

Exercice 1 :

A l'aide de la fonction `odeint()` du sous-module `integrate` de `scipy`, et du module `matplotlib.pyplot`, donner le code permettant de tracer la courbe représentative sur l'intervalle $[0; 4]$ (on prendra une subdivision de 100 points) de la solution approchée de l'équation

suivante :
$$\begin{cases} y' + \sqrt{t} \cdot y = 0 \\ y(0) = 1 \end{cases}$$

Correction

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import integrate
phi = lambda y,t: -y * t**0.5
t = np.linspace(0,4,100)
y0 = 1
y = integrate.odeint(phi,y0,t)
plt.plot(t,y)
plt.show()
```

Exercice 2 : Evolution d'une température en chimie

On note $y(t)$ la température en degrés Celsius d'une réaction chimique en fonction du temps t , t étant exprimé en heures.

Après étude, on constate que la température est solution de l'équation différentielle :

$$y' + y = e^{-0,25t}$$

Avec la condition initiale $y(0)=20^{\circ}\text{C}$.

Ecrire un programme qui donne un tableau composé des températures de la réaction chimique chaque demi-heure pendant la première journée de la réaction.

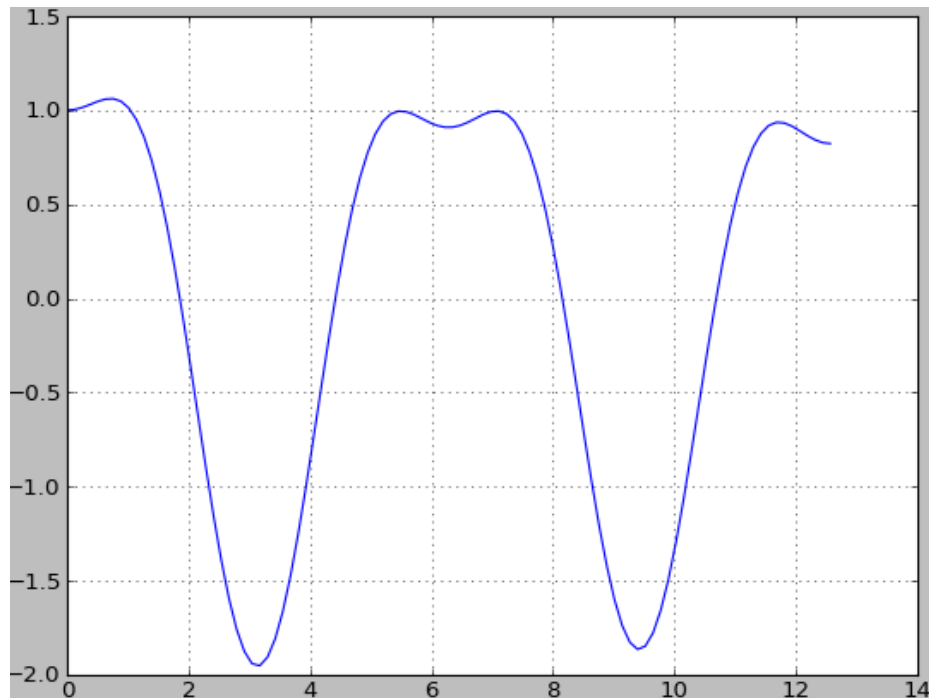
Correction

```
import numpy as np
from scipy.integrate import odeint
t=np.arange(0,24.5,0.5)
phi = lambda y,t: (exp(-0.25*t))-y
s = odeint(phi,20,t)
print(s)
```

Exercice 3 :

Résoudre l'équation : $\ddot{y} + 2y + \omega_0^2 = A \cos(\omega t)$ dans l'intervalle $[0, 4\pi]$ avec une subdivision de 100 points ; avec : $y(0)=1$; $y'(0)=0$; $A=1.5$; $\omega = 1$; $\lambda = 0.01$; $\omega_0 = 2$

Puis tracer la solution :



#importations

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

#fonction f(z,t)

```
def f(z,t) :  
    y,dy=z  
    return (dy, -2*lambda*dy-w0**2*y+A*cos(w*t))
```

#Appel

