Oscillateur conservatif linéaire à un degré de liberté

SY1924111
Justin/Jiahao HU

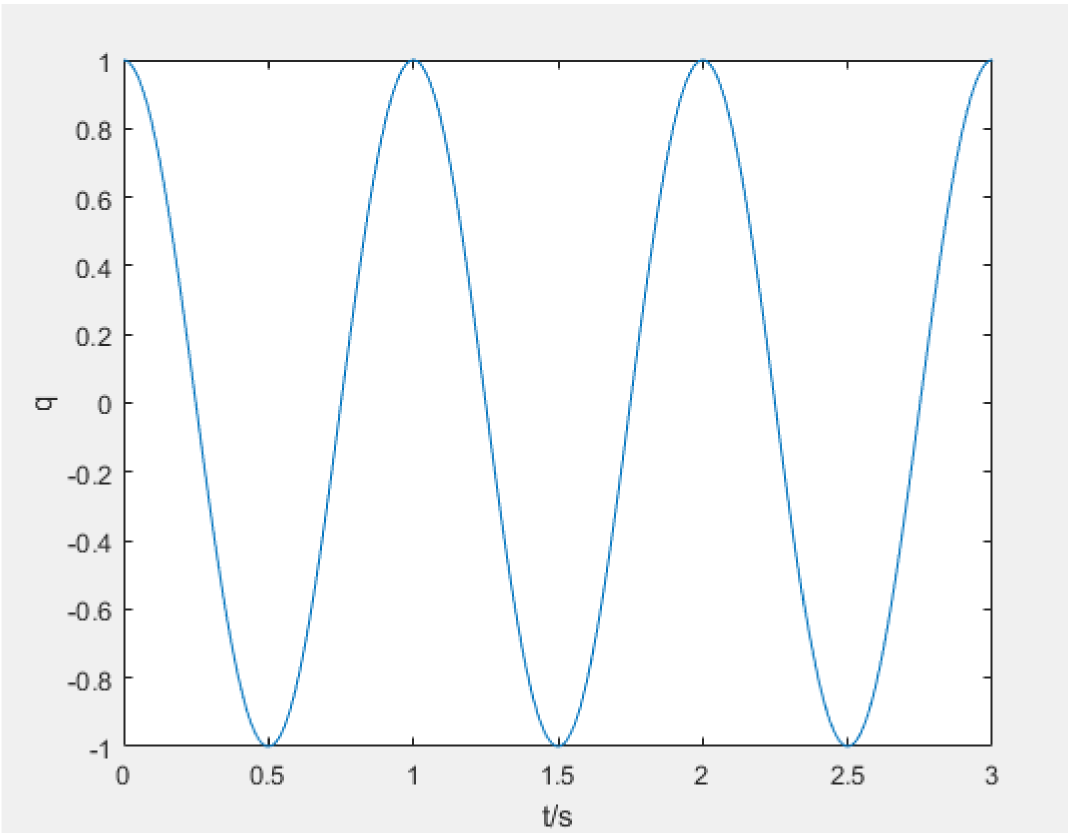
1.

