

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



No Réf :.....

Centre universitaire
Abd Elhafid Boussouf -Mila

Institut des Sciences et Technologie

Département de Mathématiques et Informatique

Mémoire préparé en vue de l'obtention du diplôme de Master

En: *Mathématiques*

Spécialité : *Mathématiques Appliquées*

Sur un algorithme pour le calcul des exposants de Lyapunov d'un système dynamique

Préparé par :

- FILALI Housna
- CHOUCANE Kaouther

Soutenue devant le jury:

S. Kaouache (M. C. B)	C. U. Abd Elhafid Boussouf	Président
M-S. Abdelouahab (M. C. A)	C. U. Abd Elhafid Boussouf	Rapporteur
W. Laouira (M. C. A)	C. U. Abd Elhafid Boussouf	Examinatrice

Année Universitaire : 2019/2020

Remerciements

Nous tenons tout d'abord à remercier Dieu
Le tout puissant, qui nous a donné
la force et la patience d'accomplir ce
modeste travail.

Nous tenons à saisir cette occasion et
à dresser nos profonds remerciements
et nos profondes reconnaissances à nos
parents.

Un remerciement particulier à notre encadreur
Dr : ABD ELOUAHAB MOHAMMED SALAH
pour sa présence, ses précieux conseils et son
aide durant toute la période du travail.

Nos vifs remerciements vont également aux membres
du jury. Dr : W. Laouira, et Dr : S. Kaouache.

Et enfin, nous tenons également à remercier
toutes les personnes qui ont participé
de près ou de loin à la réalisation de ce travail.

Dédicaces

Je dédie ce modeste travail :

A ma mère et A mon père, source de force et courage.

A mes enseignants,

A ma famille,

A tous mes amis et collègues.

Housna et Kaouther.

Table des matières

1	Généralités sur les systèmes dynamiques	12
1.1	Système dynamique	12
1.1.1	Système dynamique à temps discret	12
1.1.2	Système dynamique à temps continu	13
1.2	Flot, point fixe, portrait de phase	13
1.2.1	Ensembles limites ,attracteurs et bassin d'attraction	16
1.3	Notion de stabilité	19
1.3.1	Stabilité d'un cycle limite	24
1.4	Bifurcation	26
1.4.1	Bifurcations locales en dimension un	27
2	Chaos et exposants de Lyapunov :	31
2.1	Systèmes chaotiques :	31
2.1.1	Quelques définitions du chaos	31
2.1.2	Caractéristiques du chaos	32
2.1.3	Transitions vers le chaos	33
2.2	Exposant de Lyapunov	35
2.2.1	Cas des systèmes discrets unidimensionnelle	35
2.2.2	Cas des systèmes discrets multidimensionnels	36
2.2.3	Cas des systèmes continus multidimensionnels	38
3	Algorithmes pour le calcul des exposants de Lyapunov et applications	40
3.1	Algorithmes	40
3.1.1	Algorithme de Wolf pour ODE :	40
3.1.2	Algorithme de Wolf pour une série temporelle	41
3.1.3	Algorithme de Kantz	44
3.1.4	Algorithme de la limite inférieure d'erreur (LLE)	46
3.2	Applications	47
3.2.1	Application de L'algorithme de Wolf pour ODE :	47
3.2.2	Application de L'algorithme de Wolf pour une série temporelle	49
3.2.3	Application de L'algorithme LLE	49

Table des figures

1.1	Exemple de trajectoire pour le système Lorenz	14
1.2	Représentation du flot.	15
1.3	Orbite homocline à gauche et orbite hétérocline à droite.	15
1.4	Trois portraits de phase \mathbb{R}, \mathbb{R}^2 et \mathbb{R}^3	16
1.5	Quelques exemples d'attracteurs	18
1.6	Différentes situations	19
1.7	Stabilité globale	21
1.8	Les différents types d'états d'équilibre	23
1.9	Section de Poincare la trajectoire de phases coupe le plan	24
1.10	Les types de cycle limite	25
1.11	Trajectoire centrale spirale vers un cycle de singularités à quatre points critiques	26
1.12	Bifurcation Fourche	29
1.13	Bifurcation de Hopf	30
2.1	La sensibilité aux conditions initiales.	32
2.2	Transition vers le chaos via le doublement de période.	33
2.3	Transition vers le chaos par l'intermittence.	34
3.1	La forme de la sphère après le déplacement.	41
3.2	Calcul des exposants de Lyapounov à partir de 2 trajectoires voisines de l'espace des phases.	42
3.3	Calcul des deux plus grands exposants de Lyapounov à partir de 3 trajectoires voisines de l'espace des phases.	43
3.4	La reconstruction de l'espace de phase d'après l'algorithme de Kantz.	44
3.5	Estimation de l'exposant maximal de Lyapunov pour l'application logistique par la méthode de Kantz.	45
3.6	Attracteurs de quelques systèmes.	48
3.7	Résultat du programme de Wolf (ODE).	49
3.8	Résultat de programme de la limite inférieure d'erreur.	50
3.9	Résultats des trois algorithmes : Wolf pour ODE, Wolf pour (ST) et LLE appliquée sur le système financier.	52

3.10 L'exposant de Lyapunov de l'équation Logistique par rapport au nombre d'itérations.	53
3.11 Les exposants de Lyapunov par rapport au paramètre.	54

Liste des tableaux

2.1	Type d'attracteurs d'après l'exposant de Lyapunov.	39
3.1	Quelques systèmes dynamiques et leurs paramètres et conditions initiales.	47
3.2	Résultats de calculs des exposants de Lyapunov avec l'algorithme de Wolf ODE, Wolf (ST) et LLE.	51

Résumé

Le calcul des exposants de Lyapunov est un outil efficace pour décrire le comportement d'un système dynamique discret ou continu, il définit la divergence entre deux trajectoires initialement voisines. Si le plus grand exposant de Lyapunov est négatif alors la dynamique est chaotique mais s'il est négatif ou nul alors la dynamique est régulière. Dans ce mémoire on va étudier quelques algorithmes pour le calcul numériques des exposants de Lyapunov qui sont : l'algorithme de Wolf pour ODE, l'algorithme de Wolf pour série temporelle(ST), l'algorithme de Kantz et l'algorithme LLE ; et ces applications sur quatre systèmes dynamiques qui sont : L'équation logistique(cas discret, dim=1), le système de Lozi(cas discret, dim=2), le système de Lorenz(cas continu, dim=3)et une système financier (cas continu, dim=3) ; et après les calcul nous trouverons que tous est chaotiques pour les valeurs des paramètres choisis.

Mots clés : Chaos, système dynamique, exposant de Lyapunov, Wolf.

Introduction

Contexte historique

Les systèmes dynamiques représentent des phénomènes qui évoluent dans l'espace et / ou de temps. Ces systèmes viennent de Biologie, Physique, Chimie, ou même des sciences sociales, le système dynamique est le sujet qui fournit des outils mathématiques pour son analyse. Ils sont développés et spécialisés au cours du 19^e siècle.

Au cours de la fin de ce siècle (1890) le mathématicien, philosophe français **Henri Poincaré** avait déjà mis en évidence le phénomène de sensibilité aux conditions initiales lors de l'étude astronomique du problème des trois corps.

Toujours au 19^e siècle, de notre point de vue, les années 1970 sont une période de rupture dans le domaine que nous considérons ici : « systèmes dynamiques et chaos », le mathématicien russe **Alexandre Lyapunov** effectue des recherches sur la stabilité du mouvement. Il introduit l'idée de mesure de l'écart entre deux trajectoires ayant des conditions initiales voisines. Lorsque cet écart évolue exponentiellement on parle de sensibilité aux conditions initiales. Le météorologue **Edward Lorenz** venait de découvrir le phénomène de sensibilité aux conditions initiales, les systèmes répondant à cette propriété seront, à partir de 1975, dénommés systèmes chaotiques.

Les travaux fondateurs de Lorenz en 1963, ont donné un aperçu des scientifiques de reconnaissance un nouveau type de mouvement appelé Chaos.

"Le battement d'ailes d'un papillon du Brésil déclenche-t-il une tornade au Texas?". Question que Lorenz pose en 1972 ; lors d'une réunion de l'American Association for the Advancement of Science.

Cette formule traduit la sensibilité aux conditions initiales d'un système dynamique, atmosphérique ou autre, c'est à dire qu'elle affirme un principe de non-prédictibilité lorsque une petite cause peut avoir de grands effets.

La meilleure façon de décrire la sensibilité aux conditions initiales est à travers les exposants positifs de Lyapunov. Les exposants de Lyapunov jouent un rôle important dans les systèmes non linéaires, en particulier dans les systèmes chaotiques, essentiellement dû au fait que les systèmes chaotiques peuvent être caractérisés par la positivité de l'un des principaux (ou le plus grand) exposant de Lyapunov. Pour les systèmes dynamiques paramétrés, les exposants de Lyapunov sont bénéfiques pour l'identification de certains types de bifurcations. Ces exposants de Lyapunov sont commodés pour l'analyse et le calcul.

Dans le cas général une estimation numérique des exposants (Calcul numérique des ex-



posants de Lyapunov) est nécessaire , la façon la plus pertinente pour la description de cette caractéristique est à travers l'utilisation des exposants positifs de Lyapunov. Si le système dynamique a plus d'un exposant de Lyapunov positif, alors le système est considéré comme hyper-chaotique.

Ce mémoire contient trois chapitres.

Le premier chapitre, regroupe des concepts et notions de base de la théorie des systèmes dynamiques, nous allons illustrer des généralités sur les systèmes dynamiques pour bien comprendre la suite de ce mémoire.

Le deuxième chapitre, nous donnons des notions de la théorie du chaos partant de la définition du chaos, ensuite des propriétés et scénario de transition vers le chaos et les exposants de Lyapunov.

Le troisième chapitre, nous présentons quelques algorithmes pour le calcul des exposants de Lyapunov (algorithme de Wolf ODE, L'algorithme de Wolf série temporelle, L'algorithme de Kantz, L'algorithme de la limite inférieure d'erreur) et applications

Chapitre 1

Généralités sur les systèmes dynamiques

Ce chapitre se consacre aux notions de système dynamique, ainsi nous allons présenter la notions de stabilité d'un point fixe et d'un cycle limite, finalement on donne une brève introduction sur la notion de bifurcation locale des points fixes.

1.1 Système dynamique

Un système dynamique est une structure qui évolue au cours du temps de façon à la fois :

- Causale, où son avenir ne dépend que de phénomènes du passé ou du présent
- Déterministe, c'est-à-dire qu'à partir d'une condition initiale donnée à l'instant présent va correspondre à chaque instant ultérieur un et un seul état futur possible

L'évolution déterministe du système dynamique peut alors se modéliser de deux façons distinctes :

- * Une évolution continue dans le temps, représentée par une équation différentielle ordinaire.
- * Une évolution discrète dans le temps, l'étude théorique de ces modèles discrets est fondamentale, car elle permet de mettre en évidence des résultats importants, qui se généralisent souvent aux évolutions dynamiques continues. Elle est représentée par le modèle général des équations aux différences finie

L'évolution d'un système dynamique peut se modéliser à temps continu ou à temps discret.

1.1.1 Système dynamique à temps discret

Un système dynamique dans le cas discret est représenté par une équation aux différences sous la forme :

$$x_{i+1} = f(x_i, \alpha)$$

Où f est une fonction continue ou au moins continue par morceaux, $x_i \in \mathbb{R}^i$ et le vecteur d'état à l'instant $i (i \in N)$ et $\alpha \in \mathbb{R}^r$ est celui des paramètres. La fonction f peut, dans certains cas, être inversible, ce qui introduit la notion de réversibilité qui permet de remonter dans le temps.

1.1.2 Système dynamique à temps continu

Un système dynamique en temps continu est décrit par un système d'équations différentielles sous la forme :

$$x'(t) = F(x(t), t)$$

Où $F : \mathbb{R}^n * \mathbb{R}^+ \longrightarrow \mathbb{R}^n$

L'ensemble des solutions d'un système différentiel constitue un système dynamique. En d'autres termes, la fonction appelée champ de vecteurs, définit d'une part le système différentiel mais détermine également un système dynamique continu :

1. Le champ de vecteurs (si non linéaire condition obligatoire ou necessair)
2. Lorsque La fonction ne dépend pas explicitement alors le système dynamique est dit nonlinéaire du temps, le champ de vecteur est alors dit autonome .
3. Les systèmes étudiés sont dissipatifs(cas chaotique) : la divergence du champ de vecteurs est constante négative ou dans ce mémoire elle est en moyenne négative sur les orbites considérées.

Exemple Prenons comme exemple le système de Lorenz qui l'un des systèmes différentiels les plus connus et étudiés, il vérifie ces trois propriétés[9]

$$\begin{cases} \dot{x} &= \sigma(y - x), \\ \dot{y} &= rx - y - zx, \\ \dot{z} &= -bz - xy. \end{cases}$$

Avec x, y et z sont les variable d'état du système σ, r et b sont les paramètres de contrôle. Les paramètres pour l'exemple de trajectoire donné dans la figure ont été choisis de la manière suivante : $\sigma = 10, \rho = 28, b = 8/3$ avec la condition initiale $(x_0, y_0, z_0) = (2, 5, 20)$

1.2 Flot, point fixe, portrait de phase

Dans la suite on s'intéresse à l'étude des systèmes dynamiques continus autonomes de la forme :

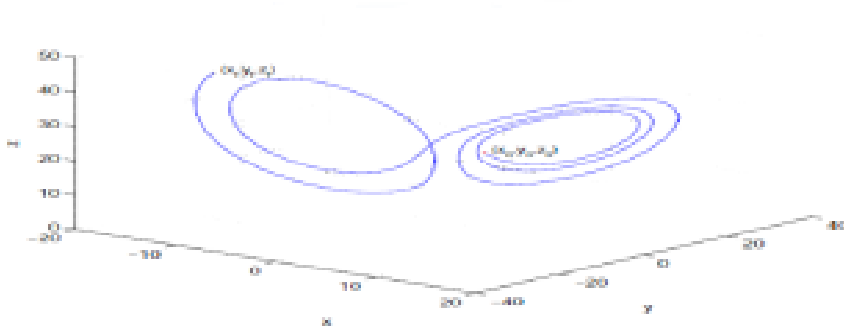


FIGURE 1.1 – Exemple de trajectoire pour le système Lorenz

$$\frac{dx}{dt} = f(x). \quad (1.1)$$

Pour assurer l'existence et l'unicité du flot du système (1.1) nous avons le théorème suivant :

Théorème 1.2.1. (*Théorème de Cauchy-Lipschitz*)

Considérons l'équation différentielle (1.1), où f est une fonction localement lipschitzienne sur un domaine U de \mathbb{R}^n et à valeurs dans \mathbb{R}^n .

Alors, pour tout $x_0 \in U$, il existe une solution maximale unique de (1.1) satisfaisant $x(0) = x_0$.

► Flot

Définition 1.2.1. La correspondance $\psi_t : x_0 \rightarrow x(t)$ qui associe à une donnée initiale x_0 la valeur de la solution maximale $x(t)$ au temps t , qui correspond à cette donnée initiale, est appelée le flot au temps t du champs de vecteurs f [10].

* Flot du champ de vecteurs utiliser [10]

Le flot du champ de vecteurs est l'application qui associe à (t, x) la solution maximale $x(t)$ au temps t qui correspond à la donnée initiale x

$$(t, x) \rightarrow \psi(t, x) = \psi_t = x(t)$$

* Complet [11]

Le flot est dit complet lorsque cette correspondance est définie pour toute valeur de $t \in]-\infty, +\infty[$

* Régularité du flot [2]

Si f est de classe C^k le flot est lui-même de classe C^k

* Transitivité du flot [2]

Le flot vérifie, pour tous t et $s \in \mathbb{R}^+$, $\psi_t \circ \psi_s = \psi_{s+t}$

* Pour un système linéaire $\dot{x} = Ax$, le flot est donné par $\psi_t = \exp(tA)x$



FIGURE 1.2 – Représentation du flot.

► Point fixe

Définition 1.2.2. :[11]

* On appelle point fixe (ou point stationnaire ou point d'équilibre ou point critique) de (1.1), le point x^* de l'espace des phases obtenu en annulant le second membre de (1.1) $f(x^*) = 0$. Par le changement de variables $X = x - x^*$, on peut ramener le point x^* à l'origine.

On peut aussi définir le point fixe par le flot (ie : pour tout $t \in \mathbb{R}$, $\psi_t(x^*) = x^*$).

* Tout point autre que les points fixe est un point ordinaire[1].

* Un point ordinaire a est dit périodique s'il existe $T > 0$ telle que $\psi_T(a) = a$.

