

Mécanique Numérique

Nicolas ChenChangyi SY1924107

Oscillateur conservatif linéaire à un degré de liberté

Question 1.1

La solution de l'équation (1) satisfaisant aux conditions initiales (4) est

$$\theta = \cos(2 * \pi * t)$$

Les codes de MATLAB :

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

Question 1.2

La quantité E^* associée à cette solution exacte est :

$$E^* = 2\pi^2$$

On peut trouver que E^* est indépendant de t

Les codes de MATLAB :

```
w0 = 2*sym(pi);  
E_etoile = simplify(0.5 * ((diff(q))^2 + (w0^2) * (q^2)));
```

Question 2.1

Montrez qu'en tenant compte des relations (1) et (5), le schéma d'EULER explicite conduit à la relation (6) entre les quantités :

On a :

$$\begin{vmatrix} q_{j+1} \\ \dot{q}_{j+1} \end{vmatrix} = \begin{vmatrix} q_j \\ \dot{q}_j \end{vmatrix} + \Delta t \times \begin{vmatrix} \dot{q}_j \\ \ddot{q}_j \end{vmatrix}$$

Donc

$$\begin{aligned} q_{j+1} &= q_j + \Delta t \times \dot{q}_j \\ \dot{q}_{j+1} &= \dot{q}_j + \Delta t \times \ddot{q}_j \end{aligned}$$

Or

$$\ddot{q} + w_0^2 q = 0$$

Alors

$$\dot{q}_{j+1} = \dot{q}_j - \Delta t \times w_0^2 q_j$$

Donc

$$\begin{vmatrix} q_{j+1} \\ \dot{q}_{j+1} \end{vmatrix} = \begin{bmatrix} 1 & \Delta t \\ -\Delta t \times w_0^2 & 1 \end{bmatrix} \begin{vmatrix} q_j \\ \dot{q}_j \end{vmatrix}$$

Question 2.2

a) Méthode 1

Les codes de MATLAB :

```
clear all;  
dt1 = 0.01;
```

```

T0 =3;
q0 =1;
dq0 = 0;
w0c = (2 * pi) ^ 2;
t1 = (0 : dt1 : T0)';
np1 = size(t1,1);
q1 = zeros(np1,1);
dq1 = 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,'r+', 'linewidth',3)

```

b) Méthode 2

Les codes de MATLAB :

```

clear all;
dt1 = 0.1;
T0 =3;
q0 =1;
dq0 = 0;
w0c = (2 * pi) ^ 2;
t1 = (0 : dt1: T0)';
np1 = size(t1,1);
q = [q0;dq0];
q1 = zeros(np1,1);
q1(1) = q0;
A = [1, dt1;-w0c * dt1,1];
for inc =2 :np1
    q = A * q;
    q1(inc) = q(1);
    dq1(inc) = q(2);
end;
plot(t1,q1,'r+', 'linewidth',3);

```

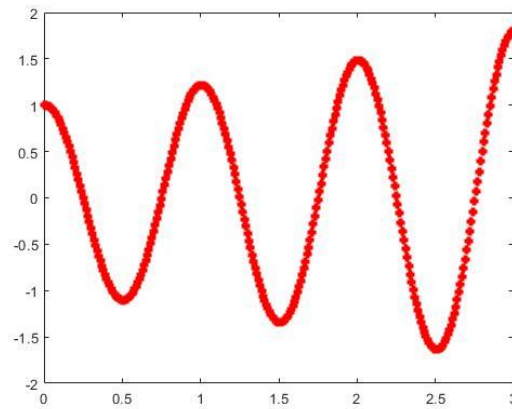


figure 1 La valeur de q fournies par Euler explicite ($\Delta t=0.01s$)

Question 2.3

Quand on change le pas de temps Δt de 0.01 à 0.001, on peut trouver la divergence est plus petit.

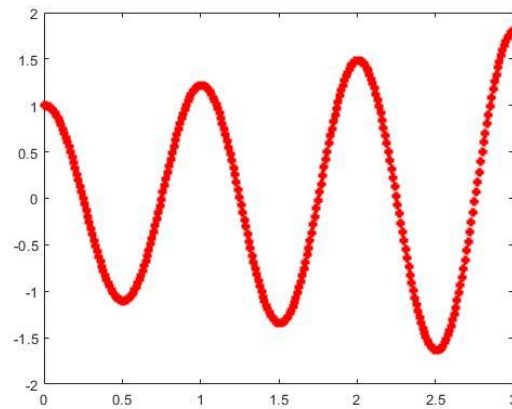


figure 2 La valeur de q fournies par Euler explicite ($\Delta t=0.01s$)

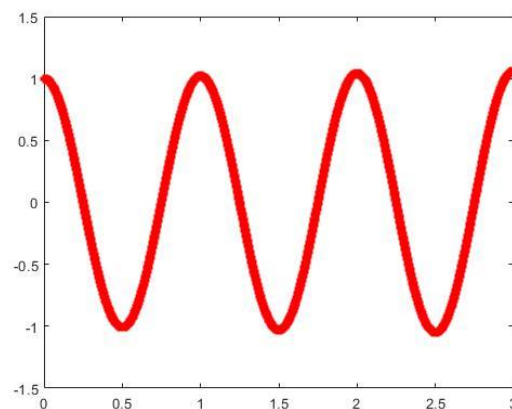


figure 3 La valeur de q fournies par Euler explicite ($\Delta t=0.001s$)

Question 2.4

Pour méthode 1 par exemple, les codes de MATLAB :

```
E_etoile = 0.5 * ((dq1).^2 + w0c * (q1.^2));
```

```
plot(t1,E_etoile,'r+', 'linewidth',3);
```

Et la comparaison de la quantité $E^\star$ est :

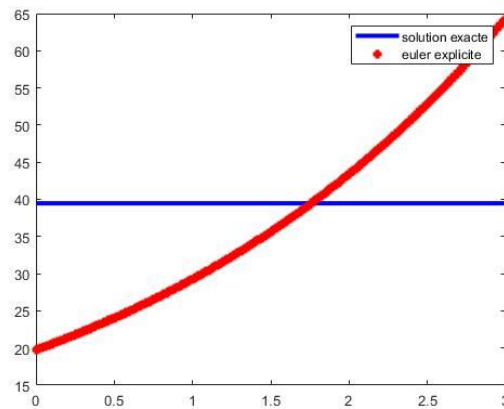


figure 4 Les valeurs de $E^\star$ fournies par les deux solutions ($\Delta t=0.01s$)

On peut trouver que la quantité $E^\star$ augmente avec le temps, quand on change le pas de temps $dt1$ de 0.01 à 0.001, on peut trouver que la quantité $E^\star$ augmente plus lentement

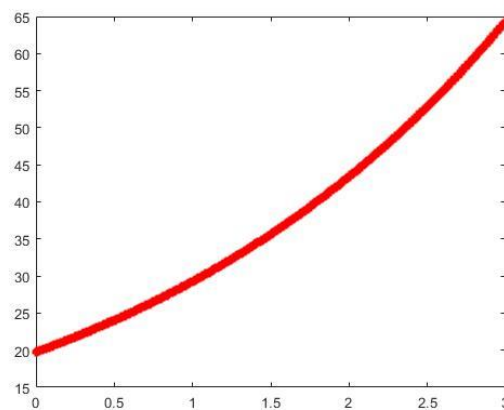


figure 5 La valeur de $E^\star$ fournies par Euler explicite ($\Delta t=0.01s$)

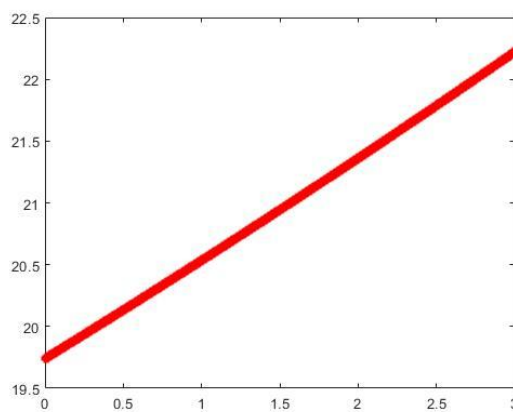


figure 6 La valeur de $E^\star$ fournies par Euler explicite ($\Delta t=0.001s$)

Question 2.5

Pour calculer les valeurs propres de la matrice d'amplification en fonction du pas de temps Δt ,

Les codes de MATLAB :

```
clear all;
w0c = (2 * pi) ^ 2;
dt1 = sym('dt1','real');
A = [1, dt1; -w0c * dt1, 1];
[z,d]=eig(A); z ; d
```

Le caractère inconditionnellement instable du schéma d'EULER explicite est alors $\Delta t > \frac{1}{2\pi^2}$

Question 3.1

a) Méthode 1

Les codes de MATLAB :

```
clear all;
dt2 = 0.01;
T0 =3;
q0 =1;
dq0 = 0;
w0c = (2 * pi) ^ 2;
t2 = (0:dt2:T0)';
np2 = size(t2,1);
q2 = zeros(np2,1);
dq2 = 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
plot(t2,q2,'g*','Linewidth',3)
```

b) Méthode 2

Les codes de MATLAB :

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

```

np2 = size(t2,1);
q = [q0;dq0];
q2b = zeros(np2,1);
dq2b = zeros(np2,1);
q2b(1) = q0;
dq2b(1) = dq0;
A = [1 , dt2 ; -w0c * dt2 , 1];
A = A / (1 + w0c * dt2 * dt2);
for inc = 2 : np2
    q = A * q;
    q2b(inc) = q(1);
    dq2b(inc) = q(2);
end;
plot(t2,q2b,'g*', 'Linewidth',3)

```

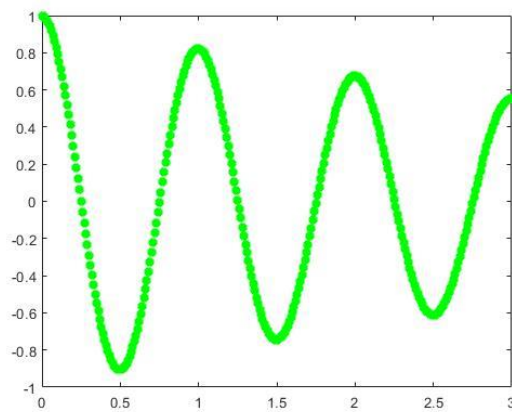


figure 7 La valeur de q fournies par Euler implicite ($\Delta t=0.01s$)

Question 3.2

Les codes de MATLAB :

```

clear all;
t = (0 : 0.01 : 3);
plot(t,cos(2 * pi * t),'b-', 'linewidth',3);
hold on;
dt1 = 0.01;
T0 =3;
q0 =1;
dq0 = 0;
w0c = (2 * pi) ^ 2;
t1 = (0 : dt1 : T0)';
np1 = size(t1,1);
q1 = zeros(np1,1);
dq1 = 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,'r+', 'linewidth',3);
hold on;
dt2 = 0.01;
T0 =3;
q0 =1;
dq0 = 0;
w0c = (2 * pi) ^ 2;
t2 = (0:dt2:T0)';
np2 = size(t2,1);
q2 = zeros(np2,1);
dq2 = 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
plot(t2,q2,'g*', 'Linewidth',3)

