



INSTITUT
NATIONAL
des SCIENCES
APPLIQUÉES



Projet de physique P6-3
STPI06 / P6-3 / 2008 - 5



Nom des étudiants

Myriam DE SAINT JEAN

Mathilde PIERRE-BOURGEON

Selima MASMOUDI

Amal ROCHD

Martin ROSALIE

Enseignant responsable du projet

Jerôme YON



Etude du mouvement
"chaotique" d'une masse
accrochée à un ressort



À TAILLE
HUMAINE
À L'ECHELLE
DU MONDE

Date de remise du rapport : 22/06/08

Référence du projet : STPI¹/P6-3/2008 – 5

Intitulé du projet : Etude du mouvement "chaotique" d'une masse accrochée à un ressort.

Type de projet : simulation et analyse numérique.

Objectifs du projet :

L'objectif de ce projet est de découvrir et mettre en évidence le mouvement chaotique d'une masse accrochée à un ressort. Nous avons d'abord cherché à caractériser le chaos avant de pouvoir le mettre en évidence. Nous avons pour cela suivi méticuleusement la démarche proposée par M. Yon dans le sujet qu'il nous a fourni (cf. Annexe A) en début de projet. En voici les grandes lignes :

1. Choisir le système
2. Résoudre numériquement les équations différentielles
3. Visualiser les résultats
4. Réaliser une analyse bibliographique
5. Analyser les résultats

Numéro de cahier de laboratoire associé : A30222

Remerciements :

Nous tenons à remercier vivement M. Yon, pour sa disponibilité et son encadrement qui nous ont permis de mener à bien ce projet sur tous les plans de Poincaré.

Nous souhaitons également remercier M. Letellier pour ses nombreux conseils, son écoute, sa compréhension, mais surtout l'intérêt qu'il a porté à l'égard de notre sujet.

Nous adressons nos vifs remerciements à MM. Goglu, Delapierre et Rello, pour s'être penchés sur notre problème dans sa dimension mathématiques.

Aussi, nous aimerions remercier M. Suire pour sa précieuse et spontanée aide lors de nos déboires en programmation.

Enfin, nous souhaiterions adresser un clin d'œil à M. Cabanial qui dès le début nous a avertis : "*Ouh là là, le chaos, c'est très compliqué ...*".

¹INSTITUT NATIONAL DES SCIENCES APPLIQUÉES DE ROUEN
DÉPARTEMENT SCIENCES ET TECHNIQUES POUR L'INGÉNIEUR
BP 8 –PLACE EMILE BLONDEL- 76131 MONT-SAINT-AIGNAN - TÉL : 33 2 35 52 83 00 - FAX : 33 2 35 52 83 69

Table des matières

Notations	5
Introduction	6
1 Méthodologie, organisation du travail	7
2 Travail réalisé et résultats	8
2.1 Modélisation du problème à l'aide d'équations différentielles	8
2.1.1 Mise en équations du problème	8
2.1.2 Conditions d'équilibre	10
2.2 Méthode de résolution des équations différentielles	12
2.2.1 Principe du Runge Kutta 4	12
2.2.2 Code en Java	15
2.3 Illustration des résultats obtenus	16
2.3.1 Principe mathématiques : projection	16
2.3.2 Code en Java	16
2.3.3 Code de Scilab	16
2.4 Analyse bibliographique des études des mouvements chaotiques	17
2.4.1 Notion de système dynamique et de chaos	17
2.4.2 Un peu d'histoire	18
2.4.3 Notion d'espace des phases	18
2.4.4 Notion d'attracteur	19
2.5 Analyse du mouvement au sens de la dynamique des systèmes	20
2.5.1 Choix des points à étudier	21
2.5.2 Outils d'étude et intérêts	22
2.5.3 Point d'équilibre : $r = 0,15$	22
2.5.4 Point Quasi-périodique : $r = 0,103\ 106\ 2$	23
2.5.5 Bouffée Chaotique : $r = 0,042\ 2$	24
2.5.6 Fenêtre stable : $r = 0,039\ 15$	25
2.5.7 Chaos : $r = 0,032$	26
2.5.8 Conclusion de l'étude	27
Conclusion et perspectives	28
Bibliographie	29
A Sujet	30
B Compte Rendu de la rencontre avec M.Christophe LETELLIER	31
B.1 Diagramme de bifurcation	31
B.2 Notion de fréquence	31

B.3	Notion de systèmes dissipatif et conservatif	32
B.4	Notion d'espace des phases	32
C	Code en Java	33
C.1	Classe principale	33
C.2	Classe du choix des conditions initiales	33
C.3	Classe de résolution des équations différentielles couplées	40
C.4	Classe de représentation graphique	45
C.5	Classe d'analyse	52
D	Code de Scilab	71
E	Graphiques et Analyse	72
E.1	Evolution du rayon en fonction de l'allongement par rapport à l'équilibre . .	72
E.2	Etude de l'état d'équilibre	73
E.3	Etude de l'état quasi-périodique	74
E.4	Etude de la bouffée chaotique	76
E.5	Etude de la fenêtre stable	78
E.6	Etude du chaos	80

Notations

Espace des phases : espace engendré par toutes les grandeurs dont la connaissance à un instant t est nécessaire pour déterminer l'évolution temporelle d'un système. Chaque point de l'espace des phases caractérise de façon unique l'état du système à un instant donné.

Système chaotique : système qui présente une grande sensibilité aux conditions initiales, mais qui évolue dans un espace des phases délimité.

Section de Poincaré : coupe, par un plan, dans l'espace de phase. C'est un sous-espace de l'espace des phases, défini par une valeur d'une coordonnée de phase. On considère ses points d'intersection avec la trajectoire dans l'espace des phases.

Attracteur : domaine de l'espace des phases vers lequel converge le système.

Fractale : objet de dimension fractionnelle, au contraire d'objet comme une droite, une surface, ou un volume dont les dimensions sont entières (1,2 ou 3).

Système continu : système dont les variables évoluent continûment, de sorte qu'à chaque instant t on puisse associer un état du système. Son évolution dans le temps peut alors être déterminée par un système d'équations différentielles qui indique, dans un état donné du système, avec quelle vitesse chaque variable évolue.

Système discret : système dont les variables se modifient par palier d'un instant t_n à un instant t_{n+1} sans que l'état du système entre ces deux instants soit connu.

Système conservatif : système pour lequel l'énergie mécanique est conservée (invariante au cours du temps).

Système dissipatif : système qui ne conserve pas l'énergie, qui perd de l'énergie au cours de son évolution temporelle.

Temps de Lyapunov : temps au bout duquel la distance entre deux trajectoires de phases est multipliée par e .

Introduction

La connaissance de l'évolution des systèmes physiques a toujours fasciné les scientifiques et intrigué tous les autres.

Ainsi, les scientifiques ont longtemps cherché à décrire l'évolution des systèmes, en particulier le système solaire. La connaissance des équations régissant le mouvement de celui-ci a donc été très convoitée. On avait fait de l'univers un système géant muni de lois relativement simples qu'il suffisait de trouver. La théorie du chaos a quant à elle prouvé que connaître les équations d'un système ne permet en rien de déterminer l'évolution à long terme du système. La dynamique des systèmes, ou théorie du chaos, fut donc à ses débuts réfutée par bon nombre de scientifiques au raisonnement cartésien qui ne pouvaient croire en un certain hasard. Tout système non linéaire peut, en théorie, être source de chaos. Celui-ci dépend alors très fortement des conditions initiales.

Ainsi, nous avons étudié une simple masse attachée à un ressort et tenté de déterminer son caractère chaotique. Notre but était donc de déterminer si ce système physique était réellement chaotique et non un simple déplacement désordonné. Cependant, il est important de signaler que nous avons uniquement fixé le point d'attache du ressort de telle sorte que la masse ait un maximum de liberté dans l'espace physique.

Afin de présenter notre étude, on retrouvera dans un premier temps les équations différentielles du mouvement dont on déterminera les solutions grâce à la méthode de résolution numérique Runge-Kutta 4.

Ensuite, les différents programmes permettant la visualisation du mouvement ainsi que l'utilisation d'outils permettant l'analyse d'un tel système seront décrits.

L'objectif majeur de ce projet étant d'appréhender la complexe théorie du chaos, une analyse approfondie de la théorie sera exposée dans l'analyse bibliographique.

Enfin nous exposerons les différents caractères périodiques, quasi-périodique et chaotique, de notre pendule élastique ; tout ceci à travers une étude détaillée dans l'espace physique et dans l'espace des phases afin d'obtenir une représentation cohérente du chaos.

Chapitre 1

Méthodologie, organisation du travail

Nous allons, dans cette partie, vous présenter la démarche que nous avons suivie dans le cadre de ce projet.

Tout d'abord, la mise en équations du problème était indispensable. Ainsi, chacun a essayé d'élaborer les équations différentielles du mouvement pour ensuite mettre en commun nos résultats. Une fois ces équations connues, nous savions que la méthode de résolution du Runge-Kutta 4 serait très pratique. Cependant, il nous a paru indispensable de vérifier la cohérence des résultats en se basant sur les équations de conditions d'équilibre plus faciles à déterminer. Plusieurs approches ont alors été abordées telles que la résolution avec la méthode de la transformée de Laplace ... mais beaucoup se sont avérées infructueuses.

Il fallait donc maintenant débiter la programmation afin de résoudre ces complexes équations différentielles. Martin ayant déjà étudié la méthode de résolution du Runge-Kutta à l'occasion d'un autre projet, il entreprit sa programmation. Ensuite, des premières ébauches de programmes permettant la visualisation de la trajectoire ont été effectuées par Martin et Myriam qui maîtrisaient mieux l'outil informatique. Il nous a alors paru judicieux de nous séparer afin que nous approfondissions en même temps la programmation et l'analyse du chaos.

Rapidement, les outils d'étude du chaos nous ont permis de visualiser l'évolution de notre système. Chacun a pu s'approprier le programme en cherchant des points judicieux et comportements particuliers. Par ailleurs, le caractère chaotique du pendule élastique nous semblait de moins en moins certain. Ainsi, Amal, Myriam, Mathilde et Selima ont repris l'étude de cas chaotiques bien connus comme l'évolution de la population et les travaux de Lorenz au cas où notre pendule ne serait pas chaotique. Les programmes informatiques ont alors été adaptés.

Dans le même temps, nos contacts avec M.Letellier, chercheur au CORIA sur le chaos, nous ont éclaircis et M.Letellier nous a alors guidés dans l'étude de notre système. Le projet arrivant à son terme, il fallait donc analyser et rédiger les observations et résultats obtenus.

Concernant l'analyse bibliographique, Amal et Mathilde ont résumé leurs connaissances acquises sur le chaos en éléments essentiels pour la compréhension du problème, même si chacun avait lu des ouvrages sur la théorie du chaos afin d'en maîtriser les principaux aspects. Chacun a ensuite pris part à la rédaction du projet dans l'analyse du mouvement. Amal, Selima et Myriam ont élaboré les différentes analyses spectrales tandis que Martin et Mathilde analysaient l'évolution du pendule dans différents espaces. La rédaction du projet achevée, il a fallu choisir les éléments essentiels pour une présentation à la fois succincte et complète de notre étude.

Chapitre 2

Travail réalisé et résultats

2.1 Modélisation du problème à l'aide d'équations différentielles

2.1.1 Mise en équations du problème

Notre étude porte sur l'analyse du mouvement d'une masse attachée au bout d'un ressort : il s'agit d'un pendule élastique. Pour la réaliser, nous avons posé le problème d'un point de vue mécanique.

Point de vue mécanique

Les premières équations posées permettaient de résoudre un problème à une dimension : les oscillations étaient alors uniquement selon l'axe vertical \vec{e}_z . Ensuite, le problème a été posé en coordonnées polaires, en prenant en compte trois forces extérieures : le poids \vec{P} , la tension \vec{T} exercée par le ressort, les frottements fluides \vec{f} à faible vitesse et en utilisant la relation fondamentale de la dynamique valable dans tout référentiel galiléen. Ici le repère $(O; \vec{e}_x; \vec{e}_y; \vec{e}_z)$ est lié au référentiel terrestre, qui peut être considéré comme galiléen sur le temps de l'expérience.

$$\sum \vec{F}_{extérieures} = m \vec{a}.$$

L'inventaire détaillé des différentes forces s'appliquant sur la masse a tout d'abord été fait. On obtient alors :

- le poids \vec{P} avec $\vec{P} = -m \vec{g}$
- la tension \vec{T} exercée par le ressort : $\vec{T} = -k(r - l_0)\vec{U}_r$
- les frottements fluides \vec{f} pour les petites vitesses : $\vec{f} = -\lambda \vec{v}$

Où

- g est la pesanteur terrestre $g = 9,81 \text{ m} \cdot \text{sec}^{-2}$
- m la masse suspendue
- r la longueur du ressort
- l_0 la longueur du ressort à vide
- k sa constante de raideur
- λ le coefficient de frottements fluides $\lambda > 0$
- \vec{v} le vecteur vitesse de la masse

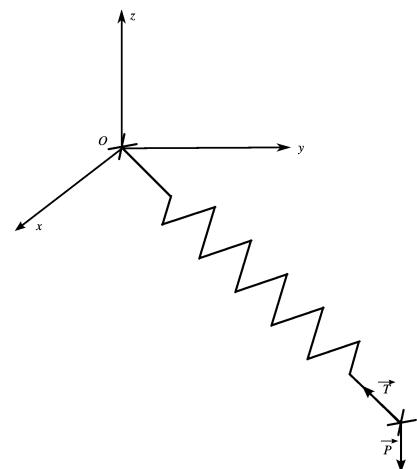


FIG. 2.1: Ressort

Mise en équations

En coordonnées sphériques, l'accélération \vec{a} d'un point matériel avec la masse m , a pour expression, d'après l'ouvrage [3] :

$$\vec{a} = \begin{cases} \ddot{r} - r\dot{\theta}^2 - r\dot{\varphi}^2 \sin^2 \theta & \vec{U}_r \\ 2\dot{r}\dot{\theta} + r\ddot{\theta} - r\dot{\varphi}^2 \sin \theta \cos \theta & \vec{U}_\theta \\ 2\dot{r}\dot{\varphi} \sin \theta + 2r\dot{\theta}\dot{\varphi} \cos \theta + r\ddot{\varphi} \sin \theta & \vec{U}_\varphi \end{cases}$$

Où $(\vec{U}_r; \vec{U}_\theta; \vec{U}_\varphi)$ sont définis comme l'indique le schéma suivant :

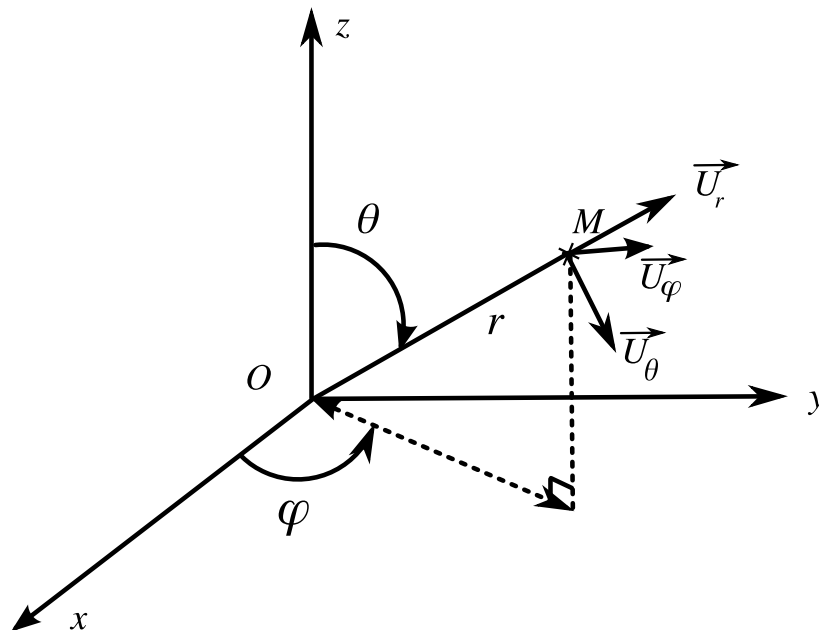


FIG. 2.2: Repère Sphérique

Les différentes forces dans le repère sphérique $(\vec{U}_r; \vec{U}_\theta; \vec{U}_\varphi)$ s'expriment de la façon suivante :

$$\vec{P} = -mg \begin{cases} \cos \theta & \vec{U}_r \\ -\sin \theta & \vec{U}_\theta \\ 0 & \vec{U}_\varphi \end{cases} \quad \vec{T} = -k(r - l_0) \begin{cases} 1 & \vec{U}_r \\ 0 & \vec{U}_\theta \\ 0 & \vec{U}_\varphi \end{cases} \quad \vec{f} = -\lambda \begin{cases} \dot{r} & \vec{U}_r \\ r\dot{\theta} & \vec{U}_\theta \\ r\dot{\varphi} \sin \theta & \vec{U}_\varphi \end{cases}$$

Ceci permet d'établir le système à trois équations différentielles totales du second ordre suivant, en projetant l'égalité

$$m\vec{a} = \vec{P} + \vec{f} + \vec{T}$$

sur chacun des trois vecteurs du repère sphérique :

$$\begin{cases} -mg \cos \theta - k(r - l_0) - \lambda\dot{r} = m(\ddot{r} - r\dot{\theta}^2 - r\dot{\varphi}^2 \sin^2 \theta) & \vec{U}_r \\ mg \sin \theta - \lambda r\dot{\theta} = m(2\dot{r}\dot{\theta} + r\ddot{\theta} - r\dot{\varphi}^2 \sin \theta \cos \theta) & \vec{U}_\theta \\ -\lambda r\dot{\varphi} \sin \theta = m(2\dot{r}\dot{\varphi} \sin \theta + 2r\dot{\theta}\dot{\varphi} \cos \theta + r\ddot{\varphi} \sin \theta) & \vec{U}_\varphi \end{cases}$$

$$\Leftrightarrow \begin{cases} -g \cos \theta - \frac{k}{m}(r - l_0) - \frac{\lambda}{m} \dot{r} = \ddot{r} - r\dot{\theta}^2 - r\dot{\varphi}^2 \sin^2 \theta & \vec{U}_r \\ \frac{g}{r} \sin \theta - \frac{\lambda}{m} \dot{\theta} - 2\frac{\dot{r}\dot{\theta}}{r} = \ddot{\theta} - \dot{\varphi}^2 \sin \theta \cos \theta & \vec{U}_\theta \\ -\lambda r \dot{\varphi} \sin \theta - 2\dot{r}\dot{\varphi} \sin \theta - 2r\dot{\theta}\dot{\varphi} \cos \theta = r\ddot{\varphi} \sin \theta & \vec{U}_\varphi \end{cases}$$

$$\Leftrightarrow \begin{cases} \ddot{r} = r\dot{\theta}^2 + r\dot{\varphi}^2 \sin^2 \theta - g \cos \theta - \frac{k}{m}(r - l_0) - \frac{\lambda}{m} \dot{r} & \vec{U}_r \\ \ddot{\theta} = \frac{g}{r} \sin \theta - \frac{\lambda}{m} \dot{\theta} - 2\frac{\dot{r}\dot{\theta}}{r} + \dot{\varphi}^2 \sin \theta \cos \theta & \vec{U}_\theta \\ \ddot{\varphi} = -2\frac{\dot{r}\dot{\varphi}}{r} - 2\frac{\dot{\theta}\dot{\varphi}}{\tan \theta} - \frac{\lambda}{m} \dot{\varphi} & \vec{U}_\varphi \end{cases}$$

Les outils mathématiques dont nous disposons actuellement ne permettent pas de résoudre ce système analytiquement. Toutefois, afin d'avoir des résultats, des méthodes de résolution numérique peuvent être utilisées, comme par exemple la méthode du Runge Kutta 4.

2.1.2 Conditions d'équilibre

Dans notre recherche de trajectoires chaotiques nous avons commencé par essayer de trouver les conditions initiales nous permettant d'obtenir une trajectoire stable, ici un cercle. Cette trajectoire est caractérisée par une vitesse angulaire $\dot{\varphi}$, un angle θ et une élongation constants. De cette façon, nous pouvons voir quels paramètres influencent cette trajectoire en observant les effets d'une légère modification de ceux-ci. Dans le cas présent, le problème se pose de la façon suivante :

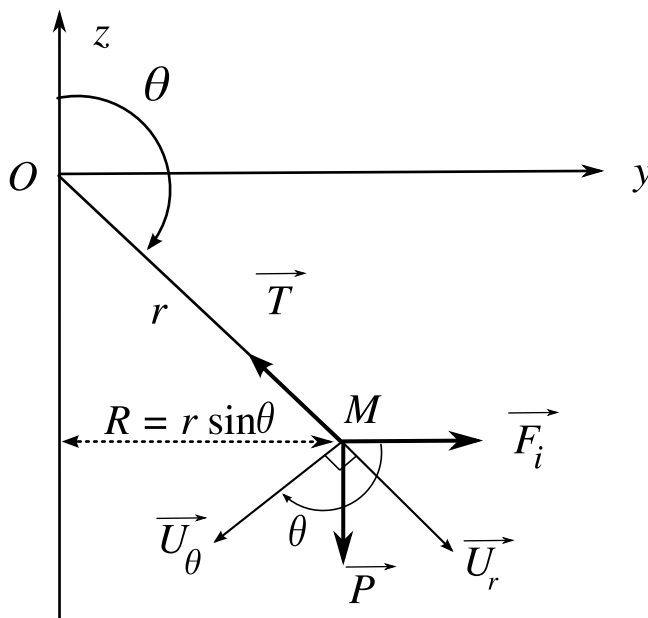


FIG. 2.3: Schéma de l'équilibre

Avec $\vec{P} = mg \cos(\theta + \pi) \vec{U}_r + mg \cos\left(\frac{\pi}{2} + \theta\right) \vec{U}_\theta$ et $\vec{T} = -k(r - l_0) \vec{U}_r$

$$\Leftrightarrow \vec{P} = -mg \cos \theta \vec{U}_r - mg \sin \theta \vec{U}_\theta$$

On prend en compte ici les forces d'inerties, qui s'apparentent à l'accélération radiale centrifuge de la masse :

$$\vec{F}_i = mR\dot{\varphi}^2 \left[\cos\left(\theta + \frac{\pi}{2}\right) \vec{U}_r + \cos(\theta) \vec{U}_\theta \right] \text{ où } R = r \sin \theta$$

$$\iff \vec{F}_i = mR\dot{\varphi}^2 [-\sin \theta \vec{U}_r + \cos \theta \vec{U}_\theta]$$

La trajectoire fermée circulaire périodique correspond en effet à une accélération de la masse se décomposant uniquement selon \vec{e}_y et dont la norme est constante. Le système est conservatif, le mouvement ne s'arrête pas, il n'y a donc pas de frottements. En projetant l'égalité $\vec{F}_i + \vec{P} + \vec{T} = 0$ sur \vec{U}_r puis sur \vec{U}_θ , on obtient le système suivant :

$$\begin{cases} -mg \cos \theta - k(r - l_0) - R \sin(\theta) m \dot{\varphi}^2 = 0 \\ -mg \sin \theta + mR\dot{\varphi}^2 \cos \theta = 0 \end{cases}$$

En remplaçant R par $r \sin \theta$, ce système équivaut à :

$$\begin{cases} g \cos \theta + r \sin^2(\theta) m \dot{\varphi}^2 = \frac{k}{m}(r_0 - l) \\ [-g + r \dot{\varphi}^2 \cos \theta] \sin \theta = 0 \Rightarrow r \dot{\varphi}^2 \cos \theta - g = 0 \Rightarrow \dot{\varphi} = \sqrt{\frac{g}{r |\cos \theta|}} \end{cases}$$

On peut considérer ici que $\frac{g}{r |\cos \theta|}$ existe et est positif. En effet, $\theta \neq \frac{\pi}{2}[\pi]$ car sinon le problème serait différent de celui abordé ici et sa mise en équation en serait modifiée.

Le système précédent conduit à :

$$\begin{cases} \cos \theta = \frac{mg}{k(l_0 - r)} \\ \dot{\varphi} = \sqrt{\frac{g}{r |\cos \theta|}} \end{cases}$$

Notons là aussi que $(r - l_0) \neq 0$ puisque le ressort doit être en légère élongation pour obtenir cette trajectoire stable et périodique. On obtient bien des valeurs appartenant à $[-1; 1]$ pour $\cos \theta$, lorsque les valeurs choisies ont une réalité et vraisemblance physiques. Ensuite, comme nous avons deux équations et six inconnues, il nous faut déterminer quatre d'entre elles. Nous choisissons arbitrairement : $\theta = \frac{5\pi}{4}$ et $r = 0,15\text{m}$, ce qui permet de calculer $\dot{\varphi}$:

$$\dot{\varphi} = \sqrt{\frac{9,81}{0,15 \cos\left(\frac{5\pi}{4}\right)}} \Rightarrow \dot{\varphi} = 9,617\ 149\ 629 \text{ rad. sec}^{-1}$$

Puis en posant $m = 0,162\text{ g}$ et $l_0 = 0,12\text{ m}$ on obtient :

$$k = \frac{mg}{\cos \theta (l_0 - r)} \Rightarrow k = \frac{0,162 \times 9,81}{(0,12 - 0,15) \cos \frac{5\pi}{4}} \Rightarrow k = 74,916\ 549\ 25 \text{ kg. sec}^{-2}.$$

2.2 Méthode de résolution des équations différentielles

2.2.1 Principe du Runge Kutta 4

Pour résoudre le couple de 3 équations différentielles du second ordre, nous avons décidé de programmer la méthode de Runge-Kutta 4. Nous allons d'abord expliquer le principe de résolution d'une équation différentielle du premier ordre, pour ensuite généraliser notre méthode aux 3 équations différentielles couplées du second ordre .