```
lambda,w0,A,w =0.01,1.0,1.5,2.  
z=[1.,0.]  
t=linspace(0,4*pi,100)  
d=odeint(f,z,t)  
S=d[:,0]  
plt.plot(t,S)  
plt.grid()  
plt.show()
```

Exercice 4 : Le parachute

La trajectoire suivie par un objet relié à un parachute est un axe vertical noté $(O, \vec{i})$.

A un instant donné, le vecteur vitesse $\vec{V}$ de l'objet est défini par $\vec{V}(t) = v(t) \vec{i}$ où v est une fonction de la variable réelle positive t (temps).

Dans ces conditions de l'expérience, le vecteur $\vec{R}$ représentant la résistance de l'air est défini par $\vec{R} = -k \vec{V}$ où k est un nombre réel strictement positif. (Voir Figure ci-après)

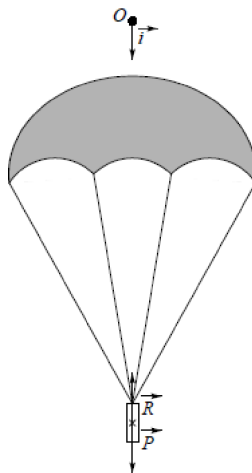
On admet que la position $x(t)$ de l'objet et du parachute vérifie l'équation différentielle suivante :

$$m x''(t) + k x(t) = mg$$

où m est la masse totale de l'objet et du parachute et g le coefficient de l'accélération de la pesanteur.

A partir de l'équation différentielle (E3) donner un programme de résolution numérique qui nous permet de donner comme résultat la distance x de la chute (l'objet+parachute) après 2 minutes avec un pas de 2 seconds.

On prendra : $m=8$ kg, $g=10$ m s⁻², $k=25$ unités SI, position initiale $x(0)=0$ m et vitesse initiale $v(0)=v_0=3,2$ m s⁻¹.



Correction

```
from numpy import arange
from scipy.integrate import odeint
t=arange(0.,121.,2)
X0=0.,3.2
m=8
g=10
k=25
def phi(X,t):
    x, x_prime = X
    return([x_prime , g-((k*x_prime)/m)])
```

```
s=odeint(phi,X0,t)
print(s[-1,0])
```

Exercice 5 : Etude d'un circuit RC

L'équation différentielle qui régit l'évolution de la tension aux bornes du condensateur d'un circuit RC en régime libre s'écrit sous la forme:

$$\frac{dU_c(t)}{dt} + \frac{1}{RC} U_c(t) = 0$$

1. Ecrire une fonction qui implémente la méthode d'Euler-Cauchy d'intégration d'une équation-différentielle du premier ordre.
2. On suppose qu'initialement $U_c(t=0) = 1V$; $R = 1k\Omega$; $C = 1 nF$. Réaliser l'intégration de l'équation-différentielle précédente par la méthode d'Euler-Cauchy entre $t=0$ et $0.1ms$, avec une subdivision de **1000** segments.
3. Donner le code permettant d'effectuer le tracé de la courbe représentative de $U_c(t)$ pour t variant de 0 à 0.1ms.

```
import numpy as np
#(1)Méthode d'Euler-Cauchy:
def Euler(f,a,b,Uo,n):
    x=np.linspace(a,b,n+1)
    y=np.empty(n+1)
    y[0]=Uo #Premier élément de la liste c'est Uo tension initiale
    h=(b-a)/n
    for i in range(n):
        y[i+1]=y[i]+h*f(x[i],y[i])
    return(x,y)

#(2)Résolution
R=1000
C=1e-9 #notation scientifique de 10**(-9)

#Définition de la fonction second membre:
f=lambda x,y:-y/(R*C) #Définition de {Uc/(RC)}

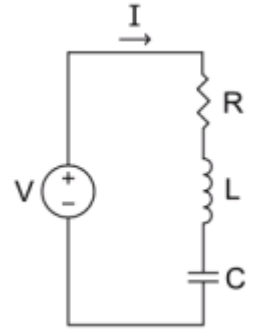
#On appelle la fonction Euler:
(x,y)=Euler(f,0,0.0001,1,1000) #n=1000, de 0 à 0,1ms avec Uc(0)=1V