```

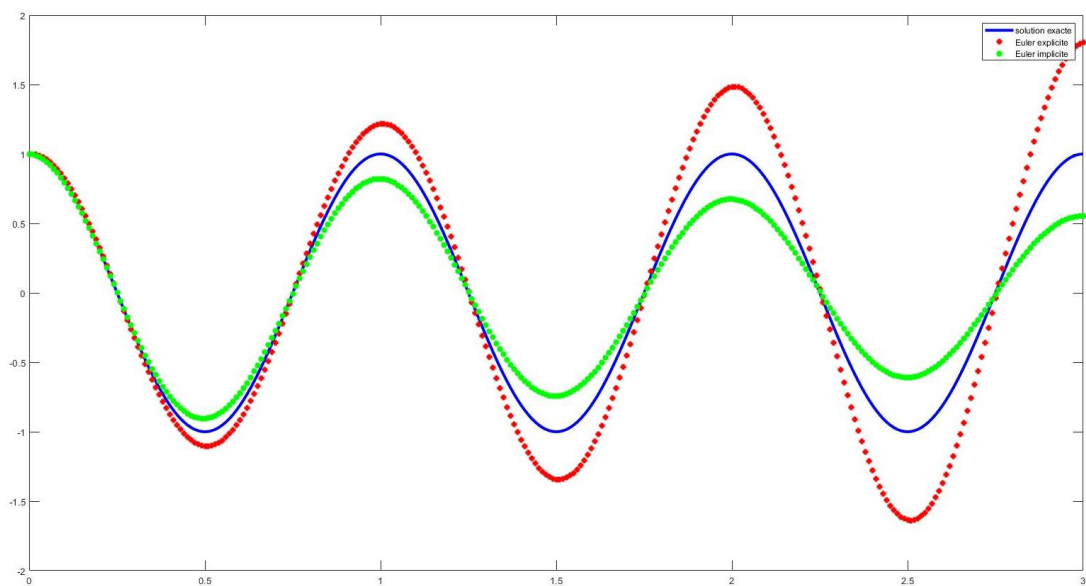


figure 8 Les valeurs de q fournies par les trois solutions ($\Delta t=0.01s$)

Question 3.3

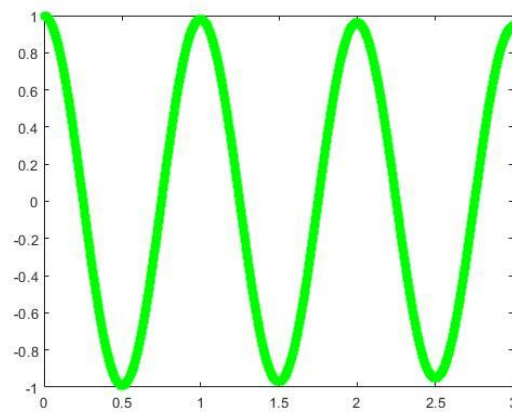


figure 9 La valeur de q fournies par Euler implicite ($\Delta t=0.001s$)

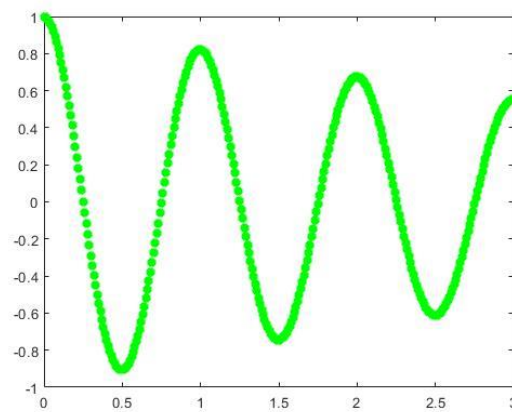


figure 10 La valeur de q fournies par Euler implicite ($\Delta t=0.01s$)

On peut trouver que le schéma d'intégration d'EULER implicite introduit un amortissement numérique.

Question 3.4

Pour méthode 1 par exemple, les codes de MATLAB :

```
E_etoile = 0.5*(dq2 .* dq2 + w0c * (q2.^2)) ;  
plot(t2,E_etoile, 'g*', 'linewidth',3);
```

Et la comparaison de la quantité E^* est :

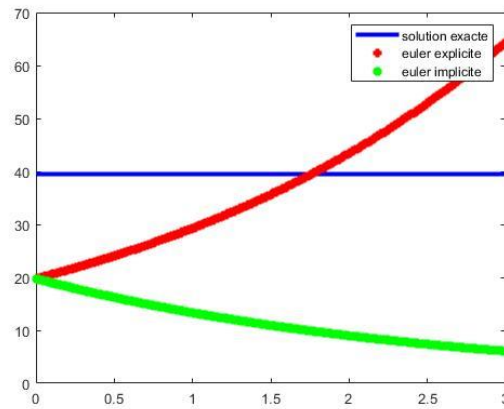


figure 11 Les valeurs de E^* fournies par les trois solutions ($\Delta t=0.01s$)

On peut trouver que la quantité E^* diminue avec le temps, quand on change le pas de temps dt_2 de 0.01 à 0.001, on peut trouver que la quantité E^* diminue plus lentement

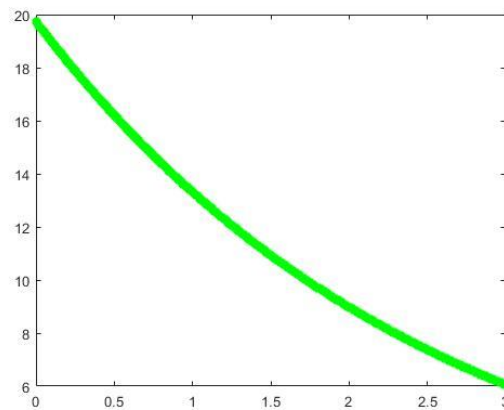


figure 12 La valeur de E^* fournies par Euler implicite ($\Delta t=0.01s$)

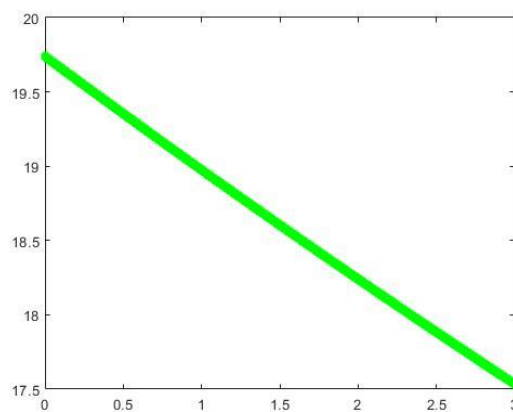


figure 13 La quantité $E^*(dt_2=0.001s)$

Question 3.5

Pour calculer les valeurs propres de la matrice d'amplification en fonction du pas de temps dt_2 , les codes de MATLAB :

```
clear all;
```

```
w0c = (2 * pi) ^ 2;
dt1 = sym('dt2','real');
A = [1, dt2;-w0c * dt2,1];
[z,d]=eig(A); z ; d
```

Le caractère inconditionnellement instable du schéma d'EULER explicite est alors $\Delta t > \frac{1}{2\pi^2}$

Question 4.1

On a équation (1) :

$$\ddot{q} + w_0^2 q = 0$$

On peut le transformer en :

$$\ddot{q} = -w_0^2 q$$

On prend

$$y(t) = \begin{pmatrix} q(t) \\ \dot{q}(t) \end{pmatrix}$$

$$\dot{y}(t) = \begin{pmatrix} \dot{q}(t) \\ \ddot{q}(t) \end{pmatrix}$$

Alors on a :

$$\dot{y}(t) = \begin{pmatrix} 0 & 1 \\ -w_0^2 & 0 \end{pmatrix} y(t)$$

C'est à dire que $\dot{y}(t) = f(y(t), t)$ avec $f(y(t), t) = \begin{pmatrix} 0 & 1 \\ -w_0^2 & 0 \end{pmatrix} y(t)$

Question 4.2

Les codes de MATLAB :

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

```
dt6 = 0.01;
t6 = (0:dt6:3)';
np6 = size(t6,1);
q0 = 1 ;
dq0 =0 ;
q6 = zeros(np6,1);
dq6 = zeros(np6,1);
q6(1) = q0;
dq6(1) = dq0;
qj = [q0 ; dq0];
for inc = 2 : np6
    tc = t6(inc-1);
    xc = qj;
```

```

k1 = cal_f(xc);
xc = qj + k1 * dt6 / 2;
k2 = cal_f(xc);
xc = qj + k2 * dt6 / 2;
k3 = cal_f(xc);
xc = qj + k3 * dt6;
k4 = cal_f(xc);
dq = (k1 + 2 * k2 + 2 * k3 + k4) / 6;
qj = qj + dq * dt6;
q6(inc) = qj(1);
dq6(inc) = qj(2);
end
plot(t6,q6,' y*', 'Linewidth',3)

```

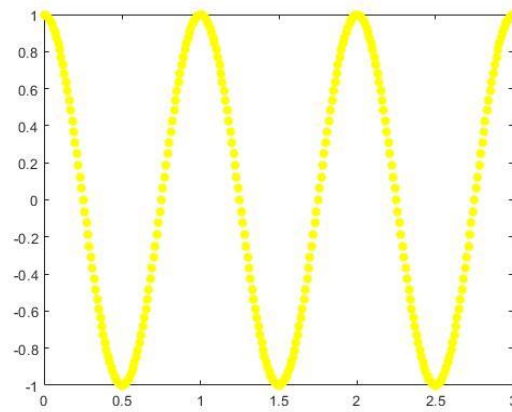


figure 14 La valeur de q fournies par Runge Kutta ($\Delta t=0.01s$)

Question 4.3

En utilisant les codes des question 3.2 et 4.2 on a cette figure :

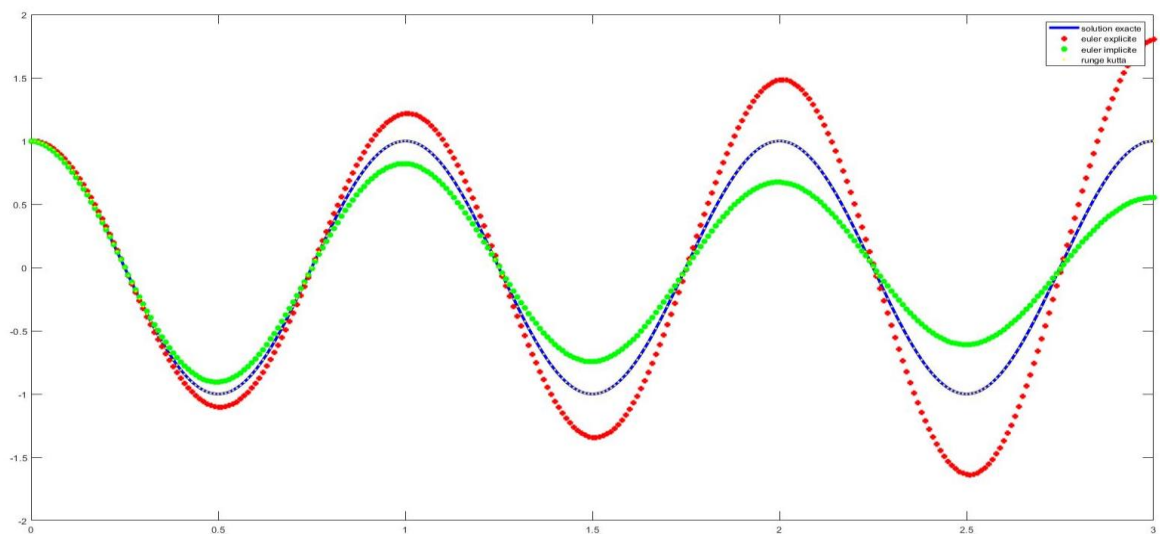


figure 15 Les valeurs de q fournies par les quatre solutions ($\Delta t=0.01s$)

On peut trouver que le schéma de RUNGE KUTTA est plus précis quand le pas du temps est même.

