Oscillateur conservatif à un degré de liberté

- Le schéma Runge Kutta

Code en Matlab :

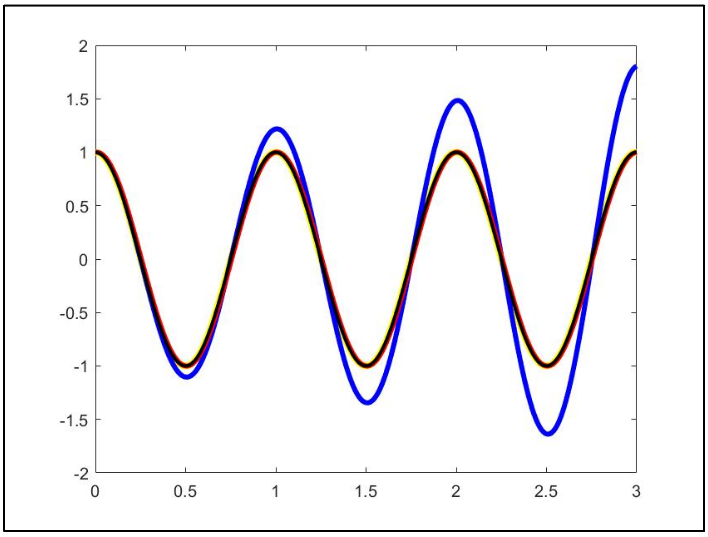
```
dt = 0.04;
t = (0:dt:3)';
np = size(t,1);
q0 = 1;
dq0 = 0;
q = zeros(np,1);
dq = zeros(np,1);
q(1) = q0;
dq(1) = dq0;
qj=[q0;dq0];
for i = 2:np
    tc = t(i-1);
    xc=qj;
    k1=cal_f(xc,tc);
    xc=qj+k1*dt/2;
    k2=cal_f(xc,tc+dt/2);
    xc=qj+k2*dt/2;
    k3=cal_f(xc,tc+dt/2);
    xc=qj+k3*dt;
    k4=cal_f(xc,tc+dt);
    dq=(k1+2*k2+2*k3+k4)/6;
    qj=qj+dq*dt;
    q(i)=qj(1)
    dq(i)=qj(2);
 end
plot(t,q,'b-','Linewidth',3)
```

```
function[dU] = cal_f(U,tc)
    w = 2 * pi;
    dU = zeros(2,1);
    dU(1) = U(2);
    dU(2) = - w*w*U(1);
end
```

Résultat obtenu :

(1) Les valeurs de la solution q(t) obtenu avec le schéma de *Runge Kutta*,

Euler explicite, *Euler implicite* et *solution exacte*

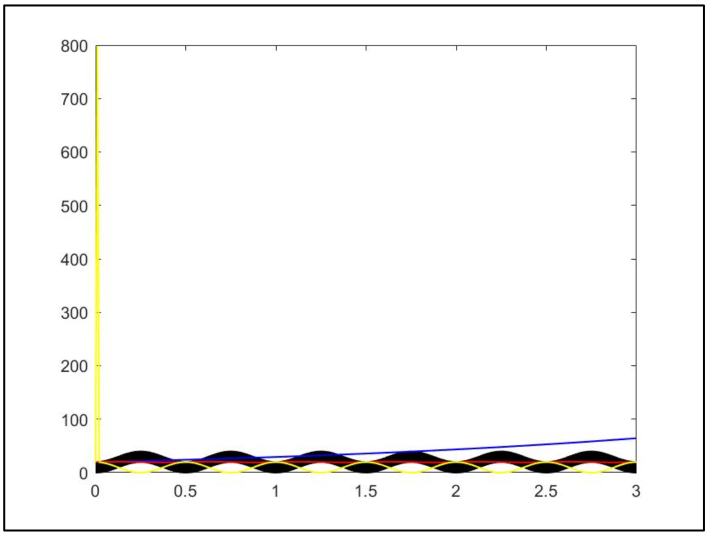


Bleu : Euler explicite
Rouge : Euler implicite
Jaune : Runge Kutta
Noir : solution exacte

C'est les résultats q(t) obtenus par des différents schémas quand on prend $\Delta t = 0.01s$

Les résultats de schémas Euler implicite, Runge Kutta et solution exacte sont presque coïncidents. Mais le résultat de schéma Euler explicite est divergent.

