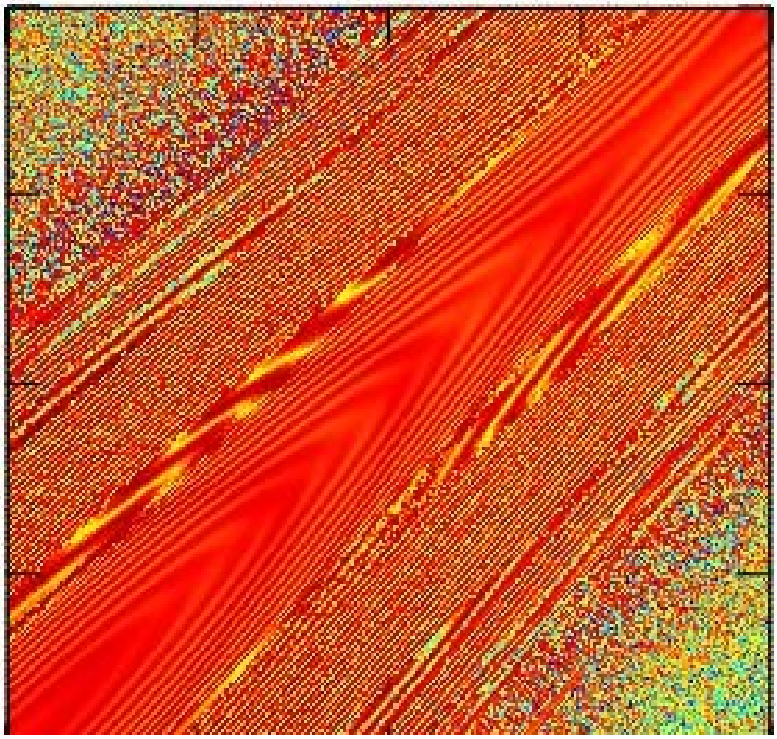
Le pendule double, un système chaotique



Noémie Jaquier Ma2-3

Travail de maturité 2009-2010
Gymnase intercantonal de la Broye

Supervisé par M. E. Jaquet

Résumé

Ce travail présente une analyse détaillée d'un cas particulier du système du pendule double, visant à démontrer la présence de chaos dans le système. Les équations caractérisant le pendule double sont établies, puis appliquées. Les méthodes de résolution numériques des équations différentielles sont présentées en détails. On observe ensuite la présence de chaos grâce à différents diagrammes illustrant la forte sensibilité aux conditions initiales, et finalement le chaos est quantifié pour des énergies différentes par le calcul de l'exposant de Lyapunov.

Table des matières

1. Introduction …. 4

2. Détermination des équations du système
2.1. Description du système 5
2.2. Description des forces agissant sur les masses …. 6
2.3. Application de la 2ème loi de Newton 6

3. Traitement numérique des équations différentielles ordinaires couplées
3.1. Les équations différentielles ….9
3.2. La méthode d'Euler ….9
3.3. La méthode du point milieu ….10
3.4. La méthode de Runge-Kutta ….11

4. Résolution numérique de ces équations
4.1. Résolution avec MuPAD …. 12
4.2. Résolution à l'aide d'un programme C …. 13

5. Représentation cinématique du pendule double
5.1. Représentation statique ….14
5.2. Animation ….15

6. Lois de conservation dans le système ….15

7. Utilisation des lois de conservation dans le système pour valider les calculs 17

8. Évaluation de la précision des calculs ….18

9. Divergence temporelle du pendule double
9.1. Choix de la représentation ….19
9.2. Représentation sous forme de diagrammes ….19
9.3. Représentation des résultats sous forme de diagrammes ….20

10. Évaluation de l'exposant de Lyapunov
10.1. Définition de l'exposant de Lyapunov ….24
10.2. Évaluation d'exposants de Lyapunov locaux ….24
10.3. Évaluation de l'exposant de Lyapunov maximal pour l'ensemble du système …. 25

11. Conclusion ….29

12. Bibliographie ….30

13. Annexes ….31

14. Attestation sur l'honneur ….40

1. Introduction

En 1812, Pierre-Simon Laplace, mathématicien, astronome et physicien, déclarait qu'il suffirait de « connaître toutes les forces qui mettent en mouvement la nature, et toutes les positions de tous les objets dont la nature est composée à un instant donné » pour connaître le passé et prédire le futur sans ambiguïté. On reconnaît aujourd'hui que certains systèmes dynamiques, mêmes simples, deviennent imprévisibles au fur et à mesure de leur évolution dans le temps. Ces systèmes, dits chaotiques sont déterministes, c'est-à-dire qu'ils obéissent aux lois de la physique classique, mais évoluent de façon totalement différente pour une très faible variation des paramètres initiaux. Ils sont ainsi imprévisibles à long terme, à cause de cette caractéristique que l'on nomme « sensibilité aux conditions initiales ».

Les exemples de systèmes chaotiques sont nombreux: parmi eux on retrouve notamment les prévisions météorologiques, pour lesquelles on parle de l'effet papillon, ou le mouvement des planètes.
La métaphore de l'effet papillon consiste à supposer qu'un simple battement d'ailes de ce petit insecte pourrait décider de la création, ou non, d'une tempête à l'autre bout du monde. En réalité, les calculs effectués par les physiciens prouvent qu'un simple battement d'aile de papillon n'a pas assez d'influence pour perturber le système atmosphérique de manière aussi significative. Mais l'image donnée par cette métaphore illustre bien le principe des systèmes chaotiques: un changement infime des conditions initiales amène une différence considérable dans l'évolution du système.

Ce travail se focalise sur un système chaotique en particulier: le pendule double. Il s'agit d'un pendule simple au bout duquel on fixe un deuxième pendule. L'ensemble est alors mis en mouvement depuis une certaine configuration, déterminée par les conditions initiales.

Afin de pouvoir décrire le système, le chapitre 2 détermine les équations le caractérisant. Les différentes méthodes permettant de résoudre ces équations sont ensuite décrites au chapitre 3, puis l'une d'entre elles est appliquée au chapitre 4. Par la suite, le pendule double est représenté graphiquement au chapitre 5. Les chapitres 6 et 7 étudient les lois de conservation de l'énergie, et vérifient l'exactitude des équations caractérisant le système et de leur résolution. Grâce à ces lois, on peut ensuite observer la précision fournie par les calculs au chapitre 8. Le chapitre 9 montre, quant à lui, la sensibilité aux conditions initiales du pendule double et sa divergence temporelle grâce à différents diagrammes. Pour terminer, le chapitre 10 caractérise enfin la quantité de chaos présente dans le système par l'évaluation de l'exposant de Lyapunov.

2. Détermination des équations du système

2.1. Description du système

Le système du pendule double est composé de deux masses: la première, m_1, est rattachée à la première tige l_1 et la deuxième, m_2, est rattachée à la deuxième tige l_2. Le point 1 est le point où se trouve m_1 au temps t, et le point 2, celui où se situe m_2 au même temps t.
Afin de faciliter la description du système, on considérera que les deux masses sont ponctuelles, et que les tiges ne peuvent ni se plier, ni se déformer d'une quelconque manière.

Les paramètres dont la connaissance est nécessaire afin de décrire précisément la cinématique d'un système sont sa position et sa vitesse pour tout temps t. Dans le cas du pendule double, il faut donc connaître quatre paramètres: la position et la vitesse de m_1, ainsi que la position et la vitesse de m_2. Les positions et les vitesses peuvent être définies par un paramètre unique qui sont respectivement un angle et une vitesse angulaire. L'angle θ_1 est l'angle formé entre l'axe y et l_1, et l'angle θ_2 est celui formé entre l'axe y et l_2. Puisque la vitesse est la dérivée de la position, la vitesse angulaire de m_1 est $\dot{\theta}_1$ et la vitesse angulaire de m_2 est $\dot{\theta}_2$.

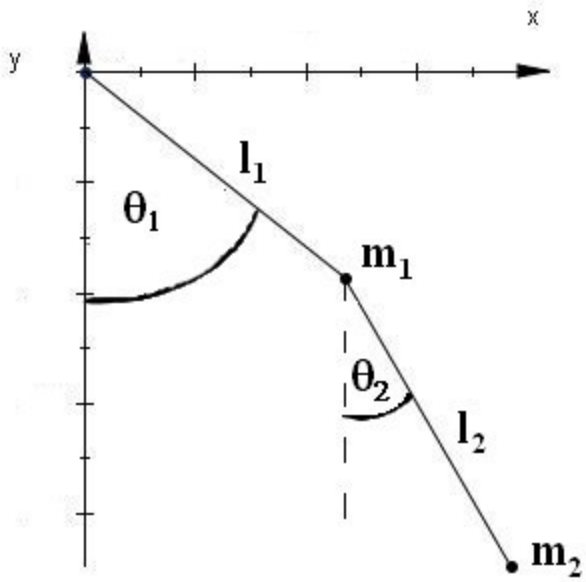


Figure 1: Définition du pendule double à l'aide des variables utilisées pour décrire l'état du système

Afin de faciliter les calculs, on traitera un cas idéal, en supposant que le pendule double n'est pas influençable par les éléments extérieurs, hors la force de gravitation. L'agissement de forces extérieures, dues, par exemple, aux frottements, sera donc négligé.
.
Il découle de cette hypothèse que le système conserve toute l'énergie mécanique et qu'il n'y a aucune dissipation, notamment d'énergie thermique. Les seules transformations d'énergie que l'on rencontre sont celles d'énergie potentielle en énergie cinétique (de mouvement), et inversement.

Le système possède ainsi trois degrés de liberté. En effet, lorsque trois des quatre paramètres décrivant le système sont connus, le quatrième peut être déterminé car toute l'énergie mécanique est conservée. L'énergie mécanique totale est définie par les conditions initiales.
Le but est de déterminer les équations du système qui permettront de connaître la position et la vitesse de chacune des masses à n'importe quel moment.

2.2. Description des forces agissant sur les masses

Afin de caractériser les forces agissant sur le système, on le décompose en deux parties: on décrit ainsi les forces appliquées sur chacune des masses séparément.
Les forces agissant sur m_1 sont la force de pesanteur $\vec{F}_{p1}$, une force de tension sur la tige (en étirement ou en compression) $\vec{T}_1$ et la force due à m_2 que l'on appelle $\vec{T}_2$. La force de pesanteur $\vec{F}_{p1}$ est dirigée vers le centre de la terre. Par définition, elle est égale à m_1*g, g étant l'accélération sur terre due à la gravitation, ayant une valeur de 9.81 m/s^2. La force $\vec{T}_1$ agit dans l'axe de l_1 vers le point de coordonnées (0;0) où est fixé l'extrémité de l_1, et $\vec{T}_2$ est dirigée dans l'axe de l_2.
Les forces agissant sur m_2 sont la force de pesanteur $\vec{F}_{p2}$ et la force de tension $\vec{T}_2$. La force de pesanteur $\vec{F}_{p2}$ agit parallèlement à $\vec{F}_{p1}$ et la force $\vec{T}_2$ est dirigée vers m_1 le long de l_2. Cette deuxième force $-\vec{T}_2$ est égale, mais opposée à la force de même nom agissant sur m_1.
Toutes ces forces sont décrites par des vecteurs, dont les origines sont les positions des masses m_1 et m_2.

L'accélération que subit chacune des deux masses est représentée par un vecteur $\vec{a}_1$ pour m_1 et par un vecteur $\vec{a}_2$ pour m_2. L'origine de ces vecteurs est la position de chacune des masses.

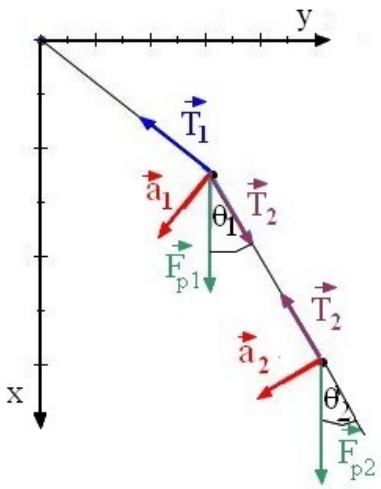


Figure 2: Description des forces agissant sur les deux masses

2.3. Application de la 2ème loi de Newton

La deuxième loi de Newton énonce que la force résultante de toutes les forces agissant sur un objet est égale à la masse de cet objet multipliée par son accélération.
On peut appliquer cette loi individuellement sur chaque masse du pendule double.

$$(1) \qquad \vec{T}_1 - \vec{T}_2 + m_1 \vec{g} = m_1 \vec{a}_1$$

$$(2) \qquad \vec{T}_2 + m_2 \vec{g} = m_2 \vec{a}_2$$

Le pendule double est un système à deux dimensions, les vecteurs décrivant les forces en jeu sont donc des vecteurs possédant deux composantes. Pour des raisons pratiques, l'axe vertical sera considéré comme le premier axe, et l'axe horizontal comme le deuxième.

Afin de déterminer la valeur des accélérations des masses, il faut utiliser la notion de dérivée. En effet, l'accélération est la dérivée de la fonction définissant la vitesse, étant elle-même la dérivée de la fonction définissant la position.

La position de m_1 est décrite pour tout angle θ_1 par le vecteur $\vec{x}_1 = l_1 \begin{pmatrix} \cos(\theta_1) \\ \sin(\theta_1) \end{pmatrix}$. Sa vitesse est alors décrite par la dérivée de $\vec{x}_1$, c'est-à-dire par le vecteur $\vec{v}_1 = l_1 \dot{\theta}_1 \begin{pmatrix} -\sin(\theta_1) \\ \cos(\theta_1) \end{pmatrix}$. Son accélération est la dérivée de $\vec{v}_1$, donc $\vec{a}_1 = l_1 \left(\ddot{\theta}_1 \begin{pmatrix} -\sin(\theta_1) \\ \cos(\theta_1) \end{pmatrix} - \dot{\theta}_1^2 \begin{pmatrix} \cos(\theta_1) \\ \sin(\theta_1) \end{pmatrix} \right)$.

De manière analogue, on peut déterminer l'accélération $\vec{a}_{12}$ de m_2. Sa position est décrite pour tout angle θ_2 par le vecteur $\vec{x}_{12} = l_2 \begin{pmatrix} \cos(\theta_2) \\ \sin(\theta_2) \end{pmatrix}$, sa vitesse par le vecteur $\vec{v}_{12} = l_2 \dot{\theta}_2 \begin{pmatrix} -\sin(\theta_2) \\ \cos(\theta_2) \end{pmatrix}$, et son accélération par le vecteur $\vec{a}_{12} = l_2 \left(\ddot{\theta}_2 \begin{pmatrix} -\sin(\theta_2) \\ \cos(\theta_2) \end{pmatrix} - \dot{\theta}_2^2 \begin{pmatrix} \cos(\theta_2) \\ \sin(\theta_2) \end{pmatrix} \right)$.

