



Projet de modélisation des systèmes chaotiques par le pendule double

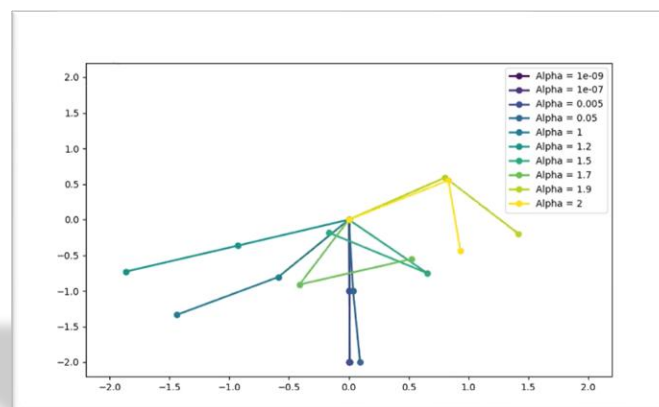


Table des matières

I.	Introduction	3
II.	Modélisation physique du pendule double	4
1.	Démonstration des équations d'évolution	4
2.	Mise sous problème de Cauchy	7
III.	Analyse numérique des équations différentielles ordinaires couplées	9
1.	Méthode d'Euler explicite	10
2.	Méthode de Runge-Kutta d'ordre 4.....	15
3.	Méthode de Verlet	18
IV.	Mouvement chaotique et sensibilité par rapport aux conditions initiales.....	22
1.	Vitesse angulaire θ_i' en fonction de l'angle θ_i pour $i = 1, 2$	22
2.	Vitesse angulaire θ_i' en fonction de l'angle θ_i pour une position très proche de celle d'équilibre.....	23
3.	Évolution du système vers un mouvement chaotique	25
V.	Conclusion	28

I. Introduction

Le double pendule notamment connu pour ses mouvements complexes et imprévisibles, est un exemple classique de système dynamique chaotique. Ce système, constitué de deux pendules attachés l'un à l'autre, illustre comment de petites variations dans les conditions initiales peuvent entraîner des comportements extrêmement différents. En raison de cette sensibilité, le double pendule est souvent étudié pour comprendre les concepts fondamentaux du chaos et des systèmes non linéaires.

Dans ce rapport, nous explorerons en profondeur la dynamique du double pendule à travers plusieurs étapes. Nous commencerons par une modélisation physique du système, où nous démontrerons les équations d'évolution du pendule en utilisant les lois fondamentales de la dynamique, et les mettrons sous forme de problème de Cauchy. Ensuite, nous procéderons à une analyse numérique des équations différentielles ordinaires couplées, en utilisant trois méthodes distinctes : la méthode d'Euler explicite, la méthode de Runge-Kutta d'ordre 4, et la méthode de Verlet. Chaque méthode sera détaillée, évaluée et comparée en termes de précision et de stabilité.

La quatrième section sera consacrée à l'étude du mouvement chaotique du double pendule et à sa sensibilité par rapport aux conditions initiales. Nous analyserons notamment la vitesse angulaire en fonction de l'angle pour différentes positions initiales, en observant comment de légères variations peuvent influencer le comportement dynamique du système.

II. Modélisation physique du pendule double

1. Démonstration des équations d'évolution

Démontrons, en utilisant le principe fondamental de la dynamique, les équations d'évolutions du pendule double avec les conditions initiales de l'exercice :

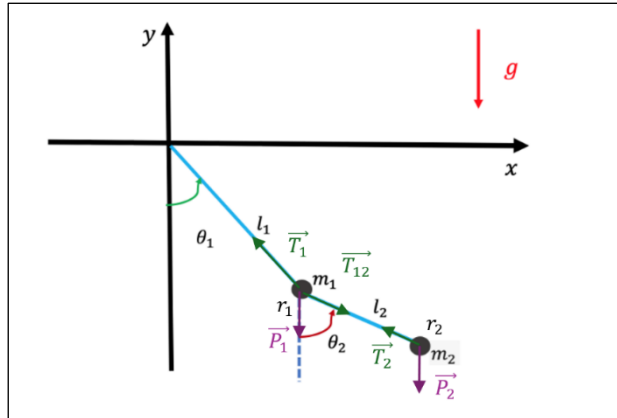


Figure 1 : Configuration du pendule double

D'après la seconde loi de Newton :

$$\sum \vec{F} = m\vec{a}$$

Pour chaque masse, nous avons :

$$\begin{cases} m_1 \vec{a}_1 = \vec{T}_1 + \vec{T}_{12} + \vec{P}_1 \text{ (1)} \\ m_2 \vec{a}_2 = \vec{P}_2 + \vec{T}_2 \text{ (2)} \end{cases}$$

où \vec{a}_1 est l'accélération de la masse m_1 et \vec{a}_2 celle de m_2

De plus,

$$\vec{v}_1 = \frac{d\vec{r}_1}{dt} = \begin{pmatrix} l_1 \dot{\theta}_1 \cos(\theta_1) \\ l_1 \dot{\theta}_1 \sin(\theta_1) \end{pmatrix}$$

et

$$\vec{a}_1 = \frac{d\vec{v}_1}{dt} = \begin{pmatrix} l_1 \ddot{\theta}_1 \cos(\theta_1) - l_1 \dot{\theta}_1^2 \sin(\theta_1) \\ l_1 \ddot{\theta}_1 \sin(\theta_1) + l_1 \dot{\theta}_1^2 \cos(\theta_1) \end{pmatrix}$$

avec \vec{r}_1 la position de m_1 , \vec{v}_1 sa vitesse et \vec{a}_1 son accélération

Effectuons le même calcul pour \vec{a}_2 :

$$\vec{a}_2 = \frac{d\vec{v}_2}{dt} = \begin{pmatrix} l_1 \ddot{\theta}_1 \cos(\theta_1) - l_1 \dot{\theta}_1^2 \sin(\theta_1) + l_2 \ddot{\theta}_2 \cos(\theta_2) - l_2 \dot{\theta}_2^2 \sin(\theta_2) \\ l_1 \dot{\theta}_1 \sin(\theta_1) + l_1 \ddot{\theta}_1 \sin(\theta_1) + l_2 \ddot{\theta}_2 \sin(\theta_2) + l_2 \dot{\theta}_2^2 \cos(\theta_2) \end{pmatrix}$$

On note que $\|\vec{T}_{12}\| = \|\vec{T}_2\|$ et $\vec{T}_{12} = -\vec{T}_2$

$$\text{alors } \vec{T}_2 = T_2 \begin{pmatrix} -\sin(\theta_2) \\ \cos(\theta_2) \end{pmatrix}$$

et $\vec{a}_{12} = \vec{a}_2 - \vec{a}_1 = \begin{pmatrix} l_2 \ddot{\theta}_2 \cos(\theta_2) - l_2 \dot{\theta}_2^2 \sin(\theta_2) \\ l_2 \dot{\theta}_2 \sin(\theta_2) + l_2 \ddot{\theta}_2 \sin(\theta_2) \end{pmatrix}$ qui est l'accélération de m_2 par rapport à m_1

Désormais, réécrivons dans l'équation (2) précédemment établie : $m_2 \vec{a}_2 = \vec{P}_2 + \vec{T}_2$

$$\Leftrightarrow m_2 \begin{bmatrix} l_1 \ddot{\theta}_1 \begin{pmatrix} \cos(\theta_1) \\ \sin(\theta_1) \end{pmatrix} + l_1 \dot{\theta}_1^2 \begin{pmatrix} -\sin(\theta_1) \\ \cos(\theta_1) \end{pmatrix} + l_2 \ddot{\theta}_2 \begin{pmatrix} \cos(\theta_2) \\ \sin(\theta_2) \end{pmatrix} + l_2 \dot{\theta}_2^2 \begin{pmatrix} -\sin(\theta_2) \\ \cos(\theta_2) \end{pmatrix} \end{bmatrix} = \begin{pmatrix} -m_2 g \\ 0 \end{pmatrix} + T_2 \begin{pmatrix} -\sin(\theta_2) \\ \cos(\theta_2) \end{pmatrix}$$

Or nous ne connaissons pas l'expression de T_2 , utilisons l'astuce de la multiplication avec la matrice de rotation $\begin{pmatrix} \cos(\theta_2) & \sin(\theta_2) \\ -\sin(\theta_2) & \cos(\theta_2) \end{pmatrix}$ afin que la composante sur l'un des axes de \vec{T}_2 soit nulle.

Par conséquent, on obtient par simplification et avec les formules trigonométriques l'équation suivante :

$$m_2 g \begin{pmatrix} -\sin(\theta_2) \\ -\cos(\theta_2) \end{pmatrix} + T_2 \begin{pmatrix} 0 \\ 1 \end{pmatrix} = m_2 l_1 \left[\ddot{\theta}_1 \begin{pmatrix} \cos(\theta_1 - \theta_2) \\ \sin(\theta_1 - \theta_2) \end{pmatrix} + \dot{\theta}_1^2 \begin{pmatrix} \sin(\theta_2 - \theta_1) \\ \cos(\theta_1 - \theta_2) \end{pmatrix} \right] + m_2 l_2 \left[\ddot{\theta}_2 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \dot{\theta}_2^2 \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right]$$

Nous extrayons la première ligne :

$$\begin{aligned} m_2 g \sin(\theta_2) + m_2 l_1 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - m_2 l_1 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) + m_2 l_2 \ddot{\theta}_2 &= 0 \\ \Leftrightarrow g \sin(\theta_2) + l_1 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - l_1 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) + l_2 \ddot{\theta}_2 &= 0 \end{aligned}$$

Nous retrouvons bien la deuxième équation de mouvement du pendule double.

