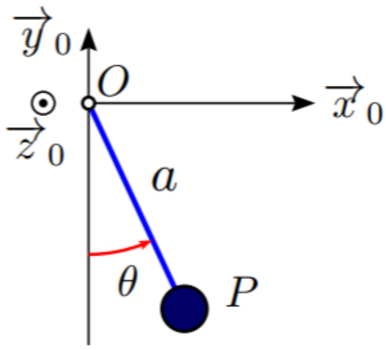
Devoir 1

0. l'équation du mouvement du pendule simple avec les équations de Lagrange



D'abord on a les expressions de l'énergie et de l'énergie potentielle, m est la masse :

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

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

$$\delta W = 0$$

En utilisant les équations de Lagrange, on obtient :

$$\frac{d}{dt}\left[\frac{\partial E_c}{\partial \dot{x}_i}\right] = ma^2\ddot{\theta}$$

$$\frac{\partial E_p}{\partial x_i} = mga\sin\theta$$

On a :

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

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

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

1.1 & 1.2 On détermine la solution analytique en utilisant le Matlab comme au dessous, d'abord on attribue des constantes.

```
w0 = 2 * pi;
q0 = 1;
T0 = 3;
dq0 = 0;
```

Et puis on utilise dsolve fonction pour obtenir la solution :

```
% solution analytique
digits(3);
syms q(t);
Dq = diff(q, t);
D2q = diff(q, t, 2);
equ = D2q + w0^2 * q == 0;
con = [q(0) == q0, Dq(0) == dq0];
q = vpa(dsolve(equ, con));
q
Dq = vpa(diff(q));
E = vpa(1/2 * (Dq^2 + w0^2 * q^2));
E
```

On obtient la solution et l'énergie :

```
>> devoir1

q =

cos(6.28*t)


E =

19.7*sin(6.28*t)^2 + 19.7*cos(6.28*t)^2
```

Selon le résultat de l'énergie, on trouve l'énergie est une constante.

2.1

D'après (5), on sait que $q_{j+1} = q_j + \delta t \dot{q}_j$ et $\dot{q_{j+1}} = \dot{q}_j + \delta t \ddot{q}_j$, on utilise l'équation $\ddot{q} + \omega_0^2 q = 0$, et puis on obtient $q_{j+1} = q_j + \delta t \dot{q}_j$, $\dot{q_{j+1}} = \dot{q}_j + \delta t \ddot{q}_j = \dot{q}_j - \omega_0^2 q_j \delta t$, c'est (6).

2.2

J'utilise la méthode 2 pour programme, le program est au-dessous : d'abord on attribue des constantes.

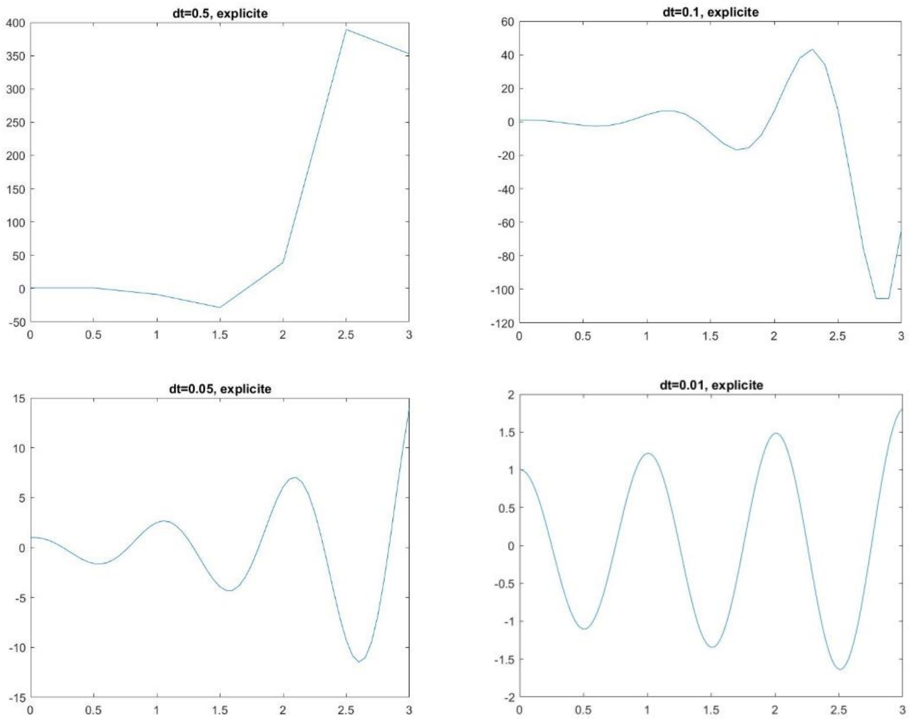
```
% EULER explicite
w0 = 2 * pi;
q0 = 1;
T0 = 3;
dq0 = 0;
dt = 0.1;
U = zeros(2, T0/dt + 1);
U(:,1) = [q0; dq0];
A = [1 dt; -w0^2 * dt 1];
```

