24

Équations différentielles d'ordre 2

◇ Ni PYTHON ni SCILAB ne disposent de fonctions pour résoudre numériquement une équation différentielle d'ordre 2. Cependant, une telle équation différentielle peut toujours être transformée en un système d'équations différentielles d'ordre 1.

◇ On considère le problème de Cauchy :

$$\begin{cases} x'' = f(x,t) \\ x(t_0) = x_0 \\ x'(t_0) = y_0 \end{cases}$$

On pose $y = x'$, alors :

$$x'' = f(x,t) \iff y' = f(x,t) \iff \begin{cases} x' = y \\ y' = f(x,t) \end{cases}$$

On obtient le problème de Cauchy, d'ordre 1 :

$$\begin{cases} x' = y \\ y' = f(x,t) \\ x(t_0) = x_0 \\ y(t_0) = y_0 \end{cases}$$

On le résout avec les techniques vues pour les systèmes d'équations différentielles.

Exemple. On considère l'équation différentielle du pendule :

$$\theta'' + \omega^2 \sin(\theta) = 0$$

avec $\omega = \sqrt{g/L}$. On prendra $L = 1$ m. On posant $v = \theta'$, on obtient le système :

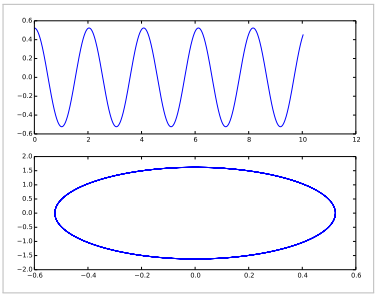
$$\begin{cases} \theta' = v \\ v' = -\omega^2 \sin(\theta) \end{cases}$$

Il s'écrit $X' = F(X,t)$ avec $X = \begin{bmatrix} \theta \\ v \end{bmatrix}$ et $F(X,t) = \begin{bmatrix} v \\ -\omega^2 \sin(\theta) \end{bmatrix}$. On résout sur l'intervalle $[0, 5T]$ avec $T = 2\pi/\omega$.

I. Avec Numpy/Scipy

```
from math import *
from scipy.integrate import odeint
import numpy as np
import matplotlib.pyplot as plt
```

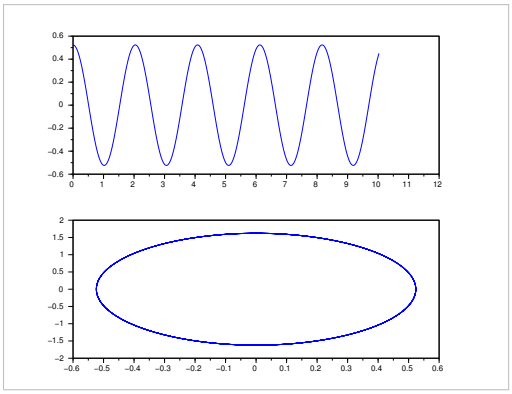
```
omega=sqrt(9.81)
T=2*pi/omega
def F(X,t):
    theta=X[0]
    v=X[1]
    thetaprim=v
    vprim=-omega**2*sin(theta)
    return np.array([thetaprim,vprim])
t=np.linspace(0,5*T,1000)
X=odeint(F,np.array([pi/6,0]),t)
plt.subplot(2,1,1)
plt.plot(t,X[:,0])
plt.subplot(2,1,2)
plt.plot(X[:,0],X[:,1])
plt.show()
```



On a représenté θ en fonction de t puis θ' en fonction de θ (portrait de phase).

