ATS

Jules Ferry

TP-info 2 : Utilisation du solveur d'équation différentielle intégré à Scilab ; corrigé

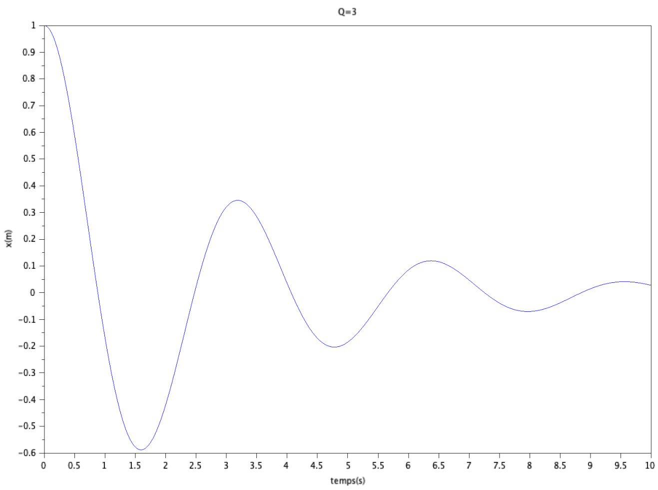
Info

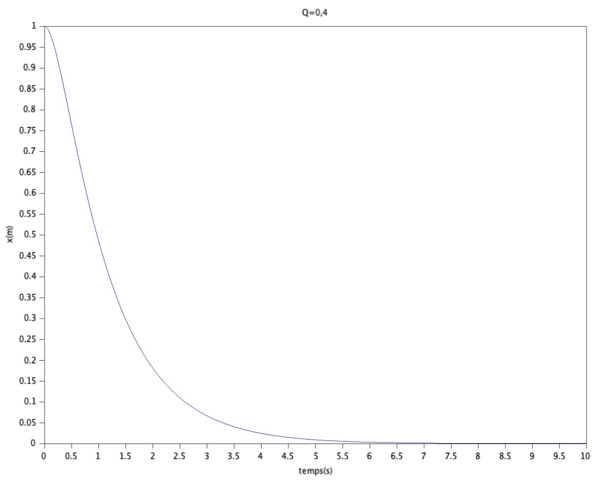
III.Exploitation (TD)

1. Oscillateur amorti

Tracer $x(t)$ pour un oscillateur amorti de pulsation propre $\omega_0=2\,rad.s^{-1}$, de facteur de qualité $Q=3$ et de position à l'équilibre $x_{eq}=0$.

```
clear;

w0 = 2; //pulsation propre
Q=3; //facteur de qualité

function f=Xprime(t, X)

f(1)=X(2)    //1ère composante de X'= 2ème composante de X
f(2)=-w0^2*X(1)-w0/Q*X(2)    // 2ème composante de X' = -w0^2.1ère composante de X

endfunction

X0 = [1;0]; // C.I. rentrées comme un vecteur
//alternative : x0=1; v0=0; X0=[x0;v0]

t0 = 0;
t = linspace(0,10,1000);
X = ode(X0,t0,t,Xprime);

clf();
plot(t,X(1,:)) //tracer x(t)
xlabel("temps(s)")
ylabel("x(m)")
title("Q=3")
```