Equation différentielle du premier ordre

Tout d'abord, nous allons expliquer le fonctionnement de la méthode de Runge-Kutta à l'ordre 4 pour une équation différentielle simple de la forme suivante : $\dot{x}(t) = f(t, x(t))$.

Le principe de la méthode de Runge-Kutta 4 est basée sur une itération successive. C'est-à-dire que l'on se base sur le principe suivant pour calculer tous les points successifs.

$$\begin{cases} x_0 = \beta \\ x_{i+1} = x_i + h_i \times \alpha_i \end{cases}$$

En effet d'après le théorème de Cauchy, il existe une unique solution si l'on fixe suffisamment de conditions initiales, ici une seule suffit : $x_0 = \beta$. Pour calculer le point x_{i+1} chercher à trouver un taux d'accroissement numérique α_i qui multiplié au pas de l'itération h_i et ajouté à la valeur précédente x_i , donnera une approximation cohérente. En partant de la condition initiale, on peut déterminer la solution approchée numériquement.

Il existe plusieurs façon de déterminer les coefficients α_i , et ici nous avons décidé de choisir celle de Runge-Kutta 4. Ce coefficient est une pondération de différentes valeurs de la dérivée prise dans l'intervalle $[t; t + h]$. Voici la forme générale :

$$\alpha_i = \frac{\alpha_{i,1} + 2\alpha_{i,2} + 2\alpha_{i,3} + \alpha_{i,4}}{6}$$

Voici donc pour l'exemple choisi les 4 valeurs des α :

$$\begin{cases} \alpha_{i,1} = f(t; y_i) \\ \alpha_{i,2} = f\left(t + \frac{h}{2}; y_i + \frac{h}{2} \times \alpha_{i,1}\right) \\ \alpha_{i,3} = f\left(t + \frac{h}{2}; y_i + \frac{h}{2} \times \alpha_{i,2}\right) \\ \alpha_{i,4} = f(t + h; y_i + h \times \alpha_{i,3}) \end{cases}$$

Généralisation aux 3 équations différentielles couplées

Nous allons maintenant généraliser le système aux 3 équations différentielles couplées suivantes :

$$\begin{cases} \ddot{r} = d(t; r(t); \dot{r}(t); \theta(t); \dot{\theta}(t); \varphi(t); \dot{\varphi}(t)) = r\dot{\theta}^2 + r\dot{\varphi}^2(\sin\theta)^2 - g\cos\theta - \frac{k}{m}(r - l_0) - \frac{\lambda}{m}\dot{r} \\ \ddot{\theta} = e(t; r(t); \dot{r}(t); \theta(t); \dot{\theta}(t); \varphi(t); \dot{\varphi}(t)) = \frac{-2m\dot{r}\dot{\theta} + m\dot{\varphi}^2 \sin\theta \cos\theta + mg\sin\theta - \lambda r\dot{\theta}}{mr} \\ \ddot{\varphi} = f(t; r(t); \dot{r}(t); \theta(t); \dot{\theta}(t); \varphi(t); \dot{\varphi}(t)) = \frac{-2m\dot{r}\dot{\varphi} - \frac{2mr\dot{\theta}\dot{\varphi}}{\tan\theta} - \lambda r\dot{\varphi}}{mr} \end{cases}$$

Soient les fonctions a, b et c , les dérivées des fonctions d, e et f . Aussi, on notera respectivement $\alpha, \beta, \gamma, \delta, \varepsilon$ et ζ les taux respectifs des 6 fonctions. Pour alléger la notation, nous utilisons les $r, \dot{r}, \theta, \dot{\theta}, \varphi, \dot{\varphi}$ d'indice $i-1$ sans pour autant les inclure.

$$\left\{ \begin{array}{l} \alpha_{i,1} = d(t, r, \dot{r}, \theta, \dot{\theta}, \varphi, \dot{\varphi}) \\ \beta_{i,1} = e(t, r, \dot{r}, \theta, \dot{\theta}, \varphi, \dot{\varphi}) \\ \gamma_{i,1} = f(t, r, \dot{r}, \theta, \dot{\theta}, \varphi, \dot{\varphi}) \\ \delta_{i,1} = \dot{r} \\ \varepsilon_{i,1} = \dot{\theta} \\ \zeta_{i,1} = \dot{\varphi} \end{array} \right.$$

$$\left\{ \begin{array}{l} \alpha_{i,2} = d\left(t + \frac{h}{2}, r + \frac{h}{2}\delta_{i,1}, \dot{r} + \frac{h}{2}\alpha_{i,1}, \theta + \frac{h}{2}\varepsilon_{i,1}, \dot{\theta} + \frac{h}{2}\beta_{i,1}, \varphi + \frac{h}{2}\zeta_{i,1}, \dot{\varphi} + \frac{h}{2}\gamma_{i,1}\right) \\ \beta_{i,2} = e\left(t + \frac{h}{2}, r + \frac{h}{2}\delta_{i,1}, \dot{r} + \frac{h}{2}\alpha_{i,1}, \theta + \frac{h}{2}\varepsilon_{i,1}, \dot{\theta} + \frac{h}{2}\beta_{i,1}, \varphi + \frac{h}{2}\zeta_{i,1}, \dot{\varphi} + \frac{h}{2}\gamma_{i,1}\right) \\ \gamma_{i,2} = f\left(t + \frac{h}{2}, r + \frac{h}{2}\delta_{i,1}, \dot{r} + \frac{h}{2}\alpha_{i,1}, \theta + \frac{h}{2}\varepsilon_{i,1}, \dot{\theta} + \frac{h}{2}\beta_{i,1}, \varphi + \frac{h}{2}\zeta_{i,1}, \dot{\varphi} + \frac{h}{2}\gamma_{i,1}\right) \\ \delta_{i,2} = \dot{r} + \frac{h}{2}\alpha_{i,1} \\ \varepsilon_{i,2} = \dot{\theta} + \frac{h}{2}\beta_{i,1} \\ \zeta_{i,2} = \dot{\varphi} + \frac{h}{2}\gamma_{i,1} \end{array} \right.$$

$$\left\{ \begin{array}{l} \alpha_{i,3} = d\left(t + \frac{h}{2}, r + \frac{h}{2}\delta_{i,2}, \dot{r} + \frac{h}{2}\alpha_{i,2}, \theta + \frac{h}{2}\varepsilon_{i,2}, \dot{\theta} + \frac{h}{2}\beta_{i,2}, \varphi + \frac{h}{2}\zeta_{i,2}, \dot{\varphi} + \frac{h}{2}\gamma_{i,2}\right) \\ \beta_{i,3} = e\left(t + \frac{h}{2}, r + \frac{h}{2}\delta_{i,2}, \dot{r} + \frac{h}{2}\alpha_{i,2}, \theta + \frac{h}{2}\varepsilon_{i,2}, \dot{\theta} + \frac{h}{2}\beta_{i,2}, \varphi + \frac{h}{2}\zeta_{i,2}, \dot{\varphi} + \frac{h}{2}\gamma_{i,2}\right) \\ \gamma_{i,3} = f\left(t + \frac{h}{2}, r + \frac{h}{2}\delta_{i,2}, \dot{r} + \frac{h}{2}\alpha_{i,2}, \theta + \frac{h}{2}\varepsilon_{i,2}, \dot{\theta} + \frac{h}{2}\beta_{i,2}, \varphi + \frac{h}{2}\zeta_{i,2}, \dot{\varphi} + \frac{h}{2}\gamma_{i,2}\right) \\ \delta_{i,3} = \dot{r} + \frac{h}{2}\alpha_{i,2} \\ \varepsilon_{i,3} = \dot{\theta} + \frac{h}{2}\beta_{i,2} \\ \zeta_{i,3} = \dot{\varphi} + \frac{h}{2}\gamma_{i,2} \end{array} \right.$$

$$\left\{ \begin{array}{l} \alpha_{i,4} = d\left(t + h, r + h\delta_{i,3}, \dot{r} + h\alpha_{i,3}, \theta + h\varepsilon_{i,3}, \dot{\theta} + h\beta_{i,3}, \varphi + h\zeta_{i,3}, \dot{\varphi} + h\gamma_{i,3}\right) \\ \beta_{i,4} = e\left(t + h, r + h\delta_{i,3}, \dot{r} + h\alpha_{i,3}, \theta + h\varepsilon_{i,3}, \dot{\theta} + h\beta_{i,3}, \varphi + h\zeta_{i,3}, \dot{\varphi} + h\gamma_{i,3}\right) \\ \gamma_{i,4} = f\left(t + h, r + h\delta_{i,3}, \dot{r} + h\alpha_{i,3}, \theta + h\varepsilon_{i,3}, \dot{\theta} + h\beta_{i,3}, \varphi + h\zeta_{i,3}, \dot{\varphi} + h\gamma_{i,3}\right) \\ \delta_{i,4} = \dot{r} + h\alpha_{i,3} \\ \varepsilon_{i,4} = \dot{\theta} + h\beta_{i,3} \\ \zeta_{i,4} = \dot{\varphi} + h\gamma_{i,3} \end{array} \right.$$

Maintenant que nous avons les 4 taux pour les 6 fonctions, nous pouvons calculer les 6 valeurs suivantes qui serviront pour la boucle suivante. On calcule ici les valeurs $r, \dot{r}, \theta, \dot{\theta}, \varphi, \dot{\varphi}$ d'indice i .

$$\left\{ \begin{array}{l} \dot{r}_i = \dot{r}_{i-1} + h \frac{\alpha_{i,1} + 2\alpha_{i,2} + 2\alpha_{i,3} + \alpha_{i,4}}{6} \\ \dot{\theta}_i = \dot{\theta}_{i-1} + h \frac{\beta_{i,1} + 2\beta_{i,2} + 2\beta_{i,3} + \beta_{i,4}}{6} \\ \dot{\varphi}_i = \dot{\varphi}_{i-1} + h \frac{\gamma_{i,1} + 2\gamma_{i,2} + 2\gamma_{i,3} + \gamma_{i,4}}{6} \\ r_i = r_{i-1} + h \frac{\delta_{i,1} + 2\delta_{i,2} + 2\delta_{i,3} + \delta_{i,4}}{6} \\ \theta_i = \theta_{i-1} + h \frac{\varepsilon_{i,1} + 2\varepsilon_{i,2} + 2\varepsilon_{i,3} + \varepsilon_{i,4}}{6} \\ \varphi_i = \varphi_{i-1} + h \frac{\zeta_{i,1} + 2\zeta_{i,2} + 2\zeta_{i,3} + \zeta_{i,4}}{6} \end{array} \right.$$

C'est ainsi que nous avons programmé la résolution des 3 équations différentielles couplées, à l'aide du langage Java, et c'est ce que nous détaillerons dans la section suivante.

2.2.2 Code en Java

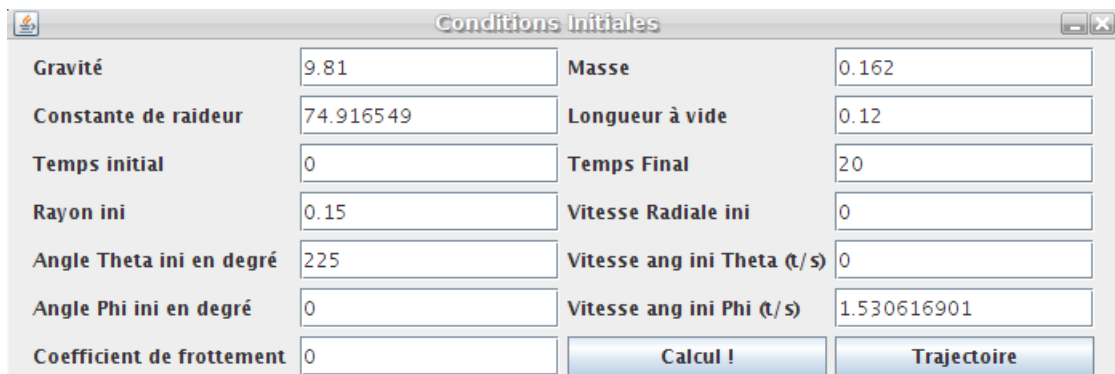
Mise en place de la programmation

Tout d'abord, nous avons réalisé un programme en C pour la résolution de ces équations différentielles. Cependant, nous l'avons très peu utilisé : il nous a juste servi de base de référence pour la résolution des équations différentielles couplées. Nous nous sommes appuyés sur ce programme pour passer au langage Java.

En effet, nous avons préféré programmer en Java, pour l'utilisation des bibliothèques graphiques et pour la représentation de notre pendule élastique. C'est ce que nous expliquerons dans la section suivante (cf. 2.3) : la représentation graphique des résultats.

Organisation du code

Voici les différentes classes que nous avons créées. Tout d'abord la classe principale JPomme (cf. Annexe C.1), qui nous permet d'appeler la classe maîtresse de notre programme : la classe JChoix (cf. Annexe C.2). Cette classe met en place l'interface graphique qui nous permet de choisir les conditions initiales à donner au pendule élastique, dont voici un aperçu, ci-dessous.



Conditions Initiales			
Gravité	9.81	Masse	0.162
Constante de raideur	74.916549	Longueur à vide	0.12
Temps initial	0	Temps Final	20
Rayon ini	0.15	Vitesse Radiale ini	0
Angle Theta ini en degré	225	Vitesse ang ini Theta (t/s)	0
Angle Phi ini en degré	0	Vitesse ang ini Phi (t/s)	1.530616901
Coefficient de frottement	0	Calcul !	Trajectoire

FIG. 2.4: Fenêtre Java du choix des conditions initiales

La classe JChoix permet de faire le lien entre toutes les autres classes que nous verrons par la suite. Ainsi c'est à travers elle que l'on décide de calculer, représenter ou analyser des mouvements particuliers.

Classe de résolution des équations différentielles

Ainsi, pour résoudre les équations différentielles, nous utilisons la classe suivante : Jresort3D (cf. Annexe C.3). Le constructeur permet de récupérer les conditions initiales sélectionnées dans la classe JChoix. Ensuite, à travers 3 fonctions, nous définissons les 3 équations différentielles. Ces 3 fonctions sont appelées dans la méthode calcul qui utilise la méthode de Runge Kutta 4 pour résoudre le système. Les résultats, pour chaque instant, sont enregistrés dans un tableau virtuel, et à travers une autre méthode, nous pouvons écrire une partie de ce tableau dans un fichier.

Aussi, dans cette classe, on trouve diverses méthodes qui permettent de récupérer les données dans le tableau virtuel, ce qui est nécessaire pour l'analyse. On trouve, par exemple, une méthode qui retourne le rayon maximal, ce qui est utile pour dimensionner la fenêtre d'affichage.

2.3 Illustration des résultats obtenus

2.3.1 Principe mathématiques : projection

Pour visualiser nos résultats, nous avons utilisé de nombreuses bibliothèques Java, mais nous avons eu aussi besoin de calculer la taille de l'image générale pour avoir l'ensemble de la trace du pendule. C'est pourquoi dans la classe `Jressort3D` on trouve, à chaque itération, le calcul du rayon maximal, pour obtenir à la fin ce rayon maximal qui permet de dimensionner correctement le tracé dans la fenêtre.

Aussi, nous avons utilisé des méthodes de projections pour pouvoir tracer une représentation cohérente du pendule élastique. Nous avons les coordonnées dans un repère sphérique, nous les transformons pour se placer dans un repère cartésien. Ensuite, nous avons défini des angles de rotations horizontaux et verticaux, de telle sorte que l'on puisse se positionner de n'importe quel endroit pour visionner la trace du pendule élastique. Sachant que α et β représentent respectivement l'angle de rotation autour de l'axe $(O; \vec{z})$ et celui autour de l'axe $(O; \vec{y})$, voici les formules de projection dans la fenêtre que l'on repère par X et Y les coordonnées de la fenêtre ayant pour *centre* la moitié de la valeur du rayon maximal.

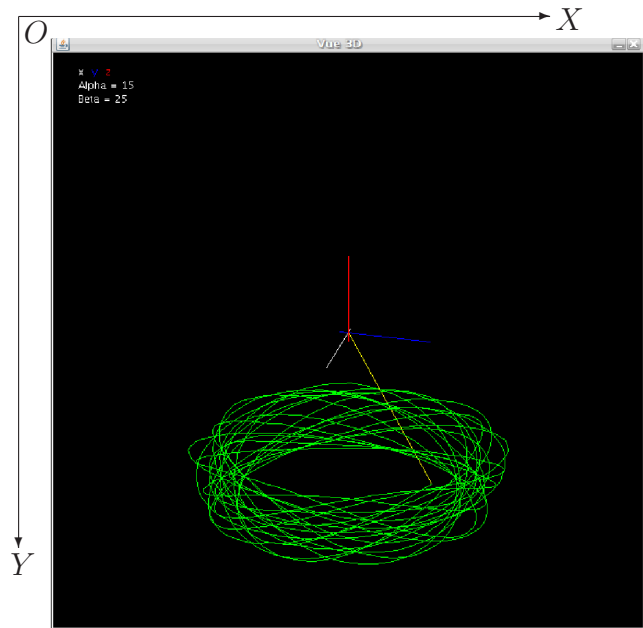


FIG. 2.5: Représentation de la trajectoire physique

$$\begin{cases} X = \text{centre} - x \sin \alpha + y \cos \alpha \\ Y = \text{hauteur de la fenetre} - (\text{centre} + x \cos \alpha \sin \beta + y \sin \alpha \sin \beta + z \cos \beta) \end{cases}$$

2.3.2 Code en Java

Nous utilisons seulement une classe `Jaffichage` (cf. Annexe C.4), pour faire le tracé du pendule en temps réel et en conserver la trace. Dans cette classe, nous utilisons les bibliothèques disponibles pour représenter le pendule élastique.

Aussi, pour conserver la trace de tous les points, nous traçons une ligne entre chaque point placé. Ensuite les valeurs sont gardées en mémoire dans un tableau dynamique pour pouvoir à tout moment recalculer la position selon une nouvelle orientation.

2.3.3 Code de Scilab

Pour procéder à une analyse fréquentielle de l'évolution du rayon en fonction du temps, nous utilisons le logiciel `Scilab`. A partir de notre fichier `"resultat.dat"` contenant pour chaque variable du tableau, sa valeur à un temps donné. Nous avons donc dans les colonnes 1 et 4, le temps et la valeur du rayon. Le programme (cf. Annexe D) établi permet de tracer le spectre de Fourier. Nous analysons directement le fichier `"resultat.dat"` car la résolution numérique par le Runge-Kutta 4 du logiciel donne des résultats aberrants.

2.4 Analyse bibliographique des études des mouvements chaotiques

2.4.1 Notion de système dynamique et de chaos

Un système dynamique est un système qui évolue au cours du temps en termes de déplacement ou mouvement. Cette évolution temporelle dépend de l'état passé ou présent. On dit que l'évolution est causale. De plus, le système dynamique évolue de façon déterministe, où pour une condition initiale correspond un unique état futur possible.

On distingue trois principaux modèles de systèmes dynamiques : le modèle déterministe, stochastique (aléatoire) et chaotique. Les systèmes stochastiques ou aléatoires évoluent au hasard dans l'espace sans qu'aucune équation différentielle ne les régit directement. Les systèmes déterministes sont des systèmes régis par des lois mathématiques bien connues, on peut donc prévoir exactement l'évolution de ces systèmes dans le temps.

Un système dynamique comme nous l'avons décrit ci-dessus fait parfois preuve de comportements imprévisibles, voire aléatoires, alors que son comportement est régi par des équations différentielles connues. C'est cette imprédictibilité de l'évolution que l'on appelle chaos. Le comportement de tels systèmes est très complexe. En terme général, on dit d'un système qu'il est chaotique s'il est régi par des lois déterministes connues mais que son évolution échappe tout de même à toute prévision sur le long terme. L'origine de ce phénomène est la dépendance de ces systèmes aux conditions initiales.

La théorie du chaos consiste donc à décrire au mieux le comportement à long terme du système dynamique. Pour cela, l'étude ne se résume pas à déterminer des solutions précises aux équations qui régissent le mouvement, car pour certains systèmes, cela devient vite très complexe. Au contraire, l'étude doit permettre de répondre aux questions suivantes : "Le système convergera-t-il vers un état stationnaire à long terme, et dans ce cas, quels sont les états stationnaires possibles?" et "Le comportement à long terme du système dépend-il des conditions initiales?".

Ceci nous amène à une définition heuristique du chaos. En effet, d'après la définition heuristique, un système est dit chaotique si lors de sa représentation dans l'espace des phases (cf. 2.4.3) il présente simultanément une forte récurrence et une importante sensibilité aux conditions initiales. En effet, pour un système chaotique, une infime erreur dans la connaissance de l'état initial dans l'espace des phases s'accroît exponentiellement en $e^{\frac{t}{\tau}}$ où τ est un temps caractéristique appelé "horizon de Lyapounov". Plus précisément, pour $t < \tau$, l'erreur est minime tandis que pour $t > \tau$, la prédiction devient impossible alors que le théorème de Cauchy-Lipschitz qui postule "l'existence et l'unicité de la solution" est toujours valable.

2.4.2 Un peu d'histoire

Nous pourrions croire que la notion de "chaos" est née avec les travaux d'Edward Lorenz dans les années 1960. Mais les premières études sur la stabilité des systèmes remontent à la fin du XIX^{ème} siècle avec les travaux d'Henri Poincaré concernant le problème à N corps. En fait, le point de départ de la dynamique des systèmes est la question de stabilité du système solaire.

Dès les années 1880, Henri Poincaré ébranla cette théorie qui faisait de l'univers un système géant muni de lois simples qu'il suffisait de trouver. Aussi, dans les années 1960-1970, l'informatique a popularisé la théorie du chaos.

C'est en 1963 que le météorologue Edward Lorenz alors professeur au MIT mit en évidence le caractère chaotique de la météo. En cherchant et en résolvant un modèle très simplifié du système d'équations qui régissent le couplage de l'atmosphère avec l'océan, E. Lorenz observa alors, par pur hasard, qu'une modification infinitésimale des données initiales (de l'ordre du millième), entraînait des résultats bien différents.

E. Lorenz venait de mettre en évidence la sensibilité aux conditions initiales qu'il illustra dans une métaphore devenue célèbre : "Le simple battement d'ailes d'un papillon au Brésil pourrait déclencher une tornade au Texas". Ce n'est qu'en 1975 que le mathématicien donna à cette théorie le nom de chaos.

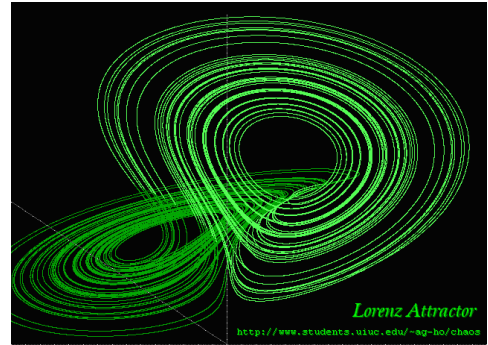


FIG. 2.6: Papillon de Lorenz [14]

Par la suite, E. Lorenz souhaita représenter graphiquement ses solutions, c'est alors qu'il découvrit ces formes appelées attracteurs. En traçant la courbe d'évolution du système météorologique avec deux séries de valeurs initiales très proches, il observa que les deux courbes étaient très proches au départ mais divergeaient de plus en plus. Cependant, le plus étonnant était que les deux courbes étaient plus ou moins identiques dans leur forme. Elles ressemblaient à des ailes de papillon. En renouvelant plusieurs fois l'expérience, E. Lorenz obtenait à chaque fois le même résultat.

Plus tard, le physicien David Ruelle qualifia ces attracteurs "d'étranges". Comme on peut l'observer sur la figure 2.6, les trajectoires ne se coupent jamais, et semblent pourtant, évoluer au hasard en formant des boucles, figures reconnaissables. Bien plus tard, Mandelbrot retrouva ces attracteurs étranges en étudiant la géométrie fractale.

2.4.3 Notion d'espace des phases

On peut dire que l'espace des phases est l'ensemble des grandeurs nécessaires à la connaissance de l'évolution temporelle du système considéré. L'espace des phases sert à représenter l'évolution des systèmes dynamiques : les axes de coordonnées de cet espace correspondent aux différents degrés de liberté caractérisant les mouvements du système, d'après l'ouvrage [5]. Prenons l'exemple d'un pendule simple : pour déterminer ce système, il faut connaître son élongation mais aussi sa vitesse.

Plan de Poincaré

Dans cet espace des phases, afin d'étudier le comportement du système, il est utile de définir un plan ou section de Poincaré. Il se définit de la manière suivante : "La section de Poincaré est une coupe par un plan dans l'espace de phases". Il permet une meilleure approche du comportement du système en retirant une dimension au problème. Chaque point de ce plan correspond à un point de l'espace des phases.

Diagramme de bifurcation

Le diagramme de bifurcation permet de mieux visualiser l'évolution d'un système vers le chaos par doublement de période. Il s'agit d'une représentation des valeurs possibles d'une variable en fonction de la variation infinitésimale de sa valeur précédente.

$$X_n + 1 = f(\Delta X_n)$$

A ce stade, il faut noter que les diagrammes de bifurcation tels que nous les avons définis ne sont utilisés que dans le cas de systèmes physiques dissipatifs. Dans le cas de systèmes conservatifs, leur obtention et étude sont très complexes (cf. [5] et [9]).

2.4.4 Notion d'attracteur

À chaque instant, l'état du phénomène est représenté par un point dans l'espace des phases. L'évolution du système est décrite par la trajectoire de ce point. Un attracteur est un ensemble compact de l'espace des phases vers lequel toutes les trajectoires environnantes convergent. Le bassin d'attraction est alors l'ensemble des points dont les trajectoires convergent vers l'attracteur.