Comme m_2 est rattachée à m_1, sa position, sa vitesse et son accélération sont en fait la somme des vecteurs de position, de vitesse et d'accélération des deux masses.

$$\vec{x}_2 = \vec{x}_1 + \vec{x}_{12}$$
$$\vec{v}_2 = \vec{v}_1 + \vec{v}_{12}$$
$$\vec{a}_2 = \vec{a}_1 + \vec{a}_{12}$$
$$\vec{x}_2 = l_1 \begin{pmatrix} \cos(\theta_1) \\ \sin(\theta_1) \end{pmatrix} + l_2 \dot{\theta}_2 \begin{pmatrix} -\sin(\theta_2) \\ \cos(\theta_2) \end{pmatrix}$$
$$\vec{v}_2 = l_1 \dot{\theta}_1 \begin{pmatrix} -\sin(\theta_1) \\ \cos(\theta_1) \end{pmatrix} + l_2 \dot{\theta}_2 \begin{pmatrix} -\sin(\theta_2) \\ \cos(\theta_2) \end{pmatrix}$$
$$\vec{a}_2 = l_1 \left(\ddot{\theta}_1 \begin{pmatrix} -\sin(\theta_1) \\ \cos(\theta_1) \end{pmatrix} - \dot{\theta}_1^2 \begin{pmatrix} \cos(\theta_1) \\ \sin(\theta_1) \end{pmatrix} \right) + l_2 \left(\ddot{\theta}_2 \begin{pmatrix} -\sin(\theta_2) \\ \cos(\theta_2) \end{pmatrix} - \dot{\theta}_2^2 \begin{pmatrix} \cos(\theta_2) \\ \sin(\theta_2) \end{pmatrix} \right)$$

En ajoutant les composantes des vecteurs dans l'équation déterminée par la deuxième loi de Newton appliquée sur m_2, on obtient l'équation suivante:

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

$\vec{T}_2$ est donné par une constante inconnue T_2 multipliant un vecteur indiquant sa direction.
De manière analogue, la force de pesanteur est décrite par m_2 multipliant g et un vecteur d'indication de direction.

Comme la valeur de T_2 est inconnue, il est nécessaire de rendre l'équation indépendante de cette constante. On applique alors une rotation d'un angle - θ_2 sur l'ensemble du système, afin que la composante de $\vec{T}_2$ sur le second axe soit nulle. Cette rotation est effectuée en multipliant chaque élément de l'équation donnée ci-dessus par la matrice $\begin{pmatrix} \cos(\theta_2) & \sin(\theta_2) \\ -\sin(\theta_2) & \cos(\theta_2) \end{pmatrix}$.

En sachant que $\sin(A-B) = \sin(A)\cos(B) - \cos(A)\sin(B)$
et que $\cos(A-B) = \cos(A)\cos(B) + \sin(A)\sin(B)$, on obtient, après rotation, l'équation suivante:

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

En tenant uniquement compte des composantes du second axe, on obtient une équation où tous les éléments sont connus.

$$(3) \qquad m_2 g \sin(\theta_2) + m_2 l_1 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - m_2 l_1 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) + m_2 l_2 \ddot{\theta}_2 = 0$$

En agissant de manière analogue avec l'addition des équations (1) et (2), il sera possible d'obtenir une deuxième équation semblable, qui sera nécessaire par la suite.
On additionne (1) et (2) de façon à ce que le vecteur $\vec{T}_2$ disparaisse.

$$(1)+(2) \qquad \vec{T}_1 + m_1 \vec{g} + m_2 \vec{g} = m_1 \vec{a}_1 + m_2 \vec{a}_2$$

$$(1)+(2) \qquad \vec{T}_1 + (m_1 + m_2) \vec{g} = (m_1 + m_2) \vec{a}_1 + m_2 \vec{a}_{12}$$

En tenant compte des composantes des vecteurs, l'équation prend la forme suivante:

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

On applique ensuite une rotation d'un angle – θ_1 sur l'ensemble du système afin que la composante de $\vec{T}_1$ sur le second axe soit nulle. Cette rotation est effectuée en multipliant chaque élément de l'équation donnée ci-dessus par la matrice $\begin{pmatrix} \cos(\theta_1) & \sin(\theta_1) \\ -\sin(\theta_1) & \cos(\theta_1) \end{pmatrix}$.

On obtient alors l'équation suivante:

$$(1)+(2)$$

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

En tenant uniquement compte des composantes du second axe, on obtient une deuxième équation où tous les éléments sont connus.

$$(4) \qquad (m_1 + m_2) g \sin(\theta_1) + (m_1 + m_2) l_1 \ddot{\theta}_1 + m_2 l_2 \ddot{\theta}_2 \cos(\theta_1 - \theta_2) + m_2 l_2 \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) = 0$$

Certaines transformations sont ensuite nécessaires afin de pouvoir résoudre le système d'équations suivant.

$$(3) \qquad m_2 g \sin(\theta_2) + m_2 l_1 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - m_2 l_1 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) + m_2 l_2 \ddot{\theta}_2 = 0$$

$$(4) \qquad (m_1 + m_2) g \sin(\theta_1) + (m_1 + m_2) l_1 \ddot{\theta}_1 + m_2 l_2 \ddot{\theta}_2 \cos(\theta_1 - \theta_2) + m_2 l_2 \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) = 0$$

Dans l'équation (3), on définit l'accélération angulaire $\ddot{\theta}_2$.

$$(3) \qquad \ddot{\theta}_2 = \frac{-g \sin(\theta_2) - l_1 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) + l_1 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2)}{l_2}$$

Puis, on substitue $\ddot{\theta}_2$ dans (4) afin que celle-ci ne dépende plus de cette variable.

$$(4) \qquad (m_1 + m_2) l_1 \ddot{\theta}_1 + m_2 \cos(\theta_1 - \theta_2)(-g \sin(\theta_2) - l_1 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) + l_1 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2))$$
$$+ m_2 l_2 \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) + (m_1 + m_2) g \sin(\theta_1) = 0$$

A partir de l'équation ci-dessus, on peut définir l'accélération angulaire $\ddot{\theta}_1$.

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

Pour terminer, on substitue $\ddot{\theta}_1$ dans (3) afin que $\ddot{\theta}_2$ ne dépende plus de $\ddot{\theta}_1$.

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

On obtient ainsi les deux équations différentielles du mouvement, basées sur la deuxième loi de Newton et définissant la cinématique du système.

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

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

3. Traitement numérique des équations différentielles ordinaires couplées

3.1. Les équations différentielles

Une équation différentielle ordinaire est une équation où la dérivée $f^{(n)}$ d'une fonction f inconnue est décrite par la fonction elle-même et par une ou plusieurs de ses dérivées $f^{(m)}$ où m est un entier appartenant à l'intervalle {1;n-1}.
L'ordre d'une équation différentielle est l'ordre de la plus haute dérivée y apparaissant, c'est-à-dire n.
Deux équations différentielles couplées sont deux équations où deux dérivées d'ordre n de deux fonctions f et g sont chacune décrites par f et g, ainsi que par une ou plusieurs dérivées de différents ordres $f^{(m)}$ et $g^{(m)}$.

Les équations du mouvement du pendule double sont deux équations différentielles ordinaires couplées d'ordre 2: $\ddot{\theta}_1$ et $\ddot{\theta}_2$ sont exprimées en fonction de θ_1 , θ_2 , $\dot{\theta}_1$ et $\dot{\theta}_2$. Lorsque le système évolue dans le temps, ces paramètres deviennent des fonctions de t.

On ne peut résoudre toutes les équations différentielles selon les méthodes analytiques de résolution: il faut utiliser des méthodes numériques. De plus, il est nécessaire de connaître les conditions initiales, c'est-à-dire les valeurs initiales de toutes les variables servant à exprimer $f^{(n)}$.

3.2. La méthode d'Euler

La méthode d'Euler est la méthode de résolution d'équations différentielles numérique la plus simple. Elle permet de résoudre des équations différentielles du premier ordre, c'est-à-dire de la forme $\dot{y}(x)=f(x,y(x))$.

Il est nécessaire de connaître les conditions initiales, donc la valeur de $y(0)$. Grâce à cela, on peut alors calculer la valeur de $\dot{y}(0)$.
La dérivée d'une fonction en un point est égale à la pente de la tangente L à cette fonction en ce même point. Proche de ce point, les trajectoires décrites par la fonction et par la tangente sont presque confondues.
Comme représenté dans la figure 3, P et Q sont des points de la courbe y(x) et Δy est leur différence d'ordonnée. P et R appartiennent à L(x), qui est la tangente à y(x) en x=0, et dy est leur différence d'ordonnée.
On ne peut pas connaître précisément les coordonnées du point Q, qui sont $(dx;y(dx))$, car on ne connaît pas la fonction $y(x)$. Par contre, la pente de L(x) est égale à $\dot{y}(0)$, qui peut être calculée grâce à $y(0)$ dont on connaît la valeur. On peut donc déterminer que les coordonnées de R sont $(dx;\dot{y}(0)*dx)$.
Pour un dx très petit, les points Q et R sont très proches: on approxime donc Δy=dy.

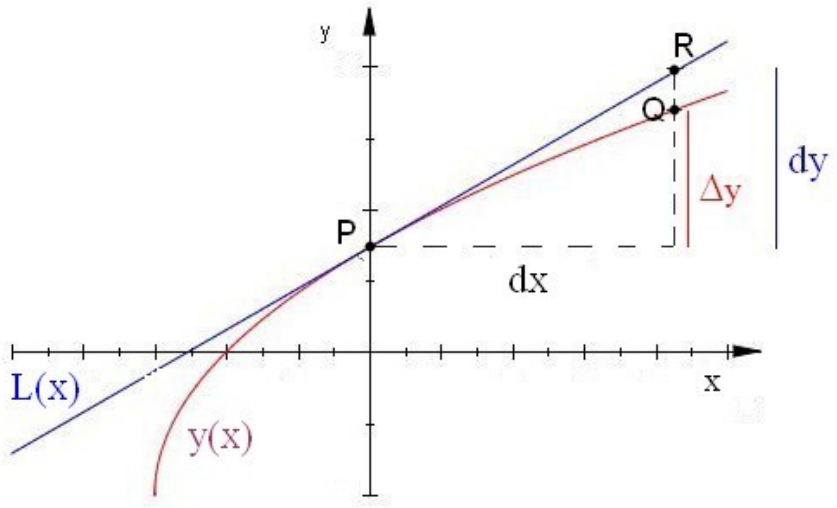


Figure 3: Représentation de la méthode d'Euler

La méthode d'Euler consiste ainsi à calculer de manière itérative, en connaissant la valeur de $y(0)$, les valeurs de la fonction y selon l'équation suivante.

$$y_{n+1}=y_n+dx*f(x_n,y_n)$$

Le désavantage de cette méthode réside dans la faible précision des résultats obtenus. Néanmoins, elle va servir à expliquer d'autres méthodes de résolutions numériques plus précises et plus fiables.

3.3. La méthode du point milieu

La méthode du point milieu est plus précise que la méthode d'Euler sur laquelle elle se base. Elle permet également de résoudre des équations différentielles du premier ordre, donc de la forme $\dot{y}(x)=f(x,y(x))$ et nécessite la connaissance des conditions initiales.

Pour chaque itération, on calcule d'abord grâce à la méthode d'Euler la valeur de la différence d'ordonnée $k1$ entre (x_n,y_n) et un point dont l'abscisse est x_n+dx en utilisant la pente de la tangente T_1 à la fonction en (x_n,y_n) . La pente peut être calculée grâce à notre connaissance des valeurs initiales de x et y. La différence d'ordonnée est alors donnée par l'équation $k_1=dx*f(x_n,y_n)$.

Cela permet ensuite de calculer la pente de la tangente T_2 à la fonction en un point milieu $(x_n+\frac{1}{2}dx, y_n+\frac{1}{2}k_1)$. Puis, on calcule à nouveau la différence d'ordonnée entre (x_n,y_n) et le point dont l'abscisse est x_n+dx , mais en utilisant cette fois la valeur de la pente de la tangente T_2 à la fonction au point milieu. Comme représenté sur la figure 4, la courbe T_3 est la parallèle à T_2 passant par (x_n,y_n) . On peut donc calculer la valeur de la différence d'ordonnée $k2$ grâce à la méthode d'Euler en utilisant la pente de T_3, qui est égale à celle de T_2. Celle-ci est alors donnée par l'équation $k_2=dx*f(x_n+\frac{1}{2}dx, y_n+\frac{1}{2}k_1)$.

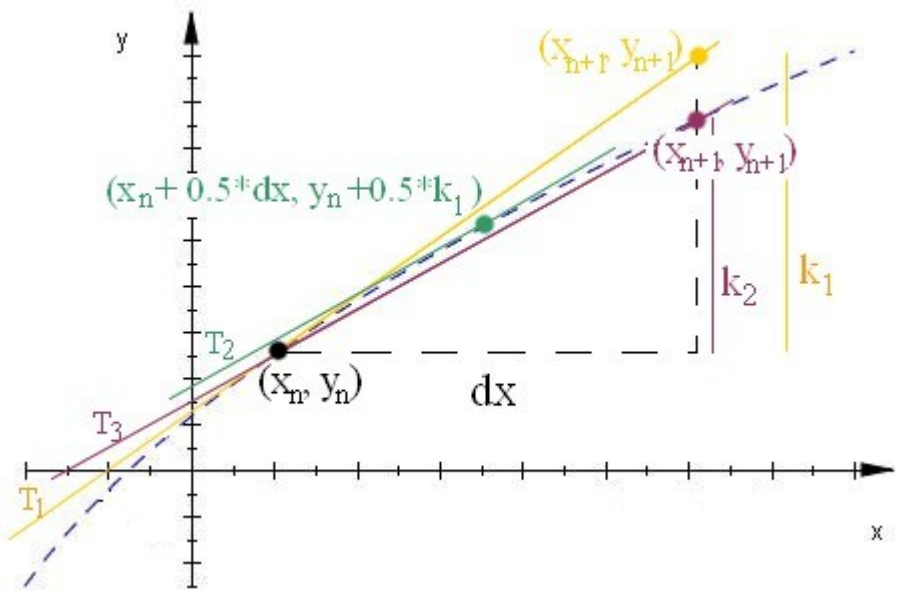


Figure 4: Représentation de la méthode du point milieu

Les valeurs de la fonction y sont alors calculées selon l'équation suivante.

$$y_{n+1}=y_n+k_2$$

L'erreur qui apparaît dans cette méthode est plus faible que celle apparaissant dans la méthode d'Euler. Il est possible d'utiliser d'autres méthodes de résolution plus complexes, mais plus précises, comme la méthode de Runge-Kutta.

3.4. La méthode de Runge-Kutta

La méthode de Runge-Kutta est une méthode de résolution d'équations différentielles souvent utilisée. Même si elle n'est pas la plus efficace au niveau de la précision et du temps de calcul, elle réussit toujours à trouver une solution satisfaisante[1].

Cette méthode est une extension de la méthode du point milieu. Elle est décrite ici comme la méthode du quatrième ordre, nécessitant l'évaluation de quatre points par itération.
En premier lieu, on commence par calculer les différences d'ordonnées $k1$ et $k2$ entre (x_n, y_n) et les points 1, d'abscisse x_n+dx , et 2, d'abscisse $x_n+\frac{1}{2}dx$, grâce à la méthode du point milieu décrite au chapitre précédent. Ces différences d'ordonnée sont donc données par les équations suivantes.

$$k_1=dx*f(x_n, y_n)$$

$$k_2=dx*f(x_n+\frac{1}{2}dx, y_n+\frac{1}{2}k_1)$$

Puis, comme représenté sur la figure 5, on calcule la différence d'ordonnée $k3$ entre (x_n, y_n) et le point 3, un deuxième point milieu d'abscisse $x_n+\frac{1}{2}dx$, en utilisant la valeur de la tangente à la fonction au point 2. Cette différence d'ordonnée est donnée par $k_3=dx*f(x_n+\frac{1}{2}dx, y_n+\frac{1}{2}k_2)$.

