

Retrouver l'équation du mouvement du pendule simple

Le process de démonstration est comme ci-dessous:

Pour un pendule simple, on a :

$$\begin{cases} E_c = \frac{I}{2} \dot{\theta}^2 \\ E_p = -mgd \cos \theta + \text{cte} \\ \delta W = 0 \end{cases}$$

avec l'équation de Lagrange $\frac{d}{dt} \left[\frac{\partial L}{\partial \dot{x}_i} \right] - \frac{\partial L}{\partial x_i} = Q_i$

$$L = E_c - E_p = \frac{I}{2} \dot{\theta}^2 + mgd \cos \theta - \text{cte}$$

$$\delta W = \sum_{i=1}^N Q_i \delta x_i = 0 \Rightarrow Q_i = 0$$

$$\frac{\partial L}{\partial \theta} = -mgd \sin \theta \quad \frac{d}{dt} \left[\frac{\partial L}{\partial \dot{\theta}} \right] = I \ddot{\theta}$$

$$\text{Donc} \quad I \ddot{\theta} + mgd \sin \theta = 0$$

c'est l'équation du mouvement.

Oscillateur conservative linéaire à un degré de liberté

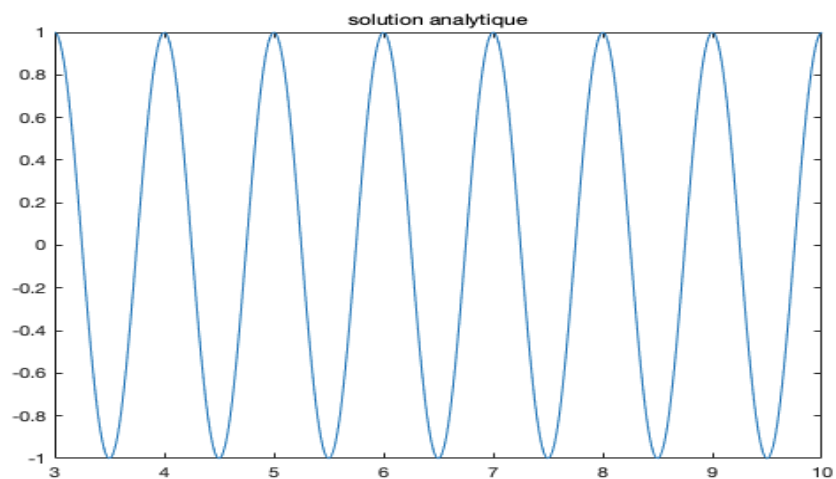
1.1

```
clear all;

%déterminer la solution de l'équation
w0 = 2*pi;
q0 = 1;
Dq0 = 0;
T0 = 3;
syms t
eq = 'D2q+w0^2*q=0';
q(t) = simplify(dsolve(eq, 'q(0)=1', 'Dq(0)=0'))

T = linspace(0, T0, 500);
q = exp(-T*w0*i)/2 + exp(T*w0*i)/2;
plot(T, q);
title('solution analytique');
```

```
q(t) =
cos(t*w0)
```



1.2

```
clf;
dq = diff(q,t)
%calculer numériquement E*
Etoile = 1/2*(dq^2+w0^2*q^2)
```

```
dq(t) =
-w0*sin(t*w0)
```

```
Etoile(t) =
(2778046668940015*cos(t*w0)^2)/140737488355328 + (w0^2*sin(t*w0)^2)/2
```

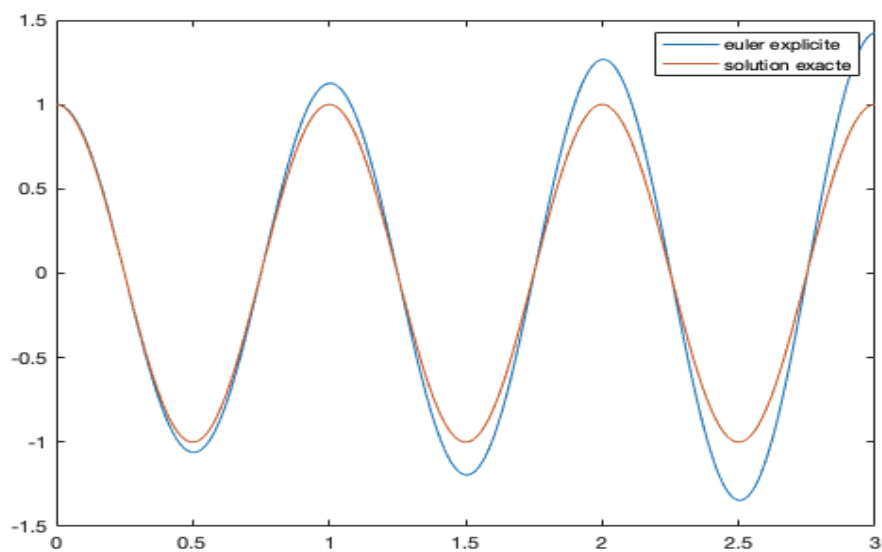
2.1 Euler explicite

$$\begin{cases} q_{j+1} \\ \dot{q}_{j+1} \end{cases} = \begin{cases} q_j + \Delta t \times \dot{q}_j \\ \dot{q}_j \end{cases}, \text{ et } \ddot{q}_j = -\omega_0^2 q_j$$

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

2.2

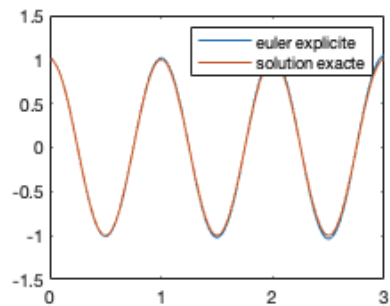
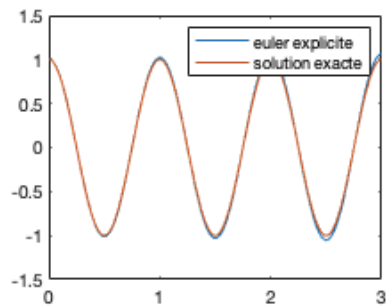
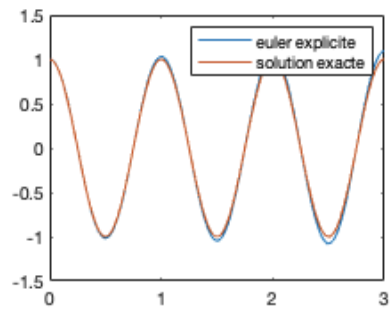
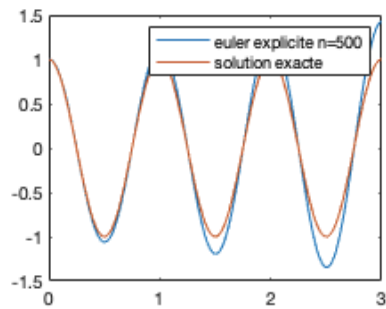
```
clf;
n = 500;
dt = T0/n;
t = 0:dt:T0;
A = [1, dt; -w0^2*dt, 1];
U1(:, 1) = [q0; Dq0];
for j = 1:length(t)-1
    U1(:, j+1) = A*U1(:, j);
end
plot(t, U1(1, :))
hold on;
plot(t, cos(2*pi*t));
legend('euler explicite', 'solution exacte')
```



