Devoir 1

0, Pendule:

$$E_c = \frac{I\dot{\theta}^2}{2}$$

$$E_d = -mgd\cos\theta + Cte$$

$$\delta W = 0$$

En utilisant les équations de Lagrange, on obtient :

$$\frac{d}{dt}[\frac{\partial E_c}{\partial \dot{x}_i}] = I\ddot{\theta} \qquad (1)$$

$$\frac{\partial E_p}{\partial x_i} = mgd\sin\theta \qquad (2)$$

(1)+(2)=0, donc :

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

En utilisant l' hypothèse de petit mouvement, on obtient une équation linéaire :

$$I\ddot{\theta} + mgd\theta = 0$$

1,1 （1） est $\ddot{q} + \omega_0^2 q = 0$, son équation caractéristique est :

$$q^2 + \omega_0^2 q = 0$$

donc son racine caractéristique est $\pm w_0 i$ qui est irréel, donc sa solution générale est :

$$q = e^0(C_1\cos\omega_0 t + C_2\sin w_0 t)$$

Valeurs initiales : t=0, $q_0 = C_1\cos 0 + C_2\sin 0 = 1$, donc $C_1 = 1$

$$\dot{q}_0 = C_2\cos 0 - C_1\sin 0 = 0, \text{ donc } C_2 = 0$$

Donc, $q = \cos\omega_0 t$ est bien le résultat.

Sur Matlab, on voir bien le même résultat.

```
>> %Lagrange exacte
syms b q w
w=2*pi;
b='D2q=-(2*pi)^2*q';
 y=dsolve(b,'q(0)=1','Dq(0)=0')

y =

cos(2*pi*t)
```

Mais je ne sais pas comment utiliser To=3s quand je résous cette équation, c' est seulement un constant dans un point.

1.2 On calcule $E^* = \frac{1}{2}(\dot{q}^2 + \omega_0^2 q^{2)}$ sur Matlab

```
>> Y=diff(y)
 E=1/2*(Y^2+4*y^2)

Y =

-2*pi*sin(2*pi*t)


E =

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

Je trouve que 2*pi^2*cos(2*pi*t)^2 + 2*pi^2sin(2*pi*t)^2=2*pi^2, parce que cos(t)^2+sin(t)^2=1, c' est-à-dire que le quantité E^* est conservatif.

2.1 D' après (5), on sait que $q_{j+1} = q_j + \delta t \dot{q}_j$ et $\dot{q_{j+1}} = \dot{q}_j + \delta t \ddot{q}_j$

Et pour (6), le premier est bien $q_{j+1} = q_j + \delta t \dot{q}_j$, mais pour le deuxième, il faut changer la forme par $\ddot{q} + \omega_0^2 q = 0$, alors, $\dot{q_{j+1}} = \dot{q}_j + \delta t \ddot{q}_j = \dot{q}_j - \omega_0^2 q_j \delta t$, c' est (6).

2.2 J' utilise la méthode 1 :

```
%Euler explicit
syms yex Y YY y0 Y0 w T E yim yvr YYY
for j=1:10
    y(1)=y0
    Y(1)=Y0
    YY(1)=-w^2*y0
    y(j+1)=y(j)+T*Y(j)
    Y(j+1)=Y(j)+T*YY(j)
    YY(j+1)=-w^2*y(j+1)
```

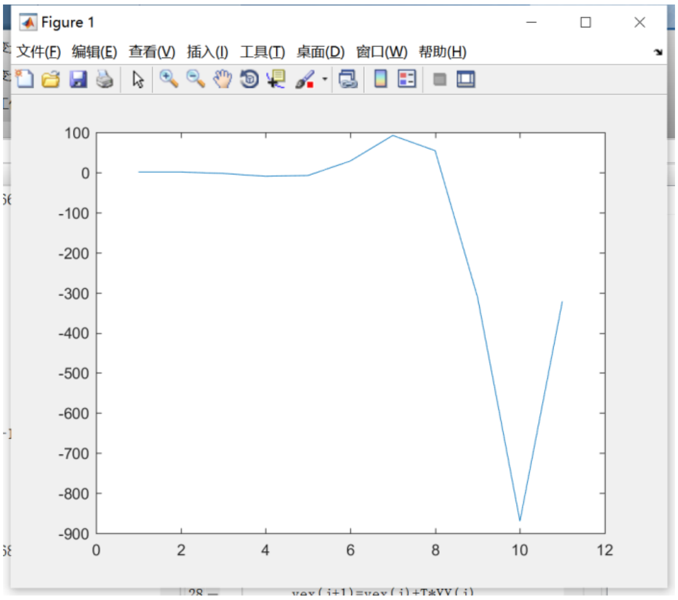
On voit que les derniers résultats sont trop longs.

