DURET Guillaume GR IMA

BURGEVIN Valentin

TP Equation Différentielles Ordinaires :

2018-2019

# EXERCICE N°1 :

## INTRODUCTION :

Nous voulons dans ce TP étudier le problème classique du pendule amorti.

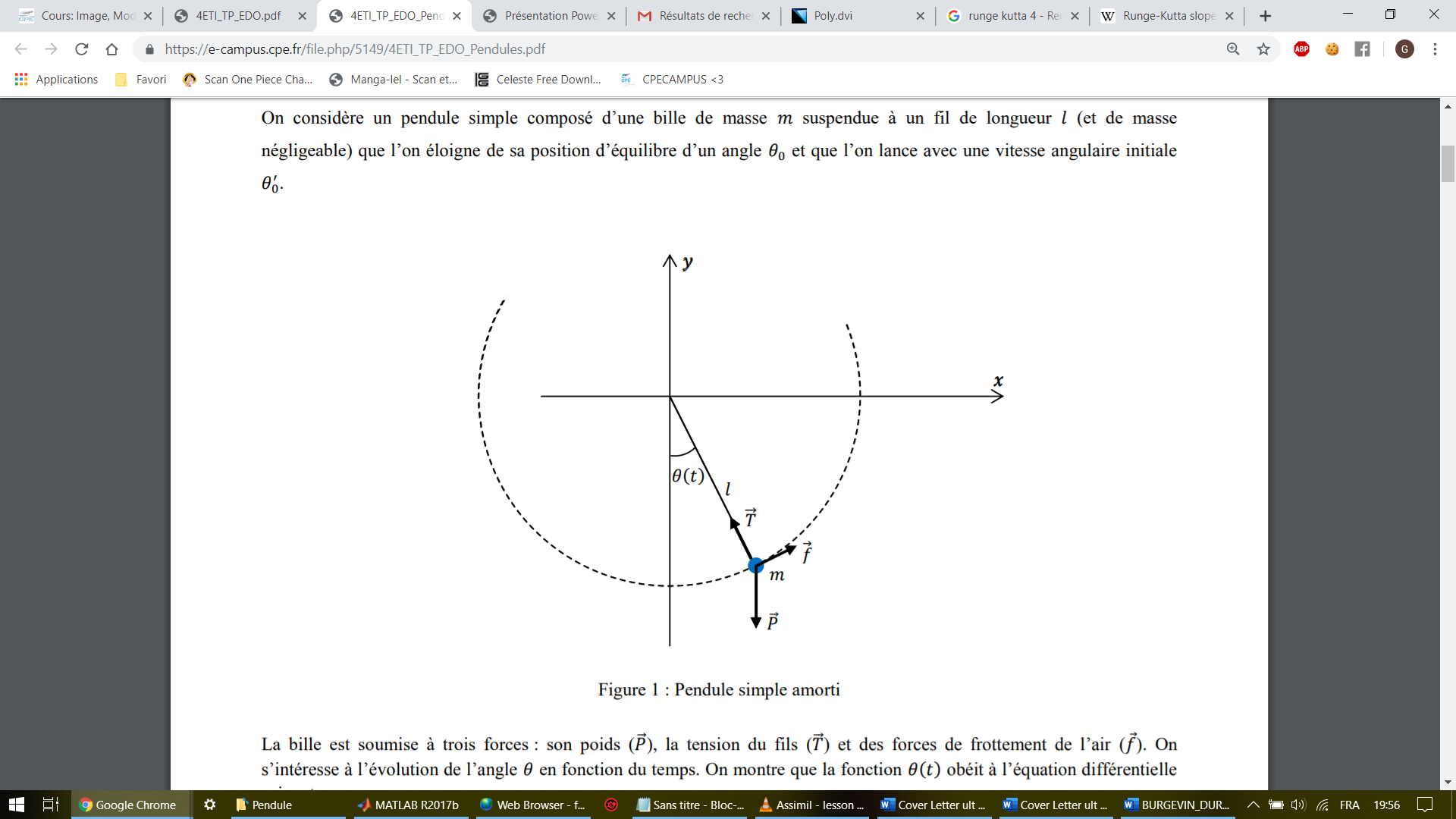


Figure 1: pendule simple amortie

En effet le problème revient à résoudre l’équation :

Avec un coefficient de résistance de l’air et la fréquence propre d’oscillation du pendule.

L’équation précédente peut être obtenue à l’aide d’un principe fondamental de la dynamique dans le repère polaire du pendule.

De plus ce système possède une énergie cinétique égale à :

Obtenue à l’aide de

Et une énergie potentiel égale à :

Cependant en présence d’amortissement (frottement de l’air) l’équation du pendule (1) ne possède pas de solutions exactes. Par conséquent nous allons approcher numériquement la solution de cette équation à l’aide de 3 méthodes qui sont : la méthode d’Euler et les méthodes de Runge-Kutta d’ordre 2 et 4

Or dans le cas du pendule on a une équation à l’ordre 2

Par conséquent on veut se ramener à une équation du premier ordre en 2 dimensions, en posant il apparait le système :

Obtenant donc bien une équation du 1er ordre avec

## La méthode Euler :

La méthode d’Euler a pour base le développement de Taylor à l’ordre 1 :

Pour h tend vers 0.

En appliquant l’équation (E) en tenant compte des conditions initiales on obtient : la fonction d’Euler 2D suivante :

function [x,y,t]=fct\_Euler\_2D(x0,y0,tmin,tmax,h,f,g)

t=tmin:h:tmax;

y=zeros(1,length(t));

x=zeros(1,length(t));

y(1)=y0;

x(1)=x0;

for k=2:length(t)

y(k)=y(k-1)+h\*g(t(k),x(k-1),y(k-1));

x(k)=x(k-1)+h\*f(t(k),x(k-1),y(k-1));

end

end

Celle-ci prend en entrée les conditions initiales (x0 et y0), l’intervalle de temps (tmin et tmax) avec le pas d’échantillonnage du temps et enfin les fonctions f et g correspondant à l’équation différentielle :

L’algorithme est, conformément à la méthode d’Euler, la construction d’une suite qui approxime en fonction du temps la solution de l’équation différentielle donnée.

## Les méthodes Runge-Kutta :

Les méthodes de Range-Kutta ressemblent à la méthode d’Euler à la différence que plutôt d’approximer la solution exacte en prenant la direction de la dérivé, cette fois-ci on utilise :

* Une évaluation supplémentaire de la pente pour l’odre2 au point donc on obtient :

]

Ce qui se met sous la forme suivante sur Matlab :

function [x,y,t]=fct\_RK2\_2D(x0,y0,tmin,tmax,h,beta,f,g)

t=tmin:h:tmax;

y=zeros(1,length(t));

y(1)=y0;

x(1)=x0;

for k=2:length(t)

x(k)=x(k-1)+h\*((1-beta)\*f(t(k),x(k-1),y(k-1))+beta\*f(t(k),x(k-1)+(h/(2\*beta))\*f(t(k),x(k-1),y(k-1)),y(k-1)+((h/(2\*beta))\*g(t(k),t(k-1),y(k-1)))));

y(k)=y(k-1)+h\*((1-beta)\*g(t(k),x(k-1),y(k-1))+beta\*g(t(k),x(k-1)+(h/(2\*beta))\*f(t(k),x(k-1),y(k-1)),y(k-1)+((h/(2\*beta))\*g(t(k),t(k-1),y(k-1)))));

end

De plus il y a pour cette fonction un paramètre qui permet de contrôler à quel moment l’évaluation supplémentaire de la pente se fait (au point ). En effet pour il y a qu’une seule évaluation et on retrouve la méthode d’Euler, pour l’évaluation supplémentaire va être en et une moyenne des pentes sera faite.