On reprend (1) et (2) :

$$\begin{aligned} &\begin{cases} m_1 \vec{a}_1 = \vec{T}_1 + \vec{T}_{12} + \vec{P}_1 \\ m_2 \vec{a}_2 = \vec{P}_2 + \vec{T}_2 \end{cases} \\ \Leftrightarrow &\begin{cases} m_1 \vec{a}_1 = \vec{T}_1 - \vec{T}_2 + m_1 \vec{g} \\ m_2 \vec{a}_1 + m_2 \vec{a}_{12} = m_2 \vec{g} + \vec{T}_2 \end{cases} \end{aligned}$$

avec $\overrightarrow{T_{12}} = -\overrightarrow{T_2}$ et $a_2 = \overrightarrow{a_1} + \overrightarrow{a_{12}}$ par composition des accélérations

Ainsi (1) + (2) :

$$\begin{aligned} m_1 \overrightarrow{a_1} + m_2 \overrightarrow{a_1} + m_2 \overrightarrow{a_{12}} &= \overrightarrow{T_1} - \overrightarrow{T_2} + \overrightarrow{T_2} + m_1 \vec{g} + m_2 \vec{g} \\ \Leftrightarrow (m_1 + m_2) \overrightarrow{a_1} + m_2 \overrightarrow{a_{12}} &= \overrightarrow{T_1} + \vec{g}(m_1 + m_2) \end{aligned}$$

Par conséquent :

$$T_1 \begin{pmatrix} -\sin(\theta_1) \\ \cos(\theta_1) \end{pmatrix} + (m_1 + m_2)g \begin{pmatrix} 0 \\ -1 \end{pmatrix} = (m_1 + m_2)l_1 \left[\ddot{\theta}_1 \begin{pmatrix} \cos(\theta_1) \\ \sin(\theta_1) \end{pmatrix} + \dot{\theta}_1^2 \begin{pmatrix} -\sin(\theta_1) \\ \cos(\theta_1) \end{pmatrix} \right] + m_2 l_2 \left[\ddot{\theta}_2 \begin{pmatrix} \cos(\theta_2) \\ \sin(\theta_2) \end{pmatrix} + \dot{\theta}_2^2 \begin{pmatrix} -\sin(\theta_2) \\ \cos(\theta_2) \end{pmatrix} \right]$$

Ici, tout comme précédemment, nous ne connaissons pas T_1 , on multiplie par la matrice de rotation précédente également ce qui donne après simplifications :

$$T_1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} + (m_1 + m_2)g \begin{pmatrix} -\sin(\theta_1) \\ -\cos(\theta_1) \end{pmatrix} = (m_1 + m_2)l_1 \left[\ddot{\theta}_1 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \dot{\theta}_1^2 \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right] + m_2 l_2 \left[\ddot{\theta}_2 \begin{pmatrix} \cos(\theta_2 - \theta_1) \\ \sin(\theta_2 - \theta_1) \end{pmatrix} + \dot{\theta}_2^2 \begin{pmatrix} \sin(\theta_1 - \theta_2) \\ \cos(\theta_1 - \theta_2) \end{pmatrix} \right]$$

Finalement, en extrayant la première ligne, on obtient :

$$\begin{aligned} T_1 \times 0 - (m_1 + m_2)g \sin(\theta_1) &= (m_1 + m_2)l_1 \ddot{\theta}_1 + m_2 l_2 \ddot{\theta}_2 \cos(\theta_2 - \theta_1) + m_2 l_2 \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) \\ \Leftrightarrow (m_1 + m_2)l_1 \ddot{\theta}_1 + m_2 l_2 \ddot{\theta}_2 \cos(\theta_2 - \theta_1) + m_2 l_2 \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) + (m_1 + m_2)g \sin(\theta_1) &= 0 \end{aligned}$$

or $\cos(\theta_2 - \theta_1) = \cos(\theta_1 - \theta_2)$ car \cos est une fonction symétrique :

$$\Leftrightarrow (m_1 + m_2)l_1 \ddot{\theta}_1 + m_2 l_2 \ddot{\theta}_2 \cos(\theta_1 - \theta_2) + m_2 l_2 \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) + (m_1 + m_2)g \sin(\theta_1) = 0$$

Ainsi, nous retrouvons bien les équations de mouvement du double pendule demandées :

$$\begin{cases} (m_1 + m_2)l_1 \ddot{\theta}_1 + m_2 l_2 \ddot{\theta}_2 \cos(\theta_1 - \theta_2) + m_2 l_2 \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) + (m_1 + m_2)g \sin(\theta_1) = 0 \\ g \sin(\theta_2) + l_1 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - l_1 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) + l_2 \ddot{\theta}_2 = 0 \end{cases}$$

2. Mise sous problème de Cauchy

Présentons le problème sous la forme d'un problème de Cauchy tel que

$$Y'(t) = F(Y(t))$$

Définissons le vecteur d'état $Y(t)$:

$$Y(t) = \begin{pmatrix} \theta_1(t) \\ \theta_2(t) \\ \dot{\theta}_1(t) \\ \dot{\theta}_2(t) \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_1' \\ y_2' \end{pmatrix}$$

alors

$$Y'(t) = \begin{pmatrix} \dot{\theta}_1(t) \\ \dot{\theta}_2(t) \\ \ddot{\theta}_1(t) \\ \ddot{\theta}_2(t) \end{pmatrix}$$

Ainsi :

$$F(Y(t)) = F\left(\begin{pmatrix} \theta_1(t) \\ \theta_2(t) \\ \dot{\theta}_1(t) \\ \dot{\theta}_2(t) \end{pmatrix}\right) = \begin{pmatrix} \dot{\theta}_1(t) \\ \dot{\theta}_2(t) \\ f_1(\theta_1(t), \theta_2(t), \dot{\theta}_1(t), \dot{\theta}_2(t)) \\ f_2(\theta_1(t), \theta_2(t), \dot{\theta}_1(t), \dot{\theta}_2(t)) \end{pmatrix} = \begin{pmatrix} y_1' \\ y_2' \\ f_1(\theta_1(t), \theta_2(t), \dot{\theta}_1(t), \dot{\theta}_2(t)) \\ f_2(\theta_1(t), \theta_2(t), \dot{\theta}_1(t), \dot{\theta}_2(t)) \end{pmatrix}$$

où f_1 et f_2 représentent les expressions des équations du mouvement pour $\ddot{\theta}_1(t)$ et $\ddot{\theta}_2(t)$ respectivement.

Isolons alors $\ddot{\theta}_1(t)$ et $\ddot{\theta}_2(t)$ issus des équations de mouvement du double pendule trouvées précédemment à l'aide du programme python « solveur equation.py » et ainsi après simplification, nous obtenons :

$$\begin{cases} f_1 = \ddot{\theta}_1 = \frac{-\frac{\dot{\theta}_1^2 l_1 m_2 \sin(2\theta_1 - 2\theta_2)}{2} - \dot{\theta}_2^2 l_2 m_2 \sin(\theta_1 - \theta_2) - g m_1 \sin(\theta_1) - \frac{g m_2 \sin(\theta_1)}{2} - \frac{g m_2 \sin(\theta_1 - 2\theta_2)}{2}}{l_1(m_1 - m_2 \cos^2(\theta_1 - \theta_2) + m_2)} \\ f_2 = \ddot{\theta}_2 = \frac{\frac{\dot{\theta}_1^2 l_1 m_1 \sin(\theta_1 - \theta_2)}{2} + \dot{\theta}_1^2 l_1 m_2 \sin(\theta_1 - \theta_2) + \frac{\dot{\theta}_2^2 l_2 m_2 \sin(2\theta_1 - 2\theta_2)}{2} - \frac{g m_1 \sin(\theta_2)}{2} + \frac{g m_1 \sin(2\theta_2)}{2} + \frac{g m_1 \sin(2\theta_1 - \theta_2)}{2} - \frac{g m_2 \sin(\theta_2)}{2} + \frac{g m_2 \sin(2\theta_1 - \theta_2)}{2}}{l_2(m_1 - m_2 \cos^2(\theta_1 - \theta_2) + m_2)} \end{cases}$$

Ce qui équivaut après simplification avec les formules trigonométriques à :

$$\begin{cases} f_1 = \ddot{\theta}_1 = \frac{-m_2 l_1 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) \cos(\theta_1 - \theta_2) + g m_2 \sin(\theta_2) \cos(\theta_1 - \theta_2) - m_2 l_2 \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) - g(m_1 + m_2) \sin(\theta_1)}{l_1(m_1 + m_2) - m_2 l_1 \cos^2(\theta_1 - \theta_2)} \\ f_2 = \ddot{\theta}_2 = \frac{m_2 l_2 \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) \cos(\theta_1 - \theta_2) + g(m_1 + m_2) \sin(\theta_1) \cos(\theta_1 - \theta_2) + (m_1 + m_2) l_1 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) - g(m_1 + m_2) \sin(\theta_2)}{l_2(m_1 + m_2) - m_2 l_2 \cos^2(\theta_1 - \theta_2)} \end{cases}$$

Finalement :

$$F(Y(t)) = \begin{pmatrix} y_1' \\ y_2' \\ f_1 = \ddot{\theta}_1 \\ f_2 = \ddot{\theta}_2 \end{pmatrix}$$

Où $m_1 = m_2 = 1kg$ et $l_1 = l_2 = 1m$

Avec

$$Y(0) = \begin{pmatrix} \frac{\pi}{2} \\ \frac{\pi}{2} \\ 0 \\ 0 \end{pmatrix}$$

Le domaine de départ de la fonction F est l'espace des états possibles du pendule, qui est un sous-ensemble de \mathbb{R}^4 (puisque chaque composante de $Y(t)$ est une valeur réelle). Le domaine d'arrivée de F est également \mathbb{R}^4 , car $Y'(t)$ est un vecteur de vitesses et d'accélérations angulaires dans \mathbb{R}^4 . Alors, on a $F: \mathbb{R}^4 \rightarrow \mathbb{R}^4$.