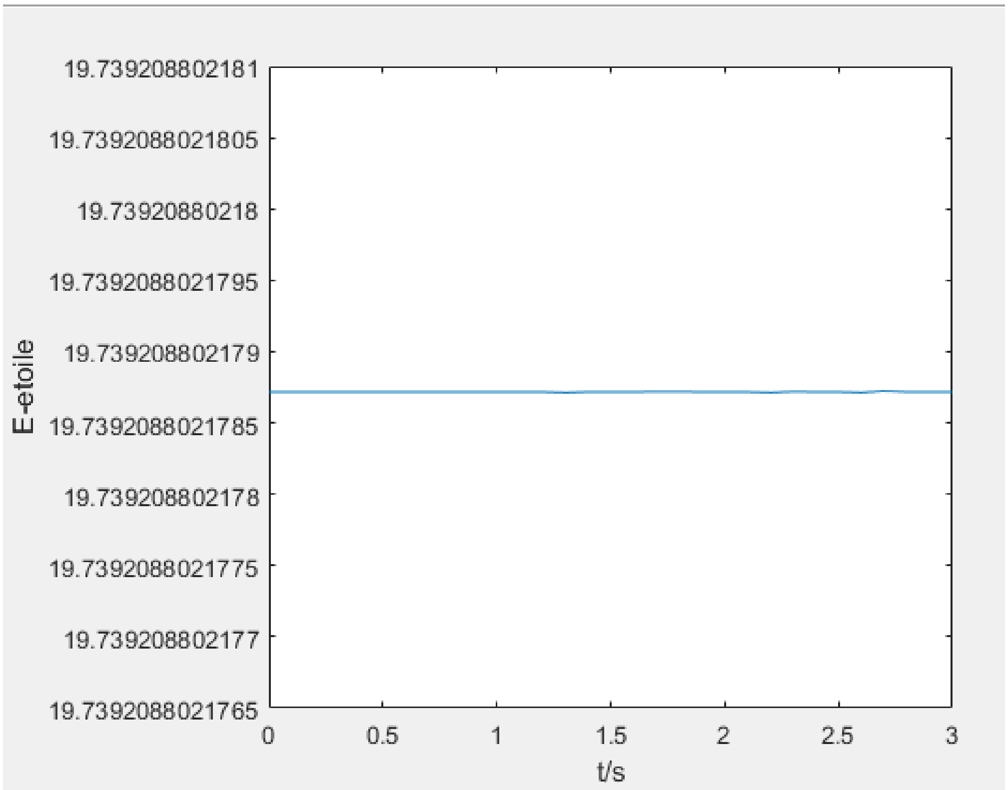
Selon l'équation différentielle et ses conditions initiales, nous pouvons obtenir la solution analytique est :

```
q = sin(2*pi*t)
```

Nous programmons donc dans matlab pour obtenir l'image de q et d'E_etoile :

```
T0 = 3;
t = 0:0.1:T0;
w0 = 2*pi;
q = cos(w0*t);
q_prime = - w0 * sin(w0*t);
plot(t,q);
xlabel('t/s');
ylabel('q');
E_etoile = 1/2*(q_prime.*q_prime+w0*w0*q.*q);
plot (t, E_etoile);
xlabel('t/s');
ylabel('E-etoile');
```





On peut trouver que E_etoile, qui est proportionnelle à l'énergie mécanique de l'oscillateur est constant.

2.1

Dans le relation 5, on remplace $\ddot{q}_j$ par q_j d'après le relation 1, alors on a le relation 6

$$\begin{vmatrix} q_{j+1} \\ \dot{q}_{j+1} \end{vmatrix} = \begin{bmatrix} 1 & \Delta t \\ -\omega_0^2\,\Delta t & 1 \end{bmatrix} \begin{vmatrix} q_j \\ \dot{q}_j \end{vmatrix}$$

2.2

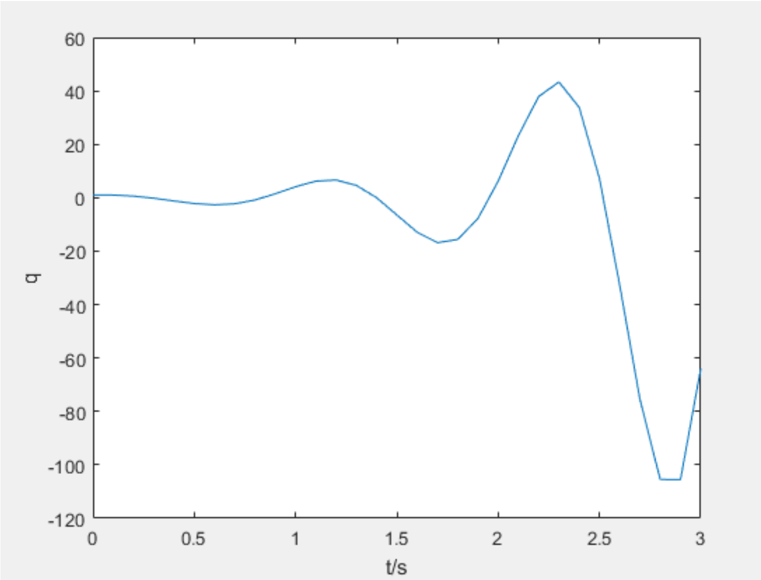
Le code est comme :

```
clear all;
T0 = 3;
w0 = 2*pi;
delta_t = 0.01;
t = 0:delta_t:T0;
A=[1            , delta_t;
   -w0*w0*delta_t, 1       ];
q(1) = 1;
q_prime(1) = 0;
U = [1;0];
for n=1:length(t)-1
```