```
y =

[ y0, y0 + T*Y0, y0 + T*Y0 + T*(- T*y0*w^2 + Y0), y0 + T*Y0 - T*(T*w^2*y0 - Y0 + T*w^2*(y0 + T*Y0)) + T*(- T*y0*w^2 + Y0), y0 + T*Y0 - T*(T*w^2*y0 - Y0 + T*w^2*(y0

Y =

[ Y0, - T*y0*w^2 + Y0, Y0 - T*w^2*y0 - T*w^2*(y0 + T*Y0), Y0 - T*w^2*y0 - T*w^2*(y0 + T*Y0) - T*w^2*(y0 + T*Y0 + T*(- T*y0*w^2 + Y0)), Y0 - T*w^2*y0 - T*w^2*(y0 +

YY =

[ -w^2*y0, -w^2*(y0 + T*Y0), -w^2*(y0 + T*Y0 + T*(- T*y0*w^2 + Y0)), -w^2*(y0 + T*Y0 - T*(T*w^2*y0 - Y0 + T*w^2*(y0 + T*Y0)) + T*(- T*y0*w^2 + Y0)), -w^2*(y0 + T*Y
```

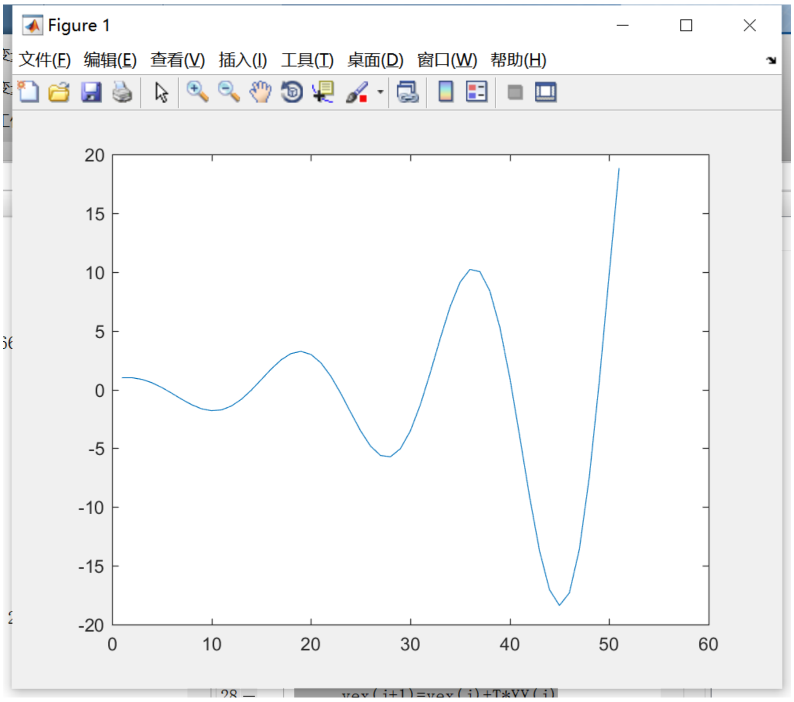
2.3 Ici je suppose que j=1 :10, c’ est-à-dire que je vais diviser To=3s en 10 parties, et T=0.3s

```
%En valeurs
 for j=1:10
     w=2*pi
     yex(1)=1
     YY(1)=0
     YYY(1)=-w^2*1
     T=0.3
     yex(j+1)=yex(j)+T*YY(j)
     YY(j+1)=YY(j)+T*YYY(j)
     YYY(j+1)=-w^2*yex(j+1)
 end
```

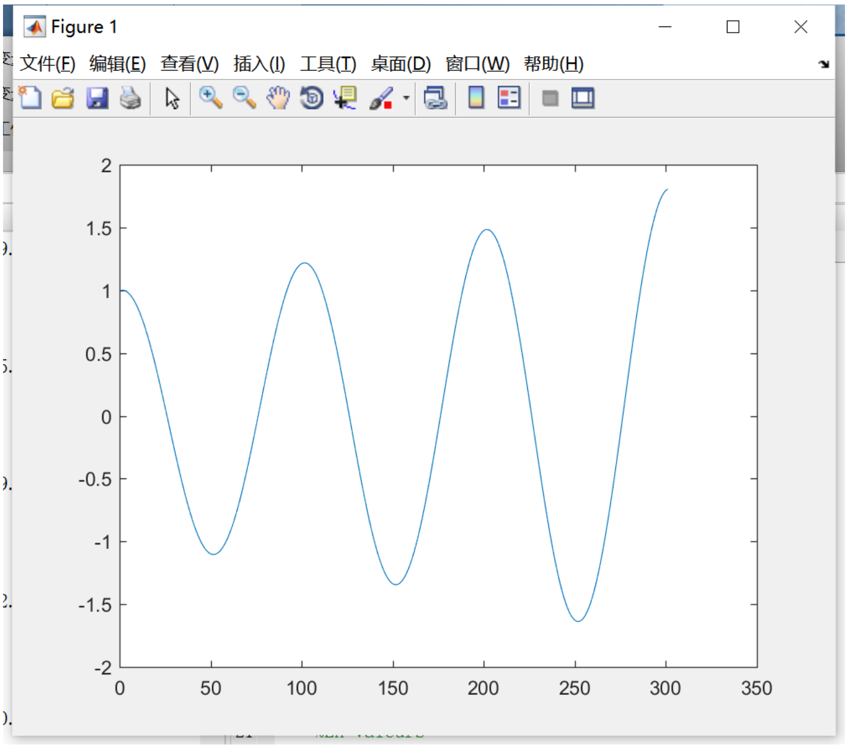
On trouve qu’ il est par hasard et il diverge au final.



Si je suppose que j=1 :50, T=0.06s, il oscille de plus en plus grand, et il diverge au final.



Si je suppose que j=1 :300, T=0.01s. On voit qu’ il oscille de plus en plus grand, et il diverge plus lentement qu’ avant.



2.4 On ajoute ci-dessous après le code.

