

Rendu sur le serveur pédagogique :

Ecrire un rapport avec vos lignes de code en Matlab ou Scilab sur l'exercice Oscillateur conservatif à un degré de liberté. Répondre à toutes les questions le schémas Euler explicite, implicite et Runge Kutta.

Oscillateur conservatif linéaire à un degré de liberté

1. Solution analytique de l'équation

1.1 La solution est sous forme de $x(t) = x_0 \cos(\omega_0 t + \varphi)$, $\omega_0 = \sqrt{\frac{k}{m}}$. En utilisant les conditions initiales données, le code Matlab est comme ci-dessous :

```
clear all ; close all ; clc
T0=3; w0=2*pi; w0c=w0*w0;
q0=1.; dq0=0.0; dte=0.01;
te=(0:dte:T0)';
npe=size(te,1);
qe=zeros(npe,1);
dqe=zeros(npe,1);
qe=q0*cos(w0*te)+dq0/w0*sin(w0*te) ;
dqe=-w0*q0*sin(w0*te)+dq0*cos(w0*te) ;
ddqe=-w0c*qe ;
```

1.2 $E^*=19.7392$ ne change pas.

2. Résolution de l'équation avec un schémas d'EULER explicite

2.1 D'après (5), on obtient

$$\begin{cases} q_{i+1} = q_i + \Delta t \times \dot{q}_i \\ \dot{q}_{i+1} = \dot{q}_i + \Delta t \times \ddot{q}_i \\ \ddot{q}_{i+1} = -\omega_0^2 q_{i+1} \end{cases}$$

Donc,

$$\begin{bmatrix} q_{i+1} \\ \dot{q}_{i+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ -\omega_0^2 \Delta t & 1 \end{bmatrix} \begin{bmatrix} q_i \\ \dot{q}_i \end{bmatrix}$$

2.2 a) La programmation directe

```
clear all ; close all ; clc
T0=3; w0=2*pi; w0c=w0*w0;
q0=1.;dq0=0;
dt1=0.01;
t1=(0:dt1:T0)';
np1=size(t1,1);
q1=zeros(np1,1);
dq1=zeros(np1,1);
ddq1=zeros(np1,1);
q1(1)=q0;
dq1(1)=dq0;
```

```

ddq1(1)=-w0c*q1(1);
for inc=2:np1
q1(inc)=q1(inc-1)+dt1*dq1(inc-1);
dq1(inc)=dq1(inc-1)+dt1*ddq1(inc-1);
ddq1(inc)=-w0c*q1(inc);
end
plot(t1,q1,'b-','Linewidth',3);
xlabel('t') ;
ylabel('q') ;
title(sprintf("pas de temps=%.2f",dt1));
legend('Schemas Euler-Explicite') ;

```

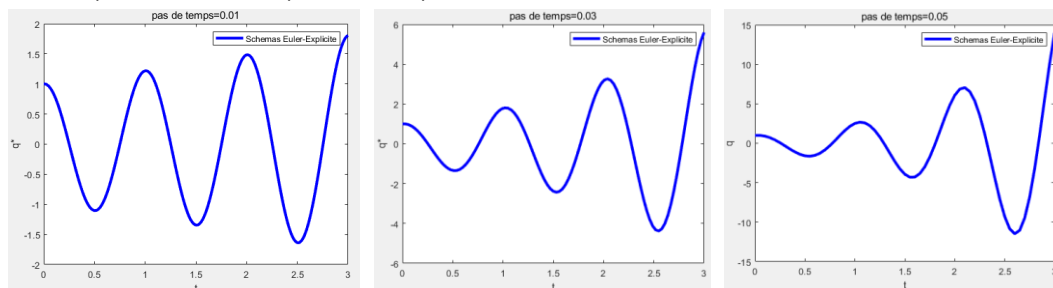
b) Avec Matrice d'amplification

```

clear all ; close all ; clc
dt1=0.05; T0=3;
q0=1 ;dq0=0 ;
w0=2*pi; w0c=w0*w0;
q=[q0 ;dq0] ;
t1=(0:dt1:T0)';
np1=size(t1,1);
q1b=zeros(np1,1);
dq1b=zeros(np1,1);
q1b(1)=q0;
dq1b(1)=dq0;
A=[1,dt1;-w0c*dt1,1];
for inc=2:np1;
q=A*q;
q1b(inc)=q(1);
dq1b(inc)= q(2);
end;
plot(t1,q1b,'g-','Linewidth',3);
xlabel('t') ;
ylabel('q') ;
legend('Euler-Explicite-Avec Matrice') ;

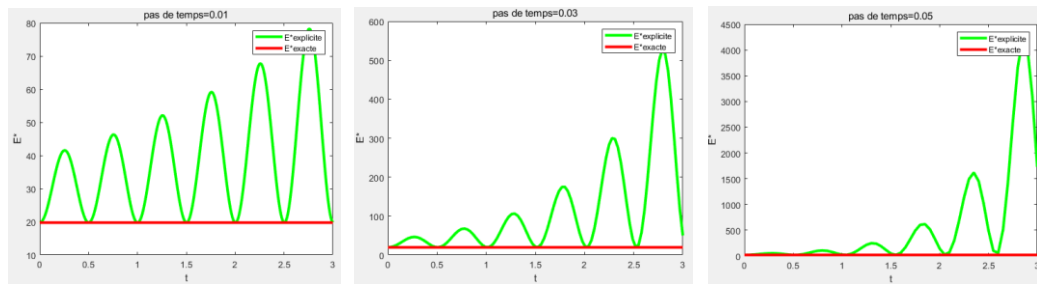
```

2.3 Comparer différents pas de temps



La solution obtenue est divergente, plus le pas de temps est petit, plus la divergence est lente.

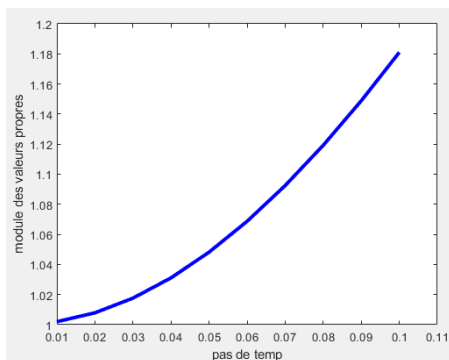
2.4 Comparer E^* en différents pas de temps avec E^* obtenue a partir de la solution exacte.



L'énergie obtenue est divergente, plus le pas de temps est petit, plus la divergence est lente.