Question 4.4

Les codes de MATLAB :

```
E_etoile = 0.5 * ((dq6).^2 + w0c * (q6.^2));  
plot(t6,E_etoile,'y*', 'linewidth',3);
```

Et la comparaison de la quantité E^* est :

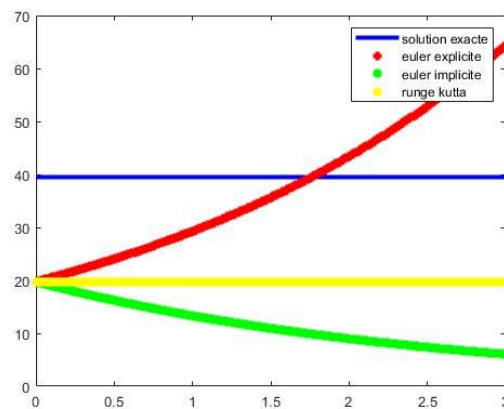


figure 16 Les valeurs de E^* fournies par les quatre solutions ($\Delta t=0.01s$)

On peut trouver que la quantité E^* est invariant du temps.

Question 5.1.1

En utilisant les trois relations (13) , (14) et (15), on a :

Les codes de MATLAB :

```
clear all;  
dt7 = 0.01;  
gamma = 0.5;  
beta = 0.25;  
w0c = (2 * pi) ^ 2;  
t7 = (0:dt7:3)';  
np7 = size(t7,1);  
q7 = zeros(np7,1);  
dq7 = zeros(np7,1);  
energ7 = zeros(np7,1);  
q7(1) = 1;  
dq7(1) = 0;  
ddq7c = -w0c;  
for inc = 2 : np7  
q7(inc) = (q7(inc-1) + dt7 * dq7(inc-1) + dt7 * dt7 * (0.5 - beta) *  
ddq7c) / (1 + dt7 * dt7 * beta * w0c) ;
```

```

ddqc7 = -w0c * q7(inc);
dq7(inc) = dq7(inc-1) + (1 - gamma) * dt7 * (ddq7c + ddqc7);
ddq7c = ddqc7;
end
plot(t7,q7,'m--','linewidth',3);

```

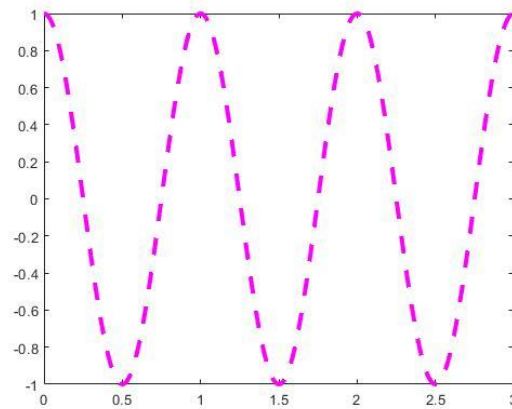


figure 17 La valeur de q fournies par Newmark($\gamma = 0.5, \beta = 0.25, \Delta t = 0.1s$)

Question 5.1.2

En utilisant les codes des question 3.2, 4.2 et 5.1.1 on a cette figure :

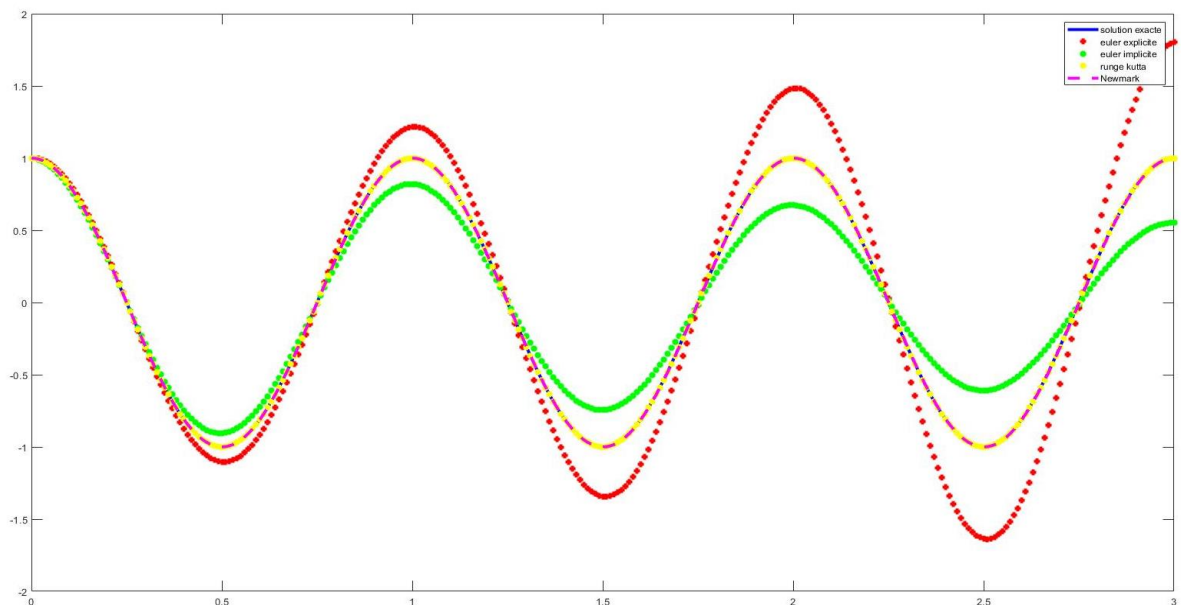


figure 18 Les valeurs de q fournies par les cinq solutions ($\Delta t = 0.01s$)

Question 5.1.3

Les codes de MATLAB :

```

E_etoile = 0.5 * ((dq7).^2 + w0c * (q7.^2));
plot(t7,E_etoile,'m--','linewidth',3);

```

Et la comparaison de la quantité E^* est :

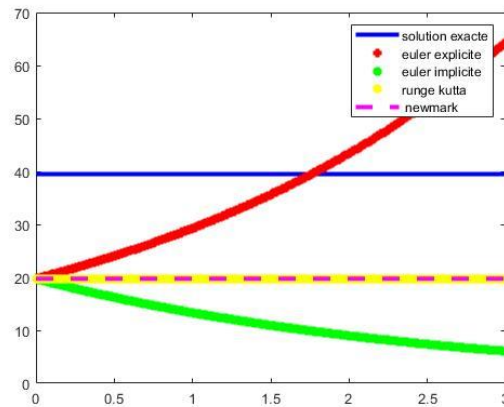


figure 19 Les valeurs de E^* fournies par les cinq solutions ($\Delta t=0.01s$)

On peut trouver que la quantité E^* est invariant du temps comme Runge Kutta.

Question 5.1.3

Les codes de MATLAB :

```
clear all;
dt7 = 0.01;
gamma = 0.5;
beta = 0.25;
w0c = (2 * pi) ^ 2;
C = [1 - dt7 * dt7 * (0.5 - beta) * w0c , dt7 ; - (1 - gamma) * dt7 * w0c , 1];
B = [1 + beta * dt7 * dt7 * w0c , 0 ; gamma * dt7 * w0c , 1];
A = inv(B) * C;
[z,d]=eig( (inv(B) ) * C);
re = real(d);
im = imag(d);
mo= abs(d);
```

Quand on change Δt de 0 à 1, on peut trouver que le module des valeurs propres est toujours égal à 1.

Question 5.2.1

Les codes de MATLAB :

```
dt8 = 0.01;
gamma = 0.5;
beta = 0;
w0c = (2 * pi) ^ 2;
t8 = (0:dt8:3)';
np8 = size(t8,1);
q8 = zeros(np8,1);
dq8 = zeros(np8,1);
```

```

energ8 = zeros(np8,1);
q8(1) = 1;
dq8(1) = 0;
ddq8c = -w0c;
for inc = 2 : np8
q8(inc) = q8(inc-1) + dt8 * dq8(inc-1) + dt8 * dt8 * (0.5 - beta) *
ddq8c;
ddqc8 = -w0c * q8(inc);
dq8(inc) = dq8(inc-1) + (1 - gamma) * dt8 * (ddq8c + ddqc8);
ddq8c = ddqc8;
end
plot(t8,q8,'k^','linewidth',3);

```

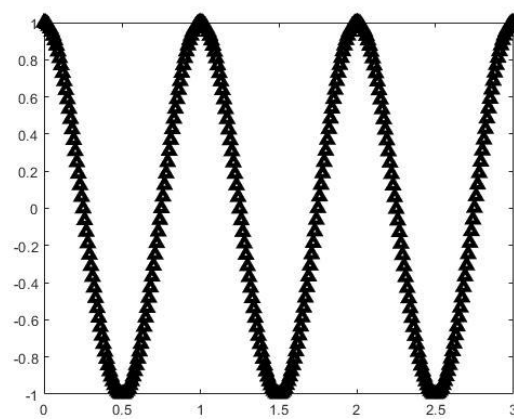


figure 20 La valeur de q fournies par Newmark ($\gamma = 0.5, \beta = 0, \Delta t = 0.1s$)

Question 5.2.2

En utilisant les codes des question 3.2, 4.2, 5.1.1 et 5.2.1 on a cette figure :

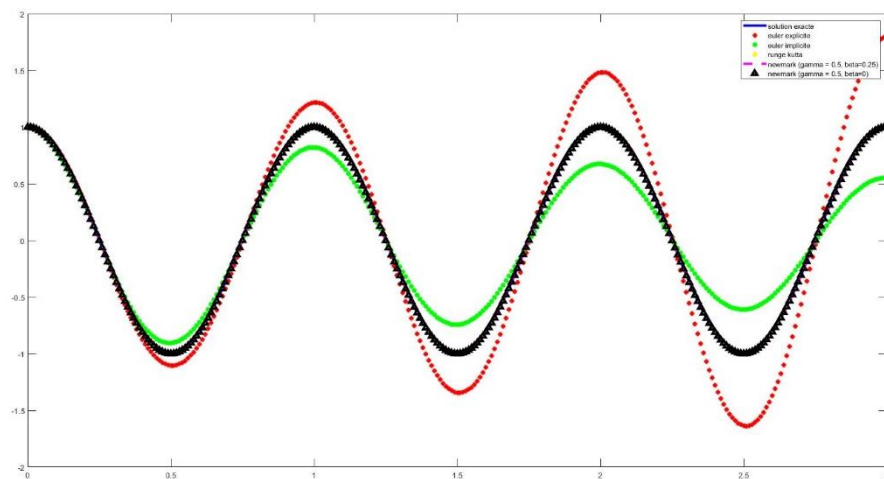


figure 21 Les valeurs de q fournies par les six solutions ($\Delta t=0.01s$)

On peut trouver que pour résoudre cette question, les deux méthodes de Newmark n'a pas beaucoup de différence.

Question 5.2.3

On trouve le pas de temps critique par faire une étude des valeurs propres de la matrice d'amplification.

Les codes de MATLAB :

```
clear all;
dt8 = 0.00;
gamma = 0.5;
beta = 0;
w0c = (2 * pi) ^ 2;
C = [1 - dt8 * dt8 * (0.5 - beta) * w0c , dt8 ; - (1 - gamma) * dt8 *
w0c , 1];
B= [1 + beta * dt8 * dt8 * w0c , 0 ; gamma * dt8 * w0c ,1];
A = inv(B) * C;
[z,d]=eig( (inv(B) ) * C);
re = real(d);
im = imag(d);
mo= abs(d);
```

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

Question 1.1.a

On peut calculer que $\frac{2\varepsilon}{\omega_0} = 0.0064s$, donc on peut choisir $\Delta t = 0.01s$ par exemple.