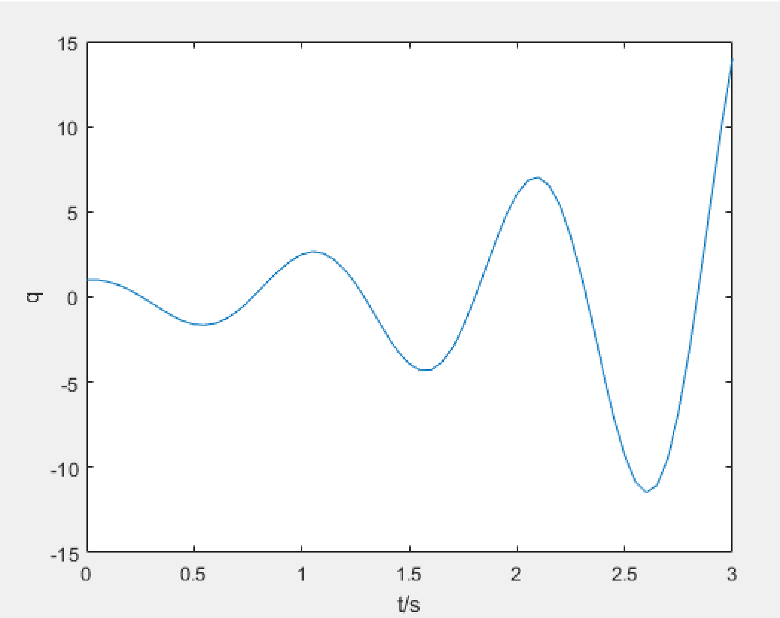
```
    U = A*U;
    q(n+1) = U(1,1);
    q_prime(n+1) = U(2,1);
end
plot(t,q)
xlabel('t/s');
ylabel('q');
E_etoile = 1/2*(q_prime.*q_prime+w0*w0*q.*q);
```

2.3

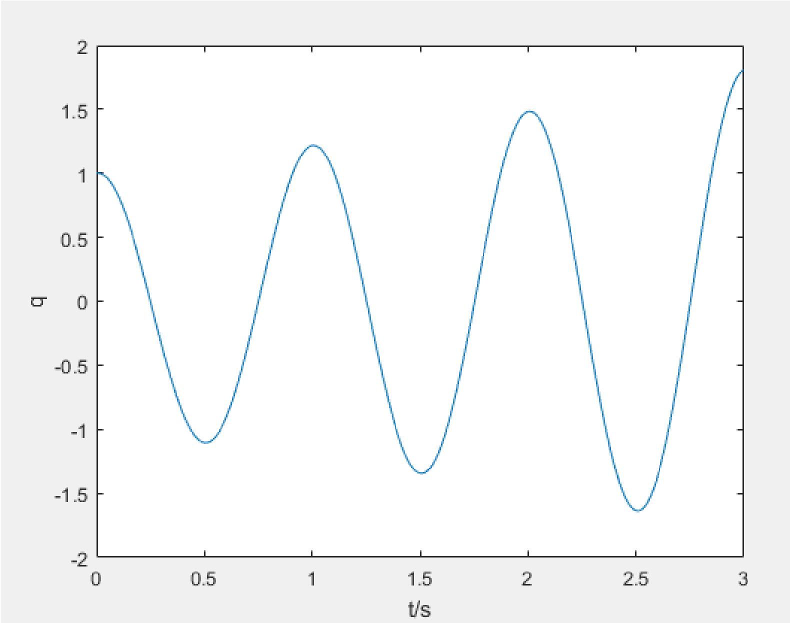
On a testé 3 valeur de delta_t : 0.1, 0.05 et 0.01, les image sont comme :