2.5 études de la stabilité en utilisant code ci-dessous :

```
clear all ; close all ; clc
dt=(0.01:0.01:0.1);
w0=2*pi;
nmp=size(dt,2);
mo=zeros(1,nmp);
interm=zeros(2,2);
for inc=1:nmp;
A=[1,dt(inc);-1*w0*w0*dt(inc),1];
[z,d]=eig(A);
interm=abs(d);
mo(inc)=interm(1,1);
end;
plot(dt,mo,'b-','Linewidth',3);
xlabel('pas de temp');
ylabel('module des valeurs propres');
```



Comme les modules des valeurs propres sont toujours supérieur a 1, la solution numérique avec un schémas Euler explicite est inconditionnellement instable. Plus le pas de temps est grand, plus l'instabilité est grande.

3. Résolution de l' équation avec un Schéma d'Euler implicite

3.1 Sans matrice d'amplification

```
clear all ; close all ; clc
dt2=0.01; T0=3;
q0=1;dq0=0;
w0=2*pi; w0c=w0*w0;
t2=(0:dt2:T0)';
```

```

np2=size(t2,1);
q2=zeros(np2,1);
dq2=zeros(np2,1);
energ2=zeros(np2,1);
q2(1)=q0;
dq2(1)=dq0;
for inc=2:np2
    q2(inc)=(q2(inc-1)+dt2*dq2(inc-1))/(1+w0c*dt2*dt2);
    ddqc=-w0c*q2(inc);
    dq2(inc)=dq2(inc-1)+dt2*ddqc;
end
energ2=0.5*(dq2.*dq2+w0c*(dq2.^2));
plot(t2,q2,'b-', 'Linewidth',3);

```

3.2 Comparer les valeurs de q fournies par les trois solutions :

```
clear all ; close all ; clc
```

```
T0=3; w0=2*pi; w0c=w0*w0;
q0=1.; dq0=0.0;
```

```
% Solution exacte
```

```
dte=0.01;
te=(0:dte:T0)';
npe=size(te,1);
qe=zeros(npe,1);
dqe=zeros(npe,1);
energe=zeros(npe,1);
qe=q0*cos(w0*te)+dq0/w0*sin(w0*te) ;
dqe=-w0*q0*sin(w0*te)+dq0*cos(w0*te) ;
ddqe=-w0c*qe ;
energe=0.5*(dqe.*dqe+w0c*(qe.*qe));
```

```
% Euler explicite
```

```
dt1=0.01;
t1=(0:dt1:T0)';
np1=size(t1,1);
q1=zeros(np1,1);
dq1=zeros(np1,1);
ddq1=zeros(np1,1);
energ1=zeros(np1,1);
q1(1)=q0;
dq1(1)=dq0;
ddq1(1)=-w0c*q1(1);
for inc=2:np1
    q1(inc)=q1(inc-1)+dt1*dq1(inc-1);

```

```

dq1(inc)=dq1(inc-1)+dt1*ddq1(inc-1);
ddq1(inc)=-w0c*q1(inc);
end
energ1=0.5*(dq1.*dq1+w0c*(q0.^2));

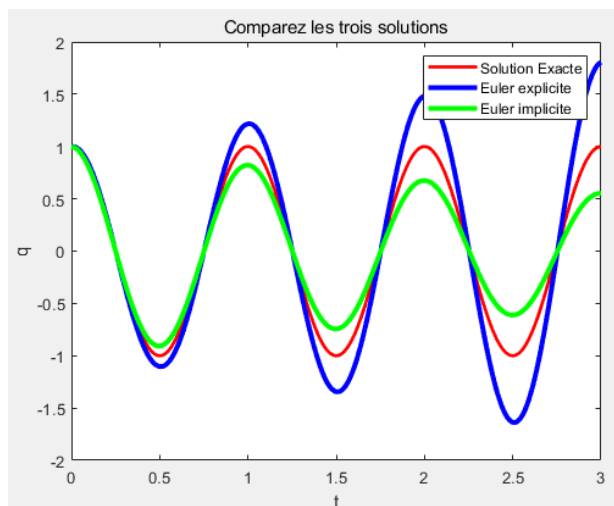
% Euler implicite
dt2=0.01;
t2=(0:dt2:T0)';
np2=size(t2,1);
q2=zeros(np2,1);
dq2=zeros(np2,1);
energ2=zeros(np2,1);
q2(1)=q0;
dq2(1)=dq0;
for inc=2:np2
    q2(inc)=(q2(inc-1)+dt2*dq2(inc-1))/(1+w0c*dt2*dt2);
    ddqc=-w0c*q2(inc);
    dq2(inc)=dq2(inc-1)+dt2*ddqc;
end
energ2=0.5*(dq2.*dq2+w0c*(dq2.^2));

%plot

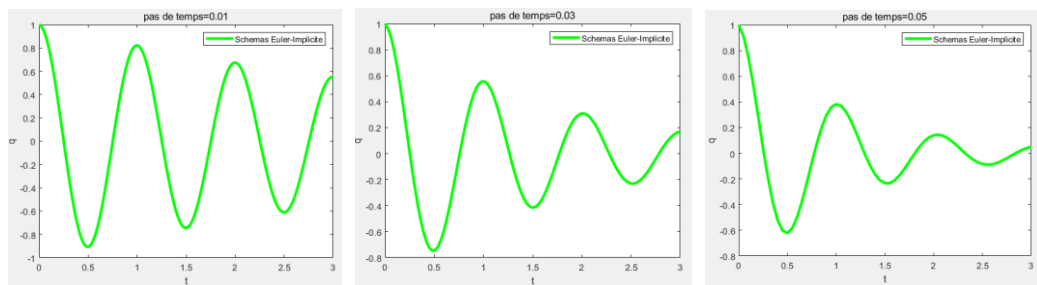
plot(te,qe, '-r', 'linewidth',2);
hold on
plot(t1,q1,'b-', 'Linewidth',3);
plot(t2,q2,'g-', 'Linewidth',3);

xlabel('t') ;
ylabel('q') ;
title('Comparez les trois solutions')
legend('Solution Exacte','Euler explicite','Euler implicite') ;