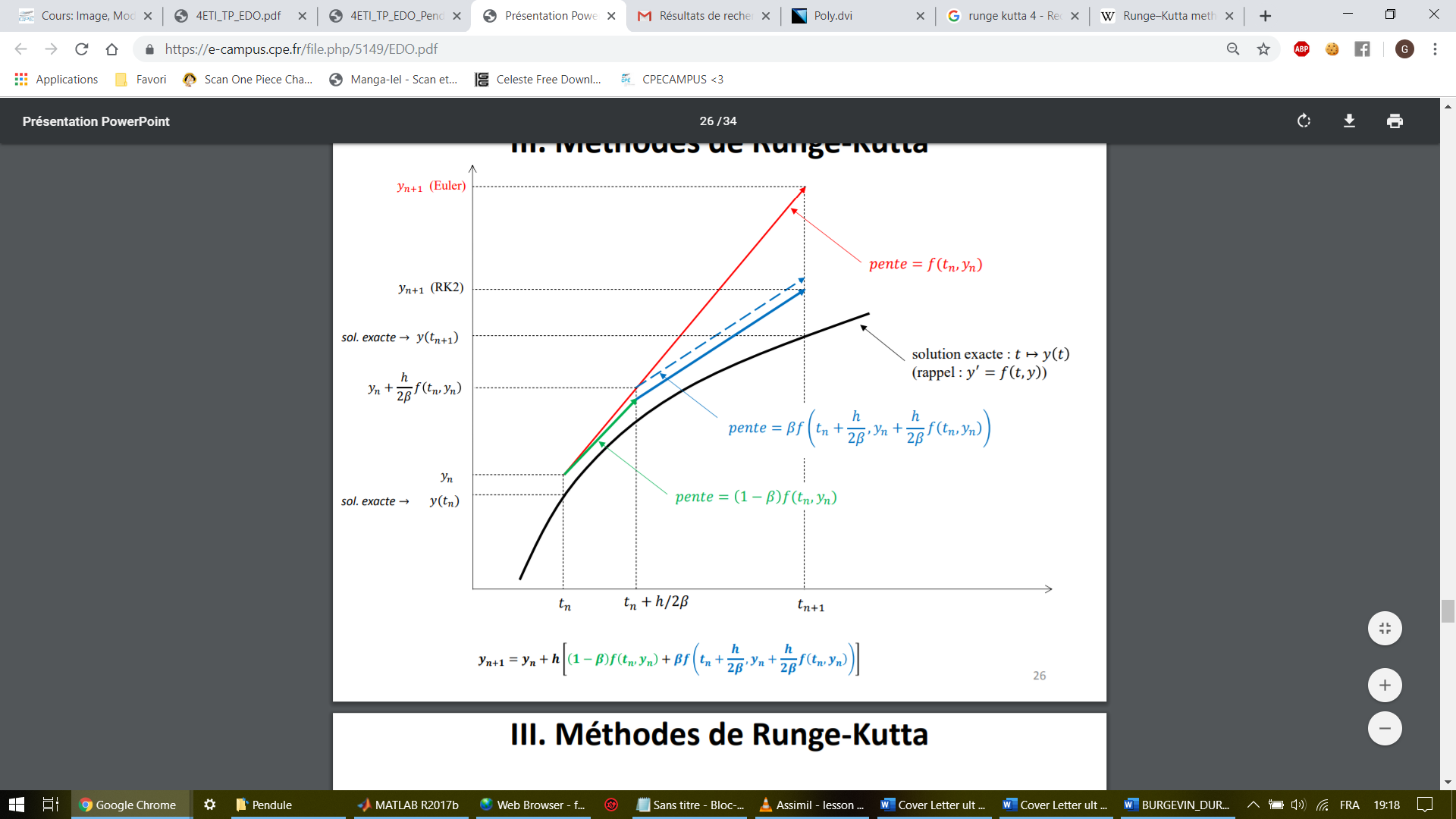


Figure 2 Runge-Kutta ordre 2

* Trois évaluations supplémentaires de la pente pour l’ordre 4 ce qui nous permet d’obtenir :

avec :

On réalise le code suivant :

% mÃ©thode de Runge-Kutta d'ordre 4

function [x,y,t]=fct\_RK4\_2D(x0,y0,tmin,tmax,pas,f,g)

nbIters=floor((tmax-tmin)/pas);

x=zeros(1,nbIters+1);

y=zeros(1,nbIters+1);

t=zeros(1,nbIters+1);

y(1)=y0;

x(1)=x0;

t(1)=tmin;

for k=1:nbIters

k1f=f(t(k),x(k),y(k));

k1g=g(t(k),x(k),y(k));

k2f=f(t(k)+pas/2,x(k)+(pas/2)\*k1f,y(k)+(pas/2)\*k1g);

k2g=g(t(k)+pas/2,x(k)+(pas/2)\*k1f,y(k)+(pas/2)\*k1g);

k3f=f(t(k)+pas/2,x(k)+(pas/2)\*k2f,y(k)+(pas/2)\*k2g);

k3g=g(t(k)+pas/2,x(k)+(pas/2)\*k2f,y(k)+(pas/2)\*k2g);

k4f=f(t(k)+pas,x(k)+pas\*k3f,y(k)+pas\*k3g);

k4g=g(t(k)+pas,x(k)+pas\*k3f,y(k)+pas\*k3g);

y(k+1)=y(k)+(pas/6)\*(k1g+2\*k2g+2\*k3g+k4g);

x(k+1)=x(k)+(pas/6)\*(k1f+2\*k2f+2\*k3f+k4f);

t(k+1)=t(k)+pas;

end

Dans la même idée de pour l’ordre 2, l’ordre 4 va faire une moyenne des 4 évaluations de la pente (voir figure ci-dessous) selon les formules ci-dessus.

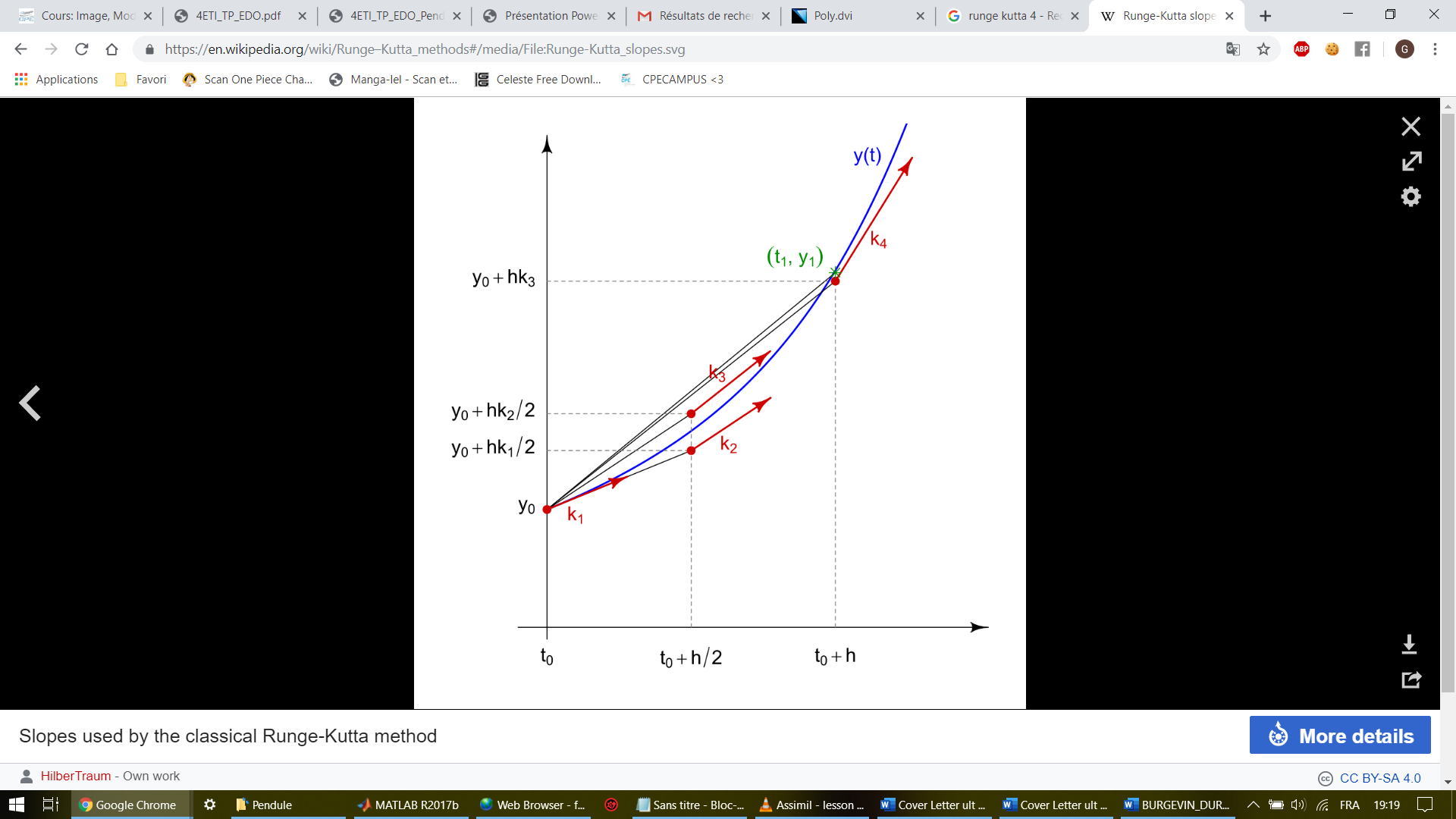


Figure 3 : Runge-Kutta ordre 2

Une fois les fonctions correspondantes faites nous réalisons le code permettant d’introduire les variables physiques et la résolution de l’équation :

clear variables;

close all;

% paramÃ¨tres physiques

m=0.1; % masse de la bille (kg)

r=0.1; % rayon de la bille (m)

eta=0.000018; % coeff. de viscositÃ© de l'air Ã  20Â°C (kg.m^-1.s^-1)

gamma=6\*pi\*r\*eta/m; % frottements (s^-1)

gr=9.8; % accÃ©leration de la pesanteur (m.s^-2)

l=2; % longueur du fil (m)

omega=sqrt(gr/l); % frÃ©quence propre (rad.s^-1)

T0=2\*pi/omega; % (pseudo-)pÃ©riode du pendule (s)

% autres paramÃ¨tres

tmin=0; % instant initialtheta

tmax=4\*T0; % instant final

pas=0.001; % pas de calcul

fprintf('DurÃ©e de l''expÃ©rience physique : %1.2f\n',tmax-tmin);

% fonctions Y'=F(Y) avec ici Y=(theta,z) et F(Y)=(f,g)

f=@(t,theta,z)(z);

g=@(t,theta,z)(-omega^2)\*sin(theta)-gamma\*z;

% conditions initiales

theta0=2\*pi/3; % angle initial (rad)

thetap0=0; % vitesse angulaire initiale (rad/s)

% choix de la mÃ©thode

method=1; % 1 : Euler

% 2 : Runge-Kutta ordre 2

% 3 : Runge-Kutta ordre 4

% calculs numÃ©riques

switch method

case 1

[theta,z,t]=fct\_Euler\_2D(theta0,thetap0,tmin,tmax,pas,f,g);

strTitle='Euler';

case 2

beta=1;

[theta,z,t]=fct\_RK2\_2D(theta0,thetap0,tmin,tmax,pas,beta,f,g);

strTitle='Runge-Kutta ordre 2';

case 3

[theta,z,t]=fct\_RK4\_2D(theta0,thetap0,tmin,tmax,pas,f,g);

strTitle='Runge-Kutta ordre 4';

end

Ce code permet, selon la valeur de la méthode choisie, quelle méthode utiliser.

