

Oscillateur conservative linéaire à un degré de liberté

Question 1

1.1

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

```
>> r
```

```
r =
```

```
cos(2*pi*t)
```

1.2

```
q=r;
```

```
dq=diff(q);
```

```
E=(dq^2+(2*pi*q)^2)/2
```

En fin on obtient que

```
E =
```

```
2*pi^2*sin(2*pi*t)^2 + 2*pi^2*cos(2*pi*t)^2
```

Donc $E = 4\pi^2$ E est un constant

Question 2

2.1

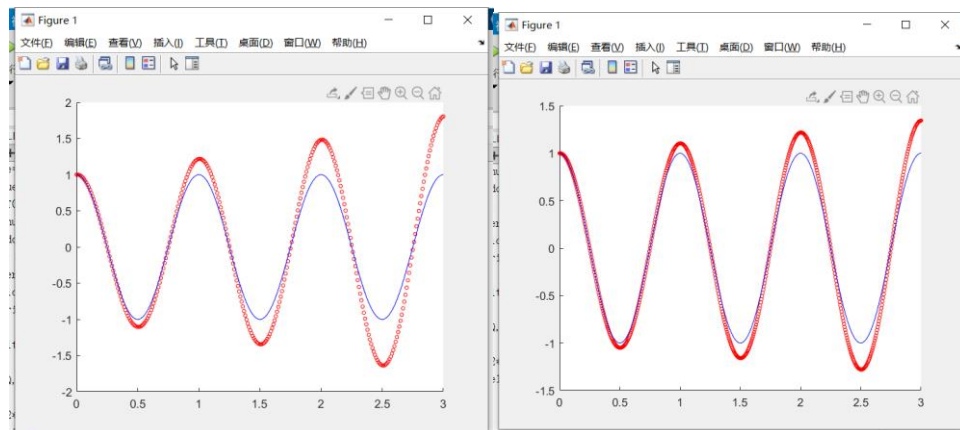
on a : $\ddot{q} + \omega^2 q = 0$
donc $\ddot{q} = -\omega^2 q$
 $\begin{vmatrix} q_{j+1} \\ \dot{q}_{j+1} \end{vmatrix} = \begin{vmatrix} q_j \\ \dot{q}_j + \Delta t X \end{vmatrix} - \omega^2 q$ $\Rightarrow \begin{vmatrix} q_{j+1} \\ \dot{q}_{j+1} \end{vmatrix} = \begin{bmatrix} 1 & \Delta t \\ -\omega^2 \Delta t & 1 \end{bmatrix} \begin{vmatrix} q_j \\ \dot{q}_j \end{vmatrix}$

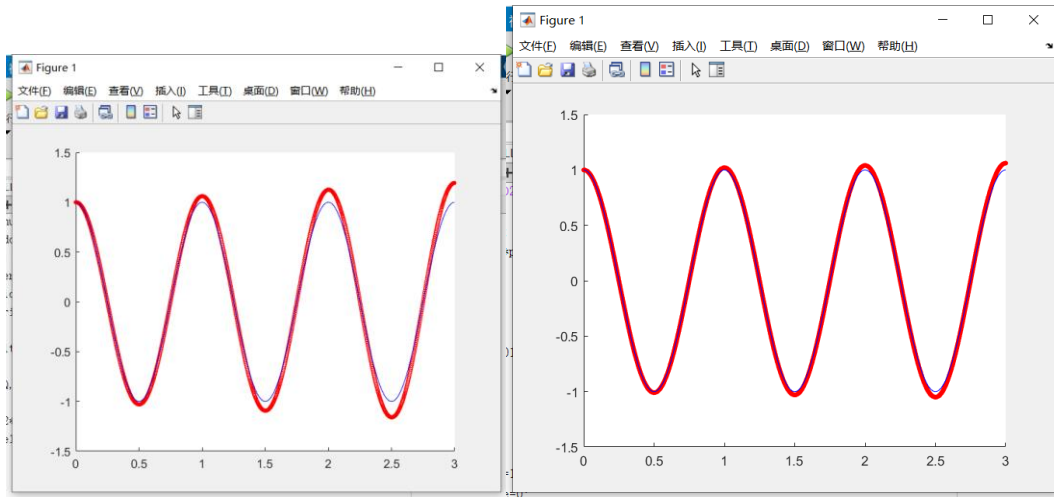
2.2

```
%Question2
T0=3;
deltat=0.001;
Q=[];
dQ=[];
T=[];
t=0;
n=1;
qnumerique=1;
dqnumerique=0;
d2qnumerique=0;
while( t<=T0 )
    Q(n)=qnumerique;
    dQ(n)=dqnumerique;
    T(n)=t;
    d2qnumerique=-(2*pi)^2*qnumerique;
    qnumerique=qnumerique+deltat*dqnumerique;
    dqnumerique=dqnumerique+deltat*d2qnumerique;
    n=n+1;
    t=t+deltat;
end
```

2.3

```
clear
scatter(T, Q, 10, 'r')
hold on
Qreel=cos(2*pi*T);
plot(T, Qreel, 'b')
```

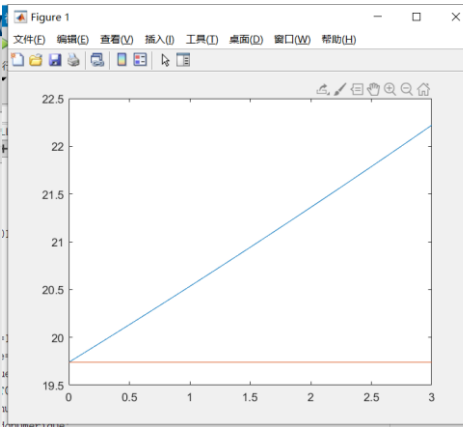
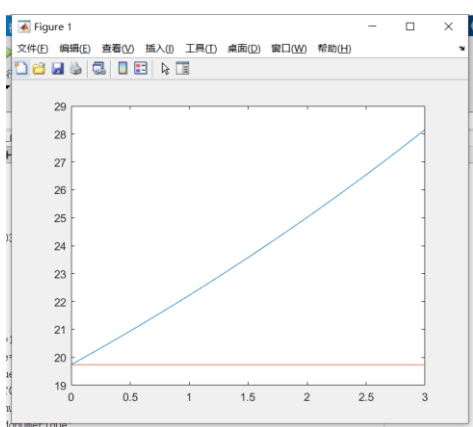
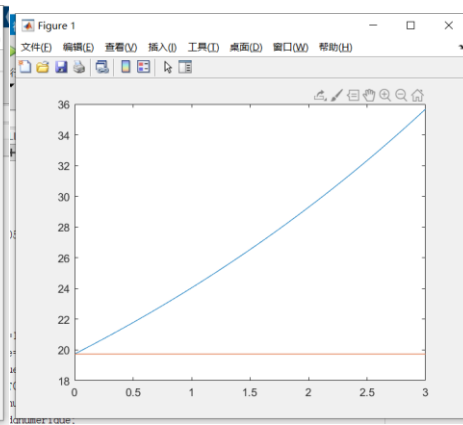
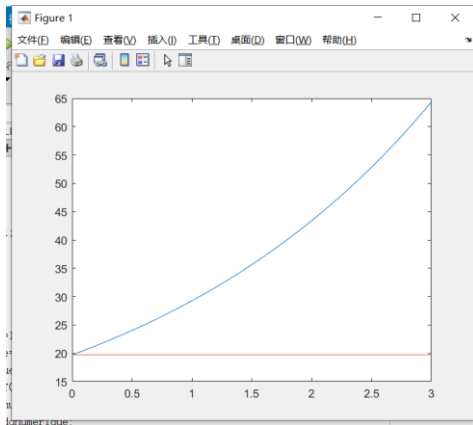




Quand le pas de temps est 0.01, 0.005, 0.003, 0.001, les figures sont comme en-dessus.
 On peut dire que ce schéma d'intégration est divergente, et le plus le pas de temps est petit,
 plus la divergence est lent.

2.4

```
% Enumerique=((dQ.^2)+(2*pi).^2*(Q.^2))/2;
% plot(T, Enumerique)
% hold on
% plot([0 3], [19.7392 19.7392])
```



On peut voir que avec l'évolution du temps, E^* augmente.

Quand le pas de temps est 0.01, 0.005, 0.003, 0.001, les figures sont comme en-dessus.

On peut dire que le plus le pas de temps est petit, la plus l'augmentation de E^* est lent, donc plus la divergence est lent.