```

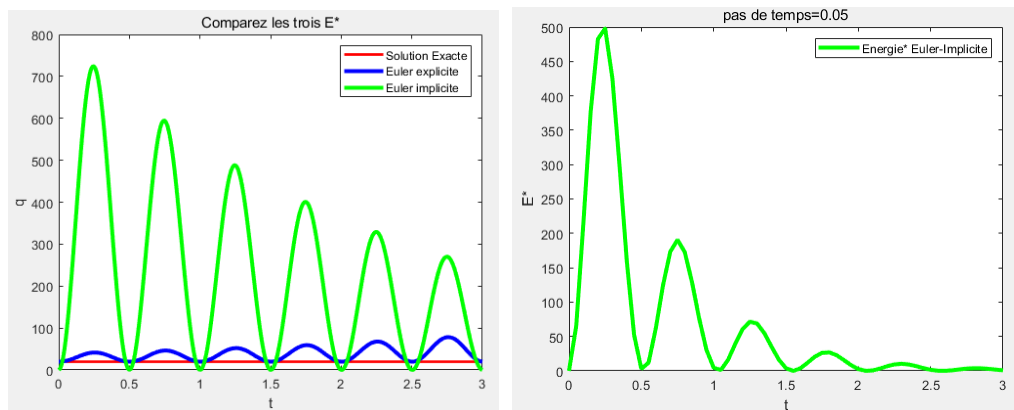


3.3 Comparer différents pas de temps



Il apparait des amortissements numériques. Plus le pas de temps est petit, plus l'atténuation des oscillations est faible.

3.4 Comparer les E^* en pas de temps 0.01

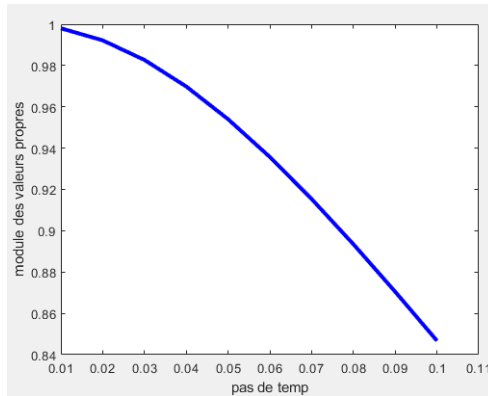


L'énergie obtenue avec schémas d'Euler Implicite est convergente.

On varie le pas de temps, et trouve que plus le pas de temps est petit, plus la convergence est lente.

3.5 Calculer numériquement les valeurs propres de la matrice d'amplification en fonction du pas de temps.

```
clear all ; close all ; clc
dt2=(0.01:0.01:0.1);
w0=2*pi;
nmp=size(dt2,2);
mo=zeros(1,nmp);
interm=zeros(2,2);
for inc=1:nmp;
A=[1,dt2(inc);-1*w0*w0*dt2(inc),1];
A=A/(1+w0*w0*dt2(inc)*dt2(inc));
[z,d]=eig(A);
interm=abs(d);
mo(inc)=interm(1,1);
end;
plot(dt2,mo,'b-','Linewidth',3);
xlabel('pas de temp');
ylabel('module des valeurs propres');
```



Comme les modules des valeurs propres sont toujours inférieure à 1, la solution numérique avec un schémas Euler implicite est inconditionnellement stable. Plus le pas de temps est grand, plus la solution numérique est stable.

4. Résolution de l'équation avec un schéma de RUNGE KUTTA

4.1 Transformation

$$\begin{bmatrix} \dot{x}(t) \\ \dot{x}(t) \end{bmatrix} = \begin{bmatrix} 0 & \Delta t \\ -\omega_0^2 & 1 \end{bmatrix} \begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix}$$

4.2 Code Matlab

Définir fonction et l'enregistrer sur le nom cal_f.m

```
function [ dUc ] = cal_f(Uc,tc,w0c);
dUc=zeros(2,1);
dUc(1)=Uc(2);
dUc(2)=-w0c*Uc(1);
end
```

Main Function:

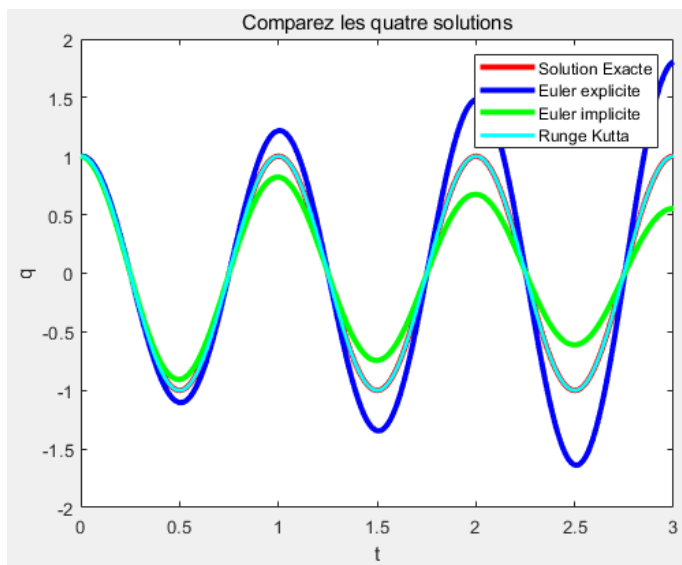
```
clear all ; close all ; clc
T0=3;w0=2*pi;w0c=w0*w0;
q0=1;dq0=0;
dt4=0.01;
t4=(0:dt4:T0)';
np4=size(t4,1);
q4=zeros(np4,1);
dq4=zeros(np4,1);
q4(1)=q0;
dq4(1)=dq0;
qj=[q0;dq0];
for i=2:np4
    tc=t4(i-1)
    xc=qj
    k1=cal_f(xc,tc,w0c)
    xc=qj+k1*dt4/2
    k2=cal_f(xc,tc+dt4/2,w0c)
    xc=qj+k2*dt4/2
    k3=cal_f(xc,tc+dt4/2,w0c)
```

```

xc=qj+k3*dt4
k4=cal_f(xc,tc+dt4,w0c)
dq=(k1+2*k2+2*k3+k4)/6
qj=qj+dq*dt4
q4(i)=qj(1)
dq4(i)=qj(2)
end
plot(t4,q4,'p-', 'Linewidth',3);
xlabel('t') ;
ylabel('q') ;
title(sprintf("pas de temps=%.2f",dt4));
legend('Schema Runge Kutta');

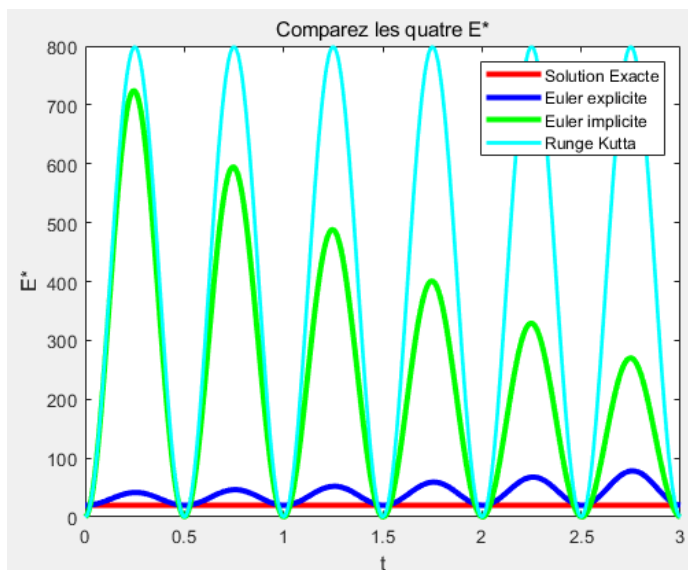
```

4.3 Comparer les 4 solutions



Le schéma Runge Kutta est parfaitement coincident avec la solution exacte.

