Oscillateur conservatif linéaire à un degré de liberté:

1.1 Pour l'équation (1), on peut obtenir:

$$q = A\cos(\omega_0 t + B)$$

avec les conditions aux limites (4), on peut obtenir:

$$q = \cos(\omega_0 t)$$

1.2 On peut calculer E* avec la résultat de 1.1:

$$E^* = \frac{1}{2}\left(\omega_0^2 \sin^2(\omega_0 t) + \omega_0^2 \cos^2(\omega_0 t)\right) = \frac{1}{2}\omega_0^2 = 2\pi^2$$

On trouve que E* est toujour une constante.

2.1 On peut utiliser l'équation (1) dans l'équation (5):

$$\left|\begin{matrix} q_{j+1} \\ \dot{q}_{j+1} \end{matrix}\right. = \left|\begin{matrix} q_j \\ \dot{q}_j \end{matrix}\right. + \Delta t \times \left|\begin{matrix} \dot{q}_j \\ -\omega_0^2 q_j \end{matrix}\right. = \left|\begin{matrix} q_j + \Delta t\,\dot{q}_j \\ -\Delta t\,\omega_0^2 q_j + \dot{q}_j \end{matrix}\right.$$

Alors, on peut l'écrire du matrice, et on peut obtenir l'équation (6).

2.2 a)

```
w0 = 2*pi;
T0 = 3;
dt = 0.01;

q0 = [1];
q1 = [0];
q2 = [-w0^2*q0(1)];

count = 1;
t = [0];
while t(count) <= T0
    q0(count+1) = q0(count)+dt*q1(count);
    q1(count+1) = q1(count)+dt*q2(count);
    q2(count+1) = -w0^2*q0(count+1);
    t = [t, dt*count];
    count = count + 1;
end
```

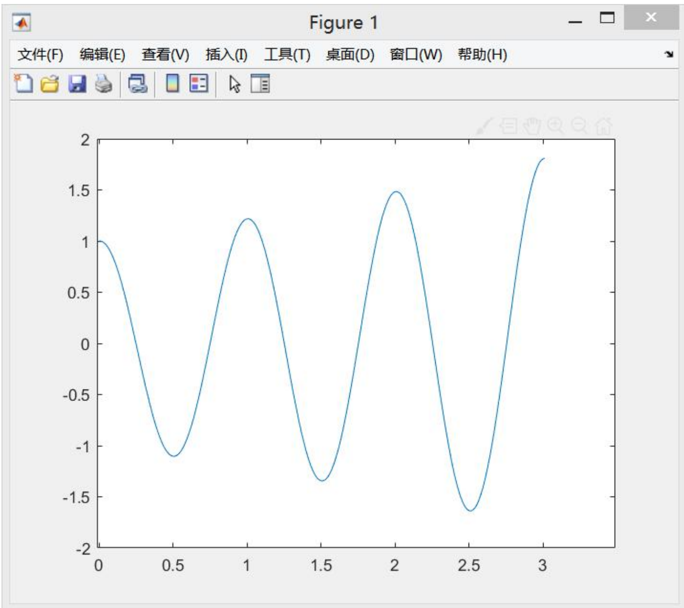
2.2 b)

```
w0 = 2*pi;
T0 = 3;
dt = 0.01;
```

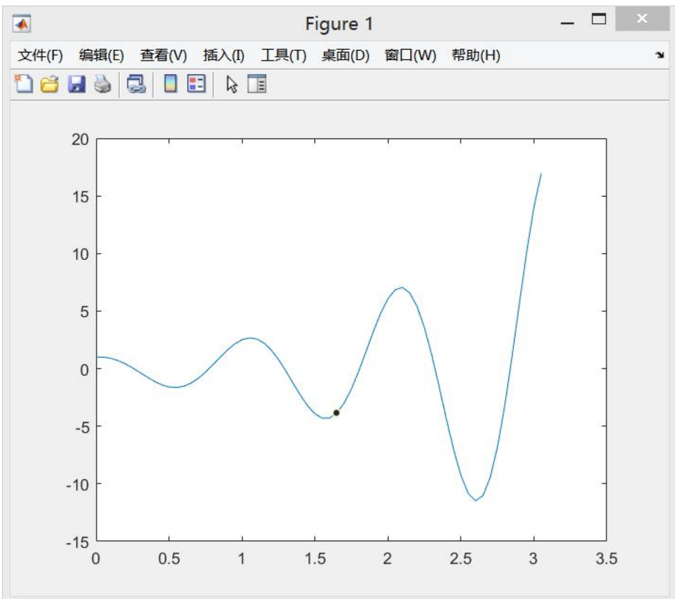
```
q0 = [1];
q1 = [0];
A = [1 dt;-w0^2*dt 1];
U = [q0;q1];

count = 1;
t = [0];
while t(count) < T0
   U(:,count+1) = A*U(:,count);
   t = [t,dt*count];
   count = count+1;
end
```