2.5

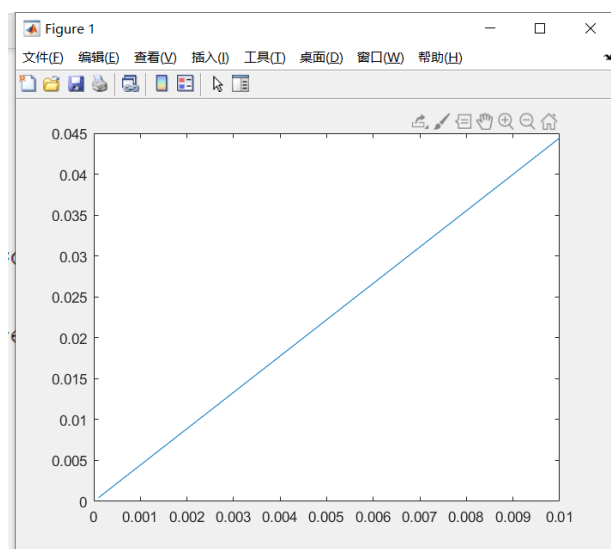
On peut voir que les valeurs propres de la matrice d'amplification est toujours sous la forme de $1 \pm x$ où x est fonction du pas de temps

Donc la valeur absolue des valeurs propres sont toujours supérieur à 1

Donc le schéma d'Euler explicite est toujours instable.

```
— delta=0.0001;
— n=1
— EIGMI=[];
— DELTAT=[];
— while (n<=100)
—     MI=[1, delta*n; -(2*pi^2)*delta*n, 1];
—     eigMi=eig(MI);
—     EIGMI(n)=abs(eigMi(1,1)-eigMi(2,1))/2;
—     DELTAT(n)=delta*n;
—     n=n+1;
— end
— plot(DELTAT, EIGMI)
```

En utilisant cette code, on peut déterminer numériquement la relation entre x et deltat



Donc , le plus le pas de temps est petit, plus la valeur absolue des valeurs propres est

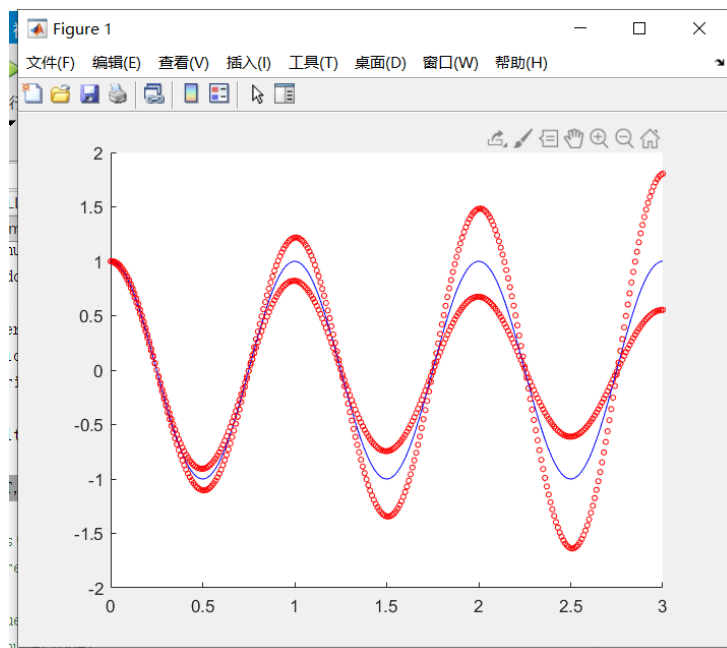
approche de 1 , donc plus la divergence est lent.

Question3

3.1

```
T0=3;
deltat=0.01;
Q2=[];
dQ2=[];
T2=[];
t=0;
n=1;
qnumerique2=1;
dqnumerique2=0;
d2qnumerique2=0;
while( t<=T0 )
    Q2(n)=qnumerique2;
    dQ2(n)=dqnumerique2;
    T(n)=t;
    qnumerique2=(qnumerique2+deltat*dqnumerique2)/(1+(deltat*2*pi)^2);
    d2qnumerique2=-(2*pi)^2*qnumerique2;
    dqnumerique2=dqnumerique2+deltat*d2qnumerique2;
    n=n+1;
    t=t+deltat;
end
scatter(T, Q2, 10, 'r')
```

3.2

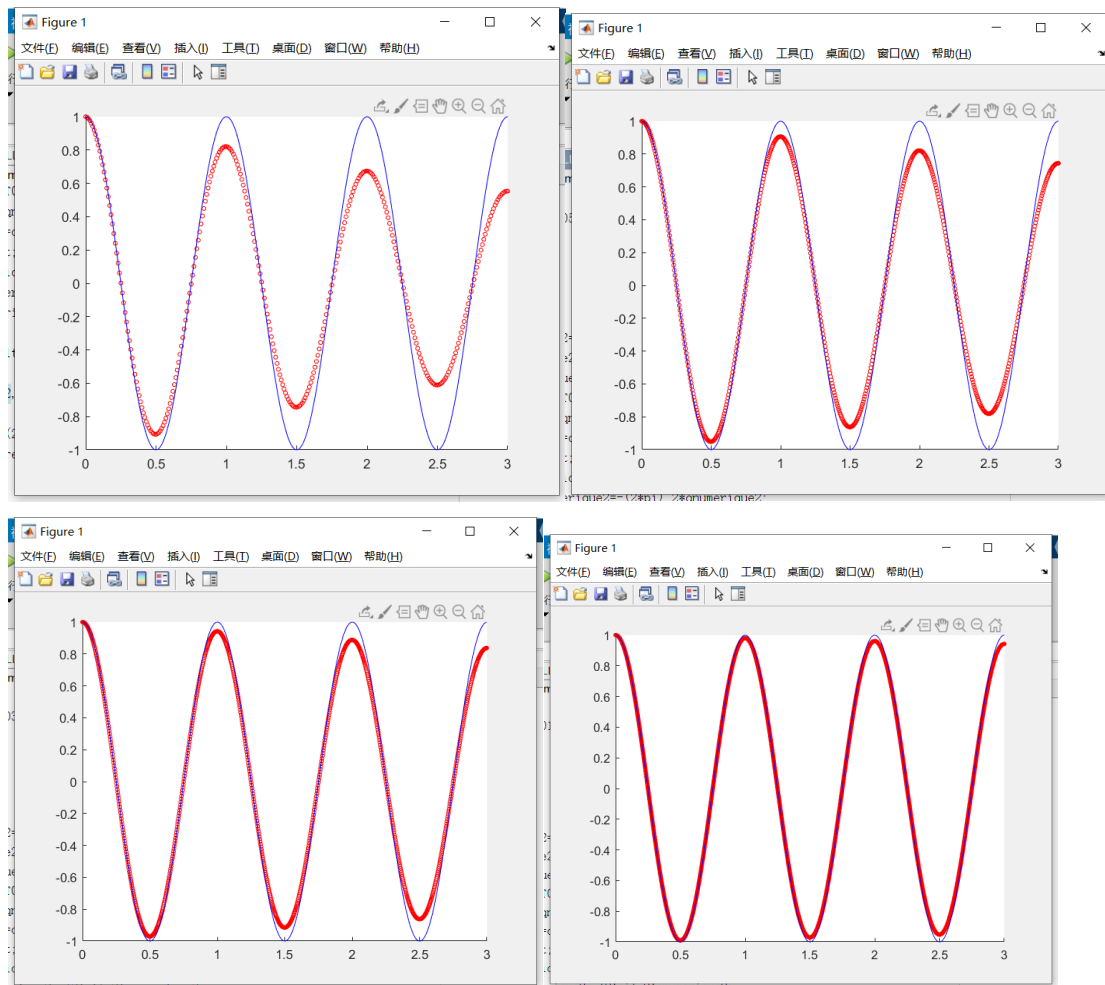


3.3

```

% scatter(T2, Q2, 10, 'r')
% hold on
% Qreel=cos(2*pi*T2);
% plot(T2, Qreel, 'b')
% clear all

```



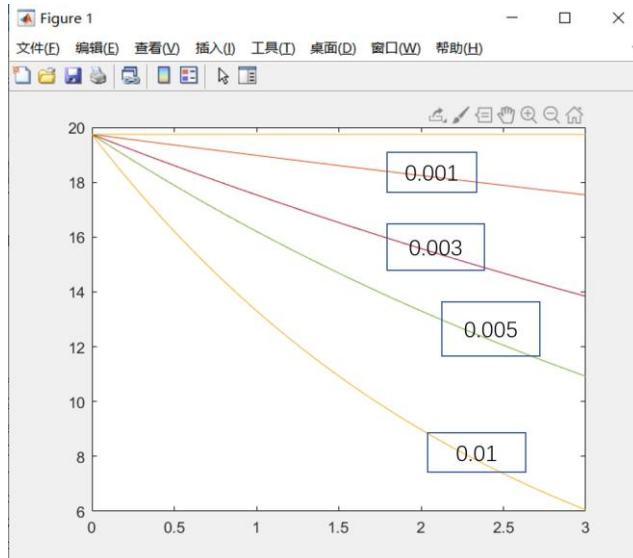
Quand le pas de temps est 0.01, 0.005, 0.003, 0.001, les figures sont comme en-dessus.
 On peut voir que le schéma implicite introduit un amortissement numérique.
 Et plus le pas de temps est petit, plus l'atténuation est faible.

3.4

```

% Enumerique=((dQ2.^2)+(2*pi)^2*(Q2.^2))/2;
% plot(T2, Enumerique)
% hold on
% plot([0 3], [19.7392 19.7392])
% hold on

```



On peut voir que avec l'évolution du temps, E^* diminue.

Quand le pas de temps est 0.01, 0.005, 0.003, 0.001, les figures sont comme en-dessus.

On peut dire que le plus le pas de temps est petit, la plus la diminution de E^* est lent, donc plus la divergence est lent.