4.4 Comparer E*



5. Résolution de l'équation avec un schéma de NEWMARK

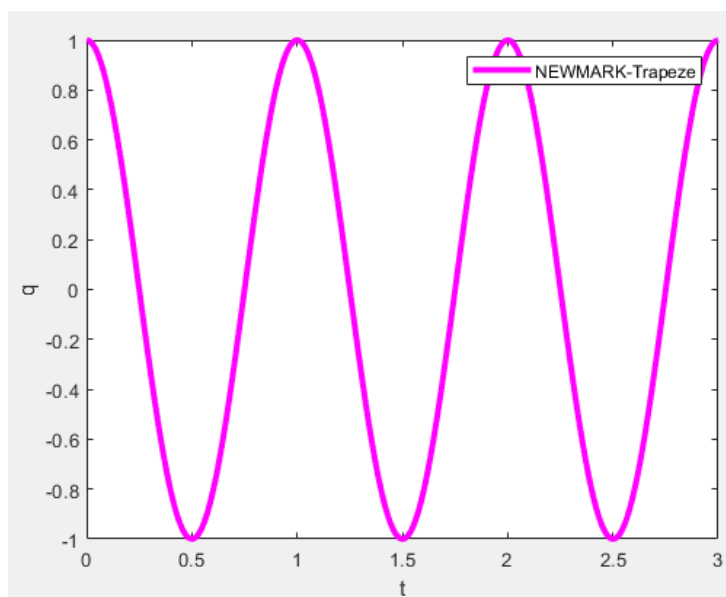
5.1 La méthode des trapèzes

5.1.1 Programmer

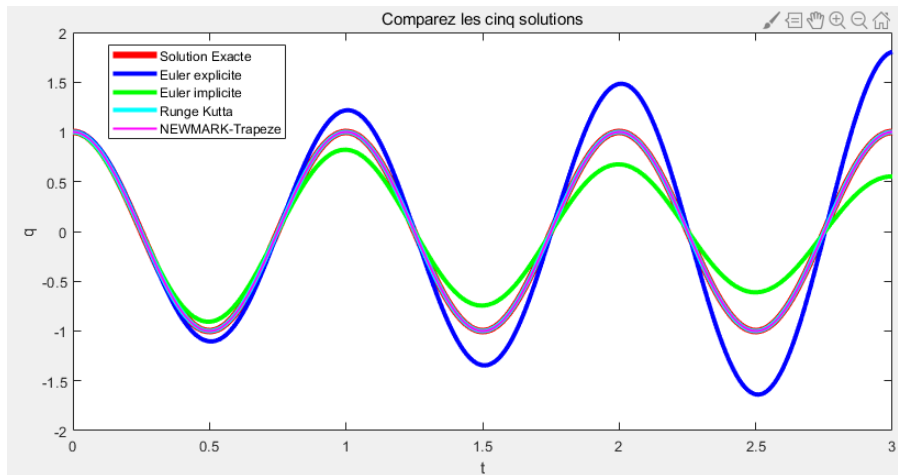
```
clear all ; close all ; clc
T0=3;w0=2*pi;w0c=w0*w0;
q0=1;dq0=0;
gamma = 0.5; beta = 0.25;
dt5=0.01;
B = [1+beta*dt5^2*w0^2 0;gamma*dt5*w0^2 1];
C = [1-dt5^2*(0.5-beta)*w0^2 dt5;-(1-gamma)*dt5*w0^2 1];
A = B\C;
q=[q0 ;dq0] ;
t5=(0:dt5:T0)';
np5=size(t5,1);
q5=zeros(np5,1);
dq5=zeros(np5,1);
q5(1)=q0;
dq5(1)=dq0;

for inc=2:np5;
q=A*q;
q5(inc)=q(1);
dq5(inc)= q(2);
end;

plot(t5,q5,'m-','Linewidth',3);
xlabel('t') ;
ylabel('q') ;
legend('NEWMARK-Trapeze') ;
```

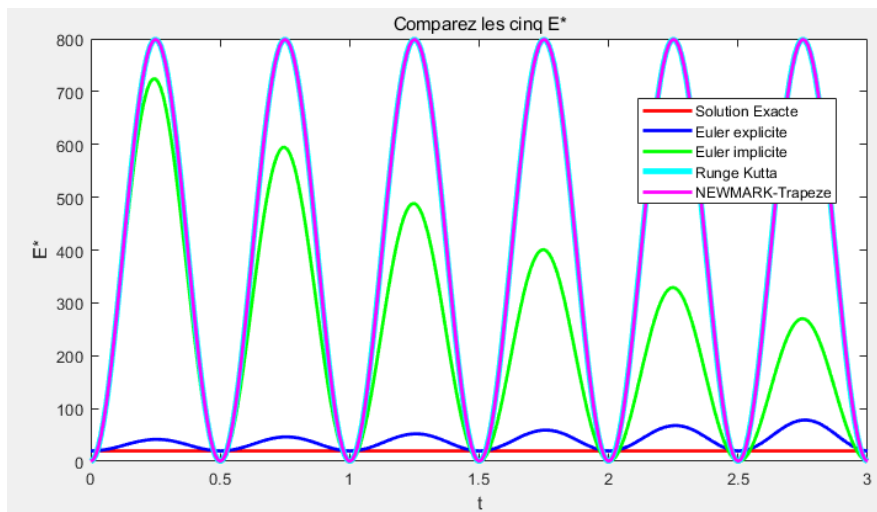


5.1.2 Comparer les cinq solutions

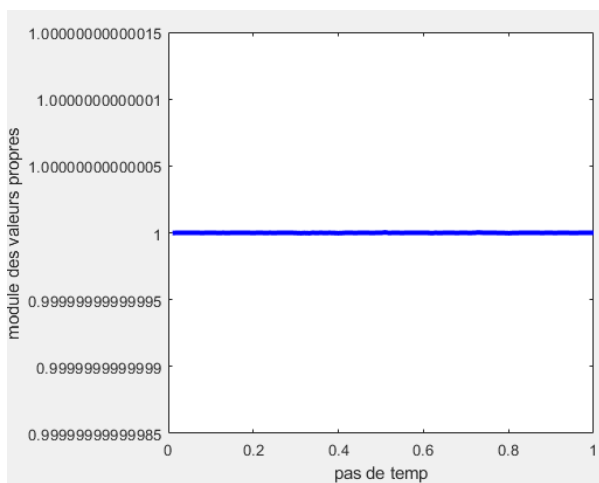


Le schéma Runge Kutta et le schéma NEWMARK sont parfaitement coincidents avec la solution exacte.