delta_t = 0.1



delta_t = 0.05

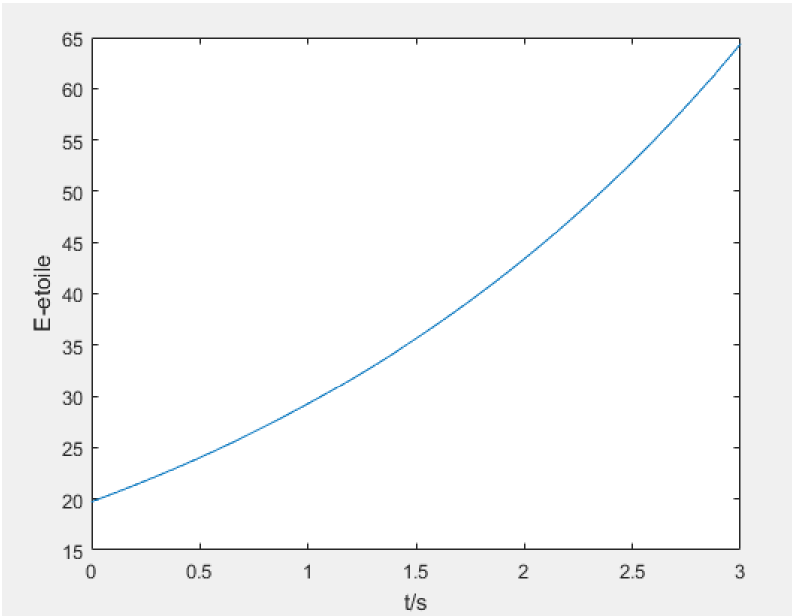


delta_t = 0.01

On peut remarquer que plus le pas de temps delta_t est petit, plus la divergence est lente.

2.4

On voir que E_etoile n'est plus constant, il est comme (delta_t = 0.01) :



Et quand on varie le delta_t, plus le pas de temps delta_t est petit, plus la vitesse d'agumentation est lente

2.5

les valeurs propres de la matrice d'amplification est

$\lambda_1 = 1 - i\omega_0 \Delta t$ et $\lambda_2 = 1 + i\omega_0 \Delta t$

On a donc |λi|<1, donc ce schéma d'EULER explicite est instable.

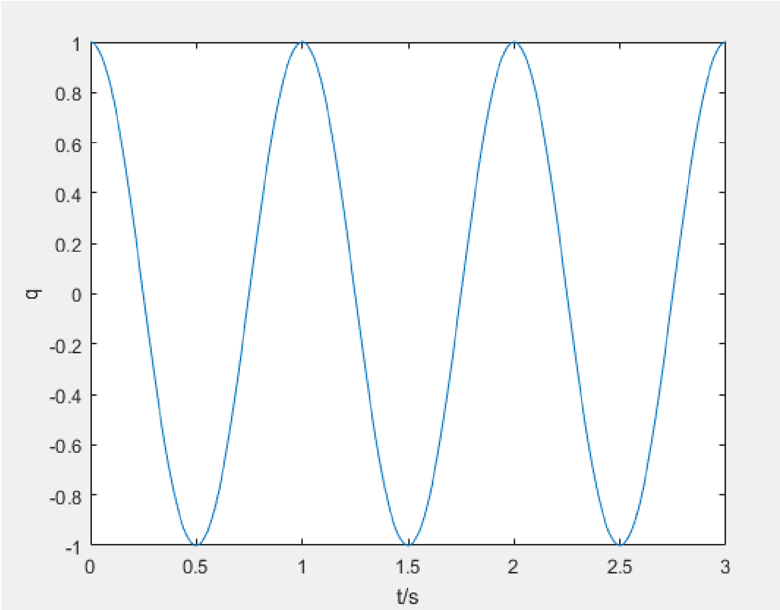
3.1

On programme dans matlab :