* Un point ordinaire et non périodique a , est dit récurrent si pour tout voisinage V de a et tout $T \in \mathbb{R}$ il existe $t > T$ tel que $\psi_t(a) \in V$

* Une orbite φ_{x_0} telle qu'il existe deux points équilibre a et b vérifiant :

$$\lim_{t \rightarrow -\infty} \psi_t(x_0) = a \text{ et } \lim_{t \rightarrow +\infty} \psi_t(x_0) = b$$

est dite orbite hétérocline si $a \neq b$ et orbite homocline si $a = b$

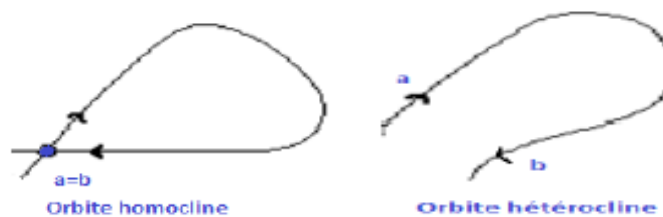


FIGURE 1.3 – Orbite homocline à gauche et orbite hétérocline à droite.

* Un ensemble $S \subset U$ est dit invariant par le flot ψ_t sur U (ou bien par le système $\dot{x} = f(x)$ correspondant) si pour tout $x \in S$ et tout $t \in \mathbb{R}$ on a $\psi_t(x) \in S$. Si S vérifie la propriété que $\psi_t \in S$ pour tout $x \in S$ et tout $t > 0$ alors on dit que S est positivement invariant.

► Portrait de phase

Définition 1.2.3. :[12]

Le portrait de phase d'un système dynamique est une représentation graphique (à l'aide des flèches) de plusieurs trajectoires représentatives dans l'espace de phase.

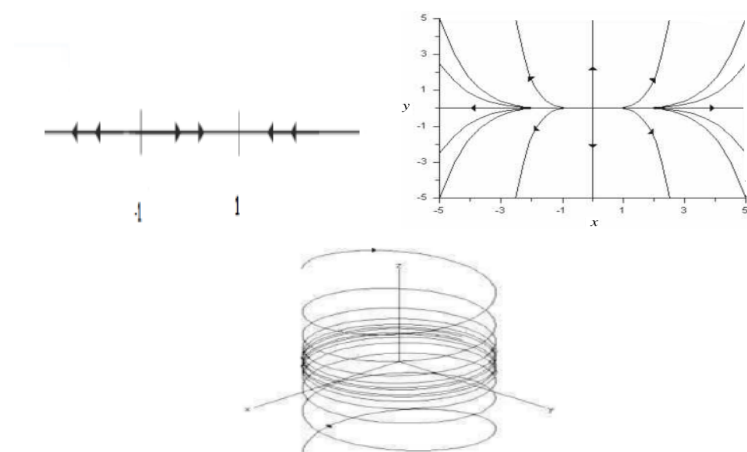


FIGURE 1.4 – Trois portraits de phase \mathbb{R}, \mathbb{R}^2 et \mathbb{R}^3 .

1.2.1 Ensembles limites ,attracteurs et bassin d'attraction

► Ensembles limites

Définition 1.2.4. :[13]

Soient X un champ de vecteurs défini sur un ouvert U de \mathbb{R}^n , Soit Γ l'orbite de X passant par x_0 , Elle est paramétrisée par une solution maximale $x(t)$ du problème de Cauchy associé : $\Gamma = \{x(t) | t \in (\alpha, \beta)\}$.

* Si $\beta = +\infty$, l'ensemble ω – limite de l'orbite (ou de x_0) est défini par :

$$\omega(x_0) = \left\{ q \in U | \exists (t_n) \in \mathbb{R}^{\mathbb{N}}, t_n \rightarrow +\infty \text{ et } (x(t_n)) \rightarrow q \right\}$$

* Si $\alpha = -\infty$, l'ensemble α – limite de l'orbite (ou de x_0) est défini par :

$$\alpha(x_0) = \left\{ q \in U | \exists (t_n) \in \mathbb{R}^{\mathbb{N}}, t_n \rightarrow -\infty \text{ et } (x(t_n)) \rightarrow q \right\}.$$

Tous les points d'une même orbite ont les mêmes ensembles α -limite et ω -limite

► Ensemble invariant

Définition 1.2.5. :[14]

Un ensemble $H \subset U$ est dit invariant par un champ de vecteur si toute solution $x(t)$ du système différentiel associé au champ de vecteurs issu de H vérifie $x(t) \subset H$ pour tout t pour lequel cette solution est définie.

► Attracteurs

Définition 1.2.6. :

Un attracteur est un objet géométrique vers lequel tendent toutes les trajectoires des points de l'espace des phases, c'est à dire une situation (ou un ensemble d'états) vers lesquels évolue un système, quelles que soient ses conditions initiales.

Mathématiquement, l'ensemble A est un attracteur si :

- . Pour tout voisinage U de A il existe un voisinage V de A tel que toute solution $x(x_0, t) = \varphi_t(x_0)$ restera dans U si $x_0 \in V$
- . $\cap \varphi_t(V) = A; t \geq 0$
- . Il existe une orbite dense dans A

Définition 1.2.7. :

Un attracteur possède les propriétés suivantes :

- . Un sous ensemble borné de l'espace est de volume nul invariant par le flot, Autrement dit, tout point de l'espace d'états qui appartient à un attracteur demeure à l'intérieur de cet attracteur pour tout t .
- . Un attracteur est indécomposable c'est-à-dire que la réunion de deux attracteurs n'est pas un attracteur.

• Les différents types d'attracteurs [1] :

Il existe deux types d'attracteurs : les attracteurs réguliers et les attracteurs étranges ou chaotiques

1- Attracteurs réguliers :

Les attracteurs réguliers caractérisent l'évolution de systèmes non chaotiques, et peuvent être de trois sortes

- * **Le point fixe [3][4]** : C'est le plus simple attracteur, le système évolue vers un état de repos (point).
- * **Le cycle limite périodique** : C'est une trajectoire fermée qui attire toutes les orbites proches.
- * **Attracteur quasi périodique (tore)** : Il est caractérisé par un régime quasi-périodique ayant n fréquences de base indépendantes.

2- Attracteurs étranges :

il s'agit d'ensembles dans l'espace des phases compacts, fermés, dans lesquels on retrouve des trajectoires chaotiques, Les caractéristiques d'un attracteur étrange sont :

- * Dans l'espace des phases l'attracteur est de volume nul.
- * La dimension de l'attracteur étrange est fractale (non entière) pour un système continue autonome $2 < d < n$, n la dimension de l'espace des phases,
- * Sensibilité aux conditions initiales où n est la dimension ...(deux trajectoires initialement voisines finissent toujours par s'écartier l'une de l'autre).

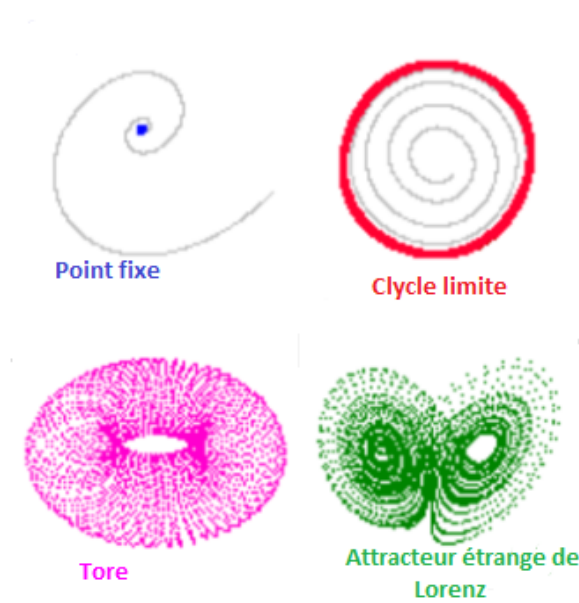


FIGURE 1.5 – Quelques exemples d'attracteurs

1.3 Notion de stabilité

La question de la stabilité se pose de la façon suivante : Si l'on écarte le système de l'équilibre, y reviendra-t-il ? Ou bien une petite perturbation qui éloigne le système légèrement de son régime stationnaire, peut avoir des conséquences importantes et être amplifiée au cours du temps ? [1][2].

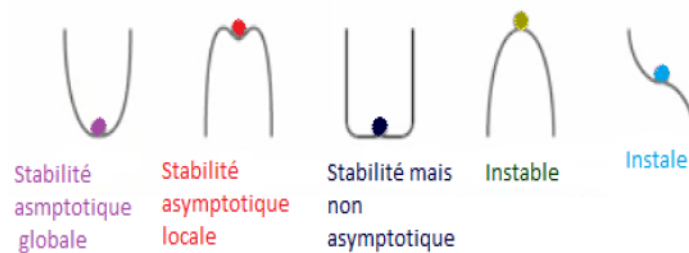


FIGURE 1.6 – Différentes situations

◆ Stabilité au sens de Lyapunov : [3]

Soit le système dynamique suivant :

$$\frac{dx}{dt} = f(x, t), \quad (1.2)$$

avec f une fonction non linéaire.

Définition 1.3.1. (Stabilité) :

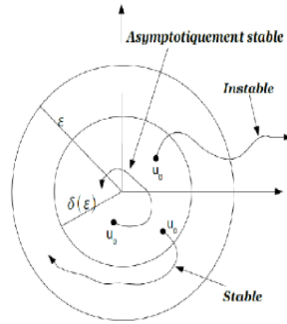
Un point d'équilibre a de (1.2) est stable au sens de Lyapunov si pour tout $\varepsilon > 0$, il existe $\eta > 0$ tel que :
pour tout X vérifiant $\|X - a\| \leq \eta$, on a $\|\Psi_t(X) - a\| \leq \varepsilon$ pour tout $t > 0$.

Définition 1.3.2. (Stabilité asymptotique) :

Un point d'équilibre a de (1.2) est asymptotiquement stable au sens de Lyapunov s'il est stable au sens de Lyapunov et de plus pour tout X suffisamment proche de a , on a :

$$\lim_{t \rightarrow +\infty} \Psi_t(X) = a$$

Les définitions précédentes sont locales, elles ne concernent que les orbites voisines d'un point d'équilibre ($\|x - a\| \leq \eta$). Nous présentons ici deux méthodes de Lyapunov pour étudier la stabilité :



▲ Méthode directe :

La méthode directe est difficile à mettre en oeuvre mais, en contrepartie, elle est d'une portée beaucoup plus générale. Elle est basée sur la définition d'une fonction particulière, notée $V(x)$ et appelée fonction de Lyapunov, qui est décroissante le long des trajectoires du système. Le théorème suivant va résumer cette méthode.

Théorème 1.3.1. (Fonction de Lyapunov et stabilité globale) :

Si a est un point d'équilibre du système (1.2) et si la fonction V de classe C^1

$$V : \mathbb{R}^n \rightarrow [0, +\infty[$$

est telle que :

* $V(a) = 0$ et $V(x) > 0$ pour $x \neq a$

* V décroît le long de toutes les trajectoires ($\frac{dV}{dt} \leq 0$).

Alors a est stable au sens de Lyapunov. Si de plus pour $x \neq a$ $\frac{dV}{dt} < 0$ alors a est asymptotiquement stable au sens de Lyapunov. Si on suppose encore que V tend vers l'infini lorsque $x \in \mathbb{R}^n$ tend vers l'infini (en norme), alors toutes les trajectoires, même celles qui démarrent loin de a , tendent vers a (on dit que a est globalement asymptotiquement stable), mais si $\frac{dV}{dt} > 0$ pour $x \neq a$ alors a est instable. V est appelé fonction de Lyapunov du système. Il n'y a aucune méthode générale pour déterminer une fonction de Lyapunov, mais en mécanique et pour les systèmes électriques on peut souvent utiliser l'énergie totale comme fonction de Lyapunov

► **Exemple** Soit le système de Lorenz[9] :

$$\begin{cases} \dot{x} &= a(y - x) \\ \dot{y} &= cx - y - xz \\ \dot{z} &= xy - bz \end{cases}$$

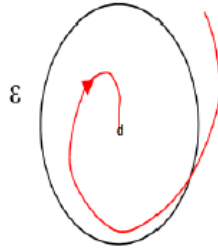


FIGURE 1.7 – Stabilité globale

pour c compris entre 0 et 1, on vérifie que l'origine est un point fixe globalement stable. Sa stabilité globale est étudiée à l'aide d'une fonction de Lyapunov qu'il faut déterminer intuitivement !

On vérifie que la fonction :

$V(x, y, z) = \frac{x^2 + a(y^2 + z^2)}{2}$ convient. On a $V(0, 0, 0) = 0$ et $V(x, y, z) > 0$ pour $(x, y, z) \neq (0, 0, 0)$

$$\begin{aligned} \frac{dV}{dt} &= a(y - x)x + ay(cx - y - xz) + az(-bz + xy) \\ &= -a(x^2 + y^2 - xy(c + 1) + bz^2) \end{aligned}$$

Comme $z^2 > 0$ et $x^2 + y^2(c + 1)xy > 0$ pour c compris entre 0 et 1, alors on a $\frac{dV}{dt} < 0$ pour $(x, y, z) \neq (0, 0, 0)$.

Par suite le point fixe $(0, 0, 0)$ est globalement asymptotiquement stable au sens de Lyapunov.

▲ Méthode indirecte (Linéarisation) :

Par un changement de coordonnées, le point fixe de (1.2) se ramène à l'origine ($f(0) = 0$) et le développement de f en série de Taylor autour de $x = 0$ donne

$$f(x) = Df(0)x + \frac{1}{2!}D^2f(0)(x, x) + \frac{1}{3!}D^3f(0)(x, x, x) + \dots$$

La méthode indirecte de Lyapunov, pour étudier la stabilité autour d'un point d'équilibre a , consiste à étudier le système linéaire

$$\dot{x} = Ax \tag{1.3}$$

Avec

$$A = Df(0) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}$$

la matrice jacobienne de f en 0 s'appelle le linéarisé du système (1.3) au point d'équilibre 0. Si A possède n valeurs propres distinctes $\lambda_i, i = 1, 2, \dots, n$ (les valeurs propres de A sont appelées exposants caractéristiques de l'équilibre 0) alors la solution de (1.3) est

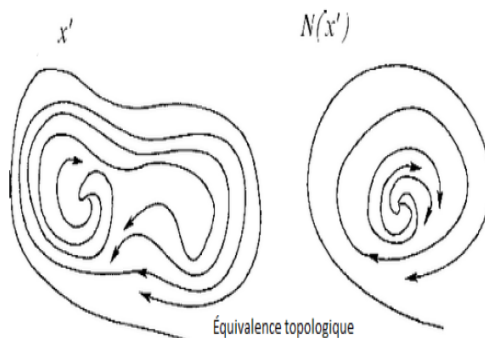
$$x(t) = \sum_{i=1}^n C_i e^{\lambda_i t} v_i, v_i \text{ est le vecteur propre associé à } \lambda_i$$

d'où le théorème suivant :

Théorème 1.3.2. *Considérons le système (1.3) avec r valeurs propres distinctes, $\lambda_1, \dots, \lambda_r$ et $a = 0$ point d'équilibre*

- Si $Re(\lambda_j) > 0$, pour $j \in \{1, \dots, r\}$, alors 0 est instable.
- Si $Re(\lambda_j) < 0$, pour tout $j \in \{1, \dots, r\}$, alors 0 est asymptotiquement stable.
- Si $Re(\lambda_j) \leq 0$, avec un pôle de multiplicité 1 est tel que ($Re\lambda_j = 0$), pour $j \in \{1, \dots, r\}$, alors 0 est stable

Définition 1.3.3. *Deux flots φ_t et ψ_t sont dits topologiquement équivalents dans un voisinages du point d'équilibre, s'il existe un homéomorphisme h qui envoie le point d'équilibre du premier flot en le point d'équilibre du deuxième flot et qui conjugue les flots, c'est à dire $h \circ \varphi_t = \psi_t \circ h$*



Définition 1.3.4. :

Le point d'équilibre a est dit hyperbolique si tous ses exposants caractéristiques ont des parties réelles non nulles.

Théorème 1.3.3. (Hartman-Grobman)

Considérons le système (1.2) de flot ϕ_t , si a est un point d'équilibre hyperbolique, alors il existe un voisinage V de a sur lequel le flot ϕ_t est topologiquement équivalent au flot du linéarisé (1.3) du système en a

Par conséquence on a le théorème suivant :

Théorème 1.3.4. Soit d un point d'équilibre de (1.2).

- * Si les valeurs propres de $Df(a)$ sont toutes à partie réelle strictement négative, alors d est un équilibre asymptotiquement stable au sens de Lyapunov.
- * Si l'une des valeurs propres de $Df(a)$ possède une partie réelle strictement positive alors a n'est pas un équilibre stable au sens de Lyapunov[15].