5.1.3 Comparer les cinq E*



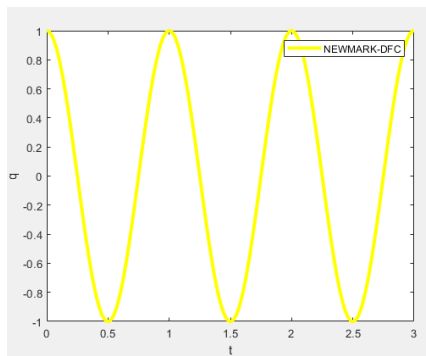
5.1.4 Calculer les valeurs propres pour $\Delta t \in [0,1]$



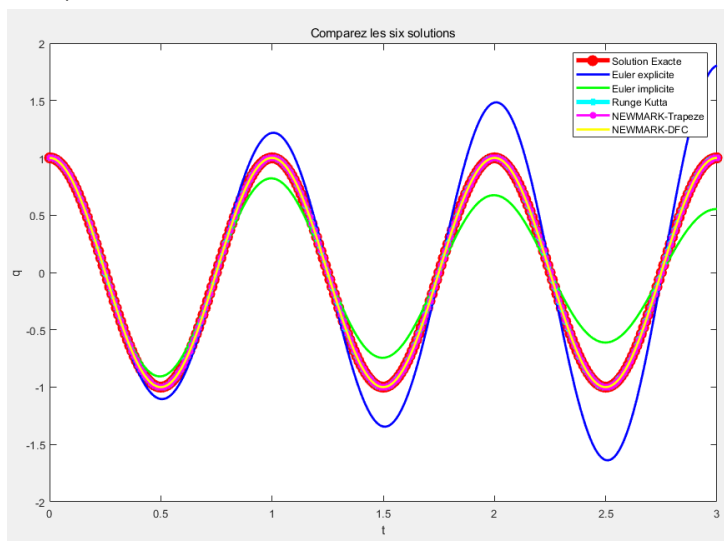
Les valeurs propres ne change pas, et la module des valeurs propres est équivalent a 1.

5.2 Schéma des différences finies centrées

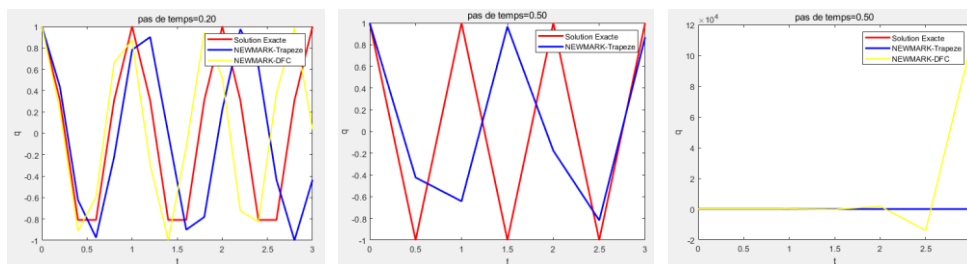
5.2.1 Programmer



5.2.2 Comparer 6 solutions

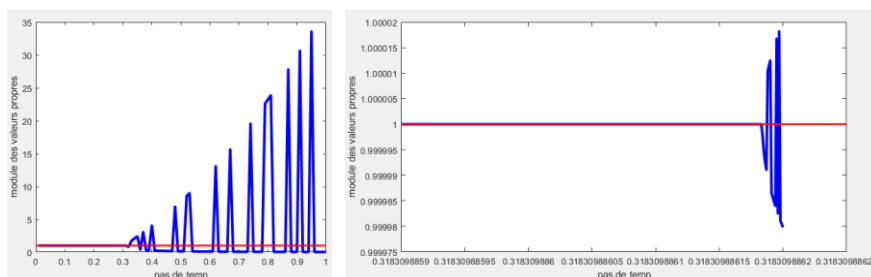


5.2.3 Comparer 3 solutions



Quand le pas de temps est 0.2s, la solution NEWMARK trapèze a une fréquence plus grande que la solution exacte, mais l'autre solution NEWMARK est le cas contraire. Quand le pas de temps est 0.5s, la solution NEWMARK différences finies centrées divergent très rapidement après 1.5 seconde.

5.2.4 Déterminer le pas de temps critique



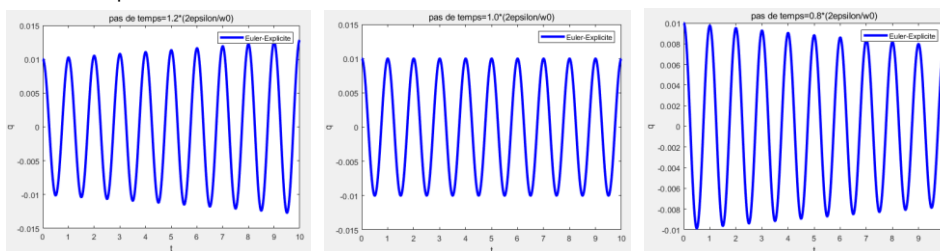
Le pas de temps critique est environ 0.3183098862s, donc $\alpha = 1$.