1 cf. réf. 1

On calcule ensuite la différence d'ordonnée $k4$ entre (x_n, y_n) et le point 4, un troisième point d'abscisse $x_n + dx$, en utilisant la valeur de la tangente à la fonction au point 3. Cette quatrième différence d'ordonnée est alors déterminée par $k_4 = dx * f(x_n + dx, y_n + k_3)$.

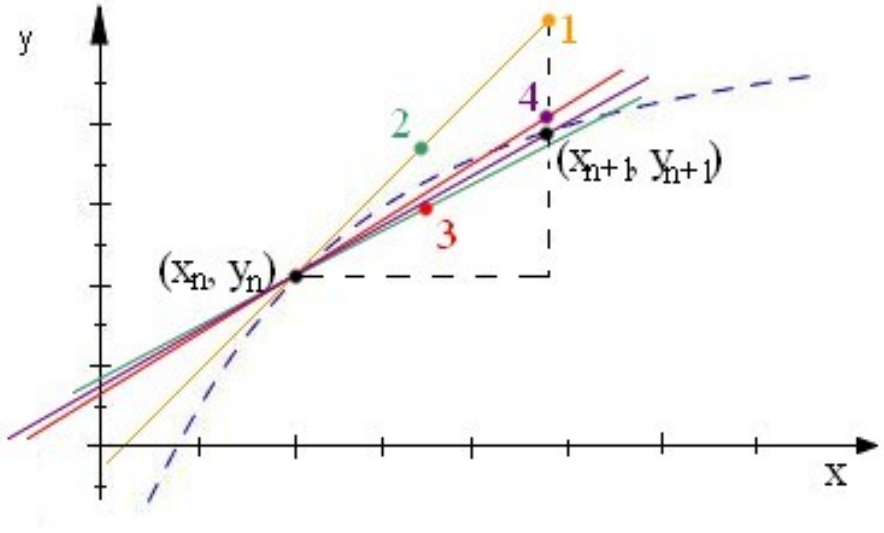


Figure 5: Représentation de la méthode de Runge-Kutta

Les valeurs de la fonction y sont alors données par l'équation suivante.

$$y_{n+1} = y_n + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6}$$

Il est possible d'augmenter le nombre de points utilisés de manière intermédiaire afin d'augmenter la précision des résultats obtenus par la méthode. Les programmes de calcul appliquent la méthode de Runge-Kutta qui peut être paramétrée en fonction de l'ordre utilisé.

4. Résolution numérique des équations du système

4.1. Résolution avec MuPAD

Les deux équations ci-dessous, qui ont été déterminées au chapitre 2.3, définissent la cinématique du système du pendule double et peuvent être résolues numériquement par la méthode de Runge-Kutta à l'aide du logiciel MuPAD.

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

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

Pour chacune des masses, il s'agit d'une équation différentielle du second ordre, qui peut être découplée en deux équations différentielles ordinaires du premier ordre. Cette opération est nécessaire étant donné que la méthode de Runge-Kutta ne résout que les équations différentielles du premier ordre.

Dans le but de limiter le nombre de cas à étudier, on utilisera dorénavant $m_1 = m_2 = 1$ kg et $l_1 = l_2 = 1$ m dans tous les calculs, les représentations et les interprétations de la suite du travail.

Dans le code MuPAD permettant de résoudre les équations, on commence par définir les valeurs des masses, des longueurs des tiges et de la constante g. Les angles θ_1 et θ_2, renommés a_1 et a_2, et les vitesses angulaires $\dot{\theta}_1$ et $\dot{\theta}_2$, renommées a_1' et a_2' sont des fonctions du temps. Ce sont leurs valeurs que l'on cherche à connaître. $\ddot{\theta}_1$ et $\ddot{\theta}_2$ sont respectivement nommées a_1'' et a_2''.
Afin de simplifier l'écriture, on substituera $m_1 + m_2$ par M et $a_1(t)-a_2(t)$ par A.

La fonction *numeric::ode2vectorfield* convertit implicitement le système d'équations différentielles du second ordre en un système d'équations différentielles du premier ordre comprenant le double d'éléments, qui est ensuite résolu par la fonction *numeric::odesolve2*. La fonction *numeric::ode2vectorfield* est composée de deux paramètres. Dans *IVP*, on définit les équations différentielles du second ordre et on donne les valeurs des conditions initiales. Puis, dans *Fields*, on donne la liste des fonctions inconnues dont on veut connaître la valeur pour chaque temps t. La fonction *numeric::odesolve2* résout ensuite par la méthode de Runge-Kutta d'ordre 8, définie par défaut dans MuPAD, les quatre équations différentielles du premier ordre qui ont été obtenues par la fonction précédente.
Le code MuPAD ci-dessous résout les équations du mouvement du pendule double. Ce code est original, de même que tous ceux qui figurent dans ce document.

```
m1:= 1 ;
m2:= 1 ;
l1:= 1 ;
l2:= 1 ;
g:= 9.81 ;
A:= a2(t)-a1(t);
M:= m1+m2;
IVP:={      a1''(t) = (m2 * l1 * (a1'(t))^2 * sin (A) * cos (A) + m2 * g *
            sin (a2(t)) * cos (A) + m2 * l2* (a2'(t))^2 * sin (A) - M * g *
            sin (a1(t))) / ( M*l1 - m2 * l1 * (cos (A))^2) ,
            a2''(t) = (-m2 * l2 * (a2'(t))^2 * sin (A) * cos (A) + M * (g *
            sin (a1(t)) * cos (A) - l1 * (a1'(t))^2 * sin (A) - g * sin
            (a2(t)))) / (M * l2 - m2 * l2 * (cos(A))^2) ,
            a1(0) = 0 , a1'(0) = 0 , a2(0) = PI/4 , a2'(0) = 1 };
Fields:= [ a1(t), a1'(t) , a2(t) , a2'(t)];
Equadiff:= numeric::ode2vectorfield(IVP, Fields);
Y:=numeric::odesolve2(Equadiff);
```

Pour connaître la valeur au temps t des fonctions recherchées, il suffit ensuite d'appeler la procédure Y. Par exemple, l'appellation ci-dessous retourne les valeurs de a_1, a_1', a_2, a_2' à t=1s, pour les conditions initiales $\theta_1 = 0$, $\theta_2 = \pi/4$, $\dot{\theta}_1 = 0$, $\dot{\theta}_2 = 1$.

```
Y(1);
[-0.1203516589, -2.314254407, -0.1224611561, 1.334458465]
```

4.2. Résolution à l'aide d'un programme C

Les deux équations différentielles du second ordre définissant la cinématique du système du pendule double peuvent également être résolues par la méthode de Runge-Kutta à l'aide du langage de programmation C[2]. Le fait de résoudre le système d'équations à l'aide d'un programme C présente

2 cf. Annexe 1

des avantages par rapport à MuPAD. Tout d'abord, le programme peut ouvrir un fichier contenant les valeurs des résultats obtenus, ce qui est pratique pour travailler avec ces valeurs par la suite. De plus, la vitesse de calcul est nettement supérieure à celle de MuPAD. Le gain de temps est considérable et permet de calculer un grand nombre de valeurs rapidement.

L'utilisation de deux méthodes de programmation différentes pour la même résolution d'équations permet de vérifier la justesse des résultats obtenus. En effet, toutes deux aboutissent aux mêmes valeurs lorsqu'on choisit les mêmes conditions de départ.

5. Représentation cinématique du pendule double

5.1. Représentation statique

Une fois les équations du mouvement du pendule double résolues, il est possible d'en faire une représentation graphique pour tout temps t à l'aide de MuPAD.

Après avoir résolu les équations, on écrit le code suivant dans MuPAD. Tout d'abord, on choisit un temps t pour lequel on demande les valeurs des angles et des vitesses angulaires. Grâce à ces valeurs, on exprime la position de chacune des masse m_1 et m_2 au temps t selon les axes x et y grâce aux fonctions sinus et cosinus et aux longueurs l_1 et l_2. La fonction *plot::Point2d* permet ensuite de représenter ces positions et la fonction *plot::Line2d* de relier les différents points entre eux.

```
t:= 0:
x1:= -sin(Y(t)[1])* l1:
y1:= -cos(Y(t)[1])* l1:
x2:= -sin(Y(t)[1])* l1 - sin (Y(t)[3]) * l2:
y2:= -cos(Y(t)[1])* l1 - cos (Y(t)[3]) * l2:
Point0:= plot::Point2d(0,0, Color=RGB::Black):
Point1:= plot::Point2d(x1(t),y1(t), Color=RGB::Black):
Point2:= plot::Point2d(x2(t), y2(t), Color=RGB::Black):
Line1:=plot::Line2d([0,0],[x1(t),y1(t)], Color=RGB::RoyalBlue):
Line2:=plot::Line2d([x1(t),y1(t)],[x2(t), y2(t)], Color=RGB::Purple):
Compplot:=plot(Point0, Point1, Point2, Line1, Line2, Scaling=Constrained);
```

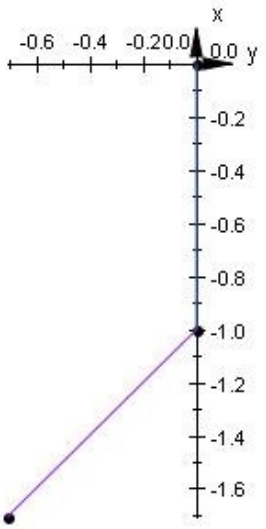


Figure 6: Le pendule double au temps 0

La figure 6 représente le pendule double au temps t=0 pour les conditions initiales θ_1 =0, θ_2 = π/4, $\dot{\theta}_1$ =0, $\dot{\theta}_2$ =1.

5.2. Animation

Le pendule double peut ensuite être animé et évoluer dans le temps à l'aide de MuPAD.

Pour cela, il faut définir les positions des masses selon les axes x et y comme des fonctions du temps et non plus comme valeurs pour un temps particulier dans le code du chapitre précédent.

```
x1:= t -> -sin(Y(t)[1])* l1:
y1:= t -> -cos(Y(t)[1])* l1:
x2:= t -> -sin(Y(t)[1])* l1 - sin (Y(t)[3]) * l2:
y2:= t -> -cos(Y(t)[1])* l1 - cos (Y(t)[3]) * l2:
Point0:= plot::Point2d(0,0):
Point1:= plot::Point2d(x1(t),y1(t), t = 0..2*PI):
Point2:= plot::Point2d(x2(t), y2(t), t = 0..2*PI):
Animation1:=plot(Point1, Point2, Point0, Line1, Line2, Scaling =
Constrained);
```

La figure 7 montre, à l'aide d'une série de figures statiques, l'évolution du pendule double dans le temps de t=0s à t=9s.

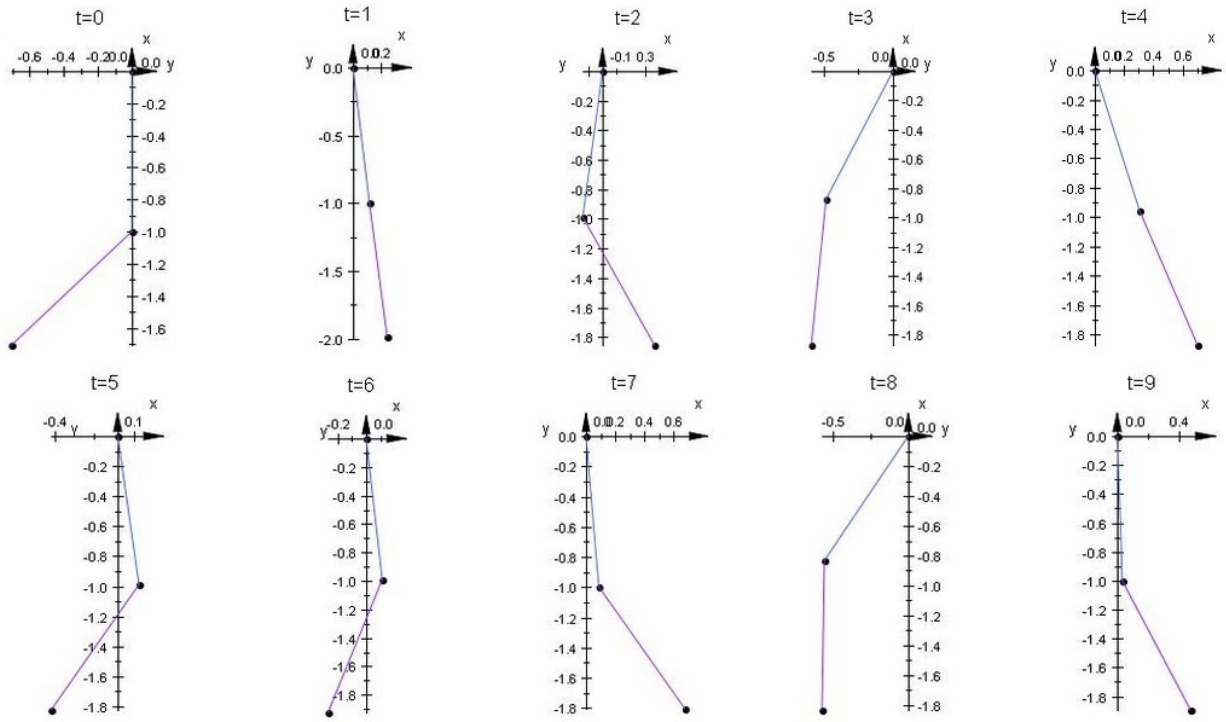


Figure 7: Cinématique du pendule double

6. Lois de conservation dans le système

L'énergie mécanique du système est conservée par le fait que le cas traité est idéal. Les seules transformations que l'on observe sont les transformations d'énergie potentielle en énergie cinétique, et d'énergie cinétique en énergie potentielle.

Ces énergies se calculent pour chaque masse m_1 et m_2 en fonction du temps à l'aide des angles θ_1 et θ_2 et des vitesses angulaires $\dot{\theta}_1$ et $\dot{\theta}_2$ qui sont connus grâce à la résolution numérique des équations différentielles décrivant le système.

L'énergie potentielle d'un corps est égale à m * g * Δh. Pour m_1 et m_2, ces énergies sont données par les équations suivantes.

$$E_{pot1} = -m_1 * g * \cos(\theta_1(t)) * l_1$$
$$E_{pot2} = -m_2 * g * (\cos(\theta_1(t)) * l_1 + \cos(\theta_2(t) * l_2))$$

Les signes – de ces deux équations sont dus au fait que l'énergie potentielle est minimale lorsque la masse concernée se trouve au point le plus bas.