II. Avec Scilab

```
omega=sqrt(9.81);
T=2*%pi/omega;
function Xprim=F(t,X)
    theta=X(1)
    v=X(2)
    dtheta=v
    dv=-omega^2*sin(theta)
    Xprim=[dtheta;dv]
endfunction
t=linspace(0,5*T,1000);
X=ode([%pi/6;0],0,t,F);
subplot(2,1,1)
plot(t,X(1,:))
subplot(2,1,2)
plot(X(1,:),X(2,:))
```



Exemples de résolution approchée d'éq. diff. d'ordre 2

```
from math import *
from scipy.integrate import odeint
import numpy as np
import matplotlib.pyplot as plt
```

1 En mathématiques

Exemple. On veut résoudre sur l'intervalle $[0,10]$ le problème de Cauchy :

$$\begin{cases} y''+3y'+2y=2\cos(t) \\ y(0)=1 \\ y'(0)=0 \end{cases}$$

- Le nouveau système en posant $z=y'$:

$$\begin{cases} y'=z \\ z'=-3z-2y+2\cos(t) \\ y(0)=1 \\ z(0)=0 \end{cases}$$

- La fonction F associée au système avec $X=\left[\begin{smallmatrix} y \\ z \end{smallmatrix}\right]$:

```
def F(X,t):
    y=X[0]
    z=X[1]
    return [z,-3*z-2*y+2*cos(t)]
```

- Le vecteur t qui représente l'intervalle de temps considéré :

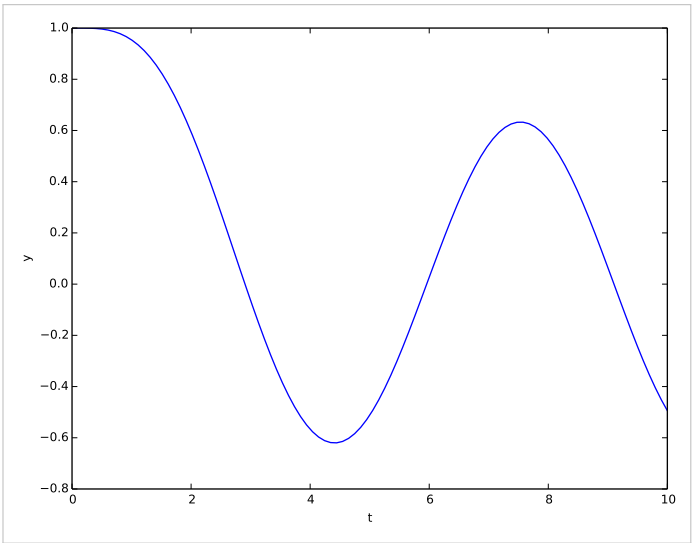
```
t=np.linspace(0,10,100)
```

- La résolution avec la commande **odeint** :

```
X=odeint(F,[1,0],t)
```

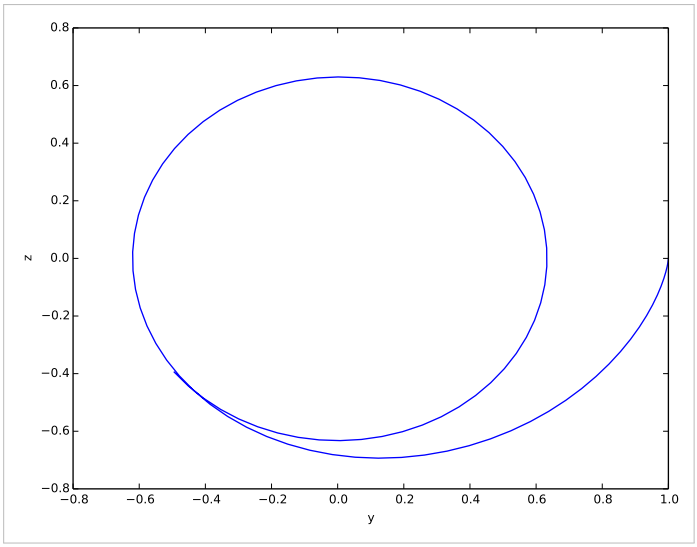
- Représentation graphique de la solution y :

```
plt.clf()
plt.plot(t,X[:,0],'b-')
plt.xlabel("t")
plt.ylabel("y")
```



- Portrait de phase :

```
plt.clf()
plt.plot(X[:,0],X[:,1],'b-')
plt.xlabel("y")
plt.ylabel("z")
```