2.3 pour dt = 0.01, on peut obtenir la résultat:



Pour dt = 0.05, on peut obtenir la résultat:



On trouve que la divergence est plus lente si dt est plus petit.

2.4 On peut trouver que E* augmente avec le temps, mais il ne change pas en fact.

2.5 On utilise [U,V]=eig(A) pour obtenir des valeurs propres, et on peut trouver des valeurs propres sont des nombres complexes. Donc, la résultat est divergente.

3.1

Méthode1:

```
w0 = 2*pi;
T0 = 3;
dt = 0.01;

q = [1];
dq = [0];
ddq = [-w0^2*q(1)];

count = 1;
t = [0];
E = [1/2*(dq(1)^2+w0^2*q(1)^2)];
while t(count) <= T0
    q(count+1) = (q(count)+dt*dq(count))/(1+dt^2*w0^2);
    ddq(count+1)  = -w0^2*q(count+1);
    dq(count+1)  = dq(count)+dt*ddq(count+1);
    E = [E, 1/2*(dq(count+1)^2+w0^2*q(count+1)^2)];
    t = [t, dt*count];
    count = count + 1;

end
```

Méthode2:

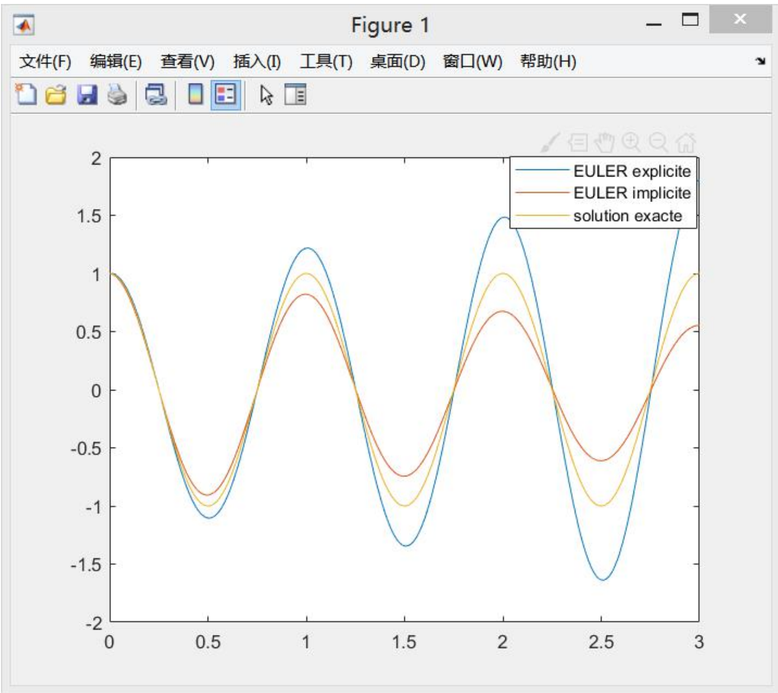
```
w0 = 2*pi;
T0 = 3;
dt = 0.01;

q = [1];
dq = [0];
A = 1/(1+w0^2*dt^2)*[1 dt;-w0^2*dt 1];
U = [q;dq];

count = 1;
t = [0];
while t(count) < T0
   U(:,count+1) = A*U(:,count);
   t = [t,dt*count];
   count = count+1;
```

