

DEPARTEMENT DES MATHÉMATIQUES

Master Mathématique et Application au Calcul Scientifique
(MACS)

MEMOIRE DE FIN D'ETUDES

Pour l'obtention du Diplôme de Master Sciences et Techniques
(MST)

**Systèmes dynamiques de type
réseaux de Hopfield et optimisation**

Réalisé par : EL-AOUNI Abderazzak

Encadré par : Pr. ETTAOUIL Mohamed

Pr. JOUDAR Nour-eddine

Soutenu le 15 juillet 2021

Devant le jury composé de :

- Pr. EL KHAOULANI EL IDRISSE Rachid FST de Fès
- Pr. ETTAOUIL Mohamed FST de Fès
- Pr. EL-FASSI Iz-iddine FST de Fès
- Pr. JOUDAR Nour-eddine ENSAM de Rabat

Année Universitaire 2020 / 2021

Table des matières

Remerciements	4
Introduction générale	5
1 Les systèmes dynamiques	7
1.1 Introduction	7
1.2 Systèmes dynamiques discrets	8
1.2.1 Systèmes dynamiques discrets linéaires de dimension 1	10
Point fixe (d'équilibre)	11
Existence et unicité du point fixe	11
Stabilité du point fixe	12
1.2.2 Systèmes dynamiques discrets non linéaire de dimension 1	16
Stabilité du point fixe	17
Stabilité du point fixe non hyperbolique	17
1.2.3 Systèmes dynamiques discrets linéaires de dimension n .	20
Point fixe	21
Stabilité du point fixe	22
1.2.4 Systèmes dynamiques discrets non linéaires de dimension n .	24
Stabilité du point fixe	24
Méthode de Lyapunov	25
1.3 Systèmes dynamiques continus	26
1.3.1 Préliminaires et définitions	27
1.3.2 Notions de stabilité	28
1.3.3 Stabilité des systèmes linéaires	30
1.3.4 Fonction de Lyapunov et méthode directe de Lyapunov	31
1.3.5 Méthode indirecte de Lyapunov	33
1.3.6 Fonctions de Lyapunov pour les systèmes linéaires continus	35
1.3.7 Méthode de construction des fonctions de Lyapunov : Méthode de Kra-	
sovskii	37
1.4 Conclusion	37
2 Réseaux de Hopfield	38
2.1 Introduction	38
2.2 Historique	38
2.3 Domaines d'application des réseaux de neurones artificiels	39
2.4 Notions de base sur les réseaux de neurones artificiels	40
2.4.1 Du Neurone biologique au neurone artificiel	40
2.4.2 Les réseaux de neurones	42
2.4.3 Réseau de neurones non bouclé	43

2.4.4 Réseau de neurones bouclé (ou récurrent)	43
2.5 Architectures de réseaux de neurones pour l'optimisation	44
2.6 Fonctions d'énergie pour l'optimisation combinatoire	44
2.7 Réseau de Hopfield	46
2.7.1 Structure et mode de fonctionnement du réseau de Hopfield discret	46
2.7.2 Stabilité du réseau et attracteur	48
2.7.3 Réseau de Hopfield continu	51
2.8 Principes de fonctionnement du réseau de Hopfield pour l'optimisation	54
2.9 Application des réseaux de Hopfield à l'optimisation	54
2.10 Limitations des réseaux de Hopfield	55
2.11 Conclusion	55
3 Application : Le problème du voyageur de commerce	56
3.1 Introduction	56
3.2 Codage du problème	56
3.3 La fonction d'énergie	59
3.4 Réglage des paramètres	60
3.5 Résultats de simulation	67
Conclusion générale	72
Bibliographie	73

Table des figures

1.1	Représentation graphique d'une orbite.	10
1.2	Point fixe stable	12
1.3	Point fixe localement asymptotiquement stable	12
1.4	Point fixe globalement asymptotiquement stable	13
1.5	Point fixe instable	13
1.6	Point fixe globalement asymptotiquement stable ($-1 < a < 0$)	14
1.7	Point fixe globalement asymptotiquement stable ($0 < a < 1$)	14
1.8	Non-existence d'un point fixe ($a = 1$ et $b \neq 0$)	15
1.9	Point fixe n'est pas asymptotiquement stable ($a = -1$)	15
1.10	Point fixe instable ($a > 1$)	16
1.11	Point fixe instable ($a < -1$)	16
1.12	Point fixe instable	18
1.13	Point fixe asymptotiquement stable	18
1.14	Point fixe semi-stable à gauche	18
1.15	Point fixe semi-stable à droite	18
1.16	Illustration de la définition intuitive de la stabilité	28
1.17	Illustration de la stabilité au sens de Lyapunov	28
1.18	Illustration de la stabilité asymptotique	29
2.1	Le neurone biologique	41
2.2	Mise en correspondance neurone biologique / neurone artificiel	41
2.3	Fonction de Heaviside	42
2.4	Fonction signe	42
2.5	Fonction linéaire à seuil	42
2.6	Fonction sigmoïde	42
2.7	Structure d'un réseau de neurones non bouclé.	43
2.8	Un réseau de neurones bouclé à deux entrées	44
2.9	Structure du réseau de Hopfield discret	47
2.10	Trois types de stabilité d'un réseau : (a) stabilité (b) oscillation (c) chaos	48
2.11	Réseau de trois neurones	51
2.12	Diagramme d'évolution de réseau de trois neurones	51
3.1	Solution acceptable du problème du voyageur de commerce pour cinq villes	57
3.2	Les sous-ensembles de H .	62
3.3	Circuit de voyage pour 5, 10, 15, 20 et 25 villes.	68
3.4	Convergence de l'énergie : 5, 10, 15, 20 et 25 villes.	69
3.5	Nombre d'itérations nécessaires à la convergence en fonction du nombre de villes.	70
3.6	Temps nécessaire pour la convergence du réseau en fonction du nombre de villes.	70

Remerciements

Ce travail est le fruit de la combinaison d'efforts de plusieurs personnes. Je remercie tout d'abord le tout puissant qui, par sa grâce m'a permis d'arriver au bout de mes efforts en me donnant la santé, la force, le courage et en me faisant entourer des merveilleuses personnes dont je tiens à remercier. Je remercie :

*Mes deux directeur de mémoire Monsieur **ETTAOUIL Mohamed**, professeur à la faculté des sciences et techniques Fès et Monsieur **JOUDAR Nour-eddine** professeur à ENSAM de Rabat. Grâce à leur savoir, leurs idées, leur disponibilité et leur soutien, j'ai pu réaliser ce travail.*

*Les membres du jury : Monsieur **EL KHAOULANI EL IDRISSE Rachid** , professeur à la Faculté des sciences et techniques de Fès et Monsieur **EL-FASSI Iz-iddine**, professeur à la Faculté des sciences et techniques de Fès, qui m'ont honoré en acceptant d'évaluer et de juger mon travail.*

Tous les enseignants de la FST Fès, pour leurs enseignements de qualité et leurs conseils qui nous ont permis de poursuivre notre itinéraire académique jusqu'à présent .

Mes parents qui m'ont toujours encouragés dans la poursuite de mes études, ainsi que pour leur aide, leur compréhension et leur soutien.

Mes Frères et sœurs pour leurs encouragements durant tout mon parcours.

Mes camarades et tous ceux qui de près ou de loin ont contribué à l'accomplissement de ce travail.

Introduction générale

Les réseaux de neurones sont des modèles mathématiques et informatiques, des assemblages d'unités de calcul appelés neurones formels, et dont l'inspiration originelle était le fonctionnement cérébral de l'être humain. Dans les années 80 les réseaux de neurones artificiels ou "RNA" formels ont été développés pour résoudre des problèmes d'ingénierie : estimation, approximation des fonctions, prédiction etc.

Aujourd'hui, les réseaux de neurones artificiels ont de nombreuses applications dans des secteurs très variés comme le traitement d'images (reconnaissance de caractères et de signatures, compression d'images, reconnaissance de formes, cryptage, classification, etc.) et le traitement du signal (filtrage, classification, identification de source, traitement de la parole etc.)

L'optimisation est aussi un domaine d'application des réseaux de neurones, telle qu'en 1982, J.J.Hopfield [25], physicien de l'institut de technologie de Californie, a publié un article sur le développement des réseaux neuronaux, proposant un réseau de neurones récurrent mono-couche, appelé par la suite le réseau de Hopfield.

En 1985, Hopfield avec D.W. Tank [26] ont également mis en place le réseau de Hopfield avec des circuits électroniques analogiques. Ils ont réussi à résoudre le problème du voyageur de commerce, en ouvrant ainsi un réseau neuronal capable de faire un traitement intelligent de l'information, et une nouvelle voie prometteuse pour le traitement des problèmes d'optimisation via les réseaux de neurones.

Contrairement aux réseaux feedforward qui généralement ne tiennent pas compte du lien entre la sortie et l'entrée des neurones, le réseau de Hopfield tient compte du lien entre la sortie et l'entrée des neurones grâce à sa dynamique qui est caractérisée par une équation différentielle. Dans ce contexte, nous proposons plusieurs modélisations de cette équation pour construire un nouveau modèle mathématique de ce type de réseau.

Les poids de ce réseau de Hopfield ne sont pas obtenus par l'apprentissage, mais calculés d'avance selon certaines règles. Les poids ne changeront pas une fois déterminés, et l'état de chaque neurone sera constamment mis à jour jusqu'à l'évolution de réseau vers un état stable. Dans le cas de la stabilité, l'état de chaque neurone est la solution du problème. Le réseau de Hopfield est composé en deux types de modèles : réseau de Hopfield discret (discrete Hopfield neural network) et réseau de Hopfield continu (continuous Hopfield neural network).

Les réseaux de Hopfield ont prouvé leur efficacité dans la résolution de problèmes d'optimisation. Toutes ces applications nécessitent un codage du problème sous la forme d'un réseau de neurones artificiels décrit au moyen d'une forme quadratique, appelée énergie du problème, et dont le minimum correspond à une solution optimale du problème. Une fois cette énergie définie, il est nécessaire de choisir la méthodologie neuronale à mettre en œuvre pour sa minimisation.

Notre mémoire est organisé comme suit : Le premier chapitre est consacré à l'étude des systèmes dynamiques autonomes. Alors que dans le deuxième chapitre, on va découvrir qu'est-ce qu'un réseau de Hopfield, ses caractéristiques et comment l'appliquer aux problèmes d'optimisation. Ce modèle neural sera appliqué au problème du voyageur de commerce dans le troisième chapitre, et au cours de cette application, on va donner le codage de ce problème, la fonction d'énergie, les équations d'évolution des neurones et les conditions de convergence du réseau . Et on va terminer ce mémoire avec une conclusion générale et des perspectives.

Chapitre 1

Les systèmes dynamiques

1.1 Introduction

Les systèmes dynamiques sont les notions mathématiques qui permettent de modéliser des phénomènes évoluant dans le temps, Le terme "système" se réfère à un ensemble de variables d'état (dont la valeur évolue au cours du temps) et aux interactions entre ces variables. Elles permettent de définir une fonction de transition qui donne l'évolution des variables d'état sur un intervalle de temps τ en fonction de l'état au temps t . Ces variables servent le plus souvent à décrire des objets physiques, comme la position d'un point, la température d'une particule ou d'un gaz, un composé d'une réaction chimique. Par extension, elles peuvent aussi correspondre aux variables d'un algorithme dont les valeurs évoluent à chaque itération de la récursion.

Depuis 1920 jusqu'à présent les systèmes dynamiques jouent un rôle très important puisque il y a des applications dans beaucoup de disciplines scientifiques par exemple : La physique (mécanique céleste, météo), la biologie (dynamique de population), la chimie (cinétique chimique), l'électronique (les circuits électroniques), l'informatique (traitement de l'images), cryptographie (chiffrement des messages, images), l'économie,..., etc.

On distingue deux types de systèmes dynamiques : les systèmes à temps continu et les systèmes à temps discret. Le système dynamique discret a une grande importance pratique, il peut également être utilisé comme modèle approximatif pour l'étude de système continu.

Définition 1.1. *Un système dynamique est défini par un triplet (X, T, ϕ) , constitué de l'espace d'états X , du domaine temporel T et d'une fonction de transition d'état $\phi : X \times T \rightarrow X$, qui possède la propriété : Pour tout $x \in X$ et $t, \tau \in T$*

$$\begin{cases} \phi(x, 0) = x \\ \phi(\phi(x, t), \tau) = \phi(x, t + \tau) \end{cases}$$

La fonction ϕ décrit la façon dont le système évolue au cours du temps.

- Si cette fonction ne dépend pas du temps, le système est autonome.
- La taille de X (nombre de variables d'état) est N . On l'appelle aussi N l'ordre du système.
- Le choix de l'espace temporel T est décisif, et dépend en général du phénomène que l'on souhaite modéliser.

- Si $T = \mathbb{R}^+$, le système est dit à temps continu, et si $T = \mathbb{N}$, le système est dit à temps discret.

Quelques définitions :

- Pour tout $x \in X$, la **trajectoire** qui a x pour origine est définie par l'application $\phi_x : T \rightarrow X$ telle que

$$\phi_x(t) = \phi(x, t).$$

Le terme trajectoire est issu de la physique classique dans le cas où les variables d'état sont les coordonnées d'un point matériel dans un espace à 3 dimensions.

- Lorsque les variables d'état représentent des grandeurs autres que des coordonnées spatiales, il est néanmoins possible de représenter les trajectoires au sein de l'espace d'état, également appelé espace des phases. Par ailleurs, on appelle **orbite** de x l'ensemble d'états

$$\Gamma(x) = \{\phi_x(t) \mid t \in T\}$$

- À l'instant t , on définit le **flot du système** par l'application $\phi_t : X \rightarrow X$ telle que

$$\phi_t(x) = \phi(x, t)$$

1.2 Systèmes dynamiques discrets

Exemple 1.1.

Supposons que nous avons une population de lapins qui au début de notre expérience compte $x(0)$ lapins. Nous savons qu'en une année la population augmente de 10%. Notons par $x(t)$ le nombre de lapins de la t -ème année. Nous voulons décrire l'évolution de $x(t)$. Après une année on obtient $x(1)$ lapins

$$\begin{aligned} x(1) &= x(0) + 0.1x(0) = 1.1x(0) \\ x(2) &= x(1) + 0.1x(1) = 1.1x(1) \end{aligned}$$

Au cours de la deuxième année la quantité de lapins augmente de la même façon. En continuant on trouve pour une année quelconque

$$x(t+1) = x(t) + 0.1x(t) = 1.1x(t)$$

Ainsi nous pouvons remarquer que pour chaque période de temps

$$x(t+1) = f(x(t))$$

avec

$$f(x) = 1.1x$$

Autrement dit, la dynamique de la population peut être décrite, comme dans l'exemple précédent, par l'itération d'une fonction $f(x)$. En connaissant cette fonction nous pouvons reconstituer l'état du système à chaque moment du temps.

Dans le cas général un système dynamique discret est décrit par un système d'équations aux différences finies, autrement dit, par une récurrence.

Définition 1.2. On appelle système dynamique discret autonome d'ordre 1 l'équation aux différences suivante :

$$\begin{cases} x_{t+1} = f(x_t) \\ x_0 \text{ donné} \end{cases}, t \in \mathbb{N} \quad (1.1)$$

Où :

- $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ une fonction différentiable.
- $x_0 \in D \subset \mathbb{R}^n$: Valeur initial.
- $x_t \in D$: Vecteur des états du système.

De (1.1) on obtient :

$$\begin{aligned} x_1 &= f(x_0) \\ x_2 &= f(x_1) = f(f(x_0)) = f^2(x_0) \end{aligned}$$

⋮

En général :

$$x_t = f^t(x_0) \quad \text{où} \quad f^t = \underbrace{f \circ \dots \circ f}_{t \text{ fois}}$$

$f(x_0)$ est appelée première itération de x_0 par la fonction f .

$f^2(x_0) = f(f(x_0))$ est appelée seconde itération de x_0 par la fonction f .

$f^t(x_0)$ est appelée t^{ime} itération de x_0 par la fonction f .

Donc, le triplet (D, \mathbb{N}, ϕ) définit un système dynamique discret autonome, où ϕ est donné par :

$$\phi(x_0, t) = f^t(x_0)$$

Remarque 1.1.

1. Si la fonction f est une fonction de l'état x_t et de la variable du temps t i.e : $x_{t+1} = f(t, x_t)$ alors le système s'appelle système dynamique discret non autonome.
2. Si l'équation aux différences est d'ordre $p \geq 2$ autonomes ou non i.e :

$$x_{t+p} = f(x_t, x_{t+1}, x_{t+2}, \dots, x_{t+p-1})$$

alors le système s'appelle systèmes dynamiques discrets d'ordre supérieur.

Définition 1.3. On appelle la suite $\{f^t(x_0)\}_{t=0}^{\infty}$ une orbite (ou un trajectoire) du point x_0 et on le note par $O(x_0)$ i.e :

$$O(x_0) = \{x_0, f(x_0) = x_1, \dots, f^t(x_0) = x_t, \dots\}$$

Représentation graphique de l'orbite :

Dans le cas où f est une fonction de \mathbb{R} vers \mathbb{R} on peut représenter sur le plan (x, y) l'évolution d'une orbite $O(x_0)$ qui commence dans le point $(x_0; 0)$ en suivant ces étapes :

Etape 1 : Tracer la courbe représentant la fonction f et la droite $y = x$.

Etape 2 : Placer le point de coordonnées $(x_0; 0)$.

Etape 3 : Chercher le point d'ordonnée $f(x_0)$, on l'obtient en traçant une droite verticale passant par $(x_0; 0)$ et en cherchant son intersection avec la courbe de la fonction f . Ce point a comme ordonnée $f(x_0)$, ce qui correspond à x_1 (puisque $x_1 = f(x_0)$).

Etape 4 : Projeter horizontalement le point de coordonnées $(x_0; x_1)$ sur la droite $y = x$ pour obtenir le point de coordonnées $(x_1; x_1)$, une projection verticale permet ensuite de reporter le point $(x_1; 0)$ sur l'axe des ordonnées.