Une fois que la solution approchée est calculée, on les utilise pour afficher au cours du temps le pendule, l’espace des phases ainsi que les énergies potentiel, cinétique et totale.

% Ã©nergies du pendule

Ec=1/2\*m\*(l\*z).^2; % Ã©nergie cinÃ©tique

Ep=m\*gr\*l\*(1-cos(theta)); % Ã©nergie potentielle

E=Ec+Ep; % Ã©nergie totale

% affichage des rÃ©sultats

figure(1);

xmin=-l;xmax=l;

ymin=-l;ymax=l;

tic;

for k=1:65:length(theta) % rÃ©gler le pas de sorte Ã  obtenir un mouvement rÃ©aliste

% espace rÃ©el

subplot(131);

x=l\*sin(theta(k));

y=-l\*cos(theta(k));

plot([0,x],[0,y],'Marker','o','MarkerFacecolor','r','MarkerSize',10);

axis('equal');

axis(1.1\*[xmin xmax ymin ymax]);

grid 'on';

t1=title(strTitle);

set(t1,'interpreter','latex');

% espace des phases

subplot(132);

hold on;

t3=title('Espace des phases');

set(t3,'interpreter','latex');

h=plot(theta(k),z(k),'ok');

set(h,'MarkerSize',2);

% axis(1.1\*[min(theta),max(theta),min(z),max(z)]);

grid on;

% Ã©nergies

subplot(133);

hold on;

t2=title('Energies');

set(t2,'interpreter','latex');

h=plot(t(k),Ec(k),'ob-',t(k),Ep(k),'om-',t(k),E(k),'or-');

set(h,'MarkerSize',2);

% axis(1.1\*[tmin,tmax,min(Ec),max(Ec)]);

pause(0.0001);

end

cputime=toc;

fprintf('DurÃ©e de la simulation numÃ©rique : %1.2f\n',cputime);

% labels des axes de la figure du portrait de phase

subplot(132);

l1=xlabel('$\theta(t)$','interpreter','latex');

set(l1,'FontSize',14);l2=ylabel('$\theta''(t)$','interpreter','latex');

set(l2,'FontSize',14);

% lÃ©gende de la figure des Ã©nergies

subplot(133);

g1=legend('Energie cinetique','Energie potentielle','Energie totale');

legend('boxoff');set(g1,'interpreter','latex');xlabel('temps','interpreter','latex');

# Résultat :

Pour la méthode d’Euler un pas h de 0.001 la figure suivante est obtenue :

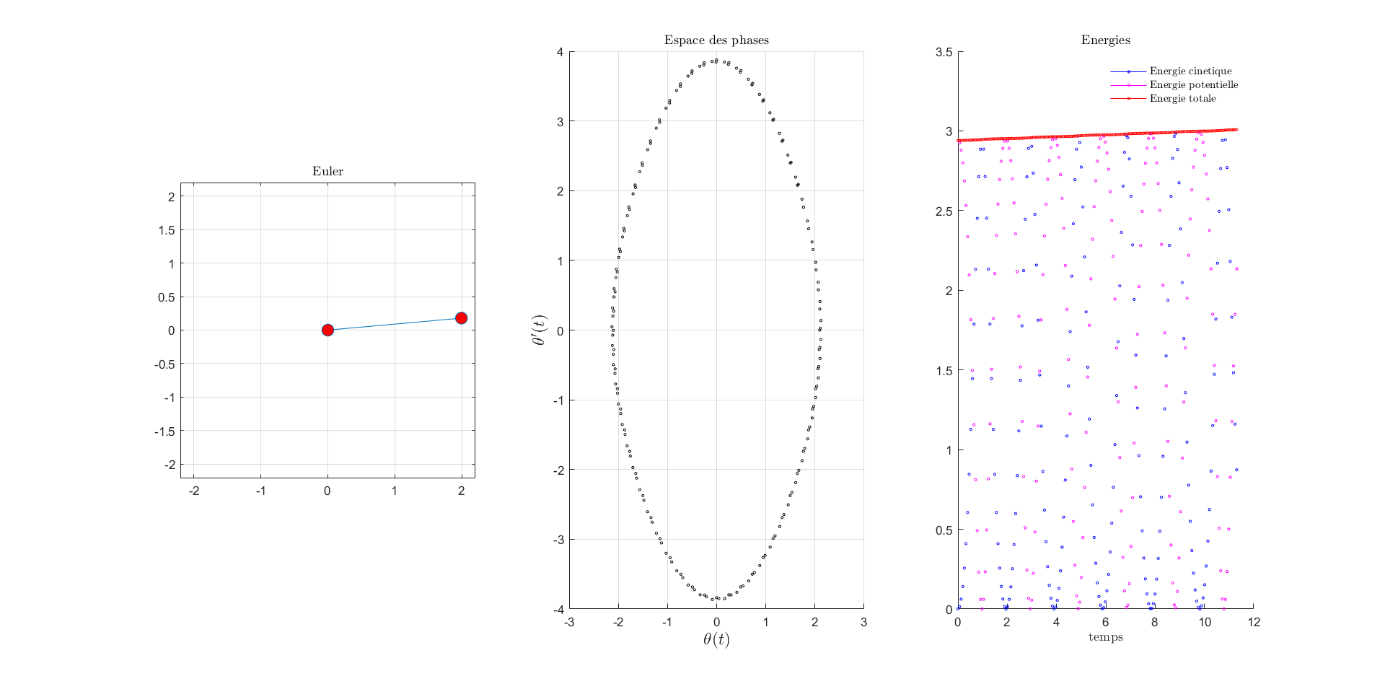


Figure 4 méthode d’Euler h = 0.001

Cette solution approchée obtenue n’est pas satisfaisante car comme on peut le remarquer, le pendule gagne de l’énergies au cours du temps ce qui est physiquement impossible.

Cela s’explique par le fait que la méthode d’Euler n’est pas assez précise et que celle-ci cumule les erreurs au cours du temps : l’erreur d’approximation est supérieure aux frottements que subit le pendule ce qui laisse croire que le pendule gagne de l’énergie.

Cependant avec un pas plus petit tel que h= 0.0001 on obtient la figure suivante :

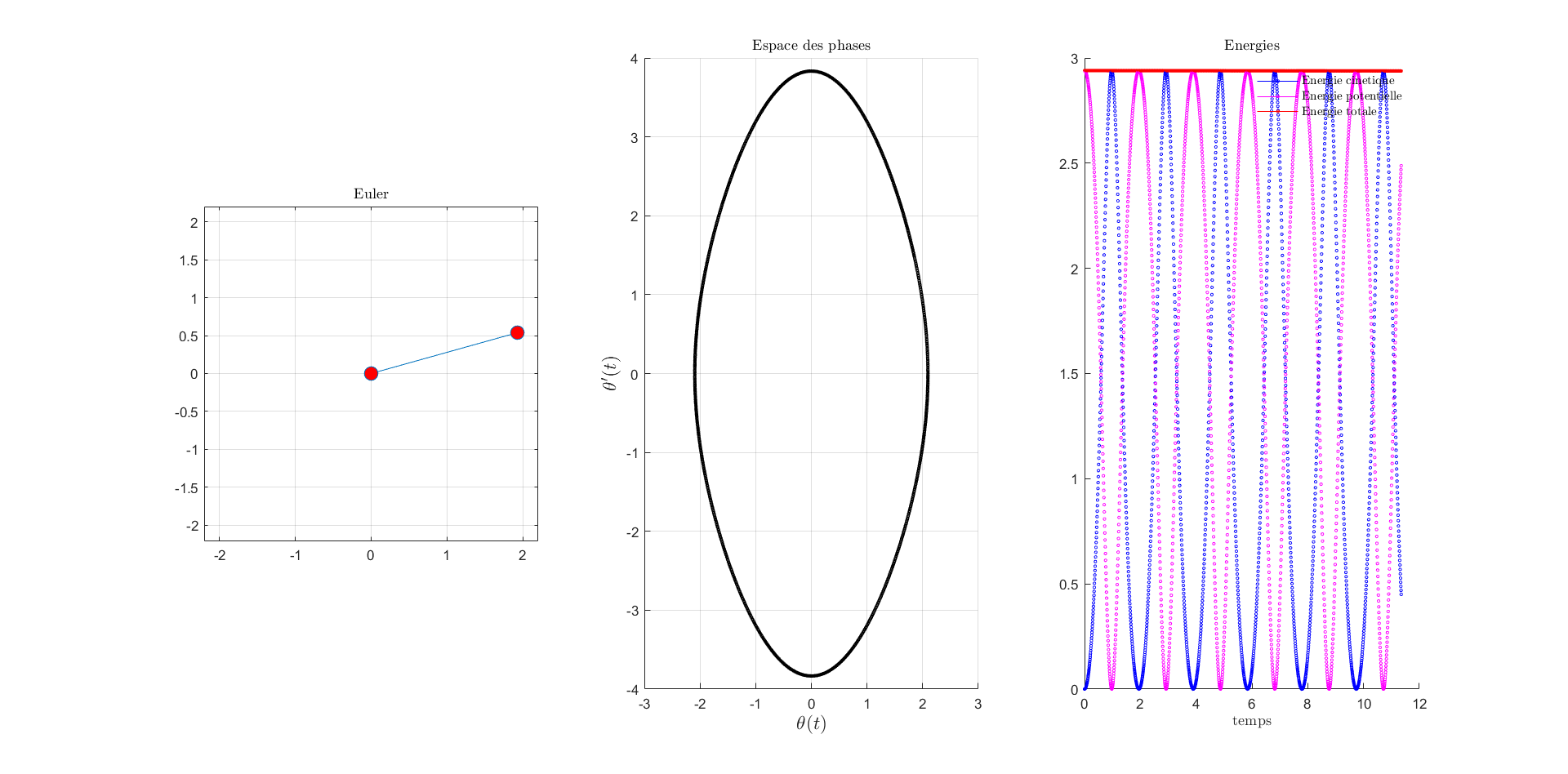


Figure 5 méthode d’Euler h = 0.0001

L’utilisation d’un pas plus petit permet d’être plus proche de la réalité mais le nombre de calculs sera largement supérieur (10x ici).