Par conséquent, nous utiliserons l'expression de $F(Y(t))$ afin de déterminer numériquement et graphiquement l'évolution du mouvement du double pendule sur nos programmes pythons.

III. Analyse numérique des équations différentielles ordinaires couplées

Dans cette partie du projet, nous avons abordé la simulation numérique d'un problème dynamique complexe, représenté mathématiquement dans la partie précédente sous forme d'un problème de Cauchy. Nous avons ainsi développé et implémenté une fonction F , qui calcule les dérivées du système à un instant donné pour deux pendules couplés. Cette fonction est essentielle car elle résume les équations différentielles décrivant le mouvement des pendules sous l'effet de la gravité, en tenant compte des paramètres tels que les longueurs des tiges l_1 et l_2 et les masses m_1 et m_2 .

Nous avons développé des fonctions pour simuler le comportement d'un système de deux pendules en utilisant la bibliothèque numpy, qui facilite la manipulation efficace des données et offre une performance optimale lors des calculs vectorisés. Trois méthodes numériques distinctes ont été implémentées pour intégrer les équations du mouvement, chacune adaptée à des besoins spécifiques en termes de précision et de stabilité :

1. Méthode d'Euler explicite : Cette approche calcule les états futurs du système en ajoutant simplement le produit du pas de temps et du vecteur dérivé à l'état courant. Bien que simple et directe, cette méthode peut être moins stable et précise pour des pas de temps plus grands, nécessitant un équilibre entre précision et coût computationnel.
2. Méthode de Runge-Kutta d'ordre 4 (RK4) : Cette méthode améliore considérablement la précision en calculant quatre approximations intermédiaires des taux de changement. Elle est plus complexe et coûteuse en termes de calculs mais offre une bien meilleure stabilité et précision pour un même pas de temps, ce qui en fait un choix privilégié pour des simulations exigeantes.
3. Méthode de Verlet : Spécialement conçue pour les problèmes de mécanique, cette méthode est idéale pour les simulations conservant l'énergie, comme celles impliquant des mouvements oscillatoires. Elle utilise une approche en deux étapes pour mettre à jour les positions et les vitesses, en assurant une meilleure conservation de l'énergie sur de longs intervalles de temps.

1. Méthode d'Euler explicite

```
#Euler explicite
def method_Euler_explicite(Y, l1, l2, m1, m2, dt, T):
    N = int(T / dt)
    temps = np.linspace(0, T, N)
    sol = np.zeros((4, N))
    sol[:, 0] = Y
    for i in range(N - 1):
        sol[:, i + 1] = sol[:, i] + dt * F(sol[:, i], l1, l2, m1, m2)
    return temps, sol
```

Figure 2 : Algorithme python de la méthode Euler explicite

Voici la méthode de d'Euler explicite que nous avons implémentée afin de résoudre des équations différentielles ordinaires en intégrant un vecteur d'état Y sur une période spécifiée T , avec un pas de temps défini dt .

Paramètres et initialisation :

- Nombre de pas (N) : Calculé comme $\frac{T}{dt}$ détermine combien de fois la fonction sera évaluée et mise à jour au cours de la période T .
- Vecteur de temps ($temps$) : Un tableau numpy généré avec « `np.linspace(0, T, N)` », il représente les instants successifs à chaque pas de temps pour lesquels la solution est calculée.
- Matrice de solutions (sol) : Initialisée à zéro avec des dimensions $4 \times N$, cette matrice stocke les états successifs du système, ici deux angles et deux vitesses angulaires pour un système de deux pendules

Processus d'intégration :

Une boucle « for » parcourt chaque intervalle de temps, commençant à 0 et s'arrêtant à $N - 1$. À chaque itération, la fonction mise à jour du vecteur d'état, F , est appelée avec l'état courant et les paramètres physiques du système (les longueurs des pendules et leurs masses). L'état suivant est estimé en ajoutant à l'état courant le produit du pas de temps dt par les dérivées calculées par F .

Résultats :

- Temps : Le vecteur des instants calculés, représentant chaque point de mesure dans l'intervalle de simulation.
- Sol : La matrice contenant la solution numérique du vecteur d'état à chaque pas de temps.

Pour tester notre méthode, nous avons tracer la trajectoire de 0 à $T = 2s$ de nos pendules en utilisant les paramètres ci-dessous :

$$\begin{cases} m_1 = m_2 = 1 \\ l_1 = l_2 = 1 \\ Y_0 = (\frac{\pi}{2}, \frac{\pi}{2}, 0, 0)^T \\ N = 10^3 \end{cases}$$

Voici la vidéo de notre simulation du pendule avec Euler explicite avec ces conditions ([lien](#)).

Vous trouverez le code traitant ce système dans le fichier « Partie 1, q1-5 ».

Dans ce projet, nous avons dû mettre en place une série de fonctions pour visualiser et analyser la dynamique du système :

- La première fonction, « *anglePlot* », est conçue pour tracer les angles $\theta_1(t)$ et $\theta_2(t)$ en fonction du temps, permettant d'observer comment chaque pendule oscille tout au long de la simulation.

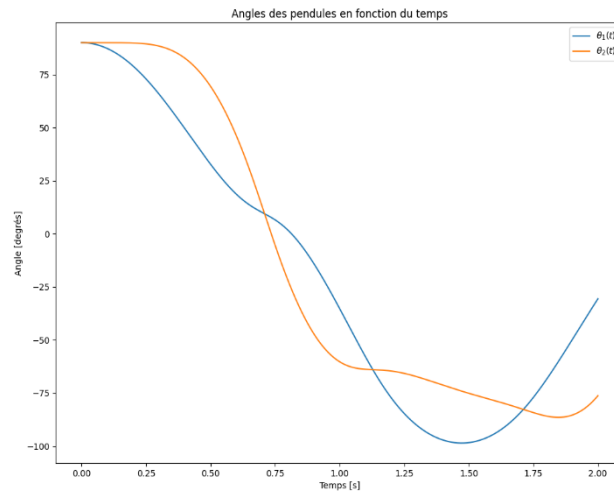


Figure 3 : Angles des pendules en fonctions du temps pour le système 1 (*angle_plot_2.png*)

La courbe présentée illustre les angles de deux pendules en fonction du temps pour le système 1. Le graphique comporte deux courbes distinctes, étiquetées $\theta_1(t)$ en bleu et $\theta_2(t)$ en orange. L'axe des abscisses représente le temps en secondes, tandis que l'axe des ordonnées indique l'angle des pendules.

Les deux courbes montrent un comportement oscillatoire caractéristique des mouvements pendulaires. Initialement, $\theta_1(t)$ commence à une valeur positive élevée et décroît progressivement, tandis que $\theta_2(t)$ débute également à une valeur positive élevée mais décroît plus rapidement que $\theta_1(t)$. Les courbes se croisent à plusieurs reprises, indiquant des moments où les deux pendules possèdent le même angle. Les amplitudes des oscillations diminuent légèrement au fil du temps, suggérant la présence d'un amortissement dans le système des pendules.

- La deuxième fonction, « *angulVelocityPlot* », sert à tracer les vitesses angulaires $\dot{\theta}_1(t)$ et $\dot{\theta}_2(t)$. Cette visualisation aide à analyser la dynamique de vitesse des pendules.

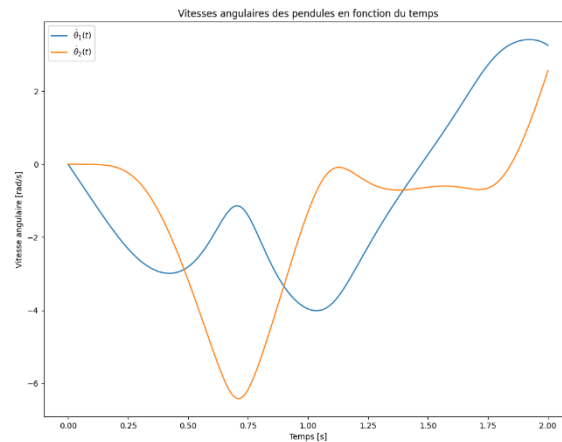


Figure 3 : Vitesses angulaires des pendules en fonctions du temps pour le système 1 (*angular_velocity_plot_2.png*)

Les deux courbes affichent un comportement oscillatoire, correspondant aux variations des vitesses angulaires des pendules au cours du temps. Les courbes montrent une inversion de signe, indiquant le changement de sens de vitesse angulaire des pendules vers 1.75s. Les pics et les creux des courbes représentent les points de vitesse angulaire maximale et minimale. Ces oscillations suggèrent que les deux pendules ont des comportements dynamiques différents mais couplés, influençant mutuellement leurs vitesses angulaires.

- La troisième fonction que nous avons dû implémenter est « *trajectoryPlot* », qui représente graphiquement les trajectoires des masses des pendules, en calculant les positions $x_i(t)$ et $y_i(t)$ à partir des angles. Cette représentation est particulièrement utile pour visualiser le chemin spatial parcouru par chaque masse pendulaire.