Les codes de MATLAB :

```
clear all;
dt1 = 0.01;
T0 = 1;
epsilon = 0.02;
w0 = (2 * pi) / T0;
x0 = 0.01;
dx0 = 0;
t1 = (0 : dt1 : 10 * T0)';
np1 = size(t1,1);
x1 = zeros(np1,1);
dx1 = zeros(np1,1);
x1(1) = x0;
dx1(1) = dx0;
ddx1(1) = -2 * epsilon * w0 * dx1(1) - w0 * w0 * x1(1);
for inc = 2 : np1
    x1(inc) = x1(inc - 1) + dt1 * dx1(inc - 1);
    dx1(inc) = dx1(inc - 1) + dt1 * ddx1(inc - 1);
    ddx1(inc) = -2 * epsilon * w0 * dx1(inc) - w0 * w0 * x1(inc);
end;
plot(t1,x1,'b-', 'linewidth',3);
```

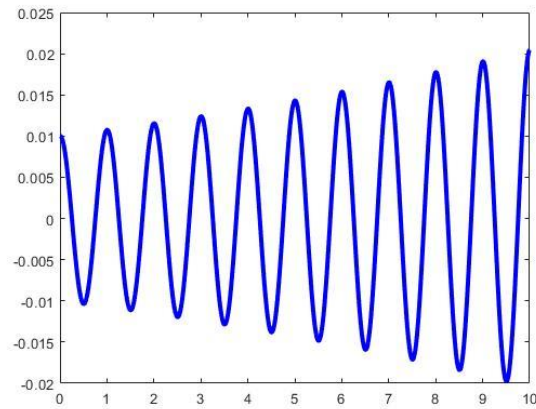


figure 22 La valeur de x fournies par Euler explicite ($\Delta t=0.01s$)

Question 1.1.b

Quand on change $\Delta t = \frac{2\varepsilon}{\omega_0} = 0.0064s$ dans les codes de MATLAB de 1.1.a, on a :

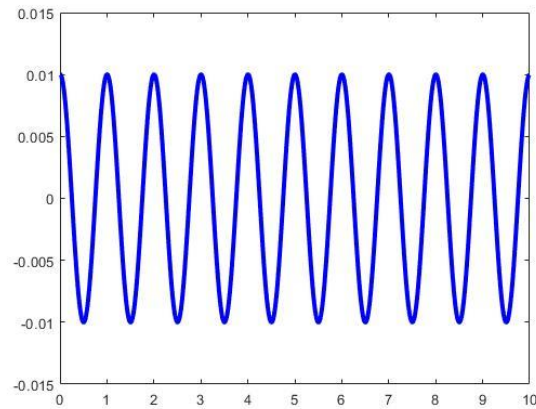


figure 23 La valeur de x fournies par Euler explicite ($\Delta t=0.0064s$)

Question 1.1.c

Quand on change $\Delta t = 0.8 \times \frac{2\varepsilon}{\omega_0} = 0.00512s$ dans les codes de MATLAB de 1.1.a, on a :

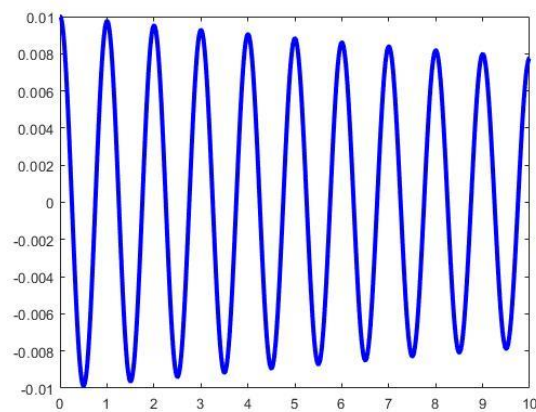


figure 24 La valeur de x fournies par Euler explicite ($\Delta t=0.00512s$)

Question 1.1.d

Les critères permettant d'étudier la précision de la solution sont la précision en période et en amplitude. A partir $\frac{\Delta t}{\frac{2\varepsilon}{\omega_0}} = 0.05$ la solution calculée présente une précision suffisante.

Question 1.2

Le pas de temps critique $\Delta t = \frac{2\varepsilon}{\omega_0} = 0.0064s$.

Les codes de MATLAB :

```
clear all;
T0 = 1;
epsilon = 0.02;
w0 = (2 * pi) / T0;
dt2 = (2 * epsilon) / w0;
x0 = 0.01;
dx0 = 0;
t2 = (0 : dt2 : 10 * T0)';
np2 = size(t2,1);
x2 = zeros(np2,1);
dx2 = zeros(np2,1) ;
x2(1) = x0;
dx2(1) = dx0;
for inc = 2 : np2
x2(inc) = ((1 + 2 * epsilon * w0 * dt2) * x2(inc-1) + dt2 * dx2(inc-1))/((1 + 2 * epsilon * w0 * dt2 ) + w0 * w0 * dt2 * dt2);
dx2(inc) = (-w0 * w0 * dt2 * x2(inc-1) + dx2(inc - 1) )/((1 + 2 * epsilon * w0 * dt2 ) + w0 * w0 * dt2 * dt2);
end
plot(t2,x2,'r','Linewidth',3)
```

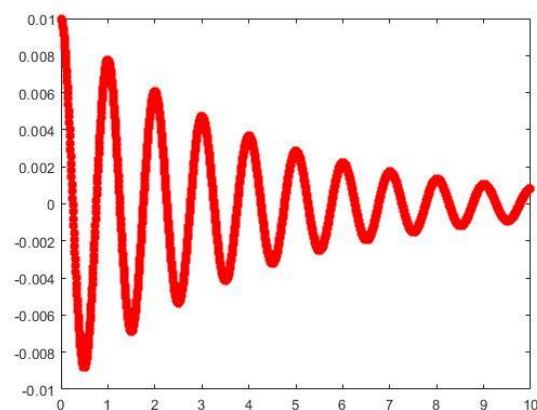


figure 25 La valeur de x fournies par Euler implicite ($\Delta t=0.0064s$)

Question 1.3.a

Les codes de MATLAB :

```

function [ dUc ] = cal_f2(Uc)
T0 = 1;
epsilon = 0.02;
w0 = (2 * pi) / T0;
dUc = zeros(2,1) ;
dUc(1)= Uc(2);
dUc(2) = -(2 * epsilon * w0) * Uc(2) - w0 * w0 * Uc(1);
end

clear all;
T0 = 1;
epsilon = 0.02;
w0 = (2 * pi) / T0;
h = 0.04;
dt3 = (2 * (2 ^ 0.5) * h) / w0;
x0 = 0.01;
dx0 = 0;
t3 = (0 : dt3 : 100 * T0)';
np3 = size(t3,1);
x3 = zeros(np3,1);
dx3 = zeros(np3,1);
x3(1) = x0;
dx3(1) = dx0;
xj = [x0 ; dx0];
for inc = 2 : np3
    tc = t3(inc-1);
    xc = xj;
    k1 = cal_f2(xc);
    xc = xj + k1 * dt3 / 2;
    k2 = cal_f2(xc);
    xc = xj + k2 * dt3 / 2;
    k3 = cal_f2(xc);
    xc = xj + k3 * dt3;
    k4 = cal_f2(xc);
    dx = (k1 + 2 * k2 + 2 * k3 + k4) / 6;
    xj = xj + dx * dt3;
    x3(inc) = xj(1);
    dx3(inc) = xj(2);
end
plot(t3,x3,'m*','Linewidth',3)

```

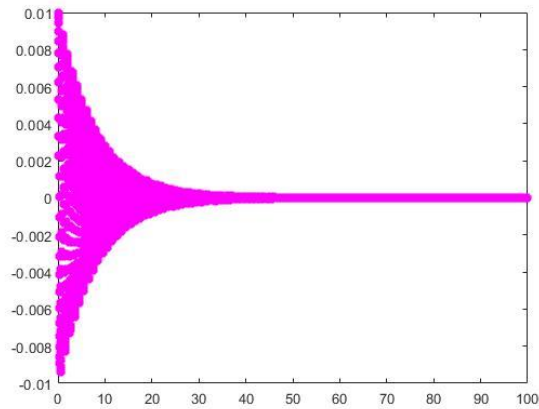


figure 26 La valeur de x fournies par Runge Kutta (h=0.04)

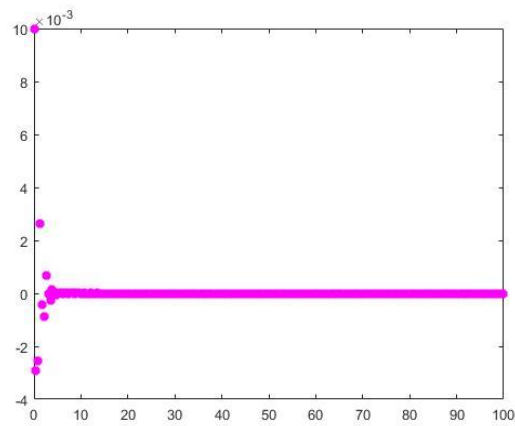


figure 27 La valeur de x fournies par Runge Kutta (h=0.96)

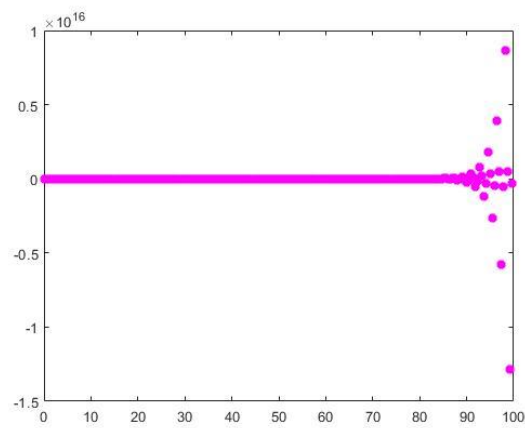


figure 28 La valeur de x fournies par Runge Kutta (h=1.04)

On peut trouver que quand $h < 1$, plus grand h , les résultats sont plus stables, mais moins précis ; mais quand $h > 1$, plus grand h , les résultats sont moins stables et moins précis.

Question 1.3.b

En utilisant la méthode de bisection, on peut déterminer le pas de temps critique est 0.5.