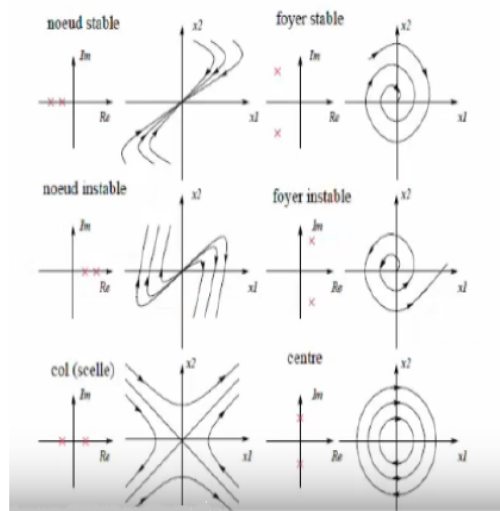


FIGURE 1.8 – Les différents types d'états d'équilibre

Exemple :[27]

Soit le système suivant :

$$\begin{cases} \dot{x} = -y + x(r^4 - 3r^2 + 1), \\ \dot{y} = x + y(r^4 - 3r^2 + 1), \end{cases} \quad (1.4)$$

avec $r^2 = x^2 + y^2$.

Le système (1.4) admet le point d'équilibre, $(0,0)$.

la matrice Jacobienne est

$$J = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$$

qui admet deux valeurs propres $\lambda_{\pm} = 1 \pm i$

comme $Re(\lambda_{\pm}) = 1 > 0$ alors l'origine est un foyer instable

1.3.1 Stabilité d'un cycle limite

Pour étudier la stabilité d'un cycle limite on a souvent recours à la section de Poincaré et l'application de premier retour.

▲ Section de Poincaré et application de premier retour :

l'application de Poincaré est un outil important pour l'analyse des orbites périodiques qui permet de ramener l'étude d'un système dynamique continu de dimension n au voisinage de l'orbite périodique, à celle d'une application (Itération ou système dynamique discret) d'une section transverse de dimension $n - 1$ dans elle même.

Définissons une hyper surface Σ de dimension $n - 1$ appelée section de Poincaré[15][6].

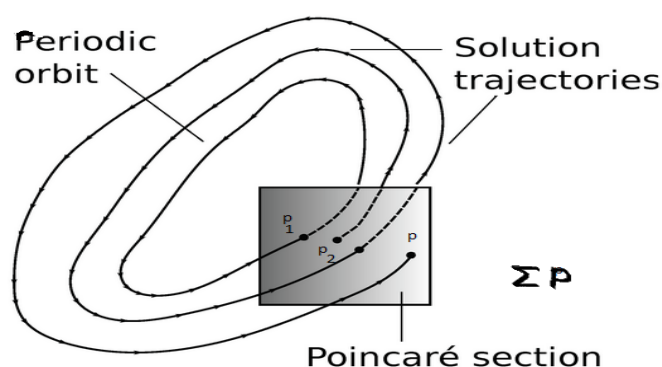


FIGURE 1.9 – Section de Poincaré la trajectoire de phases coupe le plan

Définition 1.3.5. Une orbite périodique d'un champ de vecteurs f est une orbite passant par un point x_0 , qui n'est pas un point singulier, pour lequel il existe un nombre $T > 0$ appelé période vérifiant $x(T) = x_0$

Théorème 1.3.5. On note $\Gamma = \{x : x = \phi_t(x_0), 0 \leq t \leq T\}$ l'orbite périodique de période T

Soit Σ l'hyperplan orthogonal à Γ en x_0

$$\Sigma = \{x : (x - x_0) \cdot f(x_0) = 0\}.$$

Alors il existe $\sigma > 0$ et une unique fonction $x \rightarrow \tau(x)$ de classe C^k définie pour

$$x \in \Sigma, |x - x_0| < \delta, \text{ telle que } \phi_{\tau(x)}(x) \in \Sigma$$

La fonction $x \rightarrow \tau(x)$ est appelée la fonction temps de premier retour

Définition 1.3.6. : L'application[6]

$$P : x \rightarrow P(x) = \phi_{\tau(x)}(x),$$

est appelée l'application de premier retour de Poincaré associée à l'orbite périodique τ .
Ce qui précède démontre qu'elle est de classe C^k .

Le cycle limite noté ζ est :

- * stable si toute trajectoire commençant suffisamment proche de cycle limite ; l'approche pour $t \rightarrow +\infty$ c'est à dire, s'enroule elle même sur le cycle limite ζ
- * instable si toute trajectoire commençant suffisamment proche de ζ ; l'approche pour $t \rightarrow -\infty$ c'est à dire, se déroule à partir de ζ .
- * semi-stable si les trajectoires s'approchent d'un côté et de l'autre s'éloignent (en pratique instable). Un cycle limite stable est aussi ce qu'on appelle un attracteur périodique.

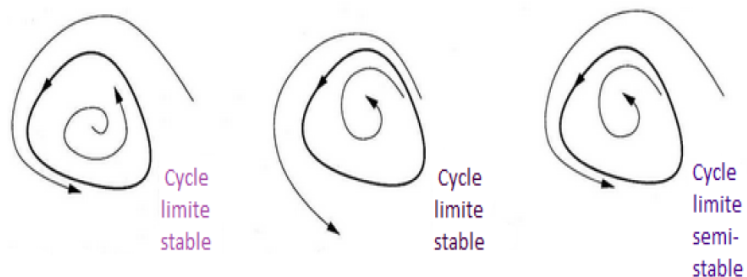


FIGURE 1.10 – Les types de cycle limite

▲ Théorème de Poincaré-Bendixson et applications :

Théorème 1.3.6. [5][6] *Un ensemble limite $\omega(x)$ ou $\alpha(x)$ non vide compact d'un système dynamique C^1 du plan, qui ne contient pas de point d'équilibre est une orbite périodique.*

★Applications du Théorème Description des ensembles limites. Le théorème de Poincaré-Bendixson permet de décrire tous les ensembles limites compacts d'un système dynamique plan. Deux cas se présentent, soit c'est une trajectoire périodique, soit c'est une union de points d'équilibres et de trajectoires reliant ces points, par exemple :



FIGURE 1.11 – Trajectoire centrale spirale vers un cycle de singularités à quatre points critiques

La trajectoire centrale spirale vers un cycle de singularités à quatre points critiques. Ce type de portrait de phase ne survit pas à de petites perturbations du champ de vecteurs. Si on bouge un peu f (un C^k – *difféomorphisme*) on peut faire disparaître les trajectoires reliant deux points critiques

1.4 Bifurcation

Le terme de bifurcation est utilisé pour désigner dans un sens large, toute modification qualitative du comportement d'un système dynamique suite à la variation de l'un de ces paramètres. Une bifurcation correspond donc à la collision de deux objets (point répulsif ou selle, attracteur) ou de deux variétés. La collision de deux objets donne naissance à une bifurcation "locale" alors que la collision de deux variétés donne naissance à une bifurcation "globale"[16]

Définition 1.4.1. :

Considérons le système

$$\dot{x} = f(x, \mu), \text{ où } x \in \mathbb{R}^n, \mu \in \mathbb{R}^m \quad (1.5)$$

et son point d'équilibre E dont la coordonnée x_E va dépendre de la valeur de μ si la nature du point d'équilibre subit un changement soudain lorsque μ varie légèrement autour d'une valeur μ_0 . on dit que le système subit une bifurcation

Définition 1.4.2. :

on d'efini' les valeurs de bifurcation du système (1.5) comme étant les valeurs de μ telle que la jacobienne $D_x f(x, \mu)$ calculée au point d'équilibre E a une valeur propre nulle.

Définition 1.4.3. :

L'équation

$$f(x, \mu) = 0$$

permet de trouver les coordonnées des points d'équilibres, défini, une equation dans le plan (x, μ) appelée diagramme de bifurcation.

Exemple Considérons le système[17]

$$\begin{cases} \dot{x} = \frac{dx}{dt} = y = f_1(x, y, r) \\ \dot{y} = \frac{dy}{dt} = -rx = f_2(x, y, r) \end{cases}$$

ce système est linéaire, il est équivalent á

$$\dot{X} = AX = \begin{pmatrix} 0 & 1 \\ -r & 0 \end{pmatrix} X, X = \begin{pmatrix} x \\ y \end{pmatrix}$$

le seul point critique est l'origine $(0, 0)$.

$$\det |A - \lambda I| = 0 \Leftrightarrow \lambda^2 + r = 0$$

1. Si $r > 0 \Rightarrow$ les valeurs propres de A sont $\lambda = \pm i\sqrt{r}$: d'oú l'origine est un "centre".
2. Si $r < 0 \Rightarrow$ les valeurs propres de A sont $\lambda = \pm\sqrt{r}$: d'oú l'origine est un "selle".

On conclut que le point $r = 0$ est un "point de bifurcation".

1.4.1 Bifurcations locales en dimension un

Nous présentons ici les quatre bifurcations de codimension un (nœud-col, transcritique, fourche et Hopf)[5][16].

1-Bifurcation nœud-col

La forme normale, dite aussi forme canonique, est la suivante :

$$\frac{dx}{dt} = \mu - \alpha x^2 \tag{1.6}$$

la transition est sous-critique pour $\alpha = -1$ et super-critique pour $\alpha = +1$.

On suppose que $\alpha = 1$.

- * Pour $\mu < 0$
il n'y a aucun point d'équilibre
- * Pour $\mu = 0$
0 est un seul point d'équilibre instable.
- * Pour $\mu > 0$
il y a deux points d'équilibres, le point d'équilibre stable $-\sqrt{\mu}$ et le point d'équilibre instable $\sqrt{\mu}$. Donc avant la valeur de bifurcation il y a une apparition de deux points fixes l'une est stable et l'autre est instable. Alors on a une bifurcation noeud-col

2-Bifurcation transcritique

La forme normale est :

$$\frac{dx}{dt} = x^2 - \mu x \quad (1.7)$$

- * Pour $\mu < 0$
il y a deux points d'équilibre, le point d'équilibre stable μ et le point d'équilibre instable 0
- * Pour $\mu = 0$
0 est le seul point d'équilibre, il est instable.
- * Pour $\mu > 0$
il y a deux points d'équilibres, le point d'équilibre stable 0 et le point d'équilibre instable μ . Donc à la valeur de bifurcation il y a un échange de stabilité entre les deux points fixes, et après la valeur de bifurcation le point fixe stable devient instable et le point fixe instable devient stable. Alors on a une bifurcation transcritique

3-Bifurcation Fourche

Les anglo-saxons désignent cette bifurcation par le terme "pitchfork" qui se traduit littéralement par "fourche". Sa forme normale est la suivante

$$\frac{dx}{dt} = \mu x - \alpha x^3 \quad (1.8)$$

où μ est un paramètre. La bifurcation sera qualifiée différemment selon la valeur de la constante α :

la transition est sous-critique pour $\alpha = -1$ et super-critique pour $\alpha = +1$.

Le fambage d'une poutre constitue un exemple classique de bifurcation fourche

4- Bifurcation de Hopf

Cette bifurcation est observée en dimension supérieure ou égale à 2. Ici, on se place en dimension 2. Pour $z \in C$, la forme normale est :

$$\frac{dz}{dt} = \lambda z - \alpha |z|^2 \cdot z \quad (1.9)$$

où $\lambda \in C$ est un paramètre. La bifurcation sera qualifiée différemment selon la valeur de la constante α :

la transition est sous-critique pour $\alpha = -1$ et super-critique pour $\alpha = +1$.

Posons $z = r.e^{i\theta}$. Dans ce cas, la forme normale s'écrit :

$$\begin{cases} \dot{r} = \mu r + \alpha r^3 \\ \dot{\theta} = \omega \end{cases}$$

On constate que la bifurcation de Hopf est "la combinaison" d'une bifurcation fourche (\dot{r}) et d'une rotation ($\dot{\theta}$) à vitesse angulaire constante[17].

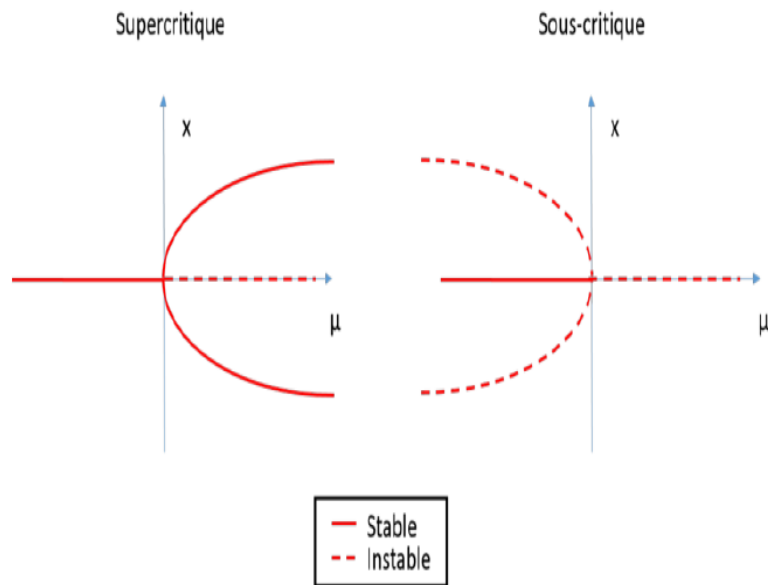


FIGURE 1.12 – Bifurcation Fourche

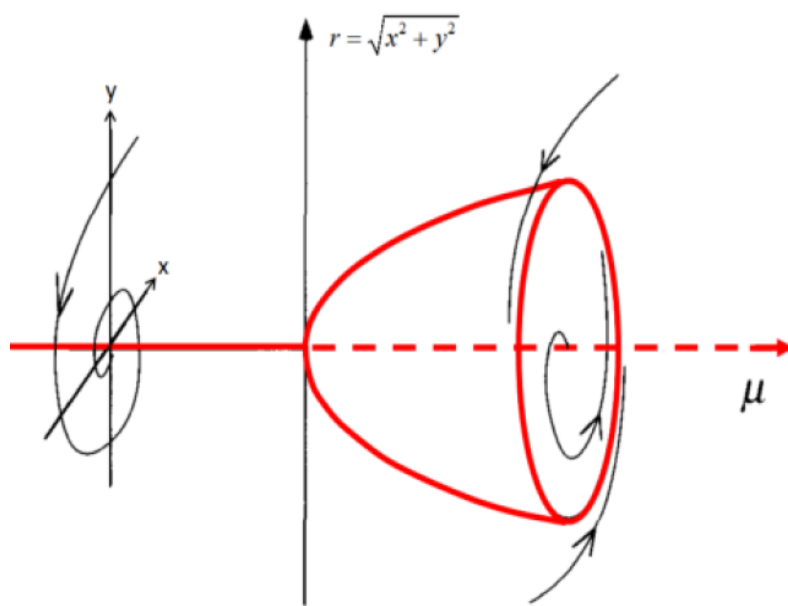


FIGURE 1.13 – Bifurcation de Hopf

Chapitre 2

Chaos et exposants de Lyapunov :

2.1 Systèmes chaotiques :

Dans l'usage courant, le «chaos» signifie "un état de désordre". Cependant, dans la théorie du chaos, ce terme est défini plus précisément. La théorie du chaos est un domaine des études en mathématiques, avec des applications dans plusieurs disciplines comme la physique, l'ingénierie, la biologie, l'économie, la météorologie, la sociologie. La théorie du chaos étudie le comportement des systèmes dynamiques qui sont très sensibles aux conditions initiales. De petites différences dans les conditions initiales (telles que celles dues à des erreurs d'arrondi dans le calcul numérique) produisent des résultats très divergents pour de tels systèmes dynamiques, ce qui rend la prévision à long terme impossible en général. Cela se produit même si ces systèmes sont déterministes, ce qui signifie que leur comportement futur est entièrement déterminé par leurs conditions initiales, sans éléments aléatoires impliqués. Ce comportement est connu sous le nom de chaos déterministe, ou tout simplement le chaos. Il n'existe pas de définition à la fois formelle et générale du chaos. Définition du chaos, quelques outils mathématiques de quantification du chaos et différentes route vers le chaos sont les grandes lignes de ce chapitre.

2.1.1 Quelques définitions du chaos

Définition 2.1 : *Chaos au sens de Li-Yorke*

Une application continue $f : I \rightarrow I$ où I est l'intervalle unitaire ($I = [0, 1]$), est chaotique au sens de Li-York s'il y a un ensemble indénombrable S tel que les trajectoire de deux points distincts x, y dans S sont proximaux et non asymptotique, ie : $\lim_{n \rightarrow +\infty} \inf d(f^n(x), f^n(y)) = 0$ et $\lim_{n \rightarrow +\infty} \sup d(f^n(x), f^n(y)) > 0$

Définition 2.2 : *Chaos au sens de Devaney*

Soit V un ensemble. Une application continue $f : V \rightarrow V$ est dite chaotique sur V si :

- f est topologiquement transitif : pour toute paire d'ensembles ouverts non vides $U, W \subset V$ il existe un $K > 0$ tel que $f^K(U) \cap W \neq \emptyset$.