Et puis utilise la euler explicite :

```
for i=2:(T0/dt) + 1
    U(:,i) = A * U(:, i - 1);
end
```

2.3

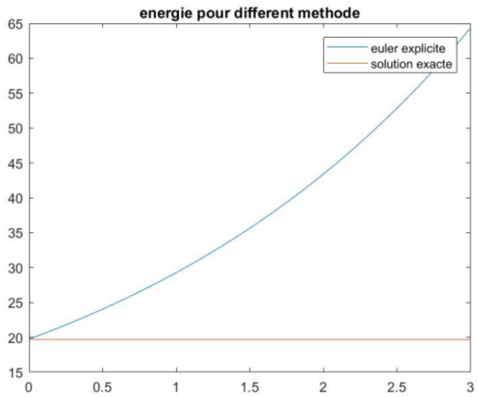
On choix dt = 0.5, 0.1, 0.05 et 0.01 pour montrer la solution numérique :



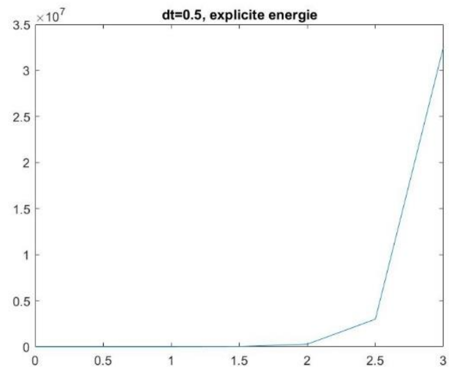
Selon les figures, on trouve la solution est divergent, et plus le pas de temps Δt est petit, plus la divergence est lente.

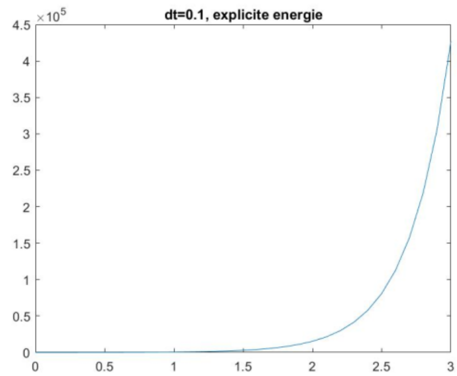
2.4

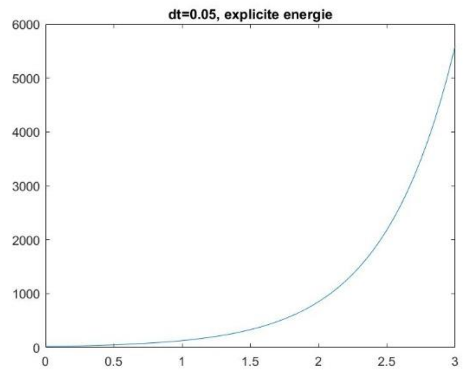
Le figure d'énergie pour solutions différentes est au-dessous, on peut voir que l'énergie obtient par euler explicite est croissant.