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

Question 1.1

De l'équation (1), on a :

$$a(2\ddot{\theta}_1 + \ddot{\theta}_2) + 2g\theta_1 = \frac{F_0}{m}\sin\omega t$$

$$a(\ddot{\theta}_1 + \ddot{\theta}_2) + g\theta_2 = \frac{F_0}{\sqrt{2}m}\sin\omega t$$

En utilisant (2) et (3), on a :

$$\begin{bmatrix} 1 & 0 & -\beta\Delta t^2 & 0 & 0 & 0 \\ 0 & 1 & -\gamma\Delta t & 0 & 0 & 0 \\ 2g & 0 & 2a & 0 & 0 & a \\ 0 & 0 & 0 & 1 & 0 & -\beta\Delta t^2 \\ 0 & 0 & 0 & 0 & 1 & -\gamma\Delta t \\ 0 & 0 & a & 0 & g & a \end{bmatrix} \begin{bmatrix} \theta_{1j+1} \\ \dot{\theta}_{1j+1} \\ \ddot{\theta}_{1j+1} \\ \theta_{2j+1} \\ \dot{\theta}_{2j+1} \\ \ddot{\theta}_{2j+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & \Delta t^2(0.5-\beta) & 0 & 0 & 0 \\ 0 & 1 & \Delta t(1-\gamma) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta t & \Delta t^2(0.5-\beta) \\ 0 & 0 & 0 & 0 & 1 & \Delta t(1-\gamma) \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta_{1j} \\ \dot{\theta}_{1j} \\ \ddot{\theta}_{1j} \\ \theta_{2j} \\ \dot{\theta}_{2j} \\ \ddot{\theta}_{2j} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{F_0}{m}\sin\omega t_{j+1} \\ 0 \\ 0 \\ \frac{F_0}{\sqrt{2}m}\sin\omega t_{j+1} \end{bmatrix}$$

On prend $\begin{bmatrix} 1 & 0 & -\beta\Delta t^2 & 0 & 0 & 0 \\ 0 & 1 & -\gamma\Delta t & 0 & 0 & 0 \\ 2g & 0 & 2a & 0 & 0 & a \\ 0 & 0 & 0 & 1 & 0 & -\beta\Delta t^2 \\ 0 & 0 & 0 & 0 & 1 & -\gamma\Delta t \\ 0 & 0 & a & 0 & g & a \end{bmatrix} = [B]$ et $\begin{bmatrix} 1 & \Delta t & \Delta t^2(0.5-\beta) & 0 & 0 & 0 \\ 0 & 1 & \Delta t(1-\gamma) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta t & \Delta t^2(0.5-\beta) \\ 0 & 0 & 0 & 0 & 1 & \Delta t(1-\gamma) \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} = [C]$

Alors la matrice d'amplification $[A] = [B]^{-1}[C]$

On peut le calculer en utilisant Matlab, les codes de MATLAB :

```
clear all;
dt1 = sym('dt1','real');
a = 0.5;
g = 9.81;
beta= 0;
gamma= 0.5;
B = [1, 0 , -beta * dt1 * dt1, 0, 0, 0;
     0, 1, -gamma * dt1, 0, 0, 0;
     2 * g, 0, 2 * a, 0, 0, a;
     0, 0, 0, 1, 0 , -beta * dt1 * dt1;
     0, 0, 0, 0, 1, -gamma * dt1;
     0, 0, a, 0, g, a];
C = [1, dt1, dt1 * dt1 * (0.5 - beta), 0, 0, 0;
     0, 1, dt1 * (1 - gamma), 0, 0, 0;
     0, 0, 0, 0, 0, 0;
     0, 0, 0, 1, dt1, dt1 * dt1 * (0.5 - beta);
     0, 0, 0, 0, 1, dt1 * (1 - gamma);
     0, 0, 0, 0, 0, 0];
A = inv(B) * C
```

Question 1.2

Les codes de MATLAB :

```
clear all;
for dt1 = 1 : -0.001 : 0
    a = 0.5;
    g = 9.81;
    beta= 0;
    gamma= 0.5;
    B = [1, 0 , -beta * dt1 * dt1, 0, 0, 0;
        0, 1, -gamma * dt1, 0, 0, 0;
        2 * g, 0, 2 * a, 0, 0, a;
        0, 0, 0, 1, 0 , -beta * dt1 * dt1;
        0, 0, 0, 0, 1, -gamma * dt1;
        0, 0, a, 0, g, a];
    C = [1, dt1, dt1 * dt1 * (0.5 - beta), 0, 0, 0;
        0, 1, dt1 * (1 - gamma), 0, 0, 0;
        0, 0, 0, 0, 0, 0;
        0, 0, 0, 1, dt1, dt1 * dt1 * (0.5 - beta);
        0, 0, 0, 0, 1, dt1 * (1 - gamma);
        0, 0, 0, 0, 0, 0];
    A = inv(B) * C;
    lamda = sym('lamda');
    d = det(A - lamda * eye(6));
    lamda = solve(d,lamda);
    lamda = double(lamda);
    if abs(lamda(1)) <= 1 && abs(lamda(2)) <= 1 && abs(lamda(3)) <= 1
    && abs(lamda(4)) <= 1 && abs(lamda(5)) <= 1 && abs(lamda(6)) <= 1
        break
    end
end
```

On peut trouver que le pas critique égal à 0.319s.

Question 1.3

On a :

$$a(2\ddot{\theta}_{10} + \ddot{\theta}_{20}) + 2g\theta_{10} = \frac{F_0}{m}\sin\omega t$$

$$a(\ddot{\theta}_{10} + \ddot{\theta}_{20}) + g\theta_{20} = \frac{F_0}{\sqrt{2}m}\sin\omega t$$

Donc on a :

$$\ddot{\theta}_{10} = \frac{1}{a}\left(\frac{F_0}{m}\sin\omega t - \frac{F_0}{\sqrt{2}m}\sin\omega t - 2g\theta_{10} + g\theta_{20}\right) = \frac{1}{a}(-2g\theta_{10} + g\theta_{20})$$

$$\ddot{\theta}_{20} = \frac{1}{a}\left(\frac{2F_0}{\sqrt{2}m}\sin\omega t - \frac{F_0}{m}\sin\omega t + 2g\theta_{10} - 2g\theta_{20}\right) = \frac{1}{a}(2g\theta_{10} - 2g\theta_{20})$$

Question 1.4

$$\begin{bmatrix} 1 & 0 & -\beta\Delta t^2 & 0 & 0 & 0 \\ 0 & 1 & -\gamma\Delta t & 0 & 0 & 0 \\ 2g & 0 & 2a & 0 & 0 & a \\ 0 & 0 & 0 & 1 & 0 & -\beta\Delta t^2 \\ 0 & 0 & 0 & 0 & 1 & -\gamma\Delta t \\ 0 & 0 & a & 0 & g & a \end{bmatrix} \begin{bmatrix} \theta_{1j+1} \\ \dot{\theta}_{1j+1} \\ \ddot{\theta}_{1j+1} \\ \theta_{2j+1} \\ \dot{\theta}_{2j+1} \\ \ddot{\theta}_{2j+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & \Delta t^2(0.5-\beta) & 0 & 0 & 0 \\ 0 & 1 & \Delta t(1-\gamma) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta t & \Delta t^2(0.5-\beta) \\ 0 & 0 & 0 & 0 & 1 & \Delta t(1-\gamma) \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta_{1j} \\ \dot{\theta}_{1j} \\ \ddot{\theta}_{1j} \\ \theta_{2j} \\ \dot{\theta}_{2j} \\ \ddot{\theta}_{2j} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{F_0}{m} \sin \omega t_{j+1} \\ 0 \\ 0 \\ \frac{F_0}{\sqrt{2}m} \sin \omega t_{j+1} \end{bmatrix}$$

Question 1.5

Les codes de MATLAB :

```
clear all;
dt1 = 0.02;
T0 = 8;
t1 = (0 : dt1 : T0)';
np1 = size(t1, 1);
m = 2;
a = 0.5;
g = 9.81;
beta= 0;
gamma= 0.5;
F0 = 20;
w = 2 * pi;
theta10 = 0;
theta20 = 0;
dtheta10 = -1.31519275;
dtheta20 = -1.85996342;
ddtheta10 = (- 2 * g * theta10 + g * theta20) / a;
ddtheta20 = (2 * g * theta10 - 2 * g * theta20) / a;
q = [theta10; dtheta10; ddtheta10; theta20; dtheta20; ddtheta20];
q1b = zeros(np1, 1);
dq1b = zeros(np1, 1);
ddq1b = zeros(np1, 1);
q2b = zeros(np1, 1);
dq2b = zeros(np1, 1);
ddq2b = zeros(np1, 1);
q1b(1) = theta10;
dq1b(1) = dtheta10;
ddq1b(1) = ddtheta10;
q2b(1) = theta20;
dq2b(1) = dtheta20;
ddq2b(1) = ddtheta20;
B = [1, 0 , -beta * dt1 * dt1, 0, 0, 0;
      0, 1, -gamma * dt1, 0, 0, 0;
      2 * g, 0, 2 * a, 0, 0, a;
      0, 0, 0, 1, 0 , -beta * dt1 * dt1;
      0, 0, 0, 0, 1, -gamma * dt1;
```

```

0, 0, a, 0, g, a];
C = [1, dt1, dt1 * dt1 * (0.5 - beta), 0, 0, 0;
0, 1, dt1 * (1 - gamma), 0, 0, 0;
0, 0, 0, 0, 0, 0;
0, 0, 0, 1, dt1, dt1 * dt1 * (0.5 - beta);
0, 0, 0, 0, 1, dt1 * (1 - gamma);
0, 0, 0, 0, 0, 0];
A = inv(B) * C;
for inc = 2 : np1
    q = A * q + inv(B) * [0; 0; F0 * sin(w * t1(inc))/ m; 0; 0; F0 *
sin(w * t1(inc))/ ((2 ^ 0.5) * m)];
    q1b(inc) = q(1);
    dq1b(inc) = q(2);
    ddq1b(inc) = q(3);
    q2b(inc) = q(4);
    dq2b(inc) = q(5);
    ddq2b(inc) = q(6);
end;
plot(t1,q1b,'b-','Linewidth',3);
hold on;
plot(t1,q2b,'r+','Linewidth',3);

```

Question 1.6

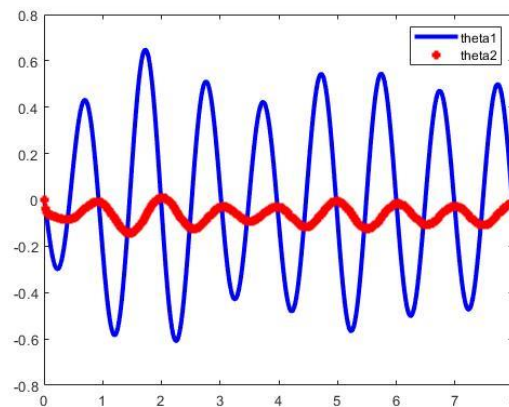


figure 29 La valeur de q fournies par Newmark explicite ($\Delta t = 0.02s$)

On a $q(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $q(\Delta t) = \begin{bmatrix} -0.0263 \\ -0.0372 \end{bmatrix}$, $q(2\Delta t) = \begin{bmatrix} -0.0624 \\ -0.0534 \end{bmatrix}$, $q(0.5s) = \begin{bmatrix} 0.145 \\ -0.0794 \end{bmatrix}$;

$\dot{q}(0) = \begin{bmatrix} -1.13 \\ -1.86 \end{bmatrix}$, $\dot{q}(\Delta t) = \begin{bmatrix} -1.56 \\ -1.34 \end{bmatrix}$, $\dot{q}(2\Delta t) = \begin{bmatrix} -1.88 \\ -0.583 \end{bmatrix}$, $\dot{q}(0.5s) = \begin{bmatrix} 2.39 \\ 0.121 \end{bmatrix}$;

$\ddot{q}(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $\ddot{q}(\Delta t) = \begin{bmatrix} -24.4 \\ 52.4 \end{bmatrix}$, $\ddot{q}(2\Delta t) = \begin{bmatrix} -7.59 \\ 22.6 \end{bmatrix}$, $\ddot{q}(0.5s) = \begin{bmatrix} -3.49 \\ 1.12 \end{bmatrix}$

Question 2.1

On a :

$$a \left[2 \left(\ddot{\theta}_{1j+1}^* + \Delta \ddot{\theta}_{1j+1} \right) + \left(\ddot{\theta}_{2j+1}^* + \Delta \ddot{\theta}_{2j+1} \right) \right] + 2g \left(\theta_{1j+1}^* + \beta \Delta t^2 \Delta \ddot{\theta}_{1j+1} \right) = \frac{F_0}{m} \sin \omega t_{j+1}$$

$$a \left[\left(\ddot{\theta}_{1j+1}^* + \Delta \ddot{\theta}_{1j+1} \right) + \left(\ddot{\theta}_{2j+1}^* + \Delta \ddot{\theta}_{2j+1} \right) \right] + g \left(\theta_{2j+1}^* + \beta \Delta t^2 \Delta \ddot{\theta}_{2j+1} \right) = \frac{F_0}{\sqrt{2}m} \sin \omega t_{j+1}$$

Car on a :

$$\Delta \theta_{1j+1} = \beta \Delta t^2 \Delta \ddot{\theta}_{1j+1}$$

$$\Delta \theta_{2j+1} = \beta \Delta t^2 \Delta \ddot{\theta}_{2j+1}$$

$$\Delta \dot{\theta}_{1j+1} = \gamma \Delta t \Delta \ddot{\theta}_{1j+1}$$

$$\Delta \dot{\theta}_{2j+1} = \gamma \Delta t \Delta \ddot{\theta}_{2j+1}$$

Donc on a :

$$\begin{bmatrix} 2a + 2g\beta\Delta t^2 & a \\ a & a + g\beta\Delta t^2 \end{bmatrix} \begin{bmatrix} \Delta \ddot{\theta}_{1j+1} \\ \Delta \ddot{\theta}_{2j+1} \end{bmatrix} = \begin{bmatrix} \frac{F_0}{m} \sin \omega t_{j+1} - 2a\ddot{\theta}_{1j+1}^* - a\ddot{\theta}_{2j+1}^* - 2g\theta_{1j+1}^* \\ \frac{F_0}{\sqrt{2}m} \sin \omega t_{j+1} - a\ddot{\theta}_{1j+1}^* - a\ddot{\theta}_{2j+1}^* - g\theta_{2j+1}^* \end{bmatrix}$$

$$\begin{bmatrix} 2a + 2g\beta\Delta t^2 & a \\ a & a + g\beta\Delta t^2 \end{bmatrix} \begin{bmatrix} \Delta \ddot{\theta}_{1j+1} \\ \Delta \ddot{\theta}_{2j+1} \end{bmatrix} = \begin{bmatrix} -2a & -a \\ -a & -a \end{bmatrix} \begin{bmatrix} \ddot{\theta}_{1j+1}^* \\ \ddot{\theta}_{2j+1}^* \end{bmatrix} + \begin{bmatrix} \frac{F_0}{m} \sin \omega t_{j+1} - 2g\theta_{1j+1}^* \\ \frac{F_0}{\sqrt{2}m} \sin \omega t_{j+1} - g\theta_{2j+1}^* \end{bmatrix}$$

On prend $\begin{bmatrix} 2a + 2g\beta\Delta t^2 & a \\ a & a + g\beta\Delta t^2 \end{bmatrix} = [B]$ et $\begin{bmatrix} -2a & -a \\ -a & -a \end{bmatrix} = [C]$

Alors la matrice d'amplification $[A] = [B]^{-1}[C]$

Question 2.2

Les codes de MATLAB :

```
clear all;
dt1 = (0 : 0.01 : 1)';
np1 = size(dt1, 1);
m = 2;
a = 0.5;
g = 9.81;
beta= 0.25;
gamma= 0.5;
F0 = 20;
w = 2 * pi;
for inc = 1 : np1
    B = [2 * a + 2 * g * beta * dt1(inc) * dt1(inc), a;
    a, a + g * beta * dt1(inc) * dt1(inc)];
    C = [-2 * a, -a;
    -a, -a];
```

```

A = inv(B) * C;
[Z, lamda] = eig(A);
lamdab(inc) = max(lamda(1), lamda(4));
end;
plot(dt1, lamdab, 'b-', 'Linewidth', 3);

```

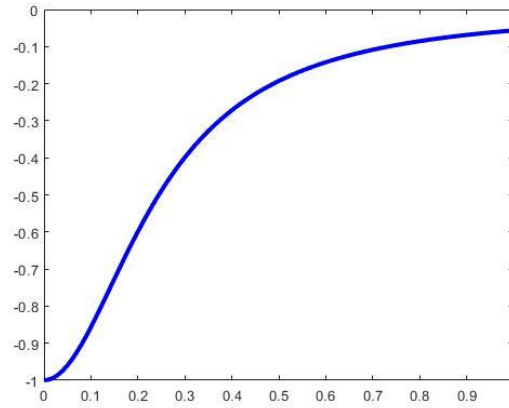


figure 30 La plus grande valeur propre de la matrice d'amplification

On peut trouver que la plus grande valeur propre de la matrice d'amplification augmente quand le pas de temps augmente.

Question 2.3

On a :

$$\ddot{\theta}_{1_0} = \frac{1}{a} \left(\frac{F_0}{m} \sin \omega t - \frac{F_0}{\sqrt{2}m} \sin \omega t - 2g\theta_{1_0} + g\theta_{2_0} \right) = \frac{1}{a} (-2g\theta_{1_0} + g\theta_{2_0})$$

$$\ddot{\theta}_{2_0} = \frac{1}{a} \left(\frac{2F_0}{\sqrt{2}m} \sin \omega t - \frac{F_0}{m} \sin \omega t + 2g\theta_{1_0} - 2g\theta_{2_0} \right) = \frac{1}{a} (2g\theta_{1_0} - 2g\theta_{2_0})$$

Question 2.4

On a :

$$\theta_{1_{j+1}}^* = \theta_{1_{j+1}}^* + \Delta \theta_{1_{j+1}}$$

$$\dot{\theta}_{1_{j+1}}^* = \dot{\theta}_{1_{j+1}}^* + \Delta \dot{\theta}_{1_{j+1}}$$

$$\ddot{\theta}_{1_{j+1}}^* = \ddot{\theta}_{1_{j+1}}^* + \Delta \ddot{\theta}_{1_{j+1}}$$

$$\theta_{2_{j+1}}^* = \theta_{2_{j+1}}^* + \Delta \theta_{2_{j+1}}$$

$$\dot{\theta}_{2_{j+1}}^* = \dot{\theta}_{2_{j+1}}^* + \Delta \dot{\theta}_{2_{j+1}}$$

$$\ddot{\theta}_{2_{j+1}}^* = \ddot{\theta}_{2_{j+1}}^* + \Delta \ddot{\theta}_{2_{j+1}}$$

Donc

$$\theta_{1_{j+1}}^* = \theta_{1_{j+1}}^* + \beta \Delta t^2 \Delta \ddot{\theta}_{1_{j+1}}$$

$$\dot{\theta}_{1_{j+1}}^* = \dot{\theta}_{1_{j+1}}^* + \gamma \Delta t \Delta \ddot{\theta}_{1_{j+1}}$$

$$\ddot{\theta}_{1j+1}^* = \ddot{\theta}_{1j+1}^* + \Delta \ddot{\theta}_{1j+1}$$

$$\theta_{2j+1}^* = \theta_{2j+1}^* + \beta \Delta t^2 \Delta \ddot{\theta}_{2j+1}$$

$$\dot{\theta}_{2j+1}^* = \dot{\theta}_{2j+1}^* + \gamma \Delta t \Delta \ddot{\theta}_{2j+1}$$

$$\ddot{\theta}_{2j+1}^* = \ddot{\theta}_{2j+1}^* + \Delta \ddot{\theta}_{2j+1}$$

Question 2.5

Les codes de MATLAB :

```
clear all;
dt2 = 0.02;
T0 = 8;
t2 = (0 : dt2 : T0)';
np2 = size(t2, 1);
m = 2;
a = 0.5;
g = 9.81;
beta = 0.25;
gamma = 0.5;
F0 = 20;
w = 2 * pi;
B = [2 * a + 2 * g * beta * dt2 * dt2, a;
      a, a + g * beta * dt2 * dt2];
C = [-2 * a, -a;
      -a, -a];
A = inv(B) * C;
theta10 = 0;
theta20 = 0;
dtheta10 = -1.31519275;
dtheta20 = -1.85996342;
ddtheta10 = (- 2 * g * theta10 + g * theta20) / a;
ddtheta20 = (2 * g * theta10 - 2 * g * theta20) / a;
q = [theta10; dtheta10; ddtheta10; theta20; dtheta20; ddtheta20];
q1b = zeros(np2, 1);
dq1b = zeros(np2, 1);
ddq1b = zeros(np2, 1);
q2b = zeros(np2, 1);
dq2b = zeros(np2, 1);
ddq2b = zeros(np2, 1);
q1b(1) = theta10;
dq1b(1) = dtheta10;
ddq1b(1) = ddtheta10;
q2b(1) = theta20;
```

```

dq2b(1) = dtheta20;
ddq2b(1) = ddtheta20;
deltatheta = zeros(2, 1);
for inc = 2 : np2
    q(1) = q(1) + dt2 * q(2) + dt2 * dt2 * (0.5 -beta) * q(3);
    q(2) = q(2) + dt2 * (1 - gamma) * q(3);
    q(3) = 0;
    q(4) = q(4) + dt2 * q(5) + dt2 * dt2 * (0.5 -beta) * q(6);
    q(5) = q(5) + dt2 * (1 - gamma) * q(6);
    q(6) = 0;
    while abs(a * (2 * q(3) + q(6)) + 2 * g * q(1) - ((F0 * sin(w *
t2(inc))) / m)) >= 0.01 || abs(a * (q(3) + q(6)) + g * q(4) - ((F0 *
sin(w * t2(inc))) / ((2 ^ 0.5) * m))) >= 0.01
        deltatheta = A * [q(3); q(6)] + (inv(B) * [(((F0 * sin(w *
t2(inc))) / m) - 2 * g * q(1)); ((F0 * sin(w * t2(inc))) / ((2 ^ 0.5)
* m) - g * q(4))]);
        q = q + [beta * dt2 * dt2 * deltatheta(1); gamma * dt2 *
deltatheta(1); deltatheta(1);
                beta * dt2 * dt2 * deltatheta(2); gamma * dt2 *
deltatheta(2); deltatheta(2)];
    end
    q1b(inc) = q(1);
    dq1b(inc) = q(2);
    ddq1b(inc) = q(3);
    q2b(inc) = q(4);
    dq2b(inc) = q(5);
    ddq2b(inc) = q(6);
end
plot(t2,q1b,'b-','Linewidth',3);
hold on;
plot(t2,q2b,'r+','Linewidth',3);

```

Question 2.6

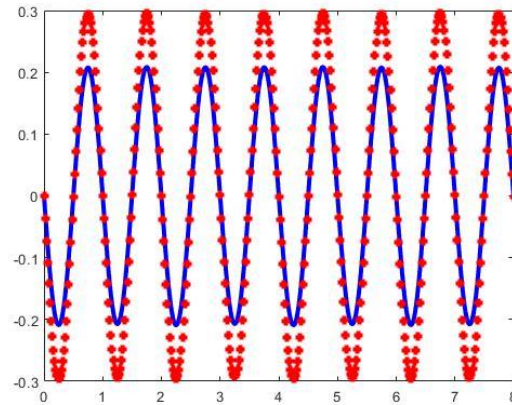


figure 31 La valeur de q fournies par Newmark implicite ($\Delta t = 0.02s$)

On a $q(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $q(\Delta t) = \begin{bmatrix} -0.0262 \\ -0.0371 \end{bmatrix}$, $q(2\Delta t) = \begin{bmatrix} -0.0520 \\ -0.0735 \end{bmatrix}$, $q(0.5s) = \begin{bmatrix} 9.22e-4 \\ -0.00130 \end{bmatrix}$;

$\dot{q}(0) = \begin{bmatrix} -1.32 \\ -1.86 \end{bmatrix}$, $\dot{q}(\Delta t) = \begin{bmatrix} -1.30 \\ -1.85 \end{bmatrix}$, $\dot{q}(2\Delta t) = \begin{bmatrix} -1.27 \\ -1.80 \end{bmatrix}$, $\dot{q}(0.5s) = \begin{bmatrix} 1.31 \\ 1.86 \end{bmatrix}$;