#Tracé:
import matplotlib.pyplot as plt
plt.plot(x,y)
plt.grid()
plt.show()
```

Exercice 6 : Circuit RLC en série.

On considère un circuit électrique constitué d'une source de tension V , d'une résistance R , d'une bobine L et d'une capacité C montés en série.

$$U_L \text{ satisfait l'équation différentielle: } \frac{d^2 u_L}{dt^2} + \frac{R}{L} \frac{du_L}{dt} + \frac{1}{LC} u_L = \frac{d^2 V}{dt^2}$$



Tracer la courbe de la tension aux bornes de la bobine pour $V(t) = \sin(2\pi ft)$ sur un intervalle de temps de 1 seconde (on prendra 10000 points) pour $f = 100\text{Hz}$, $R = 10$,

$L = 870\text{mH}$, $C = 650\text{mF}$. A l'instant initial aux bornes de la bobine la tension est nulle ainsi que sa dérivée.

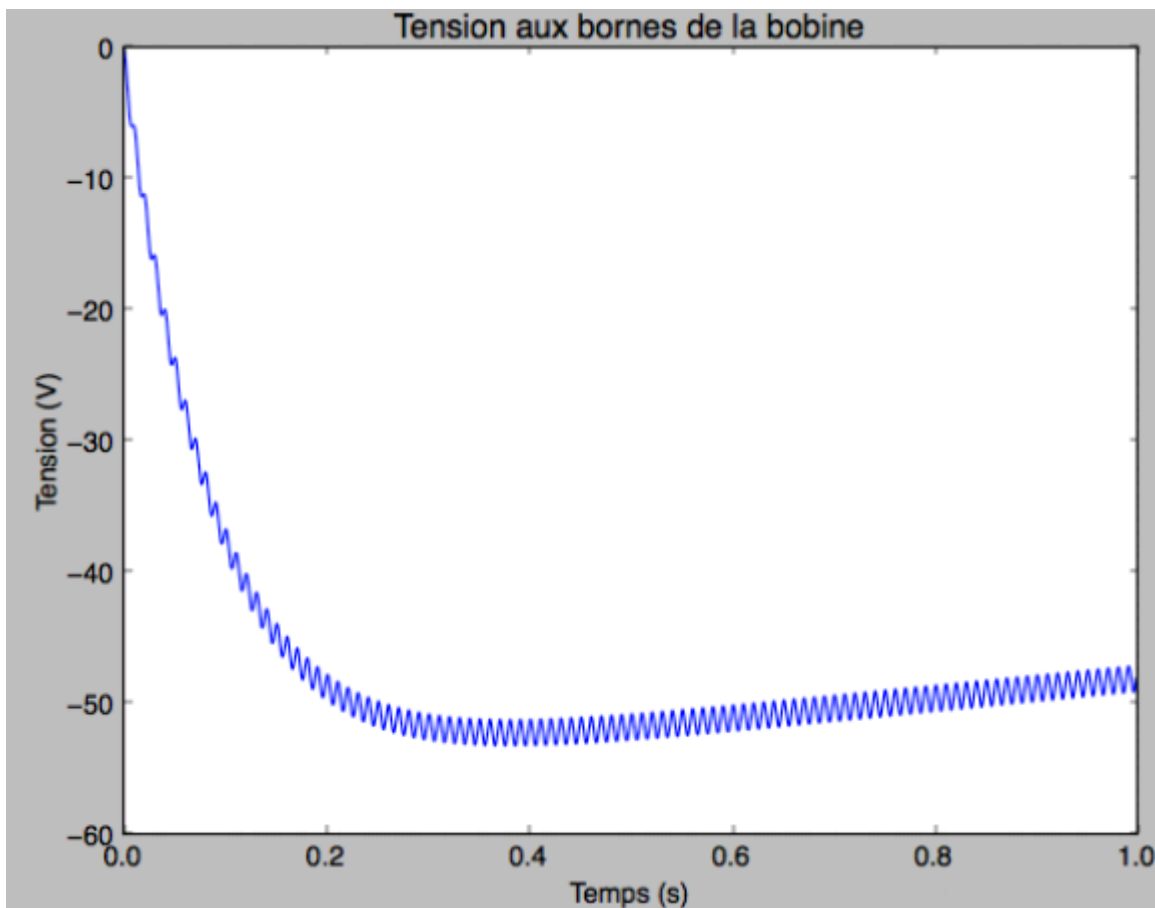
```
#Constantes du problème:
R=10 #R=10Ohms
L=0.870 #L=870mH
C=0.650 #C=650mF
f=100 #f=100Hz