On dénombre quatre sortes d'attracteurs :

Point fixe Il s'agit du plus simple. Il est tout simplement la caractéristique de l'état stationnaire atteint par le système. Pour illustrer simplement ce type d'attracteur, on peut prendre l'exemple du pendule soumis à des frottements. La représentation dans l'espace des phases du mouvement sera une spirale convergeant vers le point d'angle nul et de vitesse nulle. Le point fixe attire vers lui les trajectoires.

Cercle limite Il est la caractéristique d'un système lorsque celui-ci atteint un état répétitif. Les trajectoires dans l'espace des phases $(\dot{\theta}, \theta)$ convergent vers un cercle fermé. C'est le cas d'un cerceau que l'on fait tourner indéfiniment autour de la taille, ainsi que notre système étudié lorsque la masse décrit un cercle parfait autour de l'axe du pendule.

Tore Il caractérise les systèmes quasi-périodiques. La trajectoire de tels systèmes dans le plan de Poincaré semble se répéter mais ce n'est en fait qu'une illusion. En effet, la trajectoire se répète périodiquement mais décalée. Plusieurs fréquences indépendantes, dont on extrait un spectre de Fourier, entrent alors en jeu.

Attracteur étrange Comme son nom l'indique, ce type d'attracteur est le plus étrange. En effet, les systèmes chaotiques sont attirés par une figure géométrique de structure infiniment complexe sur laquelle ils semblent errer au hasard, mais sans jamais la quitter, ni repasser deux fois par le même point. Cette figure est appelée attracteur étrange. Il semble inclure à la fois des lois déterministes et des lois aléatoires, ce qui rend impossible toute prévision à long terme.

Par définition, l'attracteur étrange consiste en fait en une représentation du système chaotique dans un espace des phases défini. Cependant, on remarque la présence d'attracteurs dans de nombreux systèmes, qu'ils soient chaotiques ou non, conservatifs ou dissipatifs : seules leurs formes diffèrent. Aussi, parmi les attracteurs étranges, l'attracteur de Lorenz se démarque particulièrement. On peut l'appeler également "papillon de Lorenz" de par sa forme particulière (cf. 2.6). On dénombre aussi d'autres types d'attracteurs étranges bien connus comme l'attracteur de Hénon.

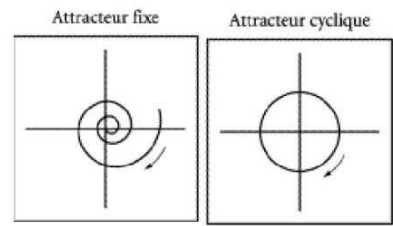


FIG. 2.7: Point fixe et Cercle limite [13]

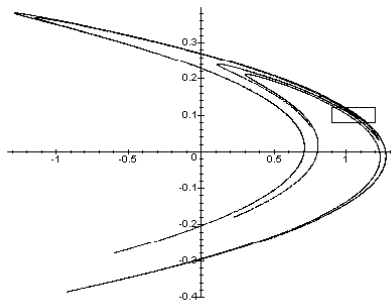


FIG. 2.8: Attracteur de Hénon [12]

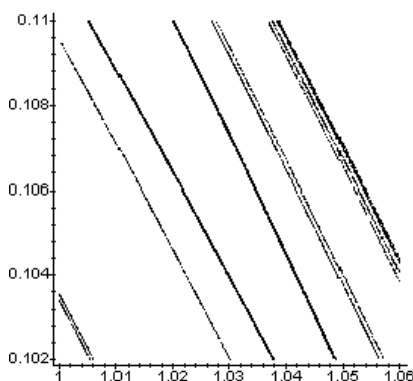


FIG. 2.9: Détail ($1 < x < 1,06$ et $0,102 < y < 0,11$) [12]

Les attracteurs étranges présentent de véritables propriétés. La plus étonnante est la structure feuilletée ou structure fractale. Une fractale est une figure géométrique qui consiste en un motif qui se répète à l'infini. Lorsque l'on fait un zoom sur une partie des fractales, puis un autre zoom, on retrouve toujours le même motif. Ces objets géométriques explorés par le mathématicien Benoît Mandelbrot révèlent une forme fragmentée qui, lorsqu'elle est subdivisée en parties, reproduit toujours la même forme, indépendamment de l'échelle.

Dans la représentation des systèmes chaotiques, les attracteurs étranges sont des courbes très complexes qui malgré leur caractère à-priori aléatoire, présentent un certain ordre. En effet, si l'on pratique un "zoom" avant ou arrière sur l'attracteur, on y retrouve la même forme, la même structure. À chaque échelle, il est semblable à lui-même. Des agrandissements successifs de l'attracteur nous montrent une structure feuilletée qui semble se répéter indéfiniment. L'attracteur étrange est donc indivisible en d'autres attracteurs. Ceci est bien visible dans l'agrandissement observable ci-contre.

De plus, cette même structure feuilletée ou fractale est observable dans un diagramme de bifurcation. À l'approche de chaque région chaotique, une succession de doublements de période s'effectue en progression géométrique tendant vers $\frac{1}{\delta F}$. (δF étant ici la constante de Feigenbaum et vaut environ 4,669 2).

2.5 Analyse du mouvement au sens de la dynamique des systèmes

Dans cette partie, nous limiterons notre étude au rayon (r) qui n'est qu'une seule des 3 composantes du système ($r; \theta; \varphi$). Nous n'étudierons que les conséquences d'une variation du rayon par rapport à sa valeur d'équilibre.

L'étude de la trajectoire dans l'espace physique, l'analyse de l'espace des phases combi-

née avec la section de Poincaré et enfin l'évolution du rayon en fonction du temps couplée avec le spectre de Fourier, nous permettront d'analyser le système. Toutes ces notions liées à la théorie du chaos sont expliquées dans l'analyse bibliographique (cf. 2.4).

2.5.1 Choix des points à étudier

Pour choisir les points à étudier, nous avons réalisé de nombreux tests à l'aide d'une nouvelle classe nommée JPoincare (cf. Annexe C.5). Cette dernière nous permet entre autre de sélectionner un espace des phases et de choisir un plan de Poincaré défini par son vecteur normal et un point lui appartenant. Pour la section de Poincaré, nous avons choisi d'étudier uniquement les points d'intersection avec le plan, mais selon un sens défini. Afin de récupérer les points d'intersection entre la trajectoire de l'espace des phases avec le plan de Poincaré, nous avons assimilé la trajectoire à une droite entre les points situés de part et d'autre de la section.

Pour faire notre analyse, et choisir les points intéressants à étudier, nous nous sommes basé sur le graphique suivant (cf. FIG. 2.10). Ce dernier a été, réalisé de la façon suivante. Tout d'abord nous avons modifié le programme de telle sorte qu'il puisse calculer l'ensemble des points du pendule pour un rayon donné, puis son espace des phases en $(r; \dot{r}; \ddot{r})$, et enfin réaliser une section de Poincaré de vecteur normal $(1; 1; 1)$ passant par le point $(0, 15; 0; 0)$, tout ceci avec un temps de calcul de 30 s avec un pas de 0, 1 ms, afin de ne pas trop surcharger la mémoire. Cette opération a été répétée pour un intervalle de $r_{initial} \in [0; 0, 15]$.

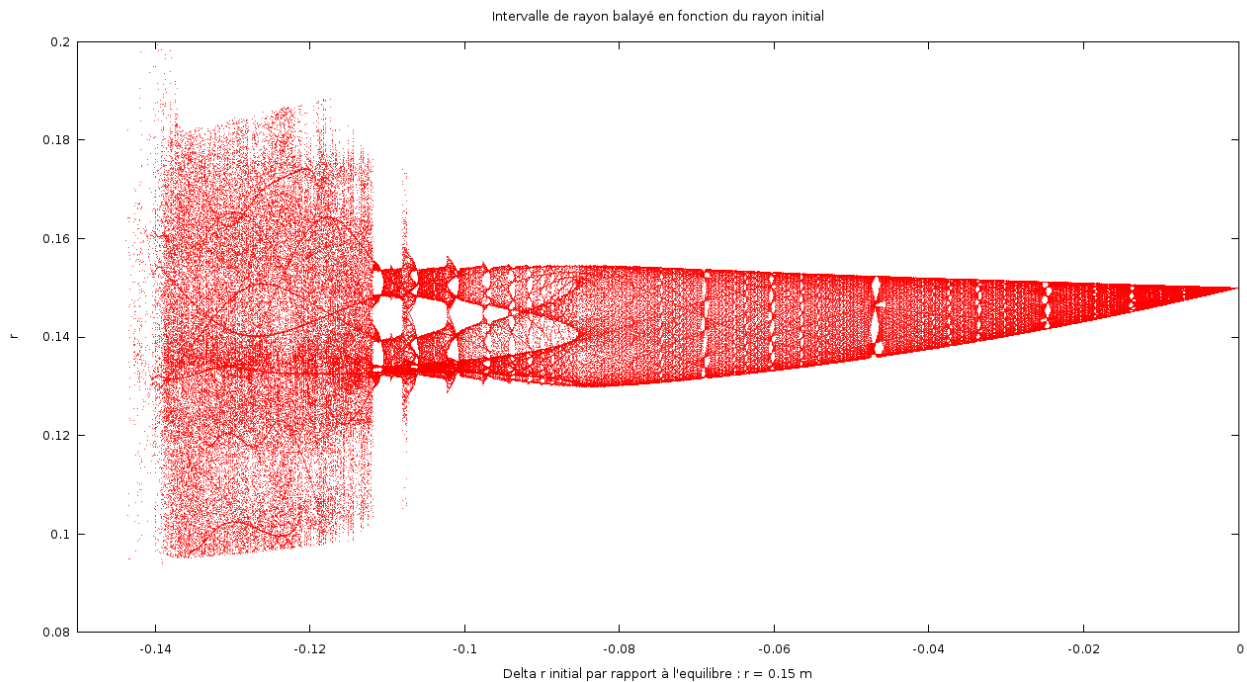


FIG. 2.10: Intervalle de rayon balayé, en fonction de $\Delta r_{initial}$

Au vu de ce diagramme, voici les points qu'il nous semble judicieux d'étudier afin de comprendre ce qu'est le chaos et ce qui le caractérise. Nous spécifions que toutes les longueurs évoquées sont exprimées en mètres. Nous avons débuté par la position d'équilibre : $r = 0,15$, puis nous avons choisi d'observer les effets d'une légère compression jusqu'à obtenir un mouvement quasi-périodique atteint en $r = 0,103\ 106\ 2$.

Ensuite, nous avons observé une bouffée chaotique pour $r = 0,042\ 2$, suivi d'une fenêtre stable, dans laquelle nous avons choisi d'étudier le cas $r = 0,039\ 15$. Enfin nous avons ter-

miné notre étude dans le domaine chaotique avec $r = 0,032$.

Cette façon de voir les choses, nous a permis de mettre en évidence le fait suivant : pour certaines conditions initiales, le rayon balayé est incroyablement grand, alors que pour une condition initiale très proche, l'intervalle balayé est faible. **Est-ce une caractérisation de l'aspect chaotique de notre pendule ?** C'est ce que nous allons essayer de comprendre avec les outils suivants.

2.5.2 Outils d'étude et intérêts

Pour étudier chacun de ces points, nous avons utilisé quatre outils, et paramétrer le système de la façon suivante, c'est-à-dire avec un calcul de 150 s, toujours avec un pas de 0,1 ms. Tout d'abord, la trajectoire physique décrite par le système permet de définir rapidement si le mouvement semble ordonné, s'il suit une certaine structure régulière, ou alors si il n'y a aucune cohérence dans la trajectoire.

Puis, dans l'espace des phases (r, \dot{r}, \ddot{r}) , l'étude de la trajectoire permet de préciser le domaine où le système étudié se situe. L'espace des phases choisi correspond uniquement à la variable r et ses dérivées, car ce sont les effets des variations du rayon initial que nous étudions.

Ensuite, nous avons choisi le plan de Poincaré avec $\dot{r} = 0$ pour la signification physique de la chose : cela nous permet de voir pour quelle valeur du rayon, la vitesse radial s'annule. La section de Poincaré présente, dans l'ensemble des cas étudiés, une forme circulaire ou un ensemble de points plus ou moins ordonnés.

Enfin, l'analyse de la valeur du rayon en fonction du temps permet d'obtenir un spectre de Fourier. Grâce à ce dernier, il est possible de mettre en relief l'existence de fréquences nettement définies dans les cas périodique et quasi-périodique, alors qu'elles sont davantage floues dans le domaine chaotique.

2.5.3 Point d'équilibre : $r = 0,15$

Comme nous l'avons vu précédemment dans la première partie concernant la position d'équilibre (cf. 2.1.2). Nous avons choisi des paramètres initiaux correspondant à l'équilibre. En voici l'analyse selon la méthodologie explicitée ci-dessus.

Trajectoire physique

Voici la trajectoire en 3D du pendule élastique à l'équilibre. On remarque que sa trajectoire décrit un cercle parfait avec l'ensemble de ses variables constantes ($r = 0,15; \dot{r} = 0; \ddot{r} = 0$).

Espace des phases $(r; \dot{r}; \ddot{r})$

Ainsi, la représentation dans l'espace des phases d'une telle trajectoire se résume à un point. En effet, les trois variables étant constantes, aucune variation n'apparaît dans l'espace des phases.

Section de Poincaré

De même, rien n'est observable dans la section de Poincaré : celle-ci étant une coupe de l'espace des phases.

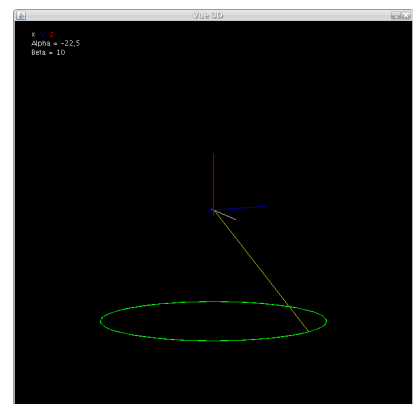


FIG. 2.11: Trajectoire physique ($r = 0,15$)

2.5.4 Point Quasi-périodique : $r = 0,103\ 106\ 2$

Trajectoire physique

Voici la trajectoire 3D du pendule élastique. Celle-ci semble régulière, ce qui caractérise la quasi-périodicité. En effet, la trajectoire a la même forme à chaque tour. Cependant la position initiale est différente de la position finale. La trajectoire est donc décalée mais se répète à l'identique. Ce qui semble à première vue indescriptible n'est en réalité que quasi-périodique.

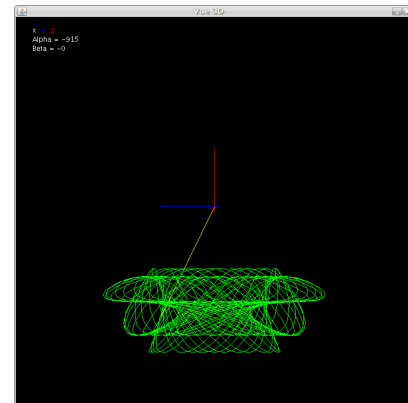


FIG. 2.12: Trajectoire physique

Espace des phases $(r; \dot{r}; \ddot{r})$

Ci-contre, la trajectoire du pendule dans l'espace des phases. On remarque que la trajectoire décrite est fermée, repliée sur elle-même en formant 5 cercles. Cette trajectoire est nettement définie : elle reste constante dans le temps.

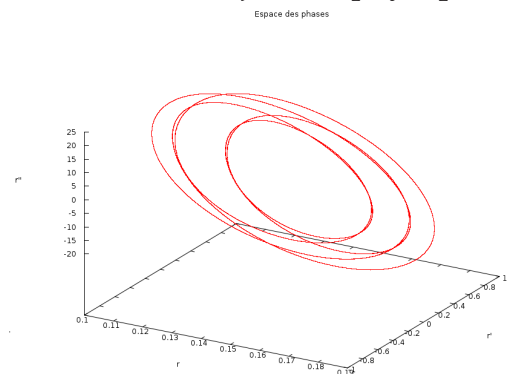


FIG. 2.13: Espace des phases

Section de Poincaré $(r; \dot{r} = 0; \ddot{r})$

Pour le plan de Poincaré, nous observons donc un ensemble de 6 points correspondant à l'intersection de ce plan avec la trajectoire dans l'espace des phases. Ainsi, on remarque qu'il n'y a que 6 endroits où la vitesse radiale est nulle. Si l'on regarde de plus près la trajectoire dans l'espace physique, ces 6 points sont facilement identifiables. Cette structure ponctuelle est caractéristique de la quasi-périodicité.

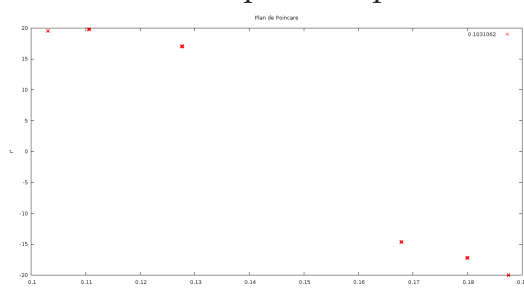


FIG. 2.14: Section de Poincare

Analyse spectrale

Nous observons un spectre comportant 3 fréquences, dont l'une est une combinaison linéaire des deux précédentes. Voici les fréquences : $f_1 = 3,89$ Hz, $f_2 = 2,30$ Hz et $f_3 = 1,55$ Hz qui correspondent respectivement à la période d'oscillation du ressort, la seconde à la période de rotation autour de l'axe et la troisième est une combinaison des deux selon la relation $0,95 \times f_2 + 1,1 \times f_3 = f_1$.

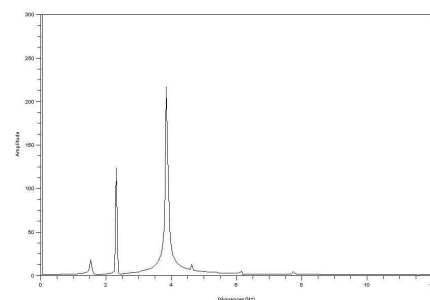


FIG. 2.15: Analyse spectrale

2.5.5 Bouffée Chaotique : $r = 0,042 \ 2$

Trajectoire physique

Voici la représentation dans l'espace physique de la trajectoire du pendule élastique. Nous observons que la masse ne passe pas régulièrement aux mêmes endroits. C'est à dire que si l'on regarde précisément l'endroit où l'élongation est maximale, la position selon l'axe $(0, z)$ n'est pas constante. La trajectoire ne semble régie par aucune loi.

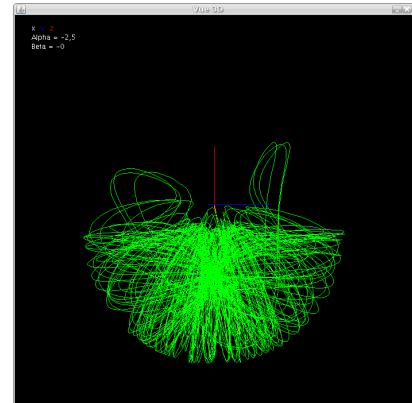


FIG. 2.16: Trajectoire physique

Espace des phases $(r; \dot{r}; \ddot{r})$

Dans cet espace des phases, nous pouvons observer une certaine régularité de la trajectoire. Cela se traduit par une courbe qui ne semble jamais se recouper, à chaque point correspond un unique état physique. La trajectoire tend à occuper un volume conséquent, et ne semble pas attirée par un point ou une trajectoire de phase particulière. Il est aussi important de noter qu'elle ne semble jamais s'approcher de la position d'équilibre $(0, 15; 0; 0)$, et même elle semblerai décrire une enveloppe autour de ce point.

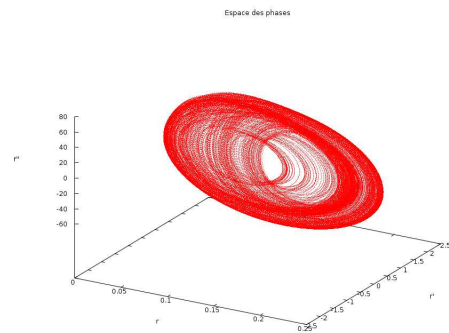


FIG. 2.17: Espace des phases

Section de Poincaré $(r; \dot{r} = 0; \ddot{r})$

Lorsque nous observons la section de Poincaré, nous pouvons remarquer que 2 formes se distinguent. En effet pour des valeurs de $r \in [0, 15 : 0, 25]$, plus grand que le rayon d'équilibre, il y a pour chaque rayon un seul état d'accélération possible : $\ddot{r} \sim -40\text{m.s}^{-1}$. A contrario, pour $r \in [0, 05 : 0, 15]$, l'accélération est décrite par un ensemble compris entre $[30; 70]$.

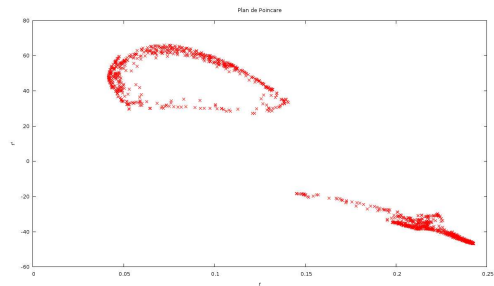


FIG. 2.18: Section de Poincare

Analyse spectrale

Dans ce spectre de Fourier, nous observons 3 fréquences principales, qui sont : $f_1 = 4 \text{ Hz}$, $f_2 = 2,6 \text{ Hz}$ et $f_3 = 1,3 \text{ Hz}$. Elles correspondent aussi à l'excitation du ressort. Aussi, on trouve l'équivalent d'un bruit de fond dû au caractère chaotique du mouvement. Par ailleurs, chaque pic de fréquence n'est pas lisse mais présente des petites irrégularités en amont et en aval. De même, dans ce cas, nous retrouvons la relation $0,95 \times f_2 + 1,1 \times f_3 = f_1$.

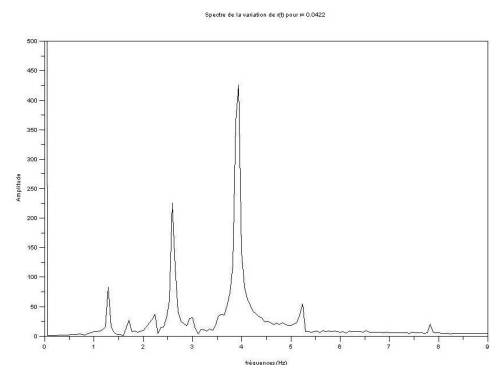


FIG. 2.19: Analyse spectrale

2.5.6 Fenêtre stable : $r = 0,039\ 15$

Trajectoire physique

Contrairement à ce que l'on vient d'observer dans le cas de la bouffée chaotique, la trajectoire est régulière. En effet, elle présente une structure ordonnée. On observe que la trajectoire est similaire à celle obtenue dans le cas de la quasi-périodicité. Bien que nous ayons franchi un domaine chaotique, la stabilité a été retrouvée.

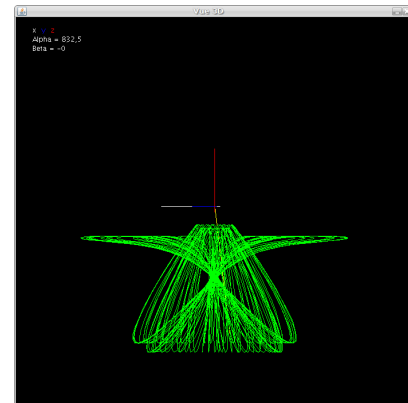


FIG. 2.20: Trajectoire physique

Espace des phases $(r; \dot{r}; \ddot{r})$

Ici, nous retrouvons également une trajectoire nettement définie. Les points que l'on observe ne gravitent pas autour de la trajectoire, ils en font intégralement partie. De plus, la trajectoire est fermée et se replie sur elle-même en 6 boucles.

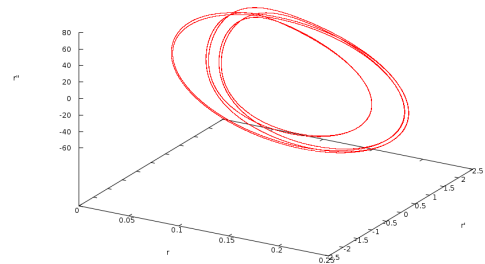


FIG. 2.21: Espace des phases

Section de Poincaré $(r; \dot{r} = 0; \ddot{r})$

En ce qui concerne la section de Poincaré, nous observons 7 points fixes dans le temps qui correspondent aux valeurs pour lesquelles la vitesse radiale est nulle, comme nous pouvons le remarquer sur la trajectoire. Le fait que la trajectoire de phase passent uniquement par ces 7 points caractérise la quasi-périodicité du mouvement du pendule, et donc la régularité qui en découle.

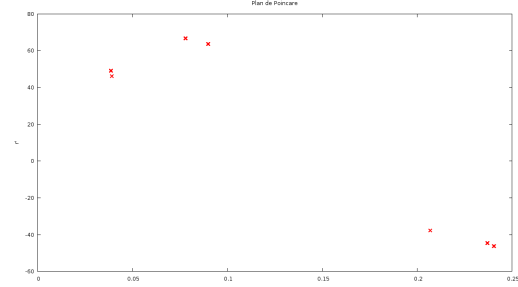


FIG. 2.22: Section de Poincaré

Analyse spectrale

Pour ce rayon initial, le spectre de Fourier nous indique que le mouvement de notre système se caractérise par un ensemble de trois fréquences $f_1 = 3,93$ Hz, $f_2 = 2,65$ Hz et $f_3 = 1,3$ Hz. Remarquons que la relation de combinaison linéaire, $0,95 \times f_2 + 1,1 \times f_3 = f_1$ est vérifiée. Là aussi, f_1 correspond à l'oscillation du ressort, tandis que les deux autres sont à relier au mouvement de rotation du ressort.