$\ddot{q}(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $\ddot{q}(\Delta t) = \begin{bmatrix} 1.04 \\ 1.46 \end{bmatrix}$, $\ddot{q}(2\Delta t) = \begin{bmatrix} 2.05 \\ 2.91 \end{bmatrix}$, $\ddot{q}(0.5s) = \begin{bmatrix} 0.0106 \\ 0.0150 \end{bmatrix}$

Oscillateur non linéaire à un degré de liberté

Question 1.1

$$\begin{bmatrix} 1 & 0 & -\beta\Delta t^2 \\ 0 & 1 & -\gamma\Delta t \\ -\omega_0^2(1 + q_{j+1}^2) & 0 & 1 \end{bmatrix} \begin{bmatrix} q_{j+1} \\ \dot{q}_{j+1} \\ \ddot{q}_{j+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & \Delta t^2(0.5 - \beta) \\ 0 & 1 & \Delta t(1 - \gamma) \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} q_j \\ \dot{q}_j \\ \ddot{q}_j \end{bmatrix}$$

Quand $\gamma = 0.5, \beta = 0$ on a :

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -0.5\Delta t \\ -\omega_0^2(1 + q_{j+1}^2) & 0 & 1 \end{bmatrix} \begin{bmatrix} q_{j+1} \\ \dot{q}_{j+1} \\ \ddot{q}_{j+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & 0.5\Delta t^2 \\ 0 & 1 & 0.5\Delta t \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} q_j \\ \dot{q}_j \\ \ddot{q}_j \end{bmatrix}$$

Question 1.2

Les codes de MATLAB :

```
clear all;
T0 = 6;
a = 0.1;
w0 = 2 * pi;
y0 = 2;
dt1 = 0.02;
q0 = 2;
dq0 = 0;
```

```

t1 = (0 : dt1 : T0)';
np1 = size(t1,1);
q1 = zeros(np1,1);
dq1 = zeros(np1,1);
q1(1) = q0;
dq1(1) = dq0;
ddq0c = - w0 * w0 * q0 * (1 + a * q0 * q0);
for inc = 2 : np1
    q1(inc) = q1(inc - 1) + dt1 * dq1(inc - 1) + dt1 * dt1 * 0.5 *
ddq0c ;
    ddqc = - w0 * w0 * q1(inc) * (1 + a * q1(inc) * q1(inc));
    dq1(inc) = dq1(inc - 1) + 0.5 * dt1 * (ddq0c + ddqc);
    ddq0c = ddqc;
end
plot(t1,q1,'b-', 'Linewidth',3)

```

Question 1.3

Quand $\Delta t = 0.02s$, on a :

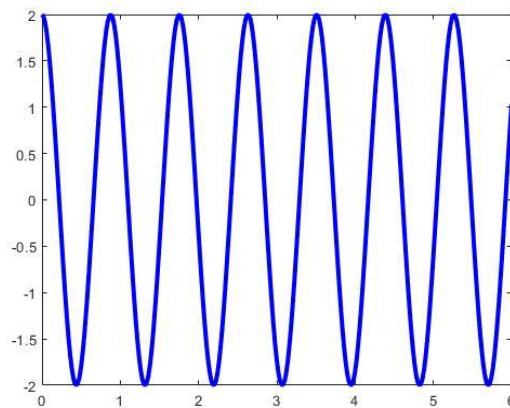


figure 32 La valeur de q fournies par Newmark explicite ($\Delta t = 0.02s$)

On a $q(0) = 2$, $q(\Delta t) = 1.9779$, $q(2\Delta t) = 1.9123$, $q(T_0) = 1.0329$.

Question 2.1

On cherche à minimiser $|\ddot{q}_{j+1} + \omega_0^2 q_{j+1}(1 + q_{j+1}^2)|$ à chaque instant.

Question 2.2

$$\ddot{q}_{j+1}^* = \ddot{q}_{j+1} + \Delta \ddot{q}_{j+1}$$

Avec

$$\ddot{q}_{j+1}^* + \Delta \ddot{q}_{j+1} + \omega_0^2 (q_{j+1}^* + \beta \Delta t^2 \Delta \ddot{q}_{j+1})(1 + (q_{j+1}^* + \beta \Delta t^2 \Delta \ddot{q}_{j+1})^2) = 0$$

Question 2.3

Les codes de MATLAB :

```

clear all;
T0 = 6;
a = 0.1;
w0 = 2 * pi;
y0 = 2;
dt2 = 0.02;
q0 = 2;
dq0 = 0;
t2 = (0 : dt2 : T0)';
np2 = size(t2,1);
q2 = zeros(np2,1);
dq2 = zeros(np2,1);
ddq2 = zeros(np2,1);
q2(1) = q0;
dq2(1) = dq0;
ddq2(1) = - w0 * q0 * (1 + a * q0 * q0);
for inc = 2 : np2
    q2b = q2(inc - 1) + dt2 * dq2(inc - 1) + 0.25 * dt2 * dt2 *
ddq2(inc - 1);
    dq2b = dq2(inc - 1) + 0.5 * dt2 * ddq2(inc - 1);
    ddq2b = 0;
    while abs(ddq2b + w0 * w0 * q2b * (1 + a * q2b * q2b)) >= 0.01
        syms deltaq2;
        eq = ddq2b + deltaq2 + w0 * w0 * (q2b + 0.25 * dt2 * dt2 *
deltaq2) * (1 + a * (q2b + 0.25 * dt2 * dt2 * deltaq2) * (q2b + 0.25
* dt2 * dt2 * deltaq2));
        deltaq2 = solve(eq,deltaq2);
        deltaq2 = double(deltaq2);
        q2b = q2b + 0.25 * dt2 * dt2 * deltaq2(1);
        dq2b = dq2b + 0.5 * dt2 * deltaq2(1);
        ddq2b = ddq2b + deltaq2(1);
    end
    q2(inc) = q2b;
    dq2(inc) = dq2b;
    ddq2(inc) = ddq2b;
end
plot(t2,q2,'r+', 'Linewidth',3)

```

Question 2.4

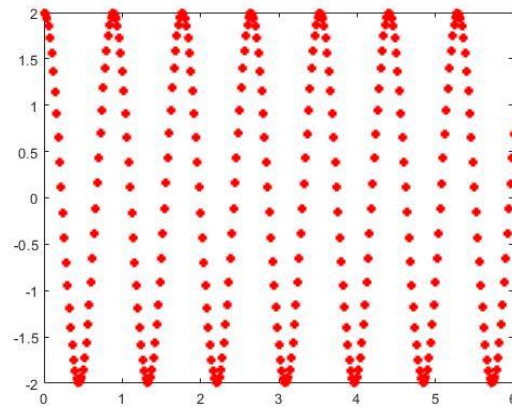


figure 33 La valeur de q fournies par Newmark implicite ($\Delta t = 0.02s$)

On a $q(0) = 2$, $q(\Delta t) = 1.9873$, $q(2\Delta t) = 1.9404$, $q(T_0) = 0.6922$.

Oscillateur non linéaire à un degré de liberté

Question 1.1.a

On a

$$\ddot{q} - \varepsilon \omega_0 \left[1 - \left(\frac{q}{y_0} \right)^2 \right] \dot{q} + \omega_0^2 q = 0$$

On peut le transformer en :

$$dU(1) = U(2)$$

$$dU(2) = \varepsilon \omega_0 \left[1 - \left(\frac{U(1)}{y_0} \right)^2 \right] U(2) - \omega_0^2 U(1)$$

Question 1.1.b

Le vecteur d'état est :

$$\{U\} = \begin{bmatrix} q(t) \\ \dot{q}(t) \end{bmatrix}$$

Question 1.2

$$\begin{bmatrix} q_{j+1} \\ \dot{q}_{j+1} \end{bmatrix} = \begin{bmatrix} q_j \\ \dot{q}_j \end{bmatrix} + K \Delta t$$

Question 1.3

Les codes de MATLAB :

```
function [ dUc ] = cal_f1(Uc)
epsilon = 4;
w0 = 2 * pi;
y0 = 2;
dUc = zeros(2,1) ;
dUc(1) = Uc(2);
```

```

dUc(2) = epsilon * w0 * (1 - ((Uc(1) / y0)^2)) * Uc(2) - w0 * w0 *
Uc(1);
end

clear all;
T0 = 20;
epsilon = 4;
w0 = 2 * pi;
y0 = 2;
dt1 = 0.025;
q0 = 0.1;
dq0 = 0;
t1 = (0 : dt1 : T0)';
np1 = size(t1,1);
q1 = zeros(np1,1);
dq1 = zeros(np1,1);
q1(1) = q0;
dq1(1) = dq0;
qj = [q0 ; dq0];
for inc = 2 : np1
    tc = t1(inc-1);
    qc = qj;
    k1 = cal_f1(qc);
    qc = qj + k1 * dt1 / 2;
    k2 = cal_f1(qc);
    qc = qj + k2 * dt1 / 2;
    k3 = cal_f1(qc);
    qc = qj + k3 * dt1;
    k4 = cal_f1(qc);
    dx = (k1 + 2 * k2 + 2 * k3 + k4) / 6;
    qj = qj + dx * dt1;
    q1(inc) = qj(1);
    dq1(inc) = qj(2);
end
plot(t1,q1,'b-','Linewidth',3)

```

Question 1.4

Quand $\varepsilon = 4$ et $\Delta t = 0.025s$, on a :

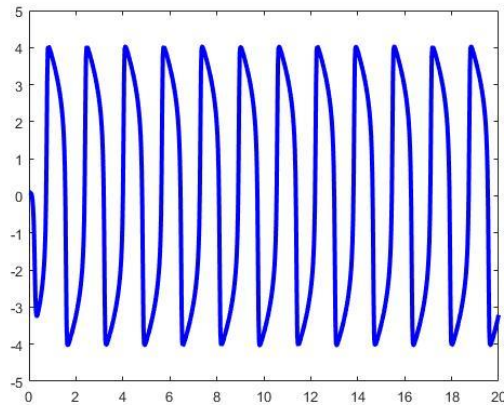


figure 34 La valeur de q fournies par Runge Kutta ($\epsilon = 4$ et $\Delta t = 0.025s$)

On a $q(0) = 0.1$, $q(\Delta t) = 0.0985$, $q(2\Delta t) = 0.0922$, $q(5\Delta t) = -0.0115$.