- Les points périodiques de f sont denses dans V .
- f possède une dépendance sensible aux conditions initiales : il existe un $\delta > 0$ tel que, pour tout $x \in V$ et tout voisinage $v(x)$, ils existent un $y \in v(x)$ et un $n \geq 0$ tel que $|f^n(x) - f^n(y)| > \delta$

2.1.2 Caractéristiques du chaos

Cependant, Le chaos est défini généralement comme un comportement particulier d'un système dynamique qui inclut :

- non-linéarité : l'évolution irrégulière du comportement d'un système chaotique est due aux non linéarités.
- le déterminisme : un système chaotique a des règles fondamentales déterministes et non probabilistes.
- la sensibilité : La sensibilité aux conditions initiales est un phénomène découvert pour la première fois, à la fin du *XIXe* siècle par Poincaré, puis a été redécouvert en 1963 par Lorenz lors de ses travaux en météorologie. Cette découverte a entraîné un grand nombre de travaux importants, principalement dans le domaine des mathématiques. Cette sensibilité explique le fait que, pour un système chaotique, une modification infime des conditions initiales peut entraîner des résultats imprévisibles sur le long terme. Le degré de sensibilité aux conditions initiales quantifie le caractère chaotique du système.
- l'imprévisibilité : en raison de la sensibilité aux conditions initiales, qui peuvent être connues seulement à un degré fini de précision.
- l'irrégularité : l'ordre caché comprenant un nombre infini de modèles périodiques instables. Cet ordre caché forme l'infrastructure des systèmes chaotiques Le comportement chaotique peut être observé dans de nombreux systèmes naturels, comme les conditions météorologiques et le climat. Ce comportement peut être étudié par l'analyse d'un modèle mathématique chaotique, ou par des techniques d'analyse telles que des parcelles de récurrence et de section de Poincaré. Dans ce présent chapitre, nous donnons une brève étude quantitative sur le chaos.

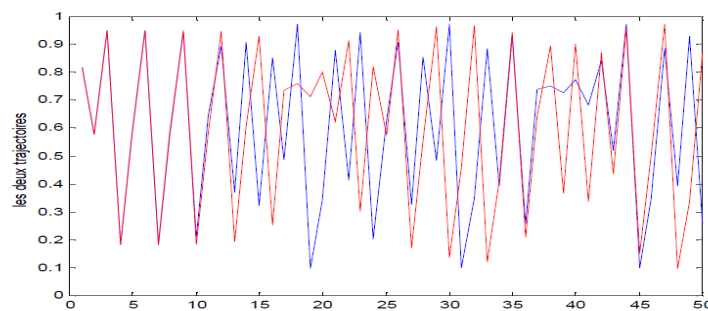


FIGURE 2.1 – La sensibilité aux conditions initiales.

2.1.3 Transitions vers le chaos

Il existe plusieurs scénarios qui décrivent le passage vers le chaos. On constate dans tous les cas que l'évolution du point fixe vers le chaos n'est pas progressive, mais marquée par des changements discontinus qu'on a déjà appelé bifurcations. On peut citer trois scénarios de transition d'une dynamique régulière à une dynamique chaotique lors de la variation d'un paramètre :

Cascade de doublements de période

Ce scénario a été observé dans les années 60 par R. May en dynamique de populations sur l'application logistique. Ce scénario est caractérisé par une succession de bifurcation fourches. A mesure que la contrainte augmente, la période d'un système forcé est multipliée par 2, puis par 4, puis par 8, etc..., ces doublements de périodes sont de plus en plus rapprochés, lorsque la période est infinie, le système devient chaotique.

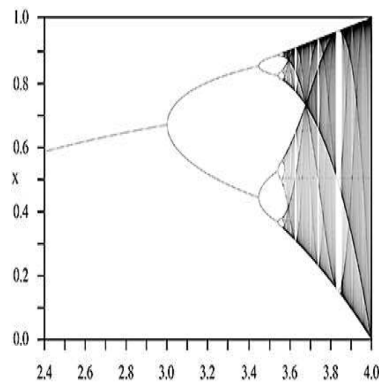


FIGURE 2.2 – Transition vers le chaos via le doublement de période.

Par intermittence

Un mouvement périodique stable est entrecoupé par des bouffées chaotiques. Lorsqu'on augmente le paramètre de contrôle, les bouffées de turbulence deviennent de plus en plus fréquentes, et finalement, la turbulence domine.

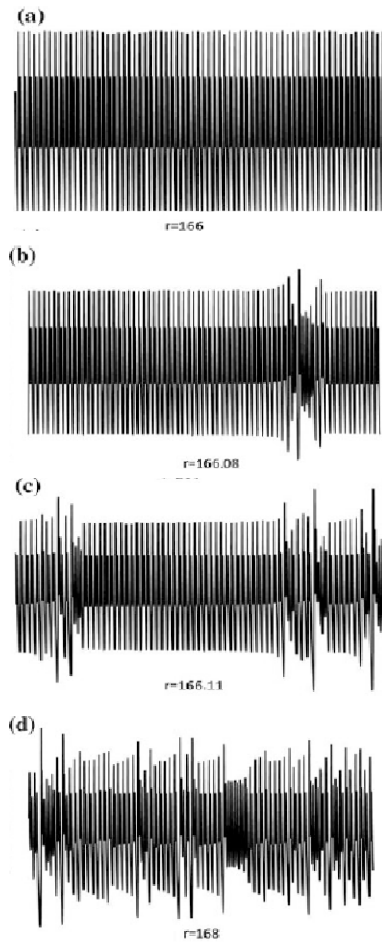


FIGURE 2.3 – Transition vers le chaos par l’intermittence.

Scénario de Ruelle et Takens

Ce scénario via la quasi-périodicité a été mis en évidence par les travaux théoriques de Ruelle et Takens. Dans un système dynamique à comportement périodique à une seule fréquence, si nous changeons un paramètre alors il apparaît une deuxième fréquence. Si le rapport entre les deux fréquences est rationnelle comportement est périodique. Mais, si le rapport est irrationnel, le comportement est le quasi périodique. Alors, on change de nouveau le paramètre et il apparaît une troisième fréquence et ainsi de suite jusqu’au chaos.

2.2 Exposant de Lyapunov

L'évolution d'un flot chaotique est difficile à appréhender, parce que la divergence des trajectoires sur l'attracteur est rapide, C'est pourquoi on essaye d'estimer ou même de mesurer la vitesse de divergence ou convergence, Cette vitesse s'appelle l'exposant lyapunov. L'exposant de Lyapunov sert à mesurer le degré de stabilité d'un système et de permet de quantifier la sensibilité aux conditions initiales d'un système chaotique. Le nombre des exposants de Lyapunov est égal à la dimension de l'espace des phases et ils sont généralement indexés du plus grand au plus petit. L'apparition du chaos exige que les exposants de Lyapunov doivent remplir trois conditions :

- Au moins l'un d'eux est positif pour expliquer la divergence des trajectoires.
- Au moins l'un d'eux est négatif pour justifier le repliement des trajectoires.
- La somme de tous les exposants est négative pour expliquer qu'un système chaotique est dissipatif, c'est-à-dire qu'il perd de l'énergie.

La valeur du plus grand exposant de Lyapunov quantifie le degré de chaos du système, mais le fait que les trois conditions énoncées ci dessus soient réunies ne suffit pas à conclure qu'un système est chaotique. Il demeure indispensable de confronter les résultats du calcul des exposants de Lyapunov avec ceux fournis par d'autres outils d'analyse non-linéaire.

2.2.1 Cas des systèmes discrets unidimensionnelle

Soit x_0 et $x_0 + \varepsilon$ deux conditions initiales, supposons qu'elle s'écartent en moyenne exponentiellement c'est à dire il existe un réel λ tel que après N itérations on a :

$$|f^N(x_0 + \varepsilon) - f^N(x_0)| \simeq \varepsilon \exp^{N\lambda},$$

alors

$$\lambda \simeq \frac{1}{N} \ln \frac{|f^N(x_0 + \varepsilon) - f^N(x_0)|}{\varepsilon},$$

lorsque $N \rightarrow +\infty$ alors :

$$\lambda = \lim_{n \rightarrow +\infty} \frac{1}{N} \ln \frac{|f^N(x_0 + \varepsilon) - f^N(x_0)|}{\varepsilon},$$

et pour $\varepsilon \rightarrow 0$, on obtient :

$$\lambda = \lim_{n \rightarrow +\infty} \frac{1}{N} \ln \left(\left| \frac{df^N(x_0)}{dx_0} \right| \right),$$

donc on a :

$$\lambda = \lim_{n \rightarrow +\infty} \frac{1}{N} \ln |(f^N)'(x_0)|,$$

finalement :

$$\lambda = \lim_{n \rightarrow +\infty} \sum_{i=0}^{n-1} \frac{1}{N} \ln (|f'(x_i)|). \quad (2.1)$$

Exemple :

L'application logistique :

$$f(x_i) = 4x_i(1 - x_i), x_i \in [0, 1]$$

Donc, l'exposant de Lyapunov de f est défini par :

$$\lambda = \lim_{n \rightarrow +\infty} \sum_{i=0}^n \frac{1}{N} \ln |4(1 - 2x_i)|.$$

Alors : on effectuant quelques changement et on utilisant quelques techniques de majoration et de minoration on trouve $\lambda = \ln 2 > 0$ d'où le comportement est chaotique. [15].

2.2.2 Cas des systèmes discrets multidimensionnels

Dans ce paragraphe nous allons généraliser les concepts du paragraphe précédent à des trajectoires multidimensionnelles de type :

$$f : R^p \rightarrow R^p, x_{n+1} = f(x_n)$$

Comme précédemment on s'intéresse à :

$$f^n(x_0 + \varepsilon) - f^n(x_0) \simeq \varepsilon e^{n\lambda}$$

Écrivons un développement en série limitée d'ordre 1 de $f^n(x_0)$ au voisinage de x'_0 :

$$\begin{aligned} x_n - x'_n &\simeq f^n(x_0) - f^n(x'_0) \\ &\simeq \frac{df^n(x_0)}{dx} (x - x_0) \\ &\simeq J(x_0)J(x_1)\dots J(x_n)(x_0 - x'_0) \\ &\simeq \prod_{i=0}^{n-1} J(x_i)(x_0 - x'_0) \end{aligned}$$

On note $\prod_{i=0}^{n-1} J(x_i)$ par $J^n(x_0)$, ainsi :

$$x_n - x'_n \simeq J^n(x_0)(x_0 - x'_0)$$

Où $J^n(x_0)$ représente la matrice Jacobienne de $f^n(\cdot)$ au point x_0 . Il s'agit d'une matrice carrée $p \times p$, si elle est diagonalisable, alors il existe une matrice inversible P_p telle que $D_p^n = P_p^{-1} J^n P_p$ est une matrice diagonale des valeurs propres $q_i(f^k(x_0)) (i = 1, \dots, p)$ de J^n .

On définit alors les p exposants de Lyapunov de la manière suivante :

$$\lambda_i = \lim_{n \rightarrow +\infty} \frac{1}{n} \sum_{j=0}^{j=n} \log q_j(f^k x_0)$$

λ_i représente l'exposant de Lyapunov associé à la trajectoire issue de x_0 dans la $i^{\text{ème}}$ direction. Ainsi

$$x_0 - x'_0 = (J(x^*))^n(x_n - x'_n)$$

Tel que x^* est un point d'équilibre d'où :

$$\|x_{i,n} - x'_{i,n}\| = q_i(x^*)$$

$$^n \|x_{i,0} - x'_{i,0}\|,$$

Et puisque :

$$\|x_{i,n} - x'_{i,n}\| = e^{n\lambda_i} \|x_{i,0} - x'_{i,0}\|$$

Par comparaison avec l'égalité précédente on obtient :

$$e^{\lambda_i n} = q_i(x^*)^n$$

D'où :

$$\lambda_i = \log q_i(x^*), i = 1, 2, \dots, p. \quad (2.3)$$

Exemple :

La transformation de boulanger sur $E = [0, 1] \times [0, 1]$ définit par :

$$f(x, y) = \begin{cases} x \rightarrow 2x(\text{mod } 1), \text{ pour } 0 \leq x < \frac{1}{2} \\ y \rightarrow \begin{cases} \frac{1}{2}ay, \text{ pour } 0 \leq x < \frac{1}{2} \\ \frac{1}{2}(ay + 1), \text{ pour } \frac{1}{2} \leq x < 1 \end{cases} \end{cases} \quad (0 \leq x \leq 1)$$

$(0,0)$ est le seul point fixe de f dont la jacobienne J est donnée par :

$$J = \begin{pmatrix} 2 & 0 \\ 0 & \frac{a}{2} \end{pmatrix}$$

Les valeurs propres de J de $f^n(x_0)$ au point $x_0 = (0, 0)$ sont :

$$q_1(f^n(x_0)) = 2 \text{ et } q_2(f^n(x_0)) = \frac{a}{2} (\forall n \in \mathbb{N})$$

Appliquant la limite dans deux directions on obtient les exposants de Lyapunov :

$$\lambda_1 = \lim_{n \rightarrow +\infty} \frac{1}{n} \sum_{i=0}^{i=n} \ln 2$$

$$\lambda_2 = \lim_{n \rightarrow +\infty} \frac{1}{n} \sum_{i=0}^{i=n} \ln \frac{a}{2}$$

D'où :

$$\lambda_1 = \ln 2 \text{ et } \lambda_2 = \frac{a}{2}.$$

[15].

2.2.3 Cas des systèmes continus multidimensionnels

Pour un système différentiel de dimension n défini par f tel que :

$$\dot{x} = f(x(t)) \text{ tel que : } t \in \mathbb{R}, x(t) \in \mathbb{R}^n$$

L'exposant de Lyapunov dans la direction i est donné par :

$$\lambda_i = \lim_{t \rightarrow +\infty} \frac{1}{t} \ln \frac{\|x_i(t) - x_i'(t)\|}{\|x_i(0) - x_i'(0)\|} \quad (2.4)$$

Exemple :

soit le système de Lorenz :

$$\begin{cases} \dot{x} &= 10(y - x) \\ \dot{y} &= 28x - y - xz \\ \dot{z} &= xy - \frac{8}{3}z \end{cases}$$

[15]. Les exposants de Lyapunov pour une condition initiale x_0 choisie sont :

$$\lambda_1 \simeq 2.16, \lambda_2 \simeq 0.00, \lambda_3 \simeq -32.40$$

Les exposants de Lyapunov permettent donc de quantifier la sensibilité aux conditions initiales (SCI) mais aussi de séparer les comportements instables ou chaotiques des comportements stables et prévisibles. Si un exposant de Lyapunov est strictement positif, alors la SCI est très grande et le système peut être considéré comme chaotique. Par contre, s'ils sont tous négatifs ou égaux à zéro, on est en présence d'un phénomène stable ou périodique. Pour une application multidimensionnelle on peut résumer la correspondance entre le type de l'attracteur et le signe des exposants de Lyapunov dans le tableau ci dessous :

Type d'attracteurs	Exposant de Lyapunov
Point fixe	$0 > \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$
Cycle	$\lambda_1 = 0, \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$
Tore	$\lambda_1 = \lambda_2 = 0, 0 > \lambda_3 \geq \lambda_4 \dots \geq \lambda_n$
K-Tore	$\lambda_1 = \lambda_2 = \lambda_k = 0, 0 > \lambda_{k+1} \geq \lambda_{k+2} \geq \dots \geq \lambda_n$
Attracteur étrange	$\lambda_1 > 0, \sum \lambda_i < 0$

TABLE 2.1 – Type d'attracteurs d'après l'exposant de Lyapunov.

Chapitre 3

Algorithmes pour le calcul des exposants de Lyapunov et applications

3.1 Algorithmes

Il y a plusieurs méthodes pour déterminer les exposants de Lyapunov, chacune a son algorithme qui on peut distinguer on deux types différentes : trajectoires, ou méthodes directes(Wolf et al., 1985 ; Rosenstein et al., 1993 ; Kantz, 1994) ; et perturbation, l'espace tangent ou la méthode de la matrice jacobienne(Sano Sawada, 1985 ; Eckmann et al., 1986 ; Brown et al., 1991. Briggs, 1990 ; Krueel et al., 1993). On va étudier quatre méthodes.

3.1.1 Algorithme de Wolf pour ODE :

On considère un système dynamique continue a n dimension :

$$\frac{dZ}{dt} = F(Z, t). \quad (3.1)$$

On définit une sphère infinitésimal autour $Z(0)$ qui est la condition initial par :

$$\frac{d\delta Z}{dt} = J\delta Z. \quad (3.2)$$

ou J est la matrice jacobienne $n \times n$.