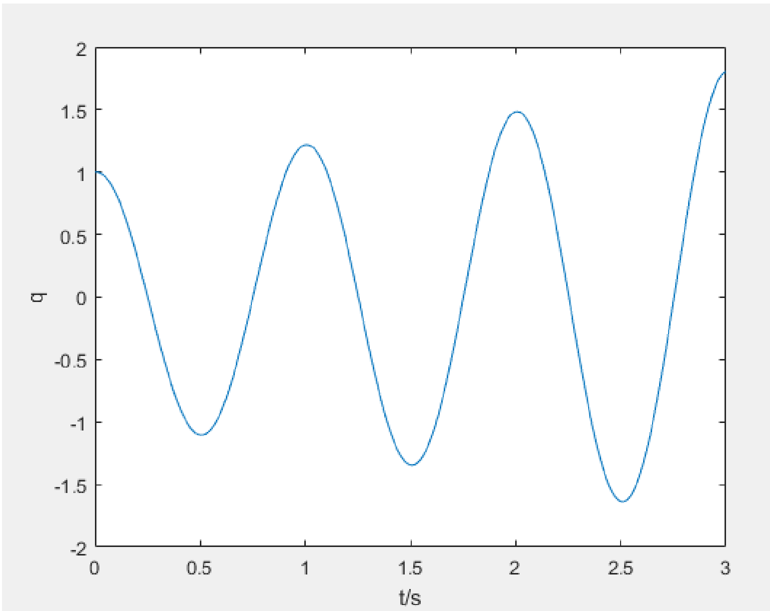
```
clear;
T0 = 3;
w0 = 2*pi;
delta_t = 0.01;
t = 0:delta_t:T0;
q(1) = 1;
q_prime(1) = 0;
for n=1:length(t)-1
    q(n+1) =
q(n)/(1+delta_t^2*w0^2)+delta_t/(1+delta_t^2*w0^2)*q_prime(n);
    q_prime2 = -w0*w0*q(n+1);
    q_prime(n+1) = q_prime(n)+delta_t*q_prime2;
end
plot(t,q)
xlabel('t/s');
ylabel('q');
E_etoile = 1/2*(q_prime.*q_prime+w0*w0*q.*q);
```

3.2

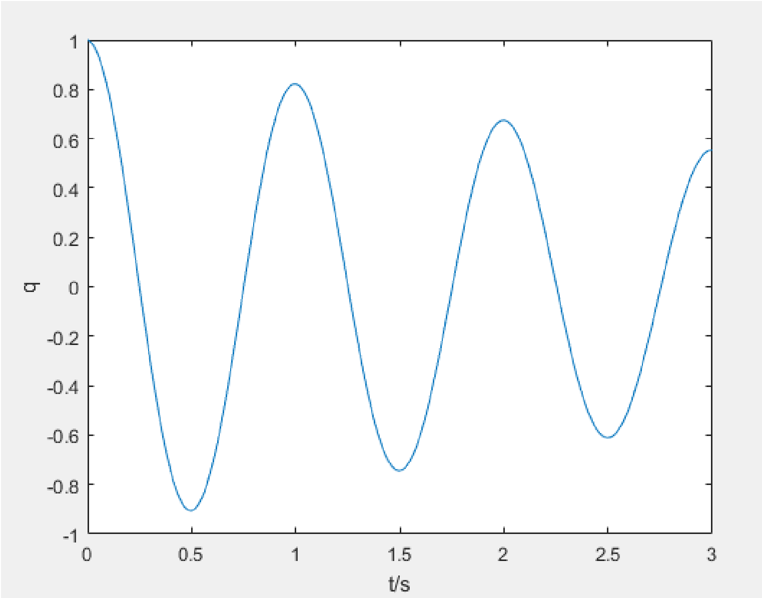
La solution exacte est comme :