3.5

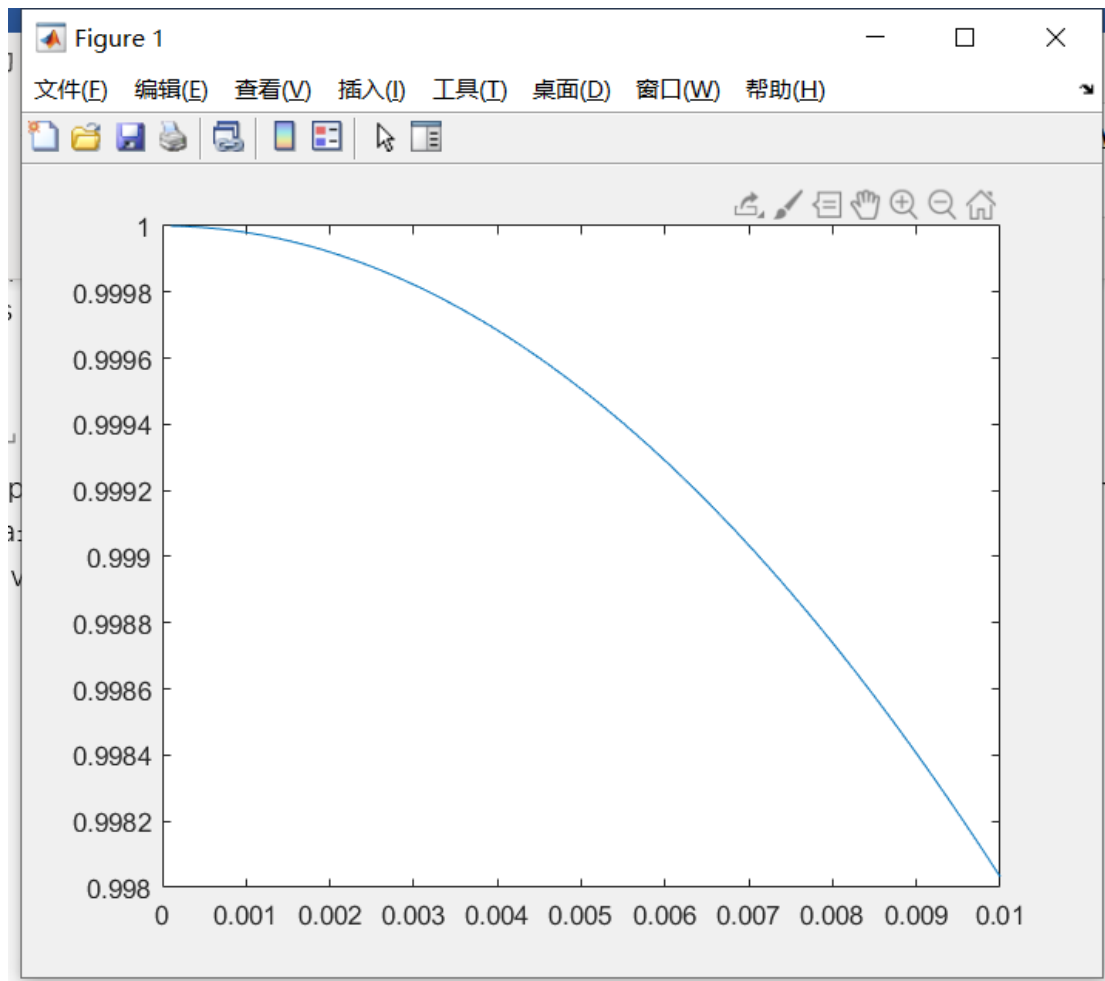
```

12
13 —   delta=0.0001;
14 —   n=1
15 —   EIGMI=[];
16 —   DELTAT=[];
17 —   while (n<=100)
18 —       MI=[1, delta*n; -(2*pi)^2*delta*n, 1]/(1+(2*pi*delta*n)^2);
19 —       eigMi=eig(MI);
20 —       EIGMI(n)=abs(eigMi(1,1));
21 —       DELTAT(n)=delta*n;
22 —       n=n+1;

```

On peut voir que les valeurs propres de la matrice d'amplification est toujours sous la forme de $a \pm bi$

Ses valeurs absolues sont en relation avec le pas de temps comme dans la figure



Donc, plus le pas de temps est petit, plus l'atténuation est faible.

Question 4

4.1

on veut résoudre $\ddot{q} + \omega^2 q = 0$
 supposons que $q_1 = \dot{q}$ alors $q_1' = \ddot{q} = -\omega^2 q$
 alors $\begin{pmatrix} \dot{q} \\ q_1 \end{pmatrix} = \begin{pmatrix} \dot{q} \\ \dot{q}_1 \end{pmatrix} = \begin{pmatrix} q_1 \\ -\omega^2 q \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\omega^2 & 0 \end{pmatrix} \begin{pmatrix} q \\ q_1 \end{pmatrix}$
 $I(\dot{q}, y(t)) = \begin{pmatrix} q \\ q_1 \end{pmatrix}$
 $f(y(t), t) = \begin{pmatrix} 0 & 1 \\ -\omega^2 & 0 \end{pmatrix} \cdot y(t)$
 $= f(y(t))$

4.2


```

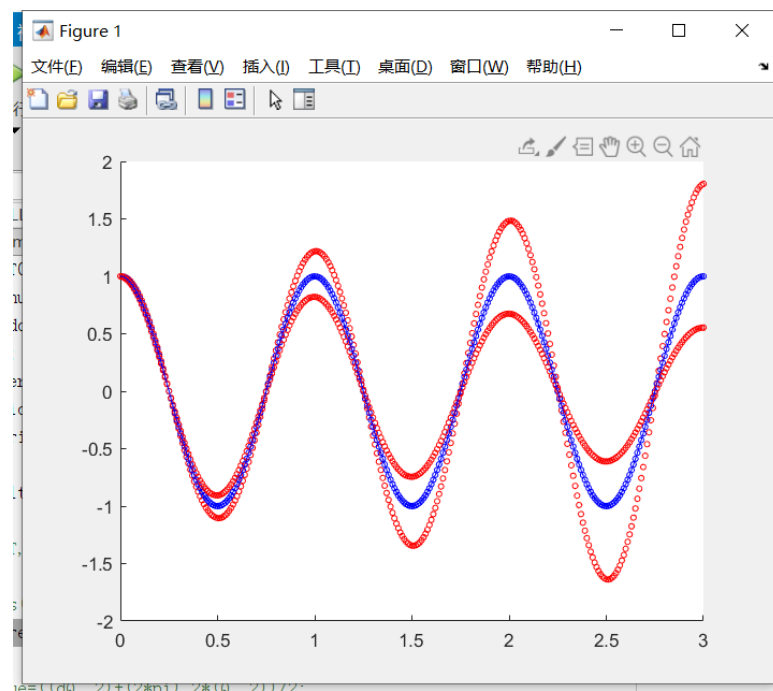
1 - T0=3;
2 - deltat=0.01;
3 - Q3=[];
4 - dQ3=[];
5 - T3=[];
6 - t=0;
7 - n=1;
8 - Qveteur=zeros(2,1);
9 - Qveteur(1,1)=1;
10 - Qveteur(2,1)=0;
11 - F=[0 1;-(2*pi)^2 0];
12 - while( t<=T0 )
13 -     Q3(n)=Qveteur(1,1);
14 -     dQ3(n)=Qveteur(2,1);
15 -     T3(n)=t;
16 -     k1=F*Qveteur;
17 -     k2=F*(Qveteur+k1*deltat/2);
18 -     k3=F*(Qveteur+k2*deltat/2);
19 -     k4=F*(Qveteur+k3*deltat);
20 -     K=(k1+2*k2+2*k3+k4)/6;
21 -     Qveteur=Qveteur+K*deltat;
22 -     n=n+1;
23 -     t=t+deltat;
24 - end
25