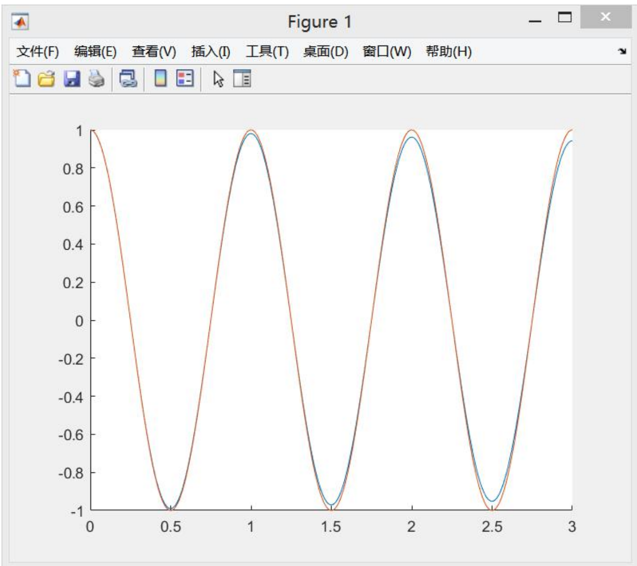
```
end
```

3.2



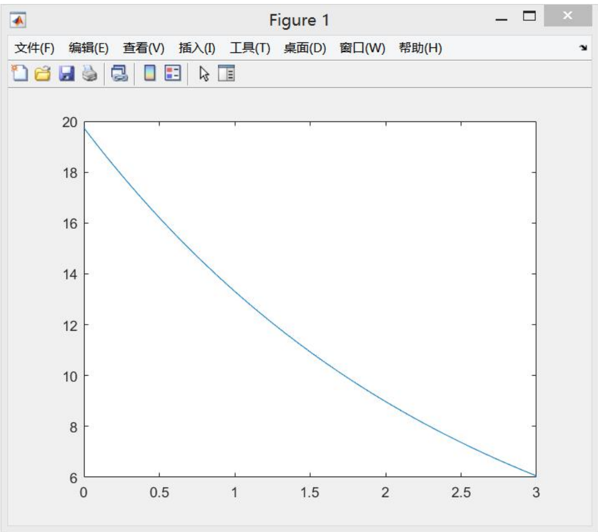
3.3

Quand dt=0.001, on trouve que l'érreur est plus petite:



3.4

E* varie comme la figure quand dt=0.01:



On trouve que il est plus en plus petit, lors la solution exacte est une constante. Cependant, comparé avec la solution d'EULER explicite, il change moins rapide. Quand dt est plus petit, il change moins rapide.

3.5

Les valeurs propres de la matrice A sont 1-i*w0*dt et 1+i*w0*dt. On trouve module de $(1-i*w0*dt)^p$ tanvers 0. Donc, on trouve la résultat.

4.1

Soit u= $\begin{pmatrix} q \\ \dot{q} \end{pmatrix}$, alors, on a du= $\begin{pmatrix} \dot{q} \\ \ddot{q} \end{pmatrix}$. Donc, on a une équation au permier ordre:

$$du = \begin{pmatrix} 0 & 1 \\ -\omega_0^2 & 0 \end{pmatrix} u$$

4.2

```
clear all
w0 = 2*pi;
T0 = 3;
dt = 0.01;

t = (0:dt:3)';
count = size(t,1);
q = zeros(count,1);
dq = zeros(count,1);

q0 = 1;
dq0 = 0;

q(1) = q0;
```

```
dq(1) = dq0;

u = [q0; dq0];
for i = 2:count
    tc = t(i -1);
    xc = u;
    k1 = cal_fc(xc,tc,w0);
    xc = u + k1/2*dt;
    k2 = cal_fc(xc,tc+dt/2,w0);
    xc = u + k2/2*dt;
    k3 = cal_fc(xc,tc+dt/2,w0);
    xc = u + k3*dt;
    k4 = cal_fc(xc,tc+dt,w0);
    k = (k1+2*k2+3*k3+k4)/6;
    u = u + k*dt;

    q(i) = u(1);
    dq(i) = u(2);
end
```

4.3

Quand on prend dt=0.01s, on trouve que la résultat est plus précise.

4.4

On trouve que E* augment avec le temps, mais plus lentement que on a vu dans un schéma d'EULER explicite.