2 En physique

Exemple. On considère l'équation différentielle du second ordre sous forme canonique :

$$\begin{cases} \dfrac{1}{\omega_0^2}\dfrac{\mathrm{d}^2u}{\mathrm{d}t^2} + \dfrac{2\xi}{\omega_0}\dfrac{\mathrm{d}u}{\mathrm{d}t} + u = 0 \\ u(t=0) = A \\ \left.\dfrac{\mathrm{d}u}{\mathrm{d}t}\right|_{t=0} = 0 \end{cases}$$

avec $w_0 = 10^4$ rad$\cdot$s^{-1} et $A = 1$ V. On pose $T = 2\pi/\omega_0$. Représenter la fonction u sur $[0, 4T]$ pour $\xi = 0.1, 0.2, \ldots, 0.9$.

- Le nouveau système, en posant $v = \dfrac{\mathrm{d}u}{\mathrm{d}t}$:

$$\begin{cases} \dfrac{\mathrm{d}u}{\mathrm{d}t} = v \\ \dfrac{\mathrm{d}v}{\mathrm{d}t} = -\omega_0^2 \left(\dfrac{2\xi}{\omega_0} v + u \right) \\ u(0) = A \\ v(0) = 0 \end{cases}$$

- La fonction F associée au système avec $X = \left[\begin{smallmatrix} u \\ v \end{smallmatrix} \right]$, en prenant ξ comme paramètre :

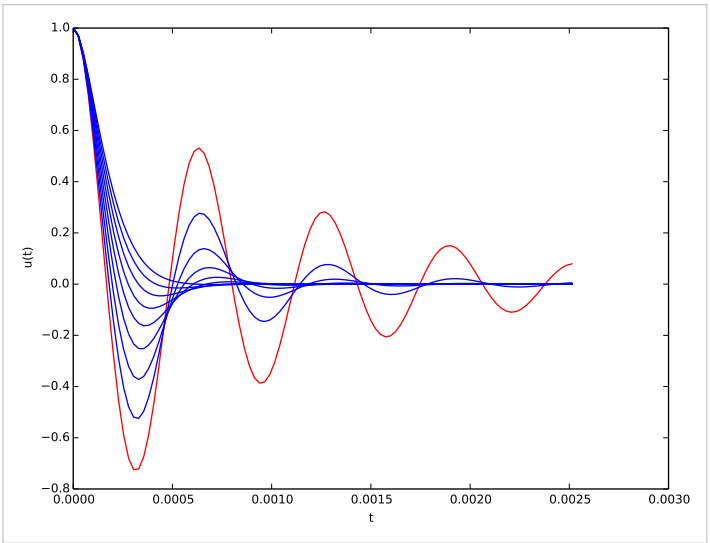
```
w0=1e4
A=1
T=2*pi/w0
def F(X,t,xi):
    u=X[0]
    v=X[1]
    return [v,-w0**2*(2*xi/w0*v+u)]
```

- Le vecteur t qui représente l'intervalle de temps considéré :

```
t=np.linspace(0,4*T,100)
```

- Représentation graphique :

```
plt.clf()
for xi in [float(k)/10 for k in range(1,10)]:
    couleur="r" if xi<0.15 else "b"
    X=odeint(F,[A,0],t,args=(xi,))
    plt.plot(t,X[:,0],couleur)
plt.xlabel("t")
plt.ylabel("u(t)")
```



3 Autre exemple en physique

Exemple. On étudie la chute d'un grêlon de masse $m = 3.6 \cdot 10^{-3}$ kg. On note y son ordonnée sur un axe vertical descendant et on a l'équation différentielle :

$$\begin{cases} m\dfrac{\mathrm{d}^2 y}{\mathrm{d}t^2} = mg - k_2\left(\dfrac{\mathrm{d}y}{\mathrm{d}t}\right)^2 \\ y(t=0) = 0 \text{ (valeur arbitraire)} \\ \left.\dfrac{\mathrm{d}y}{\mathrm{d}t}\right|_{t=0} = 0 \end{cases}$$

(modèle des frottements quadratiques, $k_2 = 9.2 \cdot 10^{-5}$). On prendra $y(t=0) = 0$ (valeur arbitraire). Représenter y en fonction de t sur l'intervalle $[0,15]$. Représenter la vitesse en fonction du temps sur le même intervalle.

