MODÉLISATION MATHÉMATIQUE DES SYSTÈMES MULTIPHYSIQUES
Intégration géométrique de systèmes hamiltoniens d'ordre 4 et extension à un système irréversible

Hèdi LEHLALI

MAM 3A

Polytech Lyon
Unversité Claude Bernard - Lyon 1

Table des matières

1 Systèmes Hamiltoniens et calculs 3
1.1 Calculs préliminaires 3

2 Méthode d'Euler Explicite 4
2.1 Méthode Euler Explicite (Matlab) 4
2.2 Les invariants dynamiques 5
2.3 Diagramme de phase 6

3 Méthode des flots intégrables 6
3.1 Méthode flots intégrables (Matlab) 6
3.2 Les invariants dynamiques 10
3.3 Diagramme de phase 10

4 Comparaison des méthodes 11
4.1 Comparaison des invariants de Casimir 11
4.2 Comparaison diagramme de phase 11
4.3 Comparaison invariant H_0 11

5 Annexe (Code Matlab) 11
5.1 Script Matlab pour la méthode Euler explicite 11
5.2 Script Matlab pour la méthode Flots Intégrables 12

L'objectif de ce TD est de développer et comparer des méthodes d'intégration des systèmes hamiltoniens définis par rapport à une matrice de structure de Poisson non-symplectique puis pour un systèmpe irréversible dont la matrice de structure antisymétrique ne vérifie pas les identités de Jacobi.
Dans cette partie, nous allons réaliser différents méthodes d'intégration numériques sur le système LC.

1 Systèmes Hamiltoniens et calculs

1.1 Calculs préliminaires

On considère l'espace d'état $\mathcal{X} = \mathbf{R}^n$ avec $x \in \mathbf{R}^n$ et un intervalle de temps $[0, T]$ avec $t \in [0, T]$ et $T > 0$. On considère aussi l'équation différentielle ordinaire d'ordre n : $\frac{dx}{dt} = f(x)$

avec $f(x) = \begin{pmatrix} f_1(x) \\ \vdots \\ f_n(x) \end{pmatrix}$ un champ de vecteur où les fonctions $f_i(x)$ sont à valeurs réelles et C^∞.

Ici, $f(x) = \frac{dx}{dt}$, or $\frac{dx}{dt} = J(x)\frac{\partial H_0}{\partial x}$ donc $f(x) = J(x)\frac{\partial H_0}{\partial x}$.

Avec $x = \begin{pmatrix} \phi_{L_1} \\ \phi_{L_2} \\ \phi_{L_3} \\ Q_C \end{pmatrix}$, $J(x) = \begin{pmatrix} 0 & 0 & 0 & B \\ 0 & 0 & 0 & A \\ 0 & 0 & 0 & 1 \\ -B & -A & -1 & 0 \end{pmatrix}$ et $H_0(x) = \frac{\phi_{L_1}^2}{2L_1} + \frac{\phi_{L_2}^2}{2L_2} + \frac{\phi_{L_3}^2}{2L_3} + \frac{Q_C^2}{2C}$

Ainsi on a :

$$\frac{\partial H_0}{\partial x} = \begin{pmatrix} \frac{\phi_{L_1}}{L_1} \\ \frac{\phi_{L_2}}{L_2} \\ \frac{\phi_{L_3}}{L_3} \\ \frac{Q_C}{C} \end{pmatrix}$$

Ainsi on peut écrire :

$$\begin{aligned} f(x) &= \begin{pmatrix} 0 & 0 & 0 & B \\ 0 & 0 & 0 & A \\ 0 & 0 & 0 & 1 \\ -B & -A & -1 & 0 \end{pmatrix} \begin{pmatrix} \frac{\phi_{L_1}}{L_1} \\ \frac{\phi_{L_2}}{L_2} \\ \frac{\phi_{L_3}}{L_3} \\ \frac{Q_C}{C} \end{pmatrix} \\ &= \begin{pmatrix} \frac{B \times Q_C}{C} \\ \frac{A \times Q_C}{C} \\ \frac{Q_C}{C} \\ -B\frac{\phi_{L_1}}{L_1} - A\frac{\phi_{L_2}}{L_2} - \frac{\phi_{L_3}}{L_3} \end{pmatrix} \end{aligned}$$

où A et B sont les coefficients des transformateurs.

