TP-info 1 : Correction

III.Exploitation (TD)

1. Chute d'une goutte d'eau (application directe)

On rappelle l'équation différentielle qui régit l'évolution temporelle de la norme de la vitesse de chute de la goutte : $\frac{dv}{dt}+\frac{\lambda}{m}v(t)=g$. Tracer $v(t)$ par une résolution numérique faisant appel à la méthode d'Euler. On prendra pour condition initiale $v(t=0)=0$.

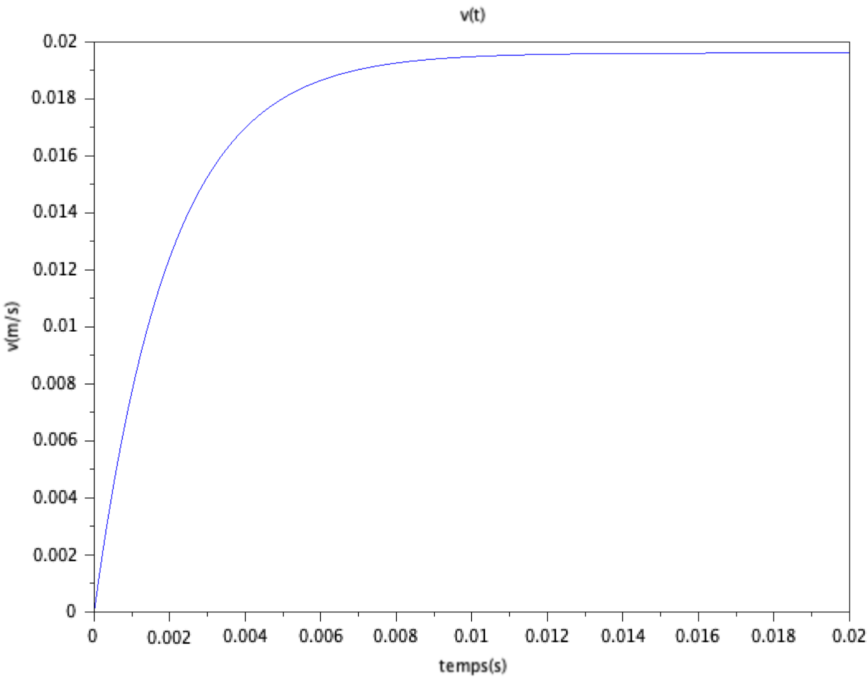
<u>Aide pour l'algorithme</u> : pour pouvoir utiliser la méthode d'Euler, il faut voir l'équation différentielle précédente sous la forme $dv=\left(g-\frac{\lambda}{m}v(t)\right)dt$ *puis on utilise* $v(t+dt)=v(t)+dv$.

```
clear; //effacer toutes les variables en mémoire

v(1) = 0; //vitesse initiale, début de la liste v
g = 9.8; //définition du champ de pesanteur
lambda = 1; //constante de frottement fluide ?
m = 0.002; //masse d'une goutte ?
t(1) = 0; //temps initial, début de la liste t
dt = (m/lambda)/100; //on impose dt << tau

for i=1:1000 //nombre d'itérations de la méthode d'Euler : on va jusqu'à tfin = n*dt
    v(i+1)=v(i)+(g-lambda*v(i)/m)*dt; //évolution de v(t) par la loi physique
    t(i+1)=t(i)+dt; //incrémentation du temps pour pouvoir tracer v(t) ultérieurement
end

plot(t,v) //tracer v(t)
xlabel("temps(s)")
ylabel("v(m/s)")
title("v(t)")
```



2. Oscillateur harmonique

Un point matériel M de masse m est accroché à l'extrémité d'un ressort de raideur k et de longueur à vide ℓ_0 dont l'autre extrémité est fixée au point O. Les forces de frottement fluide et solide sont négligées. On appelle (Ox) l'axe vertical. L'équation différentielle régissant l'évolution temporelle est : $\ddot{x}+\frac{k}{m}x(t)=\frac{k}{m}l_0$.
Tracer $x(t)$ par une résolution numérique faisant appel à la méthode d'Euler. On prendra pour condition initiale $x(t=0)=L$ et $\dot{x}(t=0)=0$.