Rendu sur le serveur pédagogique :

Ecrire un rapport avec vos lignes de code en Matlab ou Scilab, vos sorties graphiques et réponses aux questions de l'exercice « étude d'un oscillateur linéaire amorti à un degré de liberté » « étude d'un double pendule avec l'hypothèse des petits mouvements » « étude d'un oscillateur non linéaire à un degré de liberté »

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

1.1 EULER Explicite

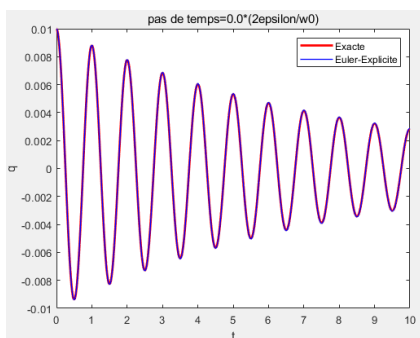


Divergente

pas de temps critique

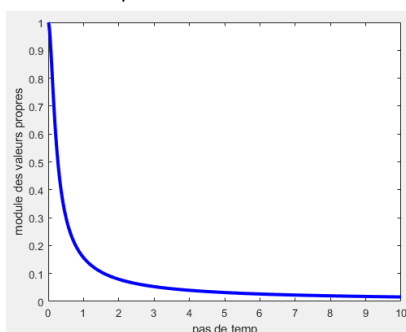
Convergente

Les erreurs avec la solution exacte sont les critères permettant d'étudier la précision.



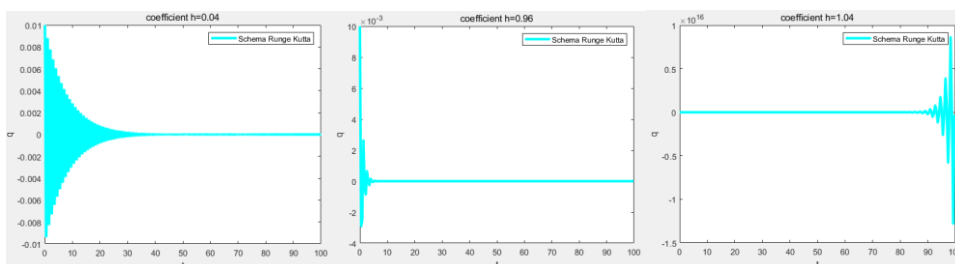
Quand le rapport supérieur à 0.01, la précision est suffisante.

1.2 EULER Implicite

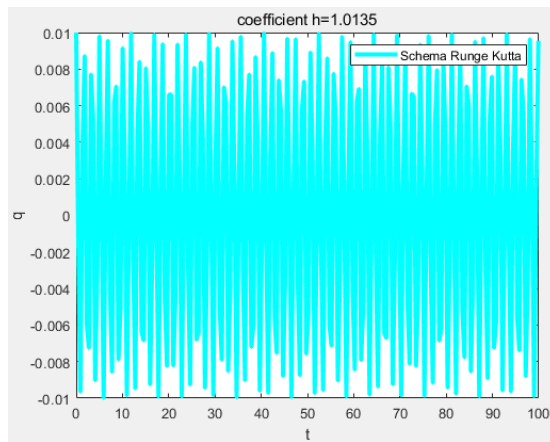


La solution numérique obtenue est inconditionnellement stable, on ne trouve pas de pas de temps critique. S'il existe, c'est peut-être 0.

1.3 RUNGE KUTTA



Quand h est 0.04, la solution est stable et a une bonne précision. Au fur et a mesure h est plus grand, la solution est convergente et puis devient divergente.
Le pas de temps critique : $hc=1.0135$.



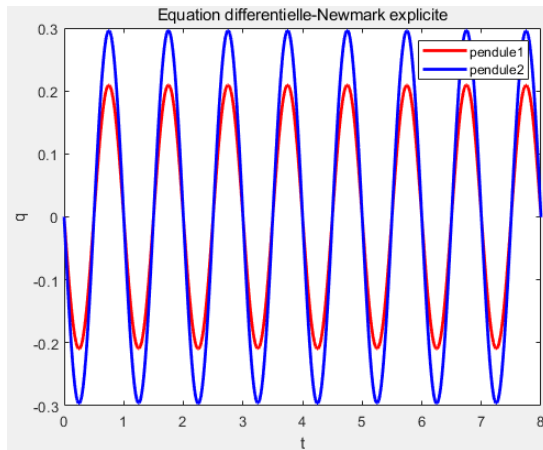
Étude d'un double pendule avec l'hypothèse des petits mouvements

1. Résolution avec un schéma de NEWMARK explicite

```
clear all ; close all ; clc
m=2;a=0.5;g=9.81;F0=20;w=2*pi;
q10=0;dq10=-1.31519275;q20=0;dq20=-1.85996342;
T0=8;
gam=0.5;bet=0;
dt4=0.02;
t4=(0:dt4:T0)';
np4=size(t4,1);
q1=zeros(np4,1);
dq1=zeros(np4,1);
ddq1=zeros(np4,1);
q2=zeros(np4,1);
dq2=zeros(np4,1);
ddq2=zeros(np4,1);
q1(1)=q10;
dq1(1)=dq10;
q2(1)=q20;
dq2(1)=dq20;
ddq1(1)=(-2*g*q1(1)+g*q2(1))/a;
ddq2(1)=(-2*g*q2(1)+2*g*q1(1))/a;

for i=2:np4
    q1(i)=q1(i-1)+dt4*dq1(i-1)+0.5*dt4^2*ddq1(i-1);
    q2(i)=q2(i-1)+dt4*dq2(i-1)+0.5*dt4^2*ddq2(i-1);
    ddq1(i)=(-2*g*q1(i)+g*q2(i))/a+F0*(1-0.5*2^0.5)*sin(w*(i-1)*dt4)/(m*a);
    ddq2(i)=(-2*g*q2(i)+2*g*q1(i))/a+F0*(2^0.5-1)*sin(w*(i-1)*dt4)/(m*a);
    dq1(i)=dq1(i-1)+0.5*dt4*(ddq1(i-1)+ddq1(i));
    dq2(i)=dq2(i-1)+0.5*dt4*(ddq2(i-1)+ddq2(i));
end
plot(t4,q1,'r-','Linewidth',2);
hold on
plot(t4,q2,'b-','Linewidth',2);
```

Quand $\Delta t = 0.02$ s



$q(t) = [\theta_1(t), \theta_2(t)]$	$t = 0$	$t = \Delta t$	$t = 2\Delta t$	$t = 0.5$
$q(t)$	[0,0]	[-0.0263, -0.0372]	[-0.0522, -0.0738]	[-2.9882e-04, -4.2260e-04]
$\dot{q}(t)$	[-1.3152, -1.8600]	[-1.3048, -1.8453]	[-1.2739, -1.8016]	[1.3143, 1.8587]
$\ddot{q}(t)$	[0,0]	[1.0365, 1.4658]	[2.0567, 2.9085]	[0.0034, 0.0049]

A l'instant $t=0$, la vitesse atteint une valeur maximale locale, l'angle est zéro, l'accélération est zéro. À l'instant $t=0.5$, l'accélération et l'angle sont presque zéro, mais la direction de la vitesse change.