```

命令行窗口

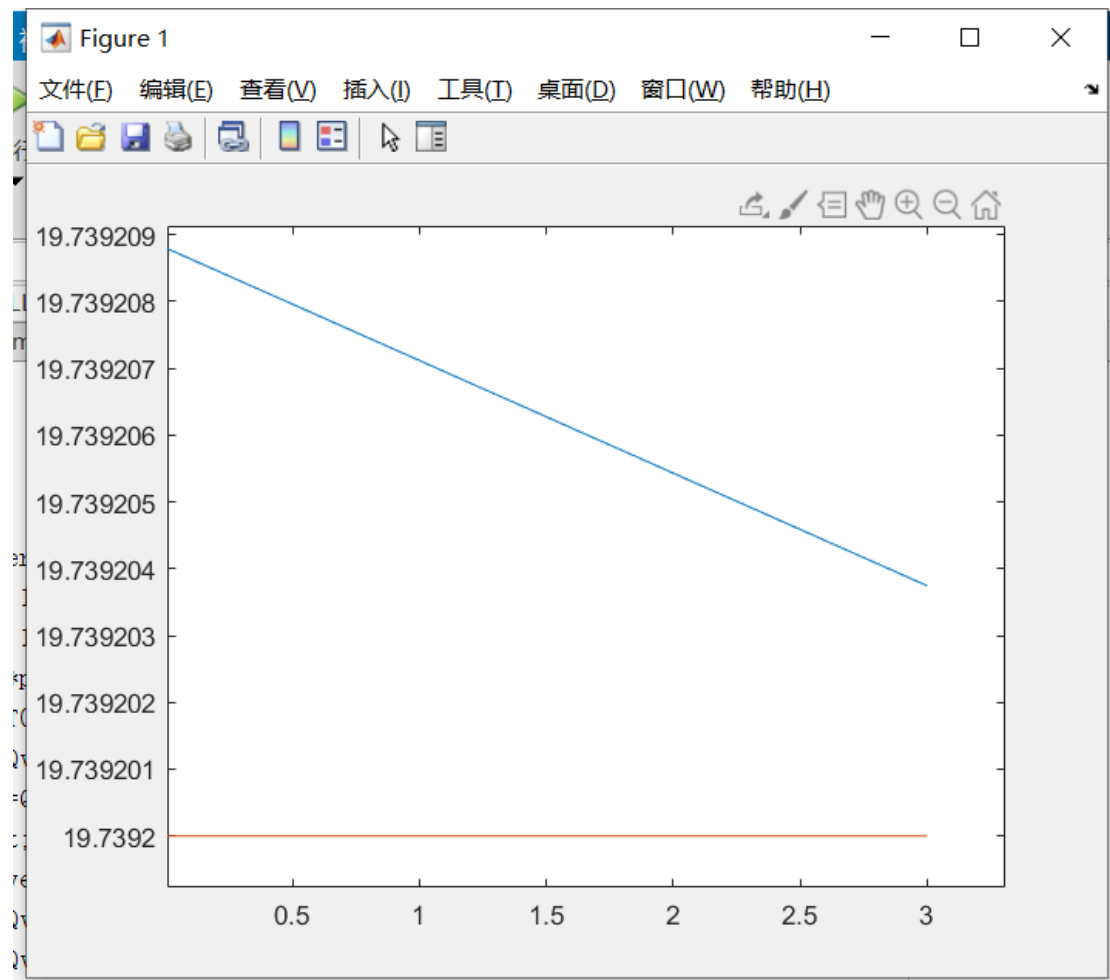
4.3

On peut dire que le schéma de RungeKutta a une performance meilleure,



4.4

```
24 - end
25 - Enumerique=((dQ3.^2)+(2*pi)^2*(Q3.^2))/2;
26 - plot(T3, Enumerique)
27 - hold on
28 - plot([0 3], [19.7392 19.7392])
```



On peut voir que ce schéma a aussi des erreurs, mais sa performance est beaucoup plus meilleure que Euler explicite et Euler implicite.

Question 5

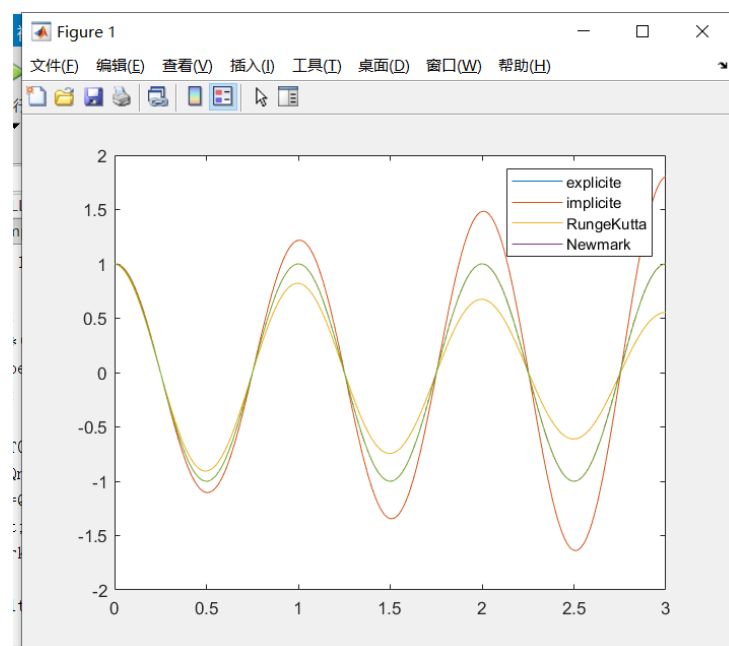
5.1

5.1.1

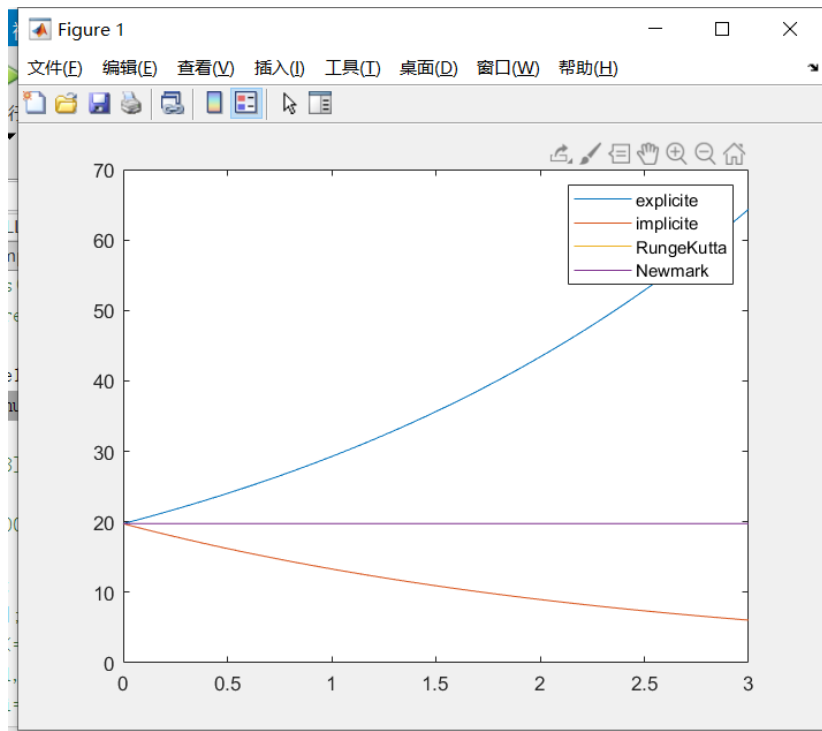
```
explicite.m x implicite.m x RungeKutta.m x Untitled.m x +
T0=3;
deltat=0.01;
Q4=[];
dQ4=[];
T4=[];
t=0;
n=1;
Qnewmark=zeros(2,1);
Qnewmark(1,1)=1;
Qnewmark(2,1)=0;
gamma=0.5;
beita=0.25;
B=[1+beita*(deltat*2*pi)^2 0 ; gamma*deltat*(2*pi)^2 1];
C=[1-(0.5-beita)*(deltat*2*pi)^2 deltat; -(1-gamma)*deltat*(2*pi)^2 1];
A=inv(B)*C;

while( t<=T0 )
    Q4(n)=Qnewmark(1,1);
    dQ4(n)=Qnewmark(2,1);
    T4(n)=t;
    Qnewmark=A*Qnewmark;
    n=n+1;
    t=t+deltat;
end
```

5.1.2



5.1.3



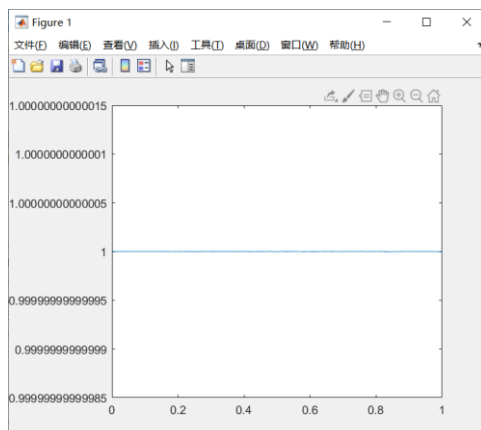
On peut voir que la permanence de Newmark et RungeKutta est presque la même, et leur performances sont meilleures que celles de Euler explicite et Euler implicite.

5.1.4

```

27 - n=1
28 - EIGMI=[];
29 - DELTAT=[];
30 - while (n<=1000)
31 -     B=[1+beita*(delta*n*2*pi)^2 0 ; gamma*delta*n*(2*pi)^2 1];
32 -     C=[1-(0.5-beita)*(delta*n*2*pi)^2 deltat; -(1-gamma)*delta*n*(2*pi)^2 1];
33 -     A=inv(B)*C;
34 -     eigMi=eig(A);
35 -     EIGMI(n)=abs(eigMi(1,1));
36 -     DELTAT(n)=delta*n;
37 -     n=n+1;
38 - end
39 - plot(DELTAT, EIGMI)

```



On peut voir que c'est toujours convergente.