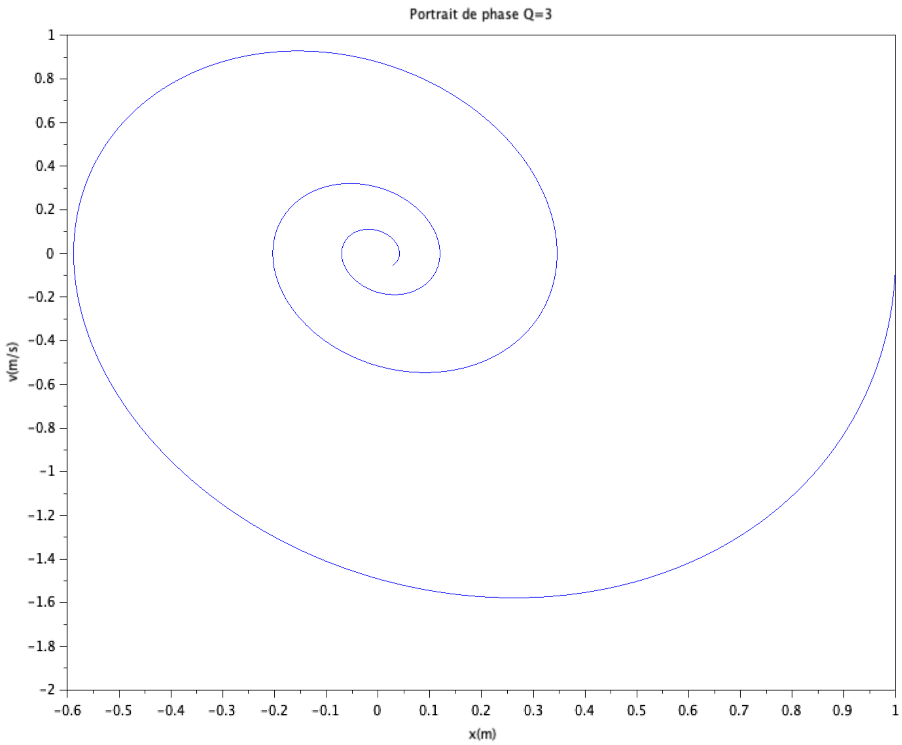


Pour $Q>0{,}5$, on observe un régime pseudo-périodique ; pour $Q<0{,}5$, on observe un régime apériodique.

2. Portrait de phase

Réaliser une trajectoire de phase de l'oscillateur amorti précédent.

```
clear;

w0 = 2; //pulsation propre
Q=3; //facteur de qualité

function f=Xprime(t, X)

f(1)=X(2)    //1ère composante de X'= 2ème composante de X
f(2)=-w0^2*X(1)-w0/Q*X(2)    // 2ème composante de X' = -w0^2.1ère composante de X

endfunction

X0 = [1;0]; // C.I. rentrées comme un vecteur
//alternative : x0=1; v0=0; X0=[x0;v0]

t0 = 0;
t = linspace(0,10,1000);
X = ode(X0,t0,t,Xprime);

clf();

plot(X(1,:),X(2,:)) // tracer le portrait de phase
xlabel("x(m)")
ylabel("v(m/s)")
title("Portrait de phase Q=3")
```



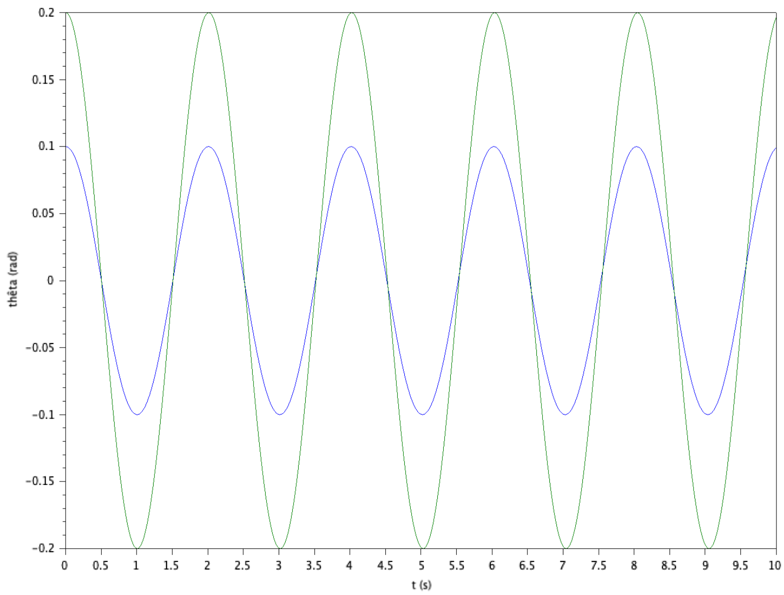
Logique d'après le cours !

3. Le pendule simple non amorti

On rappelle que l'équation différentielle qui régit l'évolution temporelle du pendule est non linéaire : $\ddot{\theta}+\frac{g}{l}\sin(\theta)=0$.

1. Superposer deux courbes $\theta(t)$ pour des conditions initiales différentes mais restant dans le cadre de l'approximation de l'oscillateur harmonique.
   Que peut-on dire des périodes ? Elles sont identiques, on parle d'isochronisme.