2.3

```
clf;
subplot(2,2,1);
plot(t,U1(1,:));
hold on;
plot(t,cos(2*pi*t));
legend('euler explicite n=500','solution exacte')

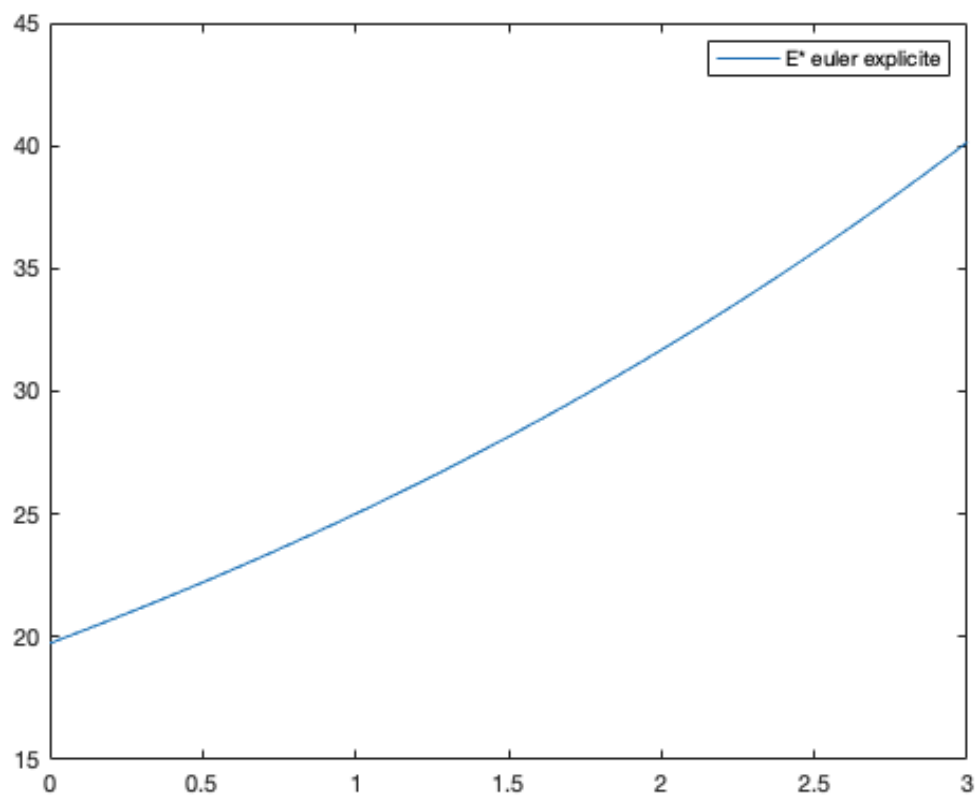
%on change le pas de temps
for i=2:4
    n = 1000*i;
    dt = T0/n;
    t = 0:dt:T0;
    U1(:,1) = [q0;Dq0];
    A = [1,dt;-w0^2*dt,1];
    for j=1:length(t)-1
        U1(:,j+1) = A*U1(:,j);
    end
    subplot(2,2,i);
    plot(t,U1(1,:));
    hold on;
    plot(t,cos(2*pi*t));
    legend('euler explicite','solution exacte')
end
```



2.4

Quand on choisit $n=500$, on peut obtenir E^* comme:

```
Etoile1=[];  
for i = 1:length(t)  
    Etoile1(i) = 1/2*(U1(2,i)*U1(2,i)+w0^2*U1(1,i)*U1(1,i));  
end  
clf;  
plot(t,Etoile1);  
legend('E* euler explicite');
```



Si on change le pas de temps, E^* se change aussi. Parce que quand dt devient plus petit, ce résultat devient plus près avec la solution exacte.

2.5

```
for i=1:100
    n = 500*i;
    dt = T0/n;
    t = 0:dt:T0;
    U1(:,1) = [q0;Dq0];
    A1 = [1,dt;-w0^2*dt,1];
    vp = eig(A1)
end
```

On change le pas de temps, et on peut obtenir que c'est instable quand n est petit. Si on change n , la valeur propre de la matrice $A1$ se change aussi.

```
vp =
    1.0000 + 0.03771i
    1.0000 - 0.03771i

vp =
    1.0000 + 0.01881i
    1.0000 - 0.01881i

vp =
    1.0000 + 0.01261i
    1.0000 - 0.01261i

vp =
    1.0000 + 0.00941i
    1.0000 - 0.00941i
```

Mais quand n est assez grand (>30000), c'est stable. La valeur propre tend vers une valeur fixée.

```
vp =
    1.0000 + 0.00041i
    1.0000 - 0.00041i

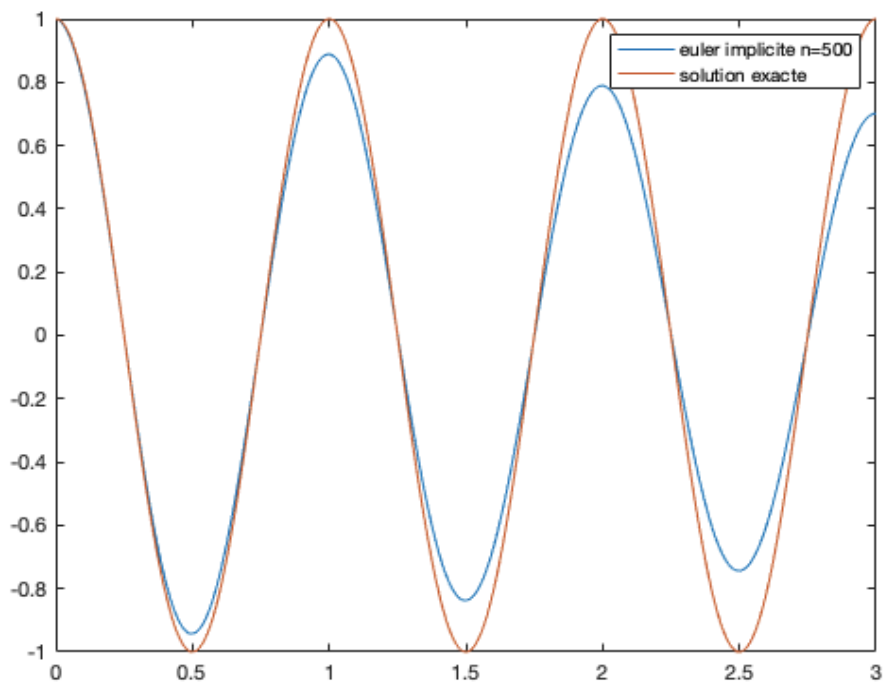
vp =
    1.0000 + 0.00041i
    1.0000 - 0.00041i

vp =
    1.0000 + 0.00041i
    1.0000 - 0.00041i

vp =
    1.0000 + 0.00041i
    1.0000 - 0.00041i
```