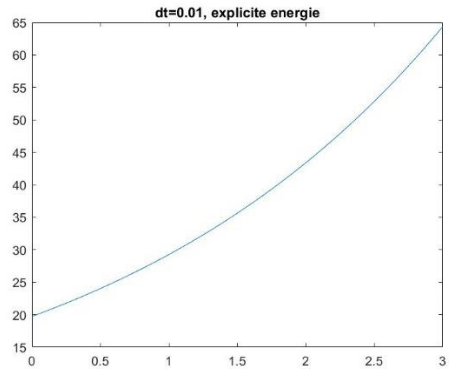


Les figures de l'énergie fait varier de dt est comme au-dessous, On choix dt = 0.5, 0.1, 0.05 et 0.01 :









On peut voir que l'énergie n'est pas stable, il est croissant, et quand le dt est plus petit, le taux de croissant est plus petit.

2.5

On utilise la fonction eig pour obtenir les valeurs propres, On trouve que ses valeurs propres sont deux complexes conjugués, car les valeurs sont plus grand que 1, alors il est divergent. Les valeurs propres sont au-dessous :

```
y =

   1.0000 + 0.0628i   0.0000 + 0.0000i
   0.0000 + 0.0000i   1.0000 - 0.0628i
```

3.1

De même, j'utilise la méthode 2 programmer. Le programme est au-dessous : d'abord on attribue des constantes.

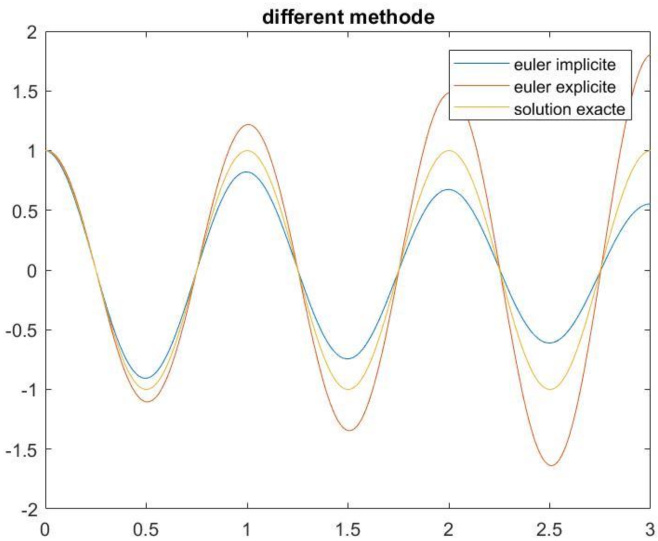
```
% EULER implicite
w0 = 2 * pi;
q0 = 1;
T0 = 3;
dq0 = 0;
dt = 0.1;
U = zeros(2, T0/dt + 1);
U(:,1) = [q0; dq0];
m = 1/(1 + w0^2 * dt^2);
A1 = [m m * dt; -(1-m)/dt m];
```