2. Résolution avec un schéma de NEWMARK implicite

On utilise la méthode d'itération, on obtient la matrice correction.

$$\begin{pmatrix} \Delta_1 \ddot{q}_{n+1} \\ \Delta_2 \ddot{q}_{n+1} \end{pmatrix} = \begin{bmatrix} 2ma^2 + 2mga\beta\Delta t^2 & ma^2 \\ ma^2 & ma^2 + mga\beta\Delta t^2 \end{bmatrix}^{-1} \begin{pmatrix} -f_1^* \\ -f_2^* \end{pmatrix}$$

Programmer :

```
clear all ; close all ; clc
m=2;a=0.5;g=9.81;F0=20;w=2*pi;
q30=0;dq30=-1.31519275;q40=0;dq40=-1.85996342;
T0=8;
dt3=0.02;

t3=(0:dt3:T0)';
np3=size(t3,1);
q3=zeros(np3,1);
dq3=zeros(np3,1);
ddq3=zeros(np3,1);
q4=zeros(np3,1);
dq4=zeros(np3,1);
ddq4=zeros(np3,1);

residu=zeros(np3,1);
niter=zeros(np3,1);
threshold=1e-8;
nitermax=20;

gam=0.5;
```



```
gam1=(1-gam)*dt3;
```

```
gam2=gam*dt3;
```

```
bet=0.25;
```

```
bet1=(0.5-bet)*dt3^2;
```

```
bet2=bet*dt3^2;
```

```
q3(1)=q30;
```

```
dq3(1)=dq30;
```

```
ddq3(1)=(-2*g*q3(1)+g*q4(1))/a;
```

```
q4(1)=q40;
```

```
dq4(1)=dq30;
```

```
ddq4(1)=(-2*g*q4(1)+2*g*q3(1))/a;
```

```
res1=m*a^2*(2*ddq3(1)+ddq4(1))+m*g*a^2*q3(1);
```

```
res2=m*a^2*(ddq3(1)+ddq4(1))+m*g*a^2*q4(1);
```

```
residu(1)=(abs(res1)^2+abs(res2)^2)^.5;
```

```
KT=[0 0;0 0];
```

```
for i=2:np3
```

```
    ddq3c=0;
```

```
    dq3c=dq3(i-1)+gam1*ddq3(i-1);
```

```
    q3c=q3(i-1)+dt3*dq3(i-1)+bet1*ddq3(i-1);
```

```
    ddq4c=0;
```

```
    dq4c=dq4(i-1)+gam1*ddq4(i-1);
```

```
    q4c=q4(i-1)+dt3*dq4(i-1)+bet1*ddq4(i-1);
```

```
    res1=m*a^2*(2*ddq3c+ddq4c)+m*g*a^2*q3c-F0*sin(w*(i-1)*dt3)*a;
```

```
    res2=m*a^2*(ddq3c+ddq4c)+m*g*a^2*q4c-F0*sin(w*(i-1)*dt3)*a*0.5*2^0.5;
```

```
    res=(abs(res1)^2+abs(res2)^2)^.5;
```

```
    iter=0;
```

```
    for j=1:nitermax
```

```
        if (abs(res)>=threshold)
```

```
            iter=iter+1;
```

```
            KT=[2*m*a^2+m*g*a*bet2,m*a^2;m*a^2,m*a^2+m*g*a*bet2];
```

```
            hj=KT\[-res1;-res2];
```

```
            correction1=hj(1);
```

```
            correction2=hj(2);
```

```
            ddq3c=ddq3c+correction1;
```

```
            dq3c=dq3c+gam2*correction1;
```

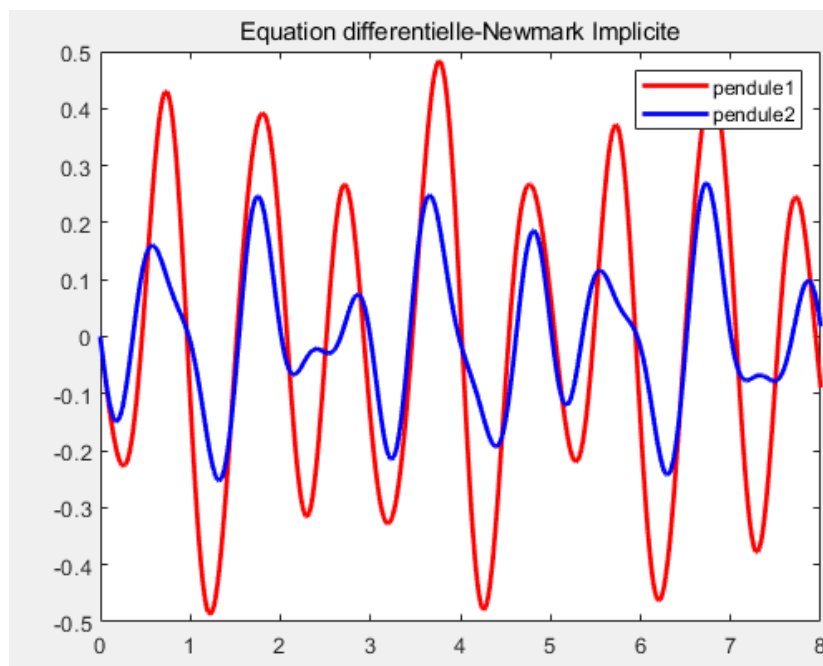
```
            q3c=q3c+bet2*correction1;
```

```

        ddq4c=ddq4c+correction2;
        dq4c=dq4c+gam2*correction2;
        q4c=q4c+bet2*correction2;
        res1=m*a^2*(2*ddq3c+ddq4c)+m*g*a*2*q3c-F0*sin(w*(i-
1)*dt3)*a;
        res2=m*a^2*(ddq3c+ddq4c)+m*g*a*2*q4c-F0*sin(w*(i-
1)*dt3)*a*0.5*2^0.5;
        res=(abs(res1)^2+abs(res2)^2)^.5;
    end
end
q3(i)=q3c;
dq3(i)=dq3c;
ddq3(i)=ddq3c;
q4(i)=q4c;
dq4(i)=dq4c;
ddq4(i)=ddq4c;
niter(i)=iter;
residu(i)=res;
end
plot(t3,q3,'r-','Linewidth',2);
hold on
plot(t3,q4,'b-','Linewidth',2);
title('Equation differentielle-Newmark Implicite');
legend('pendule1','pendule2');