La solution EULER explicite est comme :

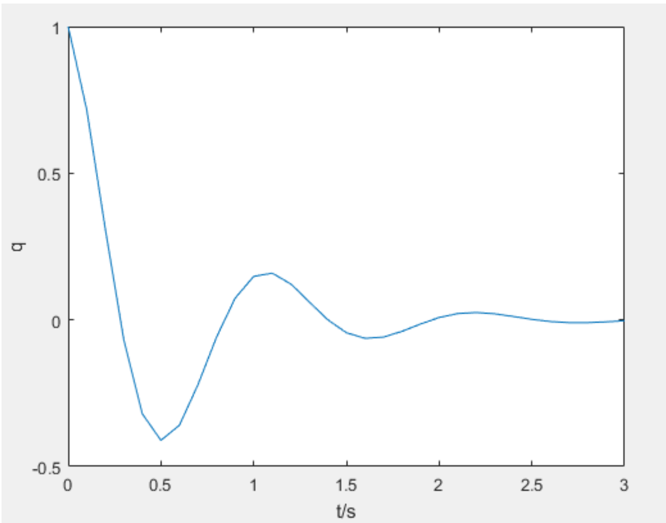


La solution EULER implicite est comme :

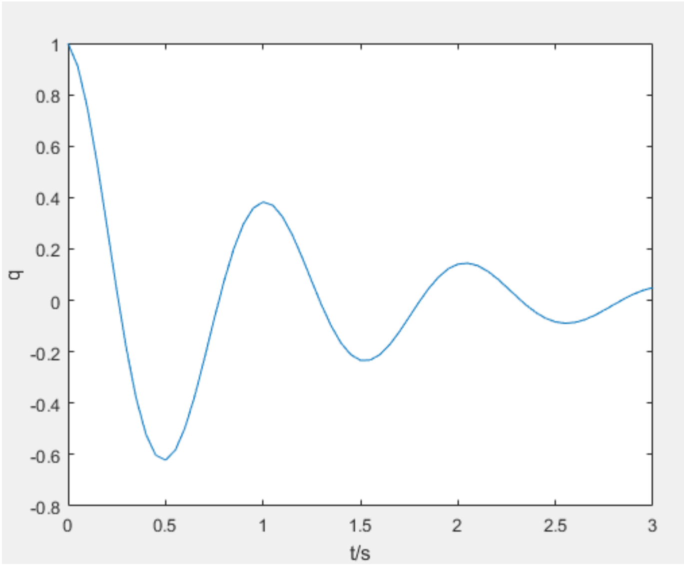


3.3

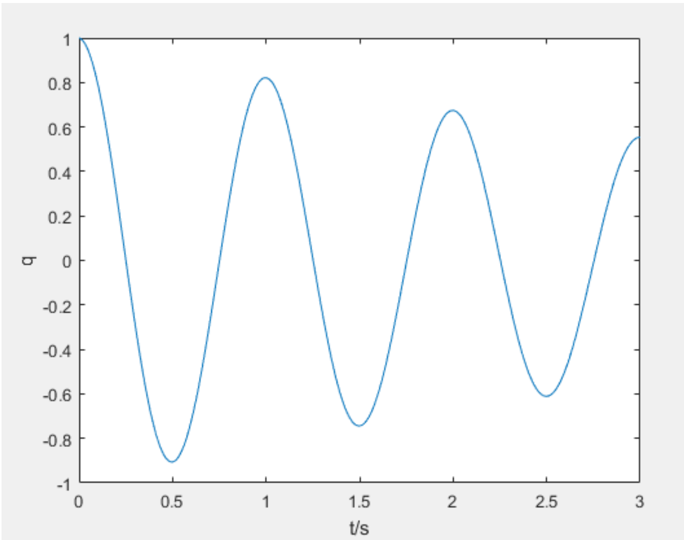
On a testé 3 valeur de delta_t : 0.1, 0.05 et 0.01, les image sont comme :