Question 2.1

Quand $\gamma = 0.5, \beta = 0$ on a :

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -0.5\Delta t \\ \omega_0^2 & -\epsilon\omega_0 \left[1 - \left(\frac{q_{j+1}}{y_0}\right)^2\right] & 1 \end{bmatrix} \begin{bmatrix} q_{j+1} \\ \dot{q}_{j+1} \\ \ddot{q}_{j+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & 0.5\Delta t^2 \\ 0 & 1 & 0.5\Delta t \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} q_j \\ \dot{q}_j \\ \ddot{q}_j \end{bmatrix}$$

Avec cette matrice on peut trouver aussi que :

$$\ddot{q}_{j+1} = \frac{\epsilon\omega_0 \left[1 - \left(\frac{q_{j+1}}{y_0}\right)^2\right] (\dot{q}_j + 0.5\Delta t)\ddot{q}_j - \omega_0^2 q_{j+1}}{\left(1 - 0.5\epsilon\omega_0\Delta t \left[1 - \left(\frac{q_{j+1}}{y_0}\right)^2\right]\right)}$$

Question 2.2

Les codes de MATLAB :

```
clear all;
T0 = 20;
epsilon = 4;
w0 = 2 * pi;
y0 = 2;
dt2 = 0.025;
q0 = 0.1;
dq0 = 0;
t2 = (0 : dt2 : T0)';
np2 = size(t2,1);
q2 = zeros(np2,1);
dq2 = zeros(np2,1);
q2(1) = q0;
dq2(1) = dq0;
ddq0c = epsilon * w0 * (1 - ((q0 / y0)^2)) * dq0 - w0 * w0 * q0;
for inc = 2 : np2
```

```

q2(inc) = q2(inc - 1) + dt2 * dq2(inc - 1) + dt2 * dt2 * 0.5 *
ddq0c ;
ddqc = (epsilon * w0 * (1 - ((q2(inc-1) / y0)^2)) * (dq2(inc - 1) +
dt2 * 0.5 * ddq0c) - w0 * w0 * q2(inc - 1)) / (1 - 0.5 * dt2 *
epsilon * w0 * (1 - ((q2(inc-1) / y0)^2)));
dq2(inc) = dq2(inc - 1) + 0.5 * dt2 * (ddq0c + ddqc);
ddq0c = ddqc;
end
plot(t2,q2,'r+', 'Linewidth',3)

```

Question 2.2

Quand $\varepsilon = 4$ et $\Delta t = 0.025s$, on a :

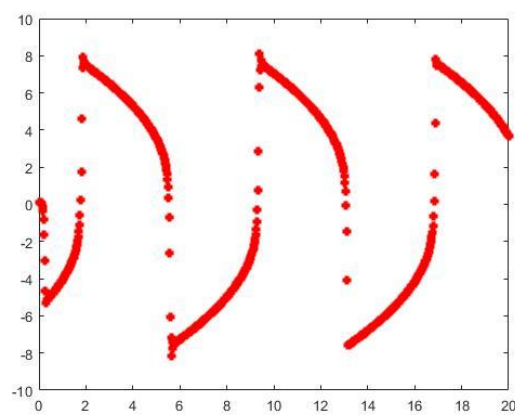


figure 35 La valeur de q fournies par Newmark explicite ($\varepsilon = 4$ et $\Delta t = 0.025s$)

On a $q(0) = 0.1$, $q(\Delta t) = 0.0988$, $q(2\Delta t) = 0.0928$, $q(5\Delta t) = -0.0179$.

Question 3.1.1

Quand $\varepsilon = 0.1$ et $\Delta t = 0.01s$, en utilisant Runge Kutta on a :

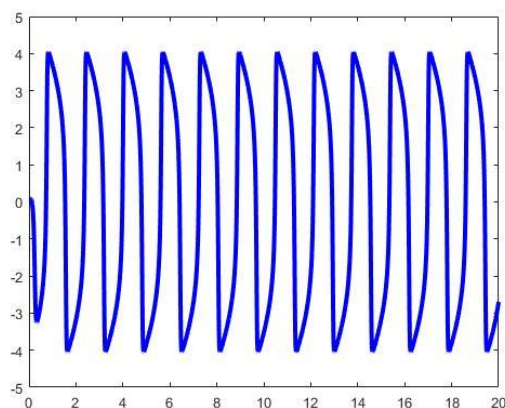


figure 36 La valeur de q fournies par Runge Kutta ($\varepsilon = 0.1$ et $\Delta t = 0.01s$)

Quand $\varepsilon = 0.1$ et $\Delta t = 0.02s$, en utilisant Runge Kutta on a :

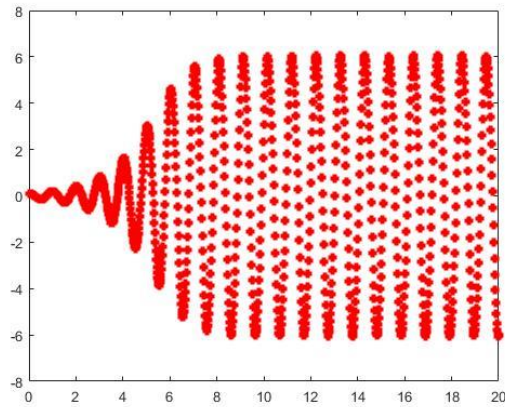


figure 37 La valeur de q fournies par Newmark explicite ($\varepsilon = 0.1$ et $\Delta t = 0.02s$)

Question 3.1.2

Pour ce cas de calcul, on peut trouver que la convergence de Runge Kutta est mieux que Newmark explicite.

Question 3.1.3

Les graphes des (t, q) est dans la question 3.1.1, les graphes des $\left(q, \frac{\dot{q}}{w_0}\right)$ est :

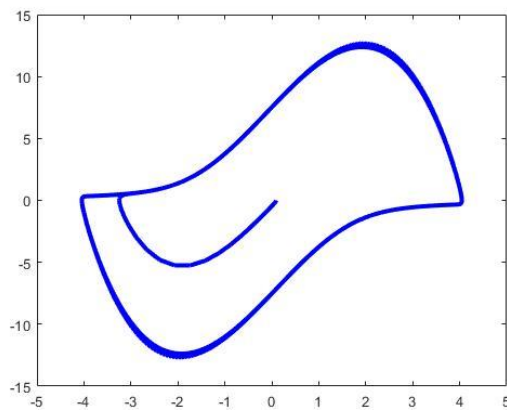


figure 38 $q, \frac{\dot{q}}{w_0}$ fournies par Runge Kutta ($\varepsilon=0.1$ et $\Delta t=0.01s$)

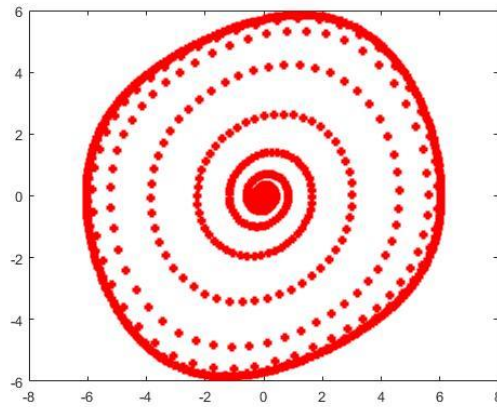


figure 39 $q, \frac{\dot{q}}{w_0}$ fournies par Newmark explicite ($\varepsilon=0.1$ et $\Delta t=0.01s$)

Question 3.2.1

Quand $\varepsilon = 5$ et $\Delta t = 0.01s$, en utilisant Runge Kutta on a :

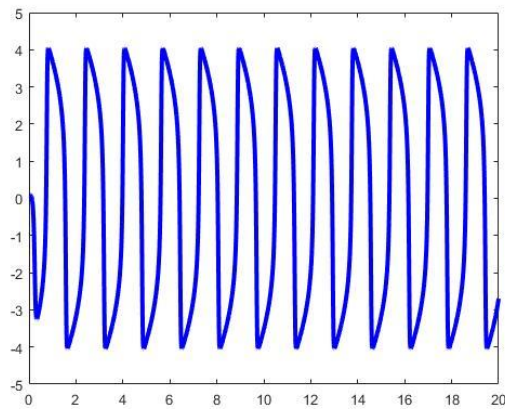


figure 40 La valeur de q fournies par Runge Kutta ($\varepsilon = 5$ et $\Delta t = 0.01s$)

Quand $\varepsilon = 5$ et $\Delta t = 0.02s$, en utilisant Runge Kutta on a :

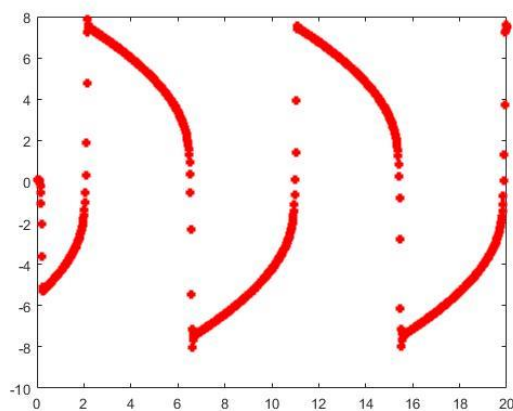


figure 41 La valeur de q fournies par Newmark explicite ($\varepsilon = 5$ et $\Delta t = 0.02s$)

Question 3.2.2

Pour Newmark explicite non, parce qu'il y a différence entre la solution correcte et la solution qu'on calculé.

Question 3.2.3

Pour Newmark explicite, on peut trouver que le pas de temps doit être $\Delta t = 0.001s$.

Question 3.2.4

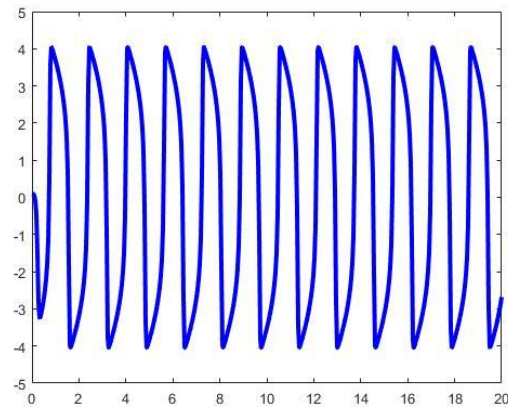


figure 42 La valeur de q fournies par Runge Kutta ($\epsilon = 5$ et $\Delta t = 0.001s$)

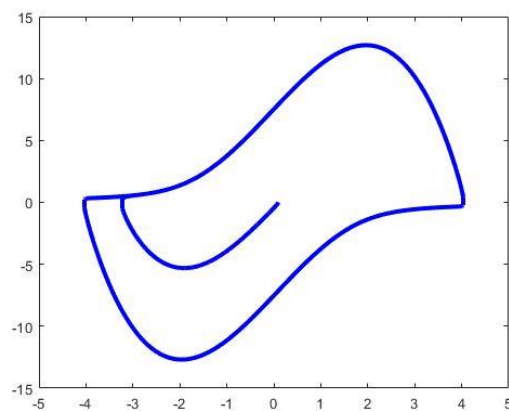


figure 43 $q, \frac{\dot{q}}{w_0}$ fournies par Runge Kutta ($\epsilon=5$ et $\Delta t=0.001s$)

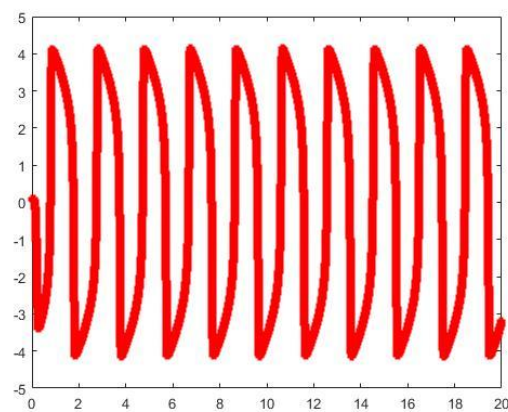


figure 44 La valeur de q fournies par Newmark explicite ($\epsilon = 5$ et $\Delta t = 0.001s$)

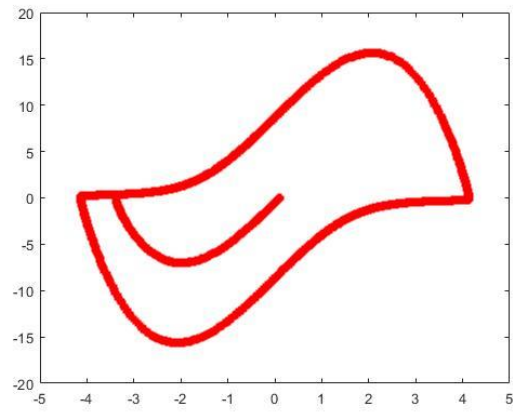


figure 45 $q, \frac{\dot{q}}{w_0}$ fournies par Newmark explicite ($\varepsilon=5$ et $\Delta t=0.001s$)