La différence est flagrante entre les deux figures, cette fois-ci le pendule ne gagne pas en énergie ce qui est bien plus cohérent. Ceci montre bien que c’était bien l’approximation qui faisait monter l’énergie du pendule au cours du temps.

Pour les mêmes paramètres que pour la méthode d’Euler on obtient la figure suivante pour un pas h de 0.001 et pour la méthode de Runge-Kutta d’ordre 2 () :

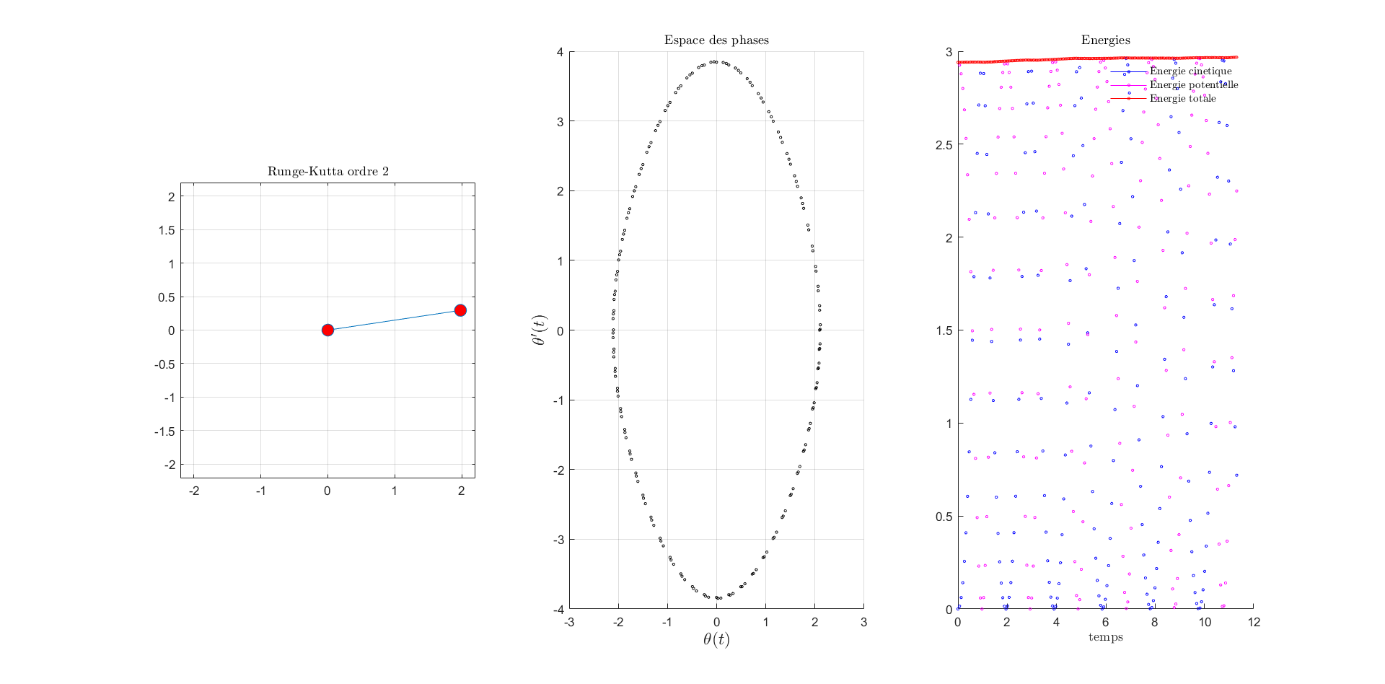


Figure 6 Runge-Kutta ordre 2 h = 0.001

Bien pour cette méthode fasse moins augmenter l’énergie du pendule que la méthode d’Euler, celle-ci l’augmente toujours ce qui rend la méthode non satisfaisante pour ce pas.

Tout comme précédemment on diminue le pas h à 0.0001 pour réduire l’erreur d’approximation :

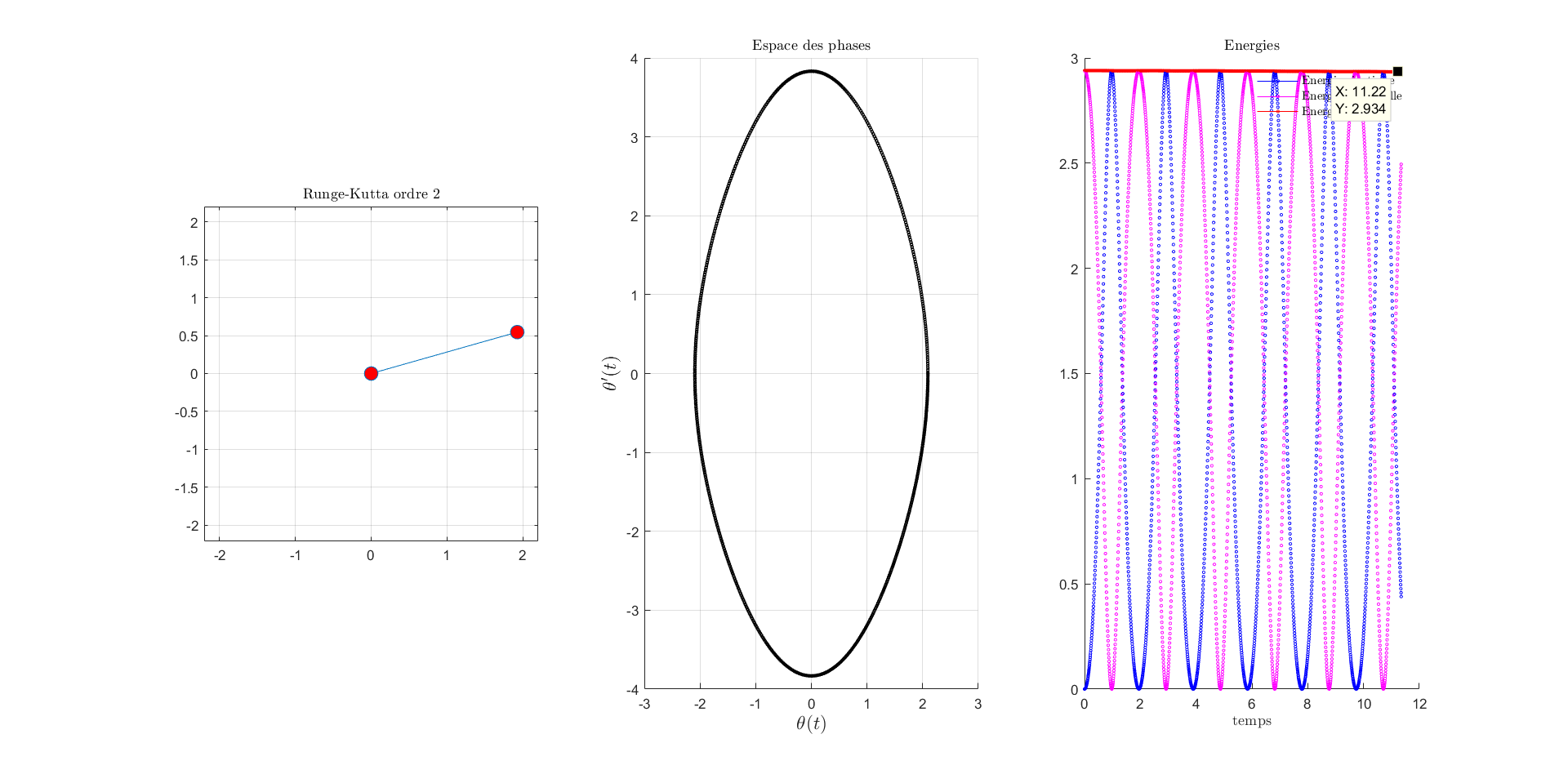


Figure 7 Runge-Kutta ordre 2 h = 0.0001

Le résultat est cette fois-ci cohérent

Enfin on teste la dernière méthode qui est Runge-Kutta à l’ordre 4 pour un pas de 0.001:

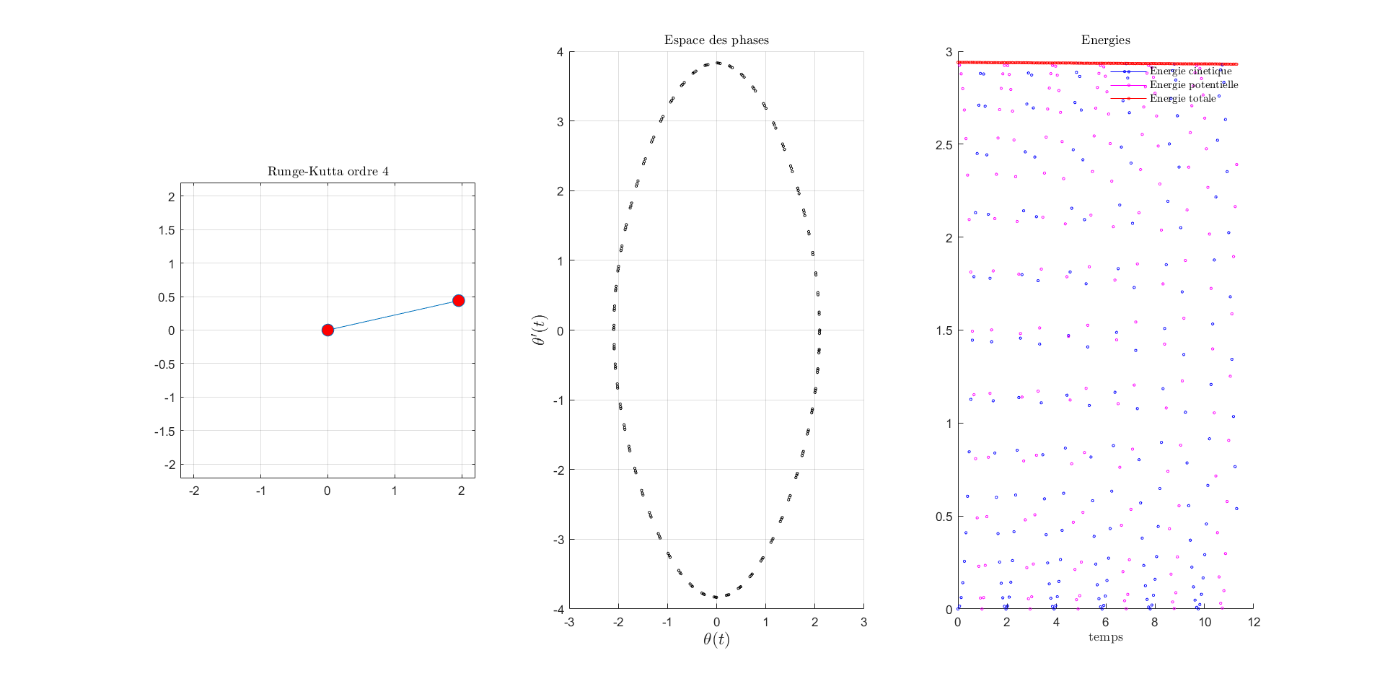


Figure 8 Runge-Kutta ordre 4 h = 0.001

Cette méthode étant la meilleure des 3 mais aussi la plus longue en calcul (4 évaluations de f à chaque pas) elle est aussi la plus précise ce qui permet d’avoir des résultats satisfaisants même avec un pas de 0.001 contrairement aux autres méthodes qui n’avaient pas de sens physique.

On test quand même cette méthode pour un pas h de 0.0001 :

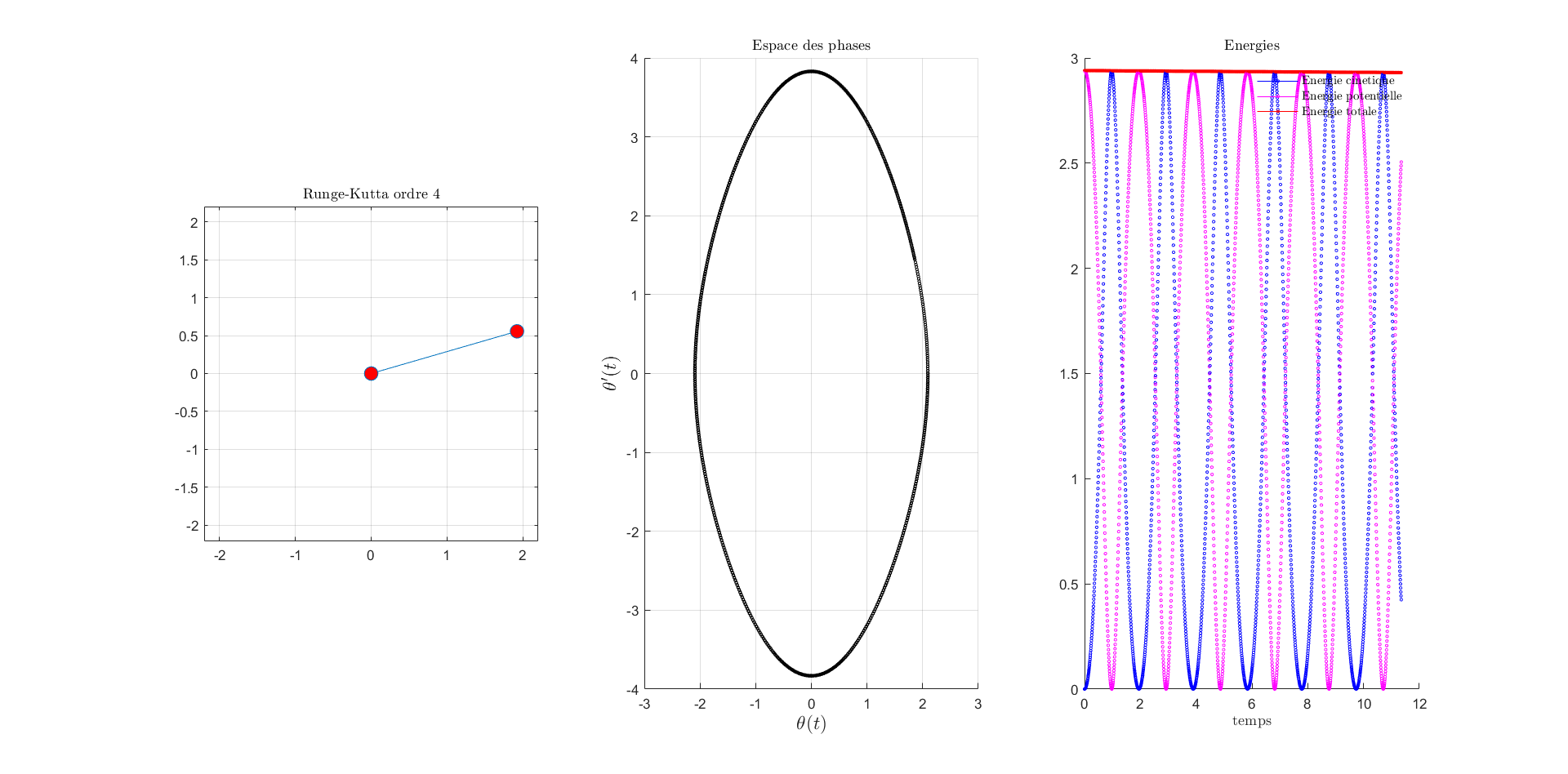


Figure 9 Runge-Kutta ordre 4 h = 0.0001

Le résultat est naturellement toujours bon car le pendule perd de l’énergie à cause des frottements.

On peut aussi faire varier les paramètres physiques pour voir l’influence sur le pendule pour s’assurer de la bonne résolution du problème. Par exemple si on augmente les frottements (eta=0.0018;), on obtient la figure suivante pour un pas de 0.001:

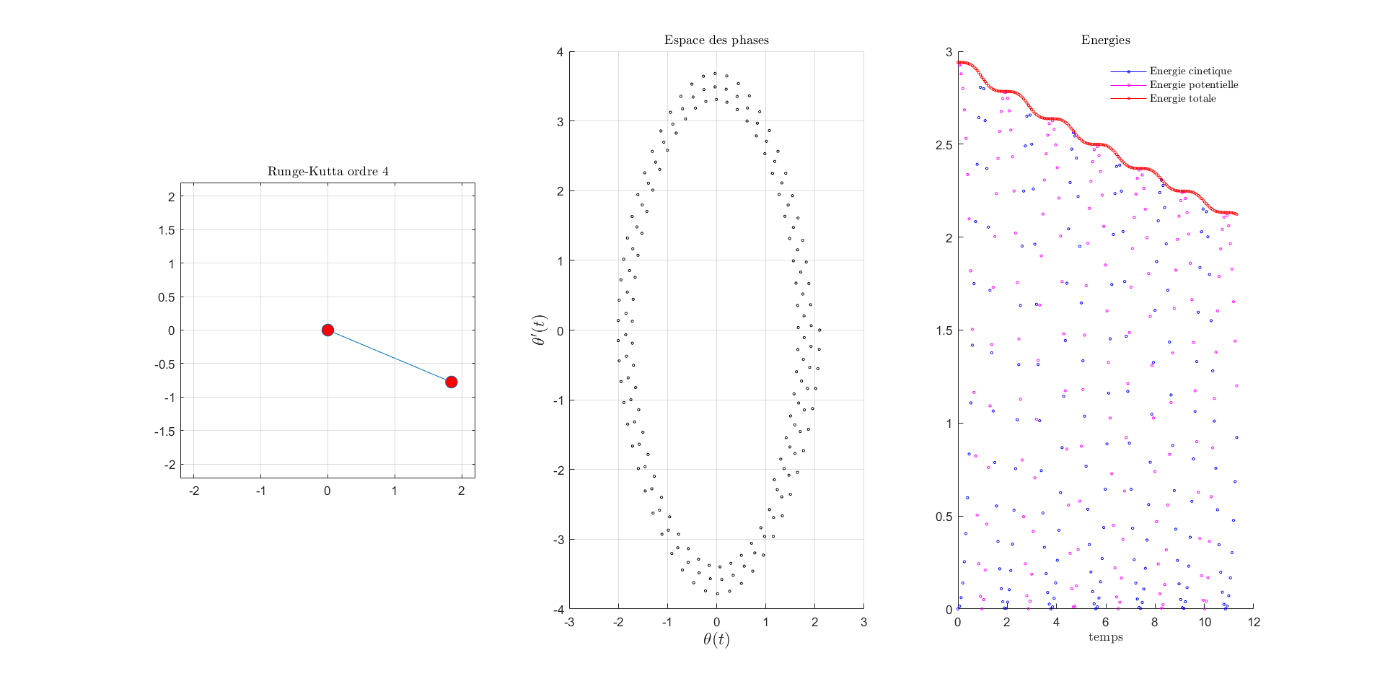


Figure 10 frottements avec Runge-Kutta ordre 4 h = 0.001

Cette figure montre bien que s’il y a plus de frottements, le pendule perd beaucoup plus d’énergie, la solution obtenue semble parfaitement cohérente à la physique

## CONCLUSION :

Nous avons testé 3 méthodes numériques pour approcher la solution de l’équation différentielle d’un pendule amorti. Les méthodes testées sont, du moins précis au plus précis, la méthode d’Euler et les méthodes de Runge-Kutta d’ordre 2 et 4.

Les tests ont montré qu’il faut toujours rester vigilant à la cohérence de la solution, en effet l’équation ne possédant pas de solution analogique nous n’avons pas la certitude que notre solution est bonne ni quelle est l’erreur d’approximation.

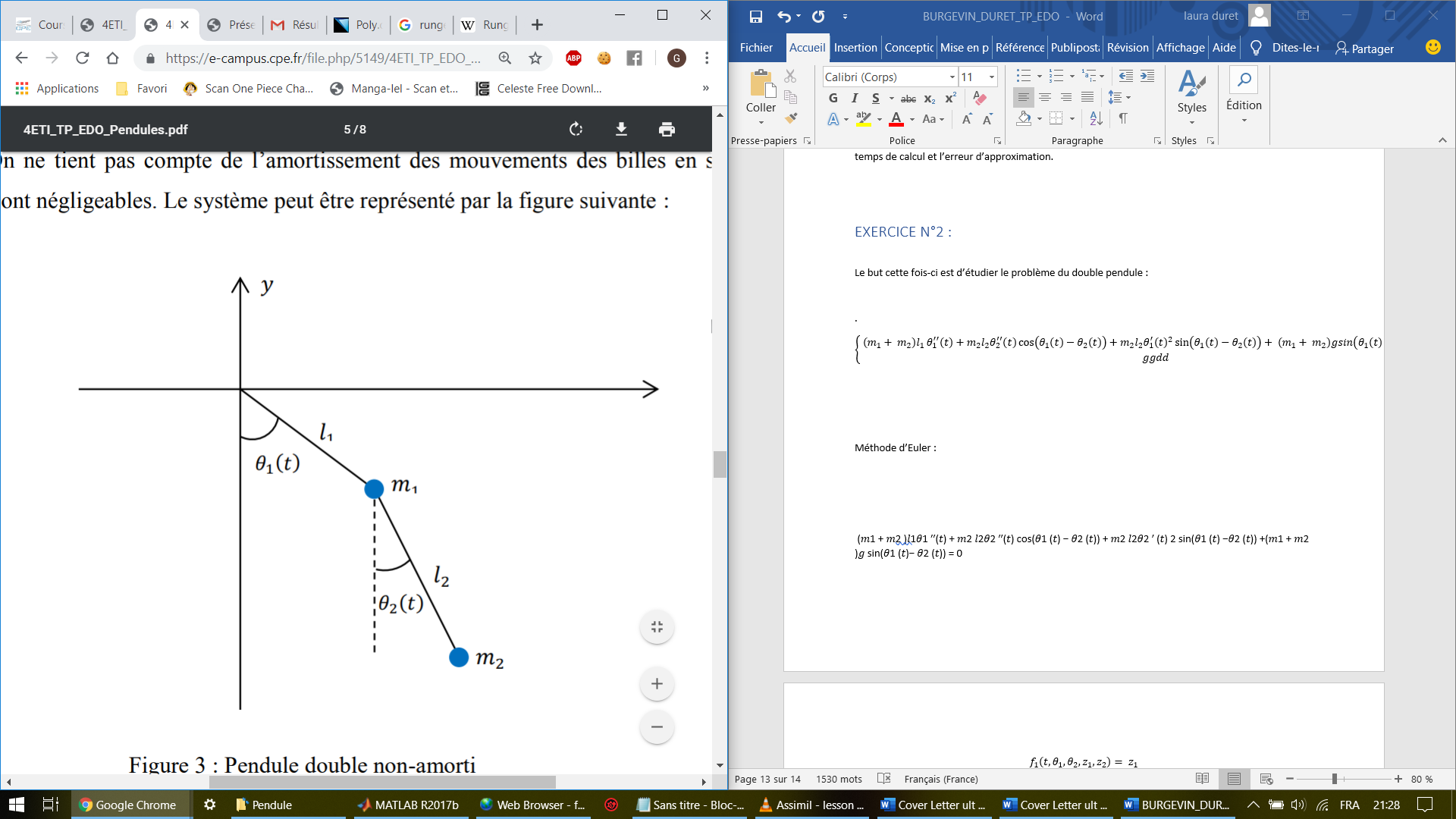
Toutefois les solutions obtenues sont satisfaisantes à la condition que l’on mette un petit pas ce qui rallonge le temps de calcul. Dans un souci d’optimisation il faudrait faire un compromis entre le temps de calcul et l’erreur d’approximation.

# EXERCICE N°2 :

## INTRODUCTION :

## :

Le but cette fois-ci est d’étudier le problème du double pendule :



.

Celui-ci obéit aux équations différentielles suivantes :

Tout comme l’exercice précédent ce système ne possède pas de solutions exactes, nous allons donc une fois de plus avoir recours aux méthodes numériques pour approcher la solution.

Il nous faut d’abord revenir à une équation différentielle d’ordre 1 on pose alors :

Ce qui nous permet d’avoir le système d’ordre 1 de dimension 4 suivant :

Avec :

## METHODE :

Nous allons dans cet exercice directement utiliser la méthode de Runge-Kutta d’ordre 4, cette fois-ci la dimension de l’équation différentielle étant 4 il nous a fallu modifier la fonction de range-Kutta d’ordre 4 en dimension 2 pour obtenir :

function [x,y,z,a,t]=fct\_RK4\_4D(x0,y0,z0,a0,tmin,tmax,pas,f,g,h,i)

nbIters=floor((tmax-tmin)/pas);

x=zeros(1,nbIters+1);

y=zeros(1,nbIters+1);

z=zeros(1,nbIters+1);

a=zeros(1,nbIters+1);

t=zeros(1,nbIters+1);

y(1)=y0;

x(1)=x0;

z(1)=z0;

a(1)=a0;

t(1)=tmin;

for k=1:nbIters

k1f=f(t(k),x(k),y(k),z(k),a(k));

k1g=g(t(k),x(k),y(k),z(k),a(k));

k1h=h(t(k),x(k),y(k),z(k),a(k));

k1i=i(t(k),x(k),y(k),z(k),a(k));

k2f=f(t(k)+pas/2,x(k)+(pas/2)\*k1f,y(k)+(pas/2)\*k1g,z(k)+(pas/2)\*k1h,a(k)+(pas/2)\*k1i);

k2g=g(t(k)+pas/2,x(k)+(pas/2)\*k1f,y(k)+(pas/2)\*k1g,z(k)+(pas/2)\*k1h,a(k)+(pas/2)\*k1i);

k2h=h(t(k)+pas/2,x(k)+(pas/2)\*k1f,y(k)+(pas/2)\*k1g,z(k)+(pas/2)\*k1h,a(k)+(pas/2)\*k1i);

k2i=i(t(k)+pas/2,x(k)+(pas/2)\*k1f,y(k)+(pas/2)\*k1g,z(k)+(pas/2)\*k1h,a(k)+(pas/2)\*k1i);

k3f=f(t(k)+pas/2,x(k)+(pas/2)\*k2f,y(k)+(pas/2)\*k2g,z(k)+(pas/2)\*k2h,a(k)+(pas/2)\*k2i);

k3g=g(t(k)+pas/2,x(k)+(pas/2)\*k2f,y(k)+(pas/2)\*k2g,z(k)+(pas/2)\*k2h,a(k)+(pas/2)\*k2i);

k3h=h(t(k)+pas/2,x(k)+(pas/2)\*k2f,y(k)+(pas/2)\*k2g,z(k)+(pas/2)\*k2h,a(k)+(pas/2)\*k2i);

k3i=i(t(k)+pas/2,x(k)+(pas/2)\*k2f,y(k)+(pas/2)\*k2g,z(k)+(pas/2)\*k2h,a(k)+(pas/2)\*k2i);

k4f=f(t(k)+pas,x(k)+pas\*k3f,y(k)+pas\*k3g,z(k)+(pas/2)\*k3h,a(k)+(pas/2)\*k3i);

k4g=g(t(k)+pas,x(k)+pas\*k3f,y(k)+pas\*k3g,z(k)+(pas/2)\*k3h,a(k)+(pas/2)\*k3i);

k4h=h(t(k)+pas,x(k)+pas\*k3f,y(k)+pas\*k3g,z(k)+(pas/2)\*k3h,a(k)+(pas/2)\*k3i);

k4i=i(t(k)+pas,x(k)+pas\*k3f,y(k)+pas\*k3g,z(k)+(pas/2)\*k3h,a(k)+(pas/2)\*k3i);

y(k+1)=y(k)+(pas/6)\*(k1g+2\*k2g+2\*k3g+k4g);

x(k+1)=x(k)+(pas/6)\*(k1f+2\*k2f+2\*k3f+k4f);

z(k+1)=z(k)+(pas/6)\*(k1h+2\*k2h+2\*k3h+k4h);

a(k+1)=a(k)+(pas/6)\*(k1i+2\*k2i+2\*k3i+k4i);

t(k+1)=t(k)+pas;

end

Une fois cette fonction réalisée nous réalisons le code suivant :

clear variables;

close all;