delta_t = 0.1



delta_t = 0.05

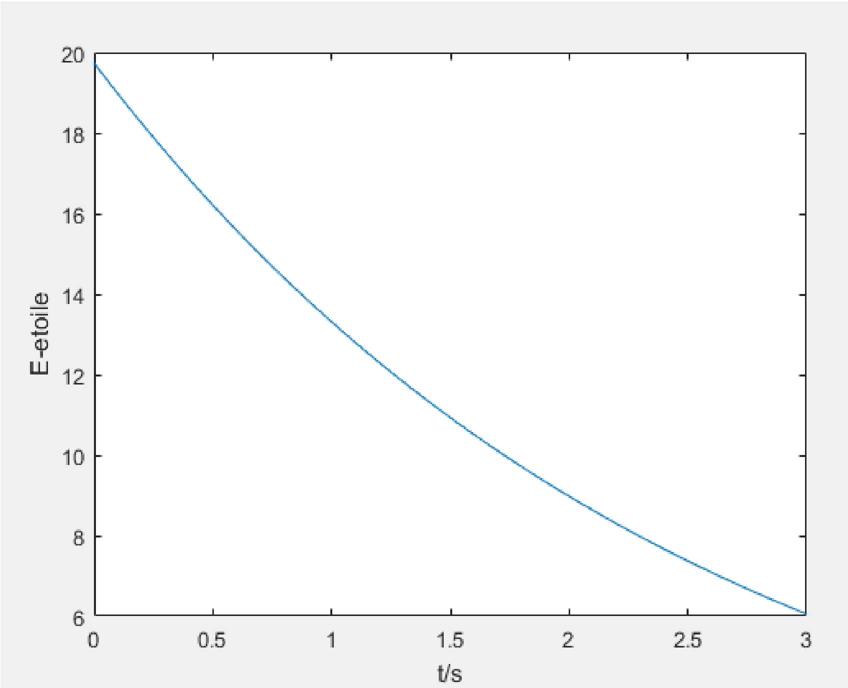


delta_t = 0.01

On peut remarquer que plus le Δt est petit, plus l'atténuation des oscillations est faible.

3.4

On voir que E_etoile n'est plus constant, il est comme (delta_t = 0.01) :



Et quand on varie le delta_t, plus le pas de temps delta_t est petit, plus la vitesse d'diminution est lente

3.5

La matrice d'amplification est comme :

$$\frac{1}{1+\omega_0^{\ 2}\ \Delta t^2}\begin{bmatrix} 1 & \Delta t \\ -\omega_0^{\ 2}\ \Delta t & 1 \end{bmatrix}$$

Et on calcul ses valeurs propres, on obtient :

$$\lambda_1 = \frac{1-i\omega_0\ \Delta t}{1+\omega_0^{\ 2}\ \Delta t^2} \quad \text{et} \quad \lambda_2 = \frac{1+i\omega_0\ \Delta t}{1+\omega_0^{\ 2}\ \Delta t^2}$$

Et donc |λi|<1, donc ce schéma est inconditionnellement stable.

4.1

On suppose U = [q ; q_prime] et F = [0, 1 ; -w0*w0, 0], U_prime = [q_prime ; q_prime2]