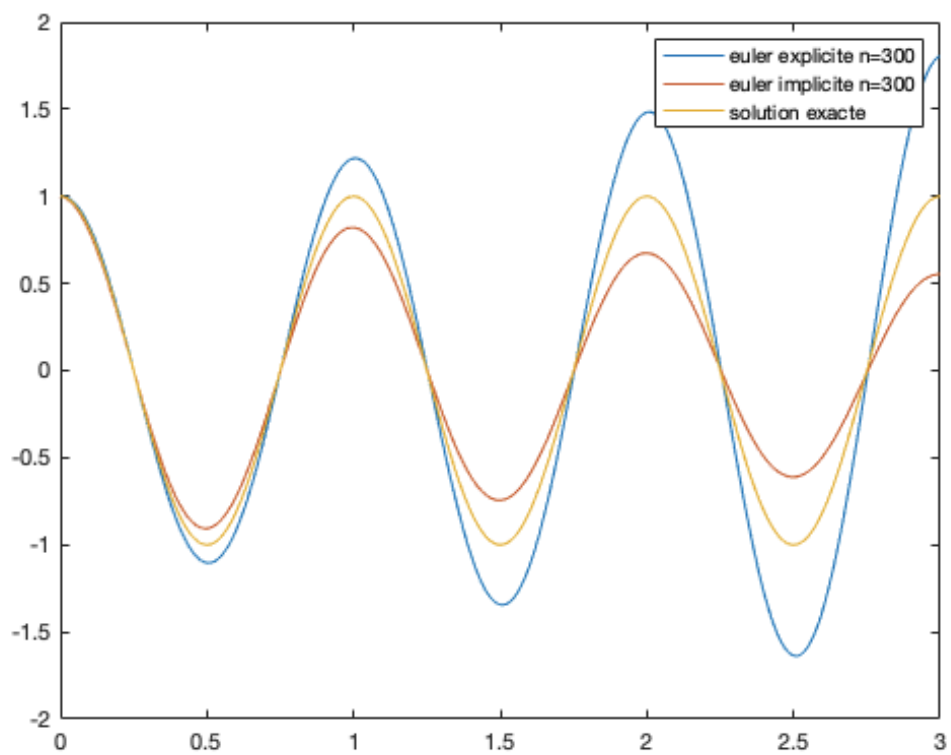
3.1 Euler implicite

```
%Euler implicite
clf;
n = 500;
dt = T0/n;
t = 0:dt:T0;
A2 = [1,-dt;w0^2*dt,1];
U2(:,1) = [q0;Dq0];
for j = 1:length(t)-1
    U2(:,j+1) = inv(A2)*U2(:,j);
end
plot(t,U2(1,:));
hold on;
plot(t,cos(w0*t))
legend('euler implicite n=500','solution exacte')
```



3.2

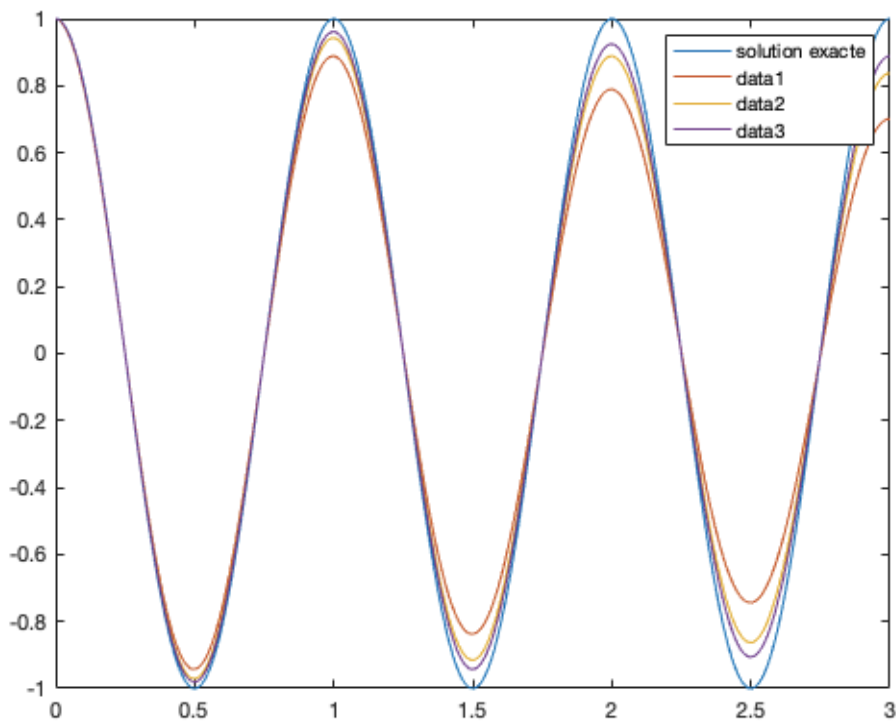
```
clf;
%on choisit n=300 pour dt=0.01
n = 300;
dt = T0/n;
t = 0:dt:T0;
A1 = [1,dt;-w0^2*dt,1];
U1(:,1) = [q0;Dq0];
A2 = [1,-dt;w0^2*dt,1];
U2(:,1) = [q0;Dq0];
for j = 1:length(t)-1
    U1(:,j+1) = A1*U1(:,j);
    U2(:,j+1) = inv(A2)*U2(:,j);
end
plot(t,U1(1,:));
hold on;
plot(t,U2(1,:));
hold on;
plot(t,cos(w0*t))
legend('euler explicite n=300','euler implicite n=300','solution exacte')
```



3.3

```
clf;
t = linspace(0,T0,500);
plot(t,cos(2*pi*t));
hold on;
legend('solution exacte');

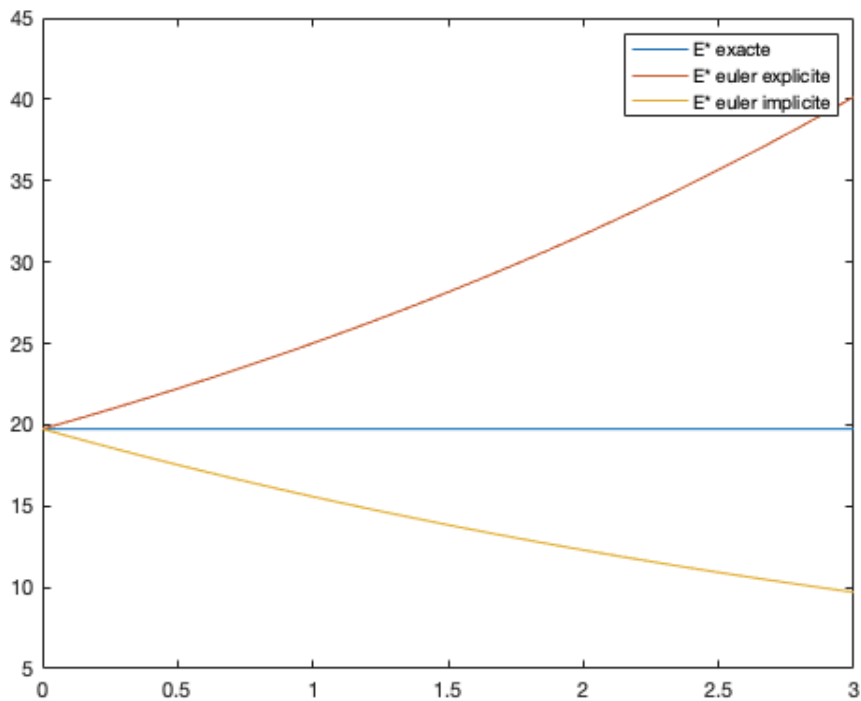
%on change le pas de temps
for i=1:3
    n = 500*i;
    dt = T0/n;
    t = 0:dt:T0;
    U2(:,1) = [q0;Dq0];
    A2 = [1,-dt;w0^2*dt,1];
    for j=1:length(t)-1
        U2(:,j+1) = inv(A2)*U2(:,j);
    end
    string = num2str(n);
    plot(t,U2(1,:));
    hold on;
end
```



On peut trouver que plus le pas de temps est petit, plus l'atténuation des oscillations est faible.