```

Quand $\Delta t = 0.02$ s ;



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

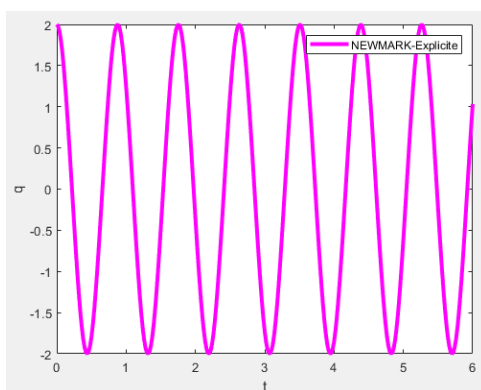
$$\ddot{q} + \omega_0^2 q(1 + aq^2) = 0$$

1.1 Relations

$$\begin{cases} q_{j+1} = q_j + \Delta t \dot{q}_j + 0.5 \Delta t^2 \ddot{q}_j \\ \dot{q}_{j+1} = \dot{q}_j + 0.5 \Delta t (\ddot{q}_j + \ddot{q}_{j+1}) \\ \ddot{q}_{j+1} = -\omega_0^2 q_{j+1} (1 + a q_{j+1}^2) \end{cases}$$

1.2 Programmation

```
clear all ; close all ; clc
T0=6;w0=2*pi;w0c=w0*w0;
q0=2;dq0=0;a=0.1;
dt6=0.02;
t6=(0:dt6:T0)';
np6=size(t6,1);
q6=zeros(np6,1);
dq6=zeros(np6,1);
q6(1)=q0;
dq6(1)=dq0;
ddq0c=-w0c*q0*(1+a*q0^2);
for inc=2:np6
q6(inc)=q6(inc-1)+dt6*dq6(inc-1)+dt6*dt6*0.5*ddq0c;
ddqc=-w0c*q6(inc)*(1+a*q6(inc)^2);
dq6(inc)=dq6(inc-1)+0.5*dt6*(ddq0c+ddqc);
ddq0c=ddqc;
end
plot(t6,q6,'m-','Linewidth',3);
xlabel('t');
ylabel('q');
legend('NEWMARK-Explicite');
```



1.3 Quand $\Delta t = 0.02s$

t/s	q(t)
0	2
Δt	1.9779
$2\Delta t$	1.9123
T0	1.0329

2.1

On cherche à minimiser les erreurs des valeurs d'estimation pour avoir une bonne précision.

2.2 Calculer la correction

$$\Delta \ddot{q}_{j+1} = \frac{-(\ddot{q}_{j+1}^* + \omega_0^2 q_{j+1}^* (1 + a q_{j+1}^{*2}))}{1 + \omega_0^2 \beta \Delta t^2 (1 + 3a q_{j+1}^{*2})}$$

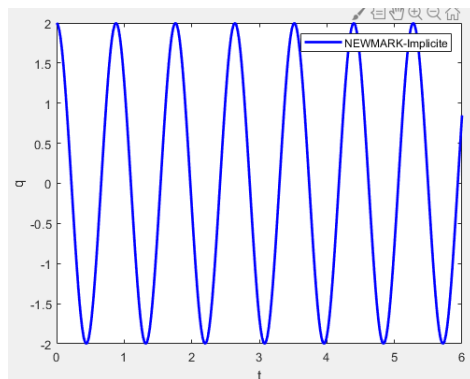
2.3 Programmation

```
clear all ; close all ; clc
T0=6;
w0=2*pi;
w0c=w0^2;
q0=2;
dq0=0.0;
a=0.1;
dt3=0.02;
t3=(0:dt3:T0)';
np3=size(t3,1);
q3=zeros(np3,1);
dq3=zeros(np3,1);
ddq3=zeros(np3,1);
energ3=zeros(np3,1);
residu=zeros(np3,1);
niter=zeros(np3,1);
threshold=1e-8;
nitermax=20;
gam=0.5;
bet=0.25;
q3(1)=q0;
dq3(1)=dq0;
ddq3(1)=-w0c*q0*(1+a*q0^2);
for i=2:np3
    ddqc=0;
    dqc=dq3(i-1)+(1-gam)*dt3*ddq3(i-1);
    qc=q3(i-1)+dt3*dq3(i-1)+(0.5-bet)*dt3^2*ddq3(i-1);
    res=ddqc+w0c*qc*(1+a*qc^2);
    iter=0;
    for j=1:nitermax
        if (abs(res)>=threshold)
            iter=iter+1;
            correction=-res/(1+w0c*bet*dt3^2*(1+3*a*qc^2));
            ddqc=ddqc+correction;
            dqc=dqc+gam*dt3*correction;
            qc=qc+bet*dt3^2*correction;
            res=ddqc+w0c*qc*(1+a*qc^2);
```

```

end
end
q3(i)=qc;
dq3(i)=dq;
ddq3(i)=ddq;
niter(i)=iter;
residu(i)=res;
end
plot(t3,q3,'b-','Linewidth',2);

```



2.4 Quand $\Delta t = 0.02s$

t/s	q(t)
0	2
Δt	1.9781
$2\Delta t$	1.9131
T_0	0.8485

3. Energie mécanique

3.1 E^*

E^* , proportionnelle à l'énergie mécanique, est défini par

$$E^* = \frac{1}{2}(\dot{q}^2 + \omega_0^2 q^2) + \frac{1}{4}a\omega_0^2 q^4$$

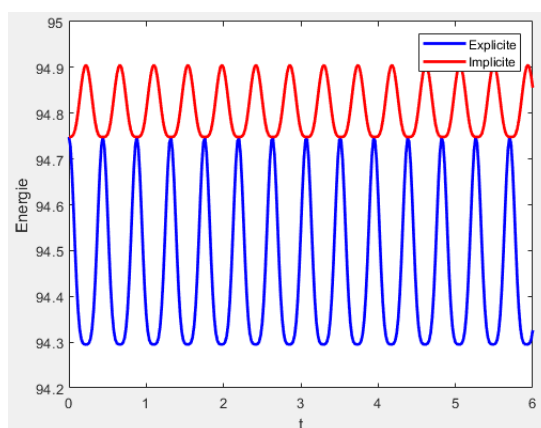
3.2 programmation

```

energ6=0.5*(dq6.*dq6+w0c*q6.^2)+(a*w0c*q6.^4)/4;

```

3.3 Quand $\Delta t = 0.02s$



On observe que Implicite Energie \geq Explicite Energie, et elles sont équivalente à quelques instants.