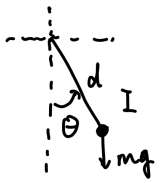
Table of Contents

TD Mécanique numérique

```
% Sébastien SY1924130
```

Retrouver l'équation du mouvement du pendule simple avec les équations de Lagrange



Prenons le pendule simple pour étudier.
θ comme la coordonnée généralisée, I comme le moment d'inertie.

Donc $E_c = \frac{I}{2}\dot{\theta}^2$, $E_p = -mgd\cos\theta$ (Croississons $E_p(\theta = 90°) = 0$)

$\delta W = 0$ à cause de la conservation de l'énergie mécanique

Lagrangien $\mathcal{L} = E_c - E_p = \frac{I}{2}\dot{\theta}^2 + mgd\cos\theta$

Équation de Lagrange :

$$\frac{d}{dt}\left[\frac{\partial L}{\partial \dot{\theta}}\right] - \frac{\partial L}{\partial \theta} = 0$$

$$\Rightarrow I\ddot{\theta} + mgd\sin\theta = 0$$

linéaire $\Rightarrow$ $I\ddot{\theta} + mgd\theta = 0$

Paramètres

```
clear all;
t_start = 0;
```

```
T0 = 3;
delta_t = 0.01;
t = t_start:delta_t:T0;
```

1.1

```
omega_0 = 2*pi;
q_exact = dsolve('D2q+omega_0^2*q=0','q(0)=1,Dq(0)=0')
```

```
Warning: Support of character vectors and strings will be removed in a
 future
release. Use sym objects to define differential equations instead.

q_exact =

exp(-omega_0*t*1i)/2 + exp(omega_0*t*1i)/2
```

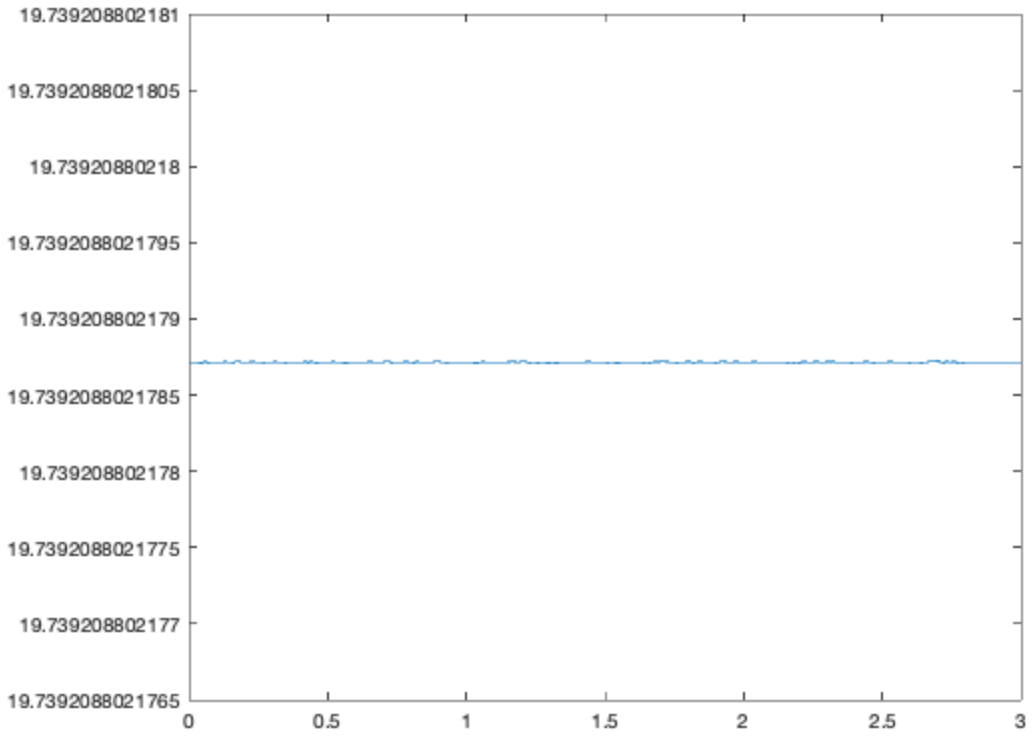
1.2

```
E_star_exact = 1/2 * ( diff(q_exact)^2 + omega_0^2  * q_exact^2)
plot(t, eval(E_star_exact));
hold off

% On peut voir que E_star est constant.
```

```
E_star_exact =

((omega_0*exp(-omega_0*t*1i)*1i)/2 -
 (omega_0*exp(omega_0*t*1i)*1i)/2)^2/2 + (2778046668940015*(exp(-
omega_0*t*1i)/2 + exp(omega_0*t*1i)/2)^2)/140737488355328
```