3.4

```
clf;
Etoile = [];
Etoile1 = [];
Etoile2 = [];
for i = 1:length(t)
    Etoile(i) = 2*pi^2;
    Etoile1(i) = 1/2*(U1(2,i)*U1(2,i)+w0^2*U1(1,i)*U1(1,i));
    Etoile2(i) = 1/2*(U2(2,i)*U2(2,i)+w0^2*U2(1,i)*U2(1,i));
end
clf;
plot(t,Etoile);
hold on;
plot(t,Etoile1);
hold on;
plot(t,Etoile2);
legend('E* exacte','E* euler explicite','E* euler implicite');
```



Après la figure, on peut voir que la valeur d'E* d'Euler explicite est plus grande que celui exacte, lors que la valeur d'E* d'Euler implicite est plus petite.

3.5

```
for i=1:100
    n = 500*i;
    dt = T0/n;
    t = 0:dt:T0;
    U2(:,1) = [q0;Dq0];
    A2 = [1,-dt;w0^2*dt,1];
    vp = eig(inv(A2))
end
```

On change le pas de temps, et on peut obtenir que c'est instable quand n est petit. Si on change n, la valeur propre de la matrice A1 se change aussi.

```
vp =
    0.9986 + 0.0376i
    0.9986 - 0.0376i
```

```
vp =
    0.9996 + 0.0188i
    0.9996 - 0.0188i
```

```
vp =
    0.9998 + 0.0126i
    0.9998 - 0.0126i
```

```
vp =
    0.9999 + 0.0094i
    0.9999 - 0.0094i
```

Mais quand n est assez grand (>30000), c'est stable. La valeur propre tend vers une valeur fixée.

```
vp =
    1.0000 + 0.0004i
    1.0000 - 0.0004i
```

```
vp =
    1.0000 + 0.0004i
    1.0000 - 0.0004i
```

```
vp =
    1.0000 + 0.0004i
    1.0000 - 0.0004i
```

```
vp =
    1.0000 + 0.0004i
    1.0000 - 0.0004i
```

4.1

l'équation du mouvement (1) =

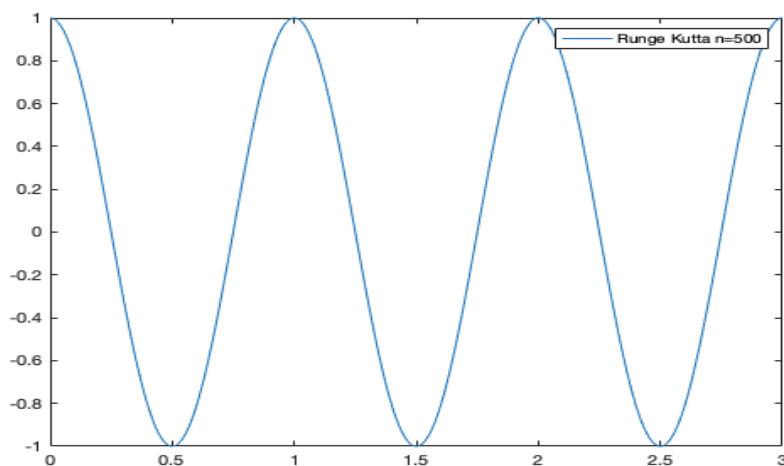
$$\ddot{q} = -\omega_0^2 q$$

$$\begin{array}{c|c} \dot{q} & \\ \hline \ddot{q} & \end{array} = \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & 0 \end{bmatrix} \begin{array}{c} q \\ \dot{q} \end{array}$$

Soit $U(t) = B U(t)$

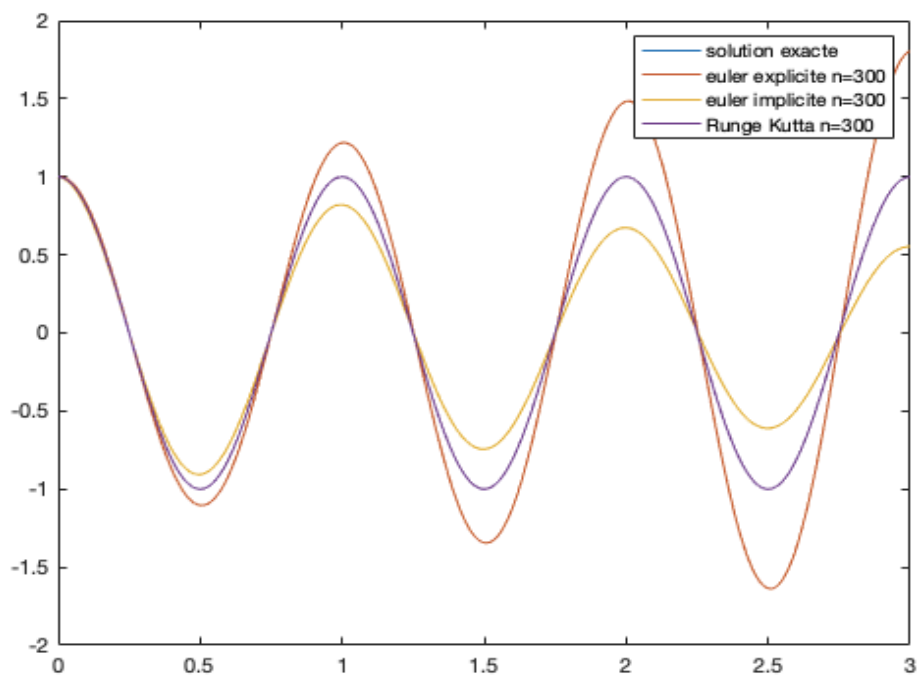
4.2

```
clf;
n = 500;
dt = T0/n;
t = 0:dt:T0;
B = [0,1;-w0^2,0];
Urk(:,1) = [q0;Dq0];
%la méthode de Runge-Kutta
for i = 1:length(t)-1
    k1 = B*Urk(:,i);
    k2 = B*(Urk(:,i)+1/2*dt*k1);
    k3 = B*(Urk(:,i)+1/2*dt*k2);
    k4 = B*(Urk(:,i)+dt*k3);
    K = 1/6*(k1+2*k2+2*k3+k4);
    Urk(:,i+1) = Urk(:,i)+K*dt;
end
plot(t,Urk(1,:))
legend('Runge Kutta n=500')
```



4.3

```
clf;
plot(t,cos(w0*t));
hold on;
plot(t,U1(1,:));
hold on;
plot(t,U2(1,:));
hold on;
plot(t,Urk(1,:));
legend('solution exacte','euler explicite n=300','euler implicite n=300','Runge Kutta n=300')
```