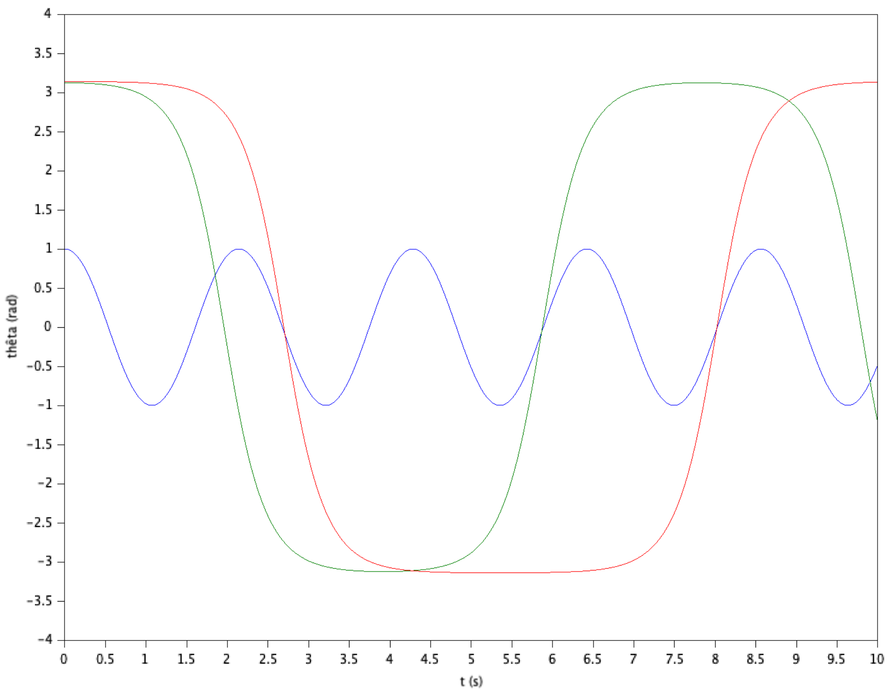
```
clear;

g = 9.8;
L=1

//Dans cet algorithme, X représente thêta ...

function f=Xprime(t, X)

f(1)=X(2)    //1ère composante de X'= 2ème composante de X
f(2)=-(g/L)*sin(X(1))    // 2ème composante de X' = -w0^2.1ère composante de X

endfunction

X10 = [0.1;0]; // C.I.1 rentrées comme un vecteur
X20 = [0.2;0]; // C.I.2 rentrées comme un vecteur

t0 = 0;

t = linspace(0,10,1000);

X1 = ode(X10,t0,t,Xprime);
X2 = ode(X20,t0,t,Xprime);

clf();
plot(t,X1(1,:),t,X2(1,:)) //aux angles << 1rad, l'équation semble linéaire et
w0=sqrt(g/L) : ils possèdent la même période !
xlabel("t (s)")
ylabel("thêta (rad)")
```



Les conditions initiales changent, la période d'oscillation non : elle est fixée par les paramètres de l'oscillateur linéaire : $\omega_0=\sqrt{(g/L)}$ et $T_0=\frac{2\pi}{\omega_0}$; on parle d'isochronisme.

2. Superposer deux courbes $\theta(t)$ pour des conditions initiales différentes dont l'une reste dans le cadre de l'approximation de l'oscillateur harmonique et l'autre est extrême ($\theta(0)=179°$ et $\dot{\theta}(0)=0$ par exemple).