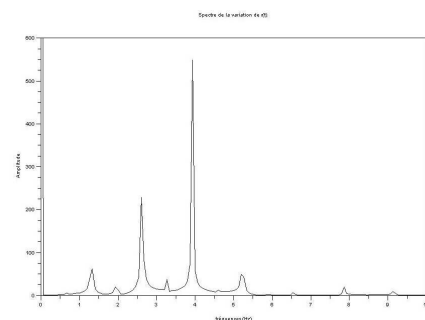


FIG. 2.23: Analyse spectrale

2.5.7 Chaos : $r = 0,032$

Trajectoire physique

Voici la représentation de la trajectoire dans l'espace physique. Il est impossible d'y trouver une quelconque structure, quelque soit le temps de l'expérience et l'angle d'observation. Nous ne pouvons rien déceler ici avec uniquement la trajectoire.

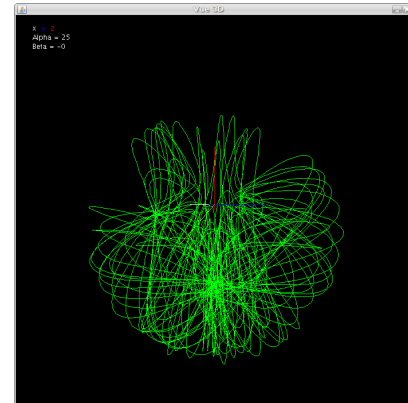


FIG. 2.24: Trajectoire physique

Espace des phases $(r; \dot{r}; \ddot{r})$

Dans l'espace des phases associé à ce cas d'étude, nous observons toujours une forme torique. Cependant cette dernière ne semble jamais repasser aux mêmes endroits. Contrairement à l'espace des phases de la bouffée chaotique, le pendule semble plus s'approcher de la position d'équilibre, car la trajectoire s'approche davantage de ce point.

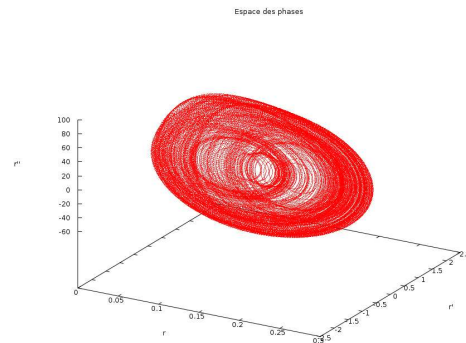


FIG. 2.25: Espace des phases

Section de Poincaré $(r; \dot{r} = 0; \ddot{r})$

Ici, dans la section de Poincaré, nous n'observons aucune structure nette et organisée. Cependant, on peut tout de même trouver les mêmes caractéristiques que pour la bouffée chaotique avec les intervalles, à ceci près que les intervalles d'accélération sont plus flous.

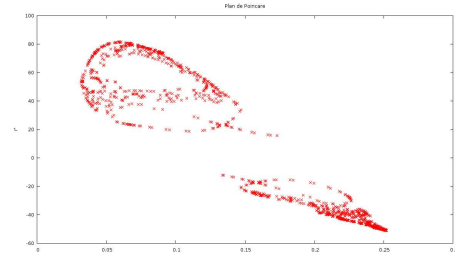


FIG. 2.26: Section de Poincaré

Analyse spectrale

Nous pouvons remarquer que le spectre de Fourier présente quatre principaux pics qui correspondent aux fréquences $f_1 = 4,07$ Hz, $f_2 = 2,5$ Hz, $f_3 = 2,2$ Hz, et $f_4 = 1,6$ Hz. Chacun de ces pics n'est pas nettement défini, la courbe n'est pas lisse à leur base. La première fréquence présentant la plus grande amplitude correspond à l'oscillation du ressort.

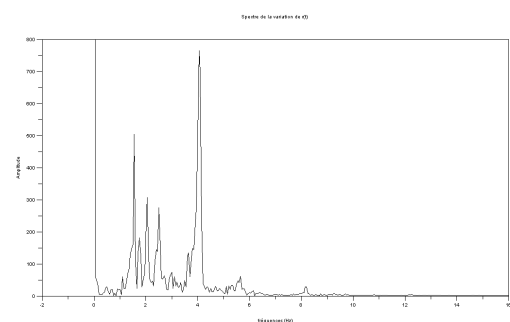


FIG. 2.27: Analyse spectrale

Par ailleurs, nous remarquons un important bruit qui est la signature du chaos. Ce spectre diffère de manière importante des spectres des mouvements périodiques et quasi-périodiques.

2.5.8 Conclusion de l'étude

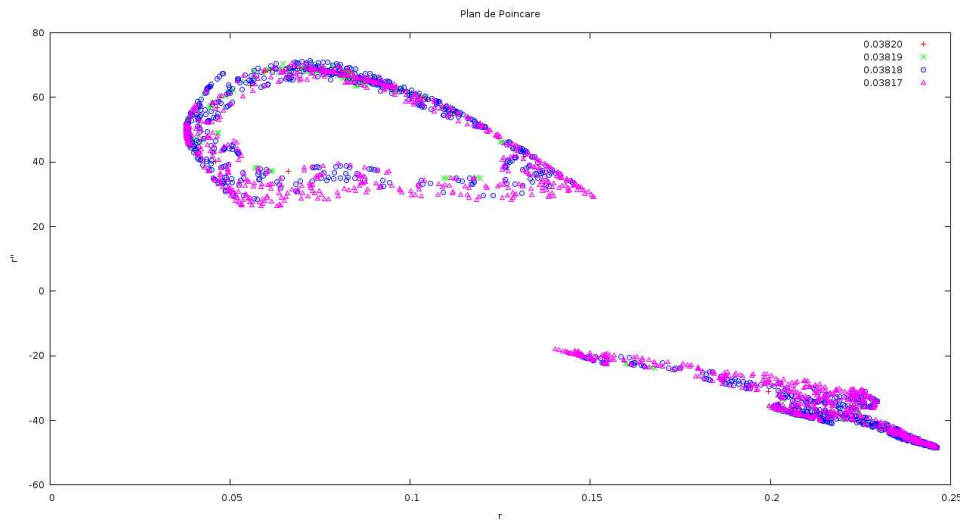


FIG. 2.28: Sensibilité aux conditions initiales dans la section de Poincaré

La principale caractéristique d'un système chaotique est sa dépendance aux conditions initiales. Nous avons donc étudié l'évolution du rayon en fonction du temps à l'aide de la section de Poincaré. Nous avons alors observé que les structures décrites divergent très rapidement. En effet, pour des conditions initiales très proches $r = \{0,038\ 20; 0,038\ 19; 0,038\ 18; 0,038\ 17\}$, la surface occupée par les points tend à s'agrandir, et cela pour une variation d'ordre 0,001 cm. La sensibilité aux conditions initiales du pendule élastique est donc vérifiée. Le graphique suivant en est la parfaite illustration.

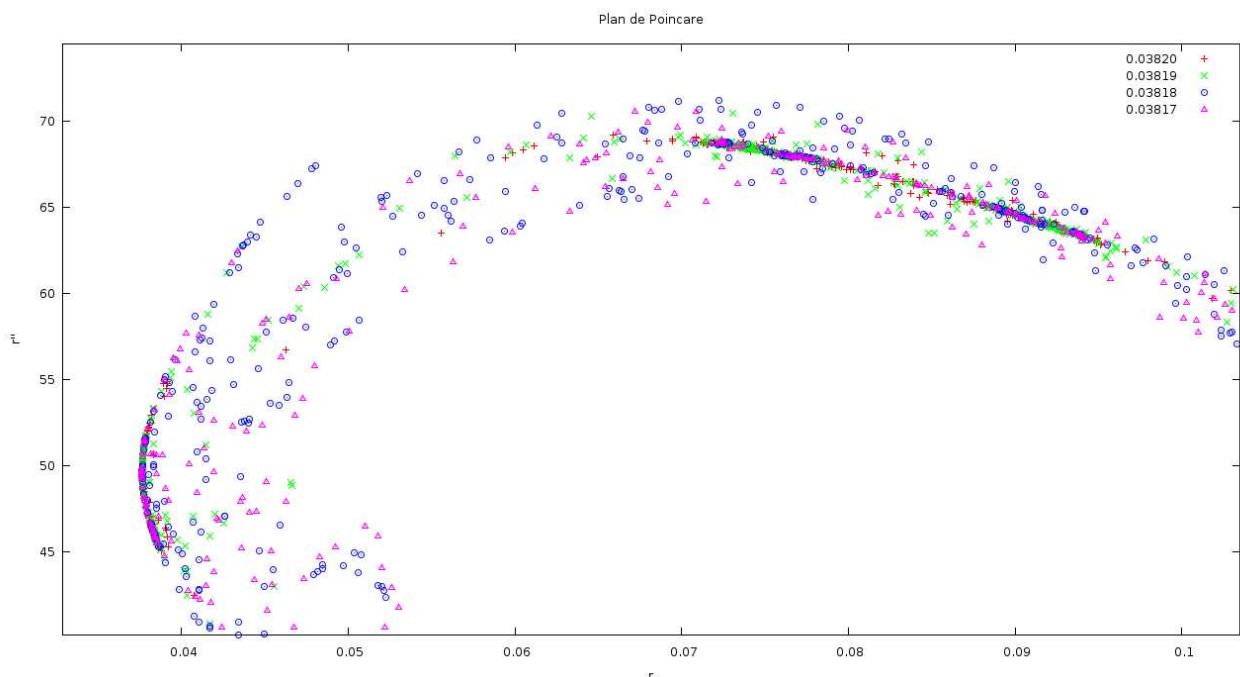


FIG. 2.29: Sensibilité aux conditions initiales - Détail

À la suite des différentes observations effectuées selon la démarche d'analyse, et à l'aide du graphique précédent, nous pouvons affirmer que notre pendule élastique est indéniablement chaotique du fait de son infime sensibilité aux conditions initiales.

Conclusion et perspectives

Bien que l'on emploie le mot chaos couramment, on ignore souvent son sens physique et ce qui en découle scientifiquement. L'objectif premier de ce projet était alors d'appréhender la théorie du chaos et d'en découvrir les principaux aspects. Ainsi, nous avons tous élargi notre champ de connaissances physiques en adaptant les méthodes du chaos à notre système. Nous avons alors rencontré plusieurs difficultés étant donné l'absence d'études préexistantes de notre système. Nous avons participé à un projet scientifique dans sa globalité. En effet, la démarche scientifique a été suivie dans son ensemble : de la recherche d'informations à la modélisation des résultats obtenus.

Pour répondre aux différents objectifs du sujet, nous avons réalisé l'étude du mouvement du pendule élastique suivant la démarche suivante.

Dans un premier temps, nous avons modélisé la trajectoire du pendule en établissant et résolvant ses équations différentielles. Après une longue étude des différents outils du chaos, nous avons entrepris la programmation permettant la visualisation et l'analyse au sens de la dynamique des systèmes du mouvement de la masse. Ceci a permis la mise en évidence de la sensibilité aux conditions initiales, propriété fondamentale d'un système chaotique.

De plus, le travail en groupe nous a appris à remettre en question nos résultats afin de se réorienter rapidement en cas de fausse route, de se motiver mutuellement et d'échanger nos doutes quant aux résultats obtenus. Enfin, la rencontre avec M.Letellier a été pour nous une occasion d'échanger nos résultats et n'a fait que renforcer notre implication dans le projet.

Bien que le chaos ait été mis en évidence grâce aux notions d'espace des phases, du plan de Poincaré et du spectre de Fourier, d'autres outils restent à approfondir tels que les langues d'Arnold, les attracteurs d'Henon et la théorie de KAM. De même, nous ne nous sommes intéressés qu'à un seul type de système, c'est-à-dire les systèmes conservatifs. La notion de chaos reste donc non étudiée dans le cas des systèmes dissipatifs.

Bibliographie

- [1] HORST Stöcker, FRANCIS Jundt, GEORGES Guillaume, *Toute la Physique*, Dunod, 1999.
- [2] RICHARD Jean-pierre, *Mathématiques pour les systèmes dynamiques*, Hermès Science : Lavoisier , 2002.
- [3] GENDREAU Bernard, *L'essentiel de la mécanique du point matériel et du système de deux points matériels*, Ellipses , 1993.
- [4] LASCAUD Patrick, THÉODOR Raymond, *Analyse numérique matricielle appliquée à l'art de l'ingénieur*, Masson , 1987.
- [5] MOON Francis C, *Choatic and fractal dynamics*, New York Wiley, 1992.
- [6] DAVENAY Robert, *An introduction to chaotic dynamical systems*, Addison-Wesley, 1987.
- [7] YON Jérôme, *Etude du système de Lorenz*,
- [8] LETELLIER Christophe, *Le chaos dans la nature : l'ordre au delà du temps*, 2005.
- [9] PARKER Thomas.S, CHUA Leon.O, *Practical Numerical Algorithms for Chaotic Systems*, Springer-Verlag, 1989.
- [10] <http://just.loic.free.fr> Site personnel, Valide à la date du 11/06/2008.
- [11] http://fr.wikipedia.org/wiki/Th%C3%A9orie_du_chaos Valide à la date du 11/06/2008.
- [12] <http://etudiants.ensimag.fr/perso/Guillaume.Molleda/tipe2.htm> Site personnel, Valide à la date du 11/06/2008.
- [13] <http://just.loic.free.fr/index.php?page=elem> Site personnel, Valide à la date du 11/06/2008.
- [14] <http://www.zeuscat.com/andrew/chaos/lorenz3d.gif> Valide à la date du 11/06/2008.

Annexe A

Sujet

Projet P6-3

Intitulé du projet : Etude du mouvement "chaotique" d'une masse attachée à un ressort.

Encadrant : Jérôme Yon

Créneau horaire : chaque mercredi de 13h à 14h30

Elèves : Martin ROSALIE, Myriam DE SAINT JEAN, Mathilde PIERRE-BOURGEON, Amal ROCHD, Selima MASMOUDI

Objectifs du projet :

1. Choisir une configuration physique : une masse m attachée à un ressort de raideur k et de longueur à vide l_0 . Retrouver les équations différentielles du mouvement (P2).
2. Choisir un utilitaire du domaine public sur internet qui permet la résolution numérique d'un tel système d'équations différentielles.
3. Mettre au point un procédé permettant l'illustration (la visualisation) du mouvement de la masse en fonction de la résolution précédente des équations différentielles.
4. Analyse bibliographique des études des mouvements chaotiques (science appelée dynamique des systèmes). Définir les outils et paramètres nécessaires à l'analyse de tels systèmes.
5. Analyse au sens de la dynamique des systèmes du mouvement de la masse attachée au ressort à partir de sa résolution numérique (item 2). On pourra observer l'effet du changement des conditions initiales et des propriétés du système étudié.
6. Mise au point d'un document pédagogique en format papier et un diaporama illustrant l'ensemble du travail effectué.

Annexe B

Compte Rendu de la rencontre avec M.Christophe LETELLIER

M.Christophe LETELLIER, chercheur au CORIA de ROUEN, le 15 mai 2008.

B.1 Diagramme de bifurcation

La première remarque porte sur ce que l'on pensait être des diagrammes de bifurcation (cf Annexe E.1). Seules les variations de paramètres physiques comme la masse ou la constante de raideur du ressort engendrent ces types de diagrammes. La variation des conditions initiales comme r, \dot{r} permet d'obtenir des diagrammes mais qui ne peuvent en aucun cas être qualifiés comme étant "de bifurcation".

B.2 Notion de fréquence

Puis, nous avons abordé la notion de quasi-périodicité : dans notre système, il semble apparaître deux fréquences f_1 et f_2 :

- lorsque $\frac{f_1}{f_2} = \frac{p}{q} =$ rationnel, le système est périodique
- lorsque $\frac{f_1}{f_2} \neq \frac{p}{q}$, le système est quasi-périodique

On peut alors obtenir un spectre discret de Fourier. De plus, une trajectoire quasi-périodique ne repasse jamais au même endroit mais balaie tout l'espace physique. Par contre, on observe une "période" pour laquelle la trajectoire reproduit une même forme, sans faire de "re-bouclage" exact sur elle même.

Concernant f_1 et f_2 , l'une des fréquences dépendrait d'un paramètre $\frac{k}{m}$ par exemple (c'est la fréquence due au ressort), et l'autre est celle rajoutée par la rotation du système. Il faut donc que l'on détermine de quoi dépendent ces deux fréquences en observant d'une part le ressort pris dans un mouvement $1D$ et d'autre part, la vitesse d'un pendule simple en rotation. Nous pourrions alors vérifier la cohérence physique des fréquences obtenue à partir du spectre de Fourier.

Les notions de systèmes périodiques, quasi-périodiques et chaotiques seront finalement présentes dans toute la suite de nos investigations sur le chaos.

B.3 Notion de systèmes dissipatif et conservatif

Notre système est conservatif, or selon C. Letellier, l'étude de ces systèmes est plus complexe, et complètement différente de celle menée avec des systèmes dissipatifs.

En effet, il n'existe pas d'attracteur pour ce type de système et l'étude de diagrammes de bifurcation est elle même très complexe. Nous ne pouvons donc pas nous familiariser avec le cas d'école des équations de Lorenz qui traduisent un système dissipatif.

B.4 Notion d'espace des phases

Nous pouvons remarquer ici qu'avant cette rencontre, le choix des axes pour tracer l'espace des phases était, à chaque séance, un sujet de désaccord. Nos questions étaient, entre autre : "Quelle est la dimension de cet espace?", et, "Que faut-il utiliser, pour tracer cet espace?" :

- chacune des variables avec le même ordre de dérivation : $(\dot{r}; \dot{\theta}; \dot{\varphi})$ ou encore $(\ddot{r}; \ddot{\theta}; \ddot{\varphi})$
- la variable et ses dérivées successives : $(r; \dot{r}; \ddot{r})$ ou encore $(\theta; \dot{\theta}; \ddot{\theta})$

En effet, dans les définitions trouvées, l'espace des phases comporte autant de dimensions que le système a de degrés de liberté.

C. Letellier nous a alors informé que le chaos se retrouvait au travers toutes les variables. Il nous a indiqué qu'il était plus adroit de traiter l'espace avec la variable et ses dérivées. Pour le pendule élastique, l'espace $(r; \dot{r}; \ddot{r})$ est celui qui permettrait une bonne visualisation du chaos.

Il faut donc nous intéresser à la section de Poincaré. Dans ce cas, notons qu'il faut un grand nombre de points (de l'ordre des milliers) afin d'obtenir des résultats cohérents. Une fois notre section de Poincaré obtenue, (r, \ddot{r}) pour $\dot{r} = 0$ dans l'espace des phases, il ne nous reste qu'à étudier cette section pour plusieurs conditions initiales r . Une fois ces différentes trajectoires obtenues, nous étudierons les variations des paramètres.

En effet, d'après la théorie de KAM (Kolmogorov-Arnold-Moser), en rapport avec la mécanique céleste, les paramètres influent sur la grandeur de la mer chaotique. Nous avons par ailleurs évoqué la représentation de f_1 en fonction de f_2 . Le diagramme normalement obtenu est connu sous le nom de "langues d'Arnold".

Il ressort de cette rencontre un bilan très positif. En effet, cela nous a permis de savoir quels outils allaient nous permettre d'étudier le chaos dans notre système, quelles nouvelles théories aborder (Hénon, application standard ...). Nous avons pu aussi nous rassurer sur la possibilité d'atteindre le chaos avec ce système. Nous avons eu la confirmation de la démarche suivie, nos efforts n'ont pas été vains.

Annexe C

Code en Java

C.1 Classe principale

```
public class JPomme
{
    public static void main (String arg[])
    {
        JChoix essai = new JChoix();
    }
}
```

C.2 Classe du choix des conditions initiales

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class JChoix extends JFrame
{
    private static final long serialVersionUID = 42L;
    private JLabel gravite = new JLabel ("Gravite");
    private JTextField graviteValeur= new JTextField ("9.81",10);
    private JLabel masse = new JLabel ("Masse");
    private JTextField masseValeur= new JTextField ("0.162",10);
    private JLabel raideur = new JLabel ("Constante de raideur");
    private JTextField raideurValeur= new JTextField ("74.916549",10);
    private JLabel longueurAVide = new JLabel ("Longueur a vide");
    private JTextField longueurAVideValeur= new JTextField ("0.12",10);
    private JLabel frottement = new JLabel ("Coefficient de frottement");
    private JTextField frottementValeur= new JTextField ("0",10);
    private JLabel tempsMin= new JLabel ("Temps initial");
    private JTextField tempsMinValeur = new JTextField ("0",10);
    private JLabel tempsMax = new JLabel ("Temps Final");
    private JTextField tempsMaxValeur = new JTextField ("20",10);
    private JLabel rayonInitial = new JLabel ("Rayon ini");
    private JTextField rayonInitialValeur = new JTextField ("0.15",10);
    private JLabel rayon_primeInitial = new JLabel ("Vitesse Radiale ini");
    private JTextField rayon_primeInitialValeur = new JTextField ("0",10);
}
```

```

private JLabel thetaInitial = new JLabel ("Angle Theta ini en degre");
private JTextField thetaInitialValeur = new JTextField ("225",10);
private JLabel theta_primeInitial = new JLabel ("Vitesse ang ini Theta
    (t/s)");
private JTextField theta_primeInitialValeur = new JTextField ("0",10);
30 private JLabel phiInitial = new JLabel ("Angle Phi ini en degre");
private JTextField phiInitialValeur = new JTextField ("0",10);
private JLabel phi_primeInitial= new JLabel ("Vitesse ang ini Phi (t/s)
    ");
private JTextField phi_primeInitialValeur= new JTextField ("1.530616901
    ",10);
private JButton validation = new JButton ("Calcul !");
35 private JButton trajectoire = new JButton ("Trajectoire");

private JLabel vecteur = new JLabel ("Vecteur Normal au plan de
    Poincare: x,y,z");
private JTextField x_vecteur= new JTextField ("1",5);
private JTextField y_vecteur= new JTextField ("2",5);
40 private JTextField z_vecteur= new JTextField ("-1",5);

private JLabel point = new JLabel ("Point du plan de Poincare : x,y,z"
    );
private JTextField x_point = new JTextField ("0",5);
private JTextField y_point = new JTextField ("0",5);
45 private JTextField z_point = new JTextField ("0",5);

private JLabel textPoincare1 = new JLabel ("Espace des phase en 3D");
private JLabel textPoincare2 = new JLabel ("( En definissant : Abscisse
    , Coordonnee et Cote )");
private JLabel textPoincare3 = new JLabel ("Choisir un plan de Poincare
    dans l'espace");
50 private JLabel textPoincare4 = new JLabel ("(En definissant un vecteur
    normal & un point du pan)");

private JButton calculPoincare = new JButton ("Plan de Poincare");
private JButton calculePhase = new JButton ("Espace des Phases en 3D")
    ;
55 private String[] phase = {"1", "Rayon", "Rayon ' ", "Rayon ''", "Theta",
    "Theta ' ", "Theta ''", "Phi", "Phi ' ", "Phi '' " };
private String[] operateur = {" x " , " / " };

private JLabel poincareX = new JLabel ("abscisse :");
60 private JComboBox phaseList1X = new JComboBox(phase);
private JComboBox phaseList2X = new JComboBox(phase);
private JComboBox phaseList3X = new JComboBox(phase);

private JComboBox operateur1X = new JComboBox(operateur);
65 private JComboBox operateur2X = new JComboBox(operateur);
  
```

```

70     private JLabel poincareY = new JLabel ("ordonnee :");

    private JComboBox phaseList1Y = new JComboBox(phase);
    private JComboBox phaseList2Y = new JComboBox(phase);
    private JComboBox phaseList3Y = new JComboBox(phase);

75     private JComboBox operateur1Y = new JComboBox(operateur);
    private JComboBox operateur2Y = new JComboBox(operateur);

    private JLabel poincareZ = new JLabel ("cote :");

80     private JComboBox phaseList1Z = new JComboBox(phase);
    private JComboBox phaseList2Z = new JComboBox(phase);
    private JComboBox phaseList3Z = new JComboBox(phase);

85     private JComboBox operateur1Z = new JComboBox(operateur);
    private JComboBox operateur2Z = new JComboBox(operateur);

    private JPanel panneau = new JPanel() ;

90     private Jressort3D ressort3D;
    private JPoincare poincare;

    private double r, rp, t, tp, p, pp, mi, ma, g, mass, constanteRaideur, longueurVide,
        lambda, an, bn, cn, xpp, ypp, zpp;
    private double xCoef1,
95         xOperateur1,
        xCoef2,
        xOperateur2,
        xCoef3 ,
        yCoef1,
100        yOperateur1,
        yCoef2,
        yOperateur2,
        yCoef3 ,
        zCoef1,
105        zOperateur1,
        zCoef2,
        zOperateur2,
        zCoef3 ;

110     public JChoix ()
    {
        super ("Conditions Initiales");
        setBounds (10,10,700,480);
        setDefaultCloseOperation(EXIT_ON_CLOSE);

115

        JPanel grilleRessort3D = new JPanel(new GridLayout(7,4,5,5));

        JPanel grillePhaseTitre = new JPanel(new GridLayout(1,2,5,5))
            ;
    
```

```
120     JPanel grillePhase = new JPanel(new GridLayout(3,6,5,5));

    JPanel grillePoincare = new JPanel(new GridLayout(4,2,5,5));

125     JPanel grilleVect = new JPanel(new GridLayout(1,3,5,5));
    JPanel grillePoin = new JPanel(new GridLayout(1,3,5,5));

    grilleRessort3D.add(gravite);
    grilleRessort3D.add(graviteValeur);

130     grilleRessort3D.add(masse);
    grilleRessort3D.add(masseValeur);

    grilleRessort3D.add(raideur);
135     grilleRessort3D.add(raideurValeur);

    grilleRessort3D.add(longueurAVide);
    grilleRessort3D.add(longueurAVideValeur);

140     grilleRessort3D.add(tempsMin);
    grilleRessort3D.add(tempsMinValeur);

    grilleRessort3D.add(tempsMax);
    grilleRessort3D.add(tempsMaxValeur);