(2) Les valeurs de la quantité E^* obtenu avec le schéma de *Runge Kutta*, *Euler explicite*, *Euler implicite* et *solution exacte*



Bleu : Euler explicite
Rouge : Euler implicite
Jaune : Runge Kutta
Noir : solution exacte

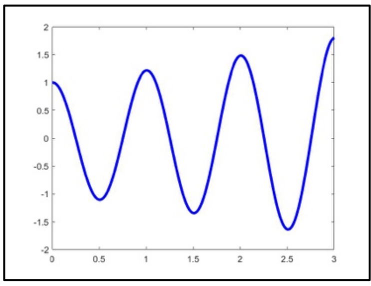
C'est la quantité E^* obtenue par des différents schémas quand on prend $\Delta t = 0.01s$

- Le schéma Euler explicite

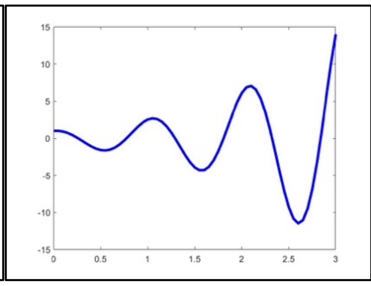
Code en Matlab :

```
dt = 0.05;
T = 3;
q0 = 1;
dq0 = 0;
w0 = 2*pi;
t = (0:dt:T)';
np = size(t,1);
q = zeros(np,1);
dq = zeros(np,1);
ddq = zeros(np,1);
q(1) = q0;
dq(1) = dq0;
ddq(1) = - w0 * w0 * q(1);
for i = 2:np
    q(i) = q(i-1)+ dt * dq(i-1);
    dq(i) = dq(i-1)+ dt * ddq(i-1);
    ddq(i) = - w0 * w0 * q(i);
end
plot(t,q,'b-','Linewidth',3)
```

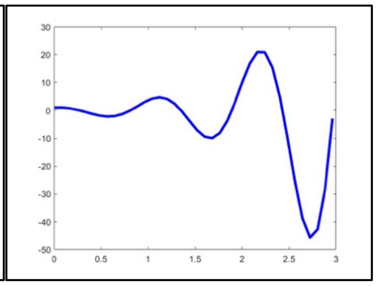
Résultat obtenu :



$\Delta t = 0.01s$



$\Delta t = 0.05s$



$\Delta t = 0.08s$

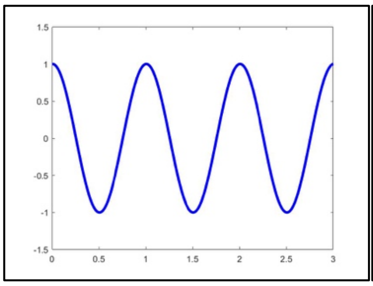
Et quand on change le pas de temps Δt , on peut voir que plus le pas de temps Δt est petit, plus la divergence est lente.

- Le schéma implicite

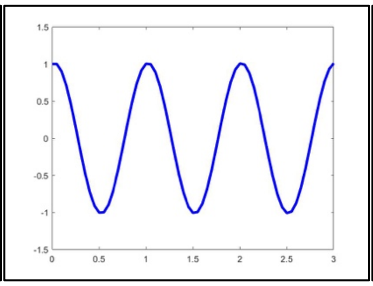
Code en Matlab :

```
dt = 0.05;
T = 3;
q0 = 1;
dq0 = 0;
w0 = 2*pi;
t = (0:dt:T)';
np = size(t,1);
q = zeros(np,1);
dq = zeros(np,1);
ddq = zeros(np,1);
q(1) = q0;
dq(1) = dq0;
ddq(1) = - w0 * w0 * q(1);
for i = 2:np
    q(i) = q(i-1)+ dt * (dq(i-1) - dt * q(i) * w0 * w0);
    ddq(i) = - w0 * w0 * q(i);
    dq(i) = dq(i-1)+ dt * ddq(i);
end
plot(t,q,'b-','Linewidth',3)
```

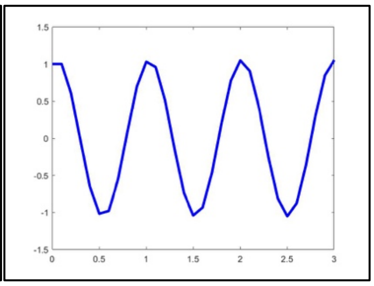
Résultat obtenu :



$\Delta t = 0.01s$



$\Delta t = 0.05s$



$\Delta t = 0.1s$

Et quand on change le pas de temps Δt , on peut voir que plus le pas de temps Δt est petit, plus l'atténuation des oscillations est faible.