Réaliser ensuite pour x_1 les mêmes opérations que pour x_0 afin d'obtenir x_2 et ainsi de suite pour les termes de rang suivant.

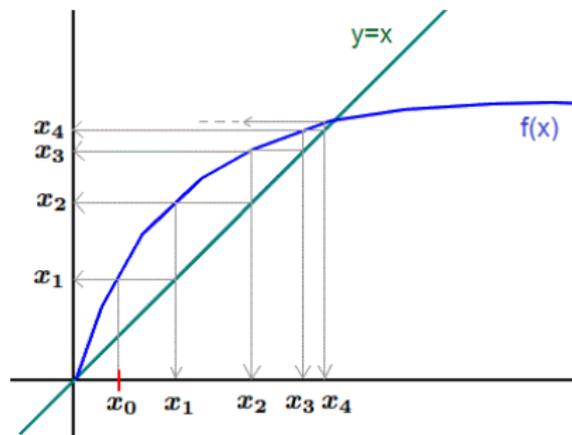


FIGURE 1.1 – Représentation graphique d'une orbite.

1.2.1 Systèmes dynamiques discrets linéaires de dimension 1

Définition 1.4. Un système dynamique linéaire discret est définie par l'équation aux différences suivant :

$$\begin{cases} x_{t+1} = ax_t + b \\ x_0 \text{ donné} \end{cases}, t \in \mathbb{N} \quad (1.2)$$

Où :

- $a, b \in \mathbb{R}$: sont des constantes.
- $x_t \in \mathbb{R}$: variable d'état.
- x_0 : valeur initiale.

A partir de la valeur initiale x_0 on déduit de (1.2) :

- à l'instant $t = 0$: $x_1 = ax_0 + b$.
- à l'instant $t = 1$: $x_2 = ax_1 + b = a(ax_0 + b) + b = a^2x_0 + ab + b$
- à l'instant $t = 2$: $x_3 = ax_2 + b = a(a^2x_0 + ab + b) + b = a^3x_0 + a^2b + ab + b$
- ⋮

En général :

$$x_t = a^t x_0 + a^{t-1}b + a^{t-2}b + \cdots + ab + b$$

$$= a^t x_0 + b \sum_{i=0}^{t-1} a^i, t \in \mathbb{N}^*$$

On a $\sum_{i=0}^{t-1} a^i$ est une somme d'une série géométrique, donc pour $t \in \mathbb{N}^*$:

$$x_t = \begin{cases} a^t x_0 + b \frac{1-a^t}{1-a} & a \neq 1 \\ x_0 + bt & a = 1 \end{cases}$$

Point fixe (d'équilibre)

La notion de points **d'équilibre** est centrale dans l'étude de la dynamique de tout système physique. Dans de nombreuses applications en biologie, économie, physique, ingénierie, etc., il est souhaitable que tous les états (solutions) d'un système donné tendent à son état d'équilibre (point d'équilibre). C'est le sujet d'étude de la théorie de la stabilité, un sujet d'une grande importance pour les scientifiques et les ingénieurs.

Définition 1.5. *Un point d'équilibre (ou point fixe) est une valeur de la variable d'état x_t qui est invariante selon la loi du mouvement dictée par le système dynamique.*

En d'autres termes, le point fixe d'un système dynamique $x_{t+1} = f(x_t)$ est $\bar{x} \in \mathbb{R}$ tel que $\bar{x} = f(\bar{x})$ où $f : \mathbb{R} \rightarrow \mathbb{R}$ une fonction dérivable.

Graphiquement, les points fixes peuvent être trouvés en identifiant les intersections de la fonction $f(x)$ avec la diagonale $y = x$.

Existence et unicité du point fixe

- On suppose que le système (1.2) est à l'état d'équilibre i.e : $\bar{x} = a\bar{x} + b$.
- pour $a \neq 1$, on a : $\bar{x} = \frac{b}{1-a}$. Donc il existe une unique point fixe.
 - pour $a = 1$ et $b = 0$, on a : $\forall t \in \mathbb{N}, x_{t+1} = x_t$. c-à-d toute condition initiale est un point fixe.
 - pour $a = 1$ et $b \neq 0$, le point fixe n'existe pas.

Finalement, on déduit que :

$$\bar{x} = \begin{cases} \frac{b}{1-a} & a \neq 1 \\ x_0 & a = 1 \text{ et } b = 0 \end{cases}$$

Propositions 1.1. *Le point fixe d'un système dynamique $x_{t+1} = ax_t + b$ existe si et seulement si $a \neq 1$ ou ($a = 1$ et $b = 0$).*

Propositions 1.2. *Le point fixe d'un système dynamique $x_{t+1} = ax_t + b$ est unique si et seulement si $a \neq 1$.*

Remarque 1.2. *La solution de (1.2) peut être écrite en termes de son point fixe \bar{x} et la valeur initial x_0 comme suit :*

$$x_t = \begin{cases} a^t (x_0 - \bar{x}) + \bar{x} & a \neq 1 \\ x_0 + bt & a = 1 \end{cases}$$

Stabilité du point fixe

Définition 1.6. On dit que le point fixe \bar{x} d'un système dynamique $x_{t+1} = f(x_t)$ (où $f : \mathbb{R} \rightarrow \mathbb{R}$ une fonction dérivable) est stable si :

$$\forall \varepsilon > 0, \exists \delta > 0 \text{ tel que si } |x_0 - \bar{x}| < \delta \Rightarrow |f^t(x_0) - \bar{x}| < \varepsilon \quad \forall t \in \mathbb{N}^*$$

Autrement dit, toutes les orbites qui commencent près du point \bar{x} restent dans un voisinage de ce point.

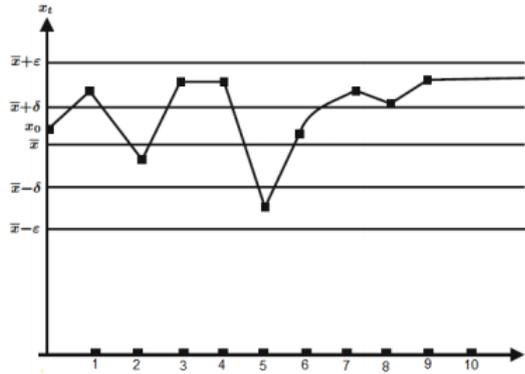


FIGURE 1.2 – Point fixe stable

Définition 1.7. On dit que le point fixe \bar{x} est localement asymptotiquement stable s'il est stable et s'il existe $V_\varepsilon(\bar{x})$ un voisinage de \bar{x} ($V_\varepsilon(\bar{x}) = \{y : |\bar{x} - y| < \varepsilon\}$) tel que

$$\forall x_0 \in V_\varepsilon(\bar{x}), \quad \lim_{t \rightarrow \infty} x_t = \bar{x}$$

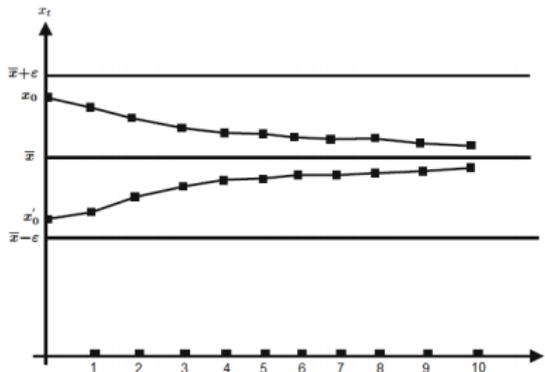


FIGURE 1.3 – Point fixe localement asymptotiquement stable

Définition 1.8. On dit que Le point fixe \bar{x} est globalement asymptotiquement stable s'il est stable et :

$$\lim_{t \rightarrow \infty} x_t = \bar{x}, \quad \forall x_0 \in \mathbb{R}$$

Remarque 1.3. La stabilité globale d'un point fixe implique l'unicité du point fixe et la stabilité locale implique l'unicité local du point fixe.

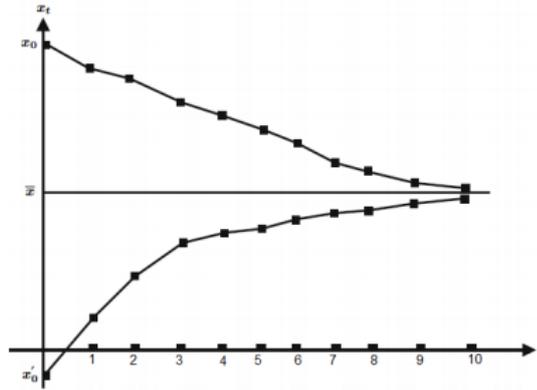


FIGURE 1.4 – Point fixe globalement asymptotiquement stable

Définition 1.9. *Un point fixe s'appelle instable s'il existe un $\varepsilon > 0$ tel que $\forall r > 0$ il existe un $x_0 \in V_r(\bar{x})$ et il existe un $t \in \mathbb{N}$ tels que $|f^t(x_0) - \bar{x}| > \varepsilon$.*

Cela signifie que pour tout voisinage du point fixe \bar{x} il existe une orbite qui, en commençant dans ce voisinage s'éloigne du point \bar{x} .

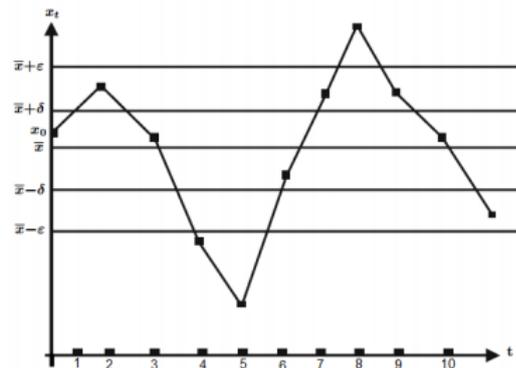


FIGURE 1.5 – Point fixe instable

D'après les définitions de la stabilité asymptotique locale et globale, la stabilité asymptotique du point fixe peut être obtenue en calculons la limite de du variable d'état x_t .

On a :

$$\lim_{t \rightarrow \infty} |x_t| = \begin{cases} \lim |(x_0 - \bar{x}) a^t + \bar{x}| & \text{si } a \neq 1 \\ \lim_{t \rightarrow \infty} |x_0 + bt| & \text{si } a = 1 \end{cases}$$

$$= \begin{cases} |\bar{x}| & \text{si } \begin{cases} |a| < 1 \\ |a| > 1 \text{ et } x_0 = \bar{x} \end{cases} \\ |x_0| & a = 1 \text{ et } b = 0 \\ |x_0| & t \text{ paire et } a = -1 \\ |b - x_0| & t \text{ impaire } a = -1 \\ \infty & \text{autres cas.} \end{cases}$$

On distingue les cas suivants :

1^{ème} cas : Si le coefficient $|a| < 1$ alors le système est globalement asymptotiquement stable, car

$$\lim_{t \rightarrow \infty} x_t = \bar{x}, \forall x_0 \in \mathbb{R}$$

En particulier, si $0 < a < 1$, la convergence de la variable d'état x_t vers le point fixe \bar{x} est monotone.

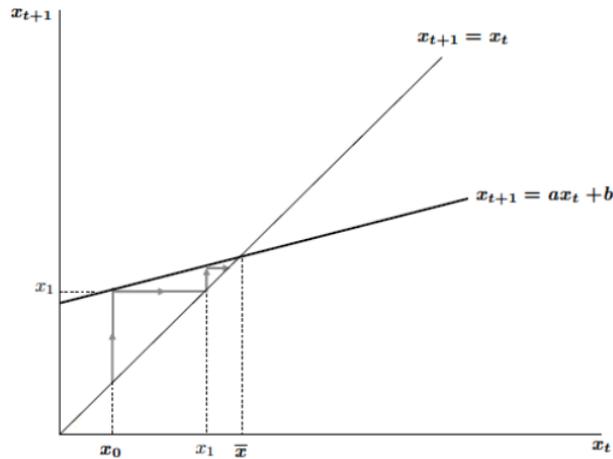


FIGURE 1.6 – Point fixe globalement asymptotiquement stable ($-1 < a < 0$)

Si $-1 < a < 0$, la convergence de la variable d'état x_t vers le point fixe \bar{x} est oscillatoire.

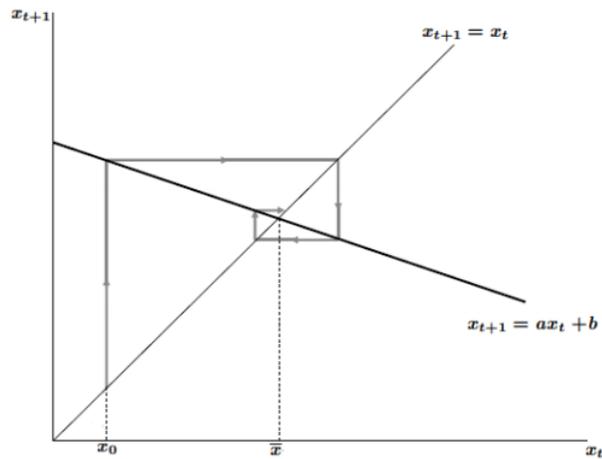
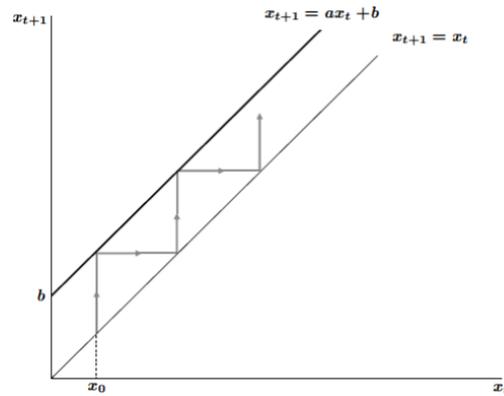


FIGURE 1.7 – Point fixe globalement asymptotiquement stable ($0 < a < 1$)

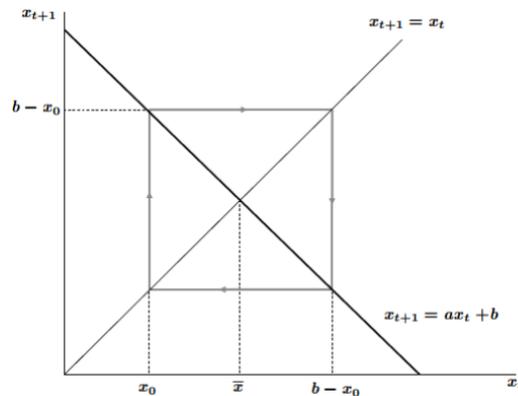
2^{ème} cas : Si $a = 1$ et $b = 0$, le système admet une infinité de points fixes et tout voisinage d'un point fixe contient un autre point fixe et il existe donc des conditions initiales dans tout voisinage d'un point fixe qui ne conduisent pas à ce point fixe à long terme. Donc chaque point fixe est instable.

3^{ème} cas : Si $a = 1$ et $b \neq 0$, n'existe pas des points fixes et de plus

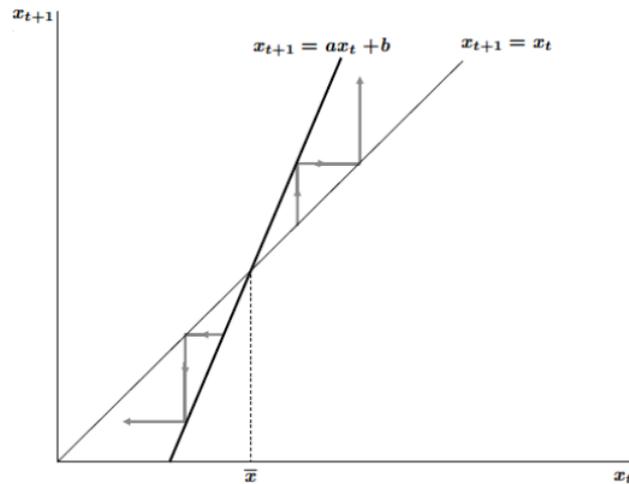
$$\lim_{t \rightarrow \infty} x_t = \lim_{t \rightarrow \infty} x_0 + bt = \begin{cases} +\infty & \text{si } b > 0 \\ -\infty & \text{si } b < 0 \end{cases}$$

FIGURE 1.8 – Non-existence d'un point fixe ($a = 1$ et $b \neq 0$)

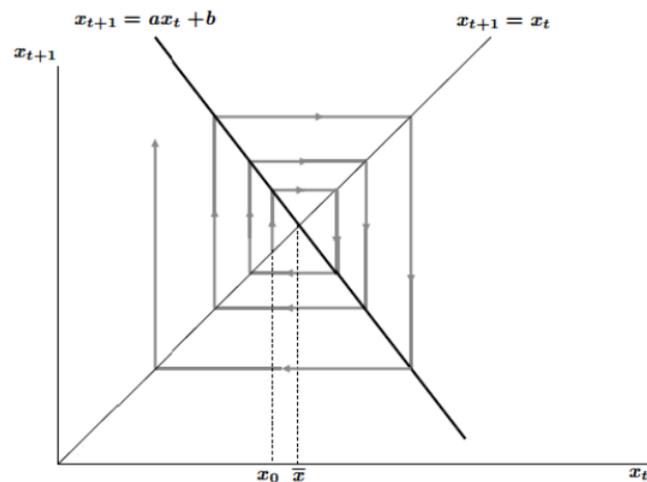
4^{ème} cas : Si $a = -1$ le système prend alternativement les valeurs x_0 et $b - x_0$ donc le point fixe $\bar{x} = \frac{b}{2}$ n'est pas asymptotiquement stable.

FIGURE 1.9 – Point fixe n'est pas asymptotiquement stable ($a = -1$)

4^{ème} cas : Si $|a| > 1$: pour $x_0 \neq \bar{x}$, $\lim_{t \rightarrow \infty} |x_t| = +\infty$ et pour $x_0 = \bar{x}$ le système commence au point fixe où il reste ensuite. Chaque perturbation mineure provoque le système à marcher sur un divergent chemin. Donc si $|a| > 1$ le point fixe $\bar{x} = \frac{b}{1-a}$ est instable. En particulier si $a > 1$, la divergence de la variable d'état x_t est monotone.