<u>Aide à l'algorithme</u> :

- *pour pouvoir utiliser la méthode d'Euler, il faut voir l'équation différentielle précédente sous la forme* $dx'=\frac{k}{m}(l_0-x(t))\,dt$;
- *du coup il va falloir créer 3 listes ici : t, x et x' ;*
- $x'(t+dt)=x'(t)+dx'$;
- $x(t+dt)=x(t)+dx$ *avec* $dx=x'(t)\times dt$ *(c'est encore Euler) donc on écrit* $x(t+dt)=x(t)+x'(t)\times dt$.

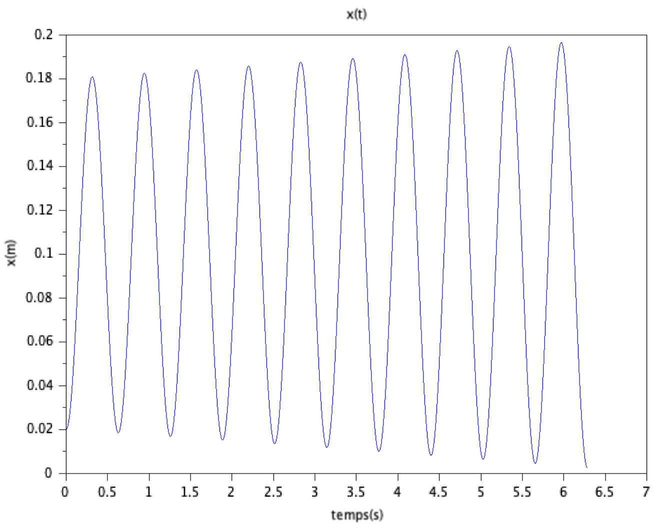
```
clear; // effacer toutes les variables en mémoire

x(1) = 0.02; //position initiale ; début liste x en m
t(1) = 0; //temps initial ; début liste t
vx(1) = 0; //vitesse initiale ; début liste vx
m = 0.01; //masse en kg
k = 1; //constante de raideur du ressort
l0 = 0.1; //longueur à vide du ressort
dt = 2*%pi*sqrt(m/k)/1000; //dt<<T0=2pi/w0

for n=1:10000
    vx(n+1) = vx(n)+(k/m)*(l0-x(n))*dt; //évolution de vx
    x(n+1) = x(n)+vx(n)*dt; //évolution de x
    t(n+1) = t(n)+dt;
end

plot(t,x)
xlabel("temps(s)")
ylabel("x(m)")
title("x(t)")
```

<u>Important</u> : afin d'observer plusieurs périodes (entre 5 et 10 par exemple), le temps de calcul est très long (imaginez si on veut en voir une centaine ou plus) … Et on voit que le calcul approché de l'algorithme diverge ! Ceci est dû à la méthode d'Euler qui est trop approximative pour une telle équation différentielle (qui n'est pourtant qu'un oscillateur harmonique). Le but du TD2 est de résoudre ces 2 problèmes.



3. Lancer (pour aller plus loin …)

Un point matériel M de masse m est lancé depuis un point O du sol avec une vitesse initiale $\vec{v}_0$ dans le plan (xOy) et faisant un angle α avec l'axe horizontal (Ox). (Oy) est verticale ascendante. On suppose que les frottements sont nuls dans un premier temps.
On rappelle les équations (qui peuvent être vues comme des équations différentielles du second ordre …) qui régissent l'évolution temporelles de $x(t)$ et $y(t)$: $\begin{cases}\ddot{x}=0\\ \ddot{y}=-g\end{cases}$.