5.2

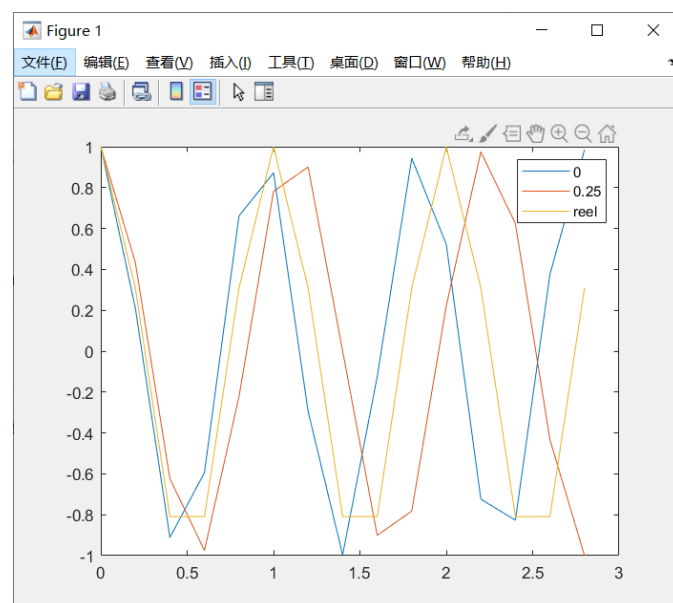
5.2.1

```
T0=3;
deltat=0.01;
Q4=[];
dQ4=[];
T4=[];
t=0;
n=1;
Qnewmark=zeros(2,1);
Qnewmark(1,1)=1;
Qnewmark(2,1)=0;
gamma=0.5;
beita=0;
B=[1+beita*(deltat*2*pi)^2 0 ; gamma*deltat*(2*pi)^2 1];
C=[1-(0.5-beita)*(deltat*2*pi)^2 deltat; -(1-gamma)*deltat*(2*pi)^2 1];
A=inv(B)*C;

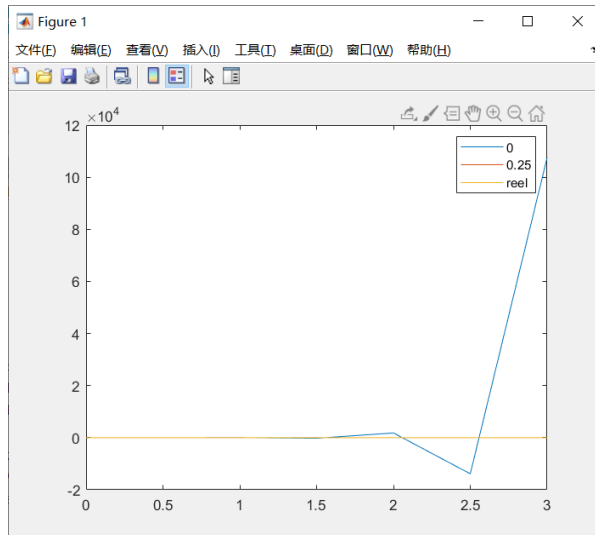
while( t<=T0 )
    Q4(n)=Qnewmark(1,1);
    dQ4(n)=Qnewmark(2,1);
    T4(n)=t;
    Qnewmark=A*Qnewmark;
    n=n+1;
    t=t+deltat;
end
```

5.2.3

Quand deltat=0.2s

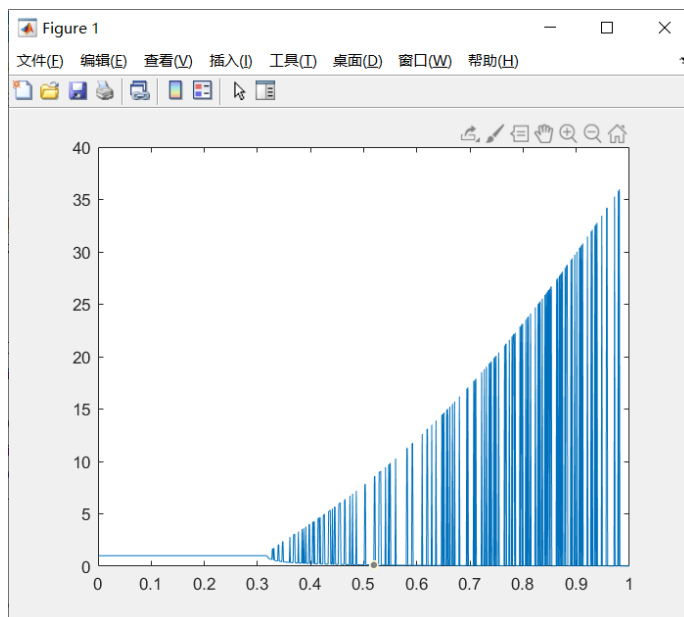


Quand deltat=0.5s



On peut conclure que,
 premièrement le coefficient beita peut faire ce schéma un peut 'retard', on doit ajouter ce coefficient pour améliorer la performance de ce méthode
 deuxièmement le coefficient beita peut retenir la divergence, on peut voir que quand beita=0, ce méthode est divergence quand deltat=0.5s.

5.2.4



On peut trouver que $t=0.318$

Donc $\alpha=0.5$

Étude d'un oscillateur linéaire amorti à un degré de liberté

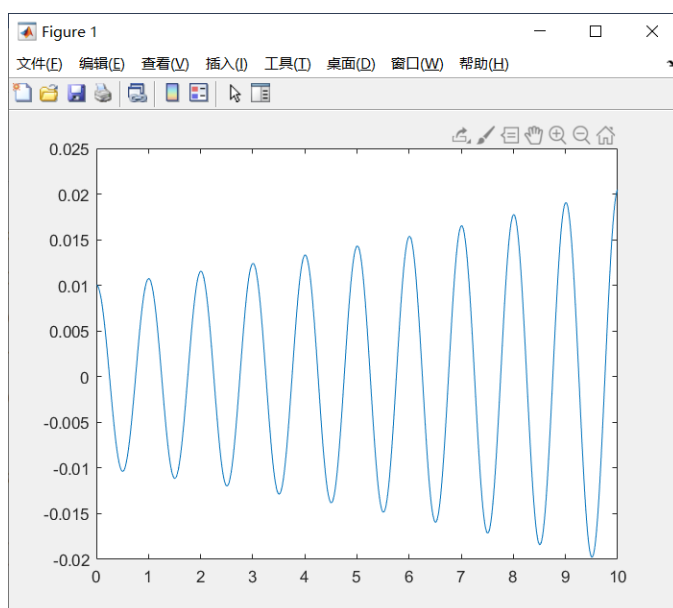
Question 1

1.1

```
T0=1;
deltat=0.001;
A=[1 deltat;-(2*pi)^2*deltat 1-2*0.02*2*pi*deltat];
Xexp=zeros(2,1);
Xexp(1,1)=0.01;
Xexp(2,1)=0;
X1=[];
X11=[];
Texp=[];
t=0;
n=1;
while( t<=T0*10 )
    X1(n)=Xexp(1,1);
    X11(n)=Xexp(2,1);
    Texp(n)=t;
    Xexp=A*Xexp;
    n=n+1;
    t=t+deltat;
end
plot(Texp,X1)
```

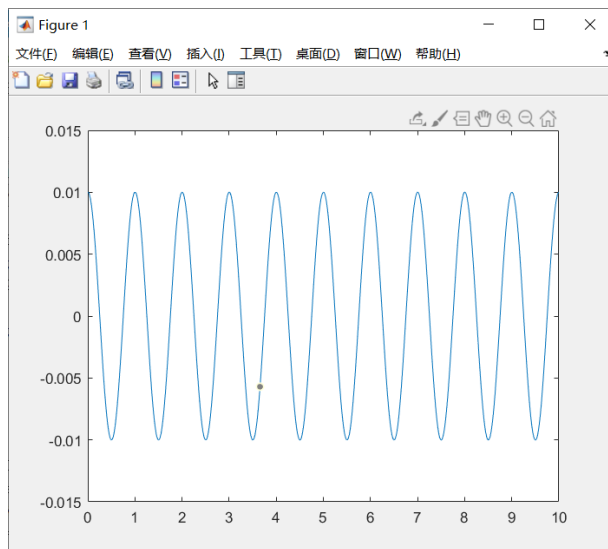
a :

quand on choisit deltat=0.01 la figure est :



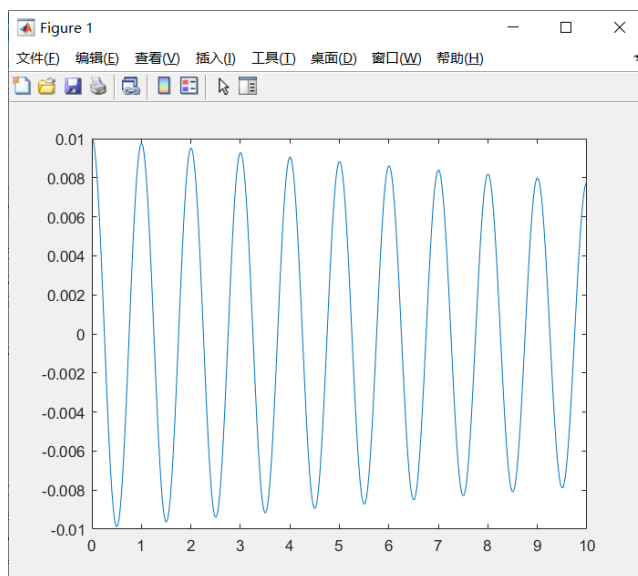
La simulation est divergente

b : quand on choisit $\Delta t = 0.0637$



La simulation n'a pas de bonne performance aussi

c : quand on choisit $\Delta t = 0.0051$



Cette simulation a une meilleure performance.