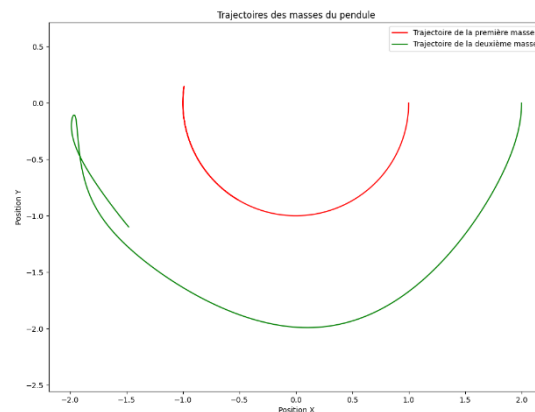


Figure 4 : Trajectoires des pendules en fonctions du temps pour le système 1 (*trajectories_2.png*)

Les deux courbes illustrent les chemins suivis par les masses des pendules au cours du temps. La trajectoire de la première masse m_1 (en rouge) décrit une courbe qui semble être une portion d'ellipse ou de cercle, suggérant un mouvement oscillatoire régulier. La trajectoire de la deuxième masse m_2 (en vert) montre une oscillation plus complexe. La trajectoire ressemble très fortement à la trajectoire attendue d'après la vidéo que vous nous avez fourni dans le sujet.

- Pour finir, nous avons dû implémenter une série de fonctions pour calculer et tracer l'évolution de l'énergie totale du système. La fonction « *energyCompute* » calcule l'énergie totale $E_T(t)$ ou $E_m(t)$, qui est la somme des énergies cinétique $E_c(t)$ et potentielle $E_p(t)$, basée sur les positions et vitesses des pendules. L'observation de la constance de cette énergie au fil du temps est une validation puissante de la justesse de la simulation numérique, conformément aux principes de conservation de l'énergie dans un système mécanique idéal. La fonction « *energyEvolution* » prend ces calculs et les représente graphiquement, illustrant comment l'énergie totale évolue, ou idéalement reste constante, au cours du temps.

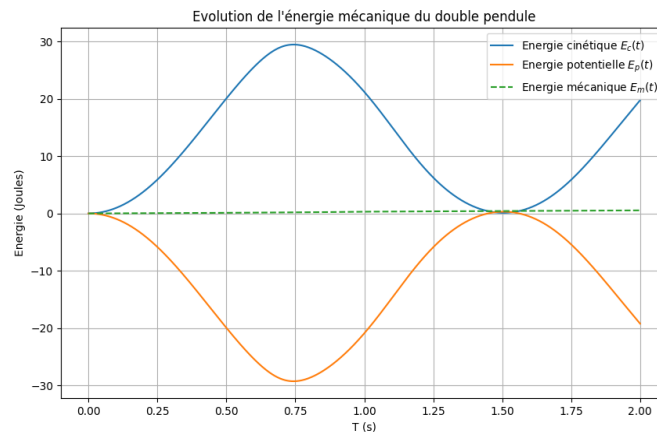


Figure 5 : Evolutions des énergies en fonctions du temps pour le système 1 (energy_evolution_1.png)

Graphiquement on voit que l'énergie semble plus ou moins conservée, mais pour plus de précision, j'ai implémenté une fonction qui calcule la conservation d'énergie de mon système durant son exécution avec un paramètre d'acceptation assez important à 10^{-5} .

```
L'énergie du double pendule est non conservée.
La différence d'énergie mécanique est 0.16432545128086443
```

Figure 6 : Sortie python d'analyse de la conservation de l'énergie du double pendule du système 1 avec Euler explicite

L'énergie totale de ce système est donc non conservée selon nos paramètres, laissant suggérer que cela est probablement dû à la précision d'Euler explicite, une méthode d'ordre 1.

Pour vérifier cela nous allons tester un nouveau système toujours avec Euler explicite en changeant les angles de départ de nos pendules :

$$\begin{cases} m_1 = m_2 = 1 \\ l_1 = l_2 = 1 \\ Y_0 = (\frac{\pi}{6}, \frac{\pi}{6}, 0, 0)^T \\ N = 10^3 \end{cases}$$

Vous trouverez le code traitant ce système dans le fichier « Partie 1, q6 ». Voici la vidéo YouTube de notre simulation du pendule avec Euler explicite et ces conditions ([lien](#)).

Nous obtenons le graphique suivant :

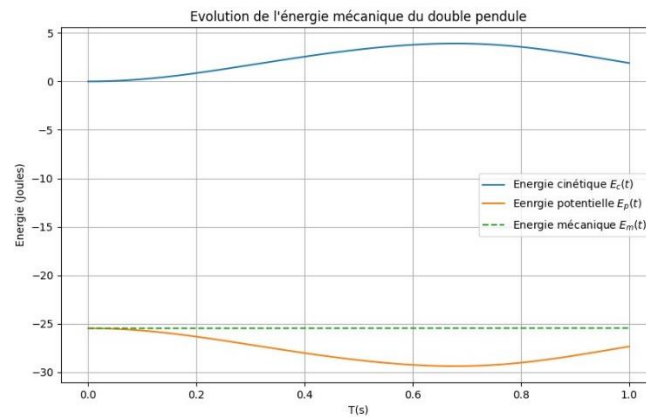


Figure 7 : Evolutions des énergies en fonctions du temps pour le système 2 (energy_evolution_2.png)

```
L'énergie du double pendule est non conservée.
La différence d'énergie mécanique est 0.00726425956934923
```

Figure 8 : Sortie python d'analyse de la conservation de l'énergie du double pendule du second système avec Euler explicite

Encore une fois, nous remarquons que l'énergie n'est pas conservée à 10^{-5} avec la méthode d'Euler explicite. C'est pourquoi dans un second temps nous allons implémenter la méthode de Runge-Kutta 4 qui est connu pour sa précision.

2. Méthode de Runge-Kutta d'ordre 4

```
# Méthode Runge-Kutta d'ordre 4
def method_RK4(Y, l1, l2, m1, m2, dt, T):
    N = int(T / dt)
    temps = np.linspace(0, T, N)
    sol = np.zeros((4, N))
    sol[:, 0] = Y
    for i in range(N - 1):
        k1 = F(sol[:, i], l1, l2, m1, m2)
        k2 = F(sol[:, i] + dt / 2 * k1, l1, l2, m1, m2)
        k3 = F(sol[:, i] + dt / 2 * k2, l1, l2, m1, m2)
        k4 = F(sol[:, i] + dt * k3, l1, l2, m1, m2)
        sol[:, i + 1] = sol[:, i] + dt / 6 * (k1 + 2 * k2 + 2 * k3 + k4)
    return temps, sol
```

Figure 9 : Algorithme python de la méthode de Runge-Kutta 4

La méthode de Runge-Kutta d'ordre 4 (RK4) que nous avons implémentée est une approche plus sophistiquée et précise que Euler explicite pour la résolution numérique des équations différentielles ordinaires. Elle est souvent utilisée pour intégrer des systèmes dynamiques complexes car elle offre une grande précision sans nécessiter un pas de temps extrêmement petit. Voici une description ma méthode RK4.

Paramètres et initialisation :

- Nombre de pas (N) : Calculé comme $\frac{T}{dt}$ détermine combien de fois la fonction sera évaluée et mise à jour au cours de la période T.
- Vecteur de temps (temps) : Un tableau numpy généré avec « np.linspace(0, T, N) », il représente les instants successifs à chaque pas de temps pour lesquels la solution est calculée.
- Matrice de solutions (sol) : Initialisée à zéro avec des dimensions $4 \times N$, cette matrice stocke les états successifs du système, ici deux angles et deux vitesses angulaires pour un système de deux pendules.

Processus d'intégration :

La méthode RK4 nécessite le calcul de quatre gradients intermédiaires (souvent appelés "slopes") pour chaque pas de temps, ce qui permet d'obtenir une estimation précise de l'état du système à l'instant $t + dt$:

- k_1 : Ce premier gradient est calculé directement à partir de l'état courant. Il représente la dérivée du vecteur d'état au début de l'intervalle.
- k_2 : Ce second gradient est estimé en utilisant l'état du système à mi-chemin à travers le pas de temps, $\frac{t+dt}{2}$, en appliquant un demi-pas en utilisant k_1 . Cela offre une prédiction du milieu de l'intervalle basée sur la pente initiale.
- k_3 : Tout comme k_2 , ce gradient est évalué au milieu du pas de temps, mais il utilise k_2 pour une estimation améliorée. Ce troisième gradient fournit une correction à k_2 , se basant sur une meilleure approximation du milieu du pas.

- k_4 : Ce dernier gradient est calculé à la fin de l'intervalle de temps, $t + dt$, en utilisant k_3 pour prédire l'état à ce moment. K4 donne une approximation de la dérivée à la fin de l'intervalle.

Après le calcul de ces quatre gradients, l'état du système à l'instant suivant ($t+dt$) est obtenu en prenant une moyenne pondérée de ces gradients. Donnant ainsi une approche d'ordre 4 de l'état suivant.

$$sol[:, i + 1] = sol[:, i] + \frac{dt}{6} \cdot (k_1 + 2 \times k_2 + 2 \times k_3 + k_4)$$

Résultats :

- Temps : Le vecteur des instants calculés, représentant chaque point de mesure dans l'intervalle de simulation.
- Sol : La matrice contenant la solution numérique du vecteur d'état à chaque pas de temps.