```
E=1/2*(Y(j+1)^2+2^2*pi^2*y(j+1)^2)
```

Et on obtient ce tableau ci-dessous :

T	E explicite
0.3	7.56e+07
0.006	1.52e+04
0.03	648.07
0.01	64.37
0.003	28.15

Donc je trouve que E diminue rapidement, mais on ne peut pas voir sa tendance.

2.5 On trouve que ses valeurs propres sont deux complexes conjugués, son module est $\sqrt{1+(6.28T)^2} > 1$, parce que T>0, donc ce schéma est inconditionnellement divergent.

```
>> a=[1 T;-2^2*pi^2*T 1]
[x,y] =eig(a)

a =

[                                  1,  T]
[ -(2778046668940015*T)/70368744177664,  1]


x =

[ 70368744177664/(2778046668940015*T) + (70368744177664*((2778046668940015^(1/2)*T*1i)/838
[


y =

[ 1 - (2778046668940015^(1/2)*T*1i)/8388608,                                          0]
[                                         0, (2778046668940015^(1/2)*T*1i)/8388608 + 1]

>> vpa(y)

ans =

[ 1.0 - T*6.283185307179586277494736737083 8i,                                        0]
[                                          0, 1.0 + T*6.2831853071795862774947367370838i]
```

3.1 En changeant y(j+1) on obtient Euler implicite :

```
%Euler implicite
 syms y Y YY y0 Y0 w T E
for j=1:10
    y(1)=y0
    Y(1)=Y0
    YY(1)=-w^2*y0
    y(j+1)=(y(j)+T*Y(j))/(1+w^2*T^2)
    Y(j+1)=Y(j)+T*YY(j)
    YY(j+1)=-w^2*y(j+1)
end
```

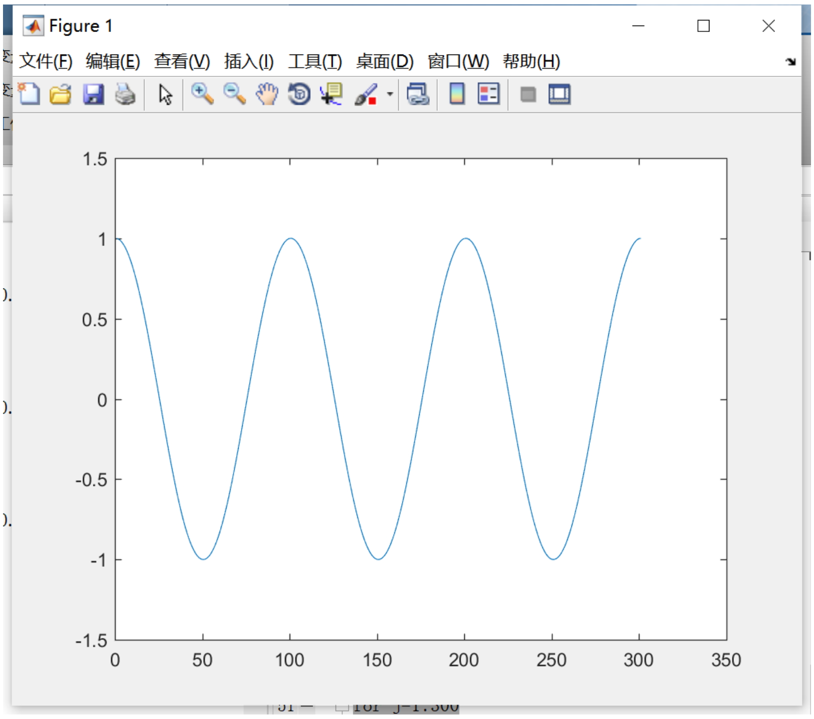
Le résultat est long comme le suivant :

```
y =

[ y0, (y0 + T*Y0)/(T^2*w^2 + 1), (T*(- T*y0*w^2 + Y0) + (y0 + T*Y0)/(T^2*w^2 + 1))/(T^2*w^2 + 1), ((T*(- T*y0*w^2 + Y0) + (y0 + T*Y0)/(T^2*w^2 + 1))/(T^2*w^2


Y =

[ Y0, - T*y0*w^2 + Y0, Y0 - T*w^2*y0 - (T*w^2*(y0 + T*Y0))/(T^2*w^2 + 1), Y0 - T*w^2*y0 - (T*w^2*(y0 + T*Y0))/(T^2*w^2 + 1) - (T*w^2*(T*(- T*y0*w^2 + Y0) + (y