FIGURE 1.10 – Point fixe instable ($a > 1$)

Si $a < -1$ la divergence de la variable d'état x_t est oscillatoire .

FIGURE 1.11 – Point fixe instable ($a < -1$)

Propositions 1.3.

- Le point d'équilibre de (1.2) est globalement asymptotiquement stable ssi $|a| < 1$.
- Le point d'équilibre de (1.2) est instable si et seulement si $|a| > 1$.

1.2.2 Systèmes dynamiques discrets non linéaire de dimension 1

Définition 1.10. Un système dynamique discret non linéaire est définie par équation aux différences suivant :

$$\begin{cases} x_{t+1} = f(x_t) \\ x_0 \text{ donné} \end{cases}, t \in \mathbb{N} \quad (1.3)$$

où $f : \mathbb{R} \rightarrow \mathbb{R}$ une fonction dérivable et $x_t \in \mathbb{R}$ est le variable d'état.

Stabilité du point fixe

Il n'est pas facile de trouver des solutions de systèmes non linéaires. Souvent, ces solutions ne fournissent pas suffisamment d'informations sur les facteurs qui contrôlent la stabilité du système. Par conséquent, nous avons besoin de méthodes analytiques pour faciliter l'étude du comportement de ce système non linéaire. L'approximation linéaire du système non linéaire est l'un des moyens les plus efficaces d'étudier la stabilité des systèmes non linéaires.

◦ Méthode de linéarisation :

Le développement de Taylor d'ordre 1 de $f(x_t) = x_{t+1}$ au voisinage \bar{x} est :

$$x_{t+1} = f(x_t) = f(\bar{x}) + f'(\bar{x})(x_t - \bar{x}) + o(x_t - \bar{x})$$

alors :

$$x_{t+1} \simeq f'(\bar{x})x_t + f(\bar{x}) - f'(\bar{x})\bar{x} = ax_t + b$$

où $a = f'(\bar{x})$ et $b = f(\bar{x}) - f'(\bar{x})\bar{x}$.

Donc, on obtient une approximation linéaire de (1.3) au voisinage du point fixe par le développement de Taylor d'ordre 1 de la fonction f .

Nous pouvons utiliser les résultats de stabilité du système linéaire pour étudier la stabilité du système non linéaire.

Étant donné que le système linéaire n'approche du comportement du système non linéaire que dans un voisinage suffisamment petite d'un point fixe, l'analyse globale du système linéarisé ne donne qu'une analyse locale de système dynamique non linéaire.

Propositions 1.4.

- Le point d'équilibre \bar{x} de (1.3) est localement asymptotiquement stable ssi $|f'(\bar{x})| < 1$.
- Le point d'équilibre \bar{x} de (1.3) est instable si et seulement si $|f'(\bar{x})| > 1$.

Remarque 1.4. La stabilité du système non linéaire à proximité d'un point fixe \bar{x} ne peut être étudiée à partir du système linéarisé si $f'(\bar{x}) = 1$. A savoir, tout changement infinitésimal dans la dérivée au point \bar{x} entraîne un changement dans la nature du système dynamique.

Stabilité du point fixe non hyperbolique

Définition 1.11. Un point fixe \bar{x} de (1.3) est dit hyperbolique si on a $|f'(\bar{x})| \neq 1$.

Dans le cas où $|f'(\bar{x})| = 1$ c'est-à-dire \bar{x} non hyperbolique on peut utiliser les théorèmes suivants pour étudier la stabilité du point fixe.

Théorème 1.1. Soit \bar{x} un point fixe de le système (1.3) avec $f \in C^3$ et $f'(\bar{x}) = 1$ et $f''(\bar{x}) = 0$ alors :

- Si $f'''(\bar{x}) > 0$ alors \bar{x} est instable.
- Si $f'''(\bar{x}) < 0$ alors \bar{x} est asymptotiquement stable.

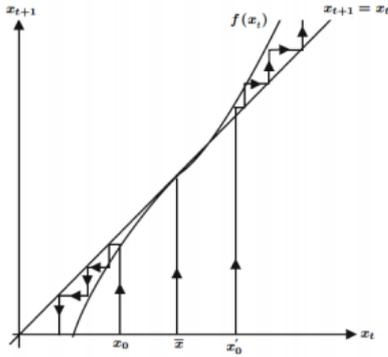


FIGURE 1.12 – Point fixe instable

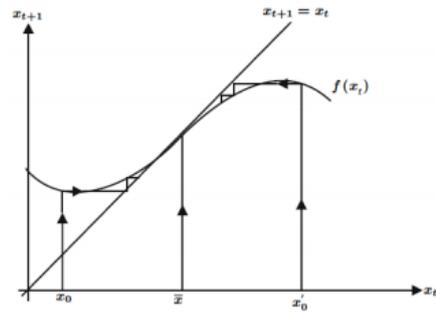


FIGURE 1.13 – Point fixe asymptotiquement stable

Exemple 1.2. Soit le système dynamique $x_{t+1} = x_t^3 + x_t$. On a :

$$f(\bar{x}) = \bar{x} \iff \bar{x}^3 + \bar{x} = \bar{x} \iff \bar{x} = 0$$

Alors, l'unique point fixe de ce système est $\bar{x} = 0$

D'autre part on a : $f'(0) = 1, f''(0) = 0$ et $f'''(0) = 6 > 0$.

Donc d'après le théorème (1.1) on déduit que le point fixe $\bar{x} = 0$ est instable.

Exemple 1.3. Soit le système dynamique $x_{t+1} = -x_t^3 + x_t$. On a :

$$f(\bar{x}) = \bar{x} \iff \bar{x} = 0$$

Alors, l'unique point fixe de ce système est $\bar{x} = 0$

D'autre part on a : $f'(0) = 1, f''(0) = 0$ et $f'''(0) = -6 < 0$.

Donc d'après le théorème (1.1) on déduit que le point fixe $\bar{x} = 0$ est asymptotiquement stable.

Théorème 1.2. Soit \bar{x} est un point fixe du système (1.3) avec $f \in C^2$, si $f'(\bar{x}) = 1$ et $f''(\bar{x}) \neq 0$ alors : \bar{x} est instable.

- Si $f'(\bar{x}) = 1$ et $f''(\bar{x}) > 0$, on dit que \bar{x} est un point fixe semi-stable à gauche .
- Si $f'(\bar{x}) = 1$ et $f''(\bar{x}) < 0$, on dit que \bar{x} est un point fixe semi-stable à droite .

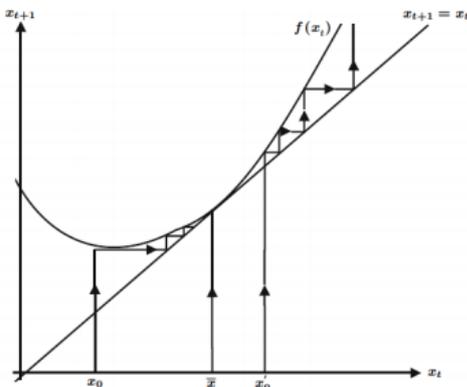


FIGURE 1.14 – Point fixe semi-stable à gauche

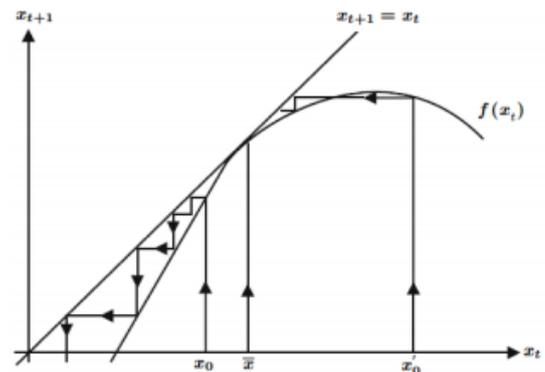


FIGURE 1.15 – Point fixe semi-stable à droite

Définition 1.12.

La dérivée Schwartzienne d'une fonction $f \in C^3$ est définie par :

$$Sf(x) = \frac{f'''(x)}{f'(x)} - \frac{3}{2} \left(\frac{f''(x)}{f'(x)} \right)^2$$

Notons que si $f'(x) = -1$, alors

$$Sf(x) = -f'''(x) - \frac{3}{2} [f''(x)]^2$$

Théorème 1.3. Si \bar{x} est un point fixe du système (1.3) avec $f \in C^3$ et $f'(\bar{x}) = -1$ alors :

- Si $Sf(\bar{x}) > 0$ alors \bar{x} est instable.
- Si $Sf(\bar{x}) < 0$ alors \bar{x} est localement asymptotiquement stable.

Preuve. Soit le système dynamique

$$x_{t+1} = g(x_t), \text{ où } g(x) = f^2(x) \quad (1.4)$$

On remarque que le point fixe \bar{x} de (1.3) est aussi un point fixe de (1.4) et si \bar{x} est asymptotiquement stable (instable) par rapport à (1.4), il est aussi asymptotiquement stable (instable) par rapport à (1.3). On a :

$$g'(x) = [f(f(x))]' = f'(x)f'(f(x)) \Rightarrow g'(\bar{x}) = [f'(\bar{x})]^2 = 1$$

Le théorème (1.1) s'applique donc à cette situation. Nous devons évaluer $g''(\bar{x})$:

$$g''(x) = [f'(x)f'(f(x))]' = [f'(x)]^2 f''(f(x)) + f'(f(x))f''(x) \Rightarrow g''(\bar{x}) = 0$$

Le théorème (1.1) dit que la stabilité asymptotique de \bar{x} est déterminée par le signe de $g'''(\bar{x})$. On a :

$$\begin{aligned} g'''(x) &= \left[[f'(x)]^2 f''(f(x)) + f'(f(x))f''(x) \right]' \\ &= 2f'(x)f''(x)f''(f(x)) + f'(x)f'''(f(x)) [f'(x)]^2 + f'(x)f''(f(x))f''(x) + f'''(x)f'(f(x)) \end{aligned}$$

Alors

$$g'''(\bar{x}) = -2f'''(\bar{x}) - 3[f''(\bar{x})]^2$$

Donc

$$g'''(\bar{x}) > 0 \iff Sf(\bar{x}) > 0 \text{ et } g'''(\bar{x}) < 0 \iff Sf(\bar{x}) < 0$$

Exemple 1.4. Soit le système dynamique $x_{t+1} = x_t^2 + 3x_t$. On a :

$$f(\bar{x}) = \bar{x} \iff \bar{x}^2 + 3\bar{x} = \bar{x} \iff \bar{x} = 0 \text{ ou } \bar{x} = -2$$

Alors, les points fixes de ce système sont : $\bar{x} = 0$ et $\bar{x} = -2$

D'autre part on a : $f'(0) = 3 > 1$

Donc d'après la proposition (1.3.1) le point fixe $\bar{x} = 0$ est instable. et on a $f'(-2) = -1$ et $Sf(-2) = -6 < 0$

D'après le théorème (1.3) on déduit que le point fixe $\bar{x} = -2$ est asymptotiquement stable.

Définition 1.13. On dit que $f : \mathbb{R} \rightarrow \mathbb{R}$ est contractante si $\exists \alpha \in]0, 1[$ tel que

$$|f(x_1) - f(x_2)| \leq \alpha |x_1 - x_2|, \forall x_1, x_2 \in \mathbb{R}$$

Corollaire 1.1. Le point fixe de (1.3) existe et est unique et globalement asymptotiquement stable si f est contractante ou si $|f'(x_t)| < 1, \forall x_t \in \mathbb{R}$.

Preuve. Nous allons d'abord démontrer que la suite $(x_t)_{t \in \mathbb{N}}$ est une suite de Cauchy : Observons que pour tout entier naturel t :

$$|x_{t+1} - x_t| = |f(x_t) - f(x_{t-1})| \leq \alpha |x_t - x_{t-1}| \leq \alpha^2 |x_{t-1} - x_{t-2}| \leq \dots \leq \alpha^t |x_1 - x_0|$$

Par inégalité triangulaire, on peut écrire :

$$|x_{t+p} - x_t| \leq \sum_{i=0}^{p-1} |x_{t+i+1} - x_{t+i}| \leq \sum_{i=0}^{p-1} \alpha^{t+i} |x_1 - x_0| \leq \alpha^t \frac{1 - \alpha^p}{1 - \alpha} |x_1 - x_0| \leq \frac{\alpha^t}{1 - \alpha} |x_1 - x_0|$$

Comme α^t tend vers 0 quand t tend vers l'infini (car $0 < \alpha < 1$), on en conclut que $|x_{t+p} - x_t|$ peut être rendu aussi petit que l'on veut pourvu que t soit suffisamment grand. Ceci prouve que la suite (x_t) est une suite de Cauchy. Donc La suite (x_t) converge donc vers $\bar{x} \in \mathbb{R}, \forall x_0 \in \mathbb{R}$ Montrons que \bar{x} est un point fixe de f :

On a par définition $\forall t \in \mathbb{N} \quad x_{t+1} = f(x_t)$ Comme f est une application contractante elle est continue donc par passage à la limite ($t \rightarrow \infty$) on obtient $\bar{x} = f(\bar{x})$

Montrons maintenant que ce point fixe est unique.

Supposons donc qu'il existe deux points fixes $\bar{x}_1, \bar{x}_2 \in \mathbb{R}$ de (2.3). On peut alors écrire :

$$|\bar{x}_1 - \bar{x}_2| = |f(\bar{x}_1) - f(\bar{x}_2)| \leq \alpha |\bar{x}_1 - \bar{x}_2| < |\bar{x}_1 - \bar{x}_2|$$

ce qui est impossible. En conséquence, le point fixe est nécessairement unique.

La condition $|f'(x)| \leq k < 1, \forall x \in \mathbb{R}$ implique que f est contractante.

Soient $x, y \in \mathbb{R}$, d'après le théorème des accroissements finis, on peut écrire : $f(x) - f(y) = f'(c)(x - y)$ ou $c \in]x, y[$ et par suite :

$$|f(x) - f(y)| = |f'(c)| |x - y| < k|x - y|$$

ce qui prouve que f est contractante.

1.2.3 Systèmes dynamiques discrets linéaires de dimension n .

Définition 1.14. Un système dynamique discret linéaire de dimension n est un systèmes de n équation aux différences linéaires d'ordre 1 i.e :

$$\begin{cases} x_{1t+1} = a_{11}x_{1t} + a_{12}x_{2t} + \dots + a_{1n}x_{nt} + b_1 \\ x_{2t+1} = a_{21}x_{1t} + a_{22}x_{2t} + \dots + a_{2n}x_{nt} + b_2 \\ \vdots \\ x_{nt+1} = a_{n1}x_{1t} + a_{n2}x_{2t} + \dots + a_{nn}x_{nt} + b_n \end{cases}$$

où $t \in \mathbb{N}$ et $X_0 = (x_{10}, x_{20}, \dots, x_{n0})$ sont donnés.

L'écriture matricielle d'un système dynamique discret linéaire de dimension n est :

$$\begin{cases} X_{t+1} = AX_t + B \\ X_0 \quad \text{donné} \end{cases}, t \in \mathbb{N} \quad (1.5)$$

- $A \in M_n(\mathbb{R})$: matrice de constantes réelles de taille $n \times n$.
- $B \in \mathbb{R}^n$: vecteur de constantes.
- $X_t \in \mathbb{R}^n$: vecteur des états du système.
- $X_0 \in \mathbb{R}^n$: vecteur initial.
- $x_{it} \in \mathbb{R}$: variable d'état.

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}, \quad X_t = \begin{pmatrix} x_{1t} \\ x_{2t} \\ \vdots \\ x_{nt} \end{pmatrix}, \quad X_0 = \begin{pmatrix} x_{10} \\ x_{20} \\ \vdots \\ x_{n0} \end{pmatrix}, \quad B = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

À partir d'un point X_0 , le système (1.5) donne :

pour $t = 1$: $X_1 = AX_0 + B$

pour $t = 2$: $X_2 = AX_1 + B = A(AX_0 + B) + B = A^2X_0 + AB + B$.

pour $t = 3$: $X_3 = AX_2 + B = A(A^2X_0 + AB + B) + B = A^3X_0 + A^2B + AB + B$

⋮

En général

$$\begin{aligned} X_t &= A^tX_0 + A^{t-1}B + A^{t-2}B + \cdots + AB + B \\ &= A^tX_0 + \sum_{i=0}^{t-1} A^iB, \forall t \in \mathbb{N}^* \end{aligned}$$

Lemme 1.1. Si $\det(I - A) \neq 0$ alors :

$$\sum_{i=0}^{t-1} A^i = (I - A^t)(I - A)^{-1}.$$

Preuve. On a :

$$\sum_{i=0}^{t-1} A^i(I - A) = I + A + A^2 + \cdots + A^{t-1} - (A + A^2 + \cdots + A^t) = I - A^t$$

Alors :

$$\sum_{i=0}^{t-1} A^i = (I - A^t)(I - A)^{-1}.$$

Selon le lemme précédent, on peut écrire la solution de (1.5) comme suit :

$$X_t = A^t(X_0 - (I - A)^{-1}B) + (I - A)^{-1}B, \text{ si } \det(I - A) \neq 0.$$

Point fixe

Définition 1.15. Un point fixe d'un système dynamique $X_{t+1} = f(X_t)$ est $\bar{X} \in \mathbb{R}^n$ tel que :

$$\bar{X} = f(\bar{X}) \text{ où } \bar{X} = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)^t$$

Existence du point fixe :

On suppose que le système (1.5) est à l'état d'équilibre c'est-à-dire : $\bar{X} = A\bar{X} + B$. Alors :

$$\bar{X} = (I - A)^{-1}B, \text{ si } \det(I - A) \neq 0$$

Donc l'unique point fixe de (1.5) si $\det(I - A) \neq 0$ est :

$$\bar{X} = (I - A)^{-1}B$$

Si $\det(I - A) = 0$, il y a un nombre infini de points fixes.