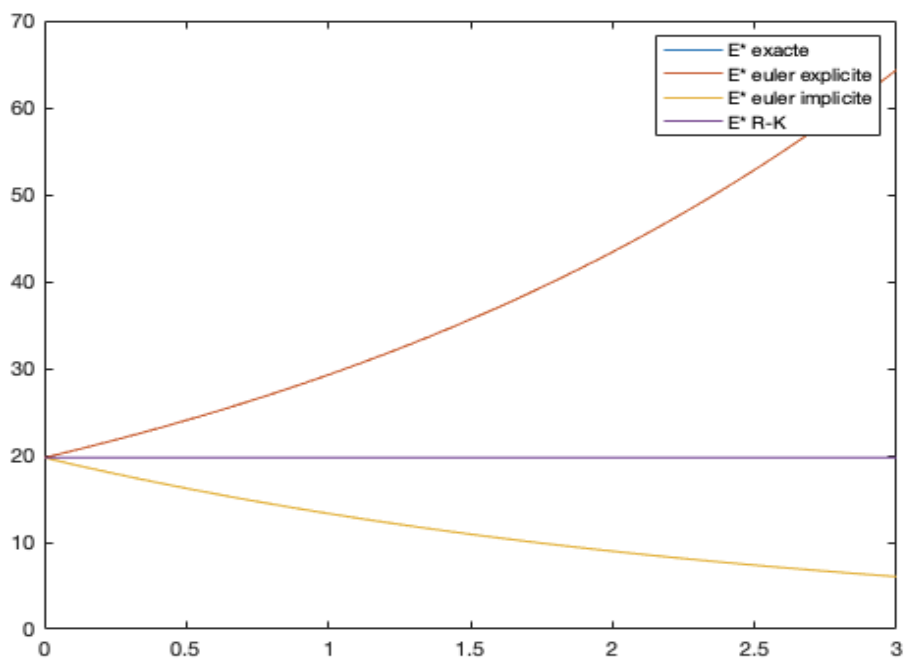


On peut obtenir selon cette figure : le résultat de la méthode de R-K est plus précise que les deux autres dans la même condition.

4.4

```
clf;
Eetoile = [];
Eetoile1 = [];
Eetoile2 = [];
Eetoile3 = [];
for i = 1:length(t)
    Eetoile(i) = 2*pi^2;
    Eetoile1(i) = 1/2*(U1(2,i)*U1(2,i)+w0^2*U1(1,i)*U1(1,i));
    Eetoile2(i) = 1/2*(U2(2,i)*U2(2,i)+w0^2*U2(1,i)*U2(1,i));
    Eetoile3(i) = 1/2*(UrK(2,i)*UrK(2,i)+w0^2*UrK(1,i)*UrK(1,i));
end

clf;
plot(t,Eetoile);
hold on;
plot(t,Eetoile1);
hold on;
plot(t,Eetoile2);
hold on;
plot(t,Eetoile3);
legend('E* exacte','E* euler explicite','E* euler implicite','E* R-K');
```

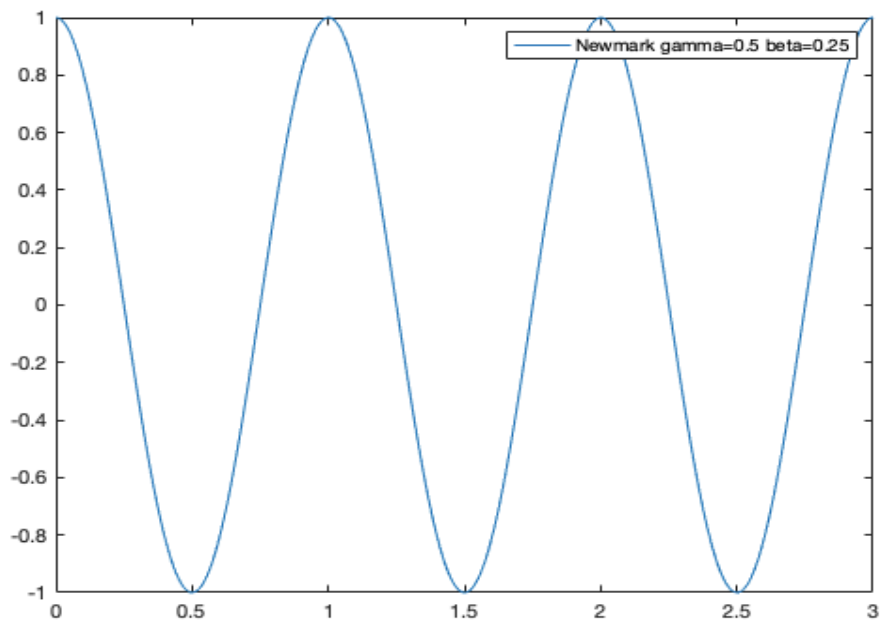


On peut obtenir selon cette figure : la quantité E^* calculée par la méthode de R-K est la même avec le résultat exacte, donc cette méthode est plus précise pour notre problème.

5.1 Résolution avec un schéma de *Newmark* $\gamma=0,5$ $\beta=0.25$

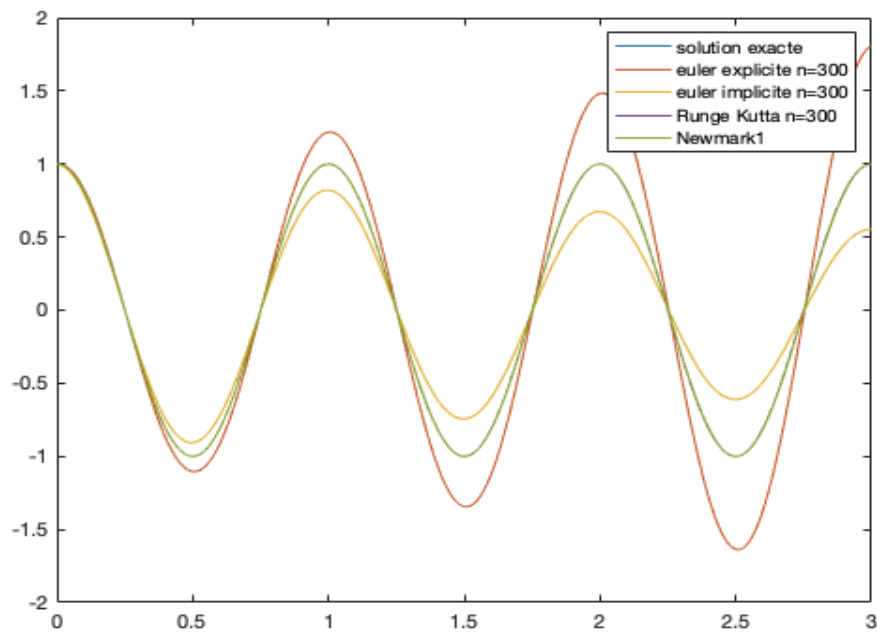
5.1.1

```
clf;
n = 300;
dt = T0/n;
t = 0:dt:T0;
gamma1 = 0.5;
beta1 = 0.25;
C1 = [1+beta1*dt^2*w0^2,0;gamma1*dt*w0^2,1];
D1 = [1-(0.5-beta1)*dt^2*w0^2,dt;-(1-gamma1)*dt*w0^2,1];
E1 = inv(C1)*D1;
Unewm1(:,1) = [q0;Dq0];
for i = 1:length(t)-1
    Unewm1(:,i+1) = E1*Unewm1(:,i);
end
plot(t,Unewm1(1,:));
legend('Newmark gamma=0.5 beta=0.25')
```