La solution de (3.2)est donné par :

$$\delta Z(t) = M(Z(t), t)\delta Z(0). \quad (3.3)$$

On explique la méthode standard pour comprendre l'idée pour calculer les exposants de Lyapunov. On choisit les n vecteurs orthogonales a $Z(0)$, le chois standard est :

$$e_1(0) = (1, 0, 0, \dots); e_2(0) = (0, 1, 0, \dots), \dots$$

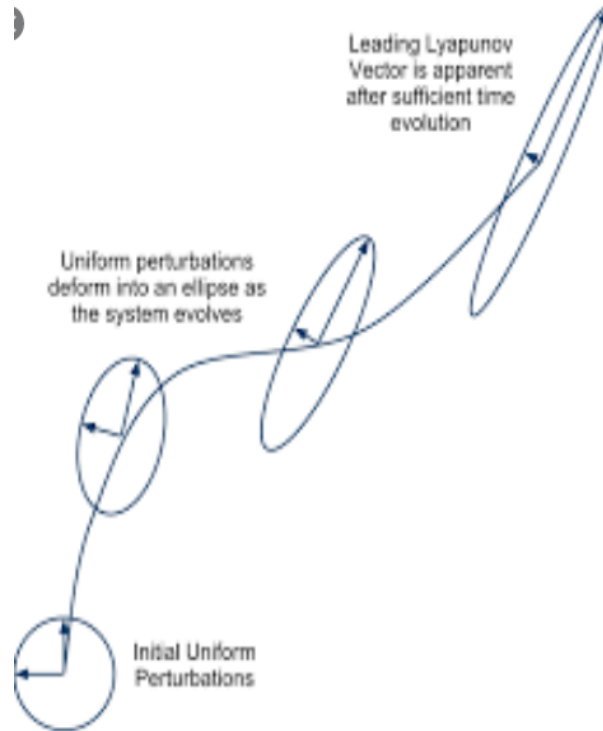


FIGURE 3.1 – La forme de la sphère après le déplacement.

Et avec le déplacement a un temps τ , la sphère précédente est former une ellipse comme il est représenter dans la figure 3.1.

Donc il suffit de reorthogonaliser les vecteurs $e_i(\tau)$ avec procédure de Gram-Schmidt [23].

$$e_1(\tau) = \frac{v_1}{\|v_1\|}, e_2(\tau) = \frac{v_2 - \langle v_2, e_1(\tau) \rangle e_1(\tau)}{\|v_2 - \langle v_2, e_1(\tau) \rangle e_1(\tau)\|}, \dots \quad (3.4)$$

tel que : les v_i sont des vecteurs.

Les normes précédent sont notés par $N_1^1, N_2^1, \dots, N_n^1$, et pour 2τ les $e_i(2\tau)$, les $v_i(2\tau)$ et les normes $N_1^2, N_2^2, \dots, N_n^2$, etc jusqu'à r itération, on définit les exposants de Lyapunov par :

$$\lambda_i = \lim_{r \rightarrow \infty} \frac{\sum_{m=1}^r \ln N_i^m}{r\tau} \quad (3.5)$$

3.1.2 Algorithme de Wolf pour une série temporelle

L'un des algorithmes utilisé pour une calcul est celui appelé algorithme de Wolf. Celui-ci permet de calculer des exposants de Lyapunov à partir du calcul effectif de la

divergence de deux trajectoires après t pas de temps. L'algorithme fait en plus varier un paramètre de contrôle choisi pour permettre de visionner l'évolution de la sensibilité aux conditions initiales selon ce dernier. Il faut remarquer que l'algorithme de Wolf calcule l'exposant de Lyapunov le plus grand. Pour ce faire, il fait la moyenne des sommes des divergences normalisées, et puisque l'exposant le plus grand aura la plus grande influence sur le système.

Les étapes de l'algorithme sont les suivantes :

1. Changement du paramètre de contrôle,
2. Choix aléatoire d'une condition initiale,
3. Évolution du système dans le but d'atteindre un attracteur,
4. Création d'une nouvelle trajectoire à partir de la trajectoire courante à laquelle on ajoute une petite perturbation,
5. Évolution dans l'attracteur de ces deux trajectoires voisines et calcul de la moyenne de la divergence renormalisée entre ces deux trajectoires,
6. Réajustement de l'écart, permettant ainsi à chaque pas de temps de l'évolution du point précédant le calcul d'une moyenne de la divergence,
7. Retour au point (5) effectué selon un nombre donné,
8. Retour au point (1),
9. Dessin de l'exposant de Lyapunov le plus grand en fonction du paramètre de contrôle donné.

L'évolution expérimentale scalaire est ainsi traduite en un ensemble de trajectoires vectorielles dans un espace des phases de dimension N . Des trajectoires voisines sont recherchées dans les données disponibles, et l'on observe la façon dont elles s'éloignent l'une de l'autre, on choisit une nouvelle trajectoire qui est plus proche que l'ancien. Moyennant quelques techniques de recalage de l'orientation des vecteurs perturbation. La figure suivante qui illustre ce principe en représentant 4 portions de trajectoires voisines d'une trajectoire de référence.

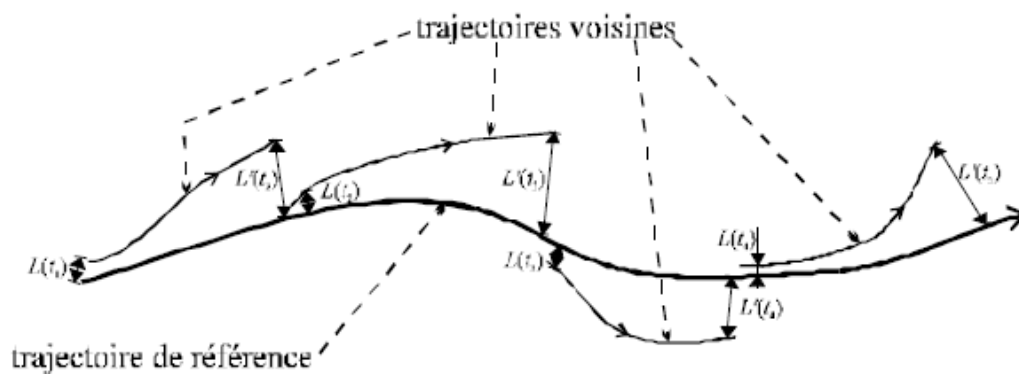


FIGURE 3.2 – Calcul des exposants de Lyapunov à partir de 2 trajectoires voisines de l'espace des phases.

Lorsqu'une trajectoire voisine s'éloigne trop de la trajectoire de référence, son écar-

tement n'est plus significatif par rapport à la trajectoire de référence, et il faut alors chercher dans les données expérimentales une nouvelle trajectoire plus proche. En répétant ces opérations sur une orbite complète de la trajectoire de référence, il est possible d'évaluer la propriété moyenne de SCI, c'est-à-dire le premier exposant de Lyapounov. Compte tenu des notations de la figure (3.2), et d'un nombre M de trajectoires voisines nécessaires au parcours d'une orbite de la trajectoire de référence, λ_1 a pour expression :

$$\lambda_1 = \frac{1}{t_m - t_0} \sum_{k=1}^M \ln \left[\frac{L'(t_k)}{L(t_{k-1})} \right] \quad (3.6)$$

Les autres exposants de Lyapounov sont également accessibles par une méthode analogue faisant intervenir des paires, des triplets, ..., de trajectoires voisines, mais les résultats sont beaucoup plus sensibles à la qualité des relevés expérimentaux. Wolf et d'autre ont proposé aussi un algorithme pour le calcul de la somme des deux plus grands exposants. Le principe est similaire : ils prennent trois trajectoires voisines et calculent l'aire du triangle de conditions initiales, soit $A(t_0)$. En un temps ultérieur t_1 , cette aire est devenue $A'(t_1)$. On effectue alors un remplacement, et on obtient une nouvelle aire de conditions initiales $A(t_1)$. On recommence alors ces opérations un grand nombre de fois pour les temps t_1, \dots, t_M et on calcule l'estimateur de la somme des deux plus grands exposants [18] :

$$\lambda_1 + \lambda_2 = \frac{1}{t_M - t_1} \sum_{k=0}^M \ln \left[\frac{A'(t_k)}{A(t_{k-1})} \right] \quad (3.7)$$

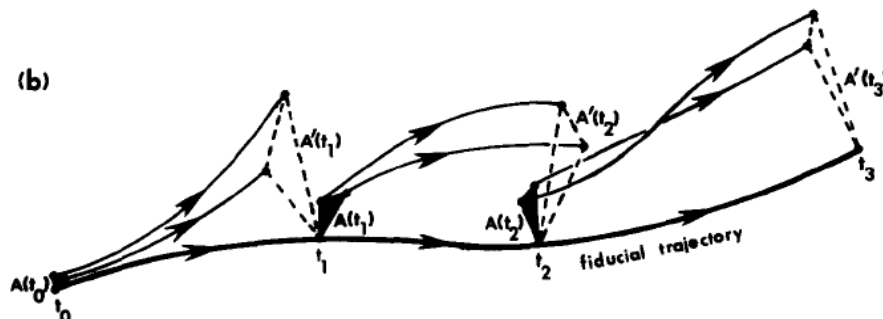


FIGURE 3.3 – Calcul des deux plus grands exposants de Lyapounov à partir de 3 trajectoires voisines de l'espace des phases.

Et finalement, malgré la méthode est applicable à plusieurs systèmes, il y a des systèmes pour lesquels il est difficile de l'appliquer. de plus, elle est très sensible aux entrées et peut facilement conduire à un résultat erroné.

3.1.3 Algorithme de Kantz

L'algorithme de Kantz (1994) explore directement la divergence des trajectoires voisines offrant la possibilité de décider de la faisabilité de l'estimation de l'exposant avant de le calculer. En effet la divergence exponentielle peut être entachée de fortes fluctuations de diverses origines. Par exemple, la mesure de l'exposant intègre une projection de la trajectoire dans l'espace de phase, et suivant l'angle adopté, cette projection peut donner l'illusion d'une convergence alors que la trajectoire subit réellement une divergence. Ainsi, le plus grand exposant de Lyapunov estimé, qui est une moyenne des différentes divergences et convergences locales [19].

Après la reconstruction de l'espace de phase, les étapes de l'algorithme se résument à choisir un point s_{k_0} dans les vecteurs résultants, de chercher le voisinage $U(s_{k_0})$, c.-à-d. tous les voisins s_k dans une distance ϵ de s_{k_0} .



FIGURE 3.4 – La reconstruction de l'espace de phase d'après l'algorithme de Kantz.

Et donc :

$$S(\Delta k) = \frac{1}{N} \sum_{k_0=1}^N \ln\left(\frac{1}{|U(s_{k_0})|} \sum_{s_K \in U(s_{k_0})} |s_{k_0+\Delta k} - s_{k+\Delta k}|\right) \quad (3.8)$$

$S(\Delta k)$ est une mesure du taux effectif de l'expansion durant un temps Δk répété pour plusieurs k_0 . La distance ϵ doit être choisie aussi petite que possible, mais assez grande pour que chaque point sur la trajectoire puisse avoir un voisinage non vide, sinon des parties de la trajectoire ne seront pas représentées dans la somme, donnant ainsi de faux résultats.

Les étapes de cet algorithme sont les suivantes :

- . Choisit point s_{k_0} sur la trajectoire.
- . Regardez les points autour s_{k_0} .
- . Mesurer à quelle distance de sont de s_{k_0} .
- . Moyenne de ces distances.
- . Regardez comment cette moyenne augmente avec le temps.
- . Prenez le log, normalisez de $S(\Delta k)$.
- . Répétez pour beaucoup de points s_{k_0} et faites la moyenne de $S(\Delta k)$.
- . Graphique $S(\Delta k)$ en fonction de n .
- . Recherchez une région de mise à l'échelle et une ligne vers celle-ci; cette pente est λ_1 .

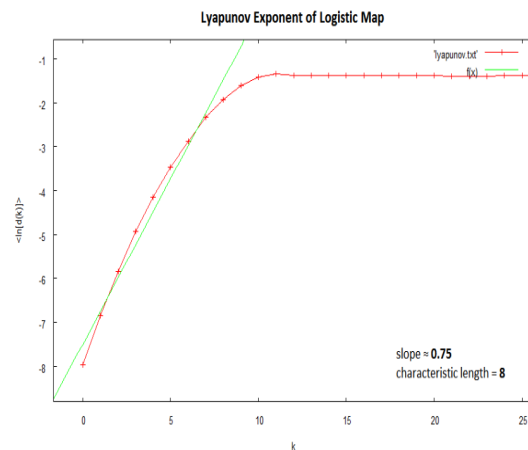


FIGURE 3.5 – Estimation de l'exposant maximal de Lyapunov pour l'application logistique par la méthode de Kantz.

Cet comme tous les méthodes il' y a toujours des problèmes, le choix de la région et des paramètres car l'algorithme a plusieurs paramètres libres : ϵ , par exemple, combien de points s_{k_0} on choisit, et plusieurs autres.

L'interprétation des résultats n'est pas délicate puisqu'il n'y a pas de définition claire de ce qui constitue une région d'échelle.

Les auteurs estiment que cet outil n'est utile que si l'on sait quelle valeur d'exposant de Lyapunov est souhaitée et peut ainsi choisir la région présentant une pente égale à cette valeur.

3.1.4 Algorithme de la limite inférieure d'erreur (LLE)

L'idée de cette méthode est de dessiner la courbe de la limite inférieure d'erreur lorsque on considère deux extensions intervalles de fonctions. Et pour comprendre la méthode, on présente premièrement quelques définitions.

Définition 3.1.1. Une extension d'intervalle de f est une fonction de valeur d'intervalle d'une variable d'intervalle X , avec la propriété :

$$F(x) = f(x) \text{ pour arguments réels} \quad (3.9)$$

Où, par un intervalle, il est censé être un ensemble fermé de nombres réels $x \in \mathbb{R}$ tels que $X = [\underline{X}, \overline{X}] = \{x : \underline{X} \leq x \leq \overline{X}\}$.

Par exemple : on a le système suivant :

$$x_{n+1} = rx_n(1 - x_n)$$

On a

$$G(X_n) = rX_n - r \cdot X_n \cdot X_n$$

et

$$H(X_n) = rX_n - rX_n^2.$$

Sont deux extensions intervalles du système précédent.

Définition 3.1.1. Une orbite est une séquence de valeurs d'une fonction, représentée par le symbole $\{x_n\}$ ou par x_0, x_1, x_2, \dots . Il n'y a pas de pseudo-orbite unique, car il y a différents matériels, logiciels, précision numérique schémas standard et de discrétisation, qui peuvent produire une sortie différente pour chaque extension intervalle.

Définition 3.1.1. Soit $i \in \mathbb{N}$ une pseudo-orbite, qui est défini par une condition initiale, un extension d'intervalle de f , certains matériels spécifiques, logiciel, norme de précision numérique et schéma de discrétisation. Une pseudo-orbite est une approximation d'une orbite et peut être représenté par $\hat{x}_{i,n}$ ou $\hat{x}_{i,0}, \hat{x}_{i,1}, \hat{x}_{i,2}, \dots$, tel que

$$|x_n - \hat{x}_{i,n}| \leq \delta_{i,n}, \quad (3.10)$$

où $\delta_{i,n} \in \mathbb{R}$ est l'erreur et $\delta_{i,n} \geq 0$. Une pseudo-orbite définit un intervalle où la vraie orbite repose. D'où un intervalle associé à chaque valeur d'une pseudo-orbite peut être définie comme

$$I_{i,n} = [\hat{x}_{i,n} - \delta_{i,n}, \hat{x}_{i,n} + \delta_{i,n}] \quad (3.11)$$

D'après (10) et (11), il est clair que

$$x_n \in I_{i,n}, \text{ pour tout } i \in \mathbb{N}. \quad (3.12)$$

Théorème 3.1. Soit deux pseudo-orbites $\{\hat{x}_{a,n}\}$ et $\{\hat{x}_{b,n}\}$ dériver de deux extensions d'intervalle. et soit $\delta_{\alpha,n} = \frac{|\hat{x}_{a,n} - \hat{x}_{b,n}|}{2}$ est l'erreur de limite inférieure d'une fonction f , avec $\delta_{a,n} \geq \delta_{\alpha,n}$ ou $\delta_{b,n} \geq \delta_{\alpha,n}$.

La méthode peut être résumée dans les étapes suivantes :

- . Deux extensions d'intervalle d'un système.
- . Choisir exactement les mêmes conditions initiales, taille de pas et schéma de discrétisation, simuler les deux extensions d'intervalle.
- . Utilisez la méthode des moindres carrés pour ajuster une ligne au pente de la courbe ln de la valeur absolue de l'erreur d'algorithmme naturelle. la pente de la ligne est le EML [20].

3.2 Applications

On va simuler quatre modèles dynamiques qui sont : L'équation Logistique, le système de Lozi dans le cas discret et Lorenz et un système financier. La simulation est avec trois méthodes : la méthode de Wolf ODE, Wolf série temporelle (Wolf ST) et la limite inférieure d'erreur(LLE), avec les même conditions initiales et les même paramètres et le nombre de points de calcul dans le langage Matlab.

Les programmes Matlab utilisés dans ce mémoire sont dans l'annexe.