L'énergie cinétique d'un corps est égale à ½ * m * v^2.
Lorsque nous avons défini les équations du mouvement du système, nous avons obtenu les vitesses vectorielles, respectivement pour les masses m_1 et m_2 égales à $\vec{v}_1 = l_1 \dot{\theta}_1 \begin{pmatrix} -\sin(\theta_1) \\ \cos(\theta_1) \end{pmatrix}$ et à $\vec{v}_2 = l_1 \dot{\theta}_1 \begin{pmatrix} -\sin(\theta_1) \\ \cos(\theta_1) \end{pmatrix} + l_2 \dot{\theta}_2 \begin{pmatrix} -\sin(\theta_2) \\ \cos(\theta_2) \end{pmatrix}$. Les vitesses quadratiques scalaires, égales aux normes au carré des vecteurs vitesse, peuvent être déterminées à partir des composantes de ces vecteurs.

Pour m_1 et m_2, les énergies cinétiques sont alors données par les équations suivantes.

$$E_{cin1} = 0.5 * m_1 * (\dot{\theta}_1(t))^2 * l_1^2$$
$$E_{cin2} = 0.5 * m_2 * ((\dot{\theta}_1(t))^2 * l_1^2 + (\dot{\theta}_2(t))^2 * l_2^2 + 2 * l_1 * l_2 * \dot{\theta}_1(t) * \dot{\theta}_2(t) * \cos(\theta_1 - \theta_2))$$

L'énergie mécanique totale est la somme des énergies potentielles et cinétiques de chaque masse. Elle doit être la même en tout temps t.

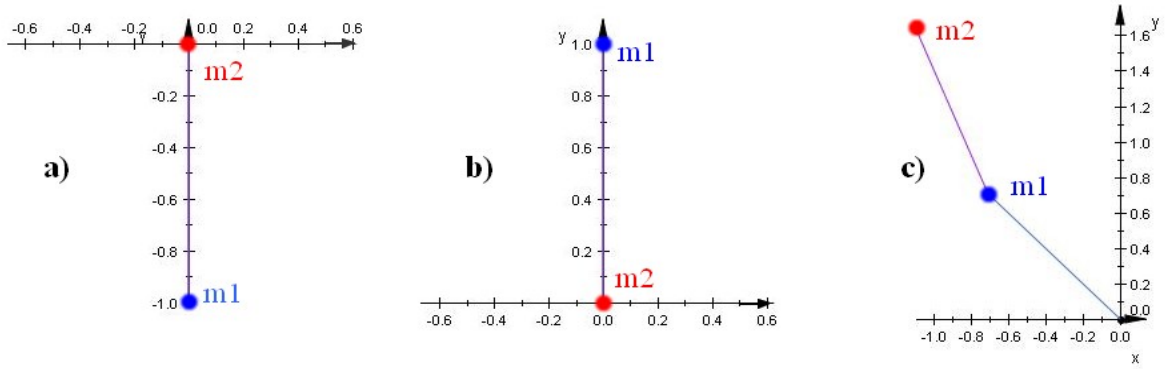


Figure 8: Représentation des différents états cinématiques du système

Le système est caractérisé par des seuils d'énergie. L'énergie minimum est atteinte lorsque les vitesses et les angles des deux masses sont nuls. Le système est alors statique. Pour des énergies légèrement supérieures à ce minimum, les masses oscillent de manière analogue au système représenté sur la figure 7. A partir d'un niveau d'énergie, pour lequel l'angle θ_2 peut atteindre 180° alors que les vitesses et l'angle θ_1 sont nuls, on peut observer une transition de par la possibilité de rotation complète de la seconde masse autour de la première (figure 8 a). En augmentant encore l'énergie, une deuxième transition, correspondant à la rotation possible de la première masse autour de son axe, intervient (figure 8 b). Enfin, pour des hauts niveaux d'énergie, les deux masses tendent à effectuer des rotations continues autour de leurs axes (figure 8 c).

7. Utilisation des lois de conservation dans le système pour valider les calculs

Le calcul de la somme des différentes énergies des masses à différents temps t permet une vérification des équations du système et de leur résolution.
On peut représenter ces énergies à l'aide de MuPAD. Dans le code suivant, introduit après celui permettant la résolution des équations différentielles du système, on définit les équations décrivant les différentes énergies en fonction du temps. La fonction plot::Function2d permet ensuite de représenter leur évolution sur le graphique de la figure 9.

```
EC1:= 0.5 * m1 * l1^2 * (Y(t)[2])^2:
EC2:= 0.5 * m2 * ( l1^2 * (Y(t)[2])^2 + l2^2 * (Y(t)[4])^2 + 2*l1*l2*(Y(t)
[2])*(Y(t)[4])*cos((Y(t)[1])-(Y(t)[3])) ):
EP1:= - (m1 * g * cos (Y(t)[1]) * l1):
EP2:= - (m2 * g * (cos(Y(t)[1])*l1 + cos(Y(t)[3])*l2)):
CE:= EC1 + EC2 + EP1 + EP2:

EC1R:=plot::Function2d(EC1, Color=RGB::Orange, t=0..20, Legend = "Energie
cinétique 1", LegendPlacement=Top):
EC2R:=plot::Function2d(EC2, Color=RGB::Purple, t=0..20, Legend = "Energie
cinétique 2"):
EP1R:=plot::Function2d(EP1, Color=RGB::Green, t=0..20, Legend = "Energie
potentielle 1"):
EP2R:=plot::Function2d(EP2, Color=RGB::Black, t=0..20, Legend = "Energie
potentielle 2"):
CER:=plot::Function2d(CE, t=0..20, Legend = "Energie mécanique totale"):

plot(EC1R,EC2R,EP1R,EP2R,CER, AxesTitles=["t[s]","Energie[J]"]);
```

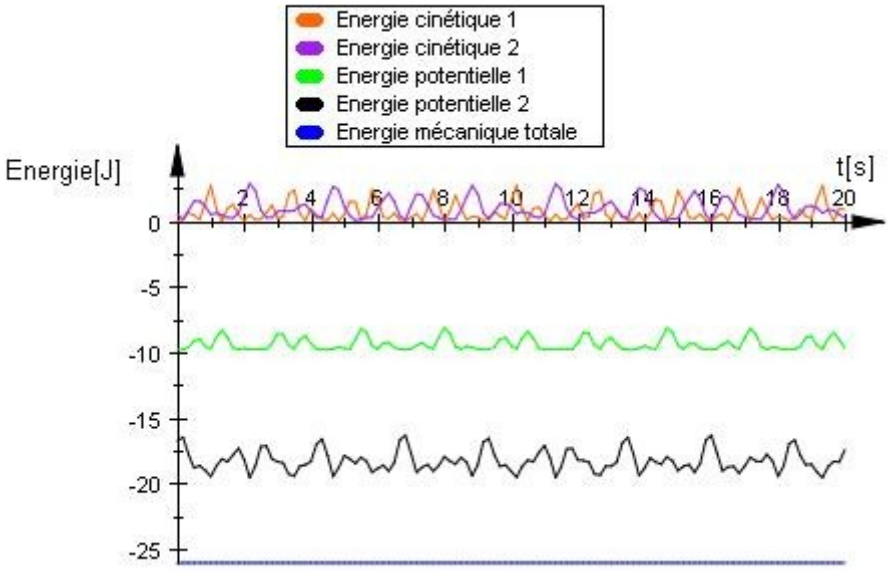


Figure 9: Évolution des énergies mécaniques dans le système

On remarque alors que l'énergie mécanique totale est belle et bien conservée, étant donné qu'elle correspond à une fonction constante sur le graphique ci-dessus.

8. Évaluation de la précision des calculs

L'évaluation de la précision de nos calculs est possible en utilisant la conservation de l'énergie dans le système. En effet, comme l'énergie mécanique totale devrait être constante, la différence entre l'énergie totale de départ et l'énergie totale à un temps t est égale à l'imprécision due aux calculs.

Dans le code MuPAD suivant, on définit l'énergie mécanique totale comme fonction du temps, puis, on calcule la différence entre l'énergie mécanique totale à t=0s et l'énergie mécanique totale pour t allant de 0 à 20s.

```
CEF:= t -> 0.5 * m1 * l1^2 * (Y(t)[2])^2 + 0.5 * m2 * ( l1^2 * (Y(t)[2])^2 +
l2^2 * (Y(t)[4])^2 + 2*l1*l2*(Y(t)[2])*(Y(t)[4])*cos((Y(t)[1])-(Y(t)[3])) )
- (m1 * g * cos (Y(t)[1]) * l1) - (m2 * g * (cos(Y(t)[1])*l1 + cos(Y(t)
[3])*l2)):
CEF(0);
-26.05671752

Imprecision:=plot(plot::Function2d(CEF(0)-CEF(t), t=0..20),
AxesTitles=["Temps [s]","E(0)-E(t) [J]"]);
```

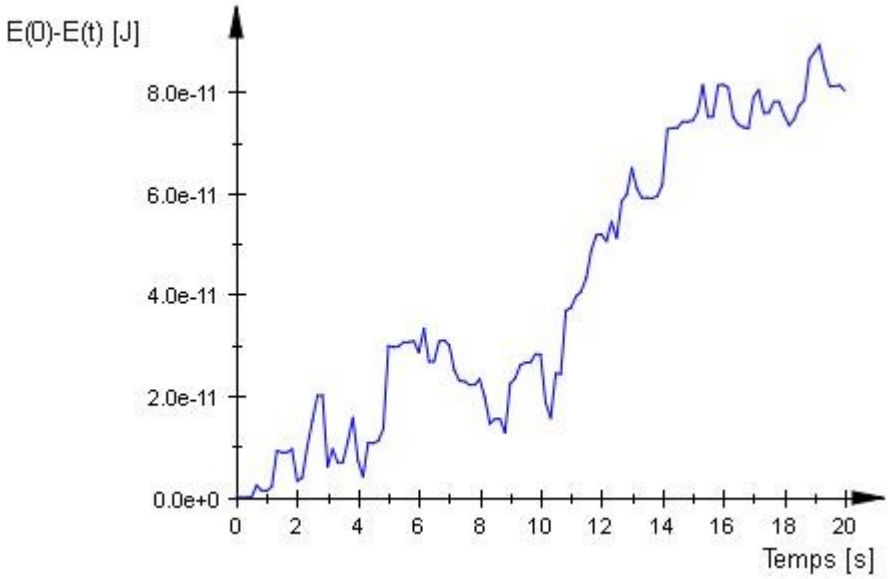


Figure 10: Visualisation de l'imprécision sur l'énergie mécanique totale du système

La figure 10 permet d'observer une imprécision absolue de 10^{-11} et une imprécision relative de l'ordre de 10^{-12}, puisque l'énergie initiale est de $-2{,}605671752*10^{1}$. Les résultats sont cohérents, étant donné que la précision relative donnée par défaut dans MuPAD est de l'ordre de 10^{-10}, ce qui garanti l'exactitude des dix premiers termes du résultat obtenu.
Il est normal que l'imprécision augmente avec le temps car nous utilisons une méthode de résolution numérique. A chaque nouvelle itération, on utilise les résultats obtenus à l'itération précédente, pour lesquels l'exactitude des dix premiers termes est garantie. On ajoute ainsi une imprécision supplémentaire à l'imprécision déjà présente à partir du onzième terme du résultat précédent.

Pour cette raison, les systèmes chaotiques sont prévisibles avec grande précision à court terme seulement. Leur extrême sensibilité aux conditions initiales rend l'imprécision des calculs très critique pour la prévision de leur comportement à plus long terme.

9. Divergence temporelle du pendule double

9.1. Choix de la représentation

Pour montrer la sensibilité aux conditions initiales du système du pendule double, il est nécessaire de disposer d'une représentation graphique appropriée. Cette représentation doit permettre de visualiser la divergence des paramètres finaux en fonction des paramètres initiaux.
Le système est défini en tout temps par quatre paramètres qui sont les angles θ_1 et θ_2 et les vitesses angulaires $\dot{\theta}_1$ et $\dot{\theta}_2$. Au temps t=0s, ces paramètres constituent les conditions initiales. Il est possible d'étudier l'évolution de ces paramètres en fonction du temps et en fonction des conditions initiales. Étant donné qu'il est impossible de représenter simultanément l'ensemble des quatre paramètres en montrant leur dépendance aux quatre conditions initiales, il est nécessaire de se restreindre à une représentation de chaque paramètre séparément en fonction d'un sous-ensemble de paramètres initiaux. Une représentation bi-dimensionnelle permet de représenter un paramètre en fonction de deux conditions initiales variant dans deux intervalles donnés. Par exemple, on fera varier deux des paramètres initiaux, θ_1 de -90° à 90° sur l'axe des abscisses et θ_2 de -90° à 90° sur l'axe des ordonnées. Les deux autres paramètres initiaux sont fixés et un temps t, pour lequel la variable est calculée, est choisi. On représente alors, sur l'ensemble de la surface, un des quatre paramètres au temps t en indiquant ses valeurs par une gradation de couleur.
La figure 11 représente la variation de θ_1 au temps t=1s en fonction des paramètres initiaux θ_1 et θ_2 , $\dot{\theta}_1$ et $\dot{\theta}_2$ étant fixés comme nuls au temps t=0s.

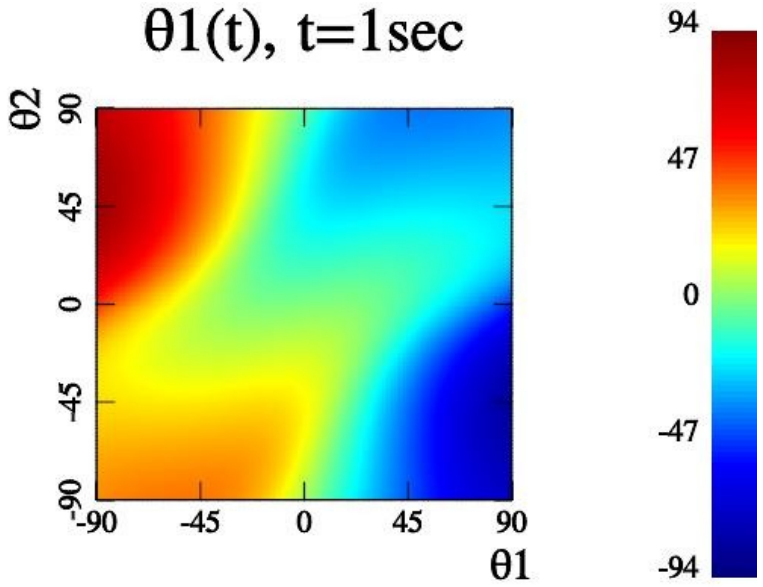


Figure 11: Valeur de la variable θ_1 au temps t=1s en fonction des valeurs initiales de θ_1 et θ_2

9.2. Représentation sous forme de diagrammes

Il est nécessaire de calculer la valeur du paramètre que l'on veut représenter sur l'ensemble de la surface pour un certain nombre de points, afin que l'on puisse bien voir son évolution en fonction du sous-ensemble de conditions initiales que l'on fait varier. La fonction espacedephase[3], qu'il faut ajouter à l'ensemble du programme C utilisé pour résoudre les équations du système, calcule les valeurs des quatre paramètres θ_1 , θ_2 , $\dot{\theta}_1$ et $\dot{\theta}_2$ à un certain temps en faisant varier deux des conditions initiales. Il ouvre ensuite quatre fichiers différents où sont répertoriées les valeurs obtenues pour chacun des paramètres sous la forme d'un tableau à double entrée. La première ligne

3 cf. Annexe 2

du tableau donne les valeurs de la première condition initiale et la première colonne donne celles de la deuxième condition initiale.