2 Méthode d'Euler Explicite

2.1 Méthode Euler Explicite (Matlab)

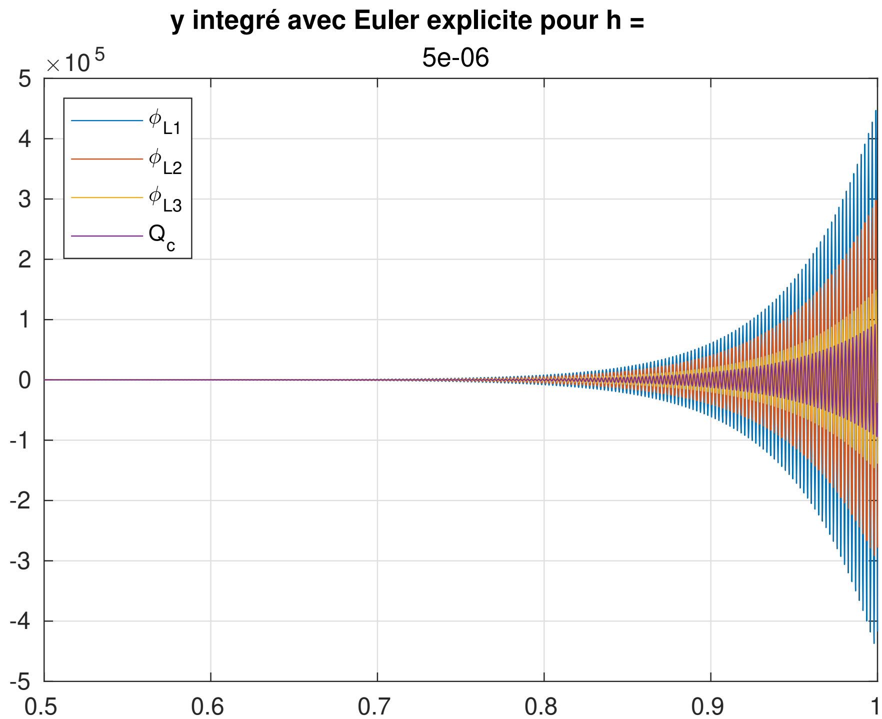
On obtient la fonction grâce au code (f.m) suivant avec la méthode Euler Explicite (EEx) :

```
function y = f(x)
% x est un vecteur de taille 4

y=zeros(4,1); % initialisation de y

C = 2.2e-4;
L1 = 1.8e-2;
L2 = 0.8e-2;
L3 = 1.3e-3;

A = 2;
B = 3;

%ici on définit f comme le produit de J avec dH0 :

y(1) = B*x(4)/C;
y(2) = A*x(4)/C;
y(3) = x(4)/C;
y(4) = -B*x(1)/L1 - A*x(2)/L2 - x(3)/L3;
end
```

et pour afficher les figures on code dans un fichier test.m :