La solution de (1.5) peut être écrite en termes de son point fixe \bar{X} et la valeur initiale X_0 comme suit :

$$X_t = A^t (X_0 - \bar{X}) + \bar{X}$$

si $\det(I - A) \neq 0$ où $\bar{X} = (I - A)^{-1}B$

Propositions 1.5. *Le point fixe de système (1.5) est unique si et seulement si $\det(I - A) \neq 0$.*

Stabilité du point fixe

Maintenant dans tout ce que suit nous ne considérons que $\det(I - A) \neq 0$.

Définition 1.16. *On dit que le point fixe \bar{X} d'un système dynamique $X_{t+1} = f(X_t)$ où f est une fonction de \mathbb{R}^n vers \mathbb{R}^n est stable si :*

$$\forall \varepsilon > 0, \exists \delta > 0 \text{ tq si } \|X_0 - \bar{X}\| < \delta \Rightarrow \|f^t(X_0) - \bar{X}\| < \varepsilon, \forall t \in \mathbb{N}^*$$

(Dans le cas où $f(X_t) = AX_t + B$ on parle d'un système dynamique linéaire).

Définition 1.17. *Le point fixe \bar{X} est localement asymptotiquement stable s'il est stable et si : $\exists V$ un voisinage de \bar{X} tel que :*

$$\forall X_0 \in V \quad \lim_{t \rightarrow \infty} X_t = \bar{X}$$

Définition 1.18. *Le point fixe \bar{X} est globalement asymptotiquement stable s'il est stable et si :*

$$\lim_{t \rightarrow \infty} X_t = \bar{X} \quad \forall X_0 \in \mathbb{R}^n$$

L'étude de la stabilité du point fixe \bar{X} de (1.5) n'est pas facile parce que nous avons une difficulté à la formule de solution de ce système dynamique, qui est dans la formule de la matrice A^t . Pour cela nous allons utiliser la décomposition de Jordan

Théorème 1.4. *Pour toute matrice A de taille $n \times n$ il existe une matrice P de taille $n \times n$ non singulière tel que $A = PJP^{-1}$ où J est une matrice diagonale par bloc c'est-à-dire :*

$$J = \begin{pmatrix} J_{\lambda_1} & 0 & \cdots & 0 \\ 0 & J_{\lambda_2} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & J_{\lambda_d} \end{pmatrix}$$

La matrice J est appelée forme normale de Jordan et

$$J_{\lambda_k} = \begin{pmatrix} \lambda_k & 1 & 0 & \cdots & 0 \\ 0 & \lambda_k & 1 & \cdots & 0 \\ & & \ddots & \ddots & \vdots \\ \vdots & & & \lambda_k & 1 \\ 0 & \cdots & 0 & 0 & \lambda_k \end{pmatrix}$$

s'appelle bloc de Jordan associé à la valeur propre λ_k de A .

Remarque 1.5.

- Les valeurs propres $\lambda_1, \lambda_2, \dots, \lambda_d$ ne sont pas nécessairement distinctes.
- Chaque bloc J_{λ_k} $k = 1, \dots, d$ ($d \leq n$) est de taille $m_k \times m_k$ ($m_k \geq 1$), avec $\sum_{k=1}^d m_k = n$.
- La taille d'un bloc de Jordan J_{λ_k} peut être égale à 1.

Étude de la stabilité à l'aide de forme normale de Jordan

D'après le théorème (1.4) :

$$X_t = A^t (X_0 - \bar{X}) + \bar{X} = PJ^tP^{-1} (X_0 - \bar{X}) + \bar{X}$$

Propositions 1.6. La solution de (1.5) est donné par :

$$X_t = PJ^tP^{-1} (X_0 - \bar{X}) + \bar{X}$$

où J est la matrice de Jordan.

Il n'est pas difficile de trouver une formule pour la matrice J^t . On a :

$$J^t = \begin{pmatrix} J_{\lambda_1}^t & 0 & \cdots & 0 \\ 0 & J_{\lambda_2}^t & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & J_{\lambda_d}^t \end{pmatrix}$$

Il reste de trouver la formule de chaque bloc.

- Si chaque bloc J_{λ_k} est de taille $1 \times 1 \forall k = 1, \dots, d$ (dans ce cas $d = n$) on dit que A est diagonalisable et on a :

$$J^t = \begin{pmatrix} \lambda_1^t & 0 & \cdots & 0 \\ 0 & \lambda_2^t & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n^t \end{pmatrix}$$

Le système dynamique (1.5) est globalement asymptotiquement stable si :

$$\lim_{t \rightarrow \infty} X_t = \bar{X} \quad \forall X_0 \in \mathbb{R}^n$$

$$\forall X_0 \in \mathbb{R}^n, \text{ on a : } \lim_{t \rightarrow \infty} X_t = \bar{X} \iff \lim_{t \rightarrow \infty} PJ^tP^{-1} (X_0 - \bar{X}) + \bar{X} = \bar{X} \iff \lim_{t \rightarrow \infty} J^t = 0.$$

Dans ce cas

$$\lim_{t \rightarrow \infty} J^t = 0 \iff |\lambda_k| < 1, \forall k = 1, \dots, n$$

- S'il existe un bloc J_{λ_k} de taille $m_k \times m_k$ ($m_k \geq 2$) alors

$$J_{\lambda_k} = \lambda_k I_{d_k} + N_k$$

où

I_{m_k} : La matrice identité de taille $m_k \times m_k$.

N_k : Une matrice de taille $m_k \times m_k$, tq :

$$N_k = \begin{pmatrix} 0 & 1 & & 0 \\ & 0 & \ddots & \vdots \\ \vdots & & \ddots & 1 \\ 0 & & & 0 \end{pmatrix}, N_k^{m_k-1} = \begin{pmatrix} 0 & 0 & \cdots & 1 \\ \vdots & \ddots & & 0 \\ & & \ddots & \vdots \\ 0 & & & 0 \end{pmatrix} \text{ et } N_k^t = 0, \forall t \geq m_k$$

Pour $t \geq m_k$, on a :

$$J_{\lambda_k}^t = (\lambda_k I_{m_k} + N_k)^t = \sum_{i=0}^{m_k-1} C_t^i \lambda_k^{t-i} N_k^i$$

Pour que le système dynamique (1.5) est globalement asymptotiquement stable il faut et il suffit que : $\lim_{t \rightarrow \infty} J^t = 0$. Dans ce cas :

$$\lim_{t \rightarrow \infty} J^t = 0 \iff \lim_{t \rightarrow \infty} J_{\lambda_k}^t = 0, \forall k = 1, \dots, d (d \leq n) \iff |\lambda_k| < 1, \forall k = 1, \dots, d$$

Corollaire 1.2. — *Le système dynamique (1.5) est globalement asymptotiquement stable si et seulement si le module de chaque valeur propre de la matrice A est inférieur à 1.*

— *Le système dynamique (1.5) est instable si et seulement s'il existe un valeur propre de la matrice A est de module supérieur à 1.*

1.2.4 Systèmes dynamiques discrets non linéaires de dimension n .

Définition 1.19. *Un système dynamique discret non linéaire de dimension n est un systèmes de n équations aux différences non linéaires d'ordre 1, i.e :*

$$\begin{cases} x_{1t+1} = f_1(x_{1t}, x_{2t}, \dots, x_{nt}) \\ x_{2t+1} = f_2(x_{1t}, x_{2t}, \dots, x_{nt}) \\ \vdots \\ x_{nt+1} = f_n(x_{1t}, x_{2t}, \dots, x_{nt}) \end{cases}$$

où $t \in \mathbb{N}$ et $X_0 = (x_{10}, x_{20}, \dots, x_{n0})$ sont donnés. Ce système dynamique est écrit comme suit.

$$\begin{cases} X_{t+1} = f(X_t) \\ X_0 \text{ donné} \end{cases} \quad t \in \mathbb{N} \quad (1.6)$$

où : $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ une fonction différentiable.

$$f(X_t) = (f_1(X_t), f_2(X_t), \dots, f_n(X_t))$$

$f_i : \mathbb{R}^n \rightarrow \mathbb{R} \quad \forall i = 1, \dots, n$ une fonction différentiable.

$X_t \in \mathbb{R}^n$: vecteur des états du système.

$X_0 \in \mathbb{R}^n$: valeur initial.

Stabilité du point fixe

STABILITÉ LOCALE :

Comme nous l'avons vu dans le cas d'un système dynamique de dimension 1 , le système dynamique (1.6) peut être aussi approximé par un système linéaire.

On suppose que le système dynamique (1.6) a un point fixe \bar{X} alors le développement de Taylor d'ordre 1 de $f_i(X_t) = x_{it+1}$ au voisinage \bar{X} est :

$$\begin{aligned} x_{it+1} &= f_i(X_t) = f_i(\bar{X}) + \sum_{k=1}^n \frac{\partial f_i(\bar{X})}{\partial x_{kt}} (x_{kt} - \bar{x}_k) + o(\|X_t - \bar{X}\|) \\ &= \frac{\partial f_i(\bar{X})}{\partial x_{1t}} x_{1t} + \frac{\partial f_i(\bar{X})}{\partial x_{2t}} x_{2t} + \dots + \frac{\partial f_i(\bar{X})}{\partial x_{nt}} x_{nt} + f_i(\bar{X}) - \sum_{k=1}^n \frac{\partial f_i(\bar{X})}{\partial x_{kt}} \bar{x}_k + o(\|X_t - \bar{X}\|) \end{aligned}$$

Où

$$o(\|X_t - \bar{X}\|) \rightarrow 0 \text{ quand } X_t \rightarrow \bar{X}$$

Donc le développement de Taylor d'ordre 1 de $f(X_t) = X_{t+1}$ au voisinage \bar{X} est :

$$\begin{pmatrix} x_{1t+1} \\ x_{2t+1} \\ \vdots \\ x_{nt+1} \end{pmatrix} \simeq \begin{pmatrix} \frac{\partial f_1(\bar{X})}{\partial x_{1t}} & \frac{\partial f_1(\bar{X})}{\partial x_{2t}} & \cdots & \frac{\partial f_1(\bar{X})}{\partial x_{nt}} \\ \frac{\partial f_2(\bar{X})}{\partial x_{1t}} & \frac{\partial f_2(\bar{X})}{\partial x_{2t}} & \cdots & \frac{\partial f_2(\bar{X})}{\partial x_{nt}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n(\bar{X})}{\partial x_{1t}} & \frac{\partial f_n(\bar{X})}{\partial x_{2t}} & \cdots & \frac{\partial f_n(\bar{X})}{\partial x_{nt}} \end{pmatrix} \begin{pmatrix} x_{1t} \\ x_{2t} \\ \vdots \\ x_{nt} \end{pmatrix} + \begin{pmatrix} f_1(\bar{X}) - \sum_{k=1}^n \frac{\partial f_1(\bar{X})}{\partial x_{kt}} \\ f_2(\bar{X}) - \sum_{k=1}^n \frac{\partial f_2(\bar{X})}{\partial x_{kt}} \\ \vdots \\ f_n(\bar{X}) - \sum_{k=1}^n \frac{\partial f_n(\bar{X})}{\partial x_{kt}} \end{pmatrix}$$

Le système non linéaire est approximée au voisinage du point fixe \bar{X} par un système linéaire

$$X_{t+1} \simeq AX_t + B.$$

Où :

$$A = \begin{pmatrix} \frac{\partial f_1(\bar{X})}{\partial x_{1t}} & \frac{\partial f_1(\bar{X})}{\partial x_{2t}} & \cdots & \frac{\partial f_1(\bar{X})}{\partial x_{nt}} \\ \frac{\partial f_2(\bar{X})}{\partial x_{1t}} & \frac{\partial f_2(\bar{X})}{\partial x_{2t}} & \cdots & \frac{\partial f_2(\bar{X})}{\partial x_{nt}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n(\bar{X})}{\partial x_{1t}} & \frac{\partial f_n(\bar{X})}{\partial x_{2t}} & \cdots & \frac{\partial f_n(\bar{X})}{\partial x_{nt}} \end{pmatrix} \text{ et } B = \begin{pmatrix} f_1(\bar{X}) - \sum_{k=1}^n \frac{\partial f_1(\bar{X})}{\partial x_{kt}} \\ f_2(\bar{X}) - \sum_{k=1}^n \frac{\partial f_2(\bar{X})}{\partial x_{kt}} \\ \vdots \\ f_n(\bar{X}) - \sum_{k=1}^n \frac{\partial f_n(\bar{X})}{\partial x_{kt}} \end{pmatrix}$$

La matrice A s'appelle la matrice jacobienne de f au point \bar{X} et on la note par $J(\bar{X})$.

Nous pouvons donc utiliser les résultats du système linéaire pour étudier la stabilité du système non linéaire au voisinage du point fixe \bar{X} . Ceci est basé sur les valeurs propres du Jacobien $J(\bar{X})$ de l'application f .

Théorème 1.5. Soit le système (1.6), on suppose que ce système a un point fixe \bar{X} et soit $\lambda_1, \dots, \lambda_p$ ($p \leq n$) les valeurs propres de $J(\bar{X})$ la matrice jacobienne de f au point \bar{X} . Alors :

1. Si $|\lambda_i| < 1 \quad \forall i = 1, \dots, p$, alors \bar{X} est localement asymptotiquement stable.
2. S'il existe une valeur propre $\lambda_i, i = 1, \dots, p$ tel que $|\lambda_i| > 1$, alors \bar{X} est instable.
3. Si $\max_{1 \leq i \leq p} |\lambda_i| = 1$ nous ne pouvons rien conclure.

Théorème 1.6. Le point fixe de (1.6) existe et est unique et globalement asymptotiquement stable si $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ est contractante.

Méthode de Lyapunov

Soit le système dynamique (1.6) où $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ une fonction continue. On suppose que \bar{X} est un point fixe de ce système dynamique.

Soit $V : \mathbb{R}^n \rightarrow \mathbb{R}$. La variation de V par rapport à le système dynamique (1.6) est définie par :

$$\Delta V(X_t) = V(f(X_t)) - V(X_t) = V(X_{t+1}) - V(X_t)$$

Définition 1.20. Soit G un ouvert de \mathbb{R}^n , une fonction V de G vers \mathbb{R} est une fonction de Liapunov sur l'ensemble G si :

1. V est continue sur G .
2. $\Delta V(X_t) \leq 0, \forall X_t, X_{t+1} \in G$

Définition 1.21. La fonction de Lyapunov V est dite définie positive au point fixe \bar{X} s'il existe une boule ouverte $B_\varepsilon(\bar{X})$ de centre \bar{X} et de rayon ε ($B_\varepsilon(\bar{X}) = \{Y \in \mathbb{R}^n, \|\bar{X} - Y\| < \varepsilon\}$) telle que :

1. $V(\bar{X}) = 0$.
2. $V(X_t) > 0$ pour tout $X_t \in B_\varepsilon(\bar{X})$, $X_t \neq \bar{X}$.

Théorème 1.7. — S'il existe V une fonction de Lyapunov sur une boule ouverte $B_r(\bar{X})$ définie positive au \bar{X} où \bar{X} est le point fixe de (1.6) alors \bar{X} est stable.

- Si en plus, $\Delta V(X_t) < 0, \forall X_t, X_{t+1} \in G, X_t \neq \bar{X}$ alors \bar{X} est localement asymptotiquement stable.
- Si cela est vrai quand $B_r(\bar{X})$ est étendu à tout \mathbb{R}^n et $V(X_t) \rightarrow \infty$ quand $\|X_t\| \rightarrow \infty$ alors \bar{X} est globalement asymptotiquement stable.

Preuve. Pour la démonstration de ce théorème voir [11].

Exemple 1.5. Soit le système dynamique

$$\begin{cases} x_{1t+1} = \frac{x_{2t}}{1+x_{2t}^2} \\ x_{2t+1} = \frac{x_{1t}}{1+x_{2t}^2} \end{cases}$$

On a :

$$\begin{cases} \bar{x}_1 = \frac{\bar{x}_2}{1+\bar{x}_2^2} \\ \bar{x}_2 = \frac{\bar{x}_1}{1+\bar{x}_2^2} \end{cases} \iff (\bar{x}_1, \bar{x}_2) = (0, 0)$$

Donc le point fixe de ce système est $\bar{X} = (\bar{x}_1, \bar{x}_2) = (0, 0)$.

On définit la fonction

$$V(x_1, x_2) = x_1^2 + x_2^2$$

On a la fonction V est continue sur \mathbb{R}^2 et

$$V(X_{t+1}) = \frac{x_{2t}^2}{(1+x_{2t}^2)^2} + \frac{x_{1t}^2}{(1+x_{2t}^2)^2} = \frac{V(X_t)}{(1+x_{2t}^2)^2} \leq V(X_t), \forall X_t, X_{t+1} \in \mathbb{R}^2$$

Donc V est une fonction de Lyapunov.

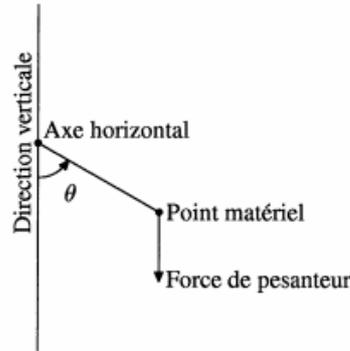
On a aussi $V(\bar{X}) = 0$ et $V(X_t) > 0; \forall X_t \in \mathbb{R}^2, X_t \neq \bar{X}$

Alors V est définie positive, en déduit de le théorème (1.7) que le point fixe $\bar{X} = (0, 0)$ est asymptotiquement stable.

1.3 Systèmes dynamiques continus

Exemple 1.6. Le mouvement d'un pendule plan :

Le pendule est constitué par un point matériel de masse m , attaché à une extrémité d'une tige rectiligne rigide de longueur l et de masse négligeable. La tige peut tourner, autour de son autre extrémité, dans un plan vertical fixe. Le frottement est supposé négligeable. On appelle θ l'angle (orienté) que fait le pendule avec la verticale, cet angle étant 0 si le pendule est en position verticale avec le point matériel au dessous du point d'attache, et $\pm\pi$ si le pendule est en position verticale avec le point matériel au dessus du point d'attache.