Nous allons reprendre le système 1 que nous avons utilisé avec Euler explicite précédemment est voir si la précision de RK4 change quelque chose.

Voici la vidéo de notre simulation du pendule avec Runge-Kutta 4 ([lien](#)). Vous trouverez le code traitant ce système dans le fichier « Partie 1, q7 ».

$$\begin{cases} m_1 = m_2 = 1 \text{ kg} \\ l_1 = l_2 = 1 \text{ m} \\ Y_0 = (\frac{\pi}{2}, \frac{\pi}{2}, 0, 0)^T \\ N = 10^3 \end{cases}$$

Voici ce que l'on obtient avec Runge-Kutta 4 :

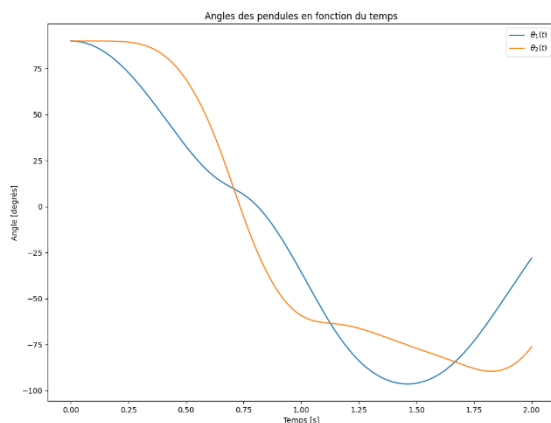


Figure 10 : Angles des pendules avec RK4 pour le système 1 (angle_plot_3.png)

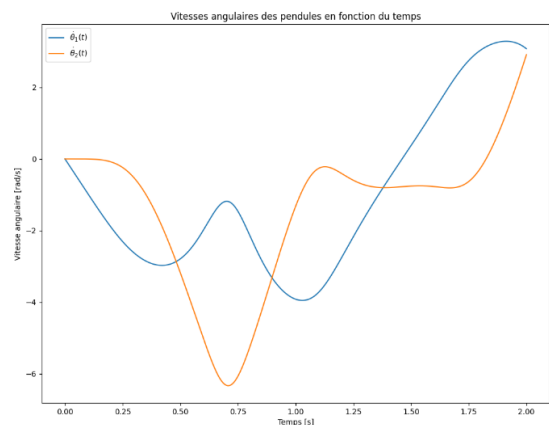


Figure 11 : Vitesses angulaires des pendules avec RK4 pour le système 1 (angle_plot_3.png)

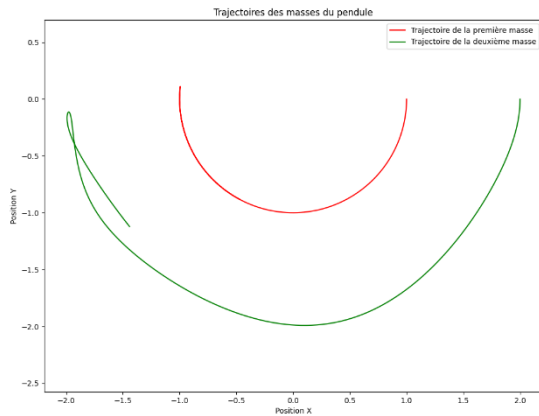


Figure 12 : Trajectoires des pendules avec RK4 pour le système 1 (trajectories_3.png)

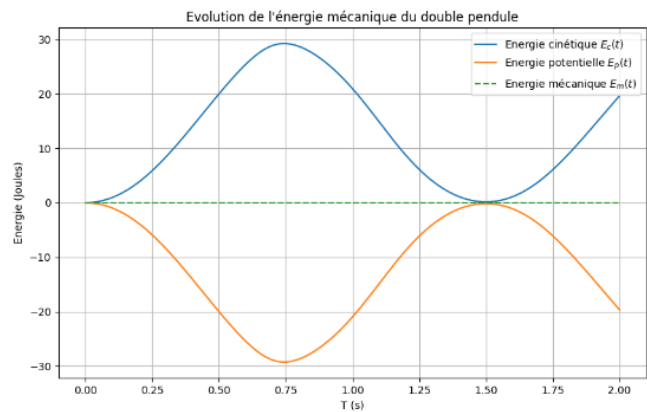


Figure 13 : Evolutions des énergies avec RK4 pour le système 1 (energy_evolution_3.png)

On trouve en effet des courbes similaires à ce que nous avons trouvé précédemment avec Euler explicite. La seule différence notable est que la différence d'énergie est de l'ordre de 10^{-10} , respectant ainsi mon critère fixé à 10^{-5} . Cela est tout à fait attendu et confirme de la précision particulièrement importante de RK4, une méthode d'ordre 4.

```
L'énergie du double pendule est conservée.
La différence d'énergie mécanique est 8.034282774073582e-10
```

Figure 14 : Sortie python d'analyse de la conservation de l'énergie du double pendule du premier système avec RK4

3. Méthode de Verlet

```
def verlet_method(Y, l1, l2, m1, m2, dt, T):
    N = int(T / dt)
    temps = np.linspace(0, T, N)
    sol = np.zeros((4, N))
    sol[:, 0] = Y

    for i in range(N - 1):
        th1, th2, dth1, dth2 = sol[:, i]
        _, _, s1, s2 = F(sol[:, i], l1, l2, m1, m2)

        # Mise à jour des positions
        th1_next = th1 + dt * dth1 + 0.5 * dt**2 * s1
        th2_next = th2 + dt * dth2 + 0.5 * dt**2 * s2

        # Calcul intermédiaire avec les nouvelles positions
        _, _, s1_new, s2_new = F([th1_next, th2_next, dth1, dth2], l1, l2, m1, m2)

        # Mise à jour des vitesses
        dth1_next = dth1 + 0.5 * dt * (s1 + s1_new)
        dth2_next = dth2 + 0.5 * dt * (s2 + s2_new)

        sol[:, i + 1] = [th1_next, th2_next, dth1_next, dth2_next]

    return temps, sol
```

Figure 15 : Algorithme python de la méthode de Verlet

La méthode de Verlet est particulièrement adaptée pour les systèmes mécaniques tel qu'un double pendule, car elle permet de conserver l'énergie et de garantir la stabilité des simulations sur des périodes prolongées. Voici une description détaillée de l'implémentation de la méthode de Verlet pour un double pendule.

Paramètres et Initialisation :

- Nombre de pas (N) : Calculé comme $\frac{T}{dt}$ détermine combien de fois la fonction sera évaluée et mise à jour au cours de la période T.
- Vecteur de temps (temps) : Un tableau numpy généré avec « np.linspace(0, T, N) », il représente les instants successifs à chaque pas de temps pour lesquels la solution est calculée.
- Matrice de solutions (sol) : Initialisée à zéro avec des dimensions $4 \times N$, cette matrice stocke les états successifs du système, ici deux angles et deux vitesses angulaires pour un système de deux pendules

Processus d'Intégration :

La méthode de Verlet pour le double pendule met à jour les positions et les vitesses à chaque pas de temps en deux étapes principales :

Premièrement, les nouvelles positions des pendules θ_1 et θ_2 sont calculées en utilisant les positions et vitesses actuelles ainsi que les accélérations.

$$\begin{aligned} \circ \quad \theta_{1,next} &= \theta_1 + dt \cdot \dot{\theta}_1 + \frac{1}{2} \cdot dt^2 \cdot \ddot{\theta}_1 \\ \circ \quad \theta_{2,next} &= \theta_2 + dt \cdot \dot{\theta}_2 + \frac{1}{2} \cdot dt^2 \cdot \ddot{\theta}_2 \end{aligned}$$

De plus, après la mise à jour des positions angulaires $\theta_{1,next}$ et $\theta_{2,next}$ nous procédons à un recalcul des accélérations angulaires. Cette étape utilise les nouvelles positions angulaires pour déterminer les accélérations angulaires $\ddot{\theta}_{1,next}$ et $\ddot{\theta}_{2,next}$. Ce processus est crucial car il permet d'obtenir des estimations plus précises des accélérations qui sont essentielles pour la mise à jour suivante des vitesses angulaires.

Les vitesses angulaires sont ensuite mises à jour en fonction des nouvelles accélérations.

$$\begin{aligned} \circ \quad \dot{\theta}_{1,next} &= \dot{\theta}_1 + \frac{1}{2} \cdot dt \cdot (\ddot{\theta}_1 + \ddot{\theta}_{1,next}) \\ \circ \quad \dot{\theta}_{2,next} &= \dot{\theta}_2 + \frac{1}{2} \cdot dt \cdot (\ddot{\theta}_2 + \ddot{\theta}_{2,next}) \end{aligned}$$

Résultats :

La fonction renvoie « temps » et « sol », le vecteur des temps et la matrice des solutions contenant les angles et vitesses angulaires à chaque instant.