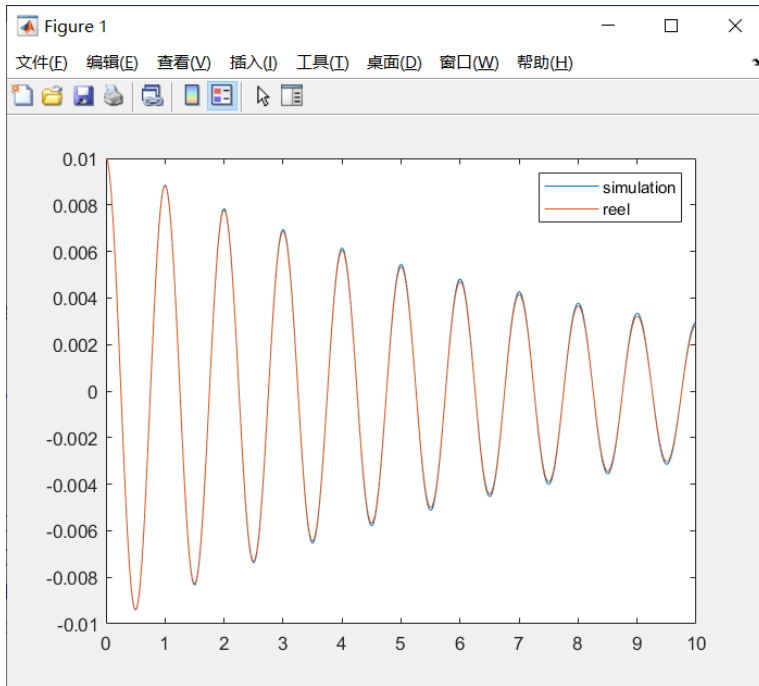
d :

```

-         t=t+deltat,
-     end
-     plot(Texp, X1)
-     hold on
-     om=2*pi*((1-0.02*0.02)^(1/2))
-     Xreel=(exp(-0.02*2*pi*Texp)).*(0.01*cos(om*Texp)+((0.02*2*pi*0.01)/om)*sin(om*Texp));
-     plot(Texp, Xreel)
-     legend('simulation', 'reel')
```

Quand $\Delta t = 0.002$, la simulation présente-t-elle une précision suffisante.

Le rapport est 0.3



1.2

On a

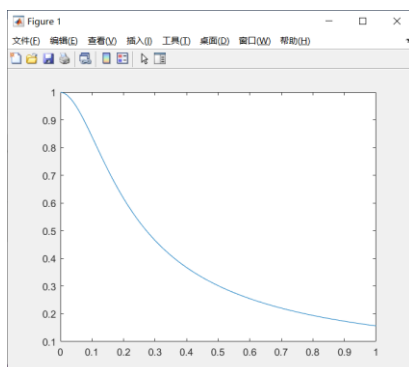
$$\begin{pmatrix} \omega_0^2 \Delta t & 1 + \sum \omega_0 \Delta t \\ 1 + \omega_0^2 \Delta t^2 & 2 \sum \omega_0 \Delta t^2 \end{pmatrix} \begin{pmatrix} x_{j+1} \\ \dot{x}_{j+1} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & \Delta t \end{pmatrix} \begin{pmatrix} x_j \\ \dot{x}_j \end{pmatrix} \quad \text{fig 1}$$

```

delta=0.001;
n=1;
EIG=[];
DELTAT=[];

while (n<=1000)
    deltat=n*delta;
    DELTAT(n)=deltat;
    Mg=[(2*pi)^2*deltat^2+2*0.02*2*pi*deltat;1+(2*pi*deltat)^2*2*0.02*2*pi*(deltat)^2];
    Md=[0 1;1 deltat];
    A2=inv(Mg)*Md;
    eigg=eig(A2);
    EIG(n)=abs(eigg(1,1));
    n=n+1;
end

```

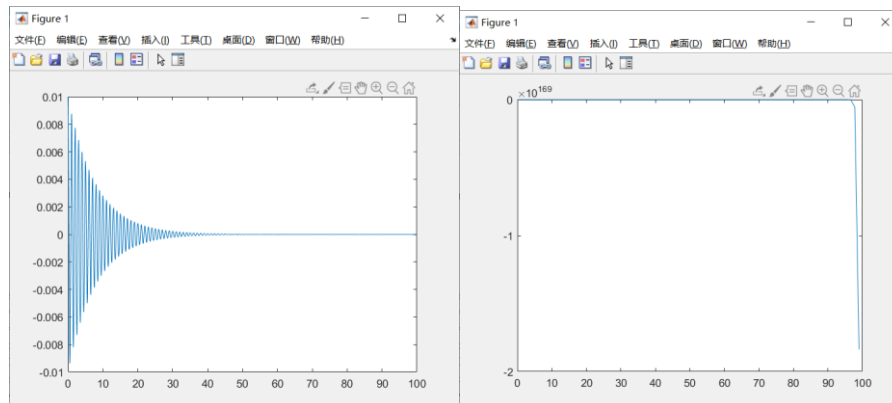


La valeur absolue des valeurs propres sont toujours inférieures à 1

1.3

```
T0=1;
h=0.04;
deltat=(h*2*(2)^2)/(2*pi);
X2=[];
X22=[];
Trunge=[];
t=0;
n=1;
Xrunge=zeros(2,1);
Xrunge(1,1)=0.01;
Xrunge(2,1)=0;
F=[0 1;-(2*pi)^2 -2*0.02*2*pi];
while( t<=T0*100 )
    X2(n)=Xrunge(1,1);
    X22(n)=Xrunge(2,1);
    Trunge(n)=t;
    k1=F*Xrunge;
    k2=F*(Xrunge+k1*deltat/2);
    k3=F*(Xrunge+k2*deltat/2);
    k4=F*(Xrunge+k3*deltat);
    K=(k1+2*k2+2*k3+k4)/6;
    Xrunge=Xrunge+K*deltat;
    n=n+1;
    t=t+deltat;
end
```

1.3a



On peut dire que avec l'augmentation de h, la précision diminue et la simulation devient instable.

1.3b

$0.358 < hc < 0.359$

Étude d'un double pendule

Question 1

On peut obtenir la relation entre q_n , \dot{q}_n , q_{n+1} et \dot{q}_{n+1} que :

$$\dot{q}_{n+1} + \Delta t \gamma A^{-1} B q_{n+1} = -\Delta t(1-\gamma)A^{-1}B \dot{q}_n + \dot{q}_n + \Delta t(1-\gamma)A^{-1}C_n + \Delta t \gamma A^{-1}C_{n+1}$$

$$\begin{pmatrix} \Delta t^2 \beta A^{-1} B & 0 \\ \Delta t \gamma A^{-1} B & I \end{pmatrix} \begin{pmatrix} q_{n+1} \\ \dot{q}_{n+1} \end{pmatrix} = \begin{pmatrix} 1 - \Delta t^2(0.5 - \beta)A^{-1}B & \Delta t I \\ -\Delta t(1-\gamma)A^{-1}B & I \end{pmatrix} \begin{pmatrix} q_n \\ \dot{q}_n \end{pmatrix} + \begin{pmatrix} \Delta t^2(0.5 - \beta)A^{-1}C_n + \Delta t^2 \beta A^{-1}C_{n+1} \\ \Delta t(1-\gamma)A^{-1}C_n + \Delta t \gamma A^{-1}C_{n+1} \end{pmatrix}$$

où

$$A = m a^2 \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$$

$$B = m g a \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$$

$$C_n = F_0 \sin(\omega t_n) \begin{pmatrix} a \\ \frac{a}{\sqrt{2}} \end{pmatrix}$$

$$q_n = \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} \quad \dot{q}_n = \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix}$$

1.1

```
Q21.m x Q22.m x Q23.m x Untitled.m* x +
m=2; a=0.5; g=9.81; F0=20; omega=2*pi;
q=zeros(2,1);
dq=[-1.31519275;-1.85996342];
deltat=0.01;
beta=0; gamma=0.5;
Mzero=zeros(2,2);
MI=[1 0;0 1];
A=m*a^2*[2 1;1 1];
B=m*g*a*[2 0;0 1];
Mg=[1+deltat^2*beta+inv(A)*B Mzero;deltat*gamma*inv(A)*B MI];
Md=[1-deltat^2*(0.5-beta)*inv(A)*B deltat*MI;-deltat*(1-gamma)*inv(A)*B MI];
Ma=inv(Mg)*Md;
```

Ma est la matrice d'amplification.