145     grilleRessort3D.add(rayonInitial);
    grilleRessort3D.add(rayonInitialValeur);

    grilleRessort3D.add(rayon_primeInitial);
150     grilleRessort3D.add(rayon_primeInitialValeur);

    grilleRessort3D.add(thetaInitial);
    grilleRessort3D.add(thetaInitialValeur);

155     grilleRessort3D.add(theta_primeInitial);
    grilleRessort3D.add(theta_primeInitialValeur);

    grilleRessort3D.add(phiInitial);
    grilleRessort3D.add(phiInitialValeur);

160     grilleRessort3D.add(phi_primeInitial);
    grilleRessort3D.add(phi_primeInitialValeur);

    grilleRessort3D.add(frottement);
165     grilleRessort3D.add(frottementValeur);

    validation.addActionListener (new ActionListener ()
    {
        public void actionPerformed (ActionEvent e)
170     {
            g=Double.parseDouble(graviteValeur.getText());
            mass=Double.parseDouble(masseValeur.getText());
```

175

```

    constanteRaideur=Double.parseDouble(raideurValeur.
        getText());
    longueurVide=Double.parseDouble(longueurAVideValeur.
        getText());
    lambda=Double.parseDouble(frottementValeur.getText());

    r = Double.parseDouble(rayonInitialValeur.getText());
    rp = Double.parseDouble(rayon_primeInitialValeur.
        getText());
  
```

180

```

    t= Math.PI*Double.parseDouble(thetaInitialValeur.
        getText())/180;
    tp= Double.parseDouble(theta_primeInitialValeur.
        getText()*(2*Math.PI));
  
```

185

```

    p= Math.PI*Double.parseDouble(phiInitialValeur.getText
        ())/180;
    pp= Double.parseDouble(phi_primeInitialValeur.getText
        ())*(2*Math.PI);
  
```

190

```

    mi= Double.parseDouble(tempsMinValeur.getText());
    ma= Double.parseDouble(tempsMaxValeur.getText());

    ressort3D = new Jressort3D(r, rp, t, tp, p, pp, g, mass,
        constanteRaideur, longueurVide, lambda) ;
    ressort3D.calcul(10000, mi, ma);
    ressort3D.ecriture();
  
```

195

```

    }
  });
  grilleRessort3D.add(validation);
  
```

200

```

  trajectoire.addActionListener (new ActionListener ()
  {
    public void actionPerformed (ActionEvent e)
    {
      Jaffichage affichage = new Jaffichage (ressort3D) ;
    }
  });
  grilleRessort3D.add(trajectoire);
  
```

205

```

  panneau.add(grilleRessort3D);
  
```

210

```

    grillePhaseTitre.add(textPoincare1);
    grillePhaseTitre.add(textPoincare2);
  panneau.add(grillePhaseTitre);
  
```

215

```

    grillePhase.add(poincareX);
    phaseList1X.setSelectedIndex(1);
    grillePhase.add(phaseList1X);
    grillePhase.add(operateur1X);

    grillePhase.add(phaseList2X);
    grillePhase.add(operateur2X);
  
```

```
grillePhase.add(phaseList3X);

220 grillePhase.add(poincareY);
    phaseList1Y.setSelectedIndex(4);
    grillePhase.add(phaseList1Y);
    grillePhase.add(operateur1Y);

225 grillePhase.add(phaseList2Y);
    grillePhase.add(operateur2Y);
    grillePhase.add(phaseList3Y);

    grillePhase.add(poincareZ);
230 phaseList1Z.setSelectedIndex(7);
    grillePhase.add(phaseList1Z);
    grillePhase.add(operateur1Z);

    grillePhase.add(phaseList2Z);
235 grillePhase.add(operateur2Z);
    grillePhase.add(phaseList3Z);

panneau.add(grillePhase);

240 grillePoincare.add(textPoincare3);

    grillePoincare.add(textPoincare4);

245 grillePoincare.add(vecteur);

    grilleVect.add(x_vecteur);
    grilleVect.add(y_vecteur);
    grilleVect.add(z_vecteur);

250 grillePoincare.add(grilleVect);

    grillePoincare.add(point);

    grillePoin.add(x_point);
255 grillePoin.add(y_point);
    grillePoin.add(z_point);

grillePoincare.add(grillePoin);

260 calculEphase.addActionListener (new ActionListener ()
{
    public void actionPerformed (ActionEvent e)
    {
265 xCoef1 = phaseList1X.getSelectedIndex();
        xOperateur1 = operateur1X.getSelectedIndex();
        xCoef2 = phaseList2X.getSelectedIndex();
        xOperateur2 = operateur2X.getSelectedIndex();
        xCoef3 = phaseList3X.getSelectedIndex();

270 yCoef1 = phaseList1Y.getSelectedIndex();
```

275
280
285
290
295
300
305
310
315

```

yOperateur1 = operateur1Y.getSelectedIndex();
yCoef2 = phaseList2Y.getSelectedIndex();
yOperateur2 = operateur2Y.getSelectedIndex();
yCoef3 = phaseList3Y.getSelectedIndex();

zCoef1 = phaseList1Z.getSelectedIndex();
zOperateur1 = operateur1Z.getSelectedIndex();
zCoef2 = phaseList2Z.getSelectedIndex();
zOperateur2 = operateur2Z.getSelectedIndex();
zCoef3 = phaseList3Z.getSelectedIndex();

mi= Double.parseDouble(tempsMinValeur.getText());
ma= Double.parseDouble(tempsMaxValeur.getText());

poincare = new JPoincare(ressort3D,10000,mi,ma);
poincare.calculEspace(xCoef1,xOperateur1,xCoef2,
    xOperateur2,xCoef3,yCoef1,yOperateur1,yCoef2,
    yOperateur2,yCoef3,zCoef1,zOperateur1,zCoef2,
    zOperateur2,zCoef3 );
poincare.ecritureEspace();
    }
});
grillePoincare.add(calculEphase);

calculPoincare.addActionListener (new ActionListener ()
{
    public void actionPerformed (ActionEvent e)
    {
        an = Double.parseDouble( x_vecteur.getText());
        bn = Double.parseDouble( y_vecteur.getText());
        cn = Double.parseDouble( z_vecteur.getText());
        xpp = Double.parseDouble( x_point.getText());
        ypp = Double.parseDouble( y_point.getText());
        zpp = Double.parseDouble( z_point.getText());

        poincare.calculPP(an,bn,cn,xpp,ypp,zpp);
        poincare.ecriturePP();
        poincare.trace();
    }
});
grillePoincare.add(calculPoincare);

panneau.add(grillePoincare);

setResizable (false);
setVisible (true);
setContentPane (panneau);
    }
}
    
```


C.3 Classe de résolution des équations différentielles couplées

```

import java.io.*;
import java.util.*;
public class Jressort3D
{ //variables de l'objet ressort3D
5   private double RayonInitial, //conditions initiales
        Rayon_primeInitial,
        ThetaInitial,
        Theta_primeInitial,
10   PhiInitial,
        Phi_primeInitial,
        gravite,
        masse,
        constanteRaideur,
        longueurVide,
15   lambda;

    private double temps,tmin,tmax;
    private double rayonMax = 0;
    private double Rayon_seconde,Rayon_prime,Rayon; //variables d'etat
    private double Theta_seconde,Theta_prime,Theta;
20   private double Phi_seconde,Phi_prime,Phi;
    private double [][] resultat;
    private int nombrePoints;

//Constructeur
25   public Jressort3D (
        double _RayonInitial,
        double _Rayon_primeInitial,
        double _ThetaInitial,
        double _Theta_primeInitial,
30   double _PhiInitial,
        double _Phi_primeInitial,
        double _gravite,
        double _masse,
        double _constanteRaideur,
        double _longueurVide,
35   double _lambda
        )
    {
        RayonInitial=_RayonInitial;
40   Rayon_primeInitial=_Rayon_primeInitial;
        ThetaInitial=_ThetaInitial;
        Theta_primeInitial=_Theta_primeInitial;
        PhiInitial=_PhiInitial;
        Phi_primeInitial=_Phi_primeInitial;
45   gravite=_gravite;
        masse=_masse;
        constanteRaideur=_constanteRaideur;
        longueurVide=_longueurVide;
        lambda=_lambda;
    }
}

```

```

50     }

//Methodes
public void calcul(long frequence, double _tmin, double _tmax)
{
55     double aRayon1, aRayon2, aRayon3, aRayon4, aRayon_prime1, aRayon_prime2
        , aRayon_prime3, aRayon_prime4;
    double aTheta1, aTheta2, aTheta3, aTheta4, aTheta_prime1, aTheta_prime2
        , aTheta_prime3, aTheta_prime4;
    double aPhi1, aPhi2, aPhi3, aPhi4, aPhi_prime1, aPhi_prime2, aPhi_prime3
        , aPhi_prime4;
    double pas;
    int i;
60     double coef, nVect;

    tmin = _tmin ;
    tmax = _tmax ;

65     pas = 1.0/frequence;
    nombrePoints= (int) ((tmax -tmin)/pas);
    resultat = new double [nombrePoints+1][18];

    Rayon=RayonInitial;
70     Rayon_prime=Rayon_primeInitial;
    Theta=ThetaInitial;
    Theta_prime=Theta_primeInitial;
    Phi=PhiInitial;
    Phi_prime=Phi_primeInitial;

75     temps=tmin;
    System.out.println("Voici le pas : "+1000*pas+" millisecondes");
    for(i=0; i<nombrePoints; i++)
    {
80         aRayon_prime1=Rayon_seconde(temps, Theta, Theta_prime, Rayon,
            Rayon_prime, Phi, Phi_prime);
        aTheta_prime1=Theta_seconde(temps, Theta, Theta_prime, Rayon,
            Rayon_prime, Phi, Phi_prime);
        aPhi_prime1=Phi_seconde(temps, Theta, Theta_prime, Rayon,
            Rayon_prime, Phi, Phi_prime);
        aRayon1=Rayon_prime;
        aTheta1=Theta_prime;
85         aPhi1=Phi_prime;

        aRayon_prime2=Rayon_seconde(temps+pas/2, Theta+pas/2*aTheta1,
            Theta_prime+pas/2*aTheta_prime1, Rayon+pas/2*aRayon1,
            Rayon_prime+pas/2*aRayon_prime1, Phi+pas/2*aPhi1, Phi_prime+
            pas/2*aPhi_prime1);
        aTheta_prime2=Theta_seconde(temps+pas/2, Theta+pas/2*aTheta1,
            Theta_prime+pas/2*aTheta_prime1, Rayon+pas/2*aRayon1,
            Rayon_prime+pas/2*aRayon_prime1, Phi+pas/2*aPhi1, Phi_prime+
            pas/2*aPhi_prime1);
        aPhi_prime2=Phi_seconde(temps+pas/2, Theta+pas/2*aTheta1,
            Theta_prime+pas/2*aTheta_prime1, Rayon+pas/2*aRayon1,

```

```

    Rayon_prime+pas/2*aRayon_prime1,Phi+pas/2*aPhi1,Phi_prime+
    pas/2*aPhi_prime1);
90  aRayon2=Rayon_prime+pas/2*aRayon_prime1;
    aTheta2=Theta_prime+pas/2*aTheta_prime1;
    aPhi2=Phi_prime+pas/2*aPhi_prime1;

    aRayon_prime3=Rayon_seconde(temps+pas/2,Theta+pas/2*aTheta2,
    Theta_prime+pas/2*aTheta_prime2,Rayon+pas/2*aRayon2,
    Rayon_prime+pas/2*aRayon_prime2,Phi+pas/2*aPhi2,Phi_prime+
    pas/2*aPhi_prime2);
95  aTheta_prime3=Theta_seconde(temps+pas/2,Theta+pas/2*aTheta2,
    Theta_prime+pas/2*aTheta_prime2,Rayon+pas/2*aRayon2,
    Rayon_prime+pas/2*aRayon_prime2,Phi+pas/2*aPhi2,Phi_prime+
    pas/2*aPhi_prime2);
    aPhi_prime3=Phi_seconde(temps+pas/2,Theta+pas/2*aTheta2,
    Theta_prime+pas/2*aTheta_prime2,Rayon+pas/2*aRayon2,
    Rayon_prime+pas/2*aRayon_prime2,Phi+pas/2*aPhi2,Phi_prime+
    pas/2*aPhi_prime2);
    aRayon3=Rayon_prime+pas/2*aRayon_prime2;
    aTheta3=Theta_prime+pas/2*aTheta_prime2;
    aPhi3=Phi_prime+pas/2*aPhi_prime2;
100

    aRayon_prime4=Rayon_seconde(temps+pas,Theta+pas*aTheta3,
    Theta_prime+pas*aTheta_prime3,Rayon+pas*aRayon3,Rayon_prime
    +pas*aRayon_prime3,Phi+pas*aPhi3,Phi_prime+pas*aPhi_prime3)
    ;
    aTheta_prime4=Theta_seconde(temps+pas,Theta+pas*aTheta3,
    Theta_prime+pas*aTheta_prime3,Rayon+pas*aRayon3,Rayon_prime
    +pas*aRayon_prime3,Phi+pas*aPhi3,Phi_prime+pas*aPhi_prime3)
    ;
    aPhi_prime4=Phi_seconde(temps+pas,Theta+pas*aTheta3,
    Theta_prime+pas*aTheta_prime3,Rayon+pas*aRayon3,Rayon_prime
    +pas*aRayon_prime3,Phi+pas*aPhi3,Phi_prime+pas*aPhi_prime3)
    ;
105  aRayon4=Rayon_prime+pas*aRayon_prime3;
    aTheta4=Theta_prime+pas*aTheta_prime3;
    aPhi4=Phi_prime+pas*aPhi_prime3;

    Rayon_prime=Rayon_prime+pas*(aRayon_prime1+2*aRayon_prime2+2*
    aRayon_prime3+aRayon_prime4)/6;
    Theta_prime=Theta_prime+pas*(aTheta_prime1+2*aTheta_prime2+2*
    aTheta_prime3+aTheta_prime4)/6;
110  Phi_prime=Phi_prime+pas*(aPhi_prime1+2*aPhi_prime2+2*
    aPhi_prime3+aPhi_prime4)/6;
    Rayon=Rayon+pas*(aRayon1+2*aRayon2+2*aRayon3+aRayon4)/6;
    Theta=Theta+pas*(aTheta1+2*aTheta2+2*aTheta3+aTheta4)/6;
    Phi=Phi+pas*(aPhi1+2*aPhi2+2*aPhi3+aPhi4)/6;

115  temps = tmin +i*pas ;

    if(Rayon>rayonMax) rayonMax = Rayon ;

    resultat[i][0]= temps;

```

```

120     resultat[i][1]= getX(Rayon,Theta,Phi);
        resultat[i][2]= getY(Rayon,Theta,Phi);
        resultat[i][3]= getZ(Rayon,Theta,Phi);
        resultat[i][4]= Rayon;
125     resultat[i][5]= Theta;
        resultat[i][6]= Phi;
        resultat[i][7]= Rayon_prime;
        resultat[i][8]= Theta_prime;
        resultat[i][9]= Phi_prime;
        resultat[i][10]= aRayon_prime1; //aRayon_prime1=
            Rayon_seconde
130     resultat[i][11]= aTheta_prime1; //aTheta_prime1=
            Theta_seconde
        resultat[i][12]= aPhi_prime1; //aPhi_prime1 = Phi_seconde
    }
    System.out.println("Le tableau est rempli et le rayon max est :"+
        rayonMax);
}

135 private double Rayon_seconde(double temps, double Theta, double
    Theta_prime, double Rayon, double Rayon_prime, double Phi, double
    Phi_prime)
    {
        return Rayon*Theta_prime*Theta_prime+Rayon*Phi_prime*Phi_prime*Math
            .sin(Theta)*Math.sin(Theta)-gravite*Math.cos(Theta)-
            constanteRaideur/masse*(Rayon-longueurVide)-lambda/masse*
            Rayon_prime ;
    }

140 private double Theta_seconde(double temps, double Theta, double
    Theta_prime, double Rayon, double Rayon_prime, double Phi, double
    Phi_prime)
    {
        return (-2*masse*Rayon_prime*Theta_prime+masse*Rayon*Phi_prime*
            Phi_prime*Math.sin(Theta)*Math.cos(Theta)+masse*gravite*Math.
            sin(Theta)-lambda*Rayon*Theta_prime)/(masse*Rayon) ;
    }

145 private double Phi_seconde(double temps, double Theta, double Theta_prime,
    double Rayon, double Rayon_prime, double Phi, double Phi_prime)
    {
        return (-2*masse*Rayon_prime*Phi_prime-2*Rayon*masse*Theta_prime*
            Phi_prime/Math.tan(Theta)-lambda*Rayon*Phi_prime)/(masse*Rayon)
            ;
    }

150 public void ecriture()
    {
        double a;
        int i,j;
155     try
        {

```

```

    BufferedWriter output = new BufferedWriter( new FileWriter(
        new File("resultat.dat")));
    for(i=0;i<nombrePoints;i=i+5)
    {
        for(j=0;j<18;j++)
        {
            a = resultat[i][j];
            output.write(a+"\t");
        }
        output.newLine();
    }
    output.close();
    System.out.println("Le fichier resultat.dat est rempli.");
}
catch(IOException e){System.out.println("Erreur");}
}

//Get
public double getX (double Rayon,double Theta,double Phi)
{
    return Rayon*Math.cos(Phi)*Math.sin(Theta);
}

public double getY (double Rayon,double Theta,double Phi)
{
    return Rayon*Math.sin(Phi)*Math.sin(Theta);
}

public double getZ (double Rayon,double Theta,double Phi)
{
    return Rayon*Math.cos(Theta);
}

public double lireCase(int ligne, int colonne)
{
    return resultat[ligne][colonne];
}

public int getXEcran(double rayon,double theta,double phi,double centreX)
{
    return (int) (centreX + rayon*((Math.sqrt(3)/2)*Math.sin(Math.PI-
        theta)*Math.cos(phi) - (Math.sqrt(3)/2)*Math.sin(Math.PI-theta)
        *Math.sin(phi)));
}

public int getYEcran(double rayon,double theta,double phi, double centreY)
{
    return (int) (centreY + rayon*(1 + Math.sin(Math.PI-theta)*Math.cos(
        phi)/2 + Math.sin(Math.PI-theta)*Math.sin(phi)/2 - (1 - Math.
        cos(Math.PI-theta))));
}

public double getRayonMax ()

```

```

205     {
        return rayonMax ;
    }

    public double getNombrePoints ()
210     {
        return nombrePoints ;
    }

    public int getNombrePointsParSeconde ()
215     {
        return (int) (nombrePoints/(tmax-tmin)) ;
    }

    public int getNombreSecondes ()
220     {
        return (int) (tmax-tmin) ;
    }
}

```

C.4 Classe de représentation graphique

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.text.*;

5
public class JAffichage extends JFrame
{
    private JAnim anim = new JAnim();
    private Graphics buffer;
10    private Image feuille;
    private DecimalFormat format = new DecimalFormat();
    private int precision = 2;
    private Jressort3D ressort3D;

15    //position
    private int x=0,y=0;
    private static int largeurFenetre = 700,
        hauteurFenetre = largeurFenetre,
        centre = largeurFenetre/2;

20    //vue
    private double alpha = 30 * ( Math.PI/180 );
    private double beta = 10 * ( Math.PI/180 );
    private double rotation = 2.5 * ( Math.PI/180 );

25    private JPanel panneau = new JPanel();

    //points
30    private int a[] = new int[5];
    private int b[] = new int[5];

```

```

35 private int c[] = new int[5];
private int d[] = new int[5];
private int e[] = new int[5];
private int f[] = new int[5];
private int o[] = new int[5];
//plan
private int a_[] = new int[5];
private int b_[] = new int[5];
40 private int c_[] = new int[5];
private int d_[] = new int[5];
private int e_[] = new int[5];
private int f_[] = new int[5];
private int g_[] = new int[5];
45 private int h_[] = new int[5];

//ressort
private int delai = 10,i=1;
private double Rayon, Theta, Phi, RayonEcranMax = hauteurFenetre/2-50,
RayonEcran=0;

50 //trace
private int trace[][];
private int debutX,debutY,finX,finY,p=2,q=0;

55 public Jaffichage(Jressort3D _ressort3D)
{
//affichage
super("Vue 3D");
setBounds(770,10,largeurFenetre, hauteurFenetre);//
setDefaultCloseOperation(EXIT_ON_CLOSE);
60 setResizable(false);
panneau.setFocusable(true);
panneau.requestFocus();
ressort3D = _ressort3D ;
int w = (int) ressort3D.getNombrePoints();
65 trace= new int [w][5];

panneau.addKeyListener(new KeyListener()
{
70 public void keyPressed(KeyEvent e)
{
int keyCode = e.getKeyCode();
//fleches
if (keyCode == KeyEvent.VK_RIGHT)
{
75 sousAlpha();
}
else if (keyCode == KeyEvent.VK_LEFT)
{
addAlpha();
}
80 else if (keyCode == KeyEvent.VK_DOWN)

```

```

    {
        addBeta();
    }
    else if (keyCode == KeyEvent.VK_UP)
    {
        sousBeta();
    }
}
public void keyReleased(KeyEvent e)
{}
public void keyTyped(KeyEvent e)
{}
});

panneau.addMouseMotionListener(new MouseMotionListener()
{
    public void mouseDragged(MouseEvent e)
    {
        if (e.getX() - x > precision)
        {
            sousAlpha();
            x = e.getX();
            y = e.getY();
        }
        else if (e.getX() - x < -precision)
        {
            addAlpha();
            x = e.getX();
            y = e.getY();
        }
        else if (e.getY() - y > precision)
        {
            addBeta();
            x = e.getX();
            y = e.getY();
        }
        else if (e.getY() - y < -precision)
        {
            sousBeta();
            x = e.getX();
            y = e.getY();
        }
    }
    public void mouseMoved(MouseEvent e)
    {}
});

setContentPane(panneau);
setVisible(true);

o[0] = 0;
o[1] = 0;
o[2] = 0;

```



```
135     a[0] = -10;
        a[1] = 0;
        a[2] = 0;

140     b[0] = 100;
        b[1] = 0;
        b[2] = 0;

        c[0] = 0;
145     c[1] = -10;
        c[2] = 0;

        d[0] = 0;
        d[1] = 100;
150     d[2] = 0;

        e[0] = 0;
        e[1] = 0;
        e[2] = -10;

155     f[0] = 0;
        f[1] = 0;
        f[2] = 100;

160     a_[0] = 35;
        a_[1] = 40;
        a_[2] = 40;

        b_[0] = 45;
165     b_[1] = 40;
        b_[2] = 40;

        c_[0] = 40;
        c_[1] = 35;
170     c_[2] = 40;

        d_[0] = 40;
        d_[1] = 45;
        d_[2] = 40;

175     e_[0] = 40;
        e_[1] = 40;
        e_[2] = 35;

        f_[0] = 40;
        f_[1] = 40;
        f_[2] = 45;
        anim.start();
    }
185     private void addAlpha()
    {
        alpha += rotation;
```

```

    }
    private void sousAlpha()
190  {
        alpha -= rotation;
    }
    private void addBeta()
    {
195     beta += rotation;
    }
    private void sousBeta()
    {
200     beta -= rotation;
    }

    private void proj(int [] m)
    {
        m[3] = (int) (centre + m[0]*( -Math.sin(alpha) ) + m[1]*( Math.cos
205     (alpha) ) + m[2]*0 );
        m[4] = hauteurFenetre - (int) (centre + ( m[0]*( -Math.cos(alpha)*
        Math.sin(beta) ) + m[1]*( -Math.sin(alpha)*Math.sin(beta) ) + m
        [2]*( Math.cos(beta) ) ) );
    }

    private void projTrace(int [][] n,int r)
    {
210     int l;
        for(l=0;l<r;l++)
        {
            n[l][3] = (int) (centre + n[l][0]*( -Math.sin(alpha) ) + n[
            l][1]*( Math.cos(alpha) ) + n[l][2]*0 );
            n[l][4] = hauteurFenetre - (int) (centre + ( n[l][0]*( -
            Math.cos(alpha)*Math.sin(beta) ) + n[l][1]*( -Math.sin(
            alpha)*Math.sin(beta) ) + n[l][2]*( Math.cos(beta) ) )
            );
215     }
    }

    public void update(Graphics g)
    {
220     paint(g);
    }
    public void paint( Graphics g )
    {
        if(buffer==null)
225     {
            feuille = createImage(largeurFenetre,hauteurFenetre);
            buffer = feuille.getGraphics();
        }

        buffer.setColor(Color.black);
230     buffer.fillRect( 0 , 0 , largeurFenetre , hauteurFenetre );

        //traits
  