Les équations du mouvement sont :

$$\begin{cases} \frac{d\theta}{dt} = \omega \\ \frac{d\omega}{dt} = -\frac{g}{l} \sin \theta \end{cases}$$

Nous avons noté g l'accélération de la pesanteur. Cette équation différentielle est associée à un champ de vecteurs qui est le générateur infinitésimal d'un système dynamique à temps continu.

1.3.1 Préliminaires et définitions

Un système dynamique continu autonome peut être représenté par l'équation :

$$\begin{cases} \dot{x}(t) = f(x(t)) \\ x(t_0) = x_0 \end{cases} \quad (1.7)$$

On suppose que pour toute condition initiale x_0 , il existe une unique solution définie sur \mathbb{R} du problème de Cauchy (1.7). Nous noterons indifféremment $x(t, x_0)$ ou $x(t)$ cette solution.

Définition 1.22. *L'état x_e est appelé état d'équilibre ou point d'équilibre pour le système (1.7) si lorsque $x(t_e) = x_e$ alors $x(t) = x_e$ pour tout $t \geq t_e$. En d'autres termes, x_e vérifie l'équation $f(x_e) = 0$.*

Remarque 1.6. *On peut toujours se ramener au cas où le point d'équilibre est l'origine 0 puisque si x_e vérifie $f(x_e) = 0$, il suffit de considérer le changement de coordonnées $z = x - x_e$, la dérivée de z est donnée par*

$$\dot{z} = \dot{x} = f(x) = f(z + x_e) \stackrel{\text{d'éf}}{=} g(z), \text{ et } g(0) = 0$$

l'origine est bien un point d'équilibre du système $\dot{z} = g(z)$.

Définition 1.23. *(Définition intuitive de la stabilité)*

Si le système dynamique est "légèrement" perturbé de son point d'équilibre, le même système reste "proche" de ce point d'équilibre. On dira alors que le point d'équilibre est stable.

Cette définition intuitive de la stabilité traduit la capacité d'un système dynamique, pour des conditions initiales données, à rester très proche d'un point d'équilibre suite à une perturbation. Considérons comme exemple illustratif une bille sur une surface sphérique comme montré sur

la figure 1.16. Dans le schéma de gauche de la figure, suite à une légère perturbation de la position d'équilibre, la bille reste proche de sa position d'équilibre appelée stable ; alors que dans le schéma de droite, la bille ne reste pas proche de sa position d'équilibre qui est alors appelée instable.

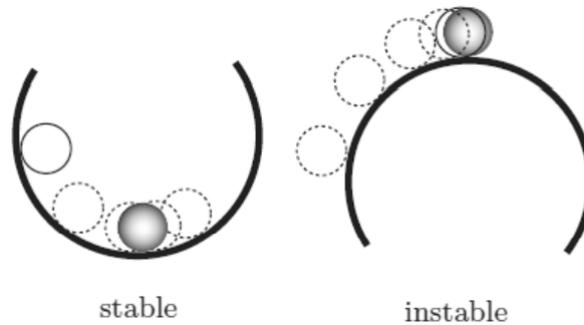


FIGURE 1.16 – Illustration de la définition intuitive de la stabilité

La traduction mathématique de cette définition intuitive de la stabilité est donnée par la définition suivante

1.3.2 Notions de stabilité

Définition 1.24. (*Stabilité au sens de Lyapunov*)

Le point d'équilibre x_e est dit stable si $\forall \varepsilon > 0$, il existe $\delta(\varepsilon) > 0$ tel que :

$$\text{si } \|x_0 - x_e\| \leq \delta \text{ alors } \|x(t) - x_e\| \leq \varepsilon, \quad \forall t \geq 0$$

Si non le point d'équilibre est dit instable.

Cette définition signifie que, quelle que soit la boule d'exigence de rayon ε , il est toujours possible de choisir une certaine sous-boule de rayon δ telle que pour toute condition initiale comprise dans cette sous-boule, la trajectoire résultante sera, en tout temps, comprise dans la boule d'exigence de rayon ε .

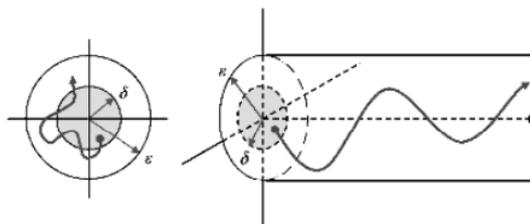


FIGURE 1.17 – Illustration de la stabilité au sens de Lyapunov

Dans un langage plus imagé, un petit déséquilibre initial n'entraîne qu'un petit déséquilibre au cours du temps, déséquilibre qui peut très bien être permanent.

Il existe plusieurs notions de stabilité :

- Stabilité asymptotique.
- Stabilité exponentielle.
- Stabilité locale et globale.

Définition 1.25. (*Stabilité asymptotique*)

Le point d'équilibre x_e est asymptotiquement stable s'il est stable et attractif, i.e., s'il existe $\delta > 0$ tel que $\forall x_0 \in B(x_e, \delta)$, $\lim_{t \rightarrow \infty} x(t) = x_e$

La stabilité asymptotique exige l'existence d'un voisinage de l'équilibre tel que toute trajectoire ayant pour condition initiale un point de ce voisinage converge vers le point d'équilibre.

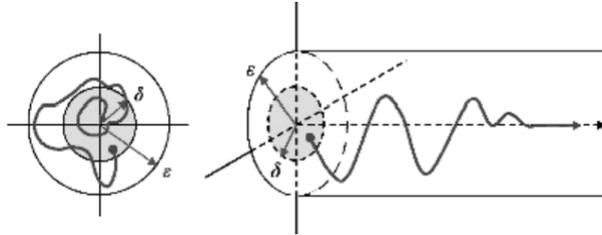


FIGURE 1.18 – Illustration de la stabilité asymptotique

Cependant, la définition de la stabilité asymptotique ne donne pas une idée sur la rapidité de convergence de la trajectoire $x(t)$ vers le point d'équilibre x_e . D'où la notion de stabilité exponentielle.

Définition 1.26. (*Stabilité exponentielle*)

Le point d'équilibre x_e est exponentiellement stable s'il existe $\alpha > 0$ tel que, quel que soit $\varepsilon > 0$, il existe $\delta(\varepsilon)$ tel que

$$\forall t \geq t_0, \|x_0 - x_e\| \leq \delta \Rightarrow \|x(t) - x_e\| < \varepsilon \|x_0 - x_e\| \exp(-\alpha(t - t_0))$$

Cette définition traduit le fait que toute trajectoire issue d'une condition initiale appartenant à la boule ouverte de rayon δ converge vers le point d'équilibre x_e plus rapidement qu'une fonction exponentielle. α est appelé dans ce cas le taux de convergence. On note que la stabilité exponentielle implique la stabilité asymptotique ainsi que la stabilité au sens de Lyapunov.

Exemple 1.7. *Considérons le système décrit par*

$$\dot{x} = -\left(1 + \sin(x^2)\right)x.$$

Il est clair que $x_e = 0$ est un point d'équilibre. La solution du système est donnée par

$$x(t) = x(0) \exp \int_0^t -\left(1 + \sin(x^2(s))\right) ds$$

On a $\forall t \geq 0$, $|x(t)| < k|x(0)| \exp(-t)$. D'où la stabilité exponentielle du système en question.

Dans chacune des définitions précédentes la stabilité est définie localement, puisque les conditions initiales sont prises dans un voisinage $\mathcal{V}(x_e)$ autour du point d'équilibre x_e . Si $\mathcal{V}(x_e) = \mathbb{R}^n$, le point d'équilibre x_e est dit globalement asymptotiquement (exponentiellement) stable.

Définition 1.27. (*Stabilité Globale*)

Si la condition de stabilité asymptotique (resp. exponentielle) est vérifiée dans tout \mathbb{R}^n , le point d'équilibre est globalement asymptotiquement (resp. exponentiellement) stable.

On note que les définitions de la stabilité de Lyapunov, de la stabilité asymptotique et de la stabilité exponentielle présentent quelques inconvénients :

- Il est difficile de calculer de manière explicite chaque solution correspondante à chacune des conditions initiales.
- Ces définitions sont exprimées sous la forme de propositions mathématiques qu'il n'est pas simple de vérifier (trouver $\delta, \varepsilon, \alpha \dots$ etc.).

C'est pourquoi on présente dans la suite des résultats permettant de déterminer la stabilité d'un système dynamique sans recours à l'intégration des équations différentielles.

1.3.3 Stabilité des systèmes linéaires

Dans ce paragraphe, on s'intéresse à l'étude de la stabilité des systèmes linéaires continus régis par l'équation

$$\begin{cases} \dot{x}(t) = Ax(t) \\ x(0) = x_0 \in \mathbb{R}^n \end{cases} \quad (1.8)$$

où A est une matrice carrée de dimension $n \times n$.

Définition 1.28. *Le système continu (1.8) est dit stable si l'origine est stable et il est dit asymptotiquement stable si l'origine est asymptotiquement stable. Dans ce cas, la matrice du système A est dite stable ou asymptotiquement stable.*

La stabilité des systèmes linéaires continus est caractérisée par le résultat suivant :

Propositions 1.7. *Le système (1.8) est stable si et seulement si :*

1. $\Re(\lambda) \leq 0$ pour toute valeur propre λ de A .
2. S'il existe une valeur propre λ de multiplicité k telle que $\Re(\lambda) = 0$ alors $\dim E_\lambda = k$, où E_λ est le sous espace propre associé à λ .

La démonstration se base essentiellement sur la décomposition de la matrice du système sous forme de Jordan.

Exemple 1.8. 1. *Considérons le système*

$$\begin{cases} \dot{x}_1(t) = x_1(t) - x_2(t) \\ \dot{x}_2(t) = 2x_1(t) - 2x_2(t) \end{cases}$$

Les valeurs propres de A sont $\lambda_1 = 0$ et $\lambda_2 = -1$, sont simples, d'où le système est stable.

2. *Considérons le système*

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = 0 \end{cases}$$

la seule valeur propre de A est 0 de multiplicité 2 et $\dim E_0 = 1$, donc le système est instable

Pour la stabilité asymptotique, on a le résultat suivant :

Propositions 1.8. *Le système (1.8) est asymptotiquement stable si et seulement si pour toute valeur propre λ de A on a $\Re(\lambda) < 0$.*

Preuve. Soient $\lambda_j, j = 1, \dots, p$, les valeurs propres distinctes de A , la solution du système (1.8) s'écrit

$$x(t) = \sum_{j=1}^p g_j(t) e^{\lambda_j t}$$

où $g_j(t)$ est un polynôme en t , donc $\lim_{t \rightarrow \infty} x(t) = 0$ si et seulement si $\Re(\lambda_j) < 0$, pour $j = 1, \dots, p$.

1.3.4 Fonction de Lyapunov et méthode directe de Lyapunov

Considérons à nouveau le système (1.7), et soit x_e un point d'équilibre de ce système, alors on a la définition suivante.

Définition 1.29. Une fonction V définie sur une région Ω qui contient x_e est une fonction de Lyapunov pour le système (1.7) et le point d'équilibre x_e si :

1. V est continue et ses dérivées partielles sont continues.
2. V admet un minimum unique en x_e sur Ω .
3. La fonction $\dot{V}(x) = \nabla V(x) \cdot f(x)$ satisfait $\dot{V}(x) \leq 0$ sur Ω .

Si $x(t)$ est la trajectoire de (1.7) alors $V(x(t))$ représente la valeur de V le long de la trajectoire. Afin que V soit décroissante le long de la trajectoire, on doit avoir $\dot{V}(x(t)) \leq 0$.

$$\dot{V}(x(t)) = \frac{\partial V}{\partial x_1} \dot{x}_1(t) + \frac{\partial V}{\partial x_2} \dot{x}_2(t) + \cdots + \frac{\partial V}{\partial x_n} \dot{x}_n(t)$$

utilisant (1.7), on obtient

$$\dot{V}(x(t)) = \frac{\partial V}{\partial x_1} f_1(x(t)) + \frac{\partial V}{\partial x_2} f_2(x(t)) + \cdots + \frac{\partial V}{\partial x_n} f_n(x(t)) = \nabla V(x(t)) \cdot f(x(t))$$

avec $f = (f_1, \dots, f_n)$, $x = (x_1, \dots, x_n)^T$ et $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$

Théorème 1.8. 1 S'il existe une fonction de Lyapunov $V(x)$ associée au système (1.7) dans une boule $B(x_e, R_0)$, alors le point d'équilibre x_e est stable.

2 Si, de plus, la fonction $\dot{V}(x) < 0$ en tout point sauf au point x_e , alors x_e est asymptotiquement stable.

Preuve :

Pour toute condition initiale x_0 dans $B(x_e, R_0)$, V est décroissante le long de la trajectoire. Donc $V(x(t)) \leq V(x_0)$ pour tout $t \geq 0$, ou encore $x(t) \in B(x_e, R_0)$

- 1) La preuve de la stabilité du point d'équilibre est identique au cas discret.
- 2) Si maintenant $\dot{V}(x) < 0$ en tout point de $B(x_e, R_0)$ sauf x_e , alors $V(x(t))$ est positive et décroît, donc converge.

Soit $\lim_{t \rightarrow \infty} V(x(t)) = l$. Alors $\lim_{t \rightarrow \infty} \dot{V}(x(t)) = 0$. Sinon, il existe une constante $a > 0$ telle que $0 < a \leq -\dot{V}(x(t))$. On a

$$V(x(t)) - V(x(0)) = \int_0^t \dot{V}(x(s)) ds \leq -ta$$

et par conséquent

$$V(x_e) \leq V(x(t)) \leq V(x(0)) - ta$$

ce qui est contradictoire lorsque $t \rightarrow \infty$.

Et par suite

$$\lim_{t \rightarrow \infty} \dot{V}(x(t)) = 0$$

Or $\dot{V}(x) < 0$ pour tout $x \in B(x_e, R_0)$ sauf x_e ; par conséquent

$$\lim_{t \rightarrow \infty} V(x(t)) = V(x_e)$$

On conclut donc que $\lim_{t \rightarrow \infty} x(t) = x_e$ pour tout $t \geq 0$.

Exemple 1.9. Soit le système régi par l'équation de Vanderpool donnée par

$$\ddot{x} + \varepsilon(x^2 - 1)\dot{x} + x = 0, \quad \varepsilon < 0$$

qui s'écrit encore sous la forme

$$\begin{cases} \dot{x} = y - \varepsilon\left(\frac{x^3}{3} - x\right) \\ \dot{y} = -x \end{cases}$$

Le seul point d'équilibre est $(0, 0)$.

On pose

$$V(x, y) = \frac{x^2 + y^2}{2}$$

Alors le long de la trajectoire on a

$$\Delta V(x, y) = y\dot{y} + x\dot{x} = -\varepsilon x^2 \left(\frac{x^2}{3} - 1\right)$$

Dans $\Omega = \{(x, y) / x^2 + y^2 < 3\}$, V est une fonction de Lyapunov associée au système avec $\dot{V} < 0$ pour $x^2 + y^2 \neq 0$. On conclut donc, que le point d'équilibre $(0, 0)$ est asymptotiquement stable.

Soit le système autonome suivant

$$\begin{cases} \dot{x} = f(x) \\ f(0) = 0 \end{cases}$$

où $f \in C^1$ et soit $A(x) = \frac{\partial f}{\partial x}$ la matrice jacobienne de f . Nous avons le résultat préliminaire suivant.

Propositions 1.9. *Si la matrice $F = A + A^T$ est définie négative sur un voisinage Ω de 0 , alors le point d'équilibre 0 est asymptotiquement stable.*

Preuve : Soit $x \in \Omega$ et $y \in \mathbb{R}^n$.

$$\langle F(x)y, y \rangle = \langle (A(x) + A^T(x))y, y \rangle = 2 \langle A(x)y, y \rangle < 0$$

Donc $A(x)$ est inversible pour tout $x \in \Omega$. D'après le théorème de la fonction inverse, f est bijective et puisque $f(0) = 0$, alors pour tout $x \in \Omega - \{0\}$, $f(x) \neq 0$.

Considérons la fonction V définie par

$$V(x) = \langle f(x), f(x) \rangle$$

On a $V(0) = 0$ et $V(x) > 0$ pour $x \in \Omega - \{0\}$. Calculons la dérivée de V le long de la trajectoire. On a

$$\dot{V}(x) = \langle F(x)f(x), f(x) \rangle < 0 \text{ pour } x \neq 0$$

On conclut donc que le point d'équilibre est asymptotiquement stable.

Remarque 1.7. *Une situation importante est celle où une fonction de Lyapunov peut être trouvée et reste constante le long de la trajectoire, ce qui correspond à $\dot{V}(x) = 0$. Dans ce cas on conclut que toute trajectoire du système se trouve dans le contour de la fonction V .*

Les théorèmes énoncés ci-dessus concernent la stabilité locale du point d'équilibre. Pour avoir la stabilité asymptotique globale d'un système, il serait logique d'étendre le voisinage (ou la boule) Ω à tout l'espace.

Théorème 1.9. (*Stabilité globale*)

Supposons qu'il existe une fonction V de Lyapunov définie sur \mathbb{R}^n , associée au système (1.7) de point d'équilibre x_e , telle que $\dot{V} < 0$ sauf en x_e et

$$V(x) \longrightarrow \infty \text{ quand } \|x\| \longrightarrow \infty$$

alors le point d'équilibre x_e est globalement asymptotiquement stable.

Preuve :

La preuve est identique au cas local ; en considérant le fait que la fonction V est non bornée et que $\dot{V} < 0$ sauf en x_e , on conclut que pour une condition initiale donnée x_0 , la trajectoire reste dans une région bornée définie par $V(x) \leq V(x_0)$.

Remarquons que le résultat de la stabilité globale implique aussi que le point x_e est le seul point d'équilibre du système.