% paramÃ¨tres physiques

%double pendule 1

m11=0.1; % masse de la bille (kg)

r11=0.1; % rayon de la bille (m)

m21=0.1; % masse de la bille (kg)

r21=0.1; % rayon de la bille (m)

eta1=0.000018; % coeff. de viscositÃ© de l'air Ã  20Â°C (kg.m^-1.s^-1)

gamma11=6\*pi\*r11\*eta1/m11; % frottements (s^-1)

gamma21=6\*pi\*r21\*eta1/m21; % frottements (s^-1)

gr1=9.8; % accÃ©leration de la pesanteur (m.s^-2)

l11=2; % longueur du fil (m)

l21=2; % longueur du fil (m)

omega11=sqrt(gr1/l11); % frÃ©quence propre (rad.s^-1)

omega21=sqrt(gr1/l21); % frÃ©quence propre (rad.s^-1)

T011=2\*pi/omega11; % (pseudo-)pÃ©riode du pendule (s)

T021=2\*pi/omega21; % (pseudo-)pÃ©riode du pendule (s)

%double pendule 2

m12=0.1; % masse de la bille (kg)

r12=0.1; % rayon de la bille (m)

m22=0.1; % masse de la bille (kg)

r22=0.1; % rayon de la bille (m)

eta2=0.000018; % coeff. de viscositÃ© de l'air Ã  20Â°C (kg.m^-1.s^-1)

gamma12=6\*pi\*r12\*eta2/m12; % frottements (s^-1)

gamma22=6\*pi\*r22\*eta2/m22; % frottements (s^-1)

gr2=9.8; % accÃ©leration de la pesanteur (m.s^-2)

l12=2; % longueur du fil (m)

l22=2; % longueur du fil (m)

omega12=sqrt(gr2/l12); % frÃ©quence propre (rad.s^-1)

omega22=sqrt(gr2/l22); % frÃ©quence propre (rad.s^-1)

T012=2\*pi/omega12; % (pseudo-)pÃ©riode du pendule (s)

T022=2\*pi/omega22; % (pseudo-)pÃ©riode du pendule (s)

% conditions initiales

%double pendule 1

theta011=2\*pi/3; % angle initial (rad)

thetap011=0.1; % vitesse angulaire initiale (rad/s)

theta021=2\*pi/3; % angle initial (rad)

thetap021=0+0.1; % vitesse angulaire initiale (rad/s)

%double pendule 2

theta012=2\*pi/3+0.01; % angle initial (rad)

thetap012=0.1; % vitesse angulaire initiale (rad/s)

theta022=2\*pi/3+0.01; % angle initial (rad)

thetap022=0+0.1; % vitesse angulaire initiale (rad/s)

% autres paramÃ¨tres

tmin=0; % instant initial theta

tmax=4\*T011; % instant final

pas=0.001; % pas de calcul

fprintf('DurÃ©e de l''expÃ©rience physique : %1.2f\n',tmax-tmin);

% fonctions Y'=F(Y) avec ici Y=(theta,z) et F(Y)=(f,g)

f1=@(t,theta1,theta2,z1,z2)(z1);

f2=@(t,theta1,theta2,z1,z2)(z2);

g1=@(t,theta1,theta2,z1,z2)-(gr1.\*(2.\*m11+m21).\*sin(theta1)+m21.\*(gr1.\*sin(theta1-2.\*theta2)+2.\*(l21.\*(z2.^2)+l11.\*(z1.^2).\*cos(theta1-theta2)).\*sin(theta1-theta2)))./(2.\*l11.\*(m11+m21.\*((sin(theta1-theta2)).^2)));

g2=@(t,theta1,theta2,z1,z2)sin(theta1-theta2).\*(((m11+m21).\*(l11.\*(z1.^2)+gr1.\*cos(theta1))+l21.\*m21.\*(z2.^2).\*cos(theta1-theta2))./(l21.\*(m11+m21.\*(sin(theta1-theta2)).^2)));

% choix de la mÃ©thode

mod=2; % 1 : simple double pendule % 2 : double double pendule

%double pendule 1

% calculs numÃ©riques

[theta11,theta21,z11,z21,t11]=fct\_RK4\_4D(theta011,thetap011,theta021,thetap021,tmin,tmax,pas,f1,f2,g1,g2);

strTitle21='Runge-Kutta ordre 4';

% Ã©nergies du pendule 1

Ec11=1/2\*m11\*(l11\*z11).^2; % Ã©nergie cinÃ©tique

Ep11=m11\*gr1\*l11\*(1-cos(theta11)); % Ã©nergie potentielle

E11=Ec11+Ep11; % Ã©nergie totale

% Ã©nergies du pendule 2

Ec21=1/2\*m21\*(l21\*z21).^2; % Ã©nergie cinÃ©tique

Ep21=m21\*gr1\*l21\*(1-cos(theta21)); % Ã©nergie potentielle

E21=Ec21+Ep21; % Ã©nergie totale

%double pendule 2

% calculs numÃ©riques

[theta12,theta22,z12,z22,t22]=fct\_RK4\_4D(theta012,thetap012,theta022,thetap022,tmin,tmax,pas,f1,f2,g1,g2);

strTitle22='Runge-Kutta ordre 4';

% Ã©nergies du pendule 1

Ec12=1/2\*m12\*(l12\*z12).^2; % Ã©nergie cinÃ©tique

Ep12=m12\*gr2\*l12\*(1-cos(theta12)); % Ã©nergie potentielle

E12=Ec12+Ep12; % Ã©nergie totale

% Ã©nergies du pendule 2

Ec22=1/2\*m22\*(l22\*z22).^2; % Ã©nergie cinÃ©tique

Ep22=m22\*gr2\*l22\*(1-cos(theta22)); % Ã©nergie potentielle

E22=Ec22+Ep22; % Ã©nergie totale

% affichage des rÃ©sultats

xmin=-l12-l22;xmax=l12+l22;

ymin=-l12-l22;ymax=l12+l22;

tic;

%affichage du pendule 1

for k=1:65:length(theta11) % rÃ©gler le pas de sorte Ã  obtenir un mouvement rÃ©aliste

% espace rÃ©el

figure(1);

if (mod==2)

subplot(231);

else

subplot(131);

end

x1=l11\*sin(theta11(k));

y1=-l11\*cos(theta11(k));

x2=l21\*sin(theta21(k));

y2=-l21\*cos(theta21(k));

plot([0,x1],[0,y1],'Marker','o','MarkerFacecolor','r','MarkerSize',10);

hold on;

plot([x1,x2],[y1,y2],'Marker','o','MarkerFacecolor','r','MarkerSize',10);

hold off;

axis('equal');

axis(1.1\*[xmin xmax ymin ymax]);

grid 'on';

t1=title(strTitle21);

set(t1,'interpreter','latex');

% espace des phases

if mod==2

subplot(232);

else

subplot(132);

end

hold on;

t3=title('Espace des phases');

set(t3,'interpreter','latex');

h=plot(theta11(k),z11(k),'ok');

set(h,'MarkerSize',2);

axis(1.1\*[min(theta11),max(theta11),min(z11),max(z11)]);

grid on;

% Ã©nergies

if mod==2

subplot(233);

else

subplot(133);

end

hold on;

t2=title('Energies');

set(t2,'interpreter','latex');

h=plot(t11(k),Ec11(k),'ob-',t11(k),Ep11(k),'om-',t11(k),E11(k),'or-');

set(h,'MarkerSize',2);

axis(1.1\*[tmin,tmax,min(Ec11),max(Ec11)]);