#Import des fonctions nécessaires des modules numpy, scipy
from numpy import linspace, array, sin, pi
from scipy import integrate
import matplotlib.pyplot as plt

#Fonction second membre
def phi(z, t):
    (y, dy) = z
    omega = 2 * pi * f
    return (dy, -omega**2 * sin(omega * t) - R / L * dy - y / (L * C))

#Résolution
N=10000 #Nbre d'échantillons
t=linspace(0,1,N)
y0=array([0,0])
result=integrate.odeint(phi,y0,t)
uL=result[:,0] #On récupère la 1ère colonne

#Tracé
plt.plot(t,uL)
plt.title('Tension aux bornes de la bobine')
plt.xlabel('Temps(s)');
plt.ylabel('Tension(V)')
plt.show()
```



Exercice 7 : Pendule simple

Soit donc un pendule de masse **m**, de longueur **l**, qui oscille dans l'air. On décide de ne pas négliger le frottement fluide de l'air sur la masse oscillante. Et l'on veut étudier l'évolution de θ , l'angle que fait le pendule avec la verticale.

Dans le cas d'amortissement nul, l'équation différentielle est très simple, comme nous l'avons vu ci-dessus :

$$\frac{d^2\theta}{dt^2} = -\omega^2 \sin(\theta)$$

Dans le cas d'un amortissement fluide, un terme supplémentaire s'ajoute à droite de l'équation pour donner:

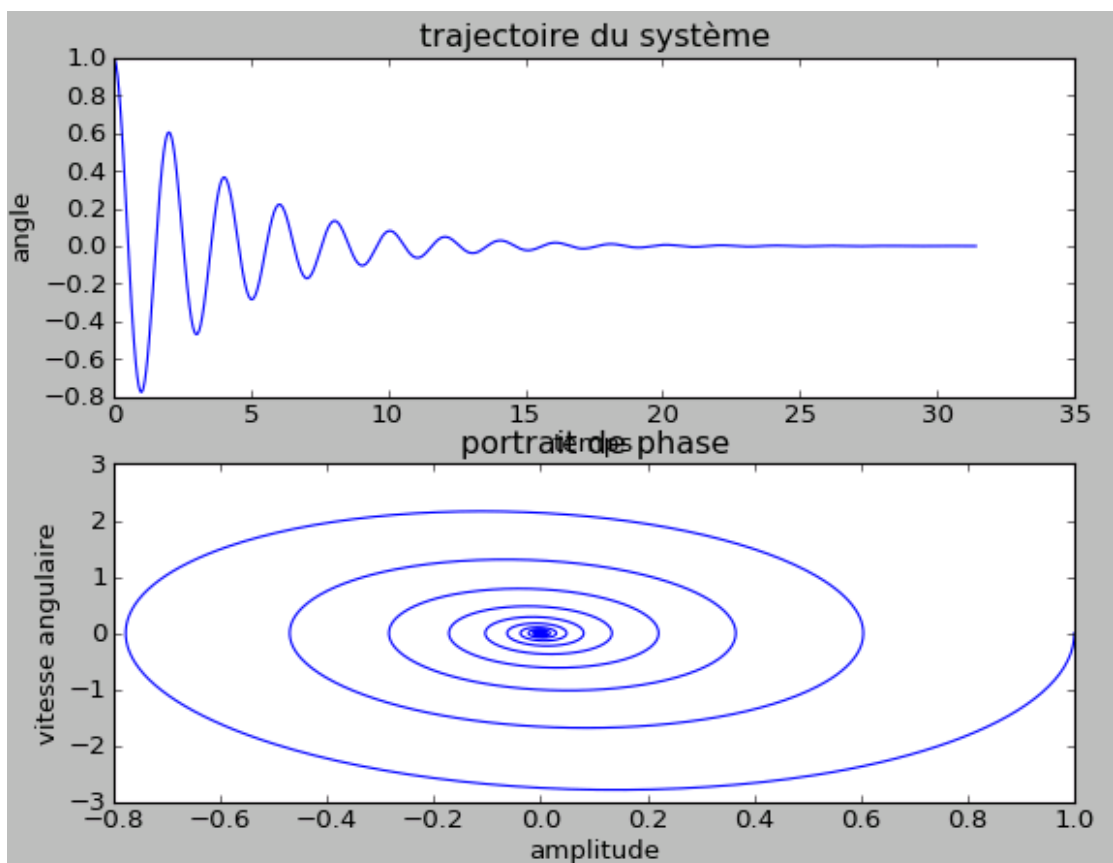
$$\frac{d^2\theta}{dt^2} = -\omega^2 \sin(\theta) - k \frac{d\theta}{dt}$$

K est une constante, qui possède la dimension de l'inverse d'un temps, et qui vaut $K = k/ml^2$, avec k le coefficient d'amortissement fluide.

On se propose d'écrire un petit [programme Python](#) qui va simuler le comportement de notre pendule et aussi d'en étudier les caractéristiques.

Ecrire le code python qui affiche les tracés suivants :

Avec $\theta_0 = 1 \text{ rad.s}^{-1}$, une vitesse initiale nulle et un coefficient d'amortissement égal à **0.5**:



```
# programme de simulation d'un pendule simple amorti

# importation des librairies
from scipy.integrate import odeint # on importe uniquement ce dont on
a besoin
from pylab import * # fonction de tracé de courbe

# Paramètres physiques de l'expérience
l = 1          # longueur en m
g = 9.81 # accélération de la pesanteur en m.s-2
Omega2 = g/l  # pulsation du pendule

# conditions initiales de l'expérience
theta0 = 1   # Angle initial (en rd)
vtheta0 = 0 # Vitesse angulaire initiale (en rd/s)
K = 0.5      # "Taux d'amortissement (en s-1): ")
C0 = [theta0,vtheta0] # angle initial et vitesse angulaire initiales

#définition du système différentiel
def Pendule(z,t):
    y,dy=z
    return (dy, -Omega2*y - K*dy)
```

```

# définition du vecteur temps de l'expérience
t0 = 0.0
tmax = 10*pi
nbpoints = 5000
t = linspace(t0, tmax, nbpoints)

# intégration du système
theta = odeint(Pendule,C0,t)

# affichage des résultats

# trajectoire du système
subplot(211)
plot(time, theta[:,0])
xlabel('temps')
ylabel('angle')
title("trajectoire du système")

# portrait de phase
subplot(212)
plot(theta[:,0],theta[:,1])
title("portrait de phase")
ylabel('vitesse angulaire')
xlabel('amplitude')
show()

```

Exercice 7 : Pendule simple

Pour le pendule simple (non amorti) l'angle avec la verticale suit une équation différentielle de la forme :

$$\ddot{\theta} = -\sin(\theta) \quad [1]$$

Q1: Résoudre numériquement et représenter la solution vérifiant les conditions initiales :

$$\theta(0) = 0.2 \text{ et } \dot{\theta}(0) = 0.2$$

Q2: Dans le cas des petits angles, on a $\sin(\theta) \approx \theta$, et l'on retrouve l'équation différentielle d'un oscillateur harmonique :

$$\ddot{\theta} = -\theta \quad [2]$$

Résoudre numériquement et représenter la solution vérifiant les conditions initiales $\theta(0) = 0.2$ et $\dot{\theta}(0) = 0.2$.

Y a-t-il une différence importante entre les solutions de [1] et [2] ?

Q3: En prenant toujours $\theta(0) = 0$, représenter les solutions de [1] vérifiant les conditions initiales :

$$\dot{\theta}(0) = \{1, 1.9, 2, 2.1, 3\}.$$

Q4: Avec les différentes conditions initiales vues plus haut, représenter le portrait de phase c'est-à-dire l'ensemble des points des couples $(\theta, \dot{\theta})$.

Q5 : Reprendre les questions précédentes avec un pendule amorti : $\ddot{\theta} = -\sin(\theta) - k\dot{\theta}$

On pourra prendre $k = 1$ dans un premier temps, et éventuellement jouer sur la valeur de ce paramètre (qui modélise l'amortissement).