Exemple 1.10. *Considérons le système*

$$\begin{cases} \dot{x}_1 = x_2 - x_1(x_1^2 + x_2^2) \\ \dot{x}_2 = -x_1 - x_2(x_1^2 + x_2^2) \end{cases}$$

L'origine est un point d'équilibre de ce système. Soit la fonction V définie par $V(x) = x_1^2 + x_2^2$. La dérivée de V le long de la trajectoire est

$$\dot{V}(x) = 2x_1\dot{x}_1 + 2x_2\dot{x}_2 = -2(x_1^2 + x_2^2)^2$$

On conclut donc que l'origine est globalement asymptotiquement stable.

1.3.5 Méthode indirecte de Lyapunov

Une approximation locale de la dynamique du système autour du point d'équilibre permet, dans certains cas, de déduire la stabilité locale du système complet. Il s'agit de la méthode indirecte de Lyapunov.

Considérons le système

$$\dot{x} = f(x)$$

où $f : D \rightarrow \mathbb{R}^n$ est une fonction de classe $C^1(D)$ et D une partie de \mathbb{R}^n contenant le point d'équilibre du système, i.e. $f(0) = 0$. Par le théorème de la moyenne, nous avons

$$f_i(x) = f_i(0) + \frac{\partial f_i}{\partial x}(z_i)x$$

où z_i est un point du segment joignant x et l'origine. Cette égalité est vraie pour tout $x \in D$ tel le segment joignant x et l'origine reste dans D . Comme $f(0) = 0$, nous pouvons écrire

$$f_i(x) = \frac{\partial f_i}{\partial x}(z_i)x = \frac{\partial f_i}{\partial x}(0)x + \left[\frac{\partial f_i}{\partial x}(z_i) - \frac{\partial f_i}{\partial x}(0) \right] x$$

D'où l'on peut déduire que

$$f(x) = Ax + g(x)$$

avec

$$A = \frac{\partial f}{\partial x}(0) \quad \text{et} \quad g_i(x) = \left[\frac{\partial f_i}{\partial x}(z_i) - \frac{\partial f_i}{\partial x}(0) \right] x$$

La fonction $g_i (i = 1, \dots, n)$ vérifie

$$|g_i(x)| \leq \left\| \frac{\partial f_i}{\partial x}(z_i) - \frac{\partial f_i}{\partial x}(0) \right\| \|x\|$$

et par la continuité de la dérivée partielle $\frac{\partial f}{\partial x}$, on remarque que

$$\lim_{\|x\| \rightarrow 0} \frac{\|g(x)\|}{\|x\|} = 0$$

Cela signifie que dans un voisinage de zéro suffisamment petit, on peut approcher le système (1.7) par son linéarisé

$$\dot{x} = Ax, \text{ où } A = \frac{\partial f}{\partial x}(0). \quad (1.9)$$

Le théorème qui suit, connu comme méthode indirecte de Lyapunov, utilise la linéarisation (1.9) du système (1.7) et peut dans certain cas apporter une réponse au problème de la stabilité locale, plus précisément :

Théorème 1.10. *Soit $x = 0$ un point d'équilibre du système*

$$\dot{x} = f(x)$$

où $f : D \rightarrow \mathbb{R}^n$ est une fonction de classe $C^1(D)$ et D un voisinage de zéro.

Soit

$$A = \left. \frac{\partial f}{\partial x}(x) \right|_{x=0}$$

Alors,

1. L'origine est asymptotiquement stable si $\Re(\lambda_i) < 0$ pour toutes les valeurs propres λ_i de A .
2. L'origine est instable si $\Re(\lambda_i) > 0$ pour au moins une valeur propre λ_i de A .

Notons que ce théorème ne permet pas de conclure sur la stabilité du système lorsque $\Re(\lambda_i) \leq 0$ pour tout i et $\Re(\lambda_i) = 0$ pour un certain i .

Exemple 1.11. *Considérons le système*

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -a \sin x_1 - b \sin x_2 \end{aligned}$$

Nous sommes en présence de deux points d'équilibres : $(x_1 = 0, x_2 = 0)$ et $(x_1 = \pi, x_2 = 0)$. La matrice jacobienne est donnée par

$$\frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -a \cos x_1 & -b \end{bmatrix}$$

Pour étudier la stabilité de l'origine, on calcule la jacobienne au point $x = 0$

$$A = \left. \frac{\partial f}{\partial x}(x) \right|_{x=0} = \begin{bmatrix} 0 & 1 \\ -a & -b \end{bmatrix}$$

Les valeurs propres de A sont

$$\lambda_{1,2} = -\frac{1}{2}b \pm \frac{1}{2}\sqrt{b^2 - 4a}$$

Pour tout $a, b > 0$, les valeurs propres de A sont purement réelles et sont strictement négatives. On conclut donc que le système est asymptotiquement stable à l'origine.

Pour étudier la stabilité du point d'équilibre $(x_1 = \pi, x_2 = 0)$, on calcule la jacobienne en ce point :

$$\tilde{A} = \left. \frac{\partial f}{\partial x}(x) \right|_{x_1=\pi, x_2=0} = \begin{bmatrix} 0 & 1 \\ a & -b \end{bmatrix}$$

Les valeurs propres de \tilde{A} sont

$$\lambda_{1,2} = -\frac{1}{2}b \pm \frac{1}{2}\sqrt{b^2 + 4a}$$

Pour tout $a > 0$ et $b \geq 0$, nous avons $\lambda_1 = -\frac{1}{2}b + \frac{1}{2}\sqrt{b^2 + 4a} > 0$. En vertu du théorème 2.10, le point d'équilibre $(x_1 = \pi, x_2 = 0)$ est instable.

1.3.6 Fonctions de Lyapunov pour les systèmes linéaires continus

Considérons le système linéaire

$$\dot{x} = Ax \tag{1.10}$$

Lorsque toutes les valeurs propres de A vérifient $\Re(\lambda_i) < 0$, alors A est dite matrice de Hurwitz ou matrice de stabilité. Supposons que la matrice A est inversible, ce là, nous garantirait l'unicité du point d'équilibre $x = 0$. Rappelons que l'origine du système (1.10) est asymptotiquement stable si et seulement si la matrice A est de Hurwitz. On peut également caractériser la stabilité asymptotique en utilisant la méthode de Lyapunov. Considérons la fonction de Lyapunov suivante

$$V(x) = x^T P x$$

où P est une matrice réelle symétrique définie positive. La dérivée de V au long des trajectoires du système (1.10) est donnée par

$$\dot{V}(x) = \dot{x}^T P x + x^T P \dot{x} = x^T (PA + A^T P) x = -x^T Q x$$

où Q est la matrice symétrique donnée par

$$Q = -(PA + A^T P) \tag{1.11}$$

L'équation (1.11) est dite équation de Lyapunov.

Le théorème suivant caractérise la stabilité asymptotique de l'origine en terme de solution de l'équation de Lyapunov :

Théorème 1.11. *Une matrice A est de Hurwitz si et seulement si pour toute matrice symétrique définie positive Q donnée, il existe une matrice P symétrique définie positive vérifiant l'équation de Lyapunov (1.11). Si de plus A est de Hurwitz, alors P est l'unique solution de l'équation (1.8).*

Preuve : La suffisance découle du théorème 1.8 avec $V(x) = x^T P x$. Pour montrer la nécessité, supposons que toutes les valeurs propres de A vérifient $\Re(\lambda_i) < 0$, et considérons la matrice P définie par

$$P = \int_0^\infty \exp(A^T t) Q \exp(At) dt$$

Cette intégrale est bien définie du fait que les éléments à intégrer sont de la forme $t^{k-1} \exp(\lambda_i t)$, avec $\Re(\lambda_i) < 0$. La matrice P est clairement symétrique.

Montrons qu'elle est définie positive. Supposons le contraire, il existe donc un vecteur $x \neq 0$ tel que $x^T P x = 0$. Comme la matrice $\exp(At)$ est inversible pour tout t , il vient que

$$\begin{aligned} x^T P x = 0 &\implies \int_0^\infty x^T \exp(A^T t) Q \exp(At) x dt = 0 \\ &\implies \exp(At)x \equiv 0, \forall t \geq 0 \implies x = 0. \end{aligned}$$

Cette contradiction montre que P est définie positive. En remplaçant l'expression de P dans l'équation de Lyapunov, on obtient

$$\begin{aligned} PA + A^T P &= \int_0^\infty \exp(A^T t) Q \exp(At) A dt + \int_0^\infty A^T \exp(A^T t) Q \exp(At) dt \\ &= \int_0^\infty \frac{d}{dt} \exp(A^T t) Q \exp(At) dt = \exp(A^T t) Q \exp(At) \Big|_0^\infty = -Q \end{aligned}$$

ce qui montre que P est bien une solution de l'équation de Lyapunov.

Pour montrer que c'est l'unique solution, supposons l'existence d'une autre solution $\tilde{P} \neq P$. Donc

$$(P - \tilde{P})A + A^T(P - \tilde{P}) = 0$$

il vient que

$$0 = \exp(A^T t) [(P - \tilde{P})A + A^T(P - \tilde{P})] \exp(At) = \frac{d}{dt} \{ \exp(A^T t) (P - \tilde{P}) \exp(At) \}$$

autrement dit

$$\exp(A^T t) (P - \tilde{P}) \exp(At) \equiv \text{constante pour tout } t$$

En particulier, comme $\exp(A0) = Id$, nous avons

$$(P - \tilde{P}) = \exp(A^T t) (P - \tilde{P}) \exp(At) \rightarrow 0 \text{ lorsque } t \rightarrow \infty$$

Finalement $P = \tilde{P}$

Remarque 1.8. L'équation (1.11) est une équation linéaire algébrique qui peut se résoudre en la réécrivant sous la forme $Mx = y$, où x et y sont deux vecteurs constitués des éléments de P et Q . Elle peut également être résolue en la considérant comme cas particulier de l'équation de Sylvester $PA + BP + C = 0$. Il existe de nombreuses méthodes numériques efficaces pour la résolution de telles équations (voir [16]) Notons que la plus part des logiciels de programmation des systèmes de contrôle contiennent des commandes pour la résolution de l'équation de Lyapunov.

Exemple 1.12. Soient

$$A = \begin{bmatrix} 0 & -1 \\ 1 & -1 \end{bmatrix}, \quad Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \text{et} \quad P = \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix}.$$

L'équation de Lyapunov a la forme suivante

$$\begin{bmatrix} 0 & 2 & 0 \\ -1 & -1 & 1 \\ 0 & -2 & -2 \end{bmatrix} \begin{bmatrix} p_{11} \\ p_{12} \\ p_{22} \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ -1 \end{bmatrix}$$

L'unique solution de cette équation est

$$\begin{bmatrix} p_{11} \\ p_{12} \\ p_{22} \end{bmatrix} = \begin{bmatrix} 1.5 \\ -0.5 \\ 1.0 \end{bmatrix} \implies P = \begin{bmatrix} 1.5 & -0.5 \\ -0.5 & 1.0 \end{bmatrix}$$

1.3.7 Méthode de construction des fonctions de Lyapunov : Méthode de Krasovskii

Il existe plusieurs méthodes de construction de fonctions de Lyapunov, entre autre la méthode de Krasovskii et la méthode du gradient conjuguée. Le trait commun de ces méthodes est de procéder par construction/correction et essais/erreurs. En d'autres termes, il n'y pas de méthode constructive directe à proprement dit. Il s'agit de procéder de manière itérative en alternant entre, d'une part, imposer la première condition de positivité de la fonction de Lyapunov, et, d'autre part, imposer la seconde condition concernant la décroissance le long des solutions de la fonction de Lyapunov. Dans ce qui suit, nous présentons la méthode de Krasovskii.

Théorème 1.12. *Soit $\dot{x} = f(x)$ tel que $f(0) = 0$. Définissons :*

$$A(x) = \frac{\partial f}{\partial x}$$

S'il existe un ouvert $\Omega \subseteq \mathbb{R}^n$ contenant l'origine $0 \in \Omega$, et que la matrice $F(x) = A(x) + A(x)^T < 0$ est définie négative dans l'ouvert Ω (i.e. $x^T F(x)x < 0, \forall x \neq 0, x \in \Omega$) alors la fonction

$$V(x) = f(x)^T f(x)$$

est une fonction de Lyapunov et l'origine 0 est localement asymptotiquement stable. Si de plus, $\Omega = \mathbb{R}^n$ et $V(x) \rightarrow \infty$ lorsque $\|x\| \rightarrow \infty$, alors $x = 0$ est globalement stable.

Ce théorème admet une certaine généralisation en considérant une équation de Lyapunov pour la matrice $F(x)$.

Théorème 1.13. *([21]) S'il existe un ouvert $\Omega \subseteq \mathbb{R}^n$, ainsi que deux matrices définies positive $P > 0$ et $Q > 0$ telles que $\forall x \neq 0, x \in \Omega$ il est vrai que*

$$F(x) = A(x)^T P + P A(x) + Q < 0$$

est une matrice définie négative, alors

$$V(x) = f(x)^T P f(x)$$

est une fonction de Lyapunov, et 0 est localement asymptotiquement stable. Si de plus, $\Omega = \mathbb{R}^n$ et $V(x) \rightarrow \infty$ lorsque $\|x\| \rightarrow \infty$, alors $x = 0$ est globalement asymptotiquement stable.

1.4 Conclusion

Tout d'abord nous avons donné les définitions des systèmes dynamiques. Puis nous avons donné un rapide aperçu des résultats relatifs à la stabilité des systèmes dynamiques. Et nous avons rappelé les résultats classiques de la théorie de Lyapunov concernant la stabilité dans un sens local et dans un sens global. Ensuite nous avons présenté quelques résultats classiques relatifs à la stabilité des systèmes linéaires. Nous avons rappelé que la stabilité d'un système linéaire peut être caractérisée par les valeurs propres de la matrice dynamique du système aussi bien que par la deuxième méthode de Lyapunov.

Ces résultats et définitions serviront ainsi de base, directement ou indirectement, pour l'étude de la stabilité du réseau de Hopfield dans le chapitre suivant.

Chapitre 2

Réseaux de Hopfield

2.1 Introduction

L'intelligence artificielle est devenue un outil d'analyse incontournable dans les domaines les plus variés de la physique et des mathématiques appliquées. Cette discipline est riche au niveau des techniques sophistiquées pour analyser divers problèmes d'optimisation de différents domaines. Parmi ces techniques, nous citons les réseaux de neurones récurrents de Hopfield . Les réseaux de neurones récurrents de Hopfield (CHN) sont communément connus sous le nom de réseau de Hopfield parce que l'apparition de ces derniers a fourni un grand coup de pouce aux réseaux connexionnistes. Les connexions, de réseaux récurrents, relient les sorties aux entrées, ce qui fournit une grande généralité dans son fonctionnement car la sortie du réseau à un certain moment dépendra non seulement des entrées courantes et des poids de connexion, mais aussi des sorties offertes par le réseau dans les moments précédents. Cette caractéristique fait que les réseaux récurrents présentent des propriétés similaires à la mémoire humaine, dans laquelle l'état des sorties des neurones du cerveau dépend, en partie, des entrées précédentes.

Nous commençons donc ce chapitre par l'historique des réseaux de neurones, nous présentons ensuite les définitions essentielles, nous expliquons ce qu'est un neurone formel, ce qu'est un réseau de neurones et nous terminerons par le réseau de de Hopfield, sa structure et son fonctionnement pour résoudre les problèmes l'optimisation combinatoire .

2.2 Historique

Les recherches sur les méthodes neuronales de traitement de l'information en vue de modéliser le comportement du cerveau humain ne sont pas récentes, en :

- 1890 : W. James, célèbre psychologue américain introduit le concept de mémoire associative, et propose ce qui deviendra une loi de fonctionnement pour l'apprentissage sur les réseaux de neurones connue plus tard sous le nom de loi de Hebb.
- 1943 : J. Mc Culloch et W. Pitts, laissent leurs noms à une modélisation du neurone biologique (un neurone au comportement binaire). Ceux sont les premiers à montrer que des réseaux de neurones formels simples peuvent réaliser des fonctions logiques, arithmétiques et symboliques complexes (tout au moins au niveau théorique).
- 1949 : D. Hebb, physiologiste américain explique le conditionnement chez l'animal par les propriétés des neurones eux-mêmes. Ainsi, un conditionnement de type pavlovien tel que, nourrir tous les jours à la même heure un chien, entraine chez cet animal la sécrétion de salive à cette heure précise même en l'absence de nourriture. La loi de modification

des propriétés des connexions entre neurones qu'il propose explique en partie ce type de résultats expérimentaux.