2.1

$$\begin{vmatrix} q_{j+1} \\ \dot{q}_{j+1} \end{vmatrix} = \begin{vmatrix} q_j \\ \dot{q}_j \end{vmatrix} + \Delta t \times \begin{vmatrix} \dot{q}_j \\ \ddot{q}_j \end{vmatrix} \quad (5)$$

$$\Rightarrow q_{j+1} = q_j + \Delta t \times \dot{q}_j$$

$$\text{et } \dot{q}_{j+1} = \Delta t \times \ddot{q}_j + \dot{q}_j$$

$$= -\omega_0^2 \Delta t\, q_j + \dot{q}_j \text{ , selon (1)}$$

$$\text{Donc (6) } \begin{vmatrix} q_{j+1} \\ \dot{q}_{j+1} \end{vmatrix} = \begin{bmatrix} 1 & \Delta t \\ -\omega_0^2 \Delta t & 1 \end{bmatrix} \begin{vmatrix} q_j \\ \dot{q}_j \end{vmatrix} \text{ est vrai}$$

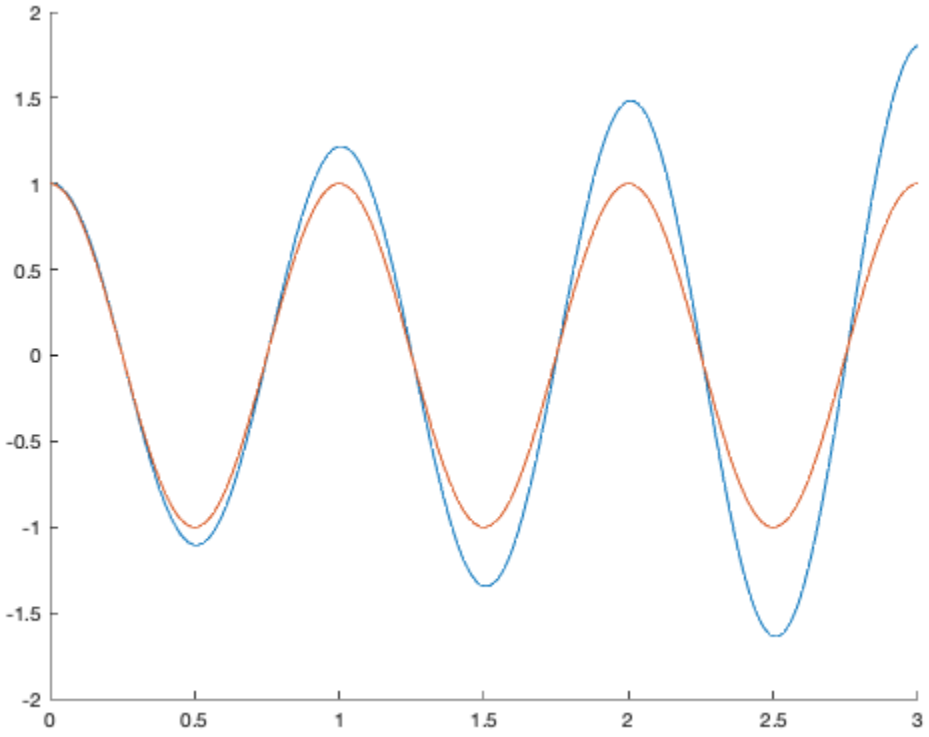
2.2

```
% On utilise la Méthode 2.
```

```
A = [1 delta_t; -omega_0^2*delta_t 1]
U = iterate(A, t);
clf;
hold on;
q_num = U(1,:);
dq_num = U(2,:);
plot(t, q_num);
plot(t, eval(q_exact));
```

```
A =

    1.0000    0.0100
   -0.3948    1.0000
```