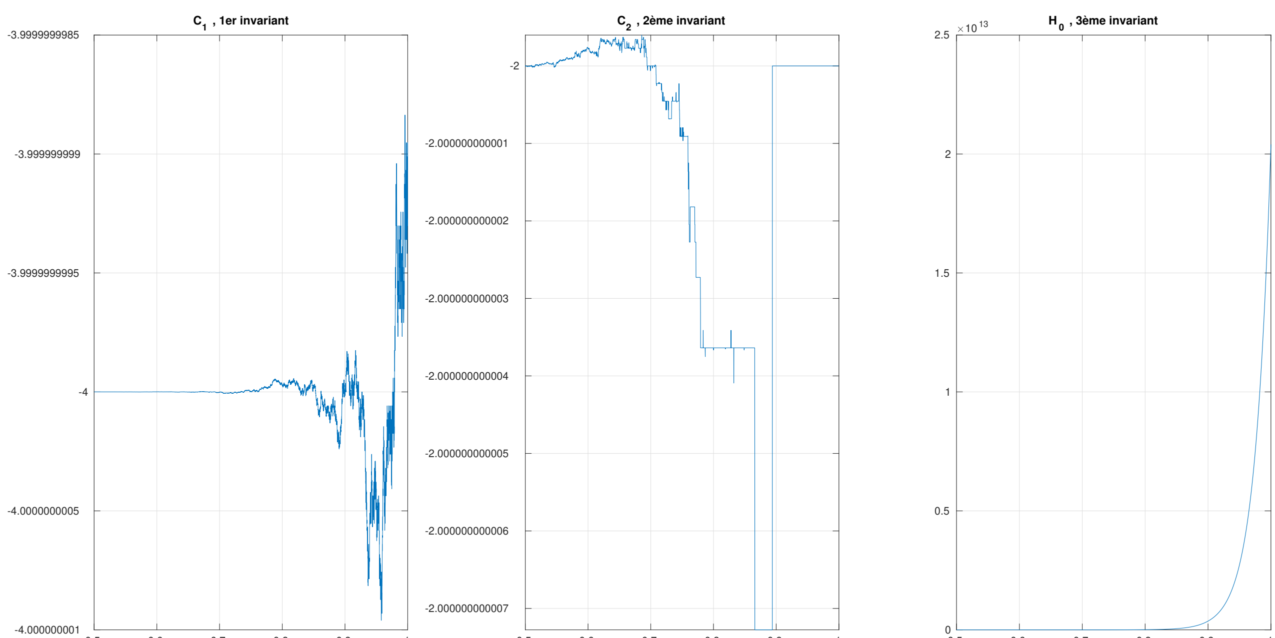
```
Yinit = [2,2,2,4]; % on prend x0 qui respecte Casimir
Xr = [0.5,1];

N = 1000;
%t = 1:N+1;
h = 5e-6;

[t,y] = EEx(Xr,Yinit,h);


%% On trace maintenant l'ensemble des variables selon le temps :
figure;

plot(t,y(1,:), t,y(2,:), t,y(3,:), t,y(4,:));
grid on;
legend('\phi_{L1}', '\phi_{L2}', '\phi_{L3}', 'Q_c', 'Location', '
    NorthWest');
title('y integré avec Euler explicite pour h = ', h);
```

On obtient la figure suivante :



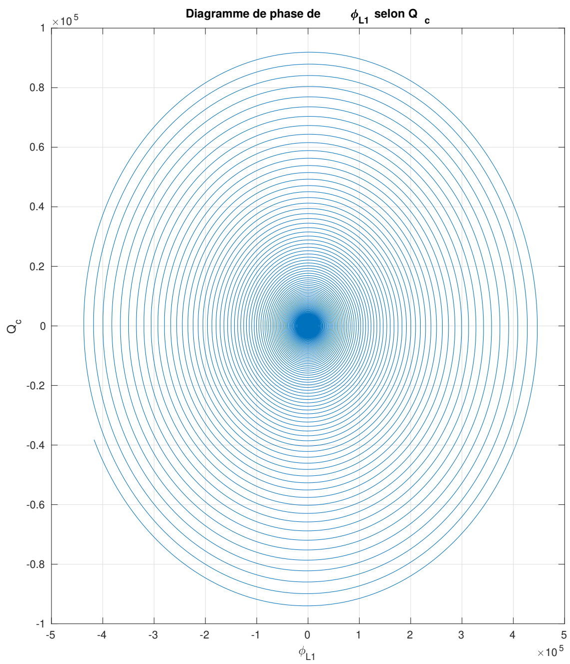
2.2 Les invariants dynamiques

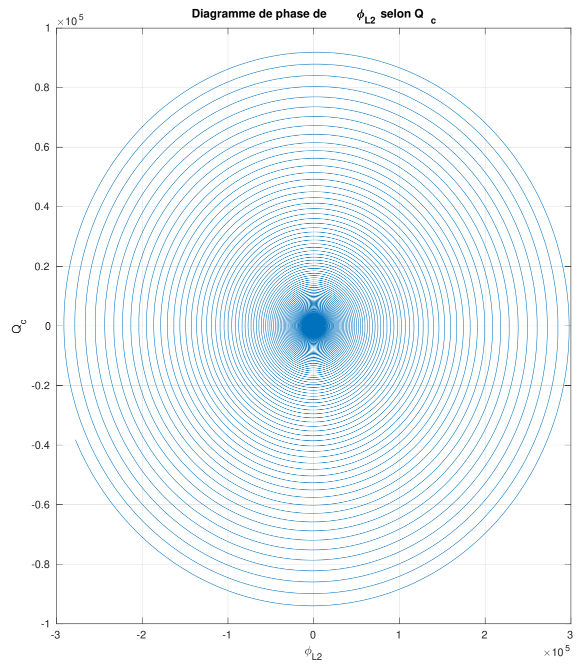
On plot les invariants et on obtient :