- 1957 : F. Rosenblatt développe le modèle du Perceptron. Il construit le premier neuroordinateur basé sur ce modèle et l'applique au domaine de la reconnaissance de formes. Notons qu'à cet époque les moyens à sa disposition sont limités et c'est une prouesse technologique que de réussir à faire fonctionner correctement cette machine plus de quelques minutes.
- 1960 : B. Widrow, un automaticien, développe le modèle ADALINE (ADaptive LInear NEuron). Dans sa structure, le modèle ressemble au Perceptron, cependant la loi d'apprentissage est différente. Celle-ci est à l'origine de l'algorithme de retro propagation de gradient très utilisé aujourd'hui avec les Perceptrons multicouches. Les réseaux de type ADALINE restent utilisés de nos jours pour certaines applications particulières.
- 1969 : M.L. Minsky et S. Papert publient ensuite un ouvrage qui met en évidence les limitations théoriques du Perceptron. Ces limitations concernent l'impossibilité de traiter des problèmes non linéaires en utilisant ce modèle.
- 1972 : T. Kohonen présente ses travaux sur les mémoires associatives et propose des applications à la reconnaissance de formes.
- 1982 : J.J. Hopfield est un physicien reconnu à qui l'on doit le renouveau d'intérêt pour les réseaux de neurones artificiels. Il présente une théorie du fonctionnement et des possibilités des réseaux de neurones.
- 1983 : La machine de Boltzmann est le premier modèle connu apte à traiter de manière satisfaisante les limitations recensées dans le cas du Perceptron. Mais l'utilisation pratique s'avère difficile, la convergence de l'algorithme étant extrêmement longue (les temps de calcul sont considérables).
- 1985 : La rétropropagation de gradient apparaît. C'est un algorithme d'apprentissage adapté aux réseaux de neurones multicouches (aussi appelés Perceptrons multicouches). Sa découverte réalisée par trois groupes de chercheurs indépendants indique que « la chose était dans l'air ». Dés cette découverte, nous avons la possibilité de réaliser une fonction non linéaire d'entrée/sortie sur un réseau en décomposant cette fonction en une suite d'étapes linéairement séparables. De nos jours, les réseaux multicouches et la rétropropagation de gradient reste le modèle le plus productif au niveau des applications

2.3 Domaines d'application des réseaux de neurones artificiels

Aujourd'hui, les réseaux de neurones artificiels ont de nombreuses applications dans des secteurs très variés :

- ◇ Traitement d'images : reconnaissance de caractères et de signatures, compression d'images, reconnaissance de forme, cryptage, classification, etc.
- ◇ Traitement du signal : filtrage, classification, identification de source, traitement de la parole etc.
- ◇ Contrôle : commande de processus, diagnostic, contrôle qualité, asservissement des robots, systèmes de guidage automatique des automobiles et des avions etc.
- ◇ Défense : guidage des missiles, suivi de cible, reconnaissance du visage, radar, sonar, compression de données, suppression du bruit etc.
- ◇ Optimisation : planification, allocation de ressource, gestion et finances, etc.
- ◇ Simulation : simulation du vol, simulation de boîte noire, prévision météorologique, copie de modèle etc

2.4 Notions de base sur les réseaux de neurones artificiels

2.4.1 Du Neurone biologique au neurone artificiel

Le neurone biologique :

- *Système Nerveux :*

Le cerveau humain, est le meilleur modèle de la machine, polyvalente incroyablement rapide et surtout douée d'une incomparable capacité d'auto organisation. Son comportement est beaucoup plus mystérieux que le comportement de ses cellules de base. Il est constitué d'un grand nombre d'unités biologiques élémentaires (1000 à 10000 synapse par neurone).

Les cellules nerveuses appelées " neurones ", sont les éléments de base du système nerveux central. Elles sont constituées de trois parties essentielles : le corps cellulaire, les dendrites et l'axone figure (2.1).

- *Le corps cellulaire :*

Il contient le noyau du neurone et effectue les transformations biochimiques nécessaires à la synthèse des enzymes et des autres molécules qui assurent la vie de neurone. Sa forme est pyramidale ou sphérique dans la plupart des cas, elle dépend souvent de sa position dans le cerveau. Ce corps cellulaire fait quelques microns de diamètre .

- *Les dendrites :*

Chaque neurone possède une chevelure de dendrites. Celles-ci sont de fines extensions tubulaires, de quelques dixièmes de microns de diamètre et d'une longueur de quelques dizaines de microns. Elles sont les récepteurs principaux du neurone qui servent à capter les signaux qui lui parviennent .

- *L'axone :*

L'axone, qui est à proprement parler la fibre nerveuse, sert de moyen de transport pour les signaux émis par le neurone. Il se distingue des dendrites par sa forme et par les propriétés de sa membrane externe. En effet, il est généralement plus long que les dendrites, et se ramifie à son extrémité, là où il communique avec les autres neurones, alors que les ramifications des dendrites se produisent plutôt près du corps cellulaire.

Pour former le système nerveux, les neurones sont connectés les uns aux autres suivant des répartitions spatiales complexes. La transmission entre deux neurones n'est pas directe. En fait, il existe un espace intercellulaire de quelques dizaines d'Angstrom (10^{-9} m) entre l'axone du neurone et les dendrites d'un autre neurone. La jonction entre deux neurones est appelée la synapse .

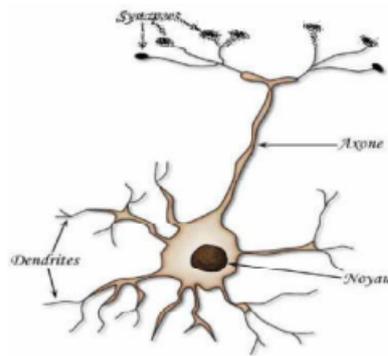


FIGURE 2.1 – Le neurone biologique

Le neurone formel :

La figure (2.2) montre la structure d'un neurone artificiel. Chaque neurone artificiel est un processeur élémentaire. Il reçoit un nombre variable d'entrées en provenance de neurones amont. A chacune de ces entrées est associée un poids w représentatif de la force de la connexion. Chaque processeur élémentaire est doté d'une sortie unique qui se ramifie ensuite pour alimenter un nombre variable de neurones aval. A chaque connexion est associé un poids.

Le modèle général du neurone artificiel est composé des éléments suivants :

- Une ou plusieurs entrées pondérées,
- Un sommateur,
- Une fonction de transfert,
- Une sortie.

Avec :

- x_i est l'entrée i ,
- w_{ij} est la valeur du poids synaptique reliant l'entrée i au neurone j .
- Σ est la somme pondérée des entrées x_i ,
- $F()$ est la fonction de transfert,
- s_j est la sortie du neurone j .

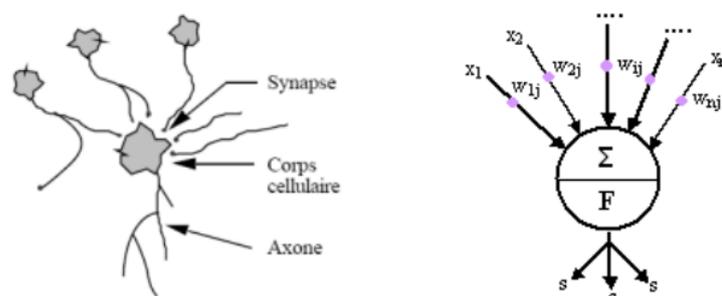


FIGURE 2.2 – Mise en correspondance neurone biologique / neurone artificiel

Définition 2.1. *Un neurone est une fonction algébrique non linéaire, paramétrée à valeurs bornées.*

- Principe de fonctionnement :

L'équation de sortie s_j du neurone j est donnée par :

$$s_j = F(\sum w_{ij} \cdot x_i - w_{0j})$$

Il y a un paramètre additionnel w_{0j} , ce terme est considéré comme la valeur du seuil interne du neurone j .

- Fonction d'activation :

C'est une fonction présentée généralement par une non linéarité appelée aussi fonction de seuil. Elle permet de définir l'état interne du neurone en fonction de son entrée totale. Les fonctions les plus souvent utilisées sont :

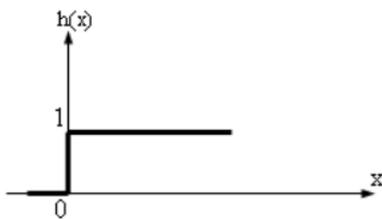


FIGURE 2.3 – Fonction de Heaviside

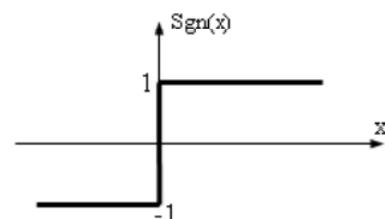


FIGURE 2.4 – Fonction signe

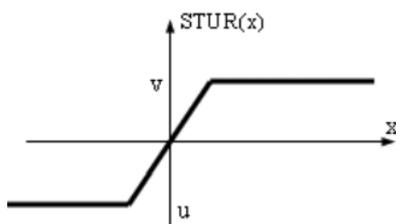


FIGURE 2.5 – Fonction linéaire à seuil

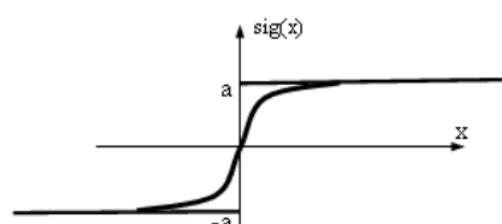


FIGURE 2.6 – Fonction sigmoïde

2.4.2 Les réseaux de neurones

Définition 2.2. *Les réseaux de neurones artificiels (RNA) sont des processeurs élémentaires fortement connectés fonctionnant en parallèle. Chaque processeur élémentaire calcule une sortie unique sur la base des informations qu'il reçoit.*

De point de vue modélisation mathématique, on peut définir un réseau de neurone artificiel par les quatre éléments suivants :

- La nature des entrées et des sorties : Elles peuvent être :
 - Binaire : $(-1; +1)$ ou $(0,1)$
 - Réelles.
- La fonction d'entrée totale qui définit le prétraitement effectué sur les entrées : Elle peut être :
 - Booléenne
 - Linéaire
 - Affine
 - Polynomiale de degré supérieur à deux

- La fonction d'activation du neurone qui définit son état en fonction de son entrée totale.
- La fonction de sortie qui calcule la sortie du réseau en fonction de son état d'activation ; en général cette fonction est considérée comme la fonction identité.

On distingue deux types de réseaux de neurones : Les réseaux non bouclés et les réseaux bouclés.

2.4.3 Réseau de neurones non bouclé

Un réseau de neurones non bouclé est donc représenté graphiquement par un ensemble de neurones connectés entre eux, l'information circulant des entrées vers les sorties sans "retour en arrière". On constate que le graphe d'un réseau non bouclé est acyclique. En effet, si on se déplace dans ce type de réseau à partir d'un neurone quelconque en suivant les connexions, on ne peut pas revenir au neurone de départ. Les neurones qui effectuent le dernier calcul de la composition de fonctions sont les neurones de sortie. Ceux qui effectuent des calculs intermédiaires sont les neurones cachés (figure 2.7).

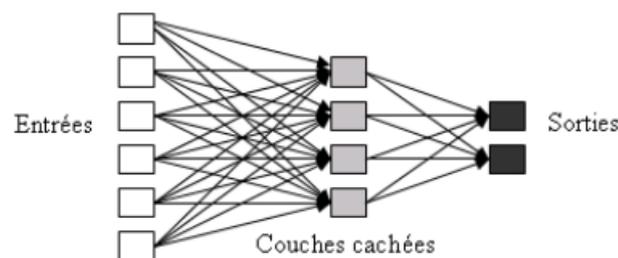


FIGURE 2.7 – Structure d'un réseau de neurones non bouclé.

Les réseaux non bouclés à couche sont structurés tel que les neurones qui appartiennent à une même couche ne soient pas connectés entre eux, chacune des couches recevant des signaux de la couche précédente, et transmettant le résultat de ses traitements à la couche suivante. Les deux couches extrêmes correspondent à la couche d'entrée qui reçoit ses entrées du milieu extérieur d'une part, et à la couche de sortie qui fournit le résultat des traitements effectués d'autre part. Les couches intermédiaires sont appelées couches cachées, leur nombre est variable.

Les réseaux de neurones non bouclés à couches dont les neurones cachés ont une fonction d'activation sigmoïde, sont souvent appelés des perceptrons multicouche (ou MLP pour Multi-Layer Perceptron)

2.4.4 Réseau de neurones bouclé (ou récurrent)

Un réseau de neurones bouclé est schématisé par un graphe des connexions qui est cyclique. Lorsqu'on se déplace dans le réseau en suivant le sens des connexions, il est possible de trouver au moins un chemin qui revient à son point de départ (un tel chemin est désigné sous le terme de "cycle"). Ainsi, un retard entier multiple de l'unité de temps choisie est attaché à chaque connexion d'un réseau de neurones bouclé (ou à chaque arrête de son graphe). Une grandeur à un instant donné ne peut pas être fonction de sa propre valeur au même instant. Tout cycle du

graphe du réseau doit avoir un retard non nul.

La figure (2.8) représente un exemple de réseau de neurones bouclé. Les chiffres dans les carrés indiquent le retard attaché à chaque connexion, exprimé en multiple de l'unité de temps. Ce réseau contient un cycle, qui part du neurone 3 et revient à celui-ci en passant par le neurone 4. La connexion de 4 vers 3 ayant un retard d'une unité de temps.

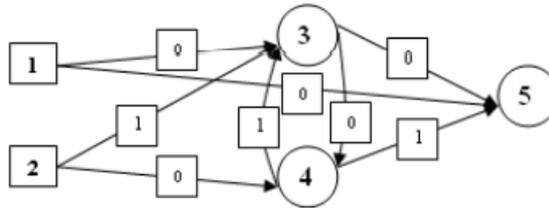


FIGURE 2.8 – Un réseau de neurones bouclé à deux entrées

2.5 Architectures de réseaux de neurones pour l'optimisation

Les réseaux de neurones bouclés (ou récurrents) constituent les techniques neuronales les plus fréquemment utilisées pour résoudre des problèmes d'optimisation. Ces réseaux sont caractérisés par le fait que le graphe de leurs connexions contient au moins un cycle. Pour l'optimisation, on utilise ces réseaux sans entrées de commande : sous leur dynamique propre, ils évoluent, à partir d'un état initial (souvent aléatoire), vers un attracteur qui code une solution du problème d'optimisation.

Dans les réseaux de neurones bouclés utilisés pour l'optimisation, les neurones sont soit binaires (leur fonction d'activation est un échelon entre -1 et $+1$), soit à fonction d'activation sigmoïde : dans ce dernier cas, la sortie x_i du neurone i est donnée par :

$$x_i = \tanh(\gamma u_i)$$

où γ est la pente à l'origine de la sigmoïde, et où u_i est le potentiel du neurone i , défini par : pour un réseau de N neurones mutuellement connectés :

$$u_i = \sum_{j=1}^N w_{ij} x_j + I_i$$

où I_i est l'entrée constante (biais) du neurone i .

Une bonne introduction aux réseaux de neurones appliqués à l'optimisation est fournie dans [48]. Les travaux de Takefuji donnent de nombreux exemples de problèmes combinatoires résolus au moyen de réseaux de neurones récurrents [47].

2.6 Fonctions d'énergie pour l'optimisation combinatoire

L'optimisation combinatoire définit un cadre formel pour de nombreux problèmes de l'industrie, de la finance ou de la vie quotidienne. Les problèmes d'optimisation combinatoire

sont habituellement définis comme une problématique de choix d'une meilleure alternative dans un ensemble très grand mais fini d'alternatives. En raison du très grand nombre d'alternatives pour ces problèmes, l'ensemble des alternatives, dit aussi ensemble de solutions réalisables, est défini en compréhension, en d'autres termes les solutions réalisables se distinguent par un ensemble de propriétés ou de conditions, dites aussi contraintes, qu'elles doivent toutes remplir. Une évaluation est associée à toute solution réalisable à l'aide d'une fonction dite fonction objectif. Résoudre un tel problème consiste donc à trouver une solution optimale, c'est-à-dire trouver une solution réalisable qui minimise ou maximise, selon le contexte, la fonction objectif.

La manière courante de traiter un problème d'optimisation combinatoire avec un réseau de neurones consiste, dans un premier temps, à associer au problème des fonctions d'énergie convenablement construites, puis à transformer le problème de leur minimisation en la résolution d'un système d'équations différentielles ou d'équations aux différences. En général, une telle fonction d'énergie s'écrit sous la forme de la somme d'un terme de coût et d'un terme qui exprime les contraintes

$$E = \text{«Coût»} + \text{«Contraintes»}$$

Dans cette équation, les deux termes sont en concurrence; cela se traduit par l'existence de nombreux minima locaux. On cherche donc à minimiser la fonction de coût tout en maximisant simultanément le nombre de contraintes satisfaites. Mathématiquement, la minimisation de la fonction d'énergie se ramène fréquemment à la minimisation d'une fonction $E(x)$ sur un ensemble fini de points X , généralement choisi comme un hypercube dans un espace de dimension N , $\{-1, 1\}^N$ ou $\{0, 1\}^N$.

Si le vecteur \mathbf{x} regroupe les variables du problème d'optimisation :

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$$

la fonction $E(\mathbf{x})$ est définie de la manière suivante :

$$E(\mathbf{x}) = E_C(\mathbf{x}) + \sum_k a_k E_k(\mathbf{x})$$

où $E_c(\mathbf{x})$ est la fonction de coût, les $E_k(\mathbf{x})$ sont les termes de pénalités associés à des violations de contraintes, et les a_k sont des coefficients de pondération qui doivent assurer un bon équilibre entre la minimisation du coût et la satisfaction des contraintes.

Pour nombre de problèmes d'optimisation, la fonction d'énergie E peut être exprimée par une forme quadratique du type :

$$\begin{aligned} E(\mathbf{x}) &= -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N w_{ij} x_i x_j - \sum_{i=1}^N I_i x_i \\ &= -\frac{1}{2} \mathbf{x}^T \cdot \mathbf{W} \cdot \mathbf{x} - \mathbf{x}^T \cdot \mathbf{I} \end{aligned}$$

où :

$$\mathbf{W} = \begin{pmatrix} w_{11} & w_{12} & w_{13} & \dots & w_{1N} \\ w_{21} & w_{22} & w_{23} & \dots & w_{2N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{N1} & w_{N2} & w_{N3} & \dots & w_{NN} \end{pmatrix} \quad \text{et} \quad \mathbf{I} = \begin{pmatrix} I_1 \\ I_2 \\ \vdots \\ I_N \end{pmatrix}$$

Dans cette expression, les coefficients w_{ij} et I_i sont complètement déterminés par la fonction énergie à minimiser.

Pour une fonction énergie $E(\mathbf{x})$ donnée, les coefficients w_{ij} expriment une courbure locale :

$$w_{ij} = -\frac{\partial^2 E}{\partial x_i \partial x_j}$$

et le coefficient I_i est donné par la dérivée première de l'énergie :

$$I_i = -\left. \frac{\partial E}{\partial x_i} \right|_{x=0}$$

Nous verrons par la suite que la fonction énergie peut être minimisée par un réseau de neurones bouclés, tel celui qui a été proposé par Hopfield.

2.7 Réseau de Hopfield

En 1982, J.J.Hopfield, physicien de l'institut de technologie de Californie, a publié un article sur le développement des réseaux neuronaux, proposant un réseau de neurones récurrent mono-couche, appelé par la suite le réseau de Hopfield.

J.J.Hopfield a introduit le concept de la fonction énergétique dans le réseau de neurones récurrent. Ce concept est d'une grande importance pour l'étude des réseaux de neurones, il fait reposer les jugements de la stabilité du réseau neuronal sur une base fiable.