Il est ensuite possible de représenter graphiquement les données que l'on a obtenues grâce au logiciel Udav, un programme qui permet la représentation graphique à partir des données d'un fichier créé par la fonction *espacedephase* du programme C[4].

On peut ainsi, grâce aux graphiques, observer la sensibilité aux conditions initiales du système du pendule double.

9.3. Représentation des résultats sous forme de diagrammes

Afin de pouvoir observer en détail la sensibilité aux conditions initiales du système, il est nécessaire de représenter l'évolution de chaque paramètre en fonction d'un sous-ensemble de conditions initiales qui varient pour plusieurs temps t. Il est également intéressant de zoomer à l'intérieur des graphiques obtenus afin d'observer si cette sensibilité est également valable sur des variations minimes des conditions de départ.

Les diagrammes de la figure 12 représentent l'évolution de θ_1 en fonction des angles initiaux pour différents temps et différents zooms (resserrement des valeurs initiales).

L'échelle des valeurs de θ_1 est donnée par la barre de couleurs au-dessous de la figure et va de -94° (bleu foncé) à 94° (rouge foncé).

Ces différents diagrammes montrent que la sensibilité aux conditions initiales du système s'accroît au cours du temps: plus le système évolue, plus certaines zones deviennent désordonnées. Ces zones sont caractérisées par une forte variation de θ_1 pour de très faibles variations des conditions initiales. On peut observer que les zones désordonnées apparaissent progressivement pour les angles initiaux les plus grands. Puis, ces zones deviennent de plus en plus grandes et s'avancent vers le centre du diagramme, où les angles initiaux sont minimaux.
Pour une variation très petite des conditions initiales, on peut observer une très grande variation de l'angle θ_1 . Ceci est facilement observable dans les zooms effectués sur les graphiques représentant θ_1 après 10 ou 20s.

Les diagrammes de la figure 13 représentent l'évolution de $\dot{\theta}_1$ en fonction des angles initiaux pour différents temps et différents zooms (resserrement des valeurs initiales).

4 cf. Annexe 3

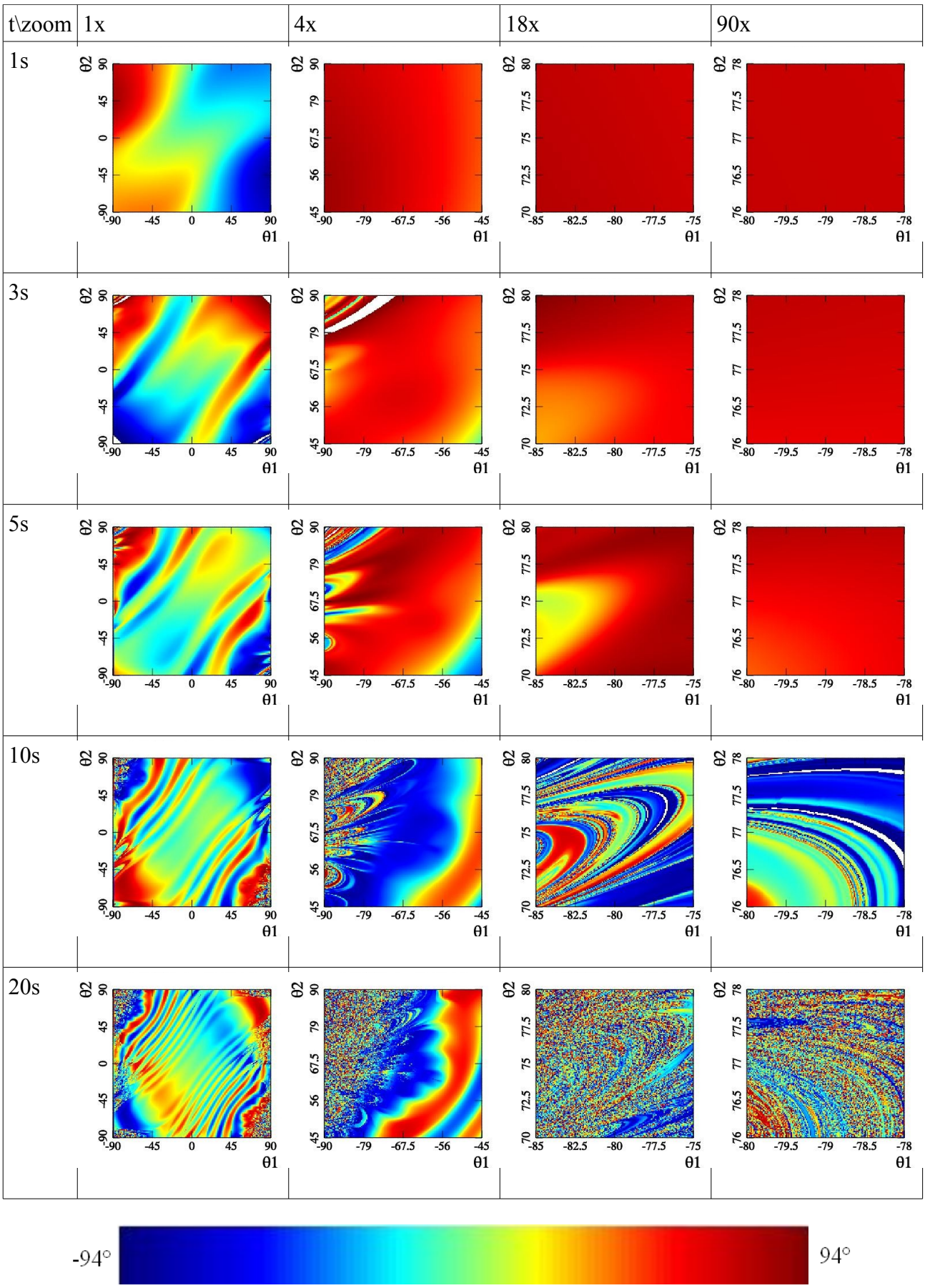


Figure 12: Représentation de θ_1 pour différents temps en fonction des paramètres initiaux

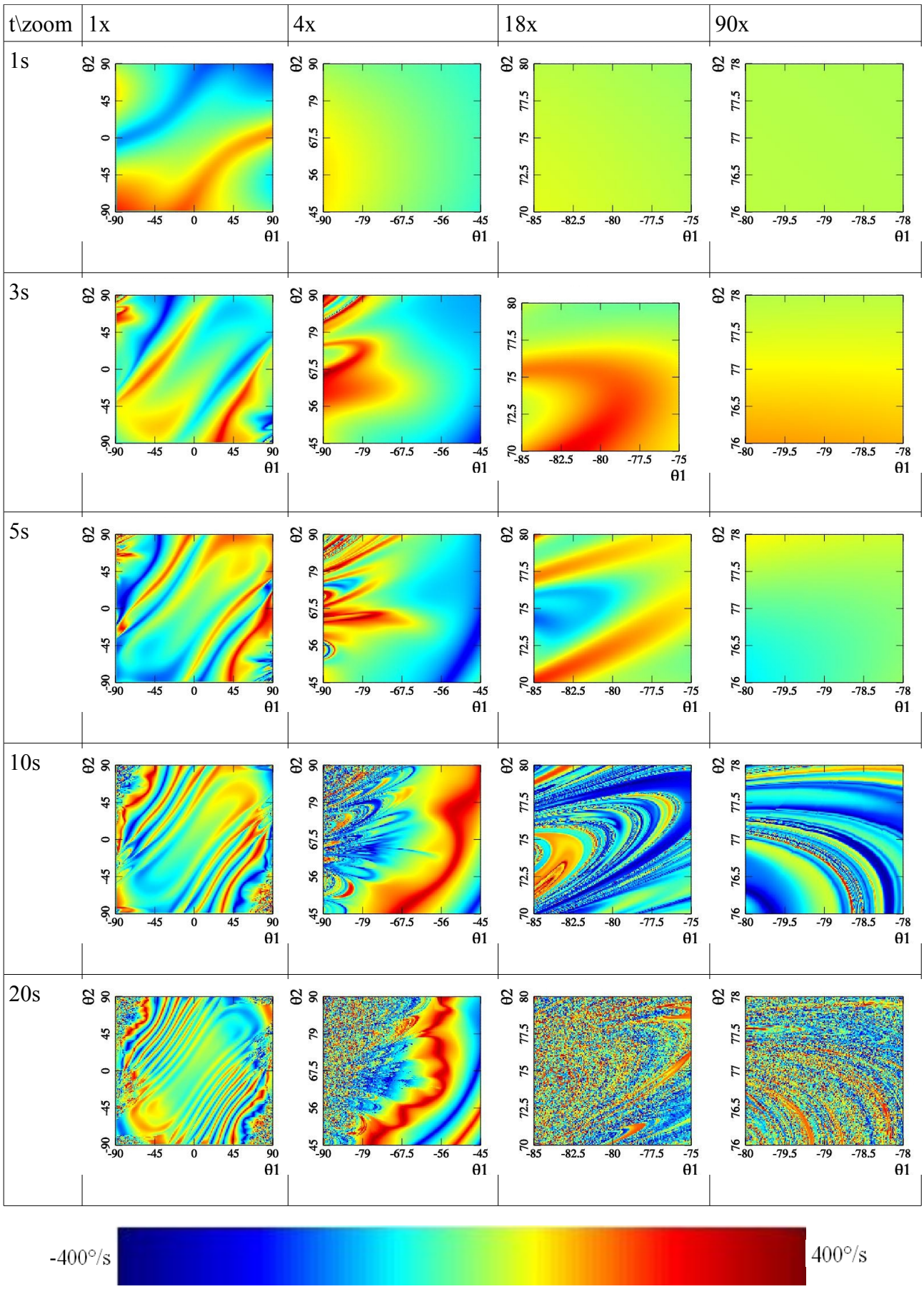


Figure 13: Représentation de $\dot{\theta}_1$ pour différents temps en fonction des angles initiaux

L'échelle des valeurs de $\dot{\theta}_1$ est donnée par la barre de couleurs ci-dessus et va de -400°/s (bleu foncé) à 400°/s (rouge foncé).

On peut observer grâce à ces diagrammes que la sensibilité de la vitesse angulaire $\dot{\theta}_1$ aux conditions initiales est très proche de celle observée pour l'angle θ_1 .

Les diagrammes de la figure 14 représentent l'évolution de $\dot{\theta}_1$ en fonction des angles initiaux après 20s pour différentes vitesses angulaires $\dot{\theta}_1$ et $\dot{\theta}_2$ initiales non-nulles.

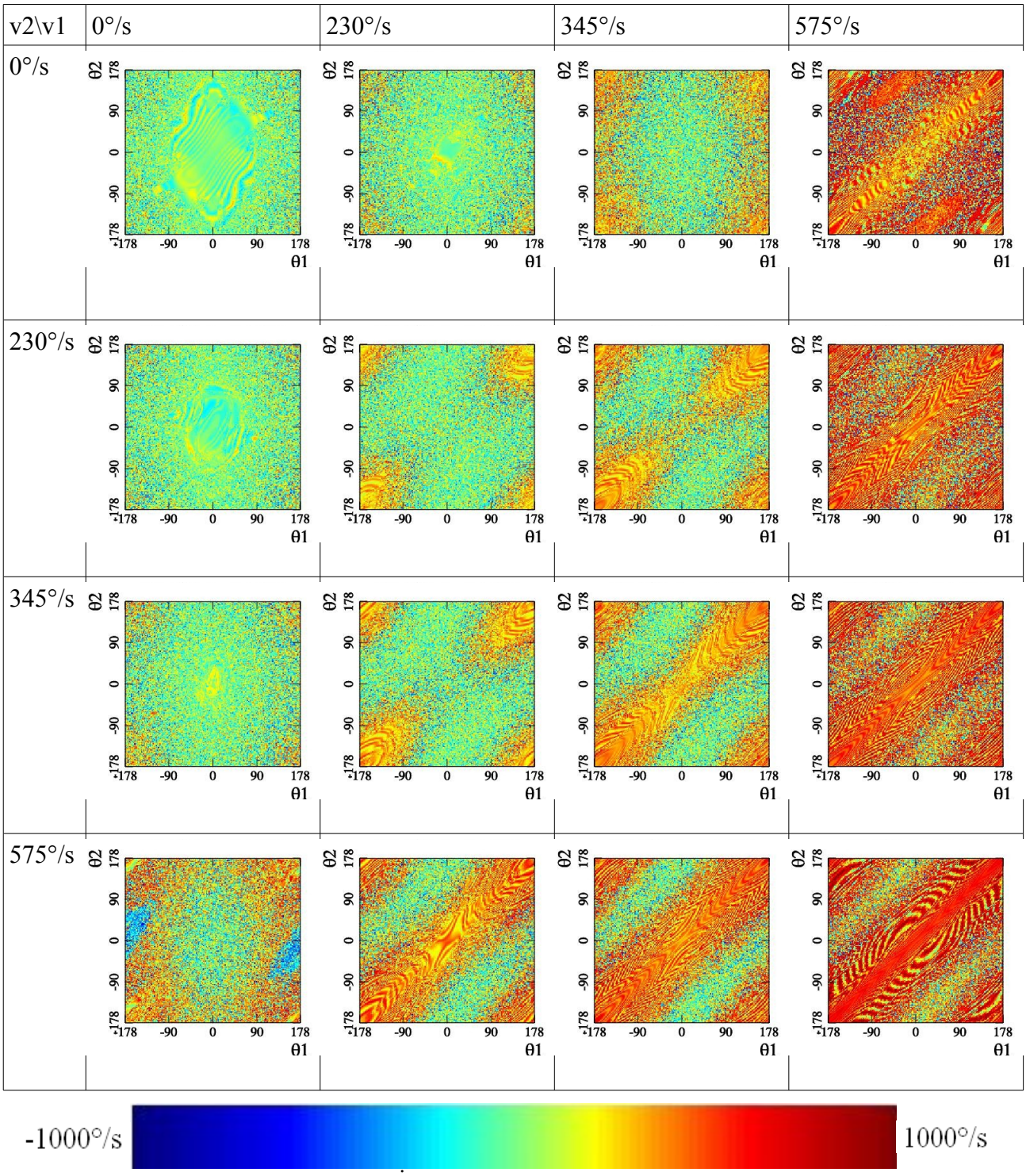


Figure 14: Représentation de $\dot{\theta}_1$ pour différents temps en fonction des vitesses initiales

L'échelle des valeurs de $\dot{\theta}_1$ est donnée par la barre de couleurs ci-dessus et va de -1000°/s (bleu foncé) à 1000°/s (rouge foncé).

Les différentes vitesses initiales sont choisies de façon à ce que l'énergie cinétique initiale de chacune des deux masses du système corresponde à la moitié, à l'entier et au double de l'énergie potentielle initiale maximum de la première masse.
Grâce à ces différents diagrammes, il est possible d'observer que la sensibilité aux conditions initiales du système semble moins marquée, soit lorsque l'énergie totale du système est faible, ce que l'on observe dans la partie centrale du diagramme pour lequel les deux vitesses angulaires initiales sont nulles, soit lorsque l'énergie cinétique initiale, et donc l'énergie totale, sont élevées, dans quel cas on peut observer la construction de structures plus régulières à l'intérieur des diagrammes.