2.3 Diagramme de phase

Et pour obtenir le diagramme des phases pour la méthodes Euler Explicite on a les figures suivantes :





3 Méthode des flots intégrables

3.1 Méthode flots intégrables (Matlab)

On rappelle qu'on a : $H_0(x) = \frac{\phi_{L_1}^2}{2L_1} + \frac{\phi_{L_2}^2}{2L_2} + \frac{\phi_{L_3}^2}{2L_3} + \frac{Q_C^2}{2C}$ et les paramètres fixés :

$$\begin{cases} C = 2,2 \times 10^{-4}[F] \\ L_1 = 1,8 \times 10^{-2}[H] \\ L_2 = 0,8 \times 10^{-}2[H] \\ L_3 = 1,3 \times 10^{-3}[H] \end{cases}$$

On voit que H_0 est séparable et on peut donc écrire :

$$H_0^{[1]} = \frac{x_1^2}{2L_1} + \frac{x_2^2}{2L_1} + \frac{x_3^2}{2L_3}$$
$$H_0^{[2]} = \frac{x_4^2}{2C}$$

On code dans des fichiers séparés les fonctions dérivées suivantes :

```
function [y] = dH0_a(x)
L1=1.8*10^(-2);
y=x/L1;
end

function [y] = dH0_b(x)
L2=0.8*10^(-2);
y=x/L2;
end

function [y] = dH0_c(x)
L3=1.3*10^(-3);
y=x/L3;
end

function [y] = dH0_1(x)
y=dH0_a(x)+dH0_b(x)+dH0_c(x);
end

function [y] = dH0_2(x)
C=2.2*10^(-4);
y=x/C;
end
```

On rappelle qu'on a :

(1) l'application de $\varphi_{h/2}^{[2]}$

$$\phi\left(k+\frac{1}{2}\right)=\begin{pmatrix} B \\ A \\ 1 \end{pmatrix}\frac{\partial H_0^{[2]}}{\partial Q}(Q(k))h+\phi(k)$$

(2) l'application de $\varphi_{h}^{[1]}$

$$Q(k+1)=\begin{pmatrix} -B & -A & -1 \end{pmatrix}\frac{\partial H_0^{[2]}}{\partial \phi}\phi\left(k+\frac{1}{2}\right)h+Q(k)$$

(3) l'application de $\varphi_{h/2}^{[2]}$

$$\phi(k+1)=\begin{pmatrix} B \\ A \\ 1 \end{pmatrix}\frac{\partial H_0^{[2]}}{\partial Q}(Q(+1))h+\phi\left(k+\frac{1}{2}\right)$$

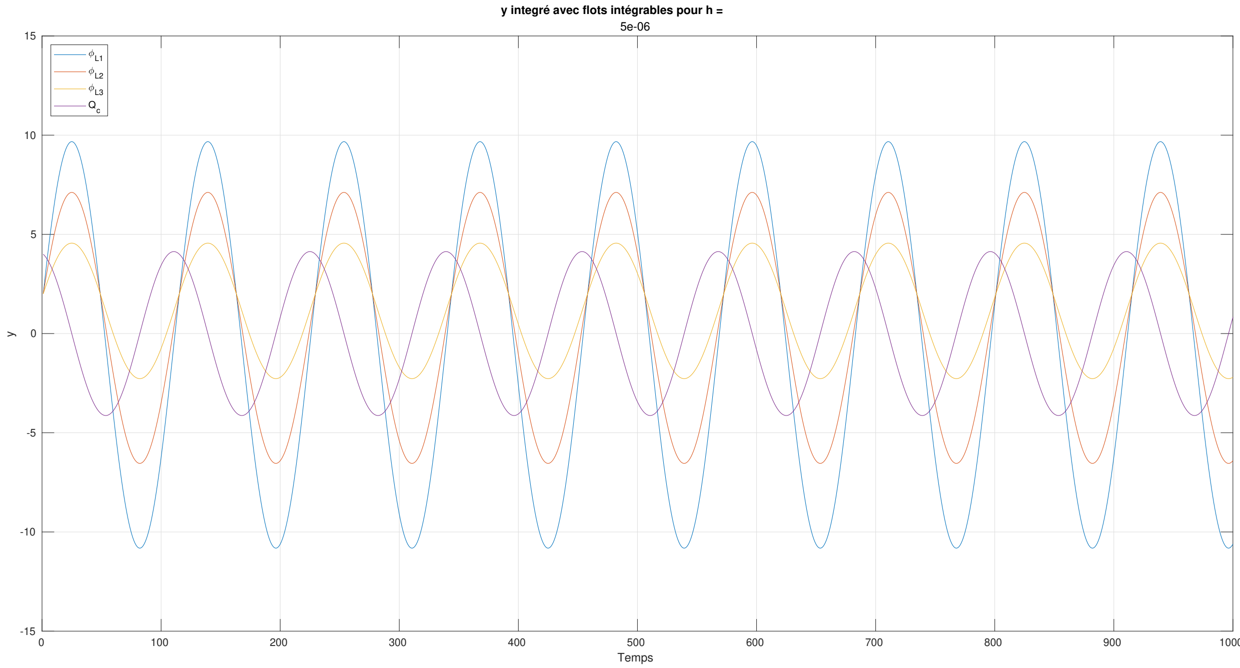
On obtient le code suivant :