Alors on a U_prime = F*U

4.2

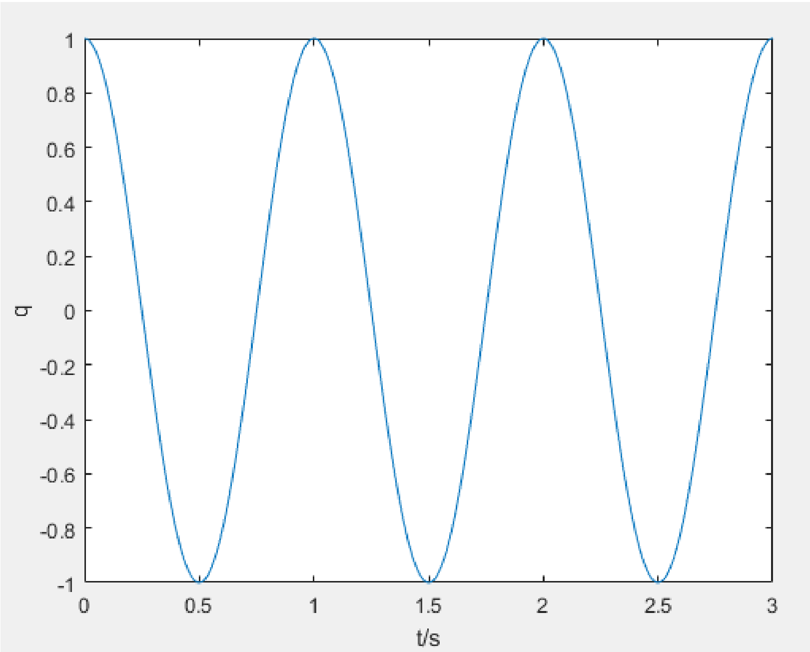
On programme dans matlab :

```
clear all;
T0 = 3;
w0 = 2*pi;
delta_t = 0.01;
t = 0:delta_t:T0;
```

```
F = [0, 1; -w0*w0, 0];
q(1) = 1;
q_prime(1) = 0;
U = [1;0];
for n=1:length(t)-1
    k1 = F*U;
    k2 = F*(U+k1*delta_t/2);
    k3 = F*(U+k2*delta_t/2);
    k4 = F*(U+k3*delta_t);
    K = (k1+2*k2+2*k3+k4)/6;
    U = U+K*delta_t;
    q(n+1) = U(1,1);
    q_prime(n+1) = U(2,1);
end
plot(t,q)
xlabel('t/s');
ylabel('q');
E_etoile = 1/2*(q_prime.*q_prime+w0*w0*q.*q);
```

4.3

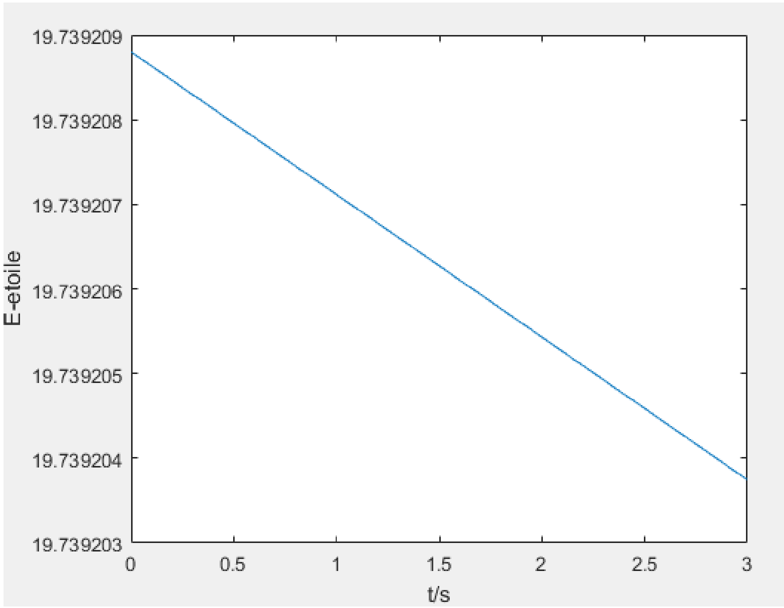
La solution est comme :



Donc on peut voir que la solution de la méthode RUNGE KUTTA est le plus proche de la solution exacte.

4.4

E_etoile est comme (delta_t = 0.01) :



On peut voir même si il n'est pas constant, mais il varie très peu en comparant avec les autres méthodes. Donc la méthode RUNGE KUTTA a la plus grande précision.