En 1985, Hopfield avec D.W. Tank ont également mis en place le réseau de Hopfield avec des circuits électroniques analogiques. Ils ont réussi à résoudre le problème du voyageur de commerce, en ouvrant ainsi un réseau neuronal capable de faire un traitement intelligent de l'information, et une nouvelle voie prometteuse pour le traitement des problèmes d'optimisation via les réseaux de neurones.

Contrairement aux réseaux feedforward qui généralement ne tiennent pas compte du lien entre la sortie et l'entrée des neurones, le réseau de Hopfield tient compte du lien entre la sortie et l'entrée des neurones grâce à sa dynamique qui est caractérisée par une équation différentielle. Dans ce contexte, nous proposons plusieurs modélisations de cette équation pour construire un nouveau modèle mathématique de ce type de réseau.

Les poids de ce réseau de Hopfield ne sont pas obtenus par l'apprentissage, mais calculés d'avance selon certaines règles. Les poids ne changeront pas une fois déterminés, et l'état de chaque neurone sera constamment mis à jour jusqu'à l'évolution de réseau vers un état stable. Dans le cas de la stabilité, l'état de chaque neurone est la solution du problème. Le réseau de Hopfield est composé en deux types de modèles : réseau de Hopfield discret (discrete Hopfield neural network) et réseau de Hopfield continu (continuous Hopfield neural network).

2.7.1 Structure et mode de fonctionnement du réseau de Hopfield discret

La structure du réseau de Hopfield discret est illustrée dans la figure (2.9). Il s'agit d'un réseau récurrent composé d'une seule couche avec n neurones. Dans ce réseau, la sortie de n'importe quel neurone x_i est renvoyée comme entrée à tous les neurones x_j à travers la connexion w_{ij} .

En d'autre terme, chaque neurone reçoit les informations renvoyées par tous les neurones à travers les connexions des poids, son but est de permettre à la sortie de n'importe quel neurone d'être contrôlée par la sortie de tous les neurones, de sorte que la sortie de chaque neurone peut se restreindre. Chaque neurone a un seuil I_i pour refléter le contrôle du bruit d'entrée.

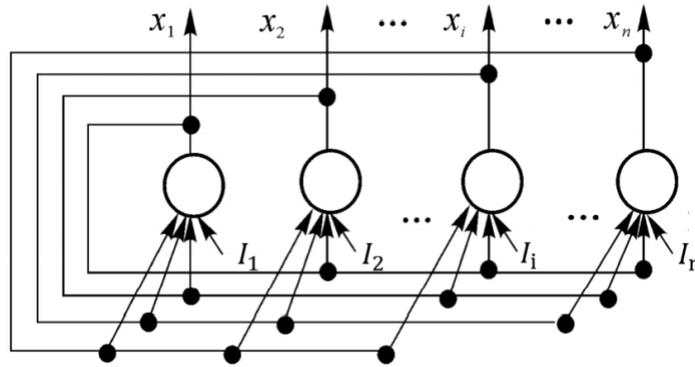


FIGURE 2.9 – Structure du réseau de Hopfield discret

États du réseau :

La sortie de chaque neurone du réseau est obtenue par une fonction de transfert, cette sortie est appelée état, notée par x_i , et l'ensemble de tous les états neuronaux constitue l'état défini par

$$X(t) = (x_1(t), x_2(t), \dots, x_n(t)).$$

L'état initial du réseau récurrent de Hopfield est exprimé par $X(0) = (x_1(0), x_2(0), \dots, x_n(0))$.

Le processus dynamique de l'évolution de neurones permet à mettre à jour l'état de chaque neurone par la formule suivante :

$$x_i = f(u_i) \quad i = 1, 2, \dots, n$$

Dans ce cas, $f(\cdot)$ est une fonction de transfert qui est souvent utilisée comme une fonction de seuil.

$$x_i = \text{sgn}(u_i) = \begin{cases} 1 & \text{si } u_i \geq 0 \\ -1 & \text{si } u_i < 0 \end{cases} \quad \forall i = 1, 2, \dots, n$$

L'entrée nette est :

$$u_i = \sum_{j=1}^n (w_{ij} \cdot x_j) - I_i, \quad \forall i = 1, 2, \dots, n$$

En ce qui concerne les poids, le réseau de Hopfield est caractérisé généralement par $w_{ii} = 0$ et $w_{ij} = w_{ji}$ (c-à-d, la matrice de poids de connexion W est une matrice symétrique de diagonale nulle). Dans l'état stable du réseau de Hopfield, la sortie de chaque neurone ne changera plus. A ce moment, on dit que la sortie du réseau est atteinte, et exprimée par :

$$\lim_{t \rightarrow \infty} X(t)$$

Modes de mise à jour :

La mise à jour de réseau de Hopfield est souvent exprimée par deux manières, soit en mode asynchrone, soit en mode synchrone.

— Mode asynchrone :

A chaque instant, un seul neurone est mis à jour et ajusté. Ce neurone peut être sélectionné au hasard ou suivi par un ordre préétabli. Le résultat de cet ajustement influence sur l'entrée du neurone suivant. A la mise à jour du $j^{\text{ème}}$ neurone, on définit l'état du réseau de Hopfield par :

$$x_i(t+1) = \begin{cases} \text{sgn}(u_i(t)) & \text{si } i = j \\ x_i(t) & \text{si } i \neq j \end{cases}$$

— Mode synchrone :

Dans le mode synchrone ou mode parallèle tous neurones mettent à jour leurs états en même temps.

$$x_i(t+1) = \text{sgn}(u_i(t)) \quad \forall i = 1, 2, \dots, n$$

2.7.2 Stabilité du réseau et attracteur

Le réseau de neurones de Hopfield est un réseau qui stocke un certain nombre d'états de stabilité prédéfinis. Lors de l'exécution, le réseau renvoie sa sortie comme entrée suivante lorsqu'elle agit sur le réseau en tant que mode d'entrée initial. Après plusieurs itérations, le réseau se stabilisera éventuellement à un état de stabilité prédéfini, à condition que la structure du réseau réponde à une certaine condition.

Soit $X(0)$ le vecteur d'activation initial du réseau, qui agit sur le réseau uniquement à l'instant initial $t = 0$. Après la mise à jour de $X(0)$, le réseau remplacera ce vecteur par l'entrée suivante $X(1)$ à l'instant $t = 1$. En tant que système dynamique non linéaire, le réseau de neurones de Hopfield a des caractéristiques dynamiques telles que la stabilité et l'instabilité de ses états.

Stabilité du réseau :

Selon l'analyse de l'état de fonctionnement du réseau, le réseau de Hopfield est essentiellement un système dynamique non linéaire [35]. À partir de l'état initial $X(0)$ si le réseau de Hopfield converge après un nombre fini d'itérations, alors son état ne change pas, c'est-à-dire $X(t+1) = X(t)$. Donc le réseau est dit stable. Dans le cas de convergence, le réseau de Hopfield peut converger de n'importe quel état initial vers un état stable comme indiqué dans la Figure (2.10)-(a). Dans le cas de l'instabilité de réseau de Hopfield, le réseau diverge. Cette divergence peut être interprétée par l'oscillation auto-entretenu de la limite, appelée réseau en anneau fini. La figure (2.10)-(b) donne son diagramme de phase. Si la trajectoire de l'état du réseau change dans une certaine plage, mais ni répète ni arrête, l'état change on un nombre infini de trajectoires, ce phénomène est appelé chaos, le diagramme de phase montré dans la figure (2.10)-(c) ci-dessous.

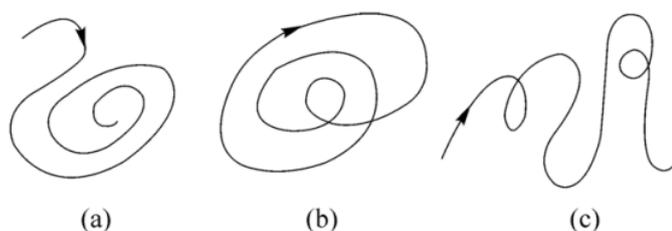


FIGURE 2.10 – Trois types de stabilité d'un réseau : (a) stabilité (b) oscillation (c) chaos

Attracteur et fonction d'énergie :

Lorsque le réseau atteint un état stable, là on parle de l'attracteur du réseau. Le comportement final d'un système dynamique est déterminé par son attracteur, et l'existence d'attracteurs constitue la base de la mémoire des informations stockées. Si nous considérons l'attracteur comme la solution du problème, alors l'évolution de l'état initial vers l'attracteur est le processus

de résolution de ce problème. La définition et le théorème de l'attracteur de réseau en mode asynchrone sont donnés ci-dessous.

Définition 2.3. *Si l'état X du réseau vérifie $X = f(WX - I)$, alors X est l'attracteur du réseau.*

Les attracteurs vers lesquels converge un tel réseau correspondent aux minima d'une fonction, appelée énergie du réseau, définie par :

$$\begin{aligned} E(t) &= -\frac{1}{2}X^T(t)WX(t) + X^T(t)I \\ &= -\frac{1}{2}\sum_{i=1}^n\sum_{j=1}^nw_{ij}x_ix_j + \sum_{i=1}^nI_ix_i \end{aligned} \quad (2.1)$$

où X est le vecteur des sorties des neurones du réseau, c'est-à-dire le vecteur d'état du système :

$$X(t) = [x_1(t), x_2(t), \dots, x_n(t)]^T$$

Cette fonction est une fonction de n variables qui comporte généralement un très grand nombre de minima locaux.

On établit naturellement un lien entre une telle énergie et la fonction d'énergie d'un problème d'optimisation combinatoire, décrite précédemment. On comprend dès lors aisément l'attrait des réseaux de neurones récurrents pour traiter des problèmes d'optimisation.

Théorème 2.1. *Pour un réseau en mode asynchrone, si l'état du réseau est réglé de manière asynchrone et que la matrice de poids de connexion W est une matrice symétrique, le réseau converge finalement vers un attracteur pour tout état initial[25].*

Dans ce qui suit, le théorème (2.1) est prouvé en analysant la fonction d'énergie. La variation de l'énergie du réseau est notée par ΔE et la variation d'état du réseau est notée par ΔX , alors :

$$\Delta E(t) = E(t+1) - E(t) \quad (2.2)$$

$$\Delta X(t) = X(t+1) - X(t) \quad (2.3)$$

En substituant (2.1) et (2.3) dans (2.2), la quantité de changement dans l'énergie du réseau peut être étendue à :

$$\begin{aligned} \Delta E(t) &= E(t+1) - E(t) \\ &= -\frac{1}{2}[X(t) + \Delta X(t)]^T W [X(t) + \Delta X(t)] + [X(t) + \Delta X(t)]^T I - \left[-\frac{1}{2}X^T(t)WX(t) + X^T(t)I\right] \\ &= -\Delta X^T(t)WX(t) - \frac{1}{2}\Delta X^T(t)W\Delta X(t) + \Delta X^T(t)I \\ &= -\Delta X^T(t)[WX(t) - I] - \frac{1}{2}\Delta X^T(t)W\Delta X(t) \end{aligned}$$

Le théorème (2.1) fournit une mise à jour de chaque neurone d'une manière asynchrone, il n'y a qu'un seul état d'un neurone est ajusté à l'instant t , soit le neurone i concerné par la mise à jour, et que $X(t) = (0, \dots, 0, \Delta x_i(t), 0, \dots, 0)^T$ et considérant que W est une matrice symétrique, alors :

$$\Delta E(t) = -\Delta x_i(t) \left[\sum_{j=1}^n (w_{ij} \cdot x_j - I_i) \right] - \frac{1}{2}\Delta x_i^2(t)w_{ii}$$

Pour $w_{ii} = 0$, et $u_i = \sum_{j=1}^n (w_{ij} \cdot x_j) - I_i$, la formule ci-dessus peut être simplifiée :

$$\Delta E(t) = -\Delta x_i(t) \cdot u_i(t) \quad (2.4)$$

Considérons toutes les situations qui peuvent survenir dans l'équation ci-dessus.

- Cas A : $x_i(t) = -1$ et $x_i(t+1) = 1$. À partir de l'équation (3.3), nous obtenons $\Delta x_i(t) = 2$ et $u_i(t) \geq 0$, et en substituant en (2.4), nous obtenons $\Delta E(t) \leq 0$
- Cas B : $x_i(t) = 1$ et $x_i(t+1) = -1$ donc $\Delta x_i(t) = -2$ et $u_i(t) \leq 0$ et en substituant en (2.4), nous obtenons $\Delta E(t) \leq 0$.
- Cas C : $x_i(t) = x_i(t+1)$ donc $\Delta x_i(t) = 0$ et en substituant en (2.4), nous obtenons $\Delta E(t) = 0$.

Tous les trois cas ci-dessus incluent les cas possibles de l'équation (2.9), à partir de laquelle nous pouvons voir que dans tous les cas $\Delta E(t) \leq 0$. En d'autre terme, lors de l'évolution dynamique du réseau, l'énergie est toujours décroissante ou reste la même.

L'analyse de la convergence de $E(t)$ vers une constante correspond à l'état stationnaire du réseau. Lorsque $E(t)$ converge vers une constante $\Delta E(t) = 0$. Ce qui correspond aux deux cas suivants :

- Cas A : $x_i(t) = x_i(t+1) = 1$ ou $x_i(t) = x_i(t+1) = -1$. Dans ce cas, l'état du neurone i ne change plus son état, indiquant que le réseau est entré en régime permanent, l'état du réseau correspondant est l'attracteur du réseau.
- Cas B : $x_i(t) = -1$, $x_i(t+1) = 1$ et $u_i(t) = 0$. Dans ce cas, le réseau continuera à évoluer, ce qui contredit le fait que $E(t)$ converge vers une constante.

Théorème 2.2. *Pour le réseau en mode synchrone, si l'état est réglé de manière synchrone et que la matrice de poids de connexion W est une matrice symétrique définie non négative, le réseau converge finalement vers un attracteur pour tout état initial[32].*

La comparaison entre le théorème (2.1) et théorème (2.2), nous révèle que lorsque le réseau fonctionne en mode synchrone, la matrice de poids W est plus exigeante, Si W ne peut pas répondre à l'exigence de matrice symétrique définie non négative, le réseau apparaîtra comme une oscillation auto-entretenu. Les méthodes asynchrones ont une meilleure stabilité que les méthodes synchrones mais avec l'inconvénient est qu'elles perdent l'avantage de traitement parallèle.

L'analyse de ci-dessus montre que dans le réseau de l'état initial au processus d'évolution stationnaire, l'énergie du réseau a toujours été de réduire la direction de l'évolution, lorsque l'énergie s'est finalement stabilisée, la constante correspond à l'état minimum de l'énergie du réseau. Le plus petit état est le puits énergétique du réseau, et l'énergie correspond bien à l'attracteur du réseau.

Exemple 2.1. *Pour un réseau en mode asynchrone à 3 neurones, qui est représenté par un graphe non orienté comme le montre la figure (3.11), Les poids et les seuils sont indiqués sur la même figure pour calculer l'état de l'évolution du réseau.*

Le réseau de la figure (2.11) doit avoir 8 type d'état de $x = (x_1, x_2, x_3)$. Nous définissons l'état initial du réseau récurrent de Hopfield par $x^1 = (0, 0, 0)$ en suivant l'ordre de mise à jour de $1 \rightarrow 2 \rightarrow 3$

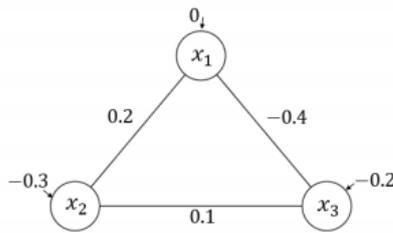


FIGURE 2.11 – Réseau de trois neurones

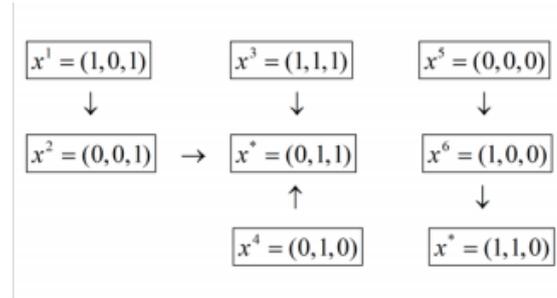


FIGURE 2.12 – Diagramme d'évolution de réseau de trois neurones

- Étape 1 : Mise à jour de x_1 , $x_1 = \text{sgn}(0 \times 0.2 + 0 \times (-0.4) - 0) = \text{sgn}(0) = 1$ et l'état des autres neurones ne change pas. Alors l'état du réseau passe de $(0, 0, 0)$ à $(1, 0, 0)$.
- Étape 2 : Maintenant, à base de l'état du réseau précédent défini par $(1, 0, 0)$, nous pouvons mettre à jour l'état de neurone x_2 par la relation $x_2 = \text{sgn}(1 \times 0.2 + 0 \times 0.1 + 0.3) = \text{sgn}(0.5) = 1$ et les autres neurones restent inchangés. Donc l'état du réseau devient $(1, 1, 0)$.
- Étape 3 : La mise à jour de neurone x_3 est obtenu par $x_3 = \text{sgn}(1 \times (-0.4) + 1 \times 0.1 + 0.2) = \text{sgn}(-0.1) = 0$, à ce stade, l'état du réseau est $(1, 1, 0)$.

La même procédure peut être faite pour l'évolution des autres états. La figure (2.12) représente les huit états possibles d'évolution. Sur la même figure les $x^* = (1, 1, 0)$ et $x^* = (0, 1, 1)$ sont les attracteurs du réseau. À partir de n'importe quel état, le réseau atteindra un de ces deux états stables après plusieurs mises à jour.

2.7.3 Réseau de Hopfield continu

En 1984, Hopfield [25] a développé un nouveau modèle de Hopfield continu. La structure de base de réseau de neurones continu est similaire à celle de réseau de neurones discret sauf que tous les neurones fonctionnent de manière synchrone et la sortie de chaque neurone est un analogue qui