1.2

On obtient que le temps critique est $\text{deltat}=0.5\text{s}$

1.3 En utilisant l'equation 1,2,3, on a

Handwritten mathematical equations on a piece of paper:

$$\ddot{q}_0 = A^{-1} [C_0 - B q_0]$$
$$\ddot{q}_0 = A^{-1} (-B) \cdot q_0$$
$$\dot{q}_0 = \begin{pmatrix} -1.31519275 \\ -1.85996342 \end{pmatrix}$$

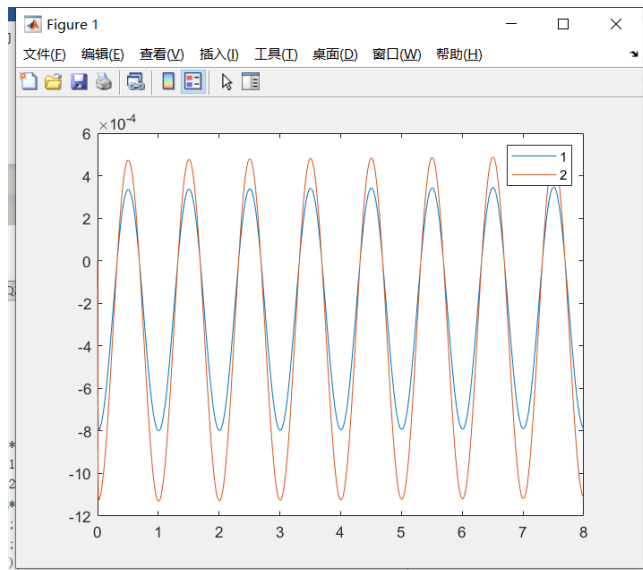
1.4

Handwritten matrix equation for a discrete-time system:

$$\begin{pmatrix} \Delta t^2 \beta A^{-1} B & 0 \\ \Delta t \gamma A^{-1} B & I \end{pmatrix} \begin{pmatrix} q_{n+1} \\ \dot{q}_{n+1} \end{pmatrix} = \begin{pmatrix} 1 - \Delta t^2 (0.5 - \beta) A^{-1} B & \Delta t \cdot I \\ -\Delta t (1 - \gamma) A^{-1} B & I \end{pmatrix} \begin{pmatrix} q_n \\ \dot{q}_n \end{pmatrix} + \begin{pmatrix} \Delta t^2 (0.5 - \beta) A^{-1} C_n + \Delta t^2 \beta A^{-1} C_{n+1} \\ \Delta t (1 - \gamma) A^{-1} C_n + \Delta t \gamma A^{-1} C_{n+1} \end{pmatrix}$$

1.5

```
Q21.m x Q22.m x Q23.m x Untitled.m x +
15 - T=[];
16 - t=0;T0=8;
17 - n=1;
18 - while( t<=T0 )
19 -     Q1(n)=q(1,1);
20 -     Q2(n)=q(2,1);
21 -     I(n)=t;
22 -     Md21=deltat^2*(0.5-beta)*inv(A)*F0*sin(omega*t)*[a;a/(2^(1/2))]+deltat^2*beta*inv(A)*F0*sin(omega*(t+deltat))*[a;a/(2^(1/2))];
23 -     Md22=deltat*(1-gamma)*inv(A)*F0*sin(omega*t)*[a;a/(2^(1/2))]+deltat*gamma*inv(A)*F0*sin(omega*(t+deltat))*[a;a/(2^(1/2))];
24 -     Md2=[Md21;Md22];
25 -     X=inv(Mg)*(Md*[q;dq]+Md2);
26 -     q(1,1)=X(1,1);
27 -     q(2,1)=X(2,1);
28 -     dq(1,1)=X(3,1);
29 -     dq(2,1)=X(4,1);
30 -     n=n+1;
31 -     t=t+deltat;
32 - end
```



Question 2

2.1

```

Q21.m x Q22.m x Q23.m x Untitled.m* x +
-   m=2; a=0.5; g=9.81; F0=20; omega=2*pi;
-   q=zeros(2,1);
-   dq=[-1.31519275;-1.85996342];
-   deltat=0.02;
-   beita=0.25; gamma=0.5;
-   Mzero=zeros(2,2);
-   MI=[1 0;0 1];
-   A=m*a^2*[2 1;1 1];
-   B=m*g*a*[2 0;0 1];
-   Mg=[1+deltat^2*beita+inv(A)*B Mzero;deltat*gamma*inv(A)*B MI];
-   Md=[1-deltat^2*(0.5-beita)*inv(A)*B deltat*MI;-deltat*(1-gamma)*inv(A)*B MI];
-   Ma=inv(Mg)*Md;
-   --

```

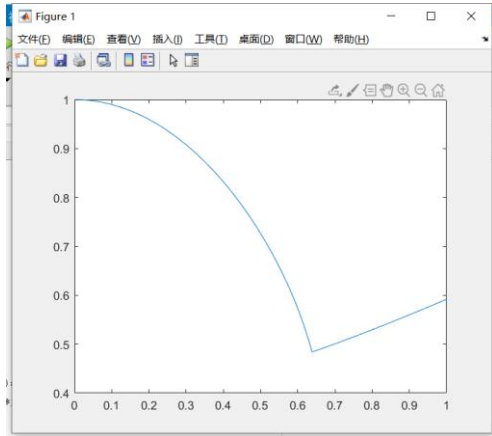
2.2

```

m=2; a=0.5; g=9.81; F0=20; omega=2*pi;
q=zeros(2,1);
dq=[-1.31519275;-1.85996342];
beita=0.25; gamma=0.5;
Mzero=zeros(2,2);
MI=[1 0;0 1];
A=m*a^2*[2 1;1 1];
B=m*g*a*[2 0;0 1];
delta=0.001;
n=1;
DELTAT=[];
EIG=[];

while (n<=1000)
    deltat=n*delta;
    DELTAT(n)=deltat;
    Mg=[1+deltat^2*beita+inv(A)*B Mzero;deltat*gamma*inv(A)*B MI];
    Md=[1-deltat^2*(0.5-beita)*inv(A)*B deltat*MI;-deltat*(1-gamma)*inv(A)*B MI];
    Ma=inv(Mg)*Md;
    EIG(n)=max(abs(eig(Ma)));
    n=n+1;
end

```



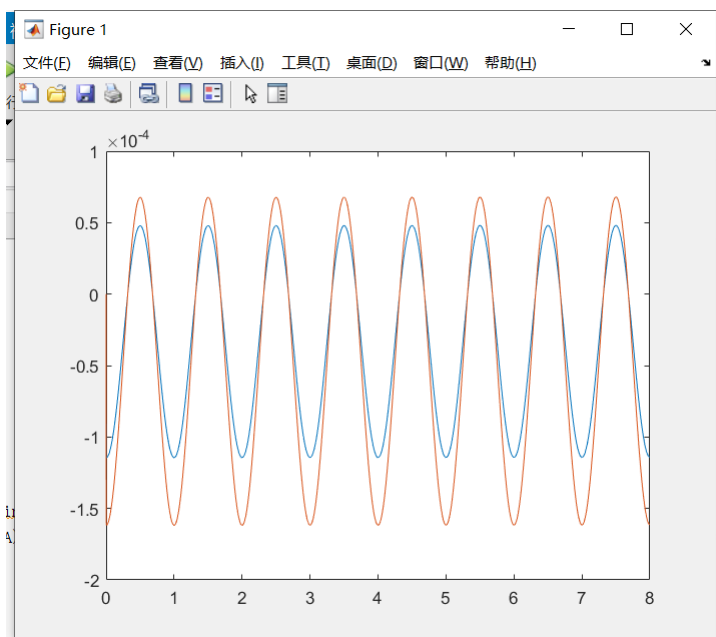
2.3.2.4 :comme 1.3 1.4

2.5

```

B=m*g*a*[2 0; 0 1];
Mg=[1+deltat^2*beita+inv(A)*B Mzero;deltat*gamma*inv(A)*B MI];
Md=[1-deltat^2*(0.5-beita)*inv(A)*B deltat*MI;-deltat*(1-gamma)*inv(A)*B MI];
Ma=inv(Mg)*Md;
Q1=[];
Q2=[];
T=[];
t=0;T0=8;
n=1;
while( t<=T0 )
    Q1(n)=q(1,1);
    Q2(n)=q(2,1);
    T(n)=t;
    Md21=deltat^2*(0.5-beita)*inv(A)*F0*sin(omega*t)*[a;a/(2^(1/2))]+deltat^2*beita*inv(A)*F0*sin(omega*(t+deltat))*[a;a/(2^(1/2))];
    Md22=deltat*(1-gamma)*inv(A)*F0*sin(omega*t)*[a;a/(2^(1/2))]+deltat*gamma*inv(A)*F0*sin(omega*(t+deltat))*[a;a/(2^(1/2))];
    Md2=[Md21;Md22];
    X=inv(Mg)*(Md*[q;dq]+Md2);
    q(1,1)=X(1,1);
    q(2,1)=X(2,1);
    dq(1,1)=X(3,1);
    dq(2,1)=X(4,1);
    n=n+1;
    t=t+deltat;
end
plot(T,Q1)

```



Oscillateur non linéaire à un degré de liberté

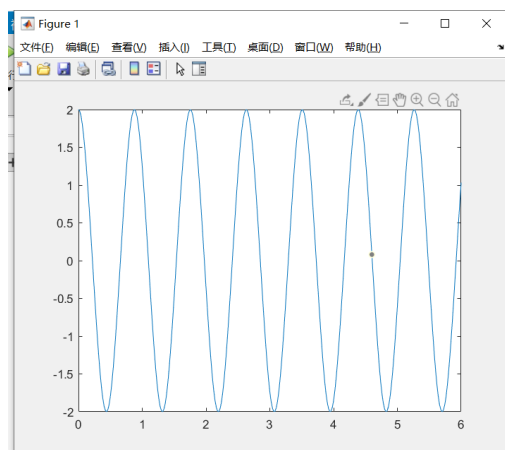
Question 1

1.1

$$\begin{cases} q_{j+1} = q_j + \Delta t \dot{q}_j + \frac{1}{2} \Delta t^2 \ddot{q}_j \\ \ddot{q}_{j+1} = -\omega^2 q_{j+1} (1 + \alpha q_{j+1}^2) \\ \dot{q}_{j+1} = \dot{q}_j + \frac{1}{2} \Delta t \ddot{q}_j + \frac{1}{2} \Delta t \ddot{q}_{j+1} \end{cases}$$