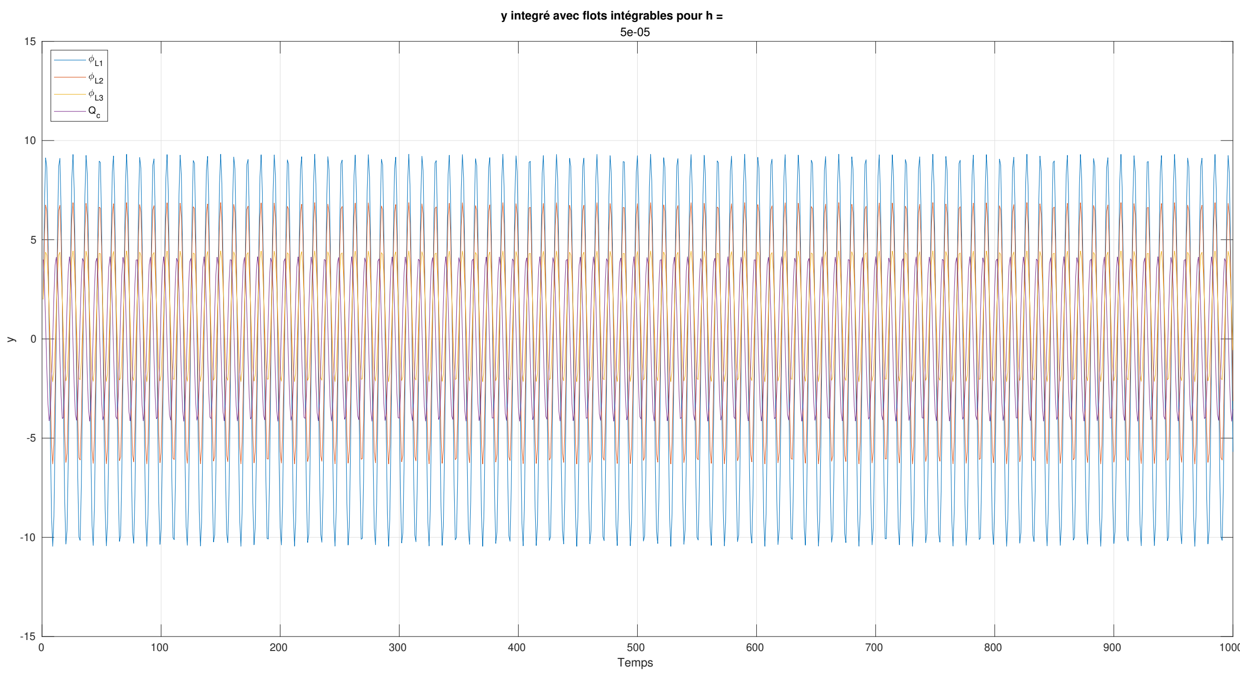
```
function [x1 ,x2 ,x3 ,x4]= flots_inteLC(x0 ,h ,N)

x1=zeros(1 ,N); x2=zeros(1 ,N); x3=zeros(1 ,N); x4=zeros(1 ,N);

x1(1)=x0(1); x2(1)=x0(2); x3(1)=x0(3); x4(1)=x0(4);
A=2; B=3;

     for k= 2:N

 %phi[2]_h/2
          temp1x1 = B*dH0_2(x4(k-1))*h + x1(k-1);
          temp1x2 = A*dH0_2(x4(k-1))*h + x2(k-1);
          temp1x3 = dH0_2(x4(k-1))*h + x3(k-1);

%phi^1_h
x4(k) = -B*dH0_1(temp1x1)*h-A*dH0_1(temp1x2)*h-dH0_1(temp1x3)*h+x4(k-1);

%phi^2_h/2
          x1(k) = B*dH0_2(x4(k))*h + temp1x1;
          x2(k) = A*dH0_2(x4(k))*h + temp1x2;
          x3(k) = dH0_2(x4(k))*h + temp1x3;

     end
end
```