5.1.2

```
clf;
plot(t,cos(w0*t));
hold on;
plot(t,U1(1,:));
hold on;
plot(t,U2(1,:));
hold on;
plot(t,Urk(1,:));
hold on;
plot(t,Unewm1(1,:));
legend('solution exacte','euler explicite n=300','euler implicite n=300','Runge Kutta n=300','Newmark1')
```

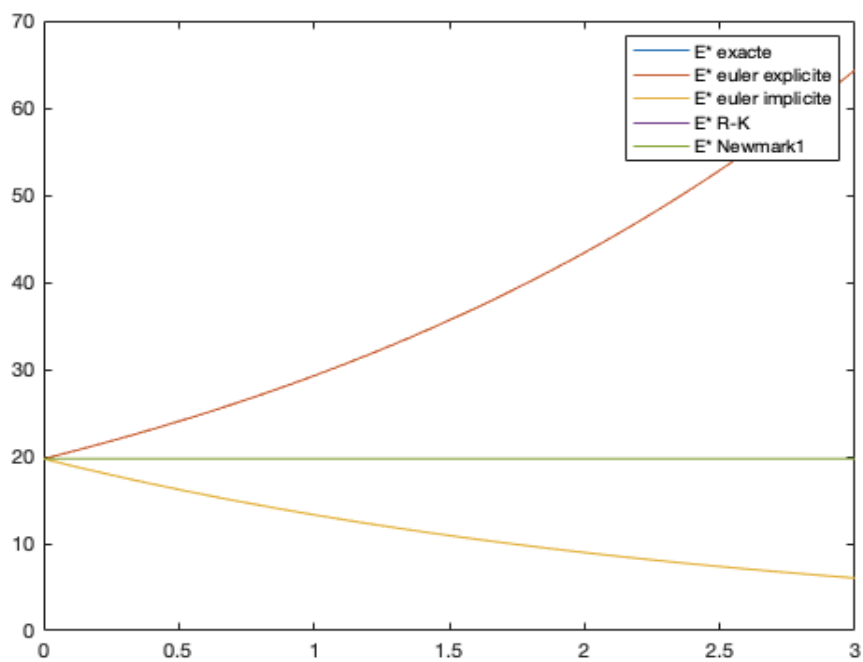


On peut obtenir selon cette figure : le résultat de la méthode Newmark est tant précise que la méthode de R-K, mieux que les deux autres dans la même condition.

5.1.3

```
clf;
Etoile = [];
Etoile1 = [];
Etoile2 = [];
Etoile3 = [];
Etoile4 = [];
for i = 1:length(t)
    Etoile(i) = 2*pi^2;
    Etoile1(i) = 1/2*(U1(2,i)*U1(2,i)+w0^2*U1(1,i)*U1(1,i));
    Etoile2(i) = 1/2*(U2(2,i)*U2(2,i)+w0^2*U2(1,i)*U2(1,i));
    Etoile3(i) = 1/2*(UrK(2,i)*UrK(2,i)+w0^2*UrK(1,i)*UrK(1,i));
    Etoile4(i) = 1/2*(Unewm1(2,i)*Unewm1(2,i)+w0^2*Unewm1(1,i)*Unewm1(1,i));
end

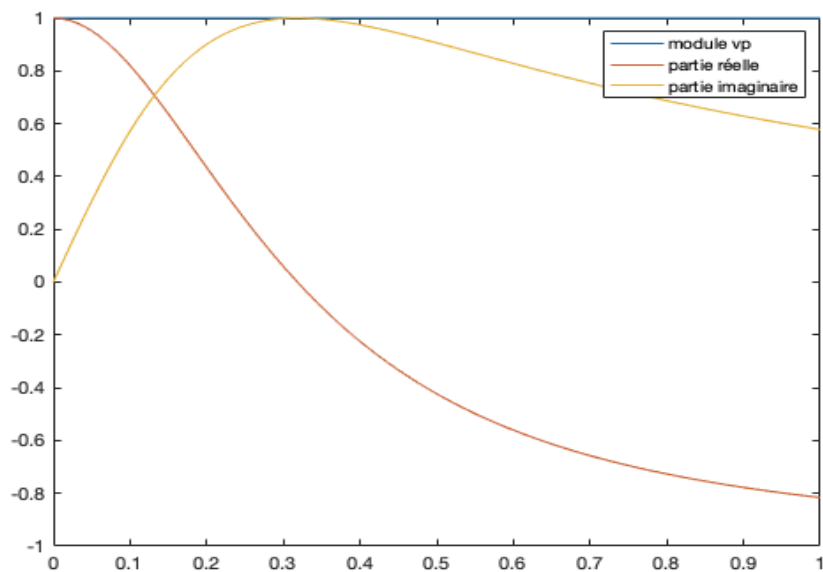
clf;
plot(t,Etoile);
hold on;
plot(t,Etoile1);
hold on;
plot(t,Etoile2);
hold on;
plot(t,Etoile3);
hold on;
plot(t,Etoile4);
legend('E* exacte','E* euler explicite','E* euler implicite','E* R-K','E* Newmark1');
```



On peut voir que avec le schéma de Newmark, $\gamma=0.5$ et $\beta=0.25$, il n'y a pas d'atténuation de E^* comme le schéma R-K.

5.1.4

```
clf;
gamma1 = 0.5;
beta1 = 0.25;
dt1 = 0:0.01:1;
vpE1 = [0;0];
module_vpE1 = [];
for i = 1:length(dt1)
    C1 = [1+beta1*dt1(i)^2*w0^2,0;gamma1*dt1(i)*w0^2,1];
    D1 = [1-(0.5-beta1)*dt1(i)^2*w0^2,dt1(i);-(1-gamma1)*dt1(i)*w0^2,1];
    E1 = inv(C1)*D1;
    vpE1(:,i) = eig(E1);
    module_vpE1(i) = sqrt((real(vpE1(1,i)))^2+(imag(vpE1(1,i)))^2);
end
plot(dt1,module_vpE1);
hold on;
plot(dt1,real(vpE1(1,:)));
hold on;
plot(dt1,imag(vpE1(1,:)));
legend('module vp','partie réelle','partie imaginaire');
```