10. Évaluation de l'exposant de Lyapunov

10.1. Définition de l'exposant de Lyapunov

L'exposant de Lyapunov permet de quantifier la sensibilité aux conditions initiales d'un système physique. En effet, il permet de calculer le taux local d'instabilité du système, c'est-à-dire que l'on peut montrer l'effet d'une petite différence sur des conditions initiales que l'on aura choisies auparavant. L'exposant de Lyapunov du système est alors défini comme le plus grand taux local d'instabilité. Ce taux doit être positif pour que le système soit chaotique.

Soit $\delta(0)$ la distance initiale entre les paramètres de deux situations proches et $\delta(t)$ la distance entre ces mêmes paramètres au temps t. On définit alors l'exposant de Lyapunov λ de la manière suivante:

$$\delta(t)=\delta(0)*e^{(\lambda*t)}$$

10.2. Évaluation d'exposants de Lyapunov locaux

Il est important de bien considérer les paramètres que l'on prend en compte dans le calcul des distances nécessaires à l'évaluation de l'exposant de Lyapunov.

Il est ainsi possible de calculer le taux d'instabilité du système en prenant en compte les quatre paramètres le définissant pour tout temps t, c'est-à-dire les angles θ_1 et θ_2 et les vitesses angulaires $\dot{\theta}_1$ et $\dot{\theta}_2$. La distance au temps t entre deux points a et b de la trajectoire du pendule pour des conditions initiales légèrement différentes est donnée par la norme du vecteur à quatre composantes, définies par les angles et les vitesses angulaires, reliant ces deux points.

$$\delta(t)=\left\|\begin{pmatrix}\theta_{1a}(t)\\ \dot{\theta}_{1a}(t)\\ \theta_{2a}(t)\\ \dot{\theta}_{2a}(t)\end{pmatrix}-\begin{pmatrix}\theta_{1b}(t)\\ \dot{\theta}_{1b}(t)\\ \theta_{2b}(t)\\ \dot{\theta}_{2b}(t)\end{pmatrix}\right\|=\sqrt{((\theta_{1a}-\theta_{1b})^2+(\dot{\theta}_{1a}-\dot{\theta}_{1b})^2+(\theta_{2a}-\theta_{2b})^2+(\dot{\theta}_{2a}-\dot{\theta}_{2b})^2)}$$

La figure 15, obtenue grâce à MuPAD[5], représente graphiquement l'exposant de Lyapunov pour t allant de 0 à 20s pour deux trajectoires du système proches. Les conditions initiales de la première trajectoire sont θ_1 =π/2, θ_2 = π/2, $\dot{\theta}_1$ =0, $\dot{\theta}_2$ =0, et celles de la deuxième sont augmentée de 10^{-8}, par rapport aux précédentes, pour chaque paramètre.

5 cf. Annexe 4

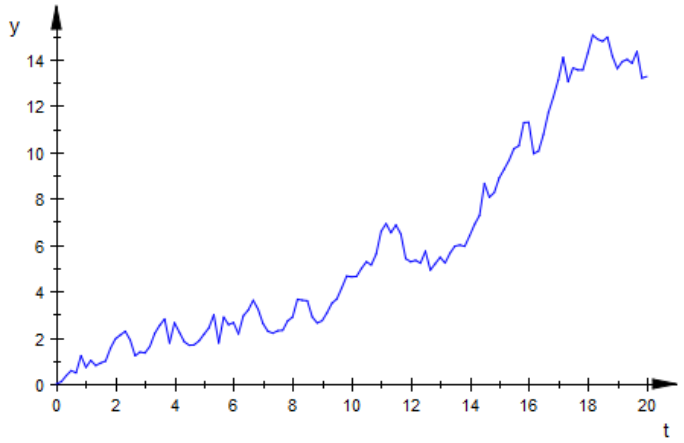


Figure 15: Représentation de l'exposant de Lyapunov par le calcul des distances entre les angles et les vitesses angulaires

Dans la figure 15, l'exposant de Lyapunov au temps t est égal à la pente de la courbe du graphique au point (t,y). On remarque que sa valeur varie au cours du temps: la pente, plus ou moins forte, est parfois positive et parfois négative. Cela est dû au fait que la distance entre les deux trajectoires est limitée par la longueur des deux tiges composant le pendule double. Par conséquent, la distance ne peut pas augmenter indéfiniment.

En calculant l'exposant de Lyapunov à l'aide de la distance définie par les quatre paramètres caractérisant le système, on manque toutefois de rigueur au niveau des unités: en effet, on additionne des angles et des vitesses angulaires. Or, l'unité des angles est le radian, alors que celle des vitesses angulaires est le radian par seconde. Pour remédier à cela, il est possible de calculer l'exposant de Lyapunov à l'aide d'une définition différente.
Soit $\delta(0)$ la distance initiale entre les mêmes paramètres de deux situations proches, $\delta(t)$ la distance entre ces paramètres au temps t, i les différents paramètres initiaux, N le nombre total de paramètres initiaux. L'exposant de Lyapunov λ est défini de la manière suivante[6]:

$$\lambda = \frac{1}{t*N} \Sigma_{i=1}^{N} \ln \frac{|(\delta_i(t))|}{|(\delta_i(0))|}$$

10.3. Évaluation de l'exposant de Lyapunov maximal pour l'ensemble du système

Les valeurs de l'exposant de Lyapunov obtenues au chapitre précédent sont valables uniquement pour le set de valeurs initiales à l'aide desquelles elles sont calculées. Pour mieux se rendre compte de la quantité de chaos dans le système, il serait préférable de calculer l'exposant de Lyapunov maximal pour l'ensemble du système à chaque temps. Cela correspond à trouver la variation la plus forte sur les diagrammes des figures 11,12 et 13 dans les colonnes de gauche, correspondant à la représentation des échelles d'angles et de vitesses complètes.

Ceci a un coût de calcul trop important. On peut alors appliquer une méthode de tirage aléatoire de sets de conditions initiales. Pour chaque paramètre, c'est-à-dire θ_1 , θ_2 , $\dot{\theta}_1$ et $\dot{\theta}_2$, on calcule la dérivée en tout temps t. En effet, la dérivée correspond au taux de variation instantané, correspondant ici à la distance entre deux paramètres de deux trajectoires très proches. En la calculant, on obtient ainsi les différents $\delta_i(t)$. On calcule également l'énergie mécanique globale, qui reste constante au cours du temps.

6 cf. réf. 7

En répétant l'opération avec une grande quantité de sets de conditions initiales - dans l'idéal il en faudrait une infinité - il est possible d'approcher l'exposant maximum de chaque temps t et ainsi de représenter l'exposant de Lyapunov maximal sur l'ensemble du système.

Par le biais d'un programme en C[7], 10'000 sets de conditions initiales ont été tiré au hasard et l'exposant de Lyapunov a été calculé pour chacun d'entre eux, pour un temps allant de 0 à 12 secondes. L'exposant de Lyapunov global[8] du système est représenté sur la figure 16. Ce graphique montre le caractère chaotique du pendule double, de par la valeur de l'exposant égale à la pente, constamment positive.

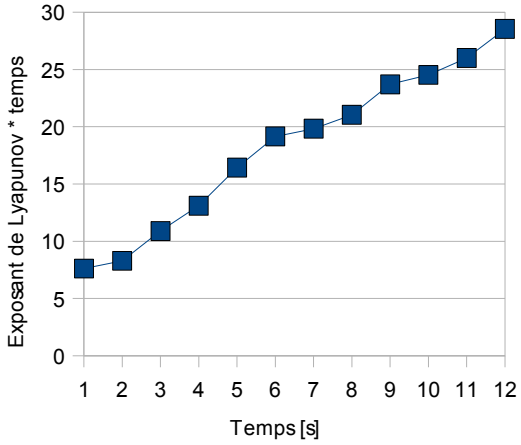


Figure 16: Représentation de l'exposant de Lyapunov (pente) en fonction du temps

On s'intéresse ensuite à l'exposant de Lyapunov en fonction de l'énergie globale du système. En effet, comme discuté au chapitre 6, la cinématique possible est fortement dépendante de l'énergie mécanique totale du pendule double.

La figure 17 représente ainsi la valeur de l'exposant de Lyapunov de chaque set de conditions initiales en fonction de l'énergie mécanique totale du système au temps t=20s. On remarque alors nettement une dépendance de la valeur de l'exposant à l'énergie.

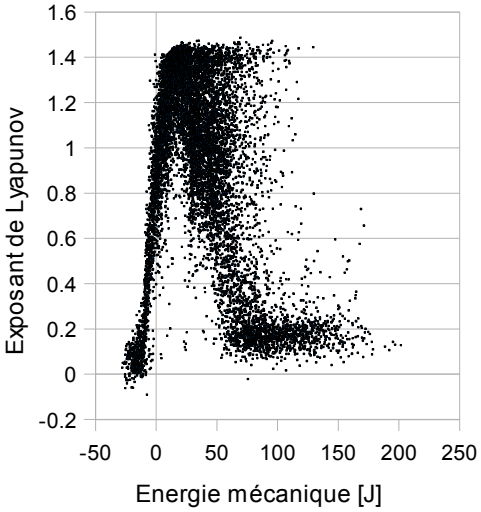


Figure 17 : Évaluation de l'exposant de Lyapunov en fonction de l'énergie mécanique totale après 20 s

7 cf. Annexe 5
8 cf. Annexe 6

Les valeurs obtenues précédemment ont ensuite été classées par niveaux d'énergie et regroupées en neuf classes comportant 1000 sets de paramètres initiaux et deux classes comportant 500 sets de paramètres initiaux. La formation de deux classes contenant moins de valeurs est due à la grande variation d'énergie entre les 1000 énergies totales les plus élevées.

La figure 18 représente l'évolution de l'exposant de Lyapunov maximal[9] de chaque classe en fonction du temps. Sur ce graphique, la pente des courbes représente l'exposant. On peut tout d'abord remarquer que ces courbes tendent à être des droites, ce qui signifierait que l'exposant tend vers une valeur fixe. De plus, les pentes les plus fortes sont produites par les classes d'énergie totale moyenne. Les classes d'énergie faibles et élevées donnent alors naissance à des pentes plus faibles.

La figure 19 représente l'évolution de l'exposant de Lyapunov maximal[10] de chaque classe en fonction du temps. La valeur de l'exposant se lit cette fois sur l'axe des ordonnées. Ce graphique tend à montrer que le chaos dans le système dépend de l'énergie mécanique totale du système. En effet, l'exposant tend vers une certaine valeur pour chaque classe d'énergie. Les deux plus petits exposants sont ceux des classes où les énergies sont les plus faibles. Ils tendent respectivement vers 1.16 et 1.57. En allant dans l'ordre croissant, les exposants suivants sont celui de la classe des énergies les plus élevées, d'une valeur de 1.63, puis celui de la classe précédente, tendant vers 1.8. Les exposants des classes d'énergie intermédiaires atteignent des valeurs tendant entre 1.9 et 2.1. L'exposant de Lyapunov maximal est ainsi donné par les énergies allant de environ 17 à 24.7 Joules.

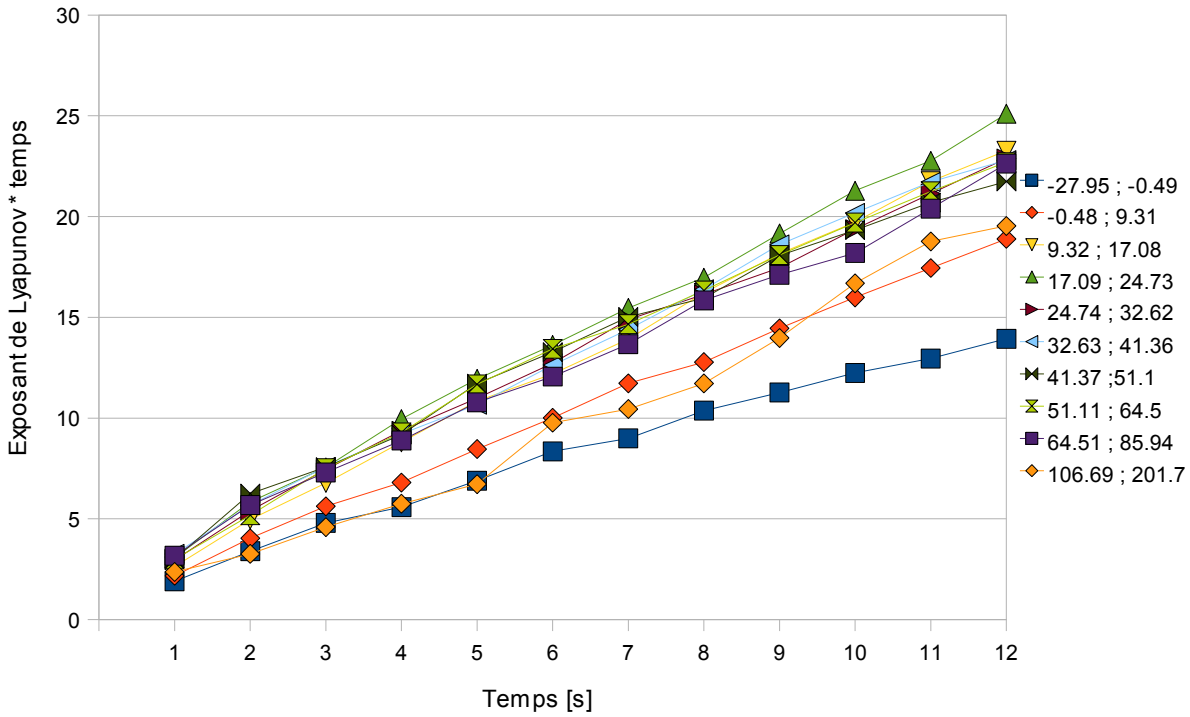


Figure 18: Évaluations par classes d'énergies d'exposants de Lyapunov, représentés par la pente