```

```

235     proj(o);
        proj(a);
        proj(b);
        proj(c);
        proj(d);
        proj(e);
240     proj(f);
        proj(a_);
        proj(b_);
        proj(c_);
        proj(d_);
245     proj(e_);
        proj(f_);

        //trace
        //durant le temps ecoule
250     if(i<ressort3D.getNombrePoints())
        {
            g_[0] = (int) ((RayonEcranMax-20)/ressort3D.getRayonMax()*
                ressort3D.lireCase(i,1));
            g_[1] = (int) ((RayonEcranMax-20)/ressort3D.getRayonMax()*
                ressort3D.lireCase(i,2));
            g_[2] = (int) ((RayonEcranMax-20)/ressort3D.getRayonMax()*
                ressort3D.lireCase(i,3));
255     trace[q][0]= g_[0];
            trace[q][1]= g_[1];
            trace[q][2]= g_[2];

            proj(g_);
260     projTrace(trace,(q+1));

            buffer.setColor(Color.yellow);
            buffer.drawLine(o[3], o[4], g_[3], g_[4]);

265     if(q>1)
            {
                int qk;
                for(qk=1;qk<q;qk++)
                {
270                 debutX = trace[(qk-1)][3];
                    debutY = trace[(qk-1)][4];
                    finX = trace[qk][3];
                    finY = trace[qk][4];
                    buffer.setColor(Color.green);
                    buffer.drawLine(debutX,debutY,finX,finY);
275                 }
            }
            q++;
        }

280     //apres le temps ecoule
        if(i>=ressort3D.getNombrePoints())
        {
    
```

```

proj(g_); //permet de pouvoir bouger.
285  buffer.setColor(Color.yellow);
    buffer.drawLine(o[3], o[4], g_[3], g_[4]);
    projTrace(trace,(q+1));
    if(q>0) //if(i>1)
    {
290         int qk;
        for(qk=1;qk<q;qk++)
        {
            debutX = trace[(qk-1)][3];
            debutY = trace[(qk-1)][4];
295             finX = trace[qk][3];
            finY = trace[qk][4];
            buffer.setColor(Color.green);
            buffer.drawLine(debutX,debutY,finX,finY);
        }
    }
300 }
//axes
buffer.setColor(Color.white);
buffer.drawLine(a[3], a[4], b[3], b[4]);
305 buffer.setColor(Color.blue);
buffer.drawLine(c[3], c[4], d[3], d[4]);
buffer.setColor(Color.red);
buffer.drawLine(e[3], e[4], f[3], f[4]);

310 //legende
buffer.setColor(Color.white);
buffer.drawString( "x", 32, 32+16 );
buffer.setColor(Color.blue);
buffer.drawString( "y", 32+1*16, 32+16 );
315 buffer.setColor(Color.red);
buffer.drawString( "z", 32+2*16, 32+16 );
buffer.setColor(Color.white);
format.setMaximumFractionDigits(2);
buffer.drawString( "Alpha = "+ format.format(alpha*180/Math.PI),
320 32, 32 + 2*16 );
buffer.drawString( "Beta = "+ format.format(beta*180/Math.PI), 32,
32 + 3*16 );

//finalement, le buffer memoire est dessine dans le buffer d'
affichage
g.drawImage(feuille, 0, 0, this);
i=i+(delai*ressort3D.getNombrePointsParSeconde()/1000);
325 }

public class JAnim extends Thread
{
    public void run()
    {
330         while(true)
        {
            try

```

```

335         {
            repaint();
            sleep( delai );
        }
        catch ( Exception e ) {}
    }
}

```

C.5 Classe d'analyse

```

import java.io.*;
import javax.swing.JFrame;
import javax.swing.JPanel;

5 public class JPoincare
{
    private Jressort3D ressort3D;
    private double rayonINI;
    private double    xVectW,      //composantes du vecteur normal
10         yVectW,
        zVectW,
        nVectW,      //norme du vecteur normal
        xVectU,
        yVectU,
15         zVectU,
        nVectU,
        xVectV,
        yVectV,
        zVectV,
20         nVectV,
        xPlanPoincare, // composantes du point qui caracterisent le
        plan
        yPlanPoincare,
        zPlanPoincare,
        Rayon, Rayon_prime , Rayon_seconde ,
25         Theta, Theta_prime , Theta_seconde ,
        Phi , Phi_prime , Phi_seconde ;

    private double tmin,tmax,pas;
    private double [][] pointsPP;
    private double [][] pointsEP;
30     private double [][] pointsPR;
    private int nombrePoints,j=0,maxtableau ;
    private double    xg,
        yg,
        zg,
35         xd,
        yd,
        zd;

    private double    xTampon,yTampon,zTampon ;

```

```

40     private double Xmax, Xmin, Ymax, Ymin, Zmax, Zmin;

    private int  xCoef1,
                xOperateur1,
45         xCoef2,
                xOperateur2,
                xCoef3 ,
                yCoef1,
                yOperateur1,
50         yCoef2,
                yOperateur2,
                yCoef3 ,
                zCoef1,
                zOperateur1,
55         zCoef2,
                zOperateur2,
                zCoef3 ;

    public JPoincare (
60         Jressort3D _ressort3D,
                double _tmin,
                double _tmax,
                double _rayonINI
        )
65     {
        ressort3D = _ressort3D;
        tmin = _tmin ;
        tmax = _tmax ;
        rayonINI = _rayonINI;
70         nombrePoints = (int) ressort3D.getNombrePoints();
    }

    public void calculEspace(
75         double _xCoef1,
                double _xOperateur1,
                double _xCoef2,
                double _xOperateur2,
                double _xCoef3,
                double _yCoef1,
80         double _yOperateur1,
                double _yCoef2,
                double _yOperateur2,
                double _yCoef3,
                double _zCoef1,
85         double _zOperateur1,
                double _zCoef2,
                double _zOperateur2,
                double _zCoef3
        )
90     {
        xCoef1 = (int) (_xCoef1);
        xOperateur1 = (int) (_xOperateur1);
    }

```

```

95     xCoef2 = (int) (_xCoef2);
        xOperateur2 = (int) (_xOperateur2);
        xCoef3 = (int) (_xCoef3);

        yCoef1 = (int) (_yCoef1);
        yOperateur1 = (int) (_yOperateur1);
100     yCoef2 = (int) (_yCoef2);
        yOperateur2 = (int) (_yOperateur2);
        yCoef3 = (int) (_yCoef3);

        zCoef1 = (int) (_zCoef1);
        zOperateur1 = (int) (_zOperateur1);
105     zCoef2 = (int) (_zCoef2);
        zOperateur2 = (int) (_zOperateur2);
        zCoef3 = (int) (_zCoef3);

        int i;

110     pointsEP = new double [nombrePoints+1][4];

        for(i=0;i<nombrePoints;i++)
        {
115             Rayon = ressort3D.lireCase(i,4);
                Rayon_prime = ressort3D.lireCase(i,7);
                Rayon_seconde = ressort3D.lireCase(i,10);
                Theta = ressort3D.lireCase(i,5);
                Theta_prime = ressort3D.lireCase(i,8);
120             Theta_seconde = ressort3D.lireCase(i,11);
                Phi = ressort3D.lireCase(i,6);
                Phi_prime = ressort3D.lireCase(i,9);
                Phi_seconde = ressort3D.lireCase(i,12);

125             xTampon = 1;
                yTampon = 1;
                zTampon = 1;

130             switch(xCoef1)
            {
                case 0 :
                    xTampon = xTampon*1;
                    break;
135                 case 1 :
                    xTampon = xTampon*Rayon;
                    break;
                case 2 :
                    xTampon = xTampon*Rayon_prime;
140                 break;
                case 3 :
                    xTampon = xTampon*Rayon_seconde;
                    break;
                case 4 :
145                 xTampon = xTampon*Theta;
            }
        }
    }

```

```

        break;
        case 5 :
            xTampon = xTampon*Theta_prime;
        break;
150     case 6 :
            xTampon = xTampon*Theta_seconde;
        break;
        case 7 :
            xTampon = xTampon*Phi;
155     break;
        case 8 :
            xTampon = xTampon*Phi_prime;
        break;
        case 9 :
            xTampon = xTampon*Phi_seconde;
160     break;
        default:
            System.out.println("Erreur");
        break;
165     }
    }
    if(xOperateur1 == 0 )
    {
170         switch(xCoef2)
        {
            case 0 :
                xTampon = xTampon*1;
            break;
            case 1 :
                xTampon = xTampon*Rayon;
175             break;
            case 2 :
                xTampon = xTampon*Rayon_prime;
            break;
            case 3 :
                xTampon = xTampon*Rayon_seconde;
180             break;
            case 4 :
                xTampon = xTampon*Theta;
            break;
            case 5 :
                xTampon = xTampon*Theta_prime;
185             break;
            case 6 :
                xTampon = xTampon*Theta_seconde;
190             break;
            case 7 :
                xTampon = xTampon*Phi;
            break;
            case 8 :
                xTampon = xTampon*Phi_prime;
195             break;
            case 9 :

```



```

        xTampon = xTampon*Phi_seconde;
    200     break;
        default:
            System.out.println("Erreur");
        break;
    }
    }
    else
    {
        switch(xCoef2)
        {
    210     case 0 :
            xTampon = xTampon/1;
        break;
        case 1 :
            xTampon = xTampon/Rayon;
    215     break;
        case 2 :
            xTampon = xTampon/Rayon_prime;
        break;
        case 3 :
            xTampon = xTampon/Rayon_seconde;
    220     break;
        case 4 :
            xTampon = xTampon/Theta;
        break;
        case 5 :
            xTampon = xTampon/Theta_prime;
    225     break;
        case 6 :
            xTampon = xTampon/Theta_seconde;
    230     break;
        case 7 :
            xTampon = xTampon/Phi;
        break;
        case 8 :
            xTampon = xTampon/Phi_prime;
    235     break;
        case 9 :
            xTampon = xTampon/Phi_seconde;
        break;
        default:
            System.out.println("Erreur");
        break;
    }
    }
    if(xOperateur2 == 0 )
    {
        switch(xCoef3)
        {
    250     case 0 :
            xTampon = xTampon*1;
        break;

```

```

255         case 1 :
                xTampon = xTampon*Rayon;
            break;
        case 2 :
                xTampon = xTampon*Rayon_prime;
            break;
260         case 3 :
                xTampon = xTampon*Rayon_seconde;
            break;
        case 4 :
                xTampon = xTampon*Theta;
            break;
265         case 5 :
                xTampon = xTampon*Theta_prime;
            break;
        case 6 :
                xTampon = xTampon*Theta_seconde;
            break;
270         case 7 :
                xTampon = xTampon*Phi;
            break;
        case 8 :
                xTampon = xTampon*Phi_prime;
            break;
275         case 9 :
                xTampon = xTampon*Phi_seconde;
            break;
        default:
            System.out.println("Erreur");
            break;
    }
}
285 else
{
    switch(xCoef3)
    {
290         case 0 :
                xTampon = xTampon/1;
            break;
        case 1 :
                xTampon = xTampon/Rayon;
            break;
295         case 2 :
                xTampon = xTampon/Rayon_prime;
            break;
        case 3 :
                xTampon = xTampon/Rayon_seconde;
            break;
300         case 4 :
                xTampon = xTampon/Theta;
            break;
        case 5 :
                xTampon = xTampon/Theta_prime;
    }
}

```

```

305         break;
           case 6 :
               xTampon = xTampon/Theta_seconde;
           break;
           case 7 :
               xTampon = xTampon/Phi;
           break;
           case 8 :
               xTampon = xTampon/Phi_prime;
           break;
315         case 9 :
               xTampon = xTampon/Phi_seconde;
           break;
           default:
               System.out.println("Erreur");
           break;
320     }
    }
// -----
325     switch(yCoef1)
    {
           case 0 :
               yTampon = yTampon*1;
           break;
           case 1 :
330               yTampon = yTampon*Rayon;
           break;
           case 2 :
               yTampon = yTampon*Rayon_prime;
           break;
335         case 3 :
               yTampon = yTampon*Rayon_seconde;
           break;
           case 4 :
               yTampon = yTampon*Theta;
340           break;
           case 5 :
               yTampon = yTampon*Theta_prime;
           break;
           case 6 :
345               yTampon = yTampon*Theta_seconde;
           break;
           case 7 :
               yTampon = yTampon*Phi;
           break;
           case 8 :
350               yTampon = yTampon*Phi_prime;
           break;
           case 9 :
               yTampon = yTampon*Phi_seconde;
355           break;
           default:
               System.out.println("Erreur");
    }
  
```

```

    }
    break;
  }
360
  if(yOperateur1 == 0 )
  {
    switch(yCoef2)
    {
365      case 0 :
          yTampon = yTampon*1;
          break;
      case 1 :
          yTampon = yTampon*Rayon;
370          break;
      case 2 :
          yTampon = yTampon*Rayon_prime;
          break;
      case 3 :
375          yTampon = yTampon*Rayon_seconde;
          break;
      case 4 :
          yTampon = yTampon*Theta;
          break;
380          case 5 :
              yTampon = yTampon*Theta_prime;
              break;
          case 6 :
              yTampon = yTampon*Theta_seconde;
385              break;
          case 7 :
              yTampon = yTampon*Phi;
              break;
          case 8 :
              yTampon = yTampon*Phi_prime;
390              break;
          case 9 :
              yTampon = yTampon*Phi_seconde;
              break;
395          default:
              System.out.println("Erreur");
              break;
    }
  }
400  else
  {
    switch(yCoef2)
    {
405      case 0 :
          yTampon = yTampon/1;
          break;
      case 1 :
          yTampon = yTampon/Rayon;
          break;
410      case 2 :

```

```

        yTampon = yTampon/Rayon_prime;
    break;
    case 3 :
        yTampon = yTampon/Rayon_seconde;
415 break;
    case 4 :
        yTampon = yTampon/Theta;
    break;
    case 5 :
420 yTampon = yTampon/Theta_prime;
    break;
    case 6 :
        yTampon = yTampon/Theta_seconde;
    break;
425 case 7 :
        yTampon = yTampon/Phi;
    break;
    case 8 :
        yTampon = yTampon/Phi_prime;
430 break;
    case 9 :
        yTampon = yTampon/Phi_seconde;
    break;
    default:
435 System.out.println("Erreur");
    break;
    }
}
440 if(yOperateur2 == 0 )
{
    switch(yCoef3)
    {
        case 0 :
445 yTampon = yTampon*1;
        break;
        case 1 :
            yTampon = yTampon*Rayon;
        break;
        case 2 :
450 yTampon = yTampon*Rayon_prime;
        break;
        case 3 :
            yTampon = yTampon*Rayon_seconde;
        break;
455 case 4 :
            yTampon = yTampon*Theta;
        break;
        case 5 :
            yTampon = yTampon*Theta_prime;
        break;
460 case 6 :
            yTampon = yTampon*Theta_seconde;
        break;
    }
}

```

```

465         case 7 :
            yTampon = yTampon*Phi;
        break;
        case 8 :
            yTampon = yTampon*Phi_prime;
        break;
470     case 9 :
            yTampon = yTampon*Phi_seconde;
        break;
        default:
            System.out.println("Erreur");
475     break;
    }
}
else
{
480     switch(yCoef3)
    {
        case 0 :
            yTampon = yTampon/1;
        break;
485     case 1 :
            yTampon = yTampon/Rayon;
        break;
        case 2 :
            yTampon = yTampon/Rayon_prime;
490     break;
        case 3 :
            yTampon = yTampon/Rayon_seconde;
        break;
        case 4 :
            yTampon = yTampon/Theta;
495     break;
        case 5 :
            yTampon = yTampon/Theta_prime;
        break;
500     case 6 :
            yTampon = yTampon/Theta_seconde;
        break;
        case 7 :
            yTampon = yTampon/Phi;
505     break;
        case 8 :
            yTampon = yTampon/Phi_prime;
        break;
        case 9 :
            yTampon = yTampon/Phi_seconde;
510     break;
        default:
            System.out.println("Erreur");
        break;
515     }
    }
}

```

```
//-----  
    switch (zCoef1)  
    {  
520         case 0 :  
            zTampon = zTampon*1;  
            break;  
         case 1 :  
            zTampon = zTampon*Rayon;  
525         break;  
         case 2 :  
            zTampon = zTampon*Rayon_prime;  
            break;  
         case 3 :  
530             zTampon = zTampon*Rayon_seconde;  
            break;  
         case 4 :  
            zTampon = zTampon*Theta;  
            break;  
535         case 5 :  
            zTampon = zTampon*Theta_prime;  
            break;  
         case 6 :  
            zTampon = zTampon*Theta_seconde;  
540         break;  
         case 7 :  
            zTampon = zTampon*Phi;  
            break;  
         case 8 :  
545             zTampon = zTampon*Phi_prime;  
            break;  
         case 9 :  
            zTampon = zTampon*Phi_seconde;  
550         break;  
         default:  
            System.out.println("Erreur");  
            break;  
    }  
  
555     if (zOperateur1 == 0 )  
    {  
        switch (zCoef2)  
        {  
560             case 0 :  
                zTampon = zTampon*1;  
                break;  
             case 1 :  
                zTampon = zTampon*Rayon;  
                break;  
565             case 2 :  
                zTampon = zTampon*Rayon_prime;  
                break;  
             case 3 :  
                zTampon = zTampon*Rayon_seconde;
```

```

570         break;
        case 4 :
            zTampon = zTampon*Theta;
        break;
        case 5 :
575             zTampon = zTampon*Theta_prime;
        break;
        case 6 :
            zTampon = zTampon*Theta_seconde;
        break;
580         case 7 :
            zTampon = zTampon*Phi;
        break;
        case 8 :
            zTampon = zTampon*Phi_prime;
585         break;
        case 9 :
            zTampon = zTampon*Phi_seconde;
        break;
        default:
590             System.out.println("Erreur");
        break;
    }
}
}
else
595 {
    switch( zCoef2)
    {
        case 0 :
            zTampon = zTampon/1;
600         break;
        case 1 :
            zTampon = zTampon/Rayon;
        break;
        case 2 :
605             zTampon = zTampon/Rayon_prime;
        break;
        case 3 :
            zTampon = zTampon/Rayon_seconde;
        break;
610         case 4 :
            zTampon = zTampon/Theta;
        break;
        case 5 :
            zTampon = zTampon/Theta_prime;
615         break;
        case 6 :
            zTampon = zTampon/Theta_seconde;
        break;
        case 7 :
620             zTampon = zTampon/Phi;
        break;
        case 8 :
    
```



```

        zTampon = zTampon/Phi_prime;
        break;
625     case 9 :
        zTampon = zTampon/Phi_seconde;
        break;
        default:
        System.out.println( "Erreur" );
630     break;
    }
}
if(zOperateur2 == 0 )
{
635     switch( zCoef3)
    {
        case 0 :
        zTampon = zTampon*1;
        break;
640     case 1 :
        zTampon = zTampon*Rayon;
        break;
        case 2 :
        zTampon = zTampon*Rayon_prime;
645     break;
        case 3 :
        zTampon = zTampon*Rayon_seconde;
        break;
        case 4 :
650     zTampon = zTampon*Theta;
        break;
        case 5 :
        zTampon = zTampon*Theta_prime;
        break;
655     case 6 :
        zTampon = zTampon*Theta_seconde;
        break;
        case 7 :
        zTampon = zTampon*Phi;
660     break;
        case 8 :
        zTampon = zTampon*Phi_prime;
        break;
        case 9 :
665     zTampon = zTampon*Phi_seconde;
        break;
        default:
        System.out.println( "Erreur" );
        break;
670     }
    }
else
{
    switch( zCoef3)
675     {

```

```

        case 0 :
            zTampon = zTampon/1;
        break;
        case 1 :
            zTampon = zTampon/Rayon;
        break;
        case 2 :
            zTampon = zTampon/Rayon_prime;
        break;
        case 3 :
            zTampon = zTampon/Rayon_seconde;
        break;
        case 4 :
            zTampon = zTampon/Theta;
        break;
        case 5 :
            zTampon = zTampon/Theta_prime;
        break;
        case 6 :
            zTampon = zTampon/Theta_seconde;
        break;
        case 7 :
            zTampon = zTampon/Phi;
        break;
        case 8 :
            zTampon = zTampon/Phi_prime;
        break;
        case 9 :
            zTampon = zTampon/Phi_seconde;
        break;
        default:
            System.out.println("Erreur");
        break;
    }
}

pointsEP[i][0] = xTampon ;
pointsEP[i][1] = yTampon ;
pointsEP[i][2] = zTampon ;

}

Xmax=pointsEP[0][0];
Xmin=pointsEP[0][0];
Ymax=pointsEP[0][1];
Ymin=pointsEP[0][1];
Zmax=pointsEP[0][2];
Zmin=pointsEP[0][2];
for(i=1;i<nombrePoints;i++)
{
    if(pointsEP[i][0]<Xmin)
        Xmin = pointsEP[i][0];
    if(pointsEP[i][0]>Xmax)
        Xmax = pointsEP[i][0];

```

```

730         if(pointsEP[i][1]<Ymin)
            Ymin = pointsEP[i][1];
        if(pointsEP[i][1]>Ymax)
            Ymax = pointsEP[i][1];
        if(pointsEP[i][2]<Zmin)
            Zmin = pointsEP[i][2];
735         if(pointsEP[i][2]>Zmax)
            Zmax = pointsEP[i][2];
    }
}

740 public void calculPP (
        double _xVect,
        double _yVect,
        double _zVect,
        double _xPlanPoincare,
745         double _yPlanPoincare,
        double _zPlanPoincare
    )
{
    xVectW=_xVect;
750     yVectW= _yVect;
    zVectW= _zVect;
    xPlanPoincare= _xPlanPoincare;
    yPlanPoincare= _yPlanPoincare;
755     zPlanPoincare= _zPlanPoincare;

    double coef,a;
    int i;

    pointsPP = new double [nombrePoints+2][6];

760     for(i=1;i<nombrePoints;i++)
    {
        xg = pointsEP[i-1][0];
        xd = pointsEP[i][0];
765         yg = pointsEP[i-1][1];
        yd = pointsEP[i][1];
        zg = pointsEP[i-1][2];
        zd = pointsEP[i][2];

770         nVectW=Math.sqrt (xVectW*xVectW+yVectW*yVectW+zVectW*zVectW) ;
        a = 0.0000001;
        if (nVectW < a)
        {
            xVectW=xVectW/a ;
775             yVectW=yVectW/a ;
            zVectW=zVectW/a ;
        }
        else
        {
780             xVectW=xVectW/nVectW ;
            yVectW=yVectW/nVectW ;

```

```

        zVectW=zVectW/nVectW ;
    }
    if ((plan(xVectW,yVectW,zVectW,xPlanPoincare,yPlanPoincare,
    zPlanPoincare,xg,yg,zg)<0) && (plan(xVectW,yVectW,zVectW,
785   xPlanPoincare,yPlanPoincare,zPlanPoincare,xd,yd,zd)>0))
    {
        coef = coefColineaire(xVectW,yVectW,zVectW,xPlanPoincare,
            yPlanPoincare,zPlanPoincare,xg,yg,zg,xd,yd,zd) ;
        pointsPP[j][0] = coef * (xd - xg)+ xg; // x du plan en 3D
        pointsPP[j][1] = coef * (yd - yg)+ yg; // y du plan en 3D
        pointsPP[j][2] = coef * (zd - zg)+ zg; // z du plan en 3D
790
        if(j==2)
        {
            xVectV=pointsPP[1][0] - pointsPP[0][0];
            yVectV=pointsPP[1][1] - pointsPP[0][1];
795   yVectV=pointsPP[1][2] - pointsPP[0][2];
            nVectV=Math.sqrt(xVectV*xVectV+yVectV*yVectV+zVectV*
                zVectV);
            xVectV=xVectV/nVectV ;
            yVectV=yVectV/nVectV ;
            zVectV=zVectV/nVectV ;

            xVectU=zVectW*yVectV-yVectW*zVectV;
            yVectU=zVectW*xVectV-xVectW*zVectV;
            yVectU=yVectW*xVectV-xVectW*yVectV;
            nVectU=Math.sqrt(xVectU*xVectU+yVectU*yVectU+zVectU*
                zVectU);
800   xVectU=xVectU/nVectU ;
            yVectU=yVectU/nVectU ;
            zVectU=zVectU/nVectU ;
        }
805
        if (j > 1)
        {
            pointsPP[j][4] = deuxDimensionY(xVectU,yVectU,zVectU,
                xVectV,yVectV,zVectV,xPlanPoincare,yPlanPoincare,
                zPlanPoincare,pointsPP[j][0],pointsPP[j][1],
                pointsPP[j][2]) ;
            pointsPP[j][3] = deuxDimensionX(pointsPP[j][4],xVectU,
                yVectU,zVectU,xVectV,yVectV,zVectV,xPlanPoincare,
                yPlanPoincare,zPlanPoincare,pointsPP[j][0],pointsPP
                [j][1],pointsPP[j][2]) ;
810
        }
        j++;
    }
    maxtableau = j;
815
}
System.out.println("Le tableau est rempli.");
820
}

public void premierRetourDuJedi()

```

```

825     {
        int i;
        pointsPR = new double [nombrePoints+1][2];
        for (i=0;i<maxtableau-2;i++)
            {
                pointsPR[i][0]= pointsPP[i][0];
830                pointsPR[i][1]= pointsPP[i+1][0];
            }
    }

    private double plan (double vectNormalX,double vectNormalY,double
        vectNormalZ,double xa,double ya,double za,double x,double y,double z)
835    {
        return vectNormalX*(x-xa)+vectNormalY*(y-ya)+vectNormalZ*(z-za);
    }

    private double coefColineaire (double vectNormalX,double vectNormalY,
        double vectNormalZ,double xa,double ya,double za,double xga,double yga
        ,double zga,double xdr,double ydr,double zdr)
840    {
        return (vectNormalX*(xa-xga)+vectNormalY*(ya-yga)+vectNormalZ*(za-
            zga))/(vectNormalX*(xdr-xga)+vectNormalY*(ydr-yga)+vectNormalZ
            *(zdr-zga));
    }

    private double deuxDimensionX(double yPrime,double xu,double yu,double zu,
        double xv,double yv,double zv,double xa,double ya,double za,double x,
        double y,double z)
845    {
        return ((x-xa)-yPrime*xv)/xu;
    }

    private double deuxDimensionY(double xu,double yu,double zu,double xv,
        double yv,double zv,double xa,double ya,double za,double x,double y,
        double z)
850    {
        return ((y-ya)*xu+yu*(xa-x))/(yv*xu-xv*yu);
    }

    public void ecritureEspace()
855    {
        double a;
        int i,j;
        try
        {
860            BufferedWriter output = new BufferedWriter( new FileWriter(
                new File("espace.dat")));
            for(i=0;i<nombrePoints;i=i+10)
            {
                for(j=0;j<3;j++)
                {
865                    a = pointsEP[i][j];
                    output.write(a+"\t");
                }
            }
        }
    }