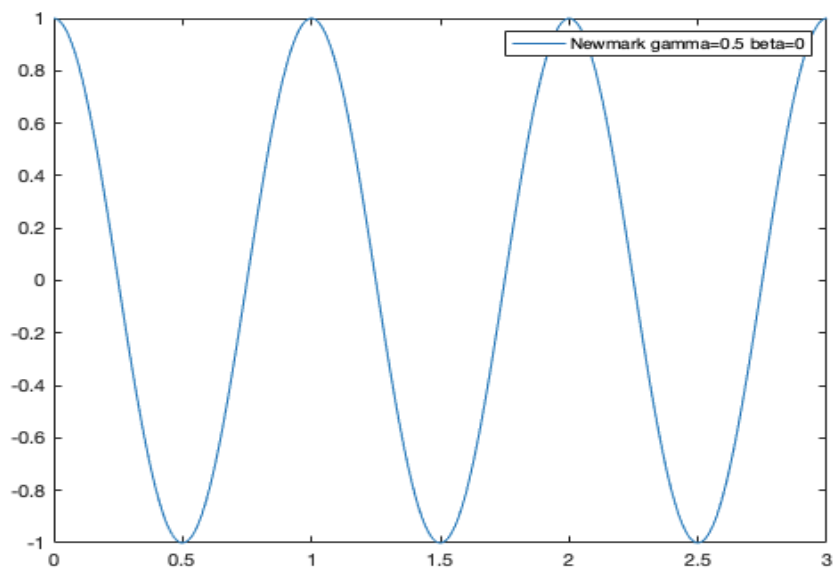


Pour conclure, on peut dire que bien que le pas de temps soit plus grand, le module des valeurs propres de la matrice d'amplification reste toujours 1.

5.2 Résolution avec un schéma de *Newmark* $\gamma=0,5$ $\beta=0$

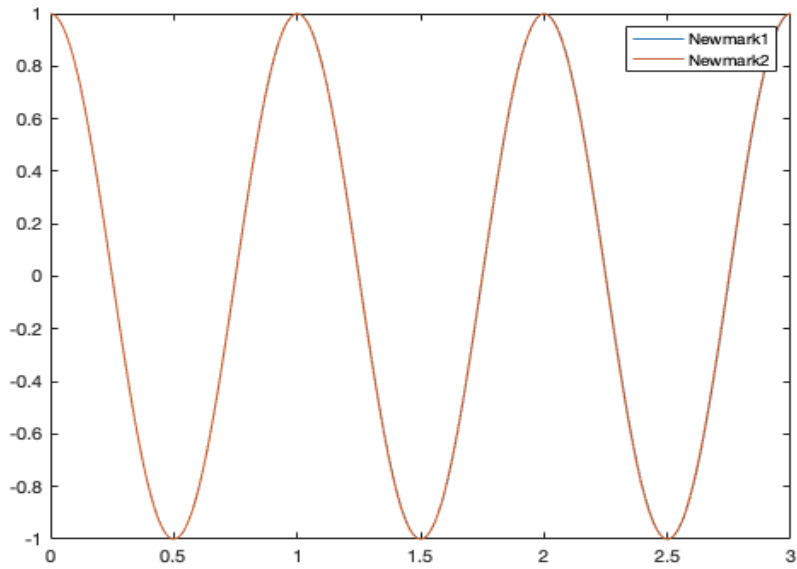
5.2.1

```
clf;
n = 300;
dt = T0/n;
t = 0:dt:T0;
gamma2 = 0.5;
beta2 = 0;
C2 = [1+beta2*dt^2*w0^2,0;gamma2*dt*w0^2,1];
D2 = [1-(0.5-beta2)*dt^2*w0^2,dt;-(1-gamma2)*dt*w0^2,1];
E2 = inv(C2)*D2;
Unewm2(:,1) = [q0;Dq0];
for i = 1:length(t)-1
    Unewm2(:,i+1) = E2*Unewm2(:,i);
end
plot(t,Unewm2(1,:));
legend('Newmark gamma=0.5 beta=0');
```



5.2.2

```
clf;  
plot(t,Unewm1(1,:));  
hold on;  
plot(t,Unewm2(1,:));  
legend('Newmark1','Newmark2');
```

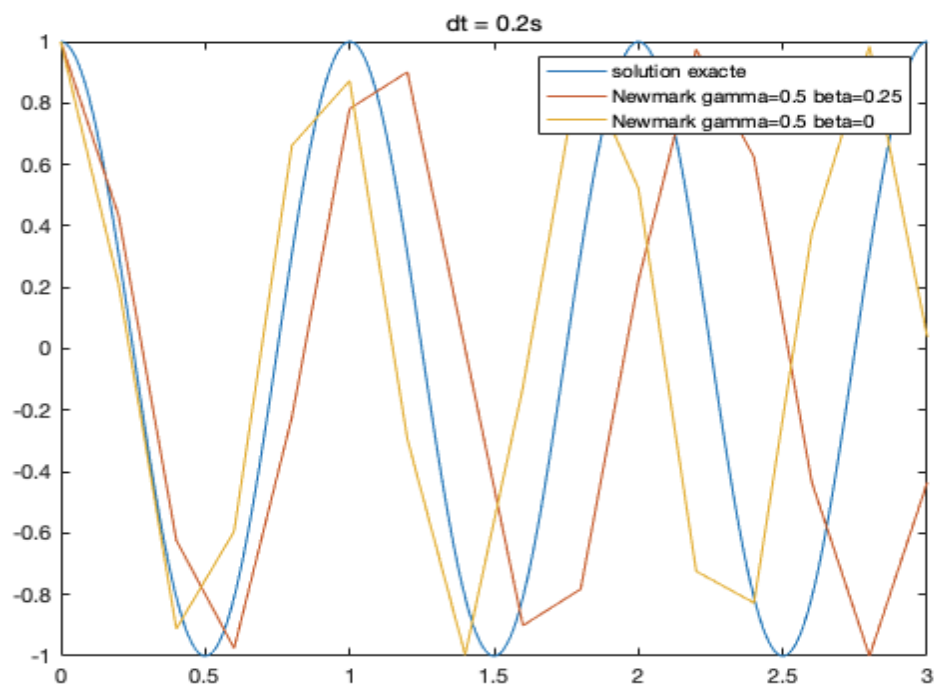


Je ne crois pas qu'il y a de différence entre ces deux résultats quand $dt = 0.01s$.