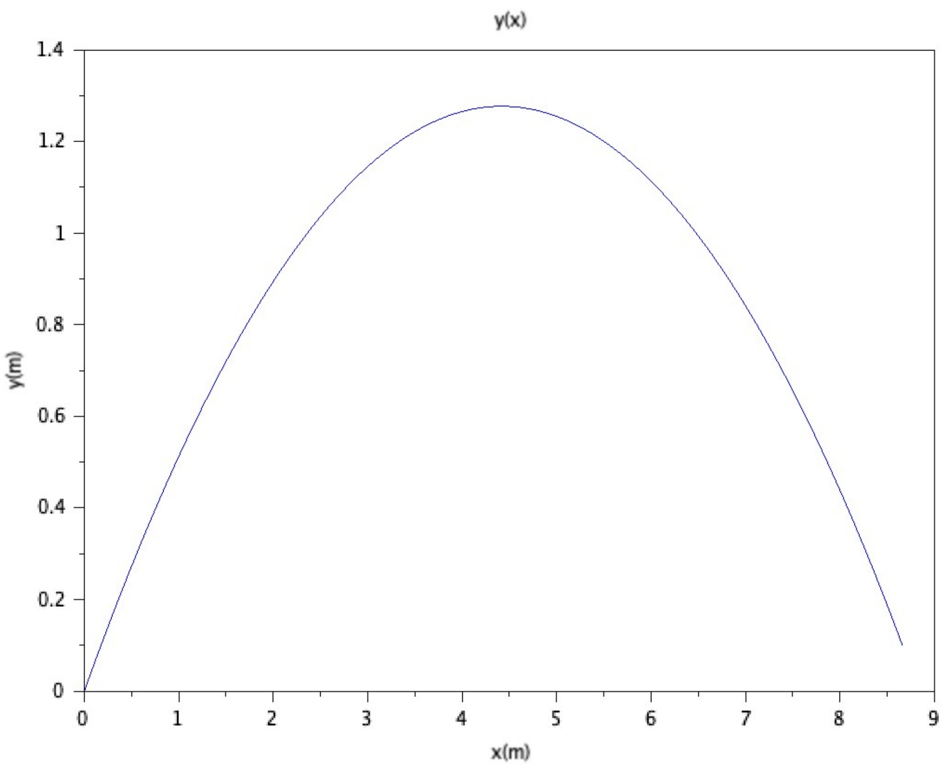
1. Tracer l'équation de la trajectoire $y=f(x)$ en utilisant la méthode de l'Euler.

```
clear; // effacer toutes les variables en mémoire

x(1) = 0; //position initiale en x ; début liste x
y(1) = 0; //position initiale en y ; début de liste y
t(1) = 0; //temps initial ; début liste t
g = 9.81; //définition du champ de pesanteur
v0 = 10; //norme de vitesse initiale
alpha = 30*%pi/180; // angle initial en rad
m = 10; //en kg mais ne sert à rien ici
dt = 1/1000; //dt très petit ... devant quoi ?!
vx = v0*cos(alpha); //définition de vx qui n'évolue pas ...
vy(1) = v0*sin(alpha); //définition de vy ; début liste vy

for n=1:1000
    vy(n+1) = vy(n)-(g)*dt; //évolution de vy par la loi physique
    y(n+1) = y(n)+vy(n)*dt; //évolution de y par la loi physique
    x(n+1) = x(n)+vx*dt; //évolution de x par la loi physique
    t(n+1) = t(n)+dt; //ne sert à rien ici (sauf si vous voulez tracer x(t) ou
y(t))
end

plot(x,y) //tracer la trajectoire y(x)
xlabel("x(m)")
ylabel("y(m)")
title("y(x)")
```



2. On rajoute une force de frottement fluide $\vec{F}_f = -\alpha \vec{v}$ donc les équations qui régissent l'évolution temporelle deviennent : $\begin{cases} \ddot{x} = -\dfrac{\alpha}{m}\dot{x} \\ \ddot{y} = -g - \dfrac{\alpha}{m}\dot{y} \end{cases}$. Tracer l'équation de la trajectoire $y = f(x)$.

```
clear; // effacer toutes les variables en mémoire

x(1) = 0;
y(1) = 0;
t(1) = 0;
g = 9.81;
v0 = 10;
alpha = 30*%pi/180;
lambda = 10; // coefficient de frottement fluide
m = 10;
dt = 1/1000; //très petit devant ? (ici on sait : tau = m/alpha)
vx(1) = v0*cos(alpha); //ici il y a évolution de vx ; début de liste
vy(1) = v0*sin(alpha);

for n=1:1000
    vy(n+1) = vy(n)-(g+lambda*vy(n)/m)*dt;
    y(n+1) = y(n)+vy(n)*dt;
    vx(n+1) = vx(n)-lambda*vx(n)/m*dt; //évolution de vx par la loi physique
    x(n+1) = x(n)+vx(n)*dt;
    t(n+1) = t(n)+dt; //ne sert à rien ici
end

plot(x,y)
xlabel("x(m)")
ylabel("y(m)")
title("y(x)")
```