```

```

        }
        output.newLine();
    }
    output.close();
    System.out.println("Le fichier espace.dat est rempli.");
}
catch(IOException e){System.out.println("Erreur");}
}

875 public void ecriturePP()
    {
        double a;
        int i,j;
880 try
        {
            BufferedWriter output = new BufferedWriter( new FileWriter(
                new File("poincare"+rayonINI+".dat")));
            for(i=3;i<maxtableau;i++)
            {
885 a = rayonINI;
                output.write(a+"\t");
                a = pointsPP[i][0];
                output.write(a+"\t");
                a = pointsPP[i][2];
890 output.write(a+"\t");
                output.newLine();
            }
            output.close();

895 BufferedWriter output2 = new BufferedWriter (new FileWriter("
                poincare.p",true));
            output2.write("\t'poincare"+rayonINI+".dat' pointtype 0 title
                '"+rayonINI+"', \\");
            output2.newLine();
            output2.close();
            System.out.println("poincare"+rayonINI+".dat est rempli et le
                fichier de tracer mis a jour");
900 }
        catch(IOException e){System.out.println("Erreur");}
    }

905 public void ecriturePR()
    {
        double a;
        int i,j;
        try
        {
910 BufferedWriter output = new BufferedWriter( new FileWriter(
                new File("premierRetour.dat")));
            for(i=0;i<maxtableau-1;i++)
            {
                for(j=0;j<2;j++)
                {

```

```
915         a = pointsPR[i][j];
           output.write(a+"\t");
           }
           output.newLine();
           }
920     output.close();
           System.out.println("Le fichier premierRetour.dat est rempli.");
           ;
           }
           catch(IOException e){System.out.println("Erreur");}
           }
925
public double lireCase(int ligne, int colonne)
{
    return pointsEP[ligne][colonne];
}
930
public double getCentreX ()
{
    return Xmax-Xmin ;
}
935
public double getCentreY ()
{
    return Ymax-Ymin ;
}
public double getCentreZ ()
940
{
    return Zmax-Zmin ;
}
public double getRayonMax ()
{
945     return Math.sqrt((Xmax-Xmin)*(Xmax-Xmin)/4+(Ymax-Ymin)*(Ymax-Ymin)
        /4+(Zmax-Zmin)*(Zmax-Zmin)/4);
}
}
```

Annexe D

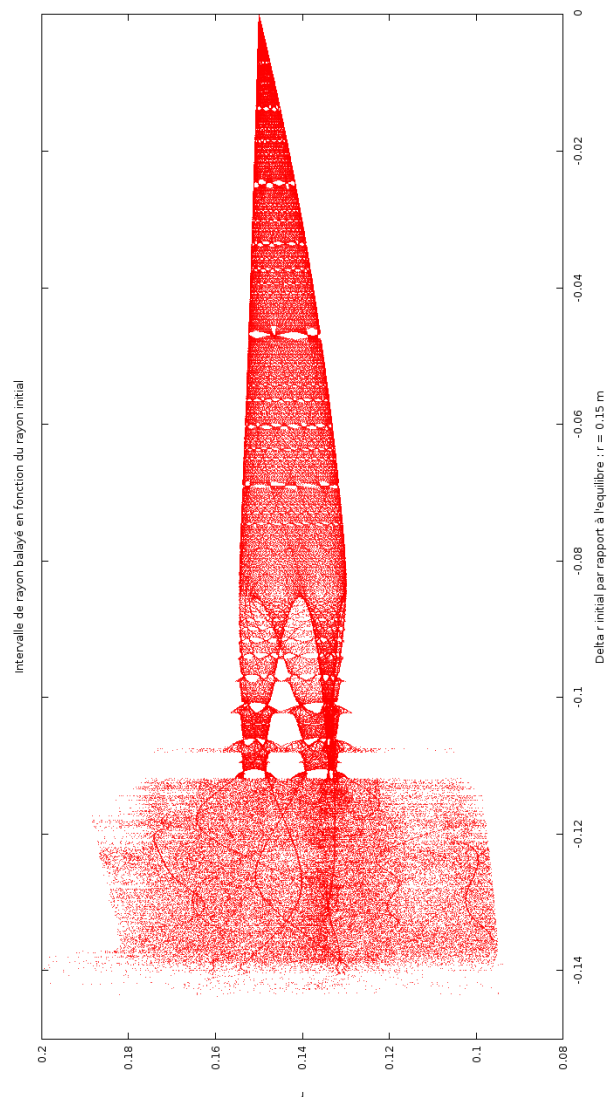
Code de Scilab

```
Mat=fscanfMat('resultat.dat'); //Recuperation des donnees sous forme
    matricielle
Taille=size(Mat);
Nbpoints=Taille(1);
Duree=Mat($,1);
5
//Analyse de Fourier
sample_rate=(Nbpoints-1)/Duree;
y=fft(Mat(:,5));
f=sample_rate*(0:(Nbpoints/2))/Nbpoints; //Vecteur associe a la frequence
10 n=size(f, '*');
plot2d(f,abs(y(1:n)))
xtitle('Spectre de la variation de r(t)', 'frequences (Hz)', 'Amplitude')
```

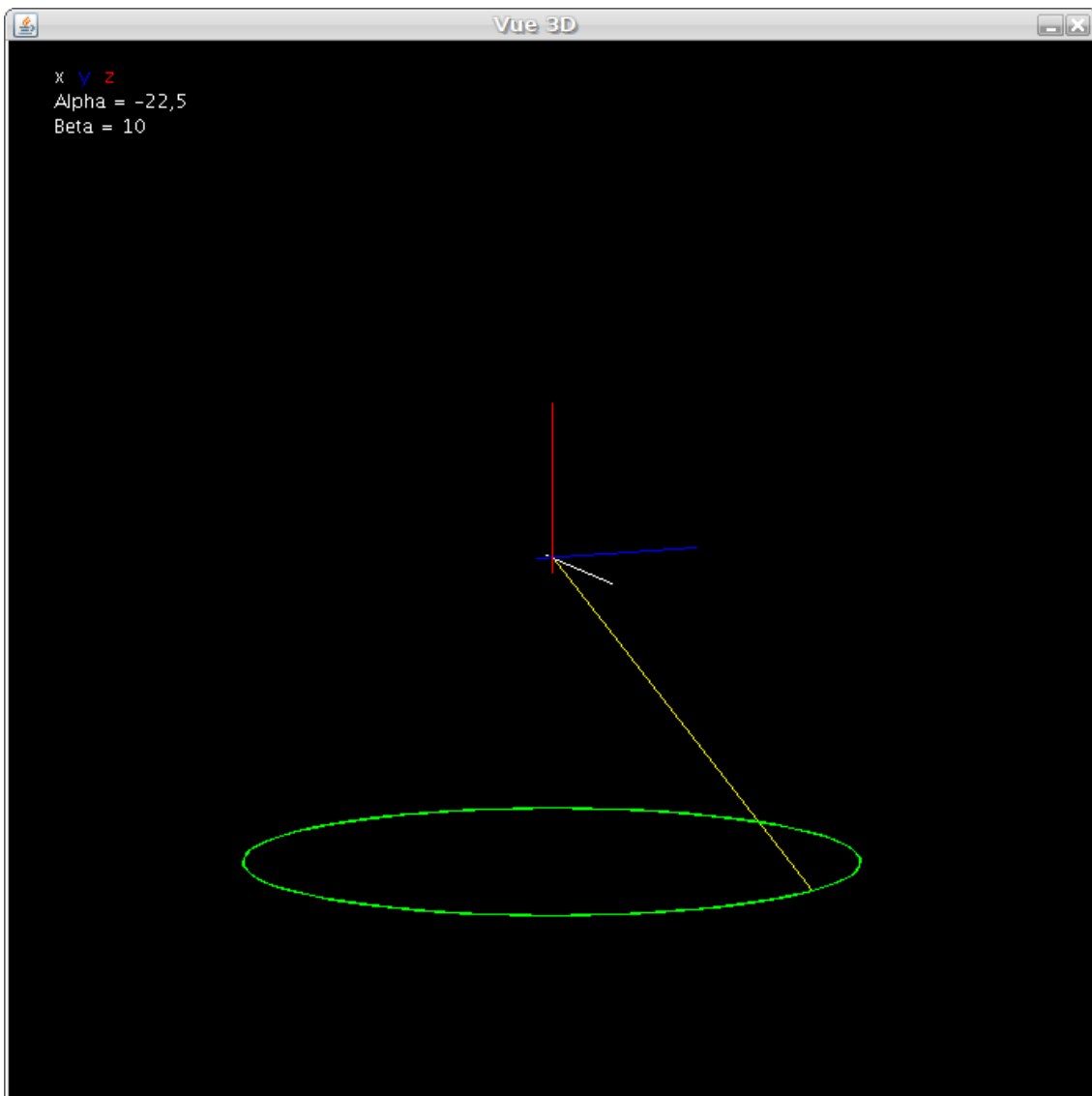

Annexe E

Graphiques et Analyse

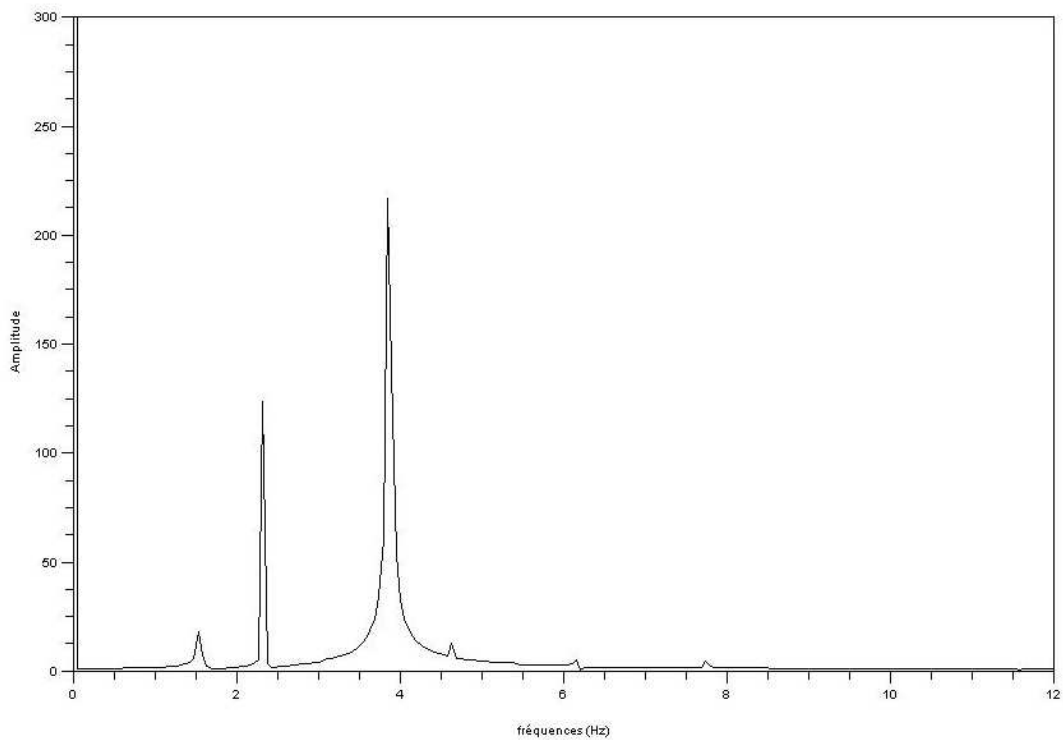
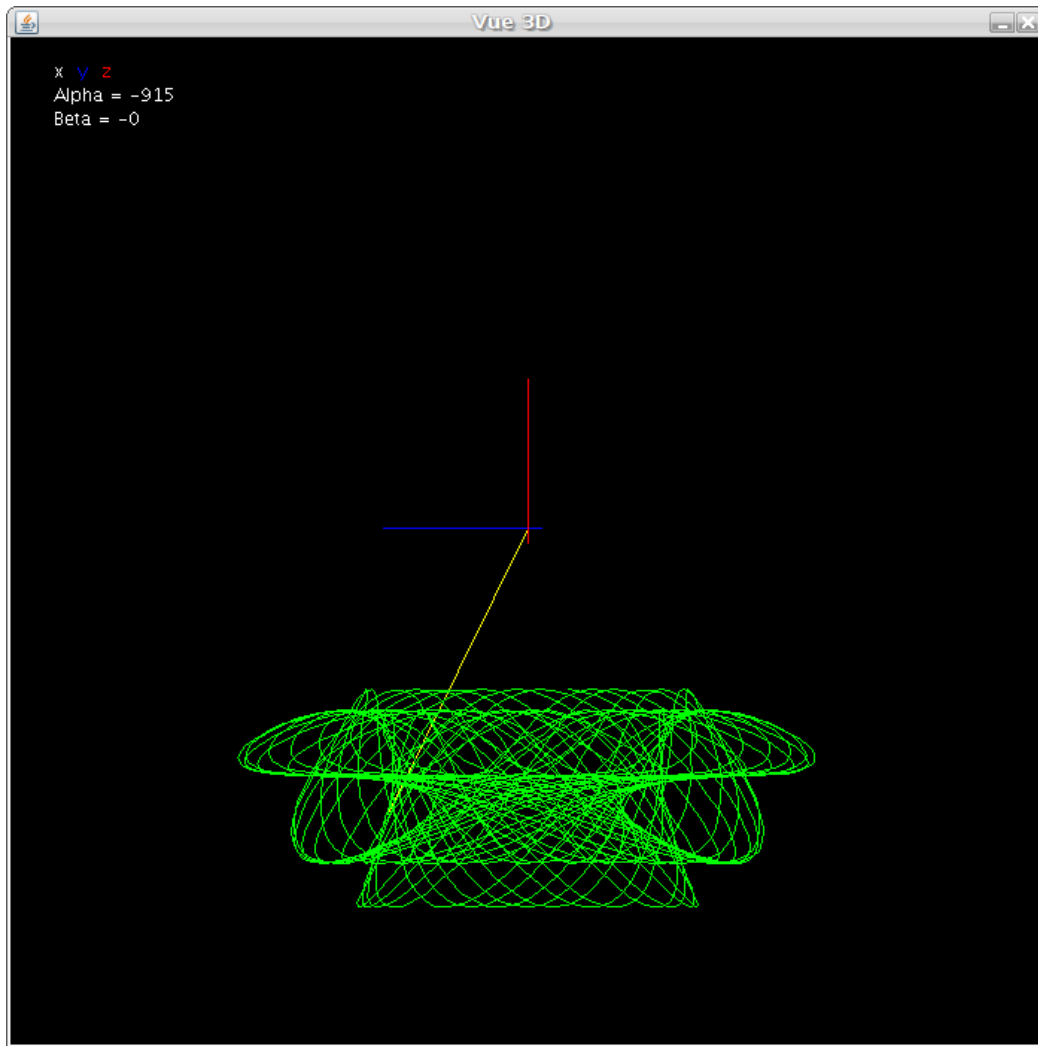
E.1 Evolution du rayon en fonction de l'allongement par rapport à l'équilibre



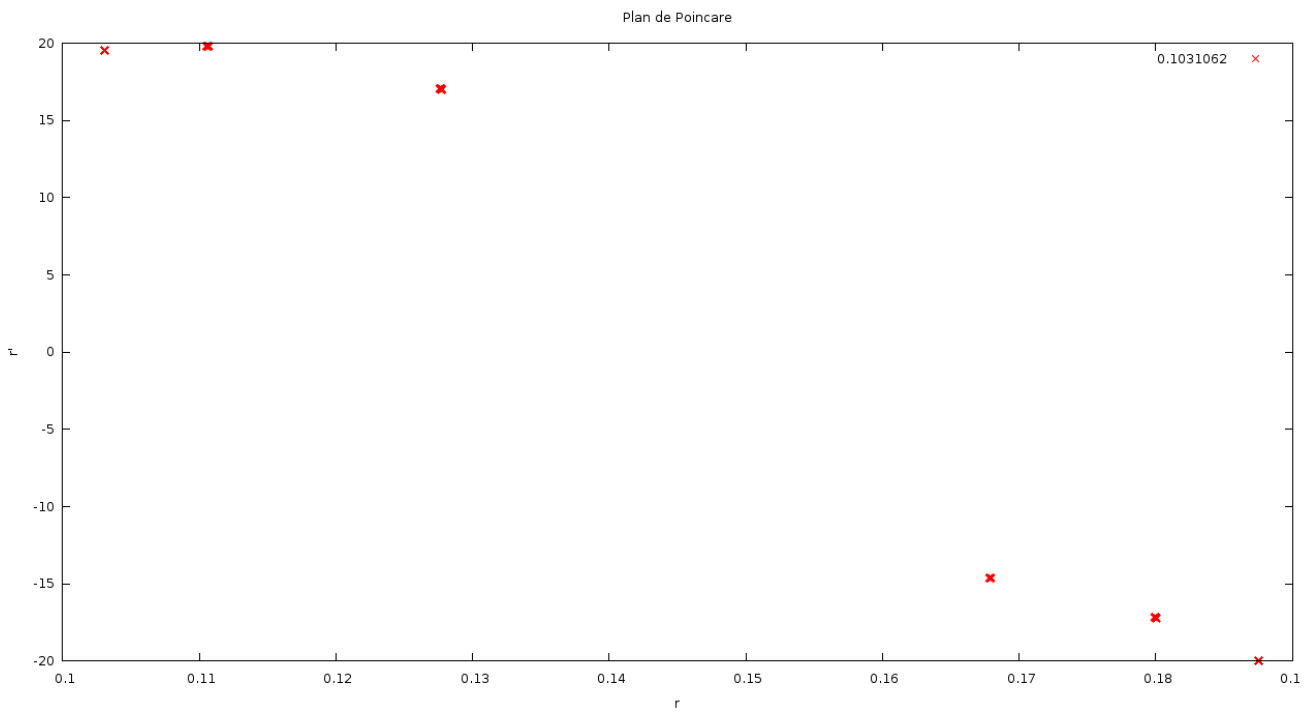
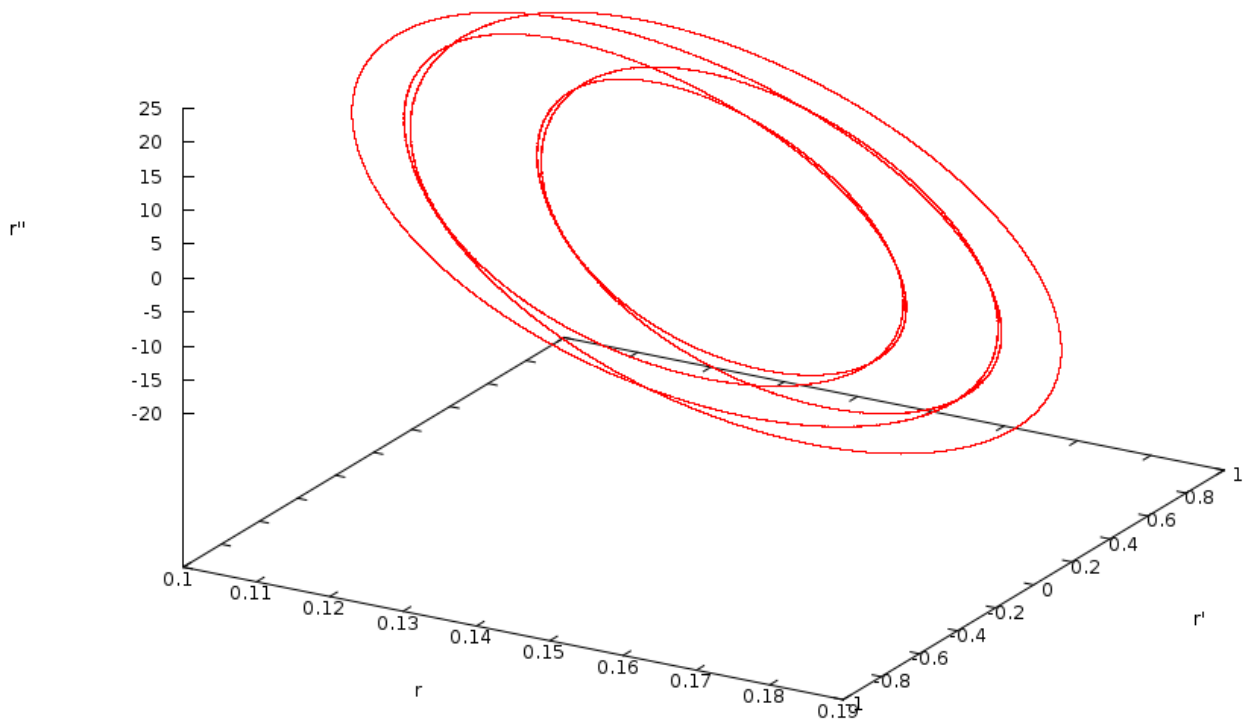
E.2 Etude de l'état d'équilibre



