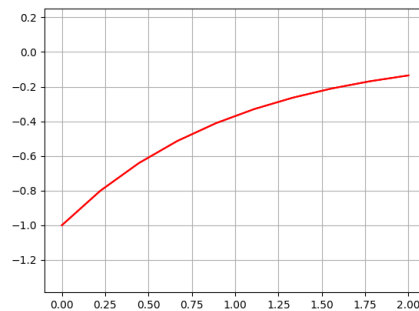
```
def f(t,u):  
    return -u
```

Le graphique de  $u$  se code de la façon suivante:

```
import matplotlib.pyplot as plt  
import numpy as np
```

```
t=np.linspace(0,2,10)  
plt.plot(t,u(t),'r-')  
plt.grid ('on')  
plt.axis ('equal')  
plt.savefig('graphU.png')
```

On obtient alors le graphique de la fonction  $u$ :



## 2 Méthode d'Euler

Pour obtenir le calcul des  $n+1$  premiers termes de la suite  $(u_k)$ , définie par la méthode d'Euler, on fait le programme suivant:

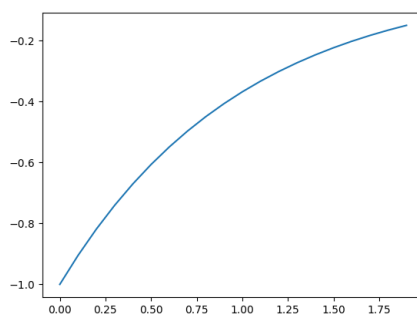
```
import numpy as np
T=2.0
n=10
h=T/n

tt=np.zeros(n+1)
for k in range (n+1):
    tt[k]=k*h

uu=np.zeros(n+1)
u0=1.0
uu[0]=u0
for k in range(1,n+1):
    u1=u0+h*f(tt[k-1],u0)
    uu[k]=u1
    u0=u1

plt.plot(tt,uu,'r',tt,u(tt))
plt.grid()
plt.savefig('graphtp1.png')
```

Nous avons fait le choix les inscrire directement dans un graphique qui est le suivant:



## 3 Fonction python de la méthode d'Euler

La fonction python *euler* qui prend en argumens une fonction python  $f$  (de deux variables scalaires)  $t, u$  une valeur initiale  $u_0$ , un réel  $T$ , un entier  $n$ , et

qui renvoie deux numpy arrays:  $tt$  de taille  $n + 1$ , contenant les valeurs  $t_k$  et  $uu$  de taille  $n + 1$ , contenant les valeurs  $u_k$  obtenues par la méthode d'Euler est la suivante:

```
def euler(f, U0, T, n):
    h=T/n
    tt=np.zeros(n+1)
    for k in range (n+1):
        tt[k]=k*h
    uu=np.zeros(n+1)
    uu[0]=u0
    for k in range (1, n+1):
        u1= u0+ h *f(tt[k-1]], u0)
        uu[k]=u1
        u0=u1
    return tt, uu
    tt=np.zeros(n)
```

## 4 Autres exemples

La fonction  $f$  de l'exemple 2 est:

```
def f(t,u):
    return -u
```

On applique alors la fonction *euler* à l'exemple 2, ce qui correspond à faire:

```
r = euler(f, 2.0, 1.0, 10) #Q4b
print r
```

On fait alors la même chose pour toutes les autres équations différentielles:

```
r = euler(f1, 2.0, 1.0, 10)
print r
```

```
r = euler(f2, 2.0, 1.0, 10)
print r
```

```
r = euler(f3, 2.0, 1.0, 10)
print r
```

avec les fonctions suivantes:

```
def f1(t,u):
    return -u + t
```

```
def f2(t,u):
```

```

        return u**2

def f3(t,u):
    return u**2 - t

```

## 5 Equations différentielles d'ordre 2

La fonction  $F$  de l'équation différentielle (5) dans une fonction python `f5`, dont on choisit  $\omega = 1.0$ .  $F$  prend en arguments un flottant  $t$  et un numpy array  $U$  de taille 2 est:

```

def f5(t,U) :
    omega=1.0
    u=U[0]
    v=U[1]
    U_out=np.zeros(2)
    U_out[0]=v
    U_out[1]=-omega**2 *u
    return U_out

```

Maintenant, on modifie la fonction *euler* pour qu'elle prenne quatre arguments: une fonction python  $F$ , une valeur initiale  $U0$  numpy array, un flottant  $T$  et un entier  $n$ , *euler2* renverra deux numpy arrays:  $tt$  de taille  $n + 1$ , contenant les valeurs  $tk$ , et  $UU$  de taille  $2 \times (n + 1)$ , contenant les valeurs  $U_k$  obtenues par la méthode d'Euler. On a alors la fonction suivante:

```

def euler_2 (f,U0,T,n):
    U=U0
    d=U0.shape[0]
    t=0.0
    h=T/n
    tt=np.zeros(n)
    UU=np.zeros((d,n))
    for i in range (n):
        UU[:,i]=U
        tt[i]=t
        U=U +h*f(t,U)
        t+=h
    return tt, UU

```

On réalise alors trois graphiques: la position  $u$  en fonction du temps  $t$ , la vitesse  $v$  en fonction du temps  $t$  et la vitesse  $v$  en fonction de la position  $u$ . On a alors le code suivant:

```

u0 = 1.0
v0 = 0.0
U0 = np.array ([u0, v0])

```

```

T = 2.0
n = 10
tt, UU = euler_2(f5, U0, T, n)
print tt
print UU

import matplotlib.pyplot as plt
plt.plot(tt, UU[0], 'r-')
plt.plot(tt, UU[1], 'b-')
plt.plot(UU[0], UU[1], 'v-')
plt.grid('on')
plt.axis('equal')
plt.savefig('graphf5.png')

```

On obtient alors le graphique suivant:

