Double pendule avec l’hypothèse des petits mouvements

Table of Contents

Sébastien SY1924130

Paramètres

```
clear global;
t_start = 0;
T0 = 8;
global F0 omega M K a g m q0 dq0 ddq0
m = 2;
a = 0.5;
g = 9.81;
omega = 2*pi;
F0 = 20;
```

1.1

Soit $q = \begin{vmatrix}\theta_1 \\ \theta_2\end{vmatrix}$, $[M] = ma^2\begin{bmatrix}2 & 1 \\ 1 & 1\end{bmatrix}$, $[K] = mga\begin{bmatrix}2 & 0 \\ 0 & 1\end{bmatrix}$, $[F] = F_0 \sin\omega t \begin{vmatrix} a \\ \frac{a}{\sqrt{2}}\end{vmatrix}$

Alors $[M]\ddot{q} + [K]q = [F]$

$$\ddot{q} = [M]^{-1}[F] - [M]^{-1}[K]q$$

Selon Newmark, on a

$$\begin{bmatrix} 1 & 0 & -\beta\Delta t^2 \\ 0 & 1 & -\gamma\Delta t \\ [K] & 0 & [M] \end{bmatrix} \begin{vmatrix} q_{n+1} \\ \dot{q}_{n+1} \\ \ddot{q}_{n+1} \end{vmatrix} = \begin{bmatrix} 1 & \Delta t & \Delta t^2(0,5-\beta) \\ 0 & 1 & \Delta t(1-\gamma) \\ 0 & 0 & 0 \end{bmatrix} \begin{vmatrix} q_n \\ \dot{q}_n \\ \ddot{q}_n \end{vmatrix} + \begin{bmatrix} 0 \\ 0 \\ [F]_{n+1} \end{bmatrix}$$

$$\Rightarrow \underbrace{\begin{bmatrix} 1 + \beta\Delta t^2[M]^{-1}[K] & 0 \\ \gamma\Delta t[M]^{-1}[K] & 1 \end{bmatrix}}_{[B]} \begin{vmatrix} q_{n+1} \\ \dot{q}_{n+1} \end{vmatrix} = \underbrace{\begin{bmatrix} 1 - \Delta t^2(0,5-\beta)[M]^{-1}[K] & \Delta t \\ -\Delta t(1-\gamma)[M]^{-1}[K] & 1 \end{bmatrix}}_{[C]} \begin{vmatrix} q_n \\ \dot{q}_n \end{vmatrix}$$

$$+ \begin{bmatrix} \beta\Delta t^2[M]^{-1}[F]_{n+1} + \Delta t^2(0,5-\beta)[F]_n \\ \gamma\Delta t[M]^{-1}[F]_{n+1} + \Delta t(1-\gamma)[F]_n \end{bmatrix}$$

$$[A] = [B]^{-1}[C]$$

```
syms delta_t B C

M = m*a^2*[2 1; 1 1];
K = m*g*a*[2 0; 0 1];

beta = 0
gamma = 0.5
B(1:2,1:2) = eye(2)+beta*delta_t^2*inv(M)*K;
B(1:2,3:4) = 0;
B(3:4,1:2) = gamma*delta_t*inv(M)*K;
B(3:4,3:4) = eye(2);

C(1:2,1:2) = eye(2)-delta_t^2*(0.5-beta)*inv(M)*K;
C(1:2,3:4) = delta_t*eye(2);
C(3:4,1:2) = -delta_t*(1-gamma)*inv(M)*K;
C(3:4,3:4) = eye(2);

A = B\C
```

```
beta =

     0


gamma =

    0.5000


A =

[                      1 - (981*delta_t^2)/50,
 (981*delta_t^2)/100,                delta_t,                      0]
[                          (981*delta_t^2)/50,                     1
 - (981*delta_t^2)/50,                      0,               delta_t]
[ (2887083*delta_t^3)/5000 - (981*delta_t)/25,  (981*delta_t)/50
 - (962361*delta_t^3)/2500, 1 - (981*delta_t^2)/50,
 (981*delta_t^2)/100]
[  (981*delta_t)/25 - (962361*delta_t^3)/1250,
 (2887083*delta_t^3)/5000 - (981*delta_t)/25,     (981*delta_t^2)/50,
 1 - (981*delta_t^2)/50]
```

1.2

```
[z,d] = eig(A);
valeurs_propres = simplify(abs(d));
temps_critique = min([vpasolve(valeurs_propres(1,1)-1,delta_t,
[1,0]),vpasolve(valeurs_propres(2,2)-1,delta_t,
[1,0]),vpasolve(valeurs_propres(3,3)-1,delta_t,
[1,0]),vpasolve(valeurs_propres(4,4)-1,delta_t,[1,0])])

% Ici on choisie la plus petite valeur comme le pas de temps critique
 pour la réponse
```

```
temps_critique =

0.132111069327
```

1.3

Comme montré dans 1.1, $\ddot{q} = [M]^{-1}[F] - [M]^{-1}[K]q$, Donc

```
q0 = [0;0]
dq0 = [-1.31519275;-1.85996342]
ddq0 = inv(M)*F(0) - inv(M)*K*q0
```

```
q0 =
```

```
      0
      0


dq0 =

   -1.3152
   -1.8600
```

1.4

C'est déjà montré dans 1.1

1.5 et 1.6

Les valeurs montrées au-dessous sont des valeurs de q(t) , dq(t) et ddq(t) pour les valeurs de t égales à 0s, delta_t, 2*delta_t et 0,5s

```
delta_t = 0.02;
t = t_start:delta_t:0.5;
B_num = eval(B);
A_num = eval(A);
[Q,ddq] = iterate(A_num,t,B_num,delta_t,beta,gamma);

disp('q(0)')
Q(1:2,1)
disp('dq(0)')
Q(3:4,1)
disp('ddq(0)')
ddq(:,1)
disp('q(delta_t)')
Q(1:2,2)
disp('dq(delta_t)')
Q(3:4,2)
disp('ddq(delta_t)')
ddq(:,2)
disp('q(2*delta_t)')
Q(1:2,3)
disp('dq(2*delta_t)')
Q(3:4,3)
disp('ddq(2*delta_t)')
ddq(:,3)
disp('q(0.5)')
Q(1:2,length(t))
disp('dq(0.5)')
Q(3:4,length(t))
disp('ddq(0.5)')
ddq(:,length(t))
```

```
q(0)
```

```
ans =

     0
     0

dq(0)

ans =

   -1.3152
   -1.8600

ddq(0)

ans =

     0
     0

q(delta_t)

ans =

   -0.0261
   -0.0370

dq(delta_t)

ans =

   -1.3122
   -1.8557

ddq(delta_t)

ans =

    1.0301
    1.4687

q(2*delta_t)

ans =

   -0.0517
   -0.0737

dq(2*delta_t)

ans =

   -1.2836
   -1.8235
```

```
ddq(2*delta_t)

ans =

    2.0412
    2.9218

q(0.5)

ans =

    0.0682
    0.0182

dq(0.5)

ans =

    1.3982
    2.3080

ddq(0.5)

ans =

   -2.3208
    1.9636
```

2.1

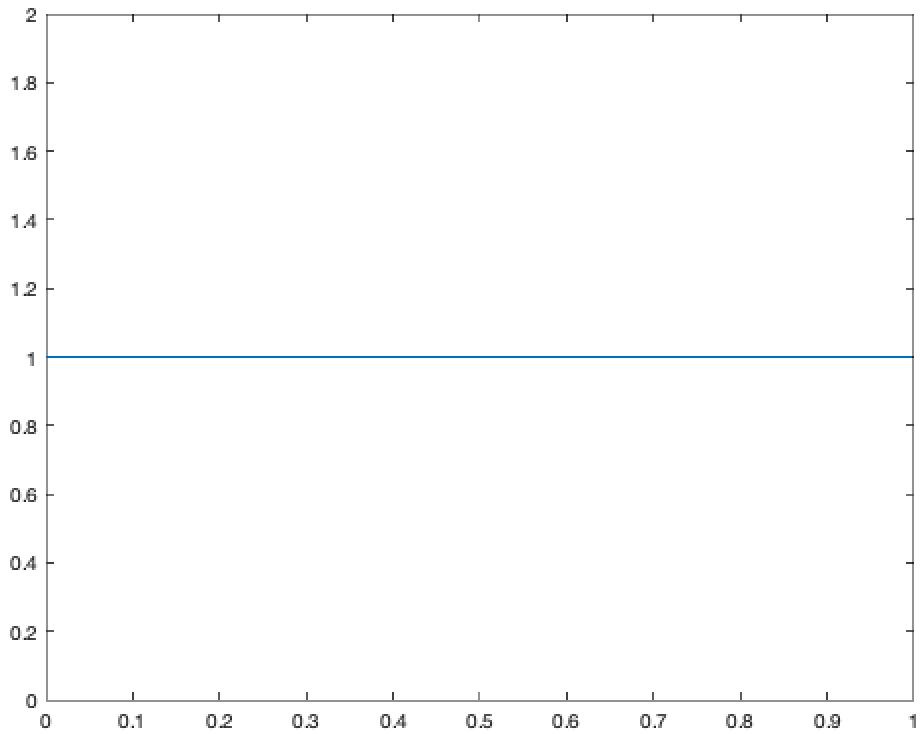
```
clear delta_t
syms delta_t
beta = 0.25
gamma = 0.5
B(1:2,1:2) = eye(2)+beta*delta_t^2*inv(M)*K;
B(1:2,3:4) = 0;
B(3:4,1:2) = gamma*delta_t*inv(M)*K;
B(3:4,3:4) = eye(2);

C(1:2,1:2) = eye(2)-delta_t^2*(0.5-beta)*inv(M)*K;
C(1:2,3:4) = delta_t*eye(2);
C(3:4,1:2) = -delta_t*(1-gamma)*inv(M)*K;
C(3:4,3:4) = eye(2);

A = B\C


beta =

    0.2500
```

```
gamma =

    0.5000


A =

[           -(962361*delta_t^4 - 20000)/(962361*delta_t^4
 + 392400*delta_t^2 + 20000),
 (196200*delta_t^2)/(962361*delta_t^4 + 392400*delta_t^2 +
 20000), (200*delta_t*(981*delta_t^2 + 100))/(962361*delta_t^4 +
 392400*delta_t^2 + 20000),                    (98100*delta_t^3)/
(962361*delta_t^4 + 392400*delta_t^2 + 20000)]
[                    (392400*delta_t^2)/(962361*delta_t^4 +
 392400*delta_t^2 + 20000),           -(962361*delta_t^4 - 20000)/
(962361*delta_t^4 + 392400*delta_t^2 + 20000),
    (196200*delta_t^3)/(962361*delta_t^4 + 392400*delta_t^2 +
 20000), (200*delta_t*(981*delta_t^2 + 100))/(962361*delta_t^4 +
 392400*delta_t^2 + 20000)]
[ -(3924*delta_t*(981*delta_t^2 + 200))/(962361*delta_t^4 +
 392400*delta_t^2 + 20000),                       (392400*delta_t)/
(962361*delta_t^4 + 392400*delta_t^2 + 20000),          -
(962361*delta_t^4 - 20000)/(962361*delta_t^4 + 392400*delta_t^2 +
 20000),                  (196200*delta_t^2)/(962361*delta_t^4 +
 392400*delta_t^2 + 20000)]
[                      (784800*delta_t)/(962361*delta_t^4 +
 392400*delta_t^2 + 20000), -(3924*(981*delta_t^3 + 200*delta_t))/
(962361*delta_t^4 + 392400*delta_t^2 + 20000),
 (392400*delta_t^2)/(962361*delta_t^4 + 392400*delta_t^2 + 20000),
      -(962361*delta_t^4 - 20000)/(962361*delta_t^4 + 392400*delta_t^2
 + 20000)]
```

2.2

On peut voir que la plus grande valeur propre de cette matrice est toujours 1 pour delta_t entre 0 et 1s. Cela veut dire que ce schéma d'intégration est toujours stable pour delta_t entre 0 et 1s. En effet, la schéma Newmark implicite est inconditionnellement stable.

```
delta_ts = 0:0.01:1;
vp = [];
i = 1;
[z,d] = eig(A);
for dt=delta_ts
    d_i = abs(subs(d,delta_t,dt));
    vp(i) = max([d_i(1,1),d_i(2,2),d_i(3,3),d_i(4,4)]);
    i=i+1;
end
plot(delta_ts,vp);
```



2.3

C'est la même que 1.3

```
q0 = [0;0]
dq0 = [-1.31519275;-1.85996342]
ddq0 = inv(M)*F(0) - inv(M)*K*q0


q0 =

     0
     0


dq0 =

   -1.3152
   -1.8600


ddq0 =

     0
     0
```

2.4

Déjà montré dans 1.1

2.5

```
delta_t = 0.02;
t = t_start:delta_t:0.5;
B_num = eval(B);
C_num = eval(C);
A_num = eval(A);
[Q,ddq] = iterate(A_num,t,B_num,delta_t,beta,gamma);
```

2.6

Les valeurs montrées au-dessous sont des valeurs de q(t) , dq(t) et ddq(t) pour les valeurs de t égales à 0s, delta_t, 2*delta_t et 0.5s

```
disp('q(0)')
Q(1:2,1)
disp('dq(0)')
Q(3:4,1)
disp('ddq(0)')
ddq(:,1)
disp('q(delta_t)')
Q(1:2,2)
disp('dq(delta_t)')
Q(3:4,2)
disp('ddq(delta_t)')
ddq(:,2)
disp('q(2*delta_t)')
Q(1:2,3)
disp('dq(2*delta_t)')
Q(3:4,3)
disp('ddq(2*delta_t)')
ddq(:,3)
disp('q(0.5)')
Q(1:2,length(t))
disp('dq(0.5)')
Q(3:4,length(t))
disp('ddq(0.5)')
ddq(:,length(t))
```

```
q(0)

ans =

     0
     0

dq(0)
```

```
ans =

   -1.3152
   -1.8600

ddq(0)

ans =

     0
     0

q(delta_t)

ans =

   -0.0261
   -0.0370

dq(delta_t)

ans =

   -1.3122
   -1.8557

ddq(delta_t)

ans =

    1.0321
    1.4656

q(2*delta_t)

ans =

   -0.0518
   -0.0736

dq(2*delta_t)

ans =

   -1.2835
   -1.8236

ddq(2*delta_t)

ans =

    2.0474
    2.9130
```

```
q(0.5)

ans =

    0.0679
    0.0136

dq(0.5)

ans =

    1.4235
    2.2889

ddq(0.5)

ans =

   -2.3984
    2.1309
```

Fonctions

```
function F = F(t)
    global F0 omega a;
    F = F0 * sin(omega*t) * [a;a/sqrt(2)];
end

function [U,ddq] = iterate(A,t,B,delta_t,beta,gamma)
global M K q0 dq0 ddq0;
i = 0;
U = [];
for t_i=t
    if i==0
        U(:,1) = [q0;dq0];
        ddq(:,1) = ddq0;
    else
        t_i_1 = t_i - delta_t;
        U(:,i+1) =
 A*U(:,i)+inv(B)*[beta*delta_t^2*inv(M)*F(t_i)+delta_t^2*(0.5-
beta)*F(t_i); gamma*delta_t*inv(M)*F(t_i_1)+delta_t*(1-
gamma)*F(t_i_1)];
        ddq(:,i+1) = inv(M)*F(t_i)-inv(M)*K*U(1:2,i+1);
    end
    i=i+1;
end
end


ddq0 =

     0
```

```
0
```

Published with MATLAB® R2019b