YY =

[ -w^2*y0, -(w^2*(y0 + T*Y0))/(T^2*w^2 + 1), -(w^2*(T*(- T*y0*w^2 + Y0) + (y0 + T*Y0)/(T^2*w^2 + 1)))/(T^2*w^2 + 1), -(w^2*((T*(- T*y0*w^2 + Y0) + (y0 + T*Y0)
```

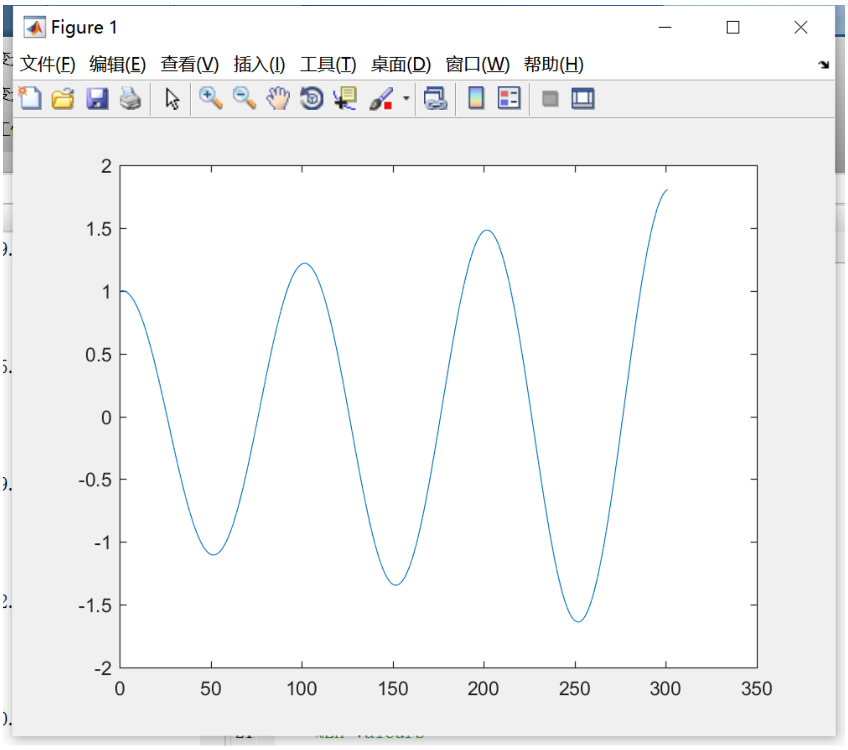
3.2 Quand T=0.01 :