Nous allons reprendre le système 1 que nous avons utilisé avec Euler explicite et RK4 précédemment afin d'observer son comportement avec la méthode de Verlet. Voici la vidéo de notre simulation du pendule avec Verlet ([lien](#)). Vous trouverez le code ainsi que tous les graphiques traitant ce système dans le fichier « Partie 2 ».

$$\begin{cases} m_1 = m_2 = 1 \text{ kg} \\ l_1 = l_2 = 1 \text{ m} \\ Y_0 = (\frac{\pi}{2}, \frac{\pi}{2}, 0, 0)^T \\ N = 10^3 \end{cases}$$

Les nouveaux graphiques que nous obtenons avec cette méthode sont semblables à première vue à ceux trouvés avec les deux autres méthodes

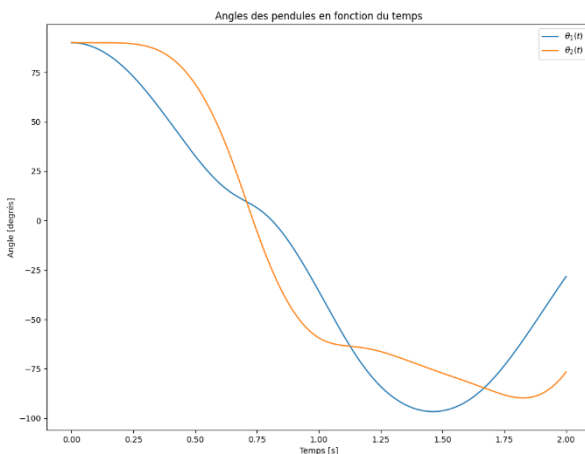


Figure 16 : Angles des pendules avec Verlet pour le système 1 (angle_plot_4.png)

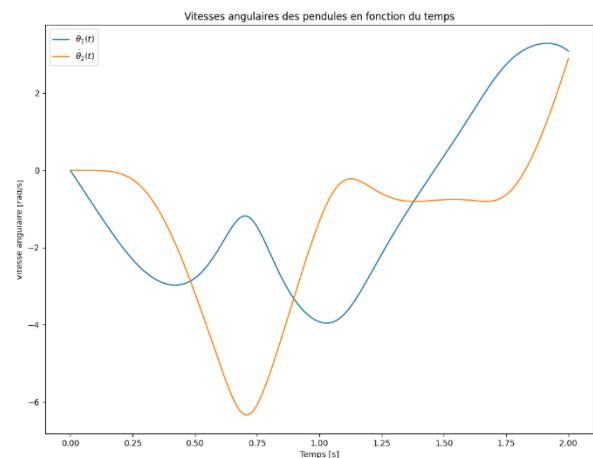


Figure 17 : Vitesses angulaires des pendules avec Verlet pour le système 1 (angle_plot_4.png)

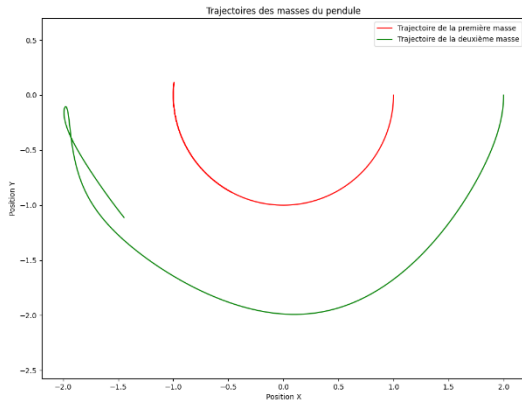


Figure 18 : Trajectoires des pendules avec Verlet pour le système 1 (trajectories_4.png)

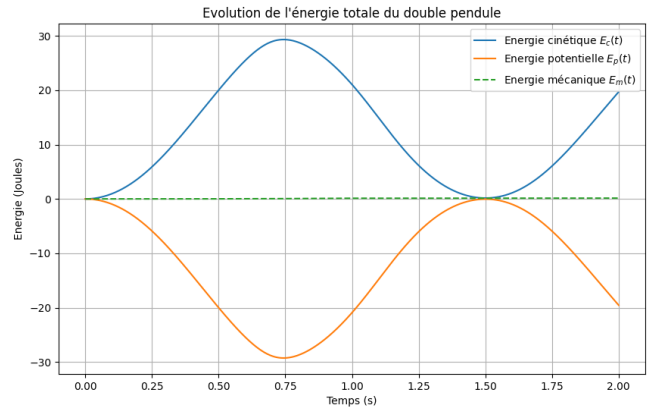


Figure 19 : Evolutions des énergies avec Verlet pour le système 1 (energy_evolution_4.png)

Nous trouvons une différence d'énergie mécanique de 0.055, ce qui est largement mieux qu'avec la méthode d'Euler explicite est son 0,16 mais nettement moins précis que la méthode RK4 avec sont 10^{-10} . Cela est tout à fait attendu car c'est une méthode d'ordre 2 donc plus précise que Euler explicite (ordre 1) et moins précise que Runge-Kutta 4 (ordre 4).

```
L'energie du double pendule est non conservée.
La différence d'energie mécanique est de 0.05534119638746919
```

Figure 20 : Sortie python d'analyse de la conservation de l'énergie du double pendule du premier système avec Verlet

Dans un second temps nous avons tracer l'évolutions des trajectoires de notre pendule double avec nos trois méthodes directement sur la même figure avec les paramètres du système 1 sur une période de 2 secondes afin de les comparer.

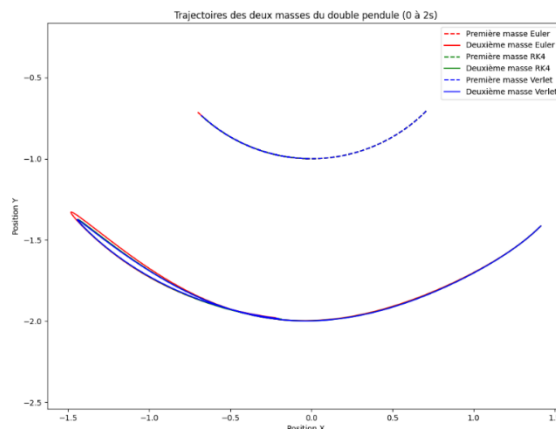


Figure 20 : Trajectoires des pendules de nos trois méthodes pour le système 1 (trajectories_3methodes.png)

La méthode d'Euler (lignes rouges pointillé pour la première masse et lignes rouges pleines pour la deuxième masse) semble produire des trajectoires moins précises, avec une déviation notable par rapport aux autres méthodes plus précises. En effet, l'erreur accumulée au fil du temps est plus importante, ce qui se traduit par des trajectoires qui divergent des trajectoires simulées par les autres méthodes.

La méthode Runge-Kutta d'ordre 4 (lignes vertes en pointillés pour la première masse et lignes vertes pleines pour la deuxième masse) produit des trajectoires très précises, avec une très bonne conservation des trajectoires attendues. Les trajectoires simulées par RK4 sont considérées comme de haute précision et sont souvent utilisées comme référence pour évaluer d'autres méthodes malgré la haute demande de ressource nécessaire pour cette méthode.

La méthode de Verlet (lignes bleues en pointillés pour la première masse et lignes bleues pleines pour la deuxième masse), tout en étant moins complexe que RK4, offre des résultats très précis et proches de ceux de RK4. La conservation de l'énergie est meilleure que celle de la méthode d'Euler, ce qui se traduit par des trajectoires stables et précises sur de plus longues périodes.

Ainsi, la méthode d'Euler, bien qu'elle soit simple à implémenter, montre ses limites en termes de précision, surtout sur des périodes plus longues ou pour des systèmes dynamiques complexes comme le double pendule. La méthode RK4 quant à elle, est la plus précise parmi les trois et sert souvent de référence. Cependant, elle est aussi la plus coûteuse en termes de calculs. Enfin, La méthode de Verlet, d'ordre 2, montre un excellent compromis entre simplicité et précision, offrant des trajectoires très proches de celles obtenues par RK4 avec une complexité de calcul moindre.

En résumé, pour des simulations nécessitant une très haute précision, la méthode RK4 est idéale, mais pour des simulations où la conservation de l'énergie sur de longues périodes est cruciale et où les calculs doivent être moins coûteux, la méthode de Verlet est souvent préférée. La méthode d'Euler, bien qu'utile pour des approximations rapides, n'est pas recommandée pour des simulations nécessitant une précision élevée.

IV. Mouvement chaotique et sensibilité par rapport aux conditions initiales

Dans cette partie, l'idée est d'étudier la variation des trajectoires des masses du double pendule selon les conditions initiales de lancement de celles-ci. Il est important à noter qu'afin d'éviter d'encombrer le rapport de graphiques, la partie 3 contient d'autres figures dans les dossiers telles que les espaces de phases, les vitesses angulaires en fonction du temps, l'évolution énergétique et les trajectoires. Seules les plus intéressantes seront expliquées. Comme vu dans la partie précédente, nous avons décidé de faire nos tests avec la méthode la plus précise, à savoir la méthode RK4.

1. Vitesse angulaire θ'_i en fonction de l'angle θ_i pour $i = 1, 2$

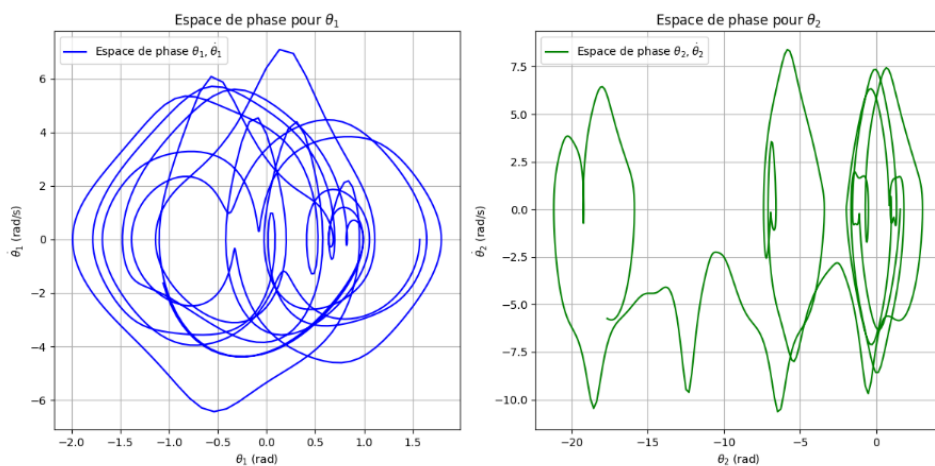


Figure 21 : Espace de phase pour θ_1 et θ_2 (phase_space_plot.png)

La vidéo du mouvement du pendule est présentée [ici](#) avec les paramètres demandés.