9 cf. Annexe 6
10 cf. Annexe 6

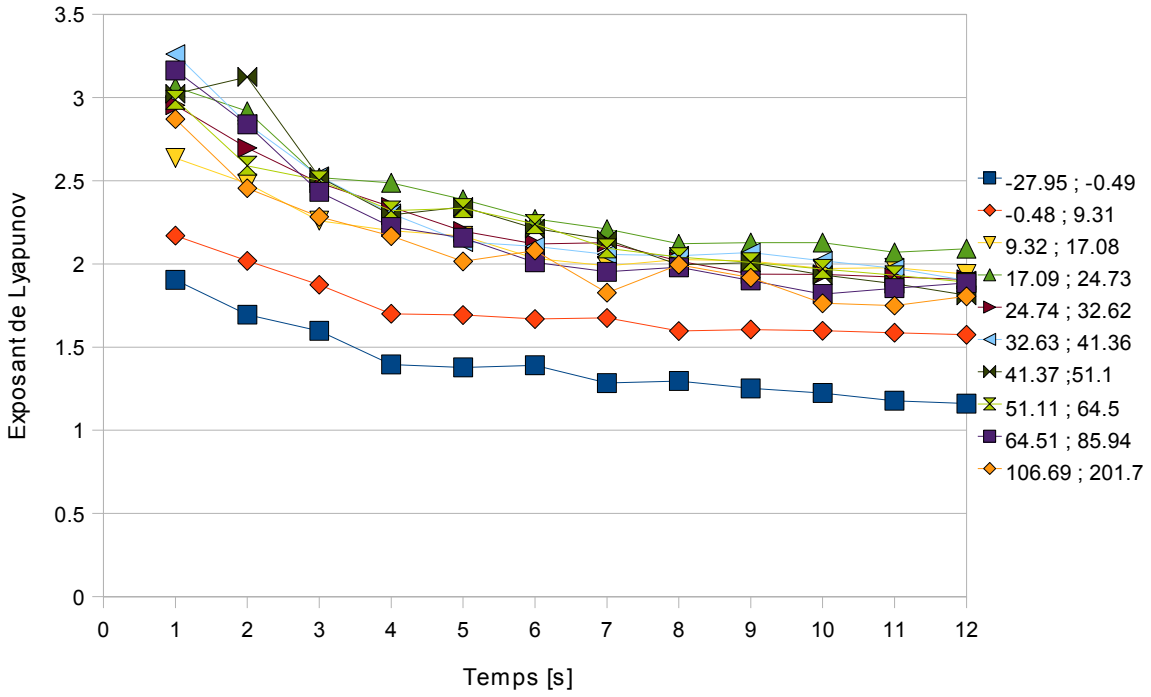


Figure 19: Évaluations par classes d'énergies d'exposants de Lyapunov

On peut alors interpréter le comportement de l'exposant de Lyapunov de la manière suivante: lorsqu'il y a peu d'énergie mécanique, le système se comporte comme deux systèmes faiblement couplés, donc de manière semblable à deux pendules simples. Par contre, lorsque l'on injecte beaucoup d'énergie, les énergies potentielles sont très faibles par rapport aux énergies cinétiques. Les vitesses angulaires sont alors élevées, et les deux masses tournent continuellement autour de leur point d'attache respectif. Dans ces deux cas, la sensibilité aux conditions initiales est réduite. Cette sensibilité est plus élevée au moment où le système possède plus de possibilités de diverger, c'est-à-dire, lorsque l'énergie totale du système a une valeur intermédiaire.

La figure 20 représente la valeur de l'exposant de Lyapunov au temps t=10s en fonction de l'énergie mécanique totale. Ce graphique confirme le fait que le chaos est moins important lorsque le système possède une énergie mécanique totale faible ou, à l'inverse, très forte. Le chaos est alors présent au maximum pour des niveaux d'énergie intermédiaires (figure 8, cas montrés sur les figures a et b).

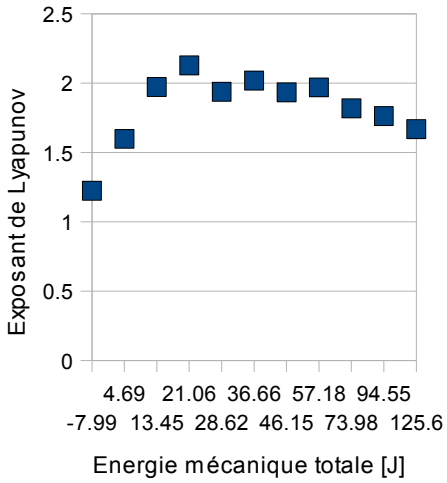


Figure 20: Évaluation de l'exposant de Lyapunov en fonction de l'énergie mécanique totale après 10 s

11. Conclusion

Au cours de ce travail, seul le cas où les longueurs des deux tiges et les deux masses sont égales à 1 a été traité. Il serait intéressant d'observer le comportement du pendule double pour des valeurs différentes de ces paramètres, par exemple en les fixant de sorte à ce que les deux pendules possèdent des caractéristiques différentes. De plus, le cas traité est idéal et imperméable à toute intervention extérieure. Si on considérait le caractère dissipatif, dû à l'influence des frottements, l'état final du pendule serait caractérisé par la valeur nulle de tous ses paramètres.

Le pendule double est un système mécanique simple car il n'est défini que par cinq paramètres fixes (les masses, les longueurs des tiges et l'accélération gravitationnelle) et quatre conditions initiales (les angles et les vitesses angulaires) et ne possède que trois degrés de liberté en raison de la loi de conservation de l'énergie. Les équations de son mouvement ont été présentées et ne peuvent être résolues que par des méthodes numériques.

La résolution de ces équations montre une grande sensibilité aux conditions initiales, que l'on observe notamment grâce à une représentation graphique d'un état au temps $t>0$ en fonction des paramètres initiaux. La précision donnée par les calculs numériques ne permet pas de connaître exactement l'état du système pour des temps éloignés.
La quantité de chaos peut être exprimée par le calcul de l'exposant de Lyapunov. Dans tous les cas, sa valeur est positive, ce qui démontre la présence du chaos pour toutes les conditions initiales, exception faite d'un système immobile. L'énergie mécanique globale du système influence l'exposant de Lyapunov, donc l'importance du chaos dans le système.

On remarque alors qu'il n'est pas nécessaire de se focaliser sur des systèmes complexes, influençables par de multiples paramètres, pour observer la présence du chaos. Celui-ci peut, en effet, occuper une place déterminante dans des systèmes mécaniques simples, tels le pendule double.

Depuis Laplace, le chaos a été appréhendé par les physiciens et fait, aujourd'hui encore, l'objet de nombreux travaux scientifiques, de publications et de revues dédiées.

12. Bibliographie

1. PRESS, William H,TEUKOLSKY, Saul A, VETTERLING, William T, FLANNERY, Brian P. *Numerical Recipes in C: The Art of Scientific Computing.* 2e éd corr. New York: Press Syndicate of the University of Cambridge, 1994. chap. 16, Integration of Ordinary Differential Equations, p.707-752.

2. KERNIGHAN, Brian W, RITCHIE, Dennis M. *Le langage C: C ANSI.* 2e éd. Paris: Masson S.A., 1994. (Collection MIM-Languages).

3. WITHERDEN, Freddie. Double Pendulum. In WITHERDEN [pdf]. http://freddie.witherden.org/tools/doublependulum/report.pdf.

4. WIKIPEDIA. Pendule double. In WIKIPEDIA [Page web]. http://fr.wikipedia.org/wiki/Pendule_double.

5. MANNEVILLE, Paul. Dynamique non linéaire et chaos. In MANEVILLE [pdf]. http://e2phy.in2p3.fr/2005/documents/apres_ecole/Textes/Manneville_txt.pdf.

6. RAMASUBRAMANIAN, K, SRIRAM, M.S. A comparative study of computation of Lyapunov spectra with different algorithms. In RAMASUBRAMANIAN [pdf]. http://ariv.org/PS_cache/chao-dyn/pdf/9909/9909029v1.pdf.

7. STACHOWIAK, Tomasz, OKADA, Toshio. A numerical analysis of chaos in the double pendulum. In *Chaos, Solitons and Fractals*, 2006, vol.29, n.2, p. 417-422.

8. WOLF, Alan, SWIFT, Jack B, SWINNEY, Harry L, VASTANO, John A. Determining Lyapunov exponents from a time series. In *Physica D: Nonlinear Phenomena*, 1985, vol.16, n.3, p. 285-317.

9. GROUSSON, Mathieu. Le chaos est finalement partout. In *Sciences & Vie*, avril 2010, n.1111, p.88-95.

10. PÉPIN, François. Hasard et déterminisme. In *Pour la Science: Hasard et incertitude*, novembre 2009, n.385, p.24-29.

11. BUZZI, Jérôme. *Chaos et stabilité*. Paris: Le Pommier, 2005, p.52-75. (Quatre à Quatre: Mathématiques)

12. WIKIPEDIA. Dérivée. In WIKIPEDIA [Page web]. http://fr.wikipedia.org/wiki/D%C3%A9riv%C3%A9e

13. WIKIPEDIA. Matrice (mathématiques). In WIKIPEDIA [Page web]. http://fr.wikipedia.org/wiki/Matrice_(math%C3%A9matiques).

13. Annexes

Annexe 1: Programme C résolvant numériquement les équations différentielles décrivant le mouvement du pendule double

Le programme ci-dessous résout ces équations pour les conditions initiales θ_1 =0, θ_2 = π/4, $\dot{\theta}_1$ =0, $\dot{\theta}_2$ =1 pour t allant de 1 à 10s et ouvre un fichier contenant les valeurs des résultats ainsi obtenus.
Les routines *rkck*, *rkqs* et *odeint* sont tirées du *Numerical Recipies in C*.

```
#include <QtCore/QCoreApplication>
#include <cmath>
#include <iostream>
#include <iomanip>
#include <limits>
#include <time.h>
#include "nr.h"
using namespace std;

// Driver for routine odeint
DP dxsav; // defining declarations
int kmax,kount;
Vec_DP *xp_p;
Mat_DP *yp_p;


//définition des constantes
const double m1=1;
const double m2=1;
const double l1=1;
const double l2=1;
const double g=9.81;
int nrhs; // counts function evaluations

// routine rkck
void NR::rkck(Vec_I_DP &y, Vec_I_DP &dydx, const DP x,
      const DP h, Vec_O_DP &yout, Vec_O_DP &yerr,
      void derivs(const DP, Vec_I_DP &, Vec_O_DP &)) { … }

//routine rkqs
void NR::rkqs(Vec_IO_DP &y, Vec_IO_DP &dydx, DP &x, const DP htry,
      const DP eps, Vec_I_DP &yscal, DP &hdid, DP &hnext,
      void derivs(const DP, Vec_I_DP &, Vec_O_DP &)) { … }

//routine odeint
extern DP dxsav;
extern int kmax,kount;
extern Vec_DP *xp_p;
extern Mat_DP *yp_p;
void NR::odeint(Vec_IO_DP &ystart, const DP x1, const DP x2, const DP eps,
      const DP h1, const DP hmin, int &nok, int &nbad,
      void derivs(const DP, Vec_I_DP &, Vec_O_DP &),
      void rkqs(Vec_IO_DP &, Vec_IO_DP &, DP &, const DP, const DP,
      Vec_I_DP &, DP &, DP &, void (*)(const DP, Vec_I_DP &, Vec_O_DP &)))
      { … }

//résolution des équations
```

```
void derivs(const DP x, Vec_I_DP &y, Vec_O_DP &dydx)
{
      nrhs++;

      const double M=m1+m2;
      const double delta=y[2]-y[0];

      //dydx[0]=theta1, dydx[1]=omega1, dydx[2]=theta2, dydx[3]=omega2
      dydx[0]=y[1];
      dydx[1]= (m2*l1*y[1]*y[1]*sin(delta)*cos(delta)
      +m2*g*sin(y[2])*cos(delta)+m2*l2*y[3]*y[3]*sin(delta)-M*g*sin(y[0]))/
      (M*l1-m2*l1*cos(delta)*cos(delta));
      dydx[2]=y[3];
      dydx[3]= (-m2*l2*y[3]*y[3]*sin(delta)*cos(delta)
      +M*(g*sin(y[0])*cos(delta)-l1*y[1]*y[1]*sin(delta)-g*sin(y[2])))/
      (M*l2-m2*l2*cos(delta)*cos(delta));
}

int equsolvestep(void)
{
      const int N=4, KMAX=100;
      const double PI=atan(1)*4;
      int i,nbad,nok;
      DP eps=1.0e-4,h1=0.1,hmin=0.0,x1=0.0,x2=10.0;
      Vec_DP ystart(N);

      kmax=KMAX;
      xp_p=new Vec_DP(KMAX);
      yp_p=new Mat_DP(N,KMAX);
      Vec_DP &xp=*xp_p;
      Mat_DP &yp=*yp_p;

      cout << endl << setw(10) << "t" << setw(15) << "theta_1";
      cout << setw(15) << "omega_1" << setw(15) << "theta_2";
      cout << setw(15) << "omega_2" << endl;

      FILE * pFile;
      pFile = fopen ("test.txt","w");

      //fonction
      for(x2=1.0;x2<=10.0;x2+=1.0) //valeurs de t
      {
            ystart[0]=0; //theta1 initial
            ystart[1]=0; //omega1 initial
            ystart[2]=PI/4; //theta2 initial
            ystart[3]=1; //omega2 initial
            nrhs=0;
            dxsav=(x2-x1)/20.0;
            NR::odeint(ystart,x1,x2,eps,h1,hmin,nok,nbad,derivs,NR::rkqs);
            i=kount-1;
            printf("%f\t%f\t%f\t%f\t%f\n",xp[i],yp[0][i],yp[1][i],yp[2]
            [i],yp[3][i]);
            fprintf(pFile,"%f\t%f\t%f\t%f\t%f\n",xp[i],yp[0][i],yp[1]
            [i],yp[2][i],yp[3][i]);
      }

      fclose (pFile);
      delete yp_p;
```

```
	delete xp_p;
	return 0;
}

int main(int argc, char *argv[])
{
	QCoreApplication a(argc, argv);

	equsolvestep();

	return a.exec();
}
```

Annexe 2: Fonction *espacedephase* utilisée pour la représentation sous forme de diagrammes

La fonction *espacedephase* doit être ajoutée à la suite du programme de l'annexe 1. Elle permet de résoudre les équations du système en faisant varier deux des conditions initiales.

```
int espacedephase(void)
{
	const int N=4, KMAX=100;
	const double PI=atan(1)*4;
	int i,j,nbad,nok;
	DP eps=1.0e-4,h1=0.1,hmin=0.0,x1=0.0,x2=1.0; //temps t
	Vec_DP ystart(N);
	double th1,th2;

	xp_p=new Vec_DP(KMAX);
	yp_p=new Mat_DP(N,KMAX);
	Vec_DP &xp=*xp_p;
	Mat_DP &yp=*yp_p;

	FILE * pFile[4];
	char s[20];
	for(i=0;i<4;i++)
	{
		sprintf(s,"test%d.txt",i);
		pFile[i] = fopen (s,"w");
	}

	for(j=0;j<4;j++)
	{
		fprintf(pFile[j],"%9.6f",0.0);
		for(th1=-PI/2;th1<=PI/2+eps;th1+=(PI/180)) fprintf(pFile[j],"\t
		%9.6f",th1*180/PI);
		fprintf(pFile[j],"\n");
	}

	for(th2=-PI/2;th2<=PI/2+eps;th2+=(PI/180)){
		for(j=0;j<4;j++)fprintf(pFile[j],"%9.6f",th2*180/PI);
		for(th1=-PI/2;th1<=PI/2+eps;th1+=(PI/180))
			{
			ystart[0]=th1; //theta1 initial
			ystart[1]=0.0; //omega1 initial
			ystart[2]=th2; //theta2 initial
			ystart[3]=0.0; //omega2 initial
			nrhs=0;
			dxsav=(x2-x1)/20.0;
```

```
                kmax=KMAX;

                NR::odeint(ystart,x1,x2,eps,h1,hmin,nok,nbad,derivs,
                NR::rkqs);

                i=kount-1;
                for(j=0;j<4;j++) fprintf(pFile[j],"\t%9.6f",yp[j]
                [i]*180/PI);
                }
            for(j=0;j<4;j++) fprintf(pFile[j],"\n");
      }
      printf("Fin\n");

      delete yp_p;
      delete xp_p;
      return 0;
}

int main(int argc, char *argv[])
{
      QCoreApplication a(argc, argv);

      espacedephase();

      return a.exec();
}
```