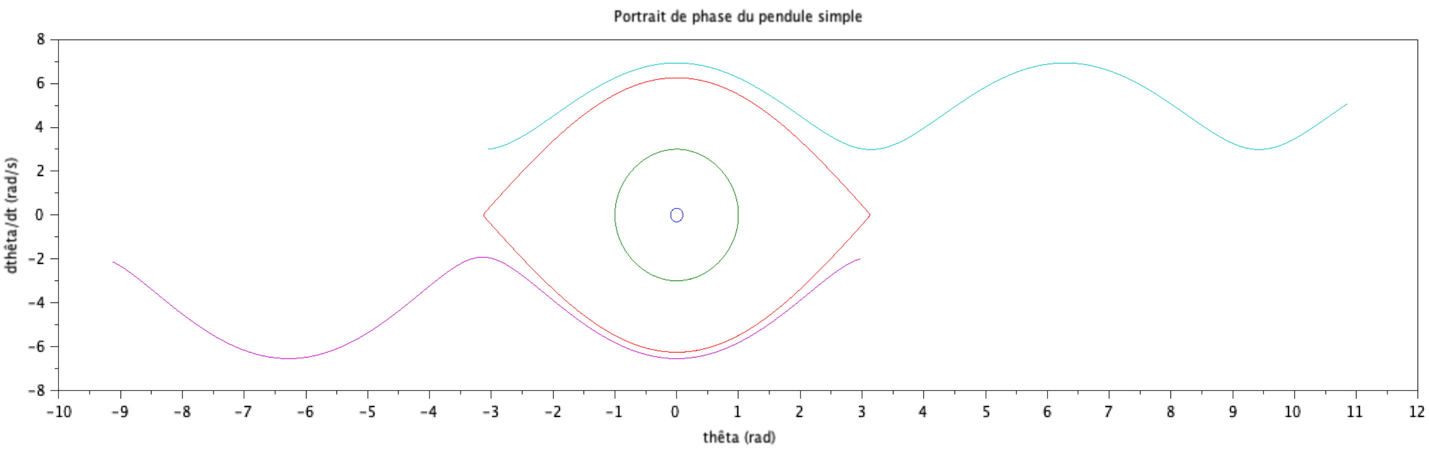
```
clear;

g = 9.8;
L=1

//Dans cet algorithme, X représente thêta ...

function f=Xprime(t, X)

f(1)=X(2)    //1ère composante de X'= 2ème composante de X

f(2)=-(g/L)*sin(X(1))    // 2ème composante de X' = -w0^2.1ère composante de X

endfunction

X10 = [1;0]; // C.I.1 rentrées comme un vecteur
X20 = [179*%pi/180;0];// C.I.2 rentrées comme un vecteur
X30 = [179.9*%pi/180;0]; // C.I.3 rentrées comme un vecteur

t0 = 0;

t = linspace(0,10,1000);

X1 = ode(X10,t0,t,Xprime);
X2 = ode(X20,t0,t,Xprime);
X3 = ode(X30,t0,t,Xprime);

clf();
plot(t,X1(1,:),t,X2(1,:),t,X3(1,:)) // superposition de 3 CI différentes
xlabel("t (s)")
ylabel("thêta (rad)")
```



Il y a clairement non isochronisme : pour un système non linéaire, la période ne dépend pas que des caractéristiques du système mais elle dépend aussi des conditions initiales !
On remarquera aussi qu'un oscillateur non linéaire n'oscille plus en sinusoïde pure (ici, on voit une forme rectangle se profiler).

3. Réaliser un portrait de phase (superposition de plusieurs trajectoires de phase) du pendule simple.

```
clear;

g = 9.8;
L=1

//Dans cet algorithme, X représente thêta ...

function f=Xprime(t, X)

f(1)=X(2)

f(2)=-(g/L)*sin(X(1))

endfunction

X10 = [0.1;0];
X20 = [1;0];
X30 = [179*%pi/180;0];
X40 = [-175*%pi/180;3];
X50 = [170*%pi/180;-2];

t0 = 0;

t = linspace(0,10,1000);
t2 = linspace(0,3,1000);

X1 = ode(X10,t0,t,Xprime);
X2 = ode(X20,t0,t,Xprime);
X3 = ode(X30,t0,t,Xprime);
X4 = ode(X40,t0,t2,Xprime);
X5 = ode(X50,t0,t2,Xprime);

clf();
plot(X1(1,:),X1(2,:),X2(1,:),X2(2,:),X3(1,:),X3(2,:),X4(1,:),X4(2,:),X5(1,:),X5(2,:))
//superposition des différents portraits de phase : X1 et X2 sont dans le domaine
linéaire, on obtient des ellipses, ce n'est évidemment pas le cas pour les autres CI
```



Les 2 premières conditions initiales (bleu foncé et vert sur le graphique) restent dans le domaine linéaire de l'oscillateur : on observe des portraits de phases de forme elliptique. Les autres conditions initiales sortent du domaine linéaire ...