Les graphiques présentés illustrent les espaces de phase pour les angles θ_1 correspondant à l'angle effectué par la masse 1 notée m_1 de la Figure 1 et θ_2 celle de la masse 2 du pendule double, représentant le comportement dynamique du système à travers des trajectoires dans un espace bidimensionnel (\mathbb{R}^2) où les axes sont l'angle et sa vitesse angulaire correspondante.

Le graphique de gauche, en bleu, montre la relation entre θ_1 et sa vitesse $\dot{\theta}_1$, avec des courbes fermées et complexes qui suggèrent un mouvement périodique présentant des comportements chaotiques et complexes, dépendant des conditions initiales et des paramètres du système. Le graphique de droite, en vert, montre une relation tout aussi peu régulière entre θ_2

et $\dot{\theta}_2$. Cela indique que le mouvement de la deuxième masse est également peu prédictible et influencé par des facteurs comme les interactions avec la première masse et les conditions initiales du système, conduisant à un comportement relativement erratique et difficile à prévoir.

Ces visualisations nous permettent d'analyser la stabilité, la périodicité, et le chaos dans les systèmes dynamiques.

2. Vitesse angulaire $\dot{\theta}_i$ en fonction de l'angle θ_i pour une position très proche de celle d'équilibre

Voici les données représentant les conditions initiales. Ici, les conditions suivantes représentent un très léger écart autour de la position d'équilibre ($\approx 10^{-8}$) :

$$\begin{cases} m_1 = m_2 = 1 \text{ kg} \\ l_1 = l_2 = 1 \text{ m} \\ Y_0 = (10^{-8}, \sqrt{2} \times 10^{-8}, 0, 0)^T \\ N = 10^3 \end{cases}$$

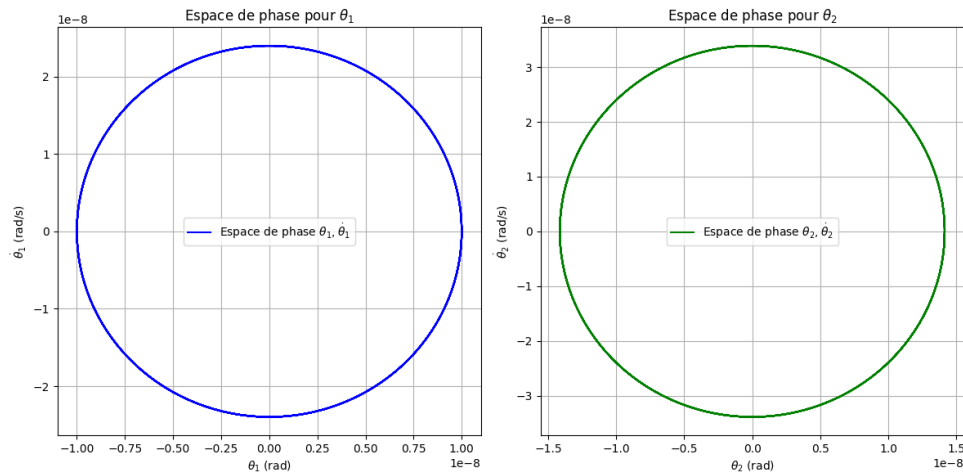


Figure 22 : Espace de phase pour θ_1 et θ_2 pour les secondes conditions (phase_space_plot.png)

Le tracé des 2 angles en ici sont parfaitement circulaires dans chacun des graphiques d'espaces de phase, ce qui indique un comportement régulier et prévisible du système sous de petites perturbations. Ce résultat suggère que le système retourne à son état initial après chaque oscillation, ce qui caractérise une stabilité harmonique simple autour de la position d'équilibre. Les cercles représentent des oscillations continues avec une amplitude constante, montrant ainsi que les deux angles oscillent de manière sinusoïdale autour de leur position de repos sans s'éloigner ou se rapprocher de manière significative de cette position au fil du temps. Nous avons

une nouvelle fois simulé ce cas de figure, afin de voir l'animation pour ces conditions, je vous invite à utiliser ce [lien](#).

Désormais modifions très légèrement les conditions initiales et examinons les solutions avec les données ci-dessous :

$$\begin{cases} m_1 = m_2 = 1 \text{ kg} \\ l_1 = l_2 = 1 \text{ m} \\ Y_0 = (10^{-8}, 10^{-8}, 0, 0)^T \\ N = 10^3 \end{cases}$$

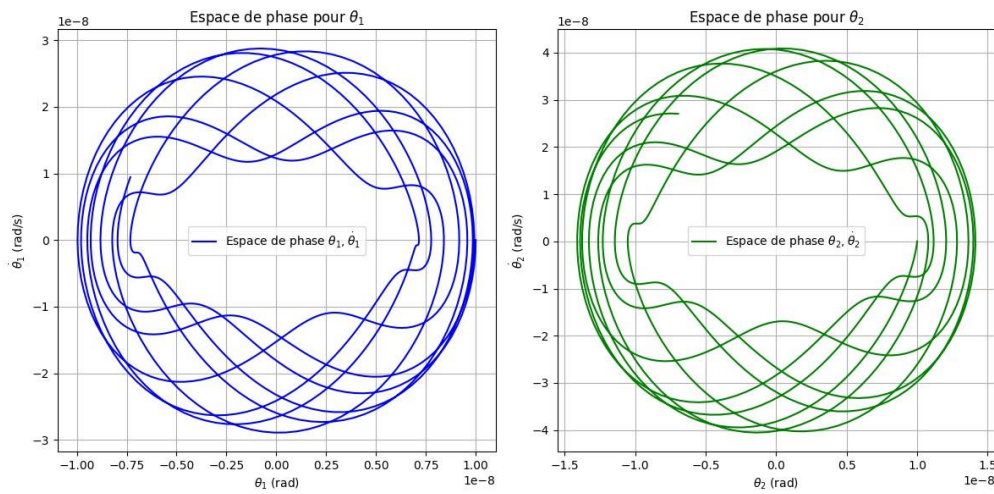


Figure 23 : Espace de phase pour θ_1 et θ_2 pour les troisièmes conditions (phase_space_plot.png)

Contrairement aux traçages circulaires précédemment observés, nous observons ici des trajectoires qui forment des motifs moins réguliers mais de façon relativement circulaire. Cela indique un comportement non linéaire plus prononcé et une réponse plus sensible aux modifications des conditions initiales.

Par rapport à la configuration précédente, où les trajectoires étaient des cercles parfaits indiquant une dynamique stable et prévisible, ces nouvelles trajectoires suggèrent que le système présente une dépendance sensible aux conditions initiales, caractéristique d'un comportement chaotique sous certaines conditions.

Cette sensibilité élevée aux conditions initiales pourrait signifier que même de petites variations peuvent entraîner des changements significatifs dans le comportement dynamique du système, ce qui est primordial pour des applications nécessitant une haute précision ou pour des études sur la stabilité et le contrôle de tels systèmes. Voici l'animation, restant semblable à la précédente et accessible via ce [lien](#).

3. Évolution du système vers un mouvement chaotique

Maintenant que nous observons le phénomène de chaos, caractérisé par une extrême sensibilité aux conditions initiales — souvent illustré par l'analogie de « l'effet papillon » — il est opportun de réaliser divers tests pour évaluer l'impact de cette sensibilité à travers des simulations :

$$\begin{cases} m_1 = m_2 = 1 \text{ kg} \\ l_1 = l_2 = 1 \text{ m} \\ Y_0 = (\alpha, \alpha, 0, 0)^T \\ P = 10^3 \end{cases} \quad \text{avec } \alpha \in \{10^{-9}, 10^{-7}, 0.005, 0.05, 1, 1.2, 1.5, 1.7, 1.9, 2\}$$

Voici quelques-uns des tracés intéressants des espaces de phases pour différent α :