On obtient les figures suivantes avec le code test.m suivant :

```
Yinit = [2 ,2 ,2 ,4]; % on prend x0 qui respecte Casimir
Xr = [0 ,1];

N = 1000;
t = 1:N;
h = 1e-6;

% On intègre y avec flots_inteLC et on plot

[x1 ,x2 ,x3 ,x4] = flots_inteLC(Yinit , h, N);
plot(t, x1, t, x2, t, x3, t, x4);
ylim([-15 ,15]);
grid on;
legend('\phi_{L1}', '\phi_{L2}', '\phi_{L3}', 'Q_c', 'Location', '
    NorthWest');
title('y integré avec flots intégrables pour h = ', h');
xlabel('Temps');
ylabel('y');
```

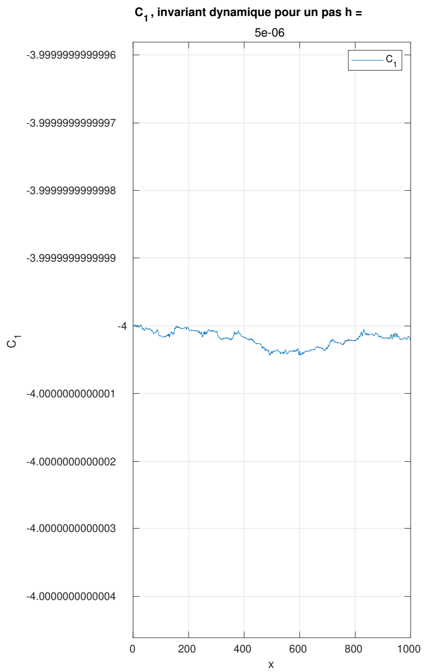
On obtient les figures suivantes :

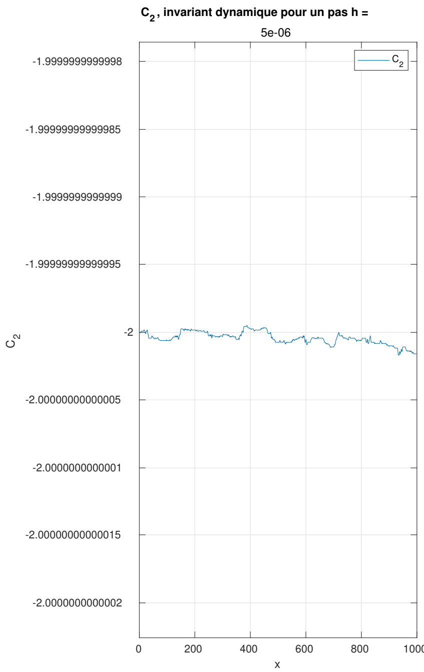


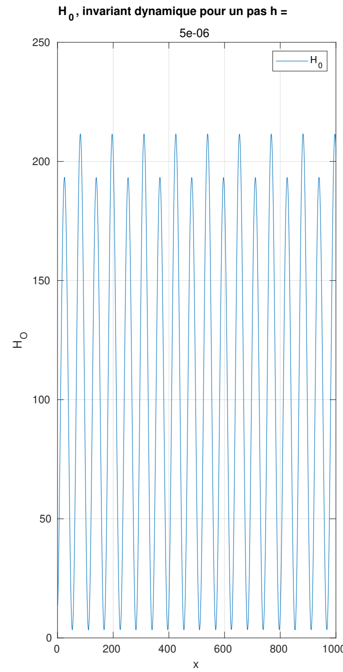


3.2 Les invariants dynamiques

Pour les invariants dynamiques on obtient les figures suivantes :