```
for j=1:300
    yim(1)=1
    Y(1)=0
    w=2*pi
    T=0.01
    YY(1)=-w^2*1
    yim(j+1)=(yim(j)+T*Y(j))/(1+w^2*T^2)
    Y(j+1)=Y(j)+T*YY(j)
    YY(j+1)=-w^2*yim(j+1)
end
```

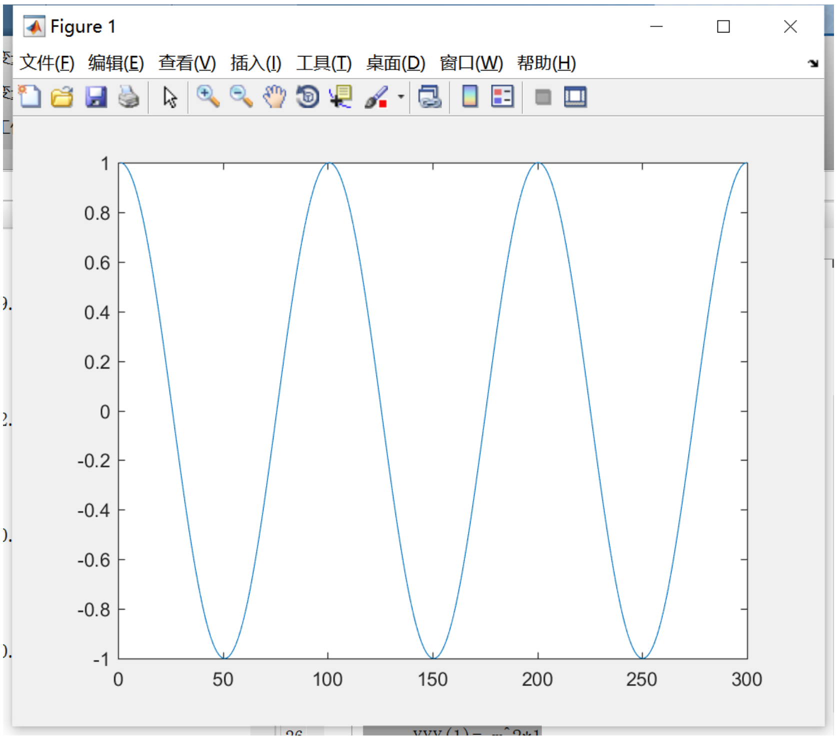
Le figure d’ Euler implicite :



Le figure d’ Euler explicite :

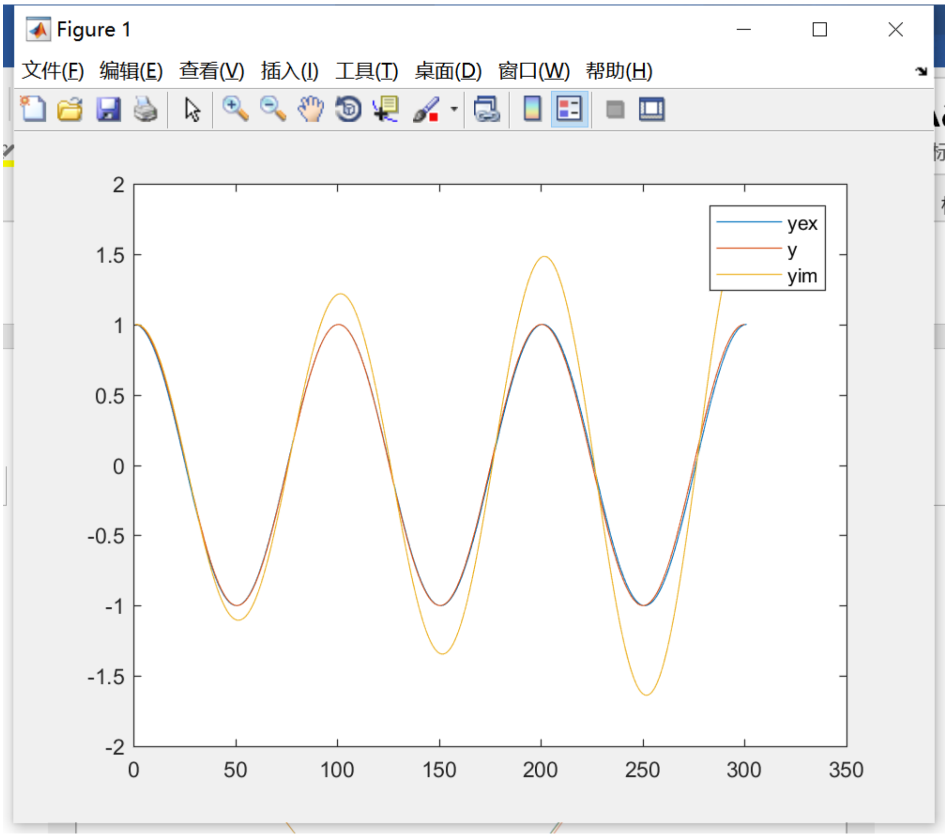


Le figure de solution exacte :



On trouve que la solution d' Euler implicite ressemble beaucoup à la solution exacte.

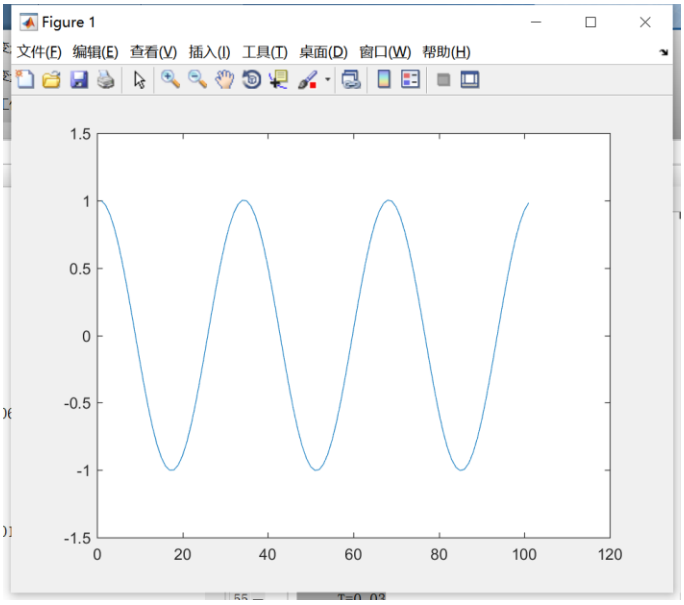
On voit plus clairement dans la figure ensemble ci-dessous :



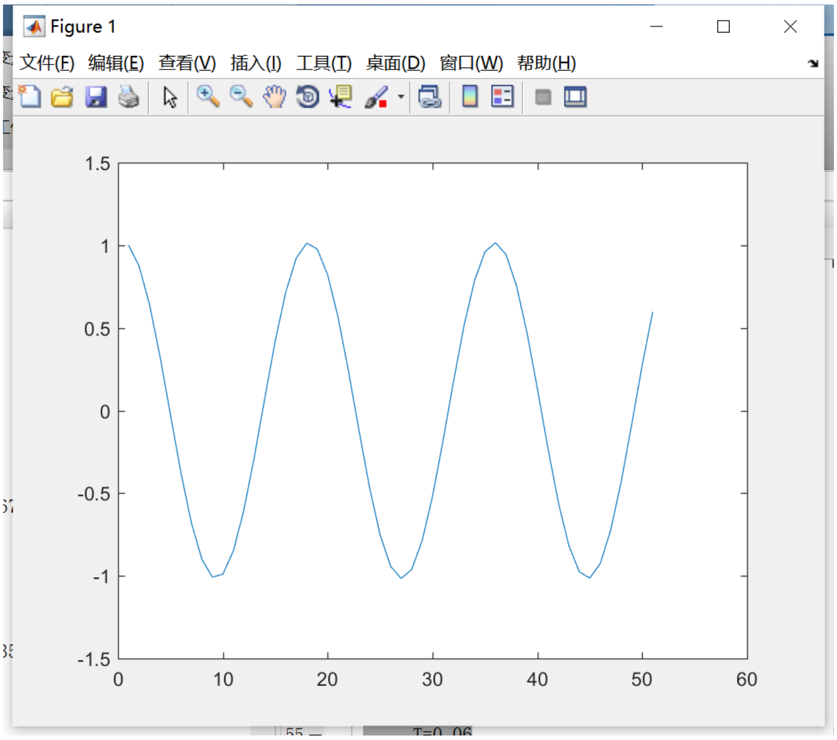
y : solution exacte ; yex : Euler explicite ; yim : Euler implicite

On trouve que y et yim superpose bien mais il y a une grande différence entre yex et y, ça implique que c’ est une fasse solution et de plus, il diverge lentement. Et Euler implicite est Une bonne solution.

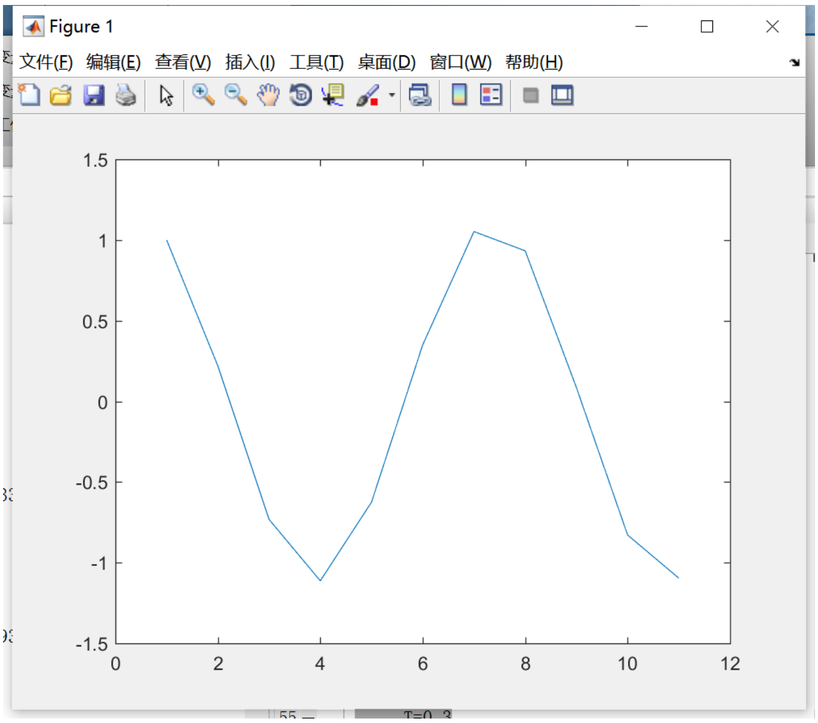
3.3 Pour le pas T=0.03, j=1 :100, on trouve qu’ il y a un peu amortissement autour de pic.



Pour le pas T=0.06, j=1 :50, on trouve qu’ il y a un amortissement clair au pic.



Pour le pas T=0.3, j=1 :10, on trouve qu' il y a un grand amortissement et le figure est faux.



Donc on obtient le résultat que : plus le pas de temps Δt est petit, plus l' atténuation des oscillations est faible.