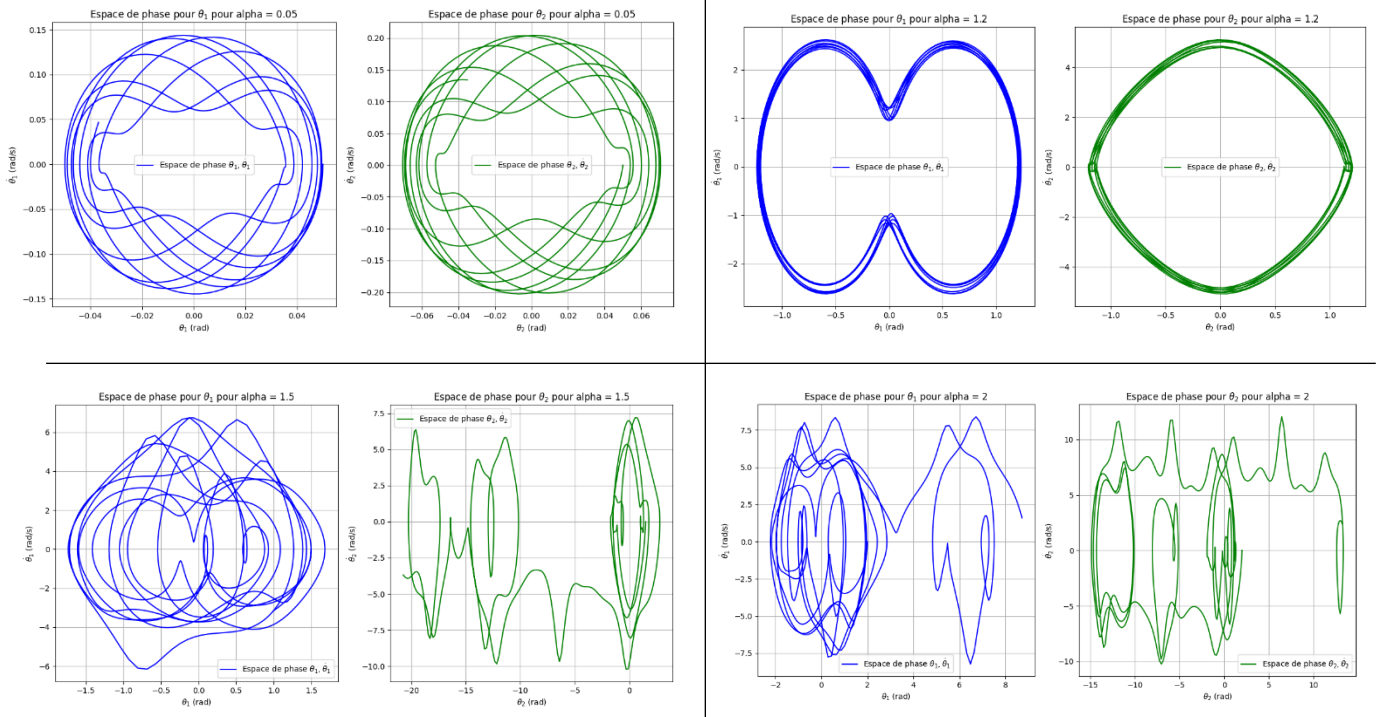


Figure 24 : Espace de phase pour θ_1 et θ_2 pour des valeurs de α (phase_space_plot_alpha.png)

Voici également, quelques-unes des trajectoires affichées de la même façon que les 4 graphiques précédents :

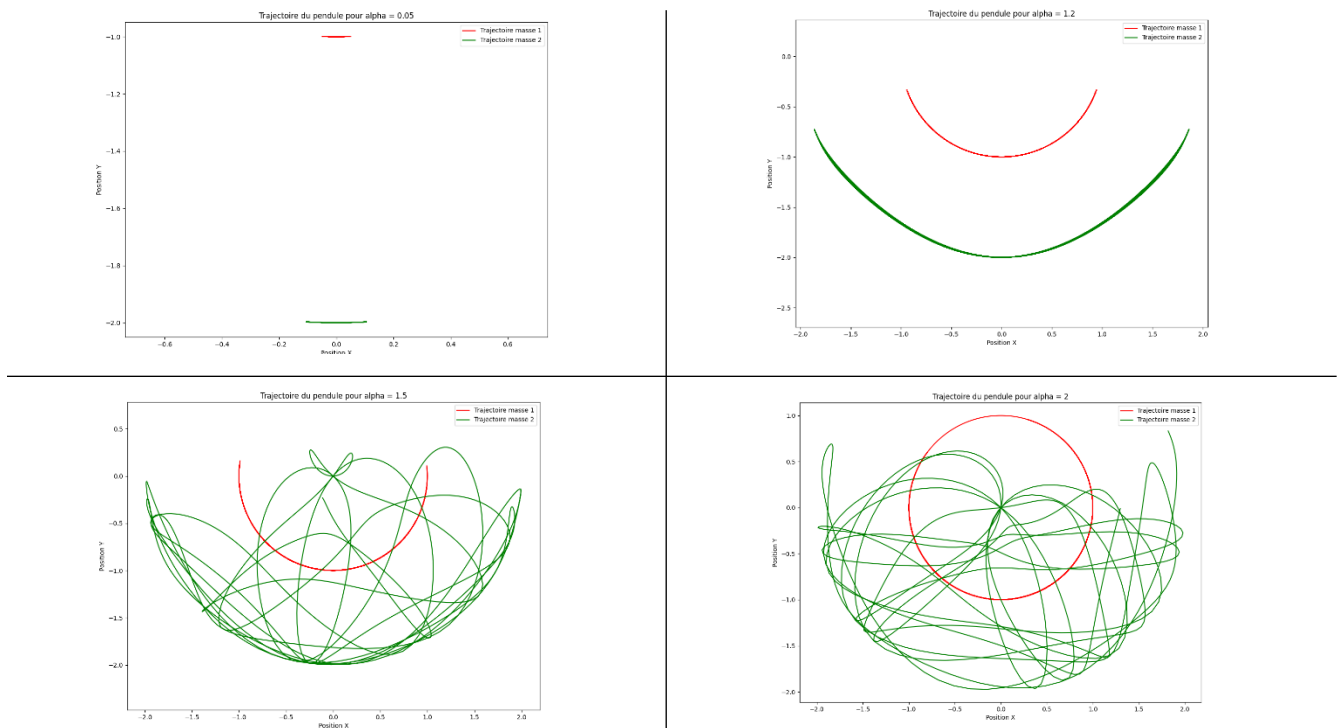


Figure 25 : Trajectoires des différentes masses du double pendule pour des valeurs de α (trajectories_alpha.png)

L'analyse des graphiques montre différentes trajectoires dans l'espace de phase (θ_i, θ_i') pour un pendule sous différentes valeurs du paramètre α . Ces graphiques illustrent comment le comportement du pendule évolue avec des variations d' α , suggérant des transitions du mouvement régulier au mouvement chaotique.

Pour les petites valeurs d' α (0.05, 0.005) : Les courbes C_i sont des ellipses fermées ou des cercles, indiquant un mouvement périodique et régulier. Le pendule oscille de manière stable avec une petite amplitude autour de sa position d'équilibre. La régularité et la prédictibilité sont évidentes ici.

Augmentation d' α (1, 1.2) : Les trajectoires commencent à montrer plus de complexité. Bien que toujours fermées, les formes deviennent moins régulières et plus étirées. Cela suggère une sensibilité déjà importante aux conditions initiales, typique d'un système non-linéaire où de petites modifications de α ou des conditions initiales peuvent considérablement changer la trajectoire.

Pour des valeurs plus élevées d' α (1.5, 2) : Les graphiques révèlent des comportements significativement plus complexes et irréguliers. Les courbes C_i deviennent de plus en plus tordues et entrelacées, indiquant un mouvement imprévisible et potentiellement chaotique. Ces trajectoires montrent des intersections multiples et des formes qui ne sont plus symétriques ou régulières, signe d'une dynamique imprévisible et chaotique.

Valeurs extrêmes d' α (2 et plus) : Les trajectoires sont extrêmement irrégulières avec des fluctuations violentes en amplitude et en forme, illustrant un comportement chaotique. Le système est très sensible aux conditions initiales, une caractéristique des systèmes chaotiques, où de petites variations dans les conditions initiales ou les paramètres peuvent mener à des dynamiques complètement différentes.

En résumé, l'augmentation du paramètre α mène à un passage du mouvement périodique et prévisible à un comportement de plus en plus erratique et chaotique. Ce comportement est typique des systèmes non linéaires où les effets des non-linéarités deviennent dominants à mesure que le paramètre de contrôle (α ici) augmente. Les structures des courbes C_i montrent clairement cette transition vers un comportement chaotique, marquée par l'augmentation de la complexité et de l'irrégularité des trajectoires dans l'espace de phase plus la condition initiale s'éloigne de la position d'équilibre.

Voici le [lien](#) permettant d'accéder à l'animation du double pendule avec les 10 angles de départs différents fonctionnant simultanément sur le même graphique.

V. Conclusion

Cette étude a exploré la dynamique complexe d'un double pendule en se concentrant sur sa modélisation physique, les méthodes numériques pour résoudre les équations différentielles ordinaires (EDO), et les effets chaotiques du système. En dérivant les équations de mouvement du double pendule à partir des lois de Newton et en les formulant sous forme de problème de Cauchy, nous avons établi une base solide pour la simulation numérique du système.

Nous avons appliqué plusieurs méthodes numériques, notamment la méthode d'Euler explicite, la méthode de Runge-Kutta d'ordre 4 (RK4) et la méthode de Verlet. La méthode d'Euler explicite, bien que simple, a montré des limites en termes de précision et de conservation de l'énergie. La méthode de Runge-Kutta d'ordre 4 a offert une meilleure précision et stabilité avec une erreur d'énergie très faible. La méthode de Verlet, particulièrement adaptée aux systèmes mécaniques, a démontré son efficacité en conservant l'énergie sur des périodes prolongées bien qu'elle soit moins efficace que notre Runge-Kutta 4.

L'analyse du mouvement chaotique a révélé la sensibilité du double pendule aux conditions initiales. De petites variations dans les conditions initiales ont conduit à des comportements dynamiques très différents, soulignant le caractère imprévisible et complexe du système.

Cette étude a non seulement mis en évidence les défis et les solutions pour modéliser et simuler un système dynamique complexe, mais a aussi ouvert des perspectives pour des recherches futures. L'optimisation des méthodes numériques, les études expérimentales pour valider les simulations, et l'application des techniques développées à d'autres systèmes dynamiques non linéaires constituent des pistes prometteuses pour approfondir notre compréhension des phénomènes chaotiques.