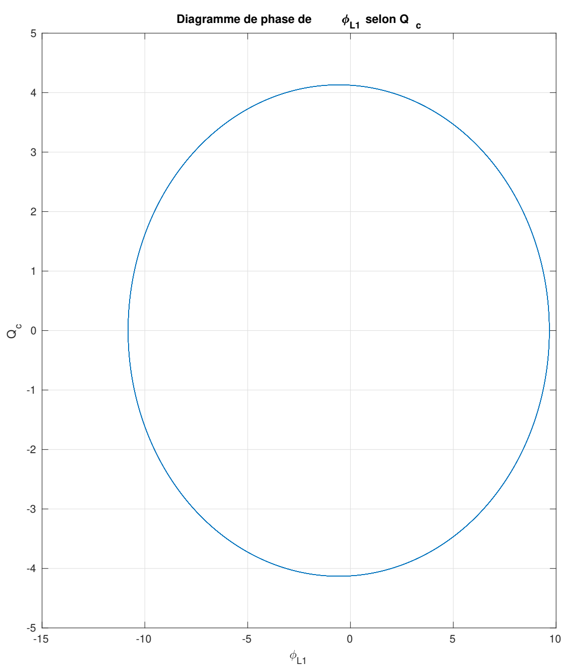


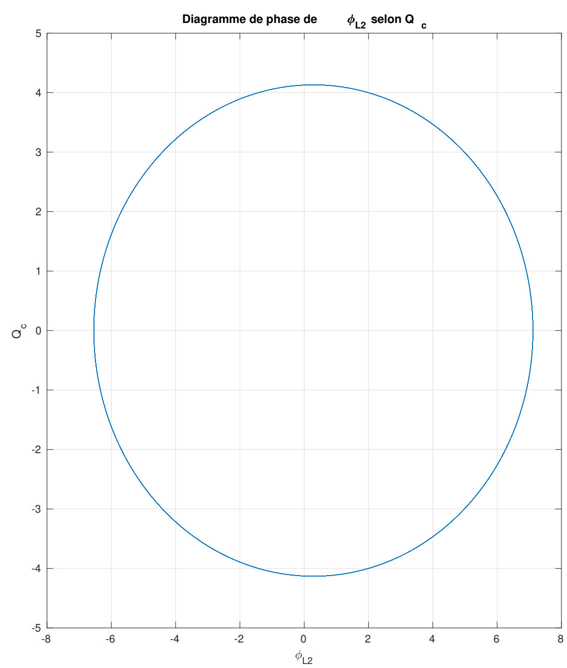




3.3 Diagramme de phase

Et pour obtenir le diagramme des phases pour la méthodes des flots intégrables on a les figures suivantes :





4 Comparaison des méthodes

4.1 Comparaison des invariants de Casimir

On remarque que la fonction de Casimir prend les mêmes valeurs pour Euler Explicite et pour Flots Intégrables. Cependant dans les deux cas les valeurs sont vraiment très faibles ($C_1 = -4$ et $C_2 = -2$). Cependant pour la méthode d'Euler Explicite on constate l'apparition d'erreur au fur et à mesure du temps. On peut considérer que cet invariant est stable pour les deux méthodes mais plus stable pour la méthode des flots intégrables que celle d'Euler explicite.

4.2 Comparaison diagramme de phase

La méthode des flots intégrables est meilleure que Euler pour le déphasage. Elle est comme la méthode Euler Symplectique vu au précédent TP.

4.3 Comparaison invariant H_0

On constate bien que la méthode des flots intégrables est meilleure pour la conservation de H_0 que la méthode d'Euler. Pour la méthode d'Euler Explicite on que H_0 diverge alors que pour la méthode des flots intégrables on a que H_0 reste bornée, comme pour la méthode Euler Symplectique.