- Le nouveau système en posant $v = \dfrac{\mathrm{d}y}{\mathrm{d}t}$:

$$\begin{cases} \dfrac{\mathrm{d}y}{\mathrm{d}t} = v \\ \dfrac{\mathrm{d}v}{\mathrm{d}t} = g - \dfrac{k_2}{m}v^2 \\ y(0) = 0 \\ v(0) = 0 \end{cases}$$

- La fonction F associée au système avec $X = \begin{bmatrix} y \\ v \end{bmatrix}$:

```
m=3.6e-3
k2=9.2e-5
g=9.81
def F(X,t):
    y=X[0]
    v=X[1]
    return [v,g-k2/m*v**2]
```

- Le vecteur t qui représente l'intervalle de temps considéré :

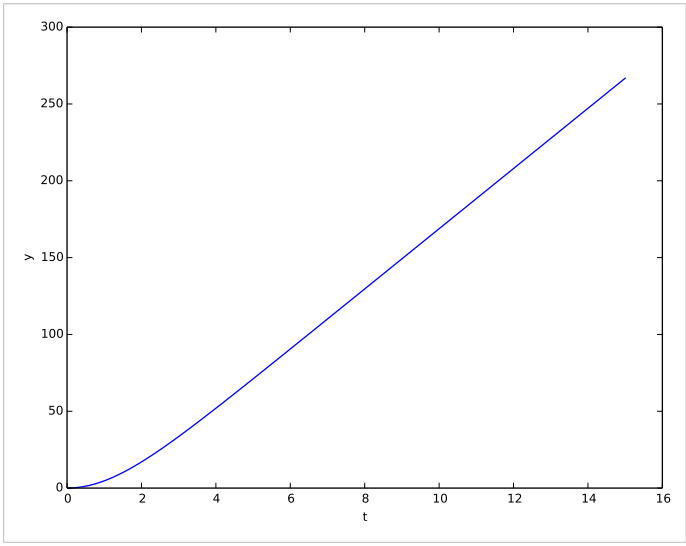
```
t=np.linspace(0,15,100)
```

- La résolution avec la commande **odeint** :

```
X=odeint(F,[0,0],t)
```

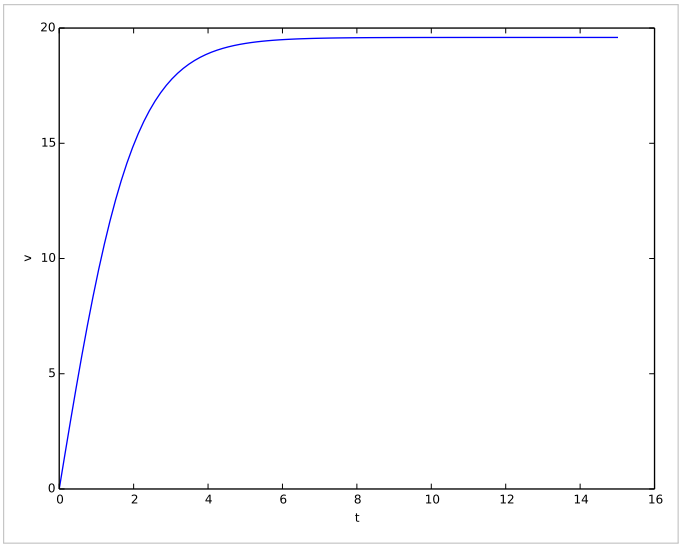
- Représentation graphique de la solution y :

```
plt.clf()
plt.plot(t,X[:,0],'b-')
plt.xlabel("t")
plt.ylabel("y")
```



- Représentation graphique de la vitesse :

```
plt.clf()
plt.plot(t,X[:,1],'b-')
plt.xlabel("t")
plt.ylabel("v")
```


20
15
10
5
0
v
0
2
4
6
8
10
12
14
16
t