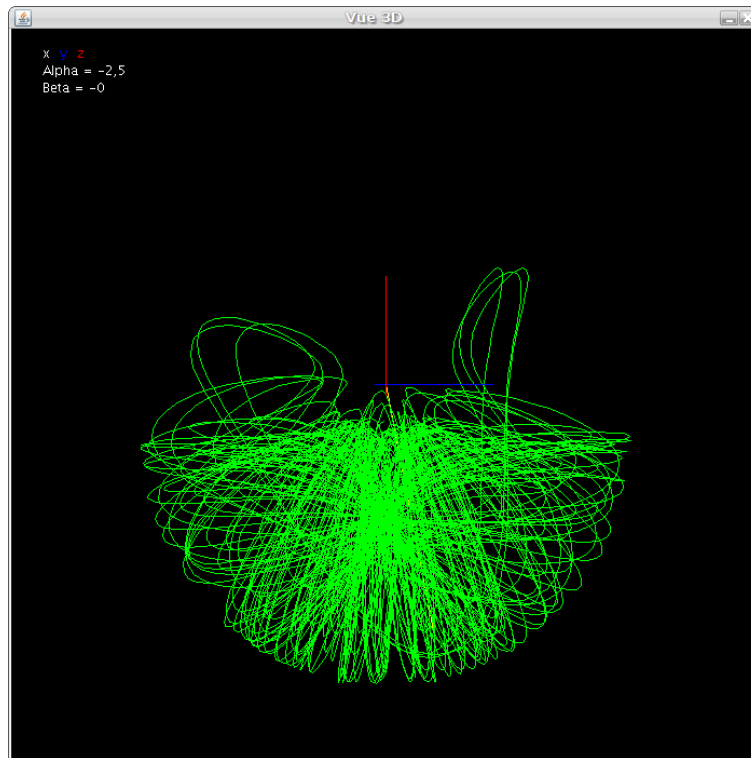
E.3 Etude de l'état quasi-périodique



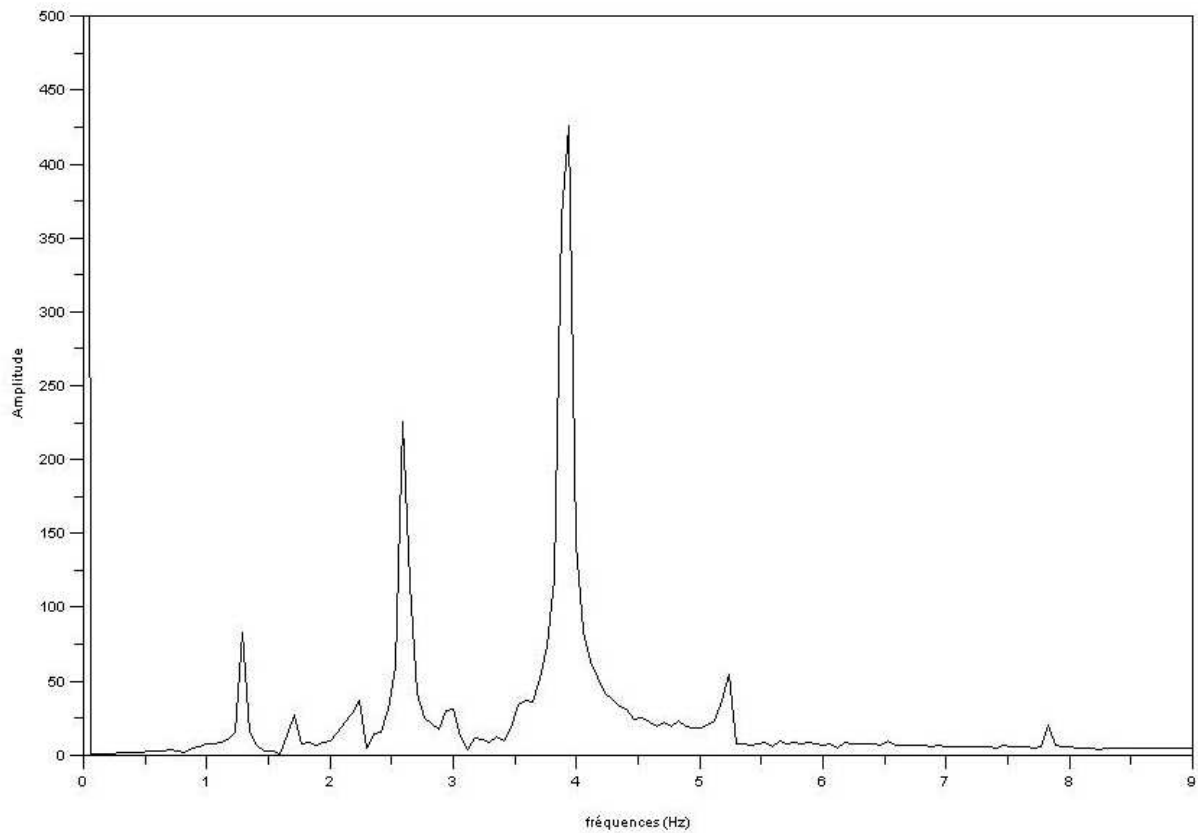
Espace des phases



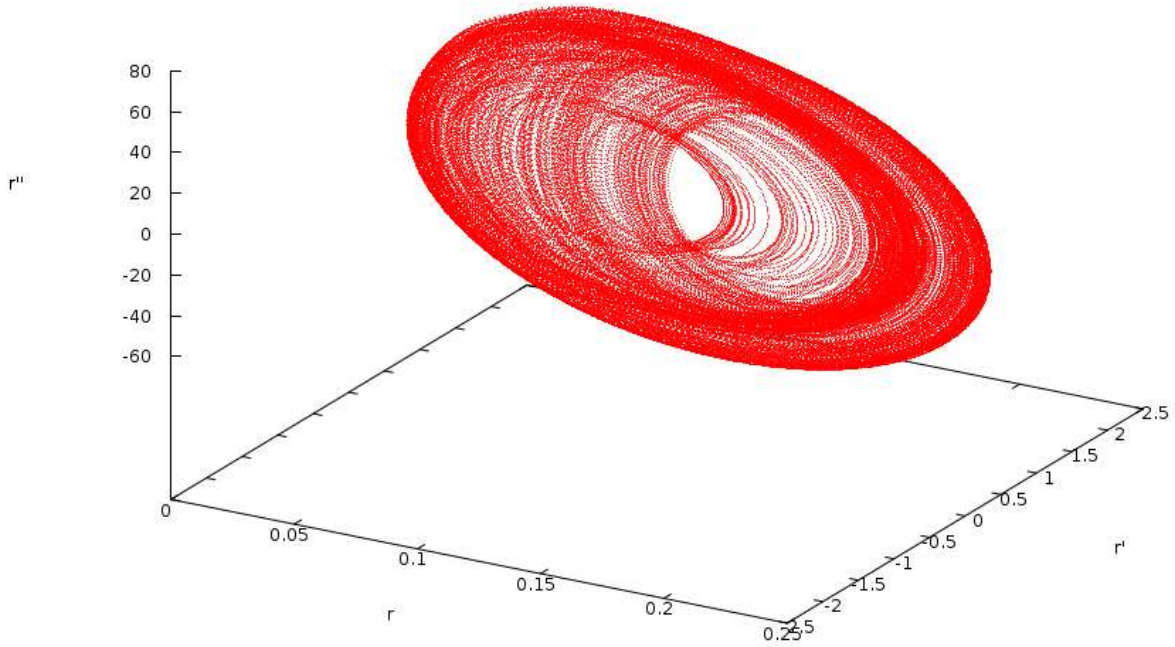
E.4 Etude de la bouffée chaotique



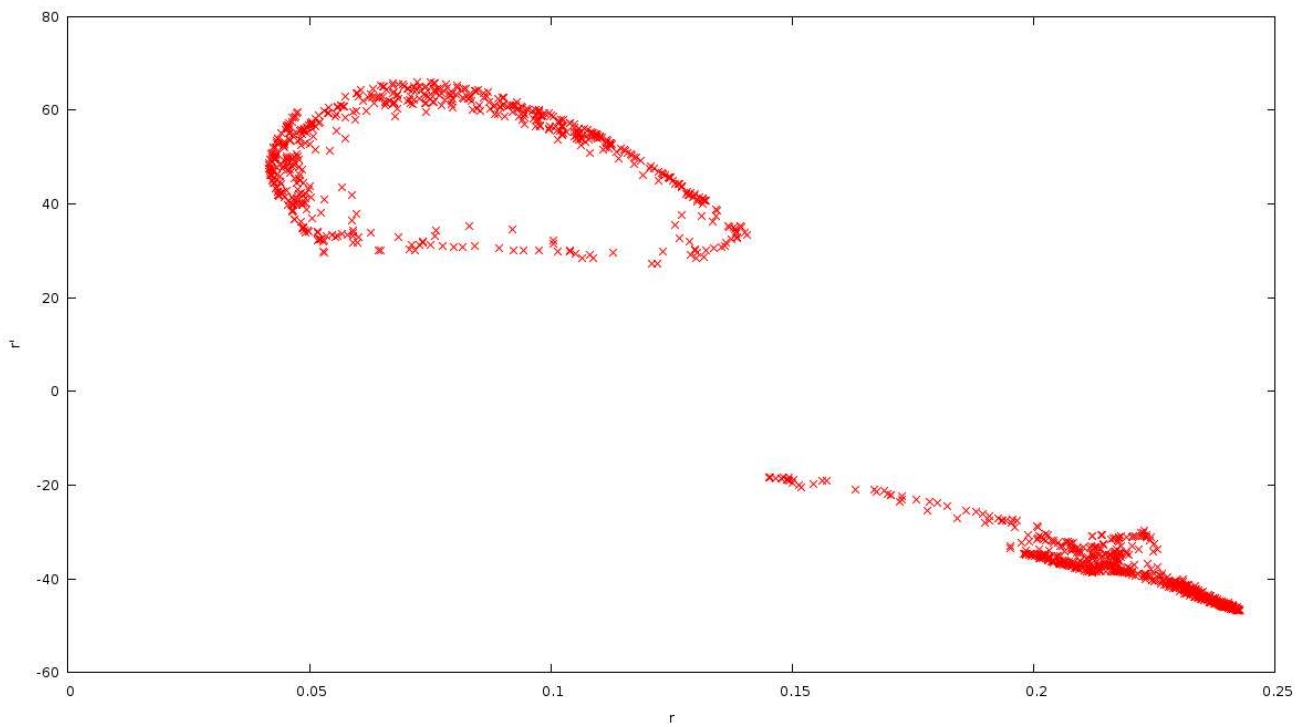
Spectre de la variation de $r(t)$ pour $\tau = 0.0422$



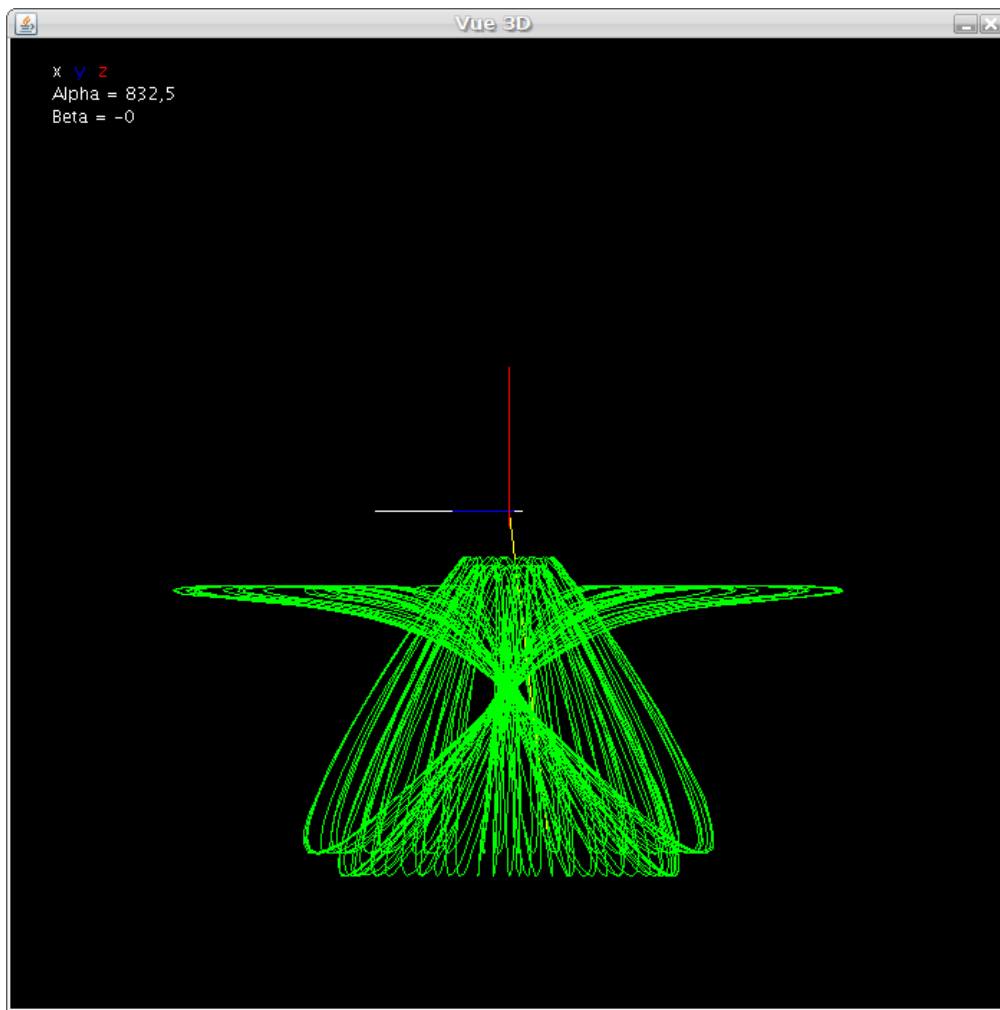
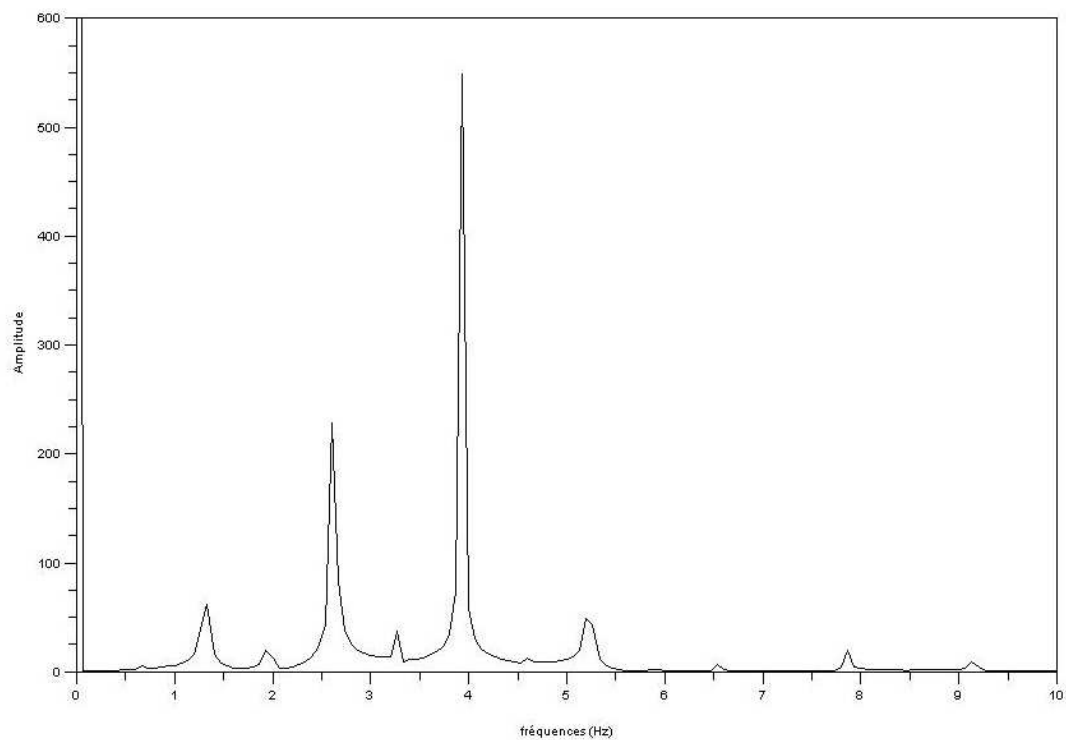
Espace des phases



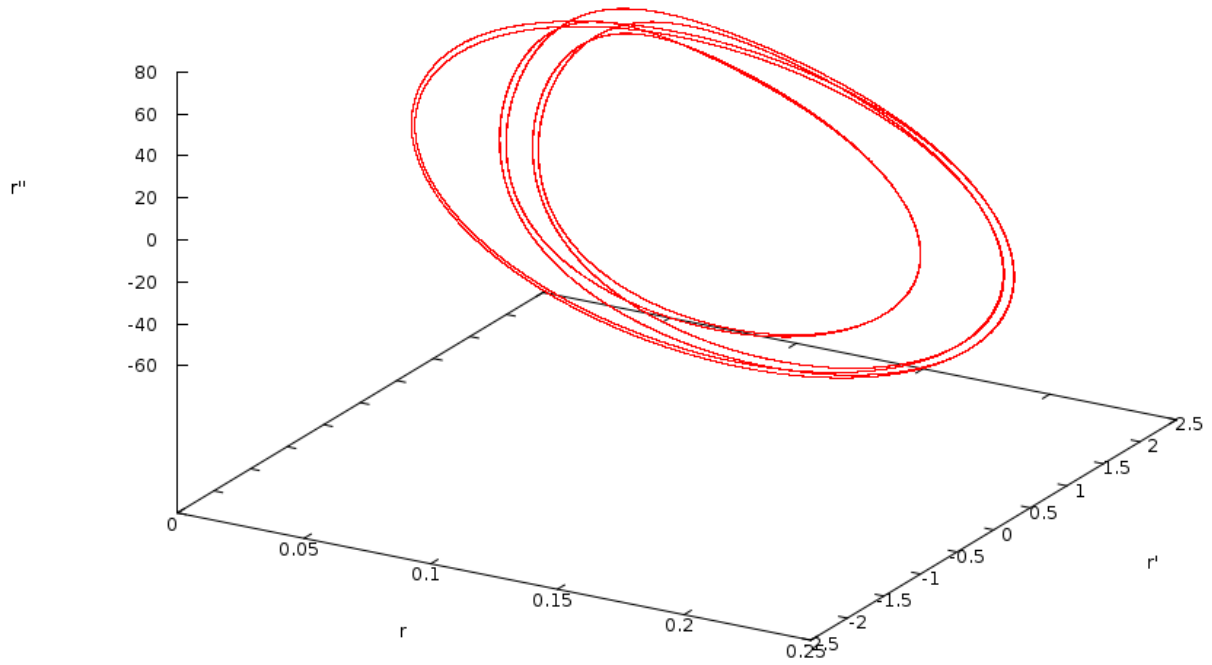
Plan de Poincare



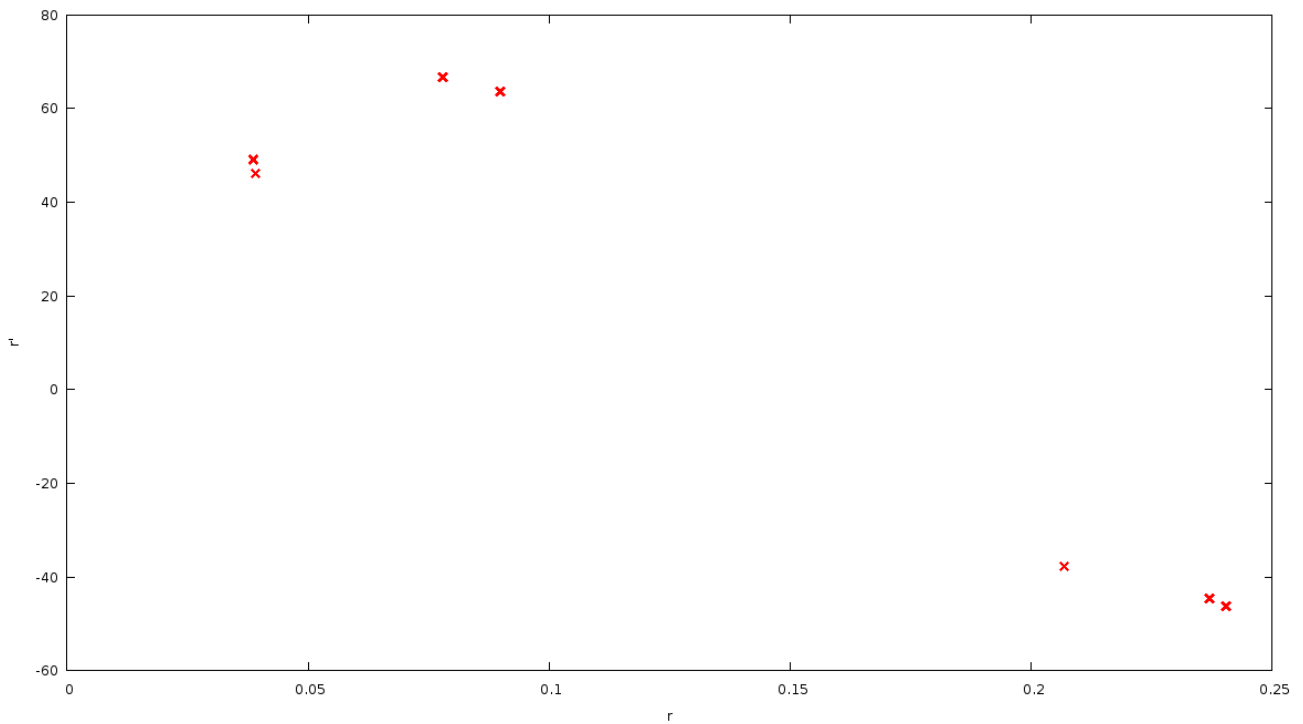
E.5 Etude de la fenêtre stable

Spectre de la variation de $x(t)$ 

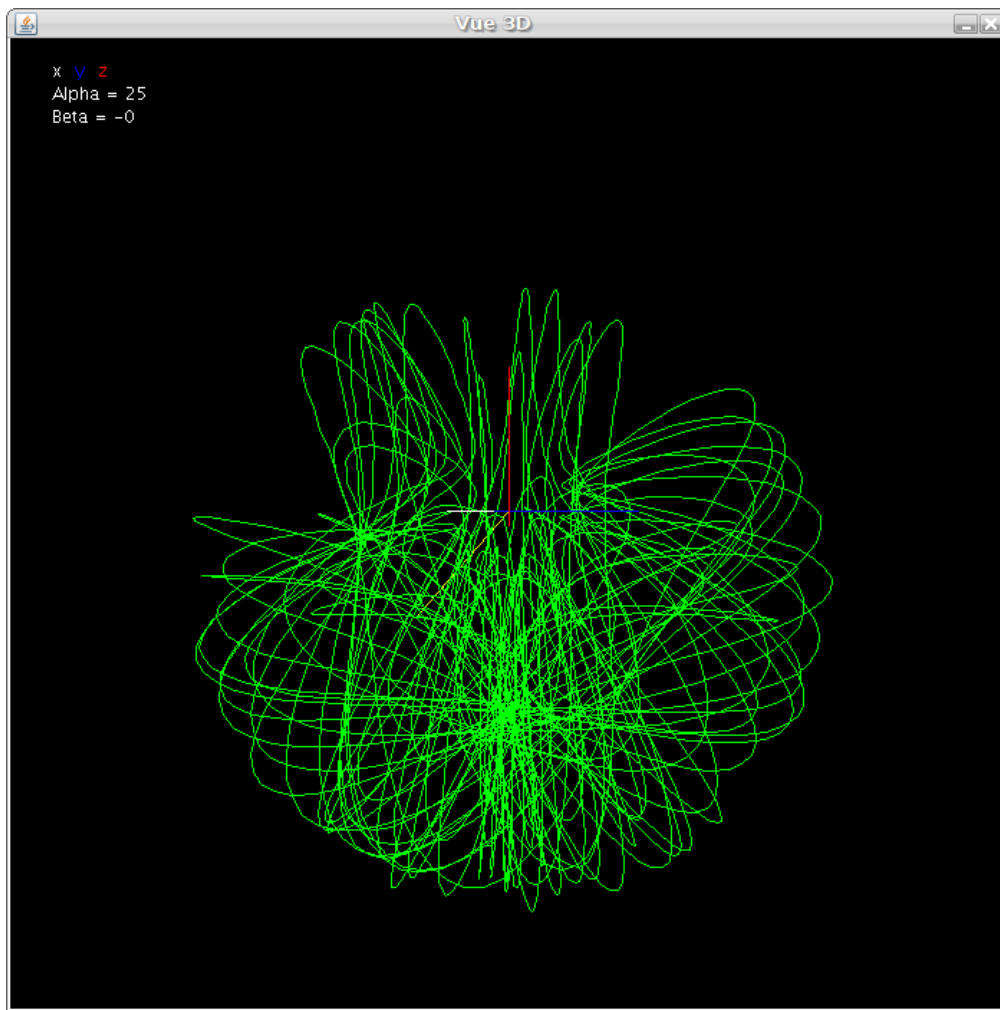
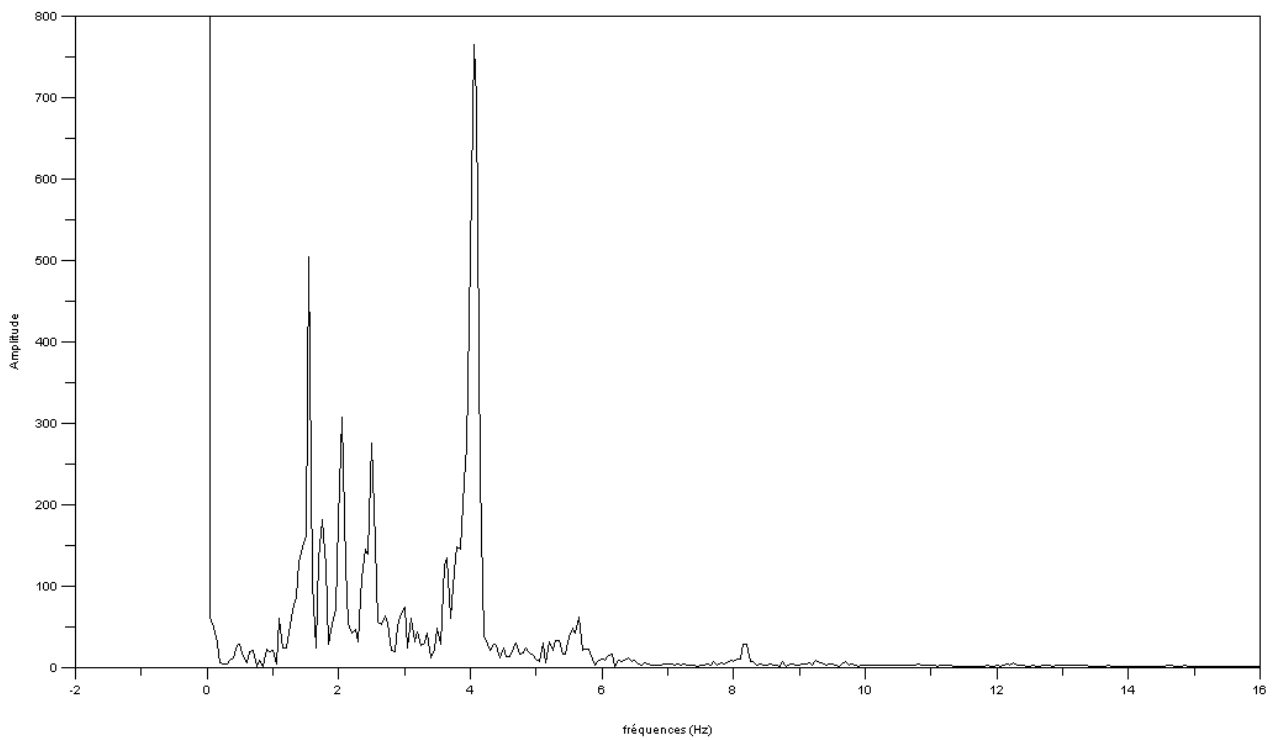
Espace des phases



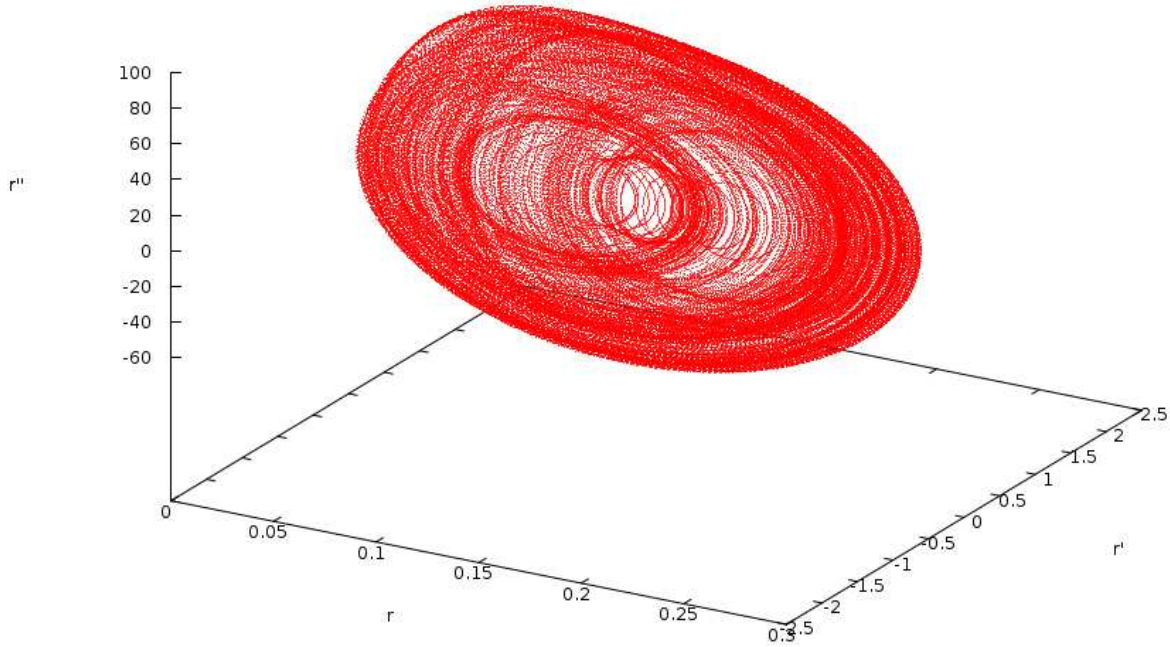
Plan de Poincare



E.6 Etude du chaos

Spectre de la variation de $\tau(t)$ 

Espace des phases



Plan de Poincare