5.2.3

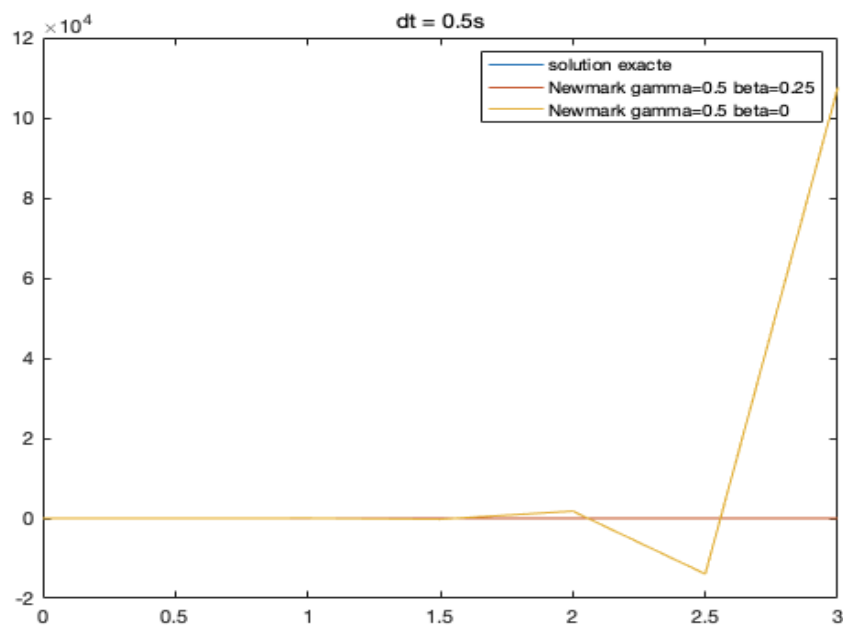
Quand on choisit $dt = 0.2s$, on peut obtenir :

```
clf;
%quand dt=0.2s
dt1 = 0.2;
t1 = 0:dt1:T0;
gamma1 = 0.5;
beta1 = 0.25;
gamma2 = 0.5;
beta2 = 0;
C1 = [1+beta1*dt1^2*w0^2,0;gamma1*dt1*w0^2,1];
D1 = [1-(0.5-beta1)*dt1^2*w0^2,dt1;-(1-gamma1)*dt1*w0^2,1];
E1 = inv(C1)*D1;
Unewm1(:,1) = [q0;Dq0];
C2 = [1+beta2*dt1^2*w0^2,0;gamma2*dt1*w0^2,1];
D2 = [1-(0.5-beta2)*dt1^2*w0^2,dt1;-(1-gamma2)*dt1*w0^2,1];
E2 = inv(C2)*D2;
Unewm2(:,1) = [q0;Dq0];
for i = 1:length(t1)-1
    Unewm1(:,i+1) = E1*Unewm1(:,i);
    Unewm2(:,i+1) = E2*Unewm2(:,i);
end
plot(t,cos(w0*t));
hold on;
plot(t1,Unewm1(1,:));
hold on;
plot(t1,Unewm2(1,:));
legend('solution exacte','Newmark gamma=0.5 beta=0.25','Newmark gamma=0.5 beta=0');
title('dt = 0.5s')
```



Si $dt = 0.5s$:

```
clf;
%quand dt=0.5s
dt1 = 0.5;
t1 = 0:dt1:T0;
gamma1 = 0.5;
beta1 = 0.25;
gamma2 = 0.5;
beta2 = 0;
C1 = [1+beta1*dt1^2*w0^2,0;gamma1*dt1*w0^2,1];
D1 = [1-(0.5-beta1)*dt1^2*w0^2,dt1;-(1-gamma1)*dt1*w0^2,1];
E1 = inv(C1)*D1;
Unewm1(:,1) = [q0;Dq0];
C2 = [1+beta2*dt1^2*w0^2,0;gamma2*dt1*w0^2,1];
D2 = [1-(0.5-beta2)*dt1^2*w0^2,dt1;-(1-gamma2)*dt1*w0^2,1];
E2 = inv(C2)*D2;
Unewm2(:,1) = [q0;Dq0];
for i = 1:length(t1)-1
    Unewm1(:,i+1) = E1*Unewm1(:,i);
    Unewm2(:,i+1) = E2*Unewm2(:,i);
end
plot(t,cos(w0*t));
hold on;
plot(t1,Unewm1(1,:));
hold on;
plot(t1,Unewm2(1,:));
legend('solution exacte','Newmark gamma=0.5 beta=0.25','Newmark gamma=0.5 beta=0');
title('dt = 0.5s')
```

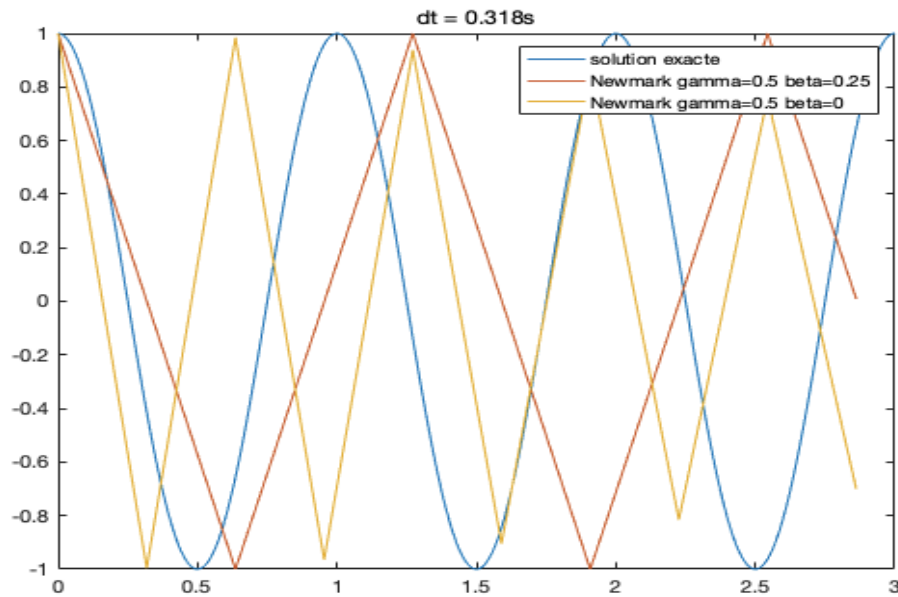


Pour conclure, on doit choisir un pas de temps assez petit, sinon le résultat de Newmark, $\gamma=0.5$ et $\beta=0$, devenira divergent.

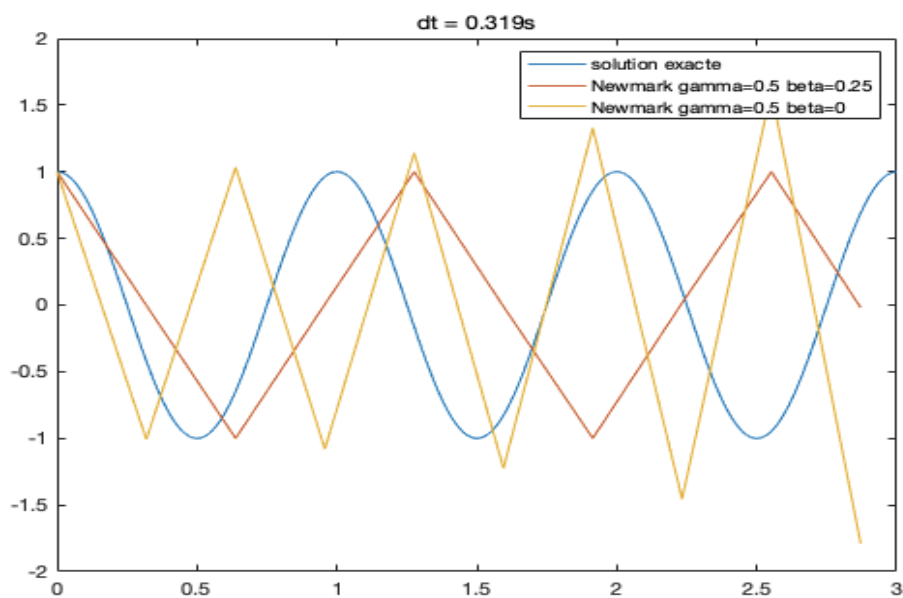
5.2.4

On utilise la méthode de dichotomie, et on peut obtenir que le pas de temps critique est environ 0.3185.

Quand $dt = 0.318s$:



Quand $dt = 0.319s$:



Donc cela est peut-être $1/\pi$.