5 Annexe (Code Matlab)

5.1 Script Matlab pour la méthode Euler explicite

```
%% Paramètres initiaux
Yinit = [2,2,2,4]; % on prend x0 qui respecte Casimir
Xr = [0.5,1];
N = 1000; h = 5e-6;
C = 2.2e-4; L1 = 1.8e-2; L2 = 0.8e-2; L3 = 1.3e-3;
A = 2; B = 3;
[t,y] = EEx(Xr,Yinit,h);

%% On trace maintenant l'ensemble des variables selon le temps :
figure;

plot(t,y(1,:), t,y(2,:), t,y(3,:), t,y(4,:));
grid on;
legend('\phi_{L1}', '\phi_{L2}', '\phi_{L3}', 'Q_c', 'Location', 'NorthWest');
title('y integré avec Euler explicite pour h = ', h);

%% On va maintenant regarder les invariants :
figure;

subplot(1,3,1);
plot(t, y(1,:) - 3*y(3,:));
grid on;
```

```
title('C_1 , 1er invariant');

subplot(1,3,2);
plot(t, y(2,:)- 2*y(3,:));
grid on;
title('C_2 , 2ème invariant');

subplot(1,3,3);
plot(t, (y(1,:).*y(1,:))/(2*L1) + (y(2,:).*y(2,:))/(2*L2) + (y(3,:).*y
    (3,:))/(2*L3) + (y(4,:).*y(4,:))/(2*C));
grid on;
title('H_0 , 3ème invariant');

%% Diagramme de phase

figure;

subplot(1,2,1);
plot(y(1,:), y(4,:));
grid on;
title('Diagramme de phase de \phi_{L1} selon Q_c');
xlabel('\phi_{L1}'); ylabel('Q_c');

subplot(1,2,2);
plot(y(2,:), y(4,:));
grid on;
title('Diagramme de phase de \phi_{L2} selon Q_c');
xlabel('\phi_{L2}'); ylabel('Q_c');
```

5.2 Script Matlab pour la méthode Flots Intégrables

```
%% Paramètres initiaux
Yinit = [2,2,2,4]; % on prend x0 qui respecte Casimir
Xr = [0,1];
N = 1000; t = 1:N; h = 5e-6;

%% On intègre y avec flots_inteLC et on plot

[x1,x2,x3,x4] = flots_inteLC(Yinit, h, N);

figure;
plot(t, x1, t, x2, t, x3, t, x4);
ylim([-15,15]);
grid on;
legend('\phi_{L1}', '\phi_{L2}', '\phi_{L3}', 'Q_c', 'Location', '
    NorthWest');
title('y integré avec flots intégrables pour h = ', h');
xlabel('Temps'); ylabel('y');

%% Diagramme de phase

figure;
[x1,x2,x3,x4] = flots_inteLC(Yinit,h,N);
```

```
subplot(1,2,1);
plot(x1,x4);
grid on;
title('Diagramme de phase de \phi_{L1} selon Q_c');
xlabel('\phi_{L1}'); ylabel('Q_c');

subplot(1,2,2);
plot(x2,x4);
grid on;
title('Diagramme de phase de \phi_{L2} selon Q_c');
xlabel('\phi_{L2}'); ylabel('Q_c');

%% Invariants dynamiques

H0 = zeros(1,N) ; C1 = zeros(1,N) ; C2 = zeros(1,N) ;
Y=[x1; x2; x3; x4];
C=2.210^(-4) ; L3=1.310^(-3) ; L2=0.810^(-2) ; L1=1.810^(-2) ;

for i =1:N
    H0(i) = (Y(4,i)^2)*(2*C) + (Y(1,i)^2)/(2*L1 ) + (Y(2,i)^2)/(2*L2) + (Y(3,i)^2)/(2*L3);
end

for i = 1:N
    C1(i) = Y(1,i) - B*Y(3,i); % Calcul de casimir avec son expression
end

for i =1:N
    C2(i) = Y(2,i) - A*Y(3,i) ; % Calcul de Casimir avec son expression
end

figure;

subplot(1,3,1);
plot(C1)
grid on;
title('C_1, invariant dynamique pour un pas h = ', h);
legend('C_1');
xlabel('x'); ylabel('C_1');

subplot(1,3,2);
plot(C2)
grid on;
title('C_2, invariant dynamique pour un pas h = ', h);
legend('C_2');
xlabel('x'); ylabel('C_2');

subplot(1,3,3);
plot(H0);
grid on;
title('H_0, invariant dynamique pour un pas h = ', h);
legend('H_0');
xlabel('x'); ylabel('H_O');
```