1.2

```
编辑器 - C:\Users\DELL\Desktop\Q41.m
Q32.m x Untitled.m x Q41.m x +
6 - d2q=-omega^2*q*(1+a*(q^2));
7 - q=[];
8 - dq=[];
9 - d2Q=[];
10 - T=[];
11 - n=1;
12 - t=0;
13 - deltat=0.02;
14 - while(t<=T0)
15 -     Q(n)=q;
16 -     dq(n)=dq;
17 -     T(n)=t;
18 -     d2Q(n)=d2q;
19 -     q=q+deltat*dq+deltat^2*d2q/2;
20 -     d2q=-omega^2*q*(1+a*(q^2));
21 -     dq=(dq(n)+deltat*d2Q(n)/2+deltat*d2q/2);
22 -     n=n+1;
23 -     t=t+deltat;
24 - end
25 - plot(T,Q)
26
```



1.3 Quand $t=0s$ $q(t)=2m$

Quand $t=0.02s$ $q(t)=1.9779m$

Quand $t=0.04s$ $q(t)=1.9123m$

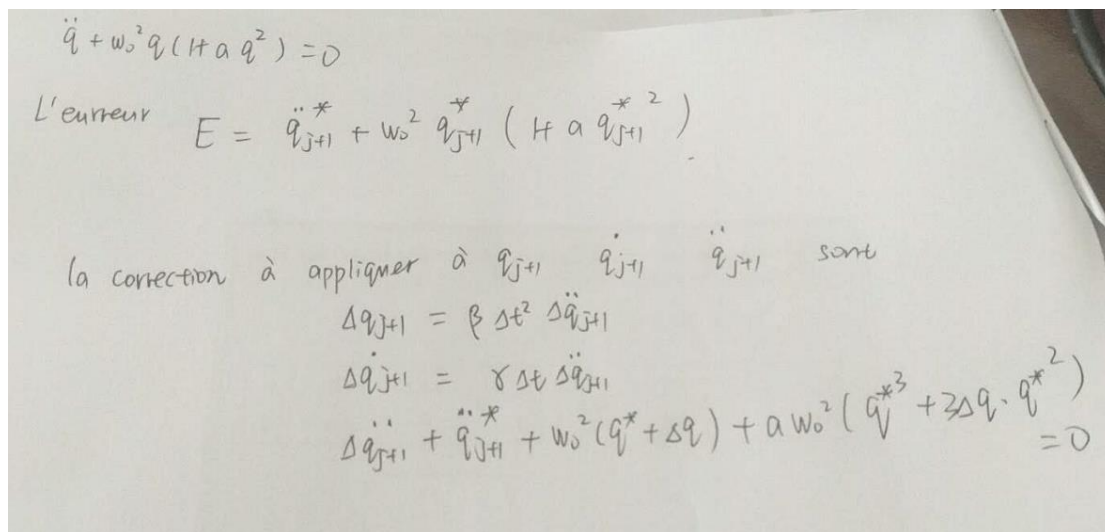
Quand $t=6s$ $q(t)=1.0329m$

Question 2

2.1

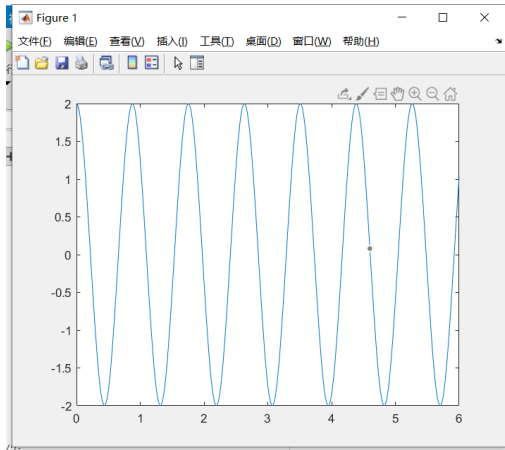
On doit minimiser l'erreur entre la simulation et la solution exacte

2.2



2.3

```
Q32.m x Untitled.m x Q41.m x Q42.m x +
1 - T0=6;
2 - omega=2*pi;
3 - a=0.1;
4 - x=2;
5 - dx=0;
6 - d2x=-omega^2*x*(1+a*(x^2));
7 - X=[];
8 - dX=[];
9 - d2X=[];
10 - T=[];
11 - n=1;
12 - t=0;
13 - deltat=0.02;
14 - while(t<=T0)
15 -     T(n)=t;
16 -     X(n)=x;
17 -     dX(n)=dx;
18 -     d2X(n)=d2x;
19 -     x=x+deltat*dx+deltat^2*(0.25)*d2x;
20 -     dx=dx+deltat*(0.5)*d2x;
21 -     d2x=0;
22 -     while(abs(d2x+omega^2*x*(a*x^2+1))>=0.01)
23 -         syms e de d2e;
24 -         [ee, dee, d2ee]=solve(ee==0.25*deltat^2*d2e, de==0.5*deltat*d2e, d2e+d2x+omega^2*(x+e)+a*omega^2*(x^3+3*e*x^2)==0, e, de, d2e);
25 -         XX=[x; dx; d2x]+[ee; dee; d2ee];
26 -         x=XX(1,1);
27 -         dx=XX(2,1);
28 -         d2x=XX(3,1);
29 -     end
30 -     t=t+deltat;
31 -     n=n+1;
32 - end
33 -
34 -
35 -
```

2.4

Quand $t=0s$ $q(t)=2m$

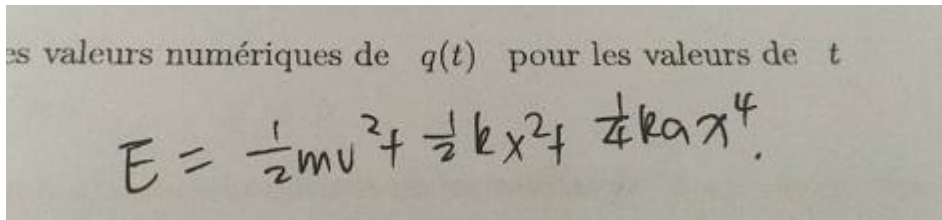
Quand $t=0.02s$ $q(t)=1.978m$

Quand $t=0.04s$ $q(t)=1.912m$

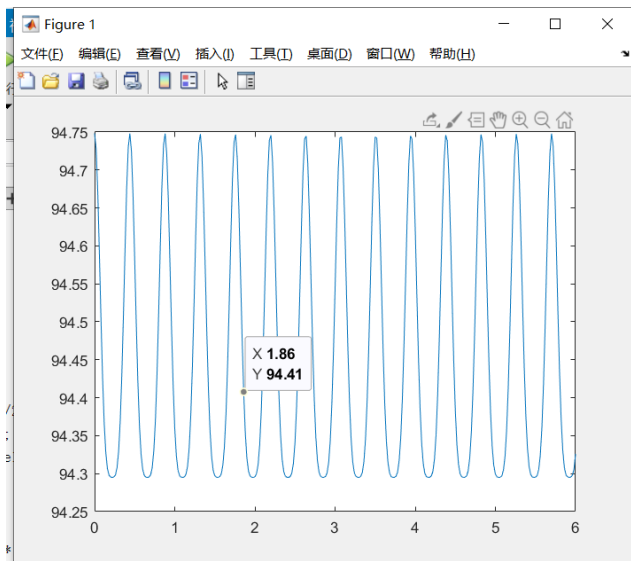
Quand $t=6s$ $q(t)=1.033m$

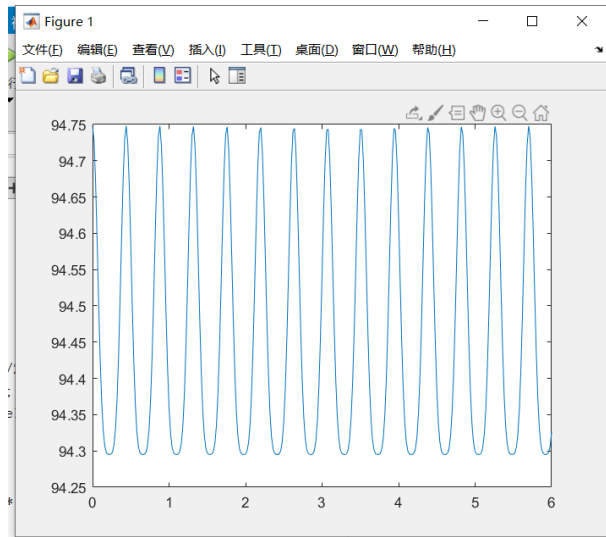
Question 3

3.1



3.2





3.3 On peut trouver que les performances sont presque la même