3.4 Pour Euler explicite, on ne peut pas voir sa tendance. Pour Euler implicite, je trouve que E diminue vers 19.7392 et je pense que quand j augmente encore et T encore diminue, E va rapprocher et converger vers sa valeur exacte, 19.7392. On trouve que pour la valeur de E, Euler implicite est le plus proche de la valeur exacte. Donc c’ est la schéma solution parmi les deux. Et pour Euler explicite, on ne peut pas voir sa tendance.

Le tableau de ces trois schémas est le suivant :

<table>
<tr><th>T</th><th>E explicite</th><th>E implicite</th><th>E exacte</th></tr>
<tr><td>0.3</td><td>7.56e+07</td><td>10.64</td><td rowspan="5">19.7392</td></tr>
<tr><td>0.006</td><td>1.52e+04</td><td>19.55</td></tr>
<tr><td>0.03</td><td>648.07</td><td>20.87</td></tr>
<tr><td>0.01</td><td>64.37</td><td>19.78</td></tr>
<tr><td>0.003</td><td>28.15</td><td>19.74</td></tr>
</table>

3.5 On trouve que ses valeurs propres sont deux réels, Quand j’ essaye de résoudre(0.5*(5.3e7*T*(- 8.3e15*T^2 - 2.8e14)^(1/2) + 2.8e15*T^2 + 1.4e14))/(2.8e15*T^2 + 7.0e13)=1 ou =-1, c’ est-à-dire que ‘1-a*T=1’ , je trouve que le résultat est 0. Je le vérifie, et c’ est correct. On sait tous les temps sont positives donc (0.5*(5.3e7*T*(- 8.3e15*T^2 - 2.8e14)^(1/2) + 2.8e15*T^2 + 1.4e14))/(2.8e15*T^2 + 7.0e13)<1 est toujours vrai.