Système	Équations	Paramètres	$\Delta t(s)$	Conditions initiales
Logistic	$x_{n+1} = rx_n(1 - x_n)$	$r=4.0$	1	$x(0) = 2/3$
Lozi	$x_{n+1} = 1 - a x_n + y_n$ $y_{n+1} = bx_n$	$a = 1.7$ $b = 0.5$	1	$x(0) = 0.3$ $y(0) = 0.3$
Lorenz	$\dot{x} = \sigma(y - x)$ $\dot{y} = x(\rho - z) - y$ $\dot{z} = xy - \beta z$	$\sigma = 16.0$ $\rho = 45.92$ $\beta = 4.0$	0.01	$x(0) = 1$ $y(0) = 0.5$ $z(0) = 0.9$
Finance	$\dot{x} = z + (y - a)x$ $\dot{y} = 1 - by - x^2$ $\dot{z} = x - cz$	$a = 2$ $b = 0.1$ $c = 1$	0.01	$x(0) = 2$ $y(0) = 3$ $z(0) = 2$

TABLE 3.1 – Quelques systèmes dynamiques et leurs paramètres et conditions initiales. La figure représente les attracteurs des quatre systèmes qui sont citer précédemment, et nous utilisons 10000 points de calcul.

3.2.1 Application de L'algorithmme de Wolf pour ODE :

Le programme permet de calculer les exposants de Lyapunov complets d'un système dynamique discret ou continue.

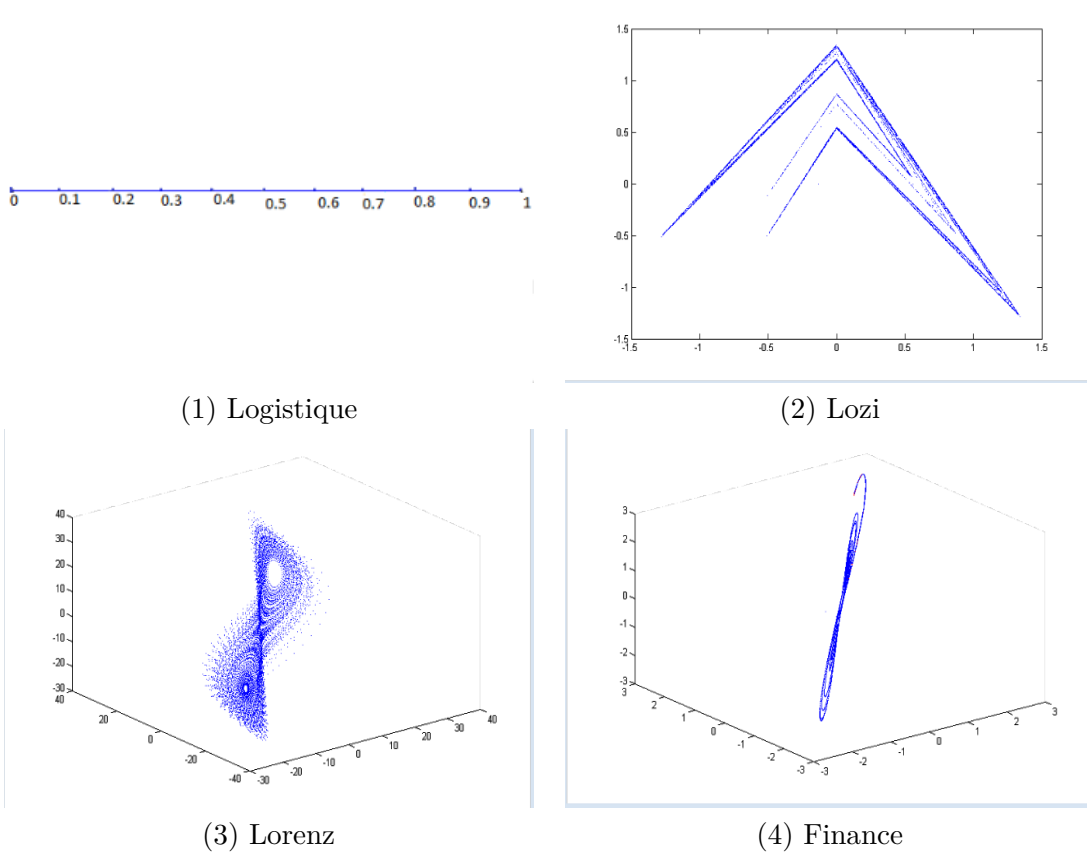
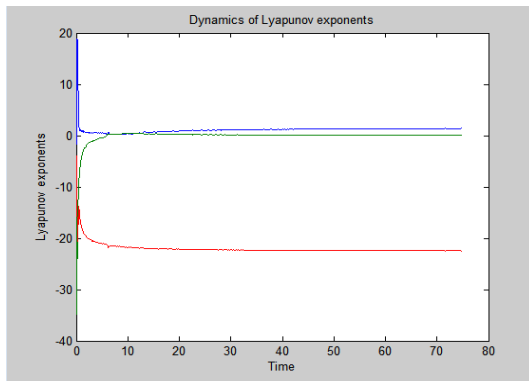


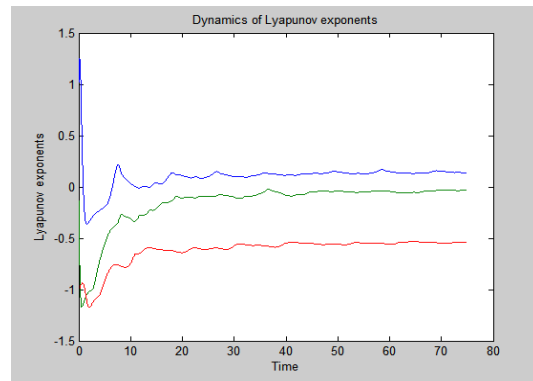
FIGURE 3.6 – Attracteurs de quelques systèmes.

Il composé de trois fichier Matlab appliquées sur le système de Lorenz, qui est facile de le modifier par d'autres systèmes.

la figure 3.7 représente les trois exposants du système de Lorenz et le système financier.



(3) Lorenz



(4) Finance

FIGURE 3.7 – Résultat du programme de Wolf (ODE).

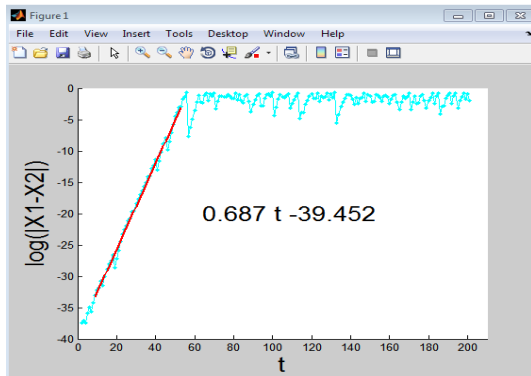
3.2.2 Application de L'algorithme de Wolf pour une série temporelle

Le programme suivant permet de calculer seulement le plus grand exposant de Lyapunov d'un système dynamique discret ou continue, ce système est défini par un fichier data (expérimentales data).

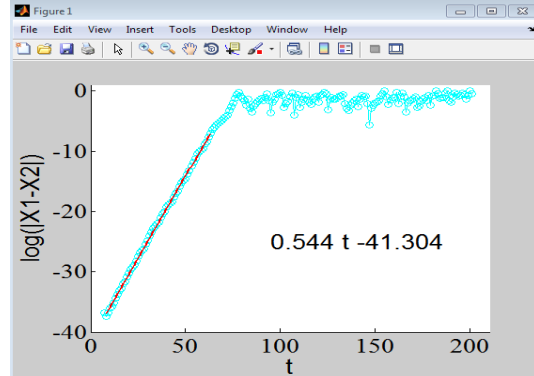
3.2.3 Application de L'algorithme LLE

Ce programme est simple a utiliser. Et aussi calcul une seul exposant de Lyapunov. Après exécution, on obtient le résultat suivant : tell que la pente de la ligne rouge est λ_1 .

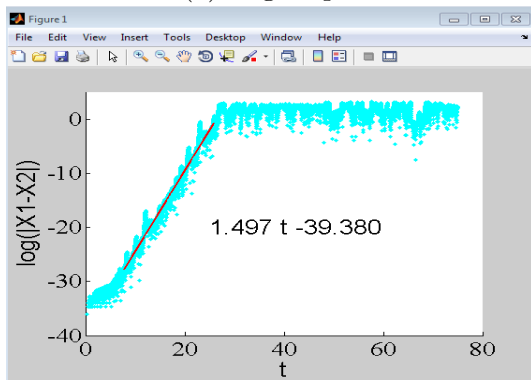
Comme représente la figure 3.8 tous les pentes de ces systèmes sont positives, et donc tous les systèmes sont chaotiques pour les valeurs des paramètres choisis.



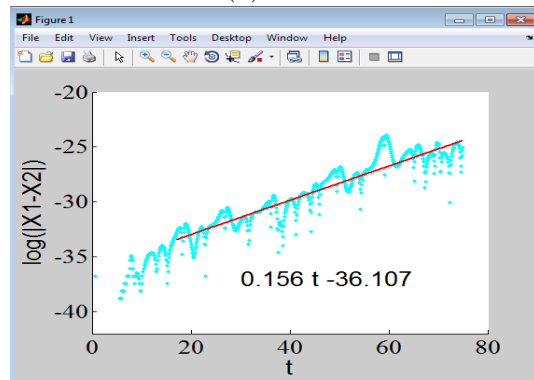
(1) Logistique



(2) Lozi



(3) Lorenz



(4) Finance

FIGURE 3.8 – Résultat de programme de la limite inférieure d'erreur.

Résultats des calculs effectués par Matlab :

Nous avons utilisé 10000 points pour obtenir les résultats qui sont organisée dans le tableau ci-dessous. la littérature λ d'équation Logistique se trouve dans [21] qui calcul par la méthode de Eckmann et Ruelle 1985, de système de Lozi dans [24] qui calcul par la méthode de différentiels équations, de Lorenz dans [25] par la méthode de Cayley, et de Finance dans [22].

le système	Wolf ODE	Wolf (ST)	LLE	littérature λ
Logistic	0.693	0.926	0.687	0.693
Lozi	0.670 -1.096	0.562	0.544	0.470
Lorenz	1.392 0.004 -22.439	1.924	1.497	1.485
Finance	0.138 -0.031 -0.544	2.815	0.156	0.022

TABLE 3.2 – Résultats de calculs des exposants de Lyapunov avec l’algorithme de Wolf ODE, Wolf (ST) et LLE.

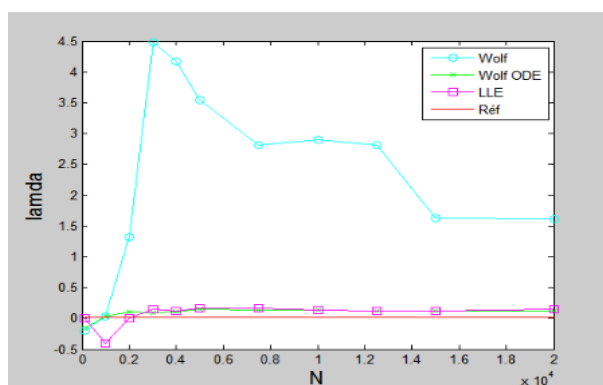


FIGURE 3.9 – Résultats des trois algorithmes : Wolf pour ODE, Wolf pour (ST) et LLE appliquée sur le système financier.

On remarque que le résultat de la méthode de Wolf pour (ST) pour un nombre petit de points de calcul n'est pas proche de la valeur de la littérature λ , par contre les deux méthodes Wolf pour ODE et LLE.

Alors, nous conseillons d'utiliser l'algorithme de Wolf pour ODE ou LLE pour calculer le plus grand exposant de Lyapunov, car elles donnent de bons résultats. Mais bien sûr si on veut calculer les exposants de Lyapunov complet d'un système, la méthode de Wolf pour ODE est une bonne candidate.

Les résultats des calculs dépendent beaucoup des choix comme le nombre de points de calcul, les paramètres du système, et aussi des conditions initiales, qui doivent être proches des points fixes.

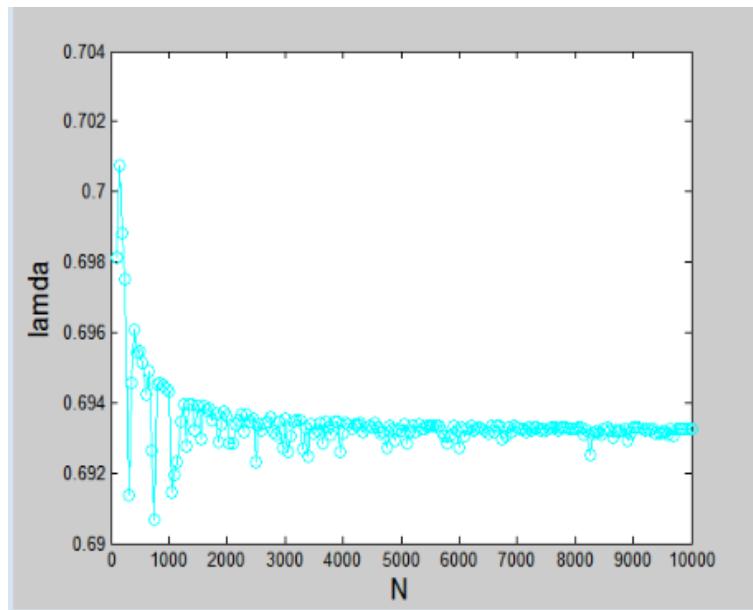


FIGURE 3.10 – L'exposant de Lyapunov de l'équation Logistique par rapport au nombre d'itérations.

On remarque que lorsque on augmente le nombre de points de calcul, le résultat tant vers la stabilité.

Donc, on doit utilisé un nombre suffisant de points pour estimer mieux l'exposant de Lyapunov. Cependant, l'utilisation d'un grand nombre de point implique un temps de calcul plus long et un espace mémoire plus grand.

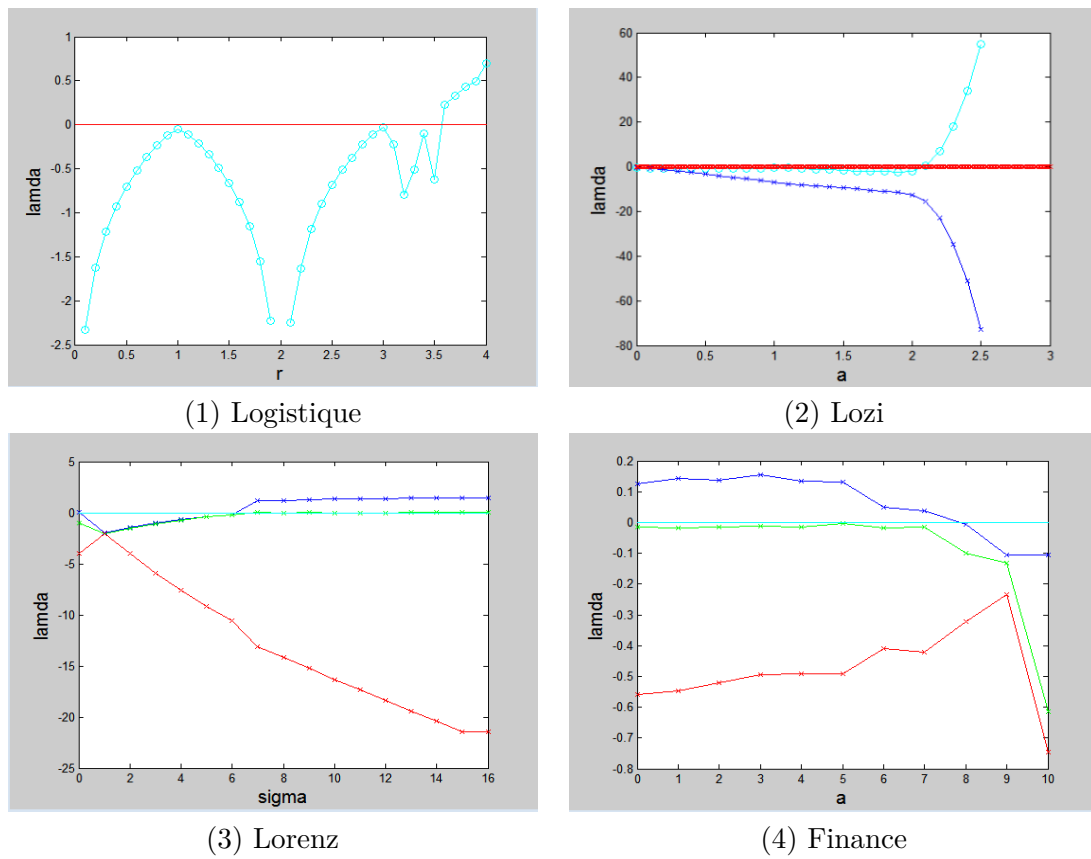


FIGURE 3.11 – Les exposants de Lyapunov par rapport au paramètre.

On remarque que pour tous les systèmes, il y a un intervalle de paramètre pour le système soit régulier et d'autre pour soit chaotique. par exemple, le système de Lorenz est régulier si $0 \leq \sigma < 6$, et chaotique si $6 \leq \sigma \leq 16$.