Et puis :

```
for i=2:(T0/dt) + 1
    U(:,i) = A1 * U(:, i - 1);
end
```

3.2

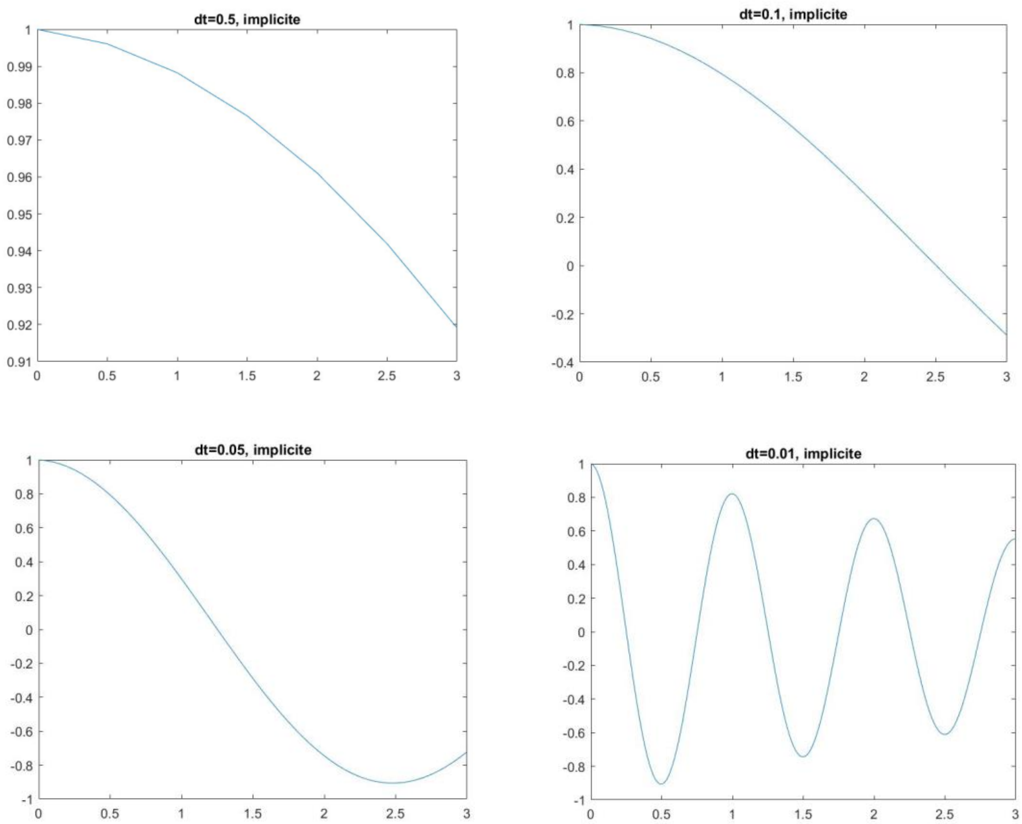
Quand dt = 0.01s, le figure est au-dessous,



Du figure, on peut voir que les trois solution ne sont pas même, et la solution implicite est amorti, elle est plus petite que la solution exacte, la solution explicite est divergent, elle est plus grande que la solution exacte.

3.3

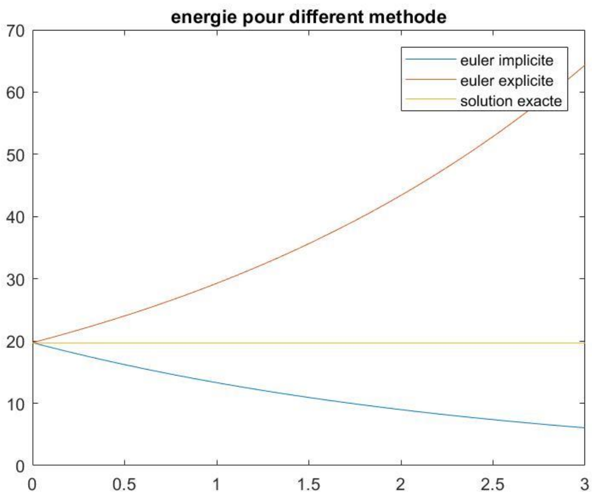
On choix dt = 0.5, 0.1, 0.05 et 0.01 pour montrer la solution numérique :



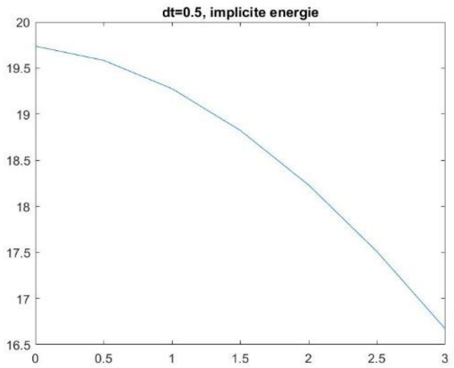
De ces quatre figures, on peut voir que le schéma d'intégration d'EULER implicite introduit un amortissement numérique. Et plus le pas de temps Δt est petit, plus l'atténuation des oscillations est faible.

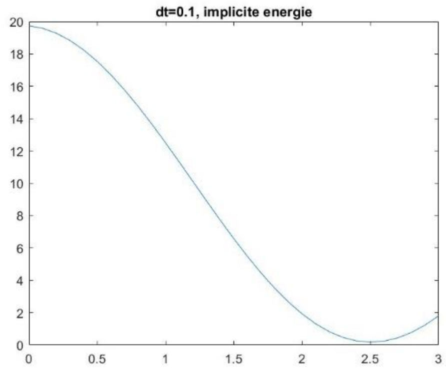
3.4

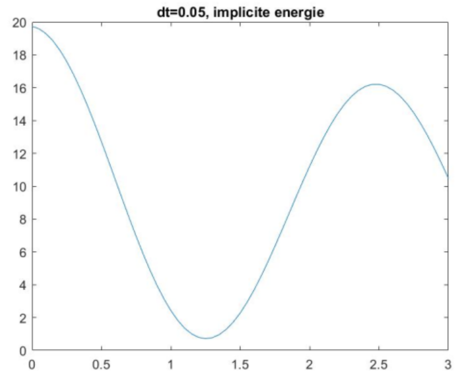
Le figure d'énergie pour solutions différentes est au-dessous, on peut voir que l'énergie obtient par euler implicite est décroissant.

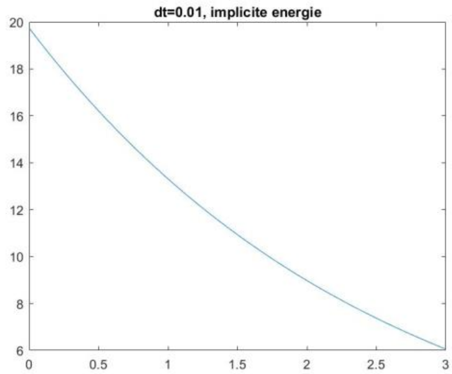


Les figures de l'énergie fait varier de dt est comme au-dessous, On choix dt = 0.5, 0.1, 0.05 et 0.01 :









On peut voir que l'énergie est décroissant, et quand le dt est plus petit, le taux de décroissant est plus petit.

3.5

On utilise la fonction eig pour obtenir les valeurs propres, On trouve que ses valeurs propres sont deux complexes conjugués, car les valeurs sont plus petit que 1, alors il est amorti et stable. Les valeurs propres sont au-dessous :

```
y =

   0.9961 + 0.0626i   0.0000 + 0.0000i
   0.0000 + 0.0000i   0.9961 - 0.0626i
```

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 :

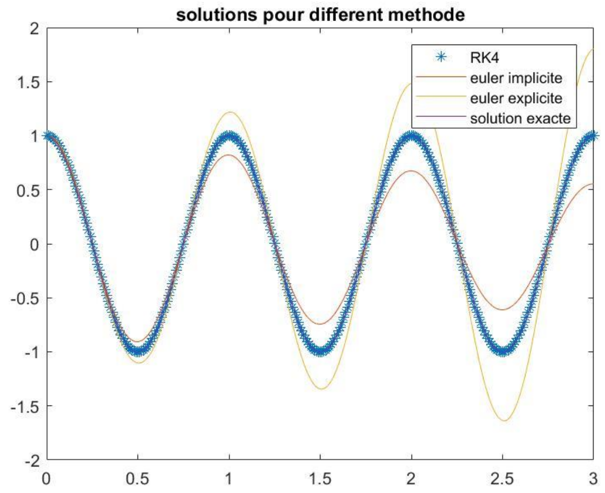
```
function [ dUc ] = cal_f( Uc, tc, w0c )
%UNTITLED Summary of this function goes here
%   Detailed explanation goes here
dUc = zeros(2,1);

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

end
```

Et puis on obtient la solution :

```
q0 = 1;
dq0 = 0;

dt = 0.01;
t = (0: dt: 3)';
np = size(t, 1);

q = zeros(np,1);
dq6 = zeros(np,1);

q(1) = q0;
dq6(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);
    dq = (k1 + 2 * k2 + 2 * k3 + k4)/6;
    qj = qj + dq * dt;
    q(inc) = qj(1);
    dq6(inc) = qj(2);
end
```

4.3

Quand dt = 0.01s, le figure est au-dessous,



De ce figure, on peut voir que la méthode RK est plus près de la solution exacte que les autres, on peut conclure la méthode RK4 est le meilleur méthode pour simuler.

4.4

Le figure d'énergie pour solutions différentes est au-dessous, on peut conclure que le RK4 peut obtenir meilleur résultat et le RK est le meilleur méthode pour simulation.