```
>> aim=[1/(1+w^2*T^2) T/(1+w^2*T^2);-w^2*T 1];
  [m,n]=eig(aim)

n =

[ 70368744177664/(2778046668940015*T) - (35184372088832*(2778046668940015*T^2 + 2778046668940015^(1/2)*T*(- 8334140006820045*T^2 - 281474976710656)^(1/2)
[


n =

[ (2778046668940015*T^2 + 2778046668940015^(1/2)*T*(- 8334140006820045*T^2 - 281474976710656)^(1/2) + 140737488355328)/(2*(2778046668940015*T^2 + 70368744
[

>> vpa(n,2)

ans =

[ (0.5*(5.3e7*T*(- 8.3e15*T^2 - 2.8e14)^(1/2) + 2.8e15*T^2 + 1.4e14))/(2.8e15*T^2 + 7.0e13),
[                                                                                           0,  (0.5*(2.8e15*T^2 - 5.3e7*T*(- 8.3e15*T^2 - 2.8e14)^(1/2) + 1.
```

```
>> solve('(0.5*(5.3e7*T*(- 8.3e15*T^2 - 2.8e14)^(1/2) + 2.8e15*T^2 + 1.4e14
警告: Support of character vectors that are not valid variable names or
define a number will be removed in a future release. To create symbolic
expressions, first create symbolic variables and then use operations on
them.
> In sym>convertExpression (line 1586)
  In sym>convertChar (line 1491)
  In sym>tomupad (line 1243)
  In sym (line 199)
  In solve>getEqns (line 406)
  In solve (line 226)
警告: Do not specify equations and variables as character vectors.
Instead, create symbolic variables with syms.
> In solve>getEqns (line 446)
  In solve (line 226)

ans =

                                        0
 (21808290^(1/2)*53i)/1557735

>> solve('(0.5*(2.8e15*T^2 - 5.3e7*T*(- 8.3e15*T^2 - 2.8e14)^(1/2) +
警告: Support of character vectors that are not valid variable names
create symbolic variables and then use operations on them.
> In sym>convertExpression (line 1586)
  In sym>convertChar (line 1491)
  In sym>tomupad (line 1243)
  In sym (line 199)
  In solve>getEqns (line 406)
  In solve (line 226)
警告: Do not specify equations and variables as character vectors. In
> In solve>getEqns (line 446)
  In solve (line 226)

ans =

                                         0
 -(21808290^(1/2)*53i)/1557735
```

Et pour (0.5*(2.8e15*T^2 - 5.3e7*T*(- 8.3e15*T^2 - 2.8e14)^(1/2) + 1.4e14))/(2.8e15*T^2 + 7.0e13)=1 ou -1, on trouve que le résultat est complexe, mais pour un temps T, il ne peut pas être un complexe, donc ce résultat n'a pas de sens, on ne le considère pas. En prenant le résultat avant, on sait que le temps convient toujours a la condition. Donc ce schéma est inconditionnellement convergent.

```
>> solve('(0.5*(5.3e7*T*(- 8.3e15*T^2 - 2.8e14)^(1/2) + 2.8e15*T^2 + 1.4e1
警告: Support of character vectors that are not valid variable names or
define a number will be removed in a future release. To create symbolic
expressions, first create symbolic variables and then use operations on
them.
> In sym>convertExpression (line 1586)
  In sym>convertChar (line 1491)
  In sym>tomupad (line 1243)
  In sym (line 199)
  In solve>getEqns (line 406)
  In solve (line 226)
警告: Do not specify equations and variables as character vectors.
Instead, create symbolic variables with syms.
> In solve>getEqns (line 446)
  In solve (line 226)

ans =

  ((371*3209^(1/2))/4693735 - 137263/4693735)^(1/2)
 -(- (371*3209^(1/2))/4693735 - 137263/4693735)^(1/2)

>> solve('(0.5*(2.8e15*T 2 - 5.3e7*T*(- 8.3e15*T 2 - 2.8e14) (1/2) + 1.4e1
警告: Support of character vectors that are not valid variable names or de
create symbolic variables and then use operations on them.
> In sym>convertExpression (line 1586)
  In sym>convertChar (line 1491)
  In sym>tomupad (line 1243)
  In sym (line 199)
  In solve>getEqns (line 406)
  In solve (line 226)
警告: Do not specify equations and variables as character vectors. Instead,
> In solve>getEqns (line 446)
  In solve (line 226)

ans =

 (- (371*3209^(1/2))/4693735 - 137263/4693735)^(1/2)
 -((371*3209^(1/2))/4693735 - 137263/4693735)^(1/2)
```

4.1 L' équation au premier ordre :

$$\begin{cases} u = \dot{q} \\ \dot{u} = -w^2 q \end{cases}$$

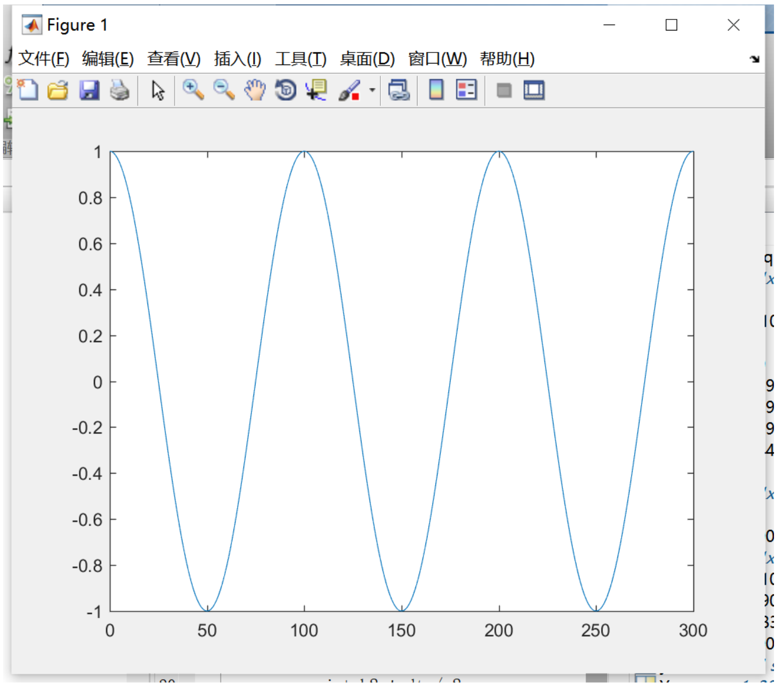
4.2 Programmer de Runge Kutta. La fonction est comme le suivant :