Nos résultats confirme l'influence du choix de paramètre d'un système sur le résultat du calcul de l'exposant de Lyapunov.

En plus des difficultés citées ci-dessus, ils existent d'autres telles que les paramètres dans la méthode de Wolf (ST) (tau, evolve, dismax...) et dans la méthode de LLE qui a le problème de l'intervalle de l'approximation de moindres carré qui risque de donner de faux résultats.

En effet, lors des calculs, nous avons constaté des résultats parfois très différents lorsqu'on

passer d'un ensemble de paramètres à un autre que se soit dans la méthode de Wolf (ST) ou dans la méthode LLE.

Conclusion

Le calcul des exposants de Lyapunov est un outil classique pour détecter le chaos dans un système dynamique déterministe.

Dans ce mémoire, nous avons étudiée trois algorithmes qui permettent de calculer le ou les exposants de Lyapunov d'un système dynamique discret ou continu. Ces algorithmes sont : l'algorithme de Wolf pour ODE, celui l'algorithme de Wolf pour (ST), l'algorithme de Kantz et l'algorithme LLE. Ensuite, nous avons appliquée ces algorithmes sur quatre systèmes dynamiques (Logistique, Lozi, Lorenz et un système financier).

L'algorithme de Wolf ODE permet de calculer les exposants de Lyapunov d'un système dynamique discret ou continue. Ce algorithme nécessite la donner des équations du système pour calculer ces exposants ; cet algorithme donne des résultats précis.

L'algorithme de Wolf (ST) calcule seulement le plus grand exposant de Lyapunov des données expérimentales d'un phénomènes, il est très utilisable dans la physique et la chimie où le système n'est pas généralement connu. Mais ces résultats comprendre des erreurs.

L'algorithme LLE aussi calcul le plus grand exposant de Lyapunov d'un système dynamique discret ou continue. Il permet de donner des résultats acceptables par rapport à la facilité d'utilisation.

Alors, chaque algorithme a des avantages et des inconvénients. Enfin, nous conséons par l'utilisation de ce dernier algorithme qui est l'algorithme de LLE a cause de ça faciliter, rapidité est ces résultats qui sont très proches de la vérité.

Nous avons présenté une étude comparative des résultats de l'estimation des exposants

de Lyapunov du système financier.

Enfin, nous devons préciser que le résultat que nous avons fourni dans ce travail ont été obtenus après utilisation de plusieurs tests sur de nombreux ensembles de paramètres.

Annexe

La figure (3.6) : (3)Lorenz.

```
function db = basgen(fname,tau,ndim,ires,datcnt,maxbox)
% Gn rateur de base de donn es pour la fonction fet.m
% Parc Taehyeun, The Cooper Union, EE'15
x = fileread(fname);
data = zeros(1,datcnt);
trck = 1;
start = 1;
fin = 0;

for ii = 1:length(x)
    if strcmp(x(ii),char(32)) || strcmp(x(ii),char(13))
        || strcmp(x(ii),char(10)) || strcmp(x(ii),char(26))
        if fin >= start
            data(trck) = str2num(x(start:fin));
            trck = trck + 1;
            if trck > 8*floor(datcnt/8)
                break
            end
        end
        start = ii + 1;
    else
        fin = ii;
    end
end

delay = 0:tau:(ndim-1)*tau;

nxtbox = zeros(maxbox, ndim);
where = zeros(maxbox, ndim);
datptr = zeros(1,maxbox);
```

```

nxtdat = zeros(1, datcnt);

datmin = min(data);
datmax = max(data);

datmin = datmin + 0.01*(datmax - datmin);
datmax = datmax - 0.01*(datmax - datmin);
boxlen = (datmax - datmin)/ires;

boxcnt = 1;

for ii = 1:(datcnt - (ndim - 1)*tau)
    target = floor((data(ii+delay) - datmin)/boxlen);
    runner = 1;
    chaser = 0;

    jj = 1;
    while jj <= ndim
        tmp = where(runner, jj) - target(jj);
        if tmp < 0
            chaser = runner;
            runner = nxtbox(runner, jj);
            if runner == 0
                continue
            end
        end
        if tmp == 0
            boxcnt = boxcnt + 1;

            if boxcnt == maxbox
                error
                ('Grid overflow, increase number of box count')
            end

            for kk = 1:ndim
                where(boxcnt, kk) = where(chaser, kk);
            end
            where(boxcnt, jj) = target(jj);
            nxtbox(chaser, jj) = boxcnt;
            nxtbox(boxcnt, jj) = runner;
            runner = boxcnt;
        end
        jj = jj + 1;
    end

```

```

    end
    nxtdat(ii) = datptr(runner);
    datptr(runner) = ii;
end

used = 0;
for ii = 1:boxcnt
    if datptr(ii) ~= 0;
        used = used + 1;
    end
end

display(['Created: ', num2str(boxcnt)]);
display(['Used: ', num2str(used)]);
db.ndim = ndim;
db.ires = ires;
db.tau = tau;
db.datcnt = datcnt;
db.boxcnt = boxcnt;
db.datmax = datmax;
db.datmin = datmin;
db.boxlen = boxlen;

db.datptr = datptr(1:boxcnt);
db.nxtbox = nxtbox(1:boxcnt, 1:ndim);
db.where = where(1:boxcnt, 1:ndim);
db.nxtdat = nxtdat(1:datcnt);
db.data = data;

```

```

function [out, SUM] =
    fet(db, dt, evolve, dismin, dismax, thmax)
% Calcule l'exposant de Lyapunov des données données (data)
% et paramètres,
% donne une sortie
% fichier texte, requête exacte de la version Fortran 77 de
% fet
% Parc Taehyeun, The Cooper Union, EE'15

out = [];

```

```

ndim = db.ndim;
ires = db.ires;
tau = db.tau;
datcnt = db.datcnt;
datmin = db.datmin;
boxlen = db.boxlen;

datptr = db.datptr;
nxtbox = db.nxtbox;
where = db.where;
nxtdat = db.nxtdat;
data = db.data;

delay = 0:tau:(ndim 1)*tau;
datuse = datcnt (ndim 1)*tau evolve;

its = 0;
SUM = 0;
savmax = dismax;

oldpnt = 1;
newpnt = 1;

fileID = fopen('fetout.txt', 'w');

goto50 = 1;
while goto50 == 1;
    goto50 = 0;
    [bstpnt, bstdis, thbest] = search(0,ndim,ires ,...
    datmin, boxlen, nxtbox, where, datptr, nxtdat ,...
    data, delay , oldpnt, newpnt, datuse, dismin ,...
    dismax thmax, evolve);
    while bstpnt == 0
        dismax = dismax * 2;
        [bstpnt, bstdis, thbest] = search(0,ndim,ires ,...
        datmin, boxlen, nxtbox, where, datptr, nxtdat ,...
        data, delay, oldpnt, newpnt, datuse, dismin ,...
        dismax, thmax, evolve);
    end

dismax = savmax;
newpnt = bstpnt;
disold = bstdis;

```

```

iang = 1;

goto60 = 1;
while goto60 == 1;
    goto60 = 0;

    oldpnt = oldpnt + evolve;
    newpnt = newpnt + evolve;
    if oldpnt >= datuse
        return
    end

    if newpnt >= datuse
        oldpnt = oldpnt + evolve;
        goto50 = 1;
        break
    end

    p1 = data(oldpnt + delay);
    p2 = data(newpnt + delay);
    disnew = sqrt(sum((p2 - p1).^2));

    its = its + 1;

    SUM = SUM + log(disnew/disold);
    zlyap = SUM/(its*evolve*dt*log(2));
    out = [out; its*evolve, disold, disnew, zlyap,
          (oldpnt + evolve), (newpnt + evolve)];

    if iang == 1
        fprintf(fileID,
            '%d\t\t\t%8.4f\t\t%8.4f\t\t%8.4f\n',
            out(end,1:4)');
    else
        fprintf(fileID,
            '%d\t\t\t%8.4f\t\t%8.4f\t\t%8.4f\t\t%d\n',
            [out(end,1:4), iang]');
    end

    if disnew <= dismax
        disold = disnew;
        iang = 1;
        goto60 = 1;
        continue
    end

```

```

end

[bstpnt, bstdis, thbest] = search(1, ndim, ires, ...
datmin, boxlen, nxtbox, where, datptr, nxdat, ...
data, delay, oldpnt newpnt, datuse, dismin, ...
dismax thmax, evolve);
if bstpnt ~= 0
    newpnt = bstpnt;
    disold = bstdis;
    iang = floor(thbest);
    goto60 = 1;
    continue
else
    goto50 = 1;
    break;
end
end
end
fclose(fileID);

```

```

function [] = makeplot(db, out, evolve, loc)
% Trace l' evolution des attracteurs 2D ou 3D par evolution ,
% Le 4 me param tre est le
% emplacement de la l gende
% Parc Taehyeun, The Cooper Union, EE'15

datcnt = db.datcnt;
ndim = db.ndim;
tau = db.tau;
dataplot = [];
freerun = 0;

delay = 0:tau:(ndim 1)*tau;
data = db.data;

for ii = 1:(datcnt (ndim 1)*tau)
    dataplot = [dataplot; data(ii+delay)];
end

figure, bar(out(:,1),out(:,3)), hold on;
mle = max(dataplot(:)) min(dataplot(:));

```



```

plot([0, out(end,1)], [mle, mle], 'r', 'LineWidth',
1.5), hold off;
set(gca, 'YTick', [0, mle])
axis([0, out(end,1), 0, 1.1*mle])
title('d_f_of_evolution_scaled_to_the_maximum_linear
extent_of_the_attractor')

if ndim == 2
    figure('Position', [100, 100, 800, 500]);
    plot(dataplot(:,1), dataplot(:,2), '.',
        'MarkerSize', 3), hold on;
    display('To see the next evolution, press enter')
    display('To clear the screen and then see the
    next evolution, type c and press enter')
    display('To proceed without stopping, type r and
    press enter')
    display('To terminate plot generating, type g
    and press enter')
    for ii = 1:size(out,1)
        if freerun == 0
            RESET = input('Next evolution?', 's');
            if strcmp(RESET, 'c')
                display('Screen cleared')
                hold off;
                clf;
                plot(dataplot(:,1), dataplot(:,2), '.',
                    'MarkerSize', 3), hold on;
            elseif strcmp(RESET, 'r')
                display('Evolving without stopping...')
                display('Press ctrl+c to terminate')
                freerun = 1;
            elseif strcmp(RESET, 'g')
                display('Plot generating stopped')
                return;
            else
                if ii > 1
                    delete(ann)
                end
            end
        end
    end
end

tmpold = out(ii,5);
oldpnt = tmpold + evolve;

```

```

tmpnew = out(ii,6);
newpnt = tmpnew + evolve;

plot(data(tmpold:oldpnt), data((tmpold+tau):
    (oldpnt+tau)), 'r', 'LineWidth', 1);
plot(data(tmpnew:newpnt), data((tmpnew+tau):
    (newpnt+tau)), 'g', 'LineWidth', 1);
for aa = 0:evolve;
    plot([data(tmpold+aa), data(tmpnew+aa)],
        [data(tmpold+aa+tau), data(tmpnew+aa+tau)],
        'LineWidth', 1)
end

ann = legend(['Iteration: ', num2str(out(ii,1)),
    '/ ', num2str(out(end,1)), char(10)...
    'd_i: ', num2str(out(ii,2)), char(10)...
    'd_f: ', num2str(out(ii,3)), char(10)...
    'Current Estimate: ' num2str(out(ii,4))] ,...
    'location', loc);
if freerun == 1
    drawnow
end
end
elseif ndim == 3
    figure('Position', [100, 100, 800, 500]);
    plot3(dataplot(:,1), dataplot(:,2), dataplot(:,3),
        '.'), 'MarkerSize', 3), hold on;
    display('To see the next evolution, press enter')
    display('To clear the screen and then see the next
    evolution, type c and press enter')
    display('To proceed without stopping,
    type r and press enter')
    display('To terminate plot generating,
    type g and press enter')

for ii = 1:size(out,1)
    if freerun == 0
        RESET = input('Next evolution? ', 's');
        if strcmp(RESET, 'c')
            display('Screen cleared')
            hold off;
            clf;
            plot3(dataplot(:,1), dataplot(:,2),

```

```

        dataplot(:,3),'.', 'MarkerSize', 3),
        hold on;
    elseif strcmp(RESET, 'r')
        display('Evolving without stopping...')
        display('Press ctrl+c to terminate')
        freerun = 1;
    elseif strcmp(RESET, 'g')
        display('Plot generating stopped')
        return;
    else
        if ii > 1
            delete(ann)
        end
    end
end

tmpold = out(ii,5);
oldpnt = tmpold + evolve;
tmpnew = out(ii,6);
newpnt = tmpnew + evolve;
plot3(data(tmpold:oldpnt), data((tmpold+tau):
(oldpnt+tau)), data((tmpold+(2*tau)):
oldpnt+(2*tau))), 'r', 'LineWidth', 1);

plot3(data(tmpnew:newpnt), data((tmpnew+tau):
(newpnt+tau)), data((tmpnew+(2*tau)):
(newpnt+(2*tau))), 'g', 'LineWidth', 1);
for aa = 0:evolve;
    plot3([data(tmpold+aa), data(tmpnew+aa)],
[data(tmpold+aa+tau), data(tmpnew+aa+tau)],
[data(tmpold+aa+(2*tau)), data(tmpnew+aa+(2*tau))]),
'LineWidth', 1)
end

ann = legend(['Iteration:', num2str(out(ii,1)), '/',
num2str(out(end,1)), char(10)...
'd_i:', num2str(out(ii,2)), char(10)...
'd_f:', num2str(out(ii,3)), char(10)...
'Current Estimate:', num2str(out(ii,4))], ...
'location', loc);
if freerun == 1
    drawnow
end

```

```
end
end
```

```
function [bstpnt , bstdis , thbest]=search( iflag , ndim , ...
    delay , datmin , boxlen , nxtbox , where , datptr , ...
    nxdat , data , oldpnt , newpnt , datuse , dismin , ...
    dismax , thmax , evolve , ires )
% Recherche le point le plus viable pour fet.m
% Parc Taehyeun, The Cooper Union, EE'15

target = zeros(1, ndim);
oldcrd = zeros(1, ndim);
zewcrd = zeros(1, ndim);

oldcrd(1:ndim) = data(oldpnt+delay);
zewcrd(1:ndim) = data(newpnt+delay);
igerds = floor((oldcrd - datmin)./ boxlen);
olddist = sqrt(sum((oldcrd - zewcrd).^2));

irange = round(dismin/boxlen);
if irange == 0;
    irange = 1;
end

thbest = thmax;
bstdis = dismax;
bstpnt = 0;

goto30 = 1;
while goto30 == 1
    goto30 = 0;
    for icnt = 0:((2*irange+1)^ndim) - 1
        goto140 = 0;
        icounter = icnt;
        for ii = 1:ndim;
            ipower = (2*irange+1)^(ndim - ii);
            ioff = floor(icounter./ipower);
            icounter = icounter - ioff*ipower;
            target(ii) = igerds(ii) - irange + ioff;
        end
    end
    goto30 = 1;
end
```

```

    if target(ii) < 0
        goto140 = 1;
        break;
    end
    if target(ii) > ires 1
        goto140 = 1;
        break
    end
end
end

if goto140 == 1;
    continue
end

if irange ~= 1
    iskip = 1;
    for ii = 1:ndim
        if abs(round(target(ii) igcrds(ii)))==irange
            iskip = 0;
        end
    end
    if iskip == 1
        continue
    end
end

runner = 1;
for ii = 1:ndim
    goto80 = 0;
    goto70 = 1;
    while goto70 == 1;
        goto70 = 0;
        if where(runner, ii) == target(ii)
            goto80 = 1;
            break
        end
        runner = nxtbox(runner, ii);
        if runner ~= 0
            goto70 = 1;
        end
    end
end

if goto80 == 1

```

```

        continue
    end
    goto140 = 1;
    break
end

if goto140 == 1
    continue
end

if runner == 0
    continue
end
runner = datptr(runner);
if runner == 0
    continue
end
goto90 = 1;
while goto90 == 1
    goto90 = 0;
    while 1;
        if abs(round(runner    oldpnt)) < evolve
            break
        end
        if abs(round(runner    datuse)) < (2*evolve)
            break
        end
    end

    bstcrd = data(runner + delay);

    abc1 = oldcrd(1:ndim)    bstcrd(1:ndim);
    abc2 = oldcrd(1:ndim)    zewcrd(1:ndim);
    tdist = sum(abc1.*abc1);
    tdist = sqrt(tdist);
    dot = sum(abc1.*abc2);

    if tdist < dismin
        break
    end
    if tdist >= bstdis
        break
    end
    if tdist == 0