%pause(0.000001);

if mod==2

subplot(234);

x1=l12\*sin(theta12(k));

y1=-l12\*cos(theta12(k));

x2=l22\*sin(theta22(k));

y2=-l22\*cos(theta22(k));

plot([0,x1],[0,y1],'Marker','o','MarkerFacecolor','r','MarkerSize',10);

hold on;

plot([x1,x2],[y1,y2],'Marker','o','MarkerFacecolor','r','MarkerSize',10);

hold off;

axis('equal');

axis(1.1\*[xmin xmax ymin ymax]);

grid 'on';

t1=title(strTitle22);

set(t1,'interpreter','latex');

end

% espace des phases

if mod==2

subplot(235);

hold on;

t3=title('Espace des phases');

set(t3,'interpreter','latex');

h=plot(theta12(k),z12(k),'ok');

set(h,'MarkerSize',2);

axis(1.1\*[min(theta12),max(theta12),min(z12),max(z12)]);

grid on;

end

% Ã©nergies

if mod==2

subplot(236);

hold on;

t2=title('Energies');

set(t2,'interpreter','latex');

h=plot(t22(k),Ec12(k),'ob-',t22(k),Ep12(k),'om-',t22(k),E12(k),'or-');

set(h,'MarkerSize',2);

axis(1.1\*[tmin,tmax,min(Ec12),max(Ec12)]);

end

end

cputime=toc;

fprintf('DurÃ©e de la simulation numÃ©rique : %1.2f\n',cputime);

% labels des axes de la figure du portrait de phase

if mod==2

subplot(232);

else

subplot(132);

end

l11=xlabel('$\theta(t)$','interpreter','latex');

set(l11,'FontSize',14);l21=ylabel('$\theta''(t)$','interpreter','latex');

set(l21,'FontSize',14);

% lÃ©gende de la figure des Ã©nergies

if mod==2

subplot(233);

else

subplot(133);

end

g1=legend('Energie cinetique','Energie potentielle','Energie totale');

legend('boxoff');set(g1,'interpreter','latex');xlabel('temps','interpreter','latex');

if mod ==2

% labels des axes de la figure du portrait de phase

subplot(235);

l11=xlabel('$\theta(t)$','interpreter','latex');

set(l11,'FontSize',14);l21=ylabel('$\theta''(t)$','interpreter','latex');

set(l21,'FontSize',14);

% lÃ©gende de la figure des Ã©nergies

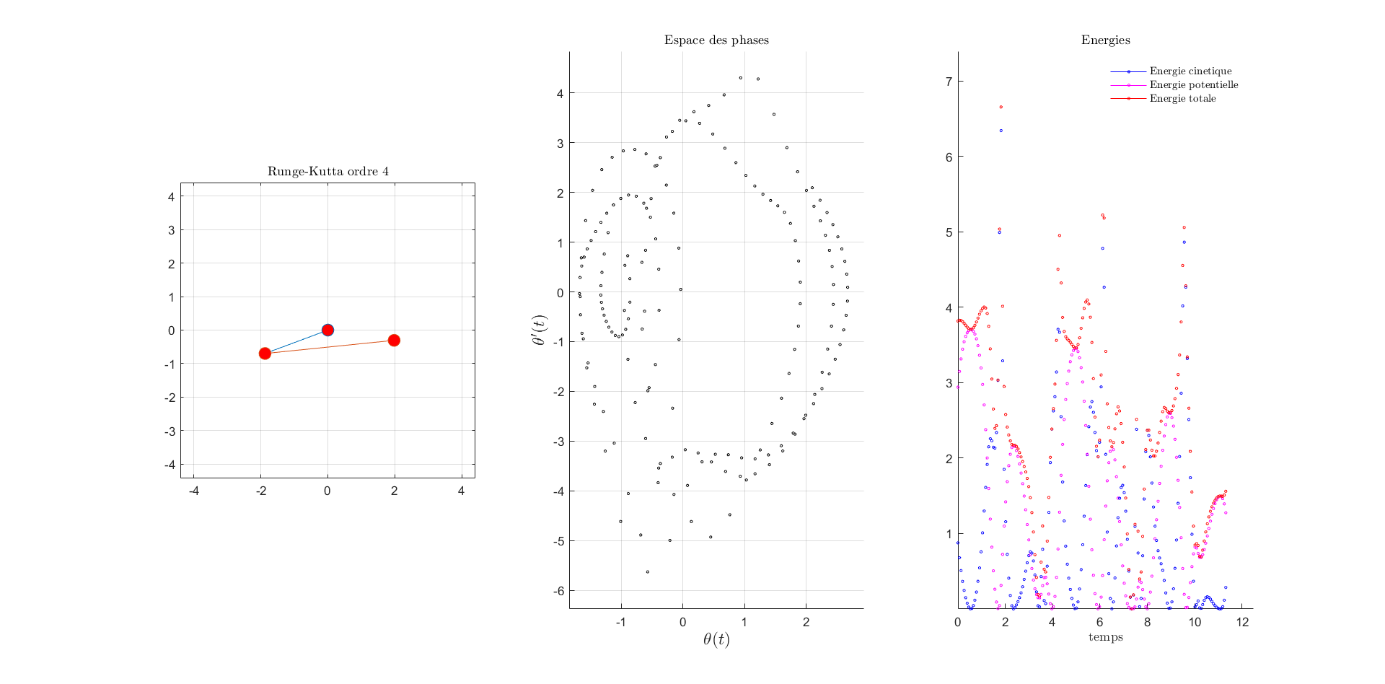
subplot(236);

g1=legend('Energie cinetique','Energie potentielle','Energie totale');

legend('boxoff');set(g1,'interpreter','latex');xlabel('temps','interpreter','latex');

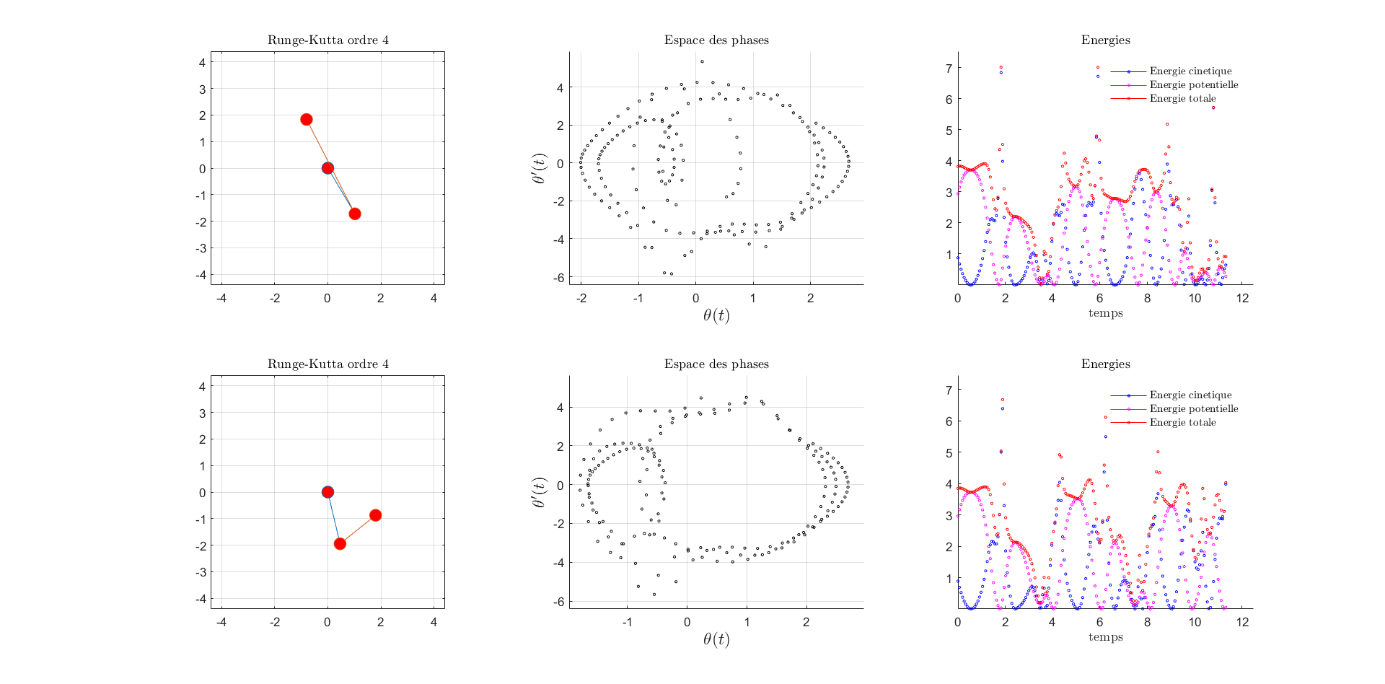
end

Celui-ci est semblable au code du 1er exercice mais il met en œuvre les équations citées en introduction. En effet on affiche aussi pour chaque double pendule l’espace des phases, ainsi que l’énergie cinétique, potentiel et total du 1er pendule (celui qui est au-dessus).



De plus la gestion de la variable mod a pour conséquence d’afficher soit 1 pendule double soit 2

En mettant une différence d’angle de 0.01 entre deux pendules on remarque que les pendules sont au début semblables mais après quelques secondes ils ont un comportement totalement différant :



Ici on peut facilement remarquer dans l’espace des phases que le comportement des deux pendules est différant. De plus si on regarde l’énergie on remarque que les courbes sont beaucoup plus chaotiques que pour le simple pendule.

## CONCLUSION :

Dans cette partie nous avons résolu un problème dont les équations différentielles était plutôt compliqué qu’il serait très compliqué de résoudre à la main voire impossible.

Cependant la résolution numérique a encore une fois été concluante, en effet visuellement le double pendule semble être réaliste (sans frottement). De plus le double pendule est considéré comme chaotique, c’est-à-dire que pour une position initiale quasiment identique (dans la réalité il est impossible que deux expériences aient rigoureusement les même conditions initiales) le comportement de celui-ci sera pas du tout le même.

La solution approchée que nous avons obtenue décrit parfaitement ce phénomène se qui accentue encore l’efficacité de la résolution numérique.