Annexe 3: Représentation sous forme de diagrammes

Le programme ci-dessous permet de représenter graphiquement les valeurs tirées d'un document extérieur, par exemple les valeurs d'un des fichiers retournés par la fonction espacedephase de l'annexe 4.

```
read a 'C:/Documents and Settings/Pierre/Mes documents/Documents GYB/Travail
de maturite/Penduledouble_1/debug/m=l=1,w1=w2=0,th1-th2,th1-t=1s.txt'
crop a 1 1000 'x'
crop a 1 1000 'y'
rotate 0 0
aspect 1 1.41 1
axis -90 -90 -30*pi 90 90 30*pi
xtick -90 '-90' -45 '-45' 0 '0' 45 '45' 90 '90'
axis
box
tile a
title '\theta 1(t), t=1sec'
xlabel '\theta 1'
ylabel '\theta 2'
colorbar
```

Annexe 4: Évaluations locales de l'exposant de Lyapunov à l'aide de MuPAD

Le code MuPAD ci-dessous calcule et représente graphiquement l'exposant de Lyapunov pour t allant de 0 à 20s pour deux trajectoires du système proches en utilisant les distances entre les angles et les vitesses angulaires.

```
m1:= 1:
```

```
m2:= 1:
l1:= 1:
l2:= 1:
g:= 9.81:
M:= m1+m2:
A:= a2-a1:
B:=a4-a3:
ai1:= PI/2:
ai2:= PI/2:
vi1:= 0:
vi2:= 0:
dai1:= 10^(-8):
dvi1:= 10^(-8):
dai2:= 10^(-8):
dvi2:= 10^(-8):

IVP1:={     a1''(t) = (m2 * l1 * (a1'(t))^2 * sin (A) * cos (A) + m2 * g *
            sin (a2) * cos (A) + m2 * l2* (a2'(t))^2 * sin (A) - M * g * sin
            (a1)) / ( M*l1 - m2 * l1 * (cos (A))^2) ,
            a2''(t) = (-m2 * l2 *(a2'(t))^2 * sin (A) * cos (A) + M * (g *
            sin (a1) * cos (A) - l1 * (a1'(t))^2 * sin (A) - g * sin (a2))) /
            (M * l2 - m2 * l2 * (cos(A))^2) ,
            a1(0) = ai1 , a1'(0) = vi1 , a2(0) = ai2 , a2'(0) = vi2 }:
Fields1:= [ a1(t), a1'(t) , a2(t) , a2'(t)]:
Equadiff1:= numeric::ode2vectorfield(IVP1,Fields1):
Y1:=numeric::odesolve2(Equadiff1):

IVP2:={     a3''(t) = (m2 * l1 * (a3'(t))^2 * sin (B) * cos (B) + m2 * g *
            sin (a4) * cos (B) + m2 * l2* (a4'(t))^2 * sin (B) - M * g * sin
            (a3)) / ( M*l1 - m2 * l1 * (cos (B))^2) ,
            a4''(t) = (-m2 * l2 * (a4'(t))^2 * sin (B) * cos (B) + M * (g *
            sin (a3) * cos (B) - l1 * (a3'(t))^2 * sin (B) - g * sin (a4))) /
            (M * l2 - m2 * l2 * (cos(B))^2) ,
            a3(0) =ai1 + dai1 , a3'(0) = vi1 + dvi1 , a4(0) = ai2 + dai2 ,
            a4'(0) = vi2 + dvi2 }:
Fields2:= [ a3(t), a3'(t) , a4(t) , a4'(t)]:
Equadiff2:= numeric::ode2vectorfield(IVP2,Fields2):
Y2:=numeric::odesolve2(Equadiff2):

d:= t-> sqrt((Y1(t)[1]-Y2(t)[1])^2+(Y1(t)[3]-Y2(t)[3])^2+(Y1(t)[2]-Y2(t)
[2])^2+(Y1(t)[4]-Y2(t)[4])^2):

PlotexpLya:= plot::Function2d(ln(d(t)/d(0)), t=0..20):
plot(PlotexpLya);
```

Pour calculer l'exposant de Lyapunov en utilisant les distances entre les énergies, il suffit de remplacer, dans le code MuPAD précédent, la ligne

```
d:= t-> sqrt((Y1(t)[1]-Y2(t)[1])^2+(Y1(t)[3]-Y2(t)[3])^2+(Y1(t)[2]-Y2(t)
[2])^2+(Y1(t)[4]-Y2(t)[4])^2):
```

par

```
d:= t-> sqrt (   (0.5 * m1 * l1^2 * (Y1(t)[2])^2 - 0.5 * m1 * l1^2 * (Y2(t)
                 [2])^2)^2 +
                 (0.5 * m2 * ( l1^2 * (Y1(t)[2])^2 + l2^2 * (Y1(t)[4])^2 +
                 2*l1*l2*(Y1(t)[2])*(Y1(t)[4])*cos((Y1(t)[1])-(Y1(t)[3])) )
                 - 0.5 * m2 * ( l1^2 * (Y2(t)[2])^2 + l2^2 * (Y2(t)[4])^2 +
                 2*l1*l2*(Y2(t)[2])*(Y2(t)[4])*cos((Y2(t)[1])-(Y2(t)
                 [3])) ))^2 +
                 ((- (m1 * g * cos (Y1(t)[1]) * l1)) - (- (m1 * g * cos
```

```
                (Y2(t)[1]) * l1)))^2 +
                ((- (m2 * g * (cos(Y1(t)[1])*l1 + cos(Y1(t)[3])*l2))) - (-
                (m2 * g * (cos(Y2(t)[1])*l1 + cos(Y2(t)[3])*l2))))^2):
```

Annexe 5: Fonction *setdedonnee*, évaluant les exposants de Lyapunov pour 10000 sets de conditions initiales tirées au hasard

Les fonctions ci-dessous doivent être ajoutées aux fonctions de l'annexe 1. Elles permettent de calculer les exposants de Lyapunov pour 10000 sets de conditions initiales tirées au hasard en calculant les distances entre des paramètres initiaux très proches à l'aide des dérivées.

```
DP ran2(int &idum)
{
      const int IM1=2147483563,IM2=2147483399;
      const int IA1=40014,IA2=40692,IQ1=53668,IQ2=52774;
      const int IR1=12211,IR2=3791,NTAB=32,IMM1=IM1-1;
      const int NDIV=1+IMM1/NTAB;
      const DP EPS=3.0e-16,RNMX=1.0-EPS,AM=1.0/DP(IM1);
      static int idum2=123456789,iy=0;
      static Vec_INT iv(NTAB);
      int j,k;
      DP temp;
      if (idum <= 0) {
            idum=(idum==0 ? 1: -idum);
            idum2=idum;
            for (j=NTAB+7;j>=0;j--) {
                  k=idum/IQ1;
                  idum=IA1*(idum-k*IQ1)-k*IR1;
                  if (idum < 0) idum += IM1;
                  if (j < NTAB) iv[j] = idum;
            }
            iy=iv[0];
      }
      k=idum/IQ1;
      idum=IA1*(idum-k*IQ1)-k*IR1;
      if (idum < 0) idum += IM1;
      k=idum2/IQ2;
      idum2=IA2*(idum2-k*IQ2)-k*IR2;
      if (idum2 < 0) idum2 += IM2;
      j=iy/NDIV;
      iy=iv[j]-idum2;
      iv[j] = idum;
      if (iy < 1) iy += IMM1;
      if ((temp=AM*iy) > RNMX) return RNMX;
      else return temp;
}

DP func4(DP x){
      const int N=4, KMAX=100;
      const double PI=atan(1)*4;
      int i,j,nbad,nok;
      DP eps=1.0e-6; //1.0e-4
      DP h1=0.1,hmin= 0.0,x1=0.0,x2=gt;
      Vec_DP ystart(N);
      double yt,ep1,ep2,ec1,ec2;
```

```
	double a,b,c,s1,s2;
	double th1,th2,w1,w2;
	double plya;

	th1 = gtwtw[0]; //theta1 initial
	w1 = gtwtw[1]; //omega1 initial
	th2 = gtwtw[2]; //theta2 initial
	w2 = gtwtw[3]; //omega2 initial

	switch (ge0i){
		case 0:
			th1 += x;
			break;
		case 1:
			w1 += x;
			break;
		case 2:
			th2 += x;
			break;
		case 3:
			w2 += x;
			break;
	}

	ystart[0]= th1;
	ystart[1]= w1;
	ystart[2]= th2;
	ystart[3] = w2;

	nrhs=0;
	x1=0.0;
	x2=gt;
	dxsav=(x2-x1)/20.0;
	kmax=KMAX;
	NR::odeint(ystart,x1,x2,eps,h1,hmin,nok,nbad,derivs,NR::rkqs);

	switch (geti){
		case 0:
			yt = ystart[0];
			break;
		case 1:
			yt = ystart[1];
			break;
		case 2:
			yt = ystart[2];
			break;
		case 3:
			yt = ystart[3];
			break;
	}

return yt;
}

int setdedonnees(void) {
	const int N=4, KMAX=100;
	const double PI=atan(1)*4;
	double energie,ep1,ep2,ec1,ec2;
```

```
double th1,th2,w1,w2;
double h=0.01,detde0,err,som2,etot;
double d,x,f0,dx,a1,b1;
long i,j;
int idum=(-1);
double a[5]={1.0/12.0,-8.0/12.0,0,8.0/12.0,-1.0/12.0};
double y[5];
double b[4];

xp_p=new Vec_DP(KMAX);
yp_p=new Mat_DP(N,KMAX);
Vec_DP &xp=*xp_p;
Mat_DP &yp=*yp_p;

FILE * pFile;
pFile = fopen ("testlyasdd3.txt","w");

fprintf(pFile,"th1\tw1\tth2\tw2\tt");
for(gt=0;gt<=100.01;gt+=1.0)
      fprintf(pFile,"\t%f",gt);
fprintf(pFile,"\n");

for (i=0;i<10000;i++){
      th1= (ran2(idum)*172+4);
      if (ran2(idum)>0.5)
      th1*=-1;
      th1*=PI/180;

      th2=(ran2(idum)*172+4);
      if (ran2(idum)>0.5)
      th2*=-1;
      th2*=PI/180;

      do {
      w1=(ran2(idum)*1000-500); // vitesses -100;100
      w1*=PI/180;
      } while (abs(w1) < 0.015);

      do {
      w2=(ran2(idum)*1000-500); //vitesses -100;100
      w2*=PI/180;
      } while (abs(w2) < 0.015);

      gtwtw[0]=th1;
      gtwtw[1]=w1;
      gtwtw[2]=th2;
      gtwtw[3]=w2;

      printf("%f\t%f\t%f\t%f\n\n",th1,w1,th2,w2);
      fprintf(pFile,"%f\t%f\t%f\t%f\t",th1,w1,th2,w2);

      etot= -g*m1*l1*cos(gtwtw[0]);
      etot+=0.5*m1*l1*l1*gtwtw[1]*gtwtw[1];
      etot+= -g*m2*(l1*cos(gtwtw[0]) + l2*cos(gtwtw[2]));
      etot+= 0.5*m2*(l1*l1*gtwtw[1]*gtwtw[1] + l2*l2*gtwtw[3]*gtwtw[3]
      + 2*l1*l2*gtwtw[1]*gtwtw[3]*cos(gtwtw[0]-gtwtw[2]));

      fprintf(pFile,"%f",etot);
```

```
            printf("%d\t%f",i,etot);

            gt=0;
            for(ge0i=0;ge0i<=3;ge0i++){
                  geti=ge0i;

                  dx=2e-12;
                  b1=0.0;
                  for(j=-2;j<=2;j++){
                        b1 +=func4(dx*j)*a[j+2];
                  }
                  b1 = b1/dx;
                  b[ge0i]=b1;
            }

            for(gt=0;gt<=20.01;gt+=1.0){
                  som2=0;
                  for(ge0i=0;ge0i<=3;ge0i++){
                        geti=ge0i;
                        dx=2e-12;
                        a1=0.0;
                        for(j=-2;j<=2;j++){
                              a1 +=func4(dx*j)*a[j+2];
                        }
                        a1 = a1/dx;
                        if(a1<0)
                              a1*=-1;

                        som2+=log(a1/b[ge0i]);
                  }
                  fprintf(pFile,"\t%f",som2/(4*gt));
            }
            fprintf(pFile,"\n");
      }
      printf("Fin\n");
      return 0;
}

int main(int argc, char *argv[])
{
      QCoreApplication a(argc, argv);

      setdedonnees();

      return a.exec();
}
```

Annexe 6 :

Les exposants de Lyapunov représentés sur les figures 16, 18, 19 et 20 ne sont pas exactement les exposants maximum de chaque classe d'énergie. En effet, le maximum est approximé par un grand nombre de tirages représentant une distribution. La plus grande valeur du tirage s'approche du maximum de manière aléatoire. Cette valeur est donc « statistiquement peu fiable ». En prenant une valeur à 99% de la distribution (soit la 10ème plus grande valeur) on obtient une valeur « statistiquement plus stable », qui semble suivre correctement le comportement du maximum absolu.

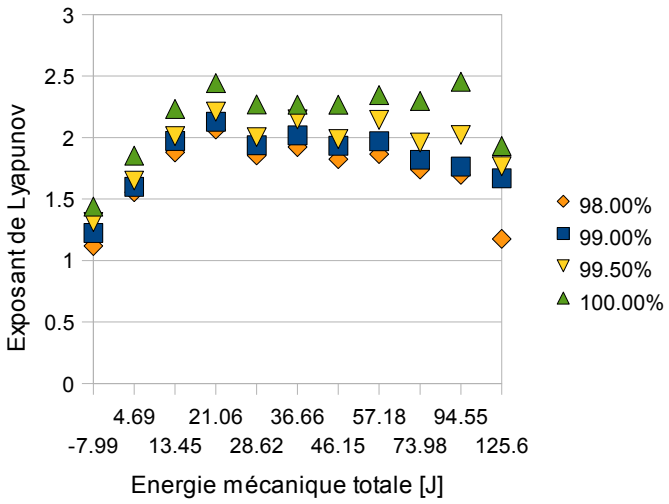


Figure 21: Évaluations d'exposants de Lyapunov en fonction de l'énergie mécanique totale après 10 s à différents pourcentages de la distribution

La figure 21 représente la valeur au temps t=10s d'exposants de Lyapunov obtenus à différents pourcentages de la distribution.

14. Attestation sur l'honneur

Je soussignée, Noémie Jaquier, demeurant au 12 Rue Centrale, 1683 Chesalles sur Moudon, atteste sur l'honneur que mon travail de maturité *Le pendule double, un système chaotique* a été accompli conformément aux conditions de réalisation du travail de maturité, sans fraude, ni plagiat, ni collusion.

Chesalles, le 29.06.2010.

Noémie Jaquier