```
function [ dUc ] = cal_f( Uc, tc, w0c )
%UNTITLED Summary of this function goes here
%   Detailed explanation goes here
dUc = zeros(2,1);

dUc(1) = Uc(2);
dUc(2) = - (2 * pi)^2 * Uc(1);

end
```

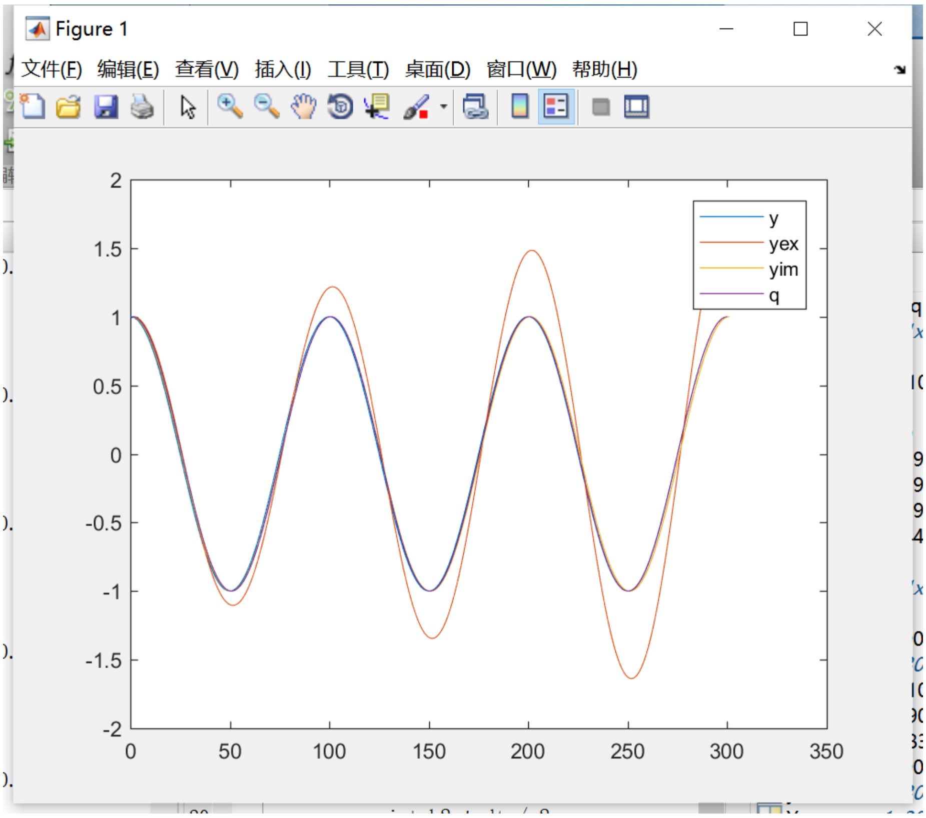
On utilise cette fonction comme le suivant :

```
q0 = 1;
dq0 = 0;
dt = 0.01;
t = (0: dt: 3)';
np = size(t, 1);
q = zeros(np, 1);
dq = zeros(np, 1);
q(1) = q0;
dq(1) = dq0;
qj = [q0; dq0];

for inc = 2: np
    tc = t(inc-1);
    xc = qj;
    k1 = cal_f(xc, tc, 0);
    xc = qj + k1 * dt / 2;
    k2 = cal_f(xc, tc + dt /2, 0);
    xc = qj + k2 * dt / 2;
    k3 = cal_f(xc, tc + dt / 2, 0);
    xc = qj + k3 * dt;
    k4 = cal_f(xc, tc + dt, 0);
    k = (k1 + 2 * k2 + 2 * k3 + k4)/6;
    qj = qj + k* dt;
    q(inc) = qj(1);
    dq(inc) = qj(2);
end
plot(t,dq)
```

4.3 La figure obtenue par RUNGE KUTTA



Je dessine la solution exacte, schéma d’ EULER explicite, schéma d’ EULER implicite et schéma de RUNGE KUTTA dans un même figure :



y : solution exacte ; yex : Euler explicite ; yim : Euler implicite ; q : Runge Kutta

On trouve que y, q et yim superpose bien mais il y a une grande différence entre yex et y, ça implique que c’ est une fasse solution et de plus yex diverge lentement. Et Euler implicite et Runge Kutta sont des bonnes solutions.

4.4 Les différentes valeurs de E pour les quatre schémas.

T	E explicite	E implicite	Runge Kutta	Solution exacte
0.01	64.37	19.78	19.74	19.7392

On trouve que pour la valeur de E, Runge Kutta est le plus proche de la valeur exacte. Donc c’ est la schéma solution parmi les trois.