```

```

        break
    end
    goto120 = 0;
    if iflag == 0
        goto120 = 1;
    end
    if goto120 == 0
        ctheta = min(abs(dot/(tdist*oldist)),1);
        theta = 57.3*acos(ctheta);
        if theta >= thbest
            break
        end
        thbest = theta;
    end
    bstdis = tdist;
    bstpnt = runner;
    break;
end
runner = nnextdat(runner);
if runner ~= 0
    goto90 = 1;
end
end
end
irange = irange + 1;
if irange <= (0.5 + round((dismax/boxlen)))
    goto30 = 1;
    continue;
end
return
end
end

```

```

clc; clear all; close all; format compact;
%
%
% Version Matlab de l'algorithme de Wolf et al. pour
% estimation de l'exposant de Lyapunov dominant partir
% d'un 1 D des series chronologiques.
%
% Physica 16D (1985) 285-317 "D termination de Lyapunov
% Exposants d'une serie chronologique "

```

```

% Alan Wolf, Jack B. Swift, Harry L. Swinney et
% John A. Vastano
%
% L'annexe B de l'article Physica D contient Fortran
% code pour une version concise, mais tr s inefficace de
% l'algorithme.
% Je distribue une version Fortran et C de
% la version efficace de l'algorithme depuis la
% Ann es 80
% La version efficace du code a t convertie en
% Matlab par Taehyeun Park, The Cooper Union, EE'15
% en septembre 2014.
%
% Des instructions d taill es pour l'utilisation de ce code
% tre affich sur l' change de fichiers de Matlab Central
%
% Ce fichier, testbench.m, prend 16 384 points de Lorenz
% donn es d'attracteur (dt = 0,01 seconde) et estime la
% exposant dominant Lyapunov en appelant d'abord "basgen"
% (pr paration d'une base de donn es pour trouver rapidemen
% voisins dans l'espace des phases reconstruit) puis
% appelant "fet" pour estimer l'exposant de Lyapunov.
% Le programme montre la fois la sortie graphique comme
% orbitale la divergence est surveill e et un fichier texte
% estimation de l'exposant en cours d'ex cution.
%
%
fname = 'Data.lor';
datcnt = 10000;
tau = 10;
ndim = 3;
ires = 10;
maxbox = 6000;

db = basgen(fname, tau, ndim, ires, datcnt, maxbox);

dt = .01;
evolve = 20;
dismin = 0.001;
dismax = 0.3;
thmax = 30;

[out, SUM] = fet(db, dt, evolve, dismin, dismax, thmax);

```



```
makeplot(db, out, evolve, 'NorthWest')
```

La figure (3.5) : (4)Finance.

```
function [Texp, Lexp]=lyapunov(n, rhs_ext_fcn,
    fcn_integrator, tstart, stept, tend, ystart, ioutp);
%Calcul de l'exposant de Lyapunov pour le syst me ODE.
%
%L'algorithme utilis dans ce fichier m pour d terminer
%Lyapunov exposants a ete propose dans
%
%    A. Wolf, J. B. Swift, H. L. Swinney et J. A. Vastano,
%    "D terminer les exposants de Lyapunov partir d'une
%    serie chronologique", Physica D,
%    Vol. 16, pages 285-317, 1985.
%
%Pour int grer le syst me ODE, vous pouvez utiliser n'importe
%quelle m thode MATLAB ODE suite.
%Cette fonction fait partie du programme MATDS  boite
%outils pour la recherche de syst me dynamique.
%    Voir:    http://www.math.rsu.ru/mermat/kvm/matds/
%
%Param tres d'entree:
% n    nombre d'equation
% rhs_ext_fcn    poignee de fonction avec le cote droit du
% syst me ODE    tendu .
%Cette fonction doit inclure la RHS du syst me ODE coupl e
% quation    variationnelle (n    lments    de syst mes lin ari s ,
%voir exemple).
%fcn_integrator    descripteur de la fonction d'int grateur
%ODE, par exemple: @ ode45.
%tstart    valeurs de d part d'une valeur ind pendante
%(temps t)
% stept    tape sur la variable t pour la proc dure de
%renormalisation de Gram Schmidt.
% tendance    valeur finale du temps
%ystart    point de d part de la trajectoire du syst me ODE.
%ioutp    tape d'impression dans la fen tre principale de
%MATLAB. ioutp == 0    pas d'impression,
%si ioutp > 0 alors chaque ioutp me point sera imprim .
%
```

```

%Param tres de sortie:
%Texp  valeurs de temps
%Lexp  Exposants de Lyapunov    chaque valeur de temps.
%
%Les utilisateurs doivent ecrire leurs propres fonctions
ODE pour leur sp cifi
%syst mes et utiliser le handle de cette fonction comme
%rhs_ext_fcn  parametre.
%
%    Exemple. Systeme Lorenz:
%          dx/dt = sigma*(y - x) = f1
%          dy/dt = r*x - y - x*z = f2
%          dz/dt = x*y - b*z = f3
%
%    Le jacobien du syst me:
%          |  sigma  sigma  0  |
%    J =  |   r z     1    x  |
%          |    y     x    b  |
%
%    Ensuite, l' equation variationnelle a une forme:
%
%    F = J*Y
%o Y est une matrice carr e de m me dimension que J.
%    Fichier m correspondant:
%          fonction f=lorenz_ext(t,X)
%          SIGMA = 10; R = 28; BETA = 8/3;
%          x=X(1); y=X(2); z=X(3);
%
%          Y= [X(4), X(7), X(10);
%              X(5), X(8), X(11);
%              X(6), X(9), X(12)];
%          f=zeros(9,1);
%f(1)=SIGMA*(y - x); f(2)= x*z+R*x - y; f(3)=x*y - BETA*z;
%
%          Jac=[ SIGMA,SIGMA,0; R z , 1 , x; y , x , BETA];
%
%          f(4:12)= Jac*Y;
%
%    Run Lyapunov exponent calcul:
%
%[T, Res]=lyapunov(3, @lorenz_ext, @ode45, 0, 0.5, 200, [0 1 0], 10);
%
%    See files: lorenz_ext , run_lyap.

```

```

%
%
% Copyright (C) 2004, Govorukhin V.N.
% Ce fichier est destin     tre utilis     avec MATLAB et
% produit pour le programme MATDS
% http://www.math.rsu.ru/mexmat/kvm/matds/
% lyapunov.m est un logiciel libre. lyapunov.m est
% distribu     dans l'espoir qu'il
% sera utile , mais SANS AUCUNE GARANTIE.
%

%
%      n = nombre d'odes non lin aires
%      n2=n*(n+1)=nombre total d'odes
%
n1=n; n2=n1*(n1+1);

% Nombre de pas

nit = round((tend tstart)/stept);

% Allocation de m moire

y=zeros(n2,1); cum=zeros(n1,1); y0=y;
gsc=cum; znorm=cum;

% Valeurs initiales

y(1:n)=ystart (:);

for i=1:n1 y((n1+1)*i)=1.0; end;

t=tstart;

% Boucle principale

for ITERLYAP=1:nit

% Solution du syst me ODE tendu

[T,Y] = feval(fcn_integrator ,rhs_ext_fcn ,[t t+stept] ,y);

```

```

t=t+stept;
y=Y(size(Y,1),:);

for i=1:n1
    for j=1:n1 y0(n1*i+j)=y(n1*j+i); end;
end;

%
%construire une nouvelle base orthonormale par gram schmidt
%

znorm(1)=0.0;
for j=1:n1 znorm(1)=znorm(1)+y0(n1*j+1)^2; end;

znorm(1)=sqrt(znorm(1));

for j=1:n1 y0(n1*j+1)=y0(n1*j+1)/znorm(1); end;

for j=2:n1
    for k=1:(j-1)
        gsc(k)=0.0;
        for l=1:n1 gsc(k)=gsc(k)+y0(n1*l+j)*y0(n1*l+k);
        end;
    end;

    for k=1:n1
        for l=1:(j-1)
            y0(n1*k+j)=y0(n1*k+j)-gsc(l)*y0(n1*k+l);
        end;
    end;

    znorm(j)=0.0;
    for k=1:n1 znorm(j)=znorm(j)+y0(n1*k+j)^2; end;
    znorm(j)=sqrt(znorm(j));

    for k=1:n1 y0(n1*k+j)=y0(n1*k+j)/znorm(j); end;
end;

%
% mettre a jour les amplitudes des vecteurs en cours
%d'execution.

```

```

for k=1:n1 cum(k)=cum(k)+log(znorm(k)); end;

%
%      normaliser l'exposant
%

for k=1:n1
    lp(k)=cum(k)/(t-tstart);
end;

% Modification de sortie

if ITERLYAP==1
    Lexp=lp;
    Texp=t;
else
    Lexp=[Lexp; lp];
    Texp=[Texp; t];
end;

if (mod(ITERLYAP,ioutp)==0)
    fprintf('t=%6.4f',t);
    for k=1:n1 fprintf('□%10.6f',lp(k)); end;
    fprintf('\n');
end;

for i=1:n1
    for j=1:n1
        y(n1*j+i)=y0(n1*i+j);
    end;
end;

end;

```

```

function f=finance_ext(t,X)
%
% Valeurs des param tres
a = 2;
b = 0.1;
c = 1;

```

```

x=X(1); y=X(2); z=X(3);

Y= [X(4), X(7), X(10);
     X(5), X(8), X(11);
     X(6), X(9), X(12)];

f=zeros(9,1);

%systeme Finance
f(1)=z+(y a)*x;
f(2)=1 b*y x^2;
f(3)= x c*z;

%Systeme lin aris

Jac=[y a, x, 1;
     2*x, b, 0;
     1, 0, c];

% quation variationnelle
f(4:12)=Jac*Y;

%Les donn es de sortie doivent tre un vecteur de colonne

```

```

[T, Res]=lyapunov(3, @finance_ext, @ode45, 0, 0.01, 75, [2 3 2], 10);
plot(T, Res);
title('Dynamics of Lyapunov exponents');
xlabel('Time'); ylabel('Lyapunov exponents');

```

figure (3.7) : (2)Lozi.

```

%systeme de Lozi
clear all; clc
x=0.3;y=0;w=0.3;z=0;
a=1.7;b=0.5;
for k=1:200
    x(k+1)=1 a*(abs(x(k)))+y(k);
    y(k+1)=b*x(k);

```

```

w(k+1)=0.6 a*(abs(w(k)))+z(k)+0.4;
z(k+1)=b*w(k);
end

T=1:k+1;
lbe=abs(x w)/2;
figure(1)
plot(T,log(lbe),'co')
axis([0 210 40 1])
hold on
t1=T';
%it rations totales
aux=[find(log10(lbe) > 3)];
iterate=aux(1);
%intervale d'approximation
aux=[find(log10(lbe) > 35)];
lbe(1:aux(1))=[];
t1(1:aux(1))=[];
aux=[find(log10(lbe) > 3)];
lbe(aux(1):end)=[];
t1(aux(1):end)=[];
aux1=polyfit(t1',log(lbe),1);

lyap=aux1(1)
plot(t1,aux1(1)*t1+aux1(2),'r','LineWidth',2)
hold off
box off
xlabel('t','FontSize',20)
ylabel('log(|X1 X2|)','FontSize',20)
set(gcf,'PaperPosition',[0 0 30 20]);
set(gcf,'PaperSize',[30 20]);
set(gca,'FontSize',20,'Fontname','Times_New_Roman')
text(95,25,sprintf('%1.3f_t_%1.3f',aux1(1),aux1(2)),
'FontSize',20)
saveas(gcf,'fig2','pdf')
%Exposant de Lyapunov
disp(lyap)
%optimal it ration
disp(iterate)

```

Figure (3.9). [26]

```

%Exposant de Lyapunov par rapport le nombre d'it ration.
clear;
fontsize=15;

i=1;
for itermax=100:50:10000
    x=0.1;xo=x;mu=4;

    for n=1:itermax
        xn=mu*xo*(1-xo);
        x=[x xn];
        xo=xn;
    end
    Lyap_exp=sum(log(abs(mu*(1-2*x))))/itermax
    M(i)= Lyap_exp
    i=i+1
end
plot((100:50:10000),M,'oc')
xlabel('N','FontSize',fontsize);
ylabel('lamda','FontSize',fontsize)

```


Bibliographie

- [1] M. S. ABDELOUAHAB. *Les systèmes chaotiques à dérivées fractionnaires*. Mémoire de magistère. Univ. Mentouri. Constantine 1, 2009.
- [2] R. AMIRA, S. BOUZRAA. *Sur le test 01 pour confirmer le comportement chaotique d'un système dynamique*. Mémoire de master. Centre. Universitaire Abd Elhafid Boussouf. Mila, 2018.
- [3] A. OUANNAS. *Sur La Synchronisation Des Systèmes Chaotiques Discrets*. Thèse de doctorat . Univ. Mentouri. Constantine 1, 2015.
- [4] I. TALBI. *Systèmes dynamiques non linéaire et phénomènes de chaos*. Mémoire de magistère. Univ. Mentouri. Constantine 1, 2010.
- [5] W. GHOUL. *La théorie de bifurcation*. Mémoire de master. Univ. Guelma, 2013.
- [6] A. BAKRIA. *Théorème de Poincaré-Bendixson et applications*. Mémoire de magistère. Univ. Kasdi Merbah. Ouargla, 2000.
- [7] O. MEGHERBI. *Etude et réalisation d'un système sécurisé à base de système chaotique*. Mémoire de magistère. Univ. Mouloud Hammeri. Tizi-ouzou, 2013.
- [8] N.W. ABDERRAHIM. *Étude et conception d'un modèle chaotique dédié aux transmissions chiffrées*. thèse de doctorat. Univ. Abou-bekr Belkaid, 2015.
- [9] A.R. KIHAL. *Systèmes chaotiques pour la transmission sécurisée de données*. Univ. Mohamed Khider. Biskra, 2013.
- [10] J. P. FRANÇOISE . *Oscillations en biologie : Analyse qualitative et modèles*. Springer-Verlag. Berlin, Heidelberg, 2005.
- [11] H. DANG-VU. C. DELCARTE . *Bifurcations et chaos : Introduction à la dynamique contemporaine avec des programmes en Pascal*. Springer-Verlag. Fortran et Mathématica, Ellipses, 2000.
- [12] J. L. PAC . *Systèmes dynamiques : Cours et exercices corrigés*. Dunod, Paris, 2016.
- [13] T. Y. LI. J. A. YORKE . *Period three implies chaos*. American Mathematical Monthly, 82 : 985-992, 1975.
- [14] T. MENACER . *Synchronisation des systèmes chaotiques à dérivées fractionnaires*. Thèse de Doctorat , Univ. Mentouri. Constantine 1, 2014.

- [15] T. AMADO. *Modélisation d'un système dynamique a jeu du groupe moto-propulse*.Thèse de Doctorat Ecole centrale de Lyon, 2006.
- [16] A. MENASRI. *CHAOS ET BIFURCATIONS DANS LES SYSTEMES DYNAMIQUES EN DIMENSIONS n ($n > 1$)*. Thèse de Doctorat, Univ. Oum el-boighi, 2016.
- [17] I. MEDJAHDI . *Bifurcation de Hopf pour un modèle proie-prédateur avec structure d'âge*. Mémoire de Master, Univ. Abou Bekr Belkaid Tlemcen , 2019.
- [18] A. WOLF, J. B. SWIFT, H. L. SWINNEY, ET J. A. VASTANO . *Determining Lyapunov Exponents from a time Series*. Physica D : Nonlinear Phenomena, 16 : 285-317, 1985.
- [19] KANTZ, HOLGER, & SCHREIBER, THOMAS. *Nonlinear time series analysis*. Second edn. Cambridge University Press. 2004.
- [20] E. M. A. M. MENDES AND E. G. NEPOMUCENO. *A very simple method to calculate the (Positive) largest lyapunov exponent using interval extensions*. International Journal of Bifurcation and Chaos, vol. 26, no. 13, Article ID 1650226, 2016.
- [21] ECKMANN, J. P.& RUELLE, D. *Ergodic theory of chaos and strange attractors*. Rev. Mod. Phys. 57,1985,617–656.
- [22] M. S. ABD-ELOUAHAB, N. HAMRI, AND JUNWEI WANG. *Chaos Control of a Fractional-Order Financial System*. Hindawi Publishing Corporation Mathematical Problems in Engineering, Article ID 270646, 2010.
- [23] K. RAMASUBRAMANIAN, M. S. SRIRAM. *A comparative study of computation of Lyapunov spectra with different algorithms*. Physica D 139, 2000, 72–86.
- [24] ZERAOLIA ELHADJ. *Lozi Mappings Theory and Applications*. CRC press A SCIENCE PUBLISHERS BOOK :98.
- [25] BERNARD J. GEURTS. DARRY D. HOLM. ERWIN LUESINK. *Lyapunov Exponents of Two Stochastic Lorenz 63 Systems*. Journal of Statistic Physics ,2019.
- [26] S. LYNCH. *Dynamical Systems with Applications using MATLAB Second Edition*. Springer International Publising. 2014. PP :60.
- [27] PIERRE. AUGER.*Modélisation Mathématique En Écologie*. Cours et exercices corrigés. Paris.2010.PP :50.