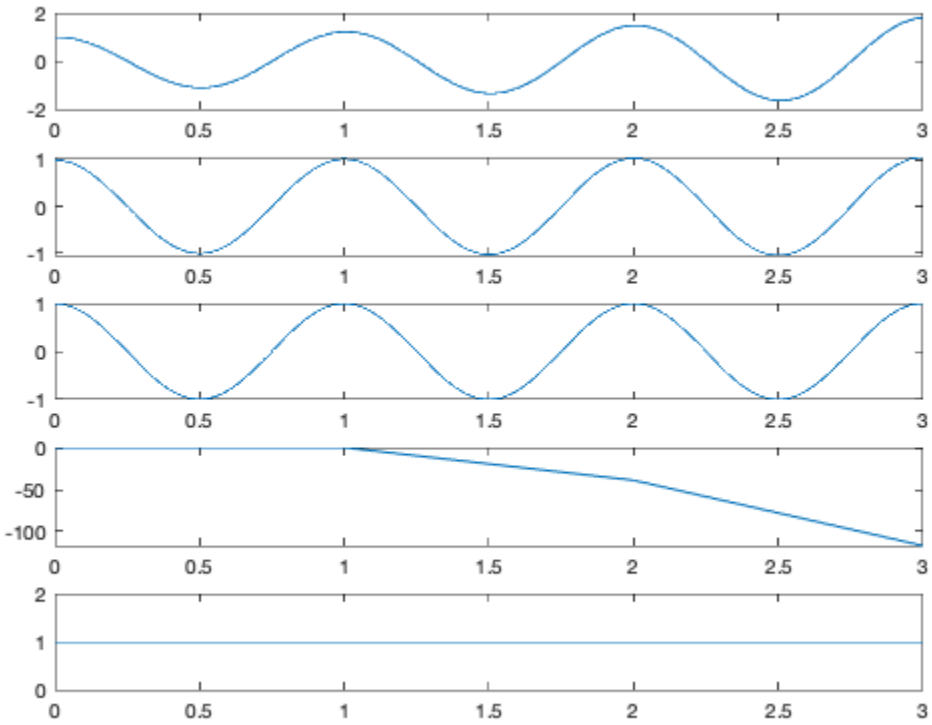


2.3

```
% Ce schéma d'intégration est divergente, voir de la figure.
% Mais plus le pas de temps est petit, plus la divergence est lente.

delta_ts = [0.01 0.001 0.0001 000001];
clf;
hold off;
l = numel(delta_ts);
```

```
for i = 1:l
    delta_t = delta_ts(i);
    t = t_start:delta_t:T0;
    A = [1 delta_t; -omega_0^2*delta_t 1];
    U = iterate(A, t);
    q_num = U(1,:);
    dq_num = U(2,:);
    subplot(l+1,1,i);
    plot(t, q_num);
end
subplot(l+1,1,i+1);
plot(t, eval(q_exact));
```

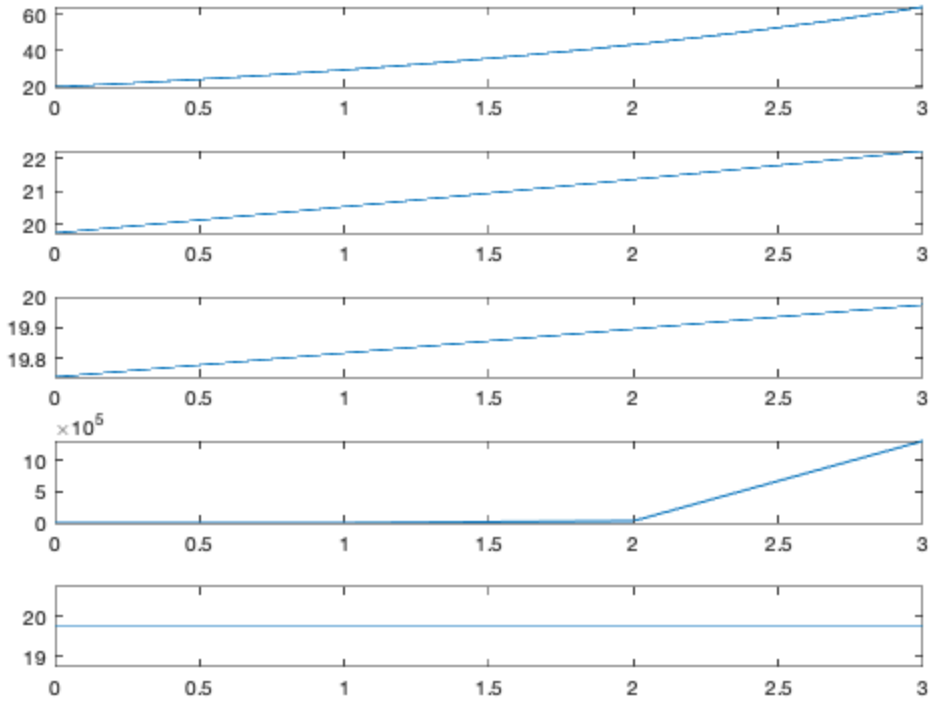


2.4

```
clf;
hold off;
for i = 1:numel(delta_ts)
    delta_t = delta_ts(i);
    t = t_start:delta_t:T0;
    A = [1 delta_t; -omega_0^2*delta_t 1];
    U = iterate(A, t);
    q_num = U(1,:);
    dq_num = U(2,:);
    E_star_num = 1/2 * ( dq_num.^2 + omega_0.^2  .* q_num.^2);
    subplot(l+1,1,i);
```

```
    plot(t, E_star_num);
end
subplot(l+1,1,i+1);
plot(t, eval(E_star_exact));

% E_star est divergent.
% Plus le pas de temps est petit, plus la divergence est lente.
```



2.5

```
syms o0 dt
eig([1 dt; -o0^2*dt 1])

% Les valeurs propres sont des complexes conjugués. Et plus le pas de
 temps
% est grand, plus les modules des valeurs propres sont grandes, mais
 aussi la norme de ces complexes sont toujours supérieures à 1, ce
% qui explique l'instabilité inconditionnelle.


ans =

 1 - dt*o0*1i
 1 + dt*o0*1i
```

Fonctions

```
function [U] = iterate(A,t)
%ITERATE U_j+1 = A*U_j, for j in t
%   Detailed explanation goes here
i = 0;
U = [];
for t_i=t
    if i==0
        U(:,1) = [1;0];
    else
        U(:,i+1)=A*U(:,i);
    end
    i=i+1;
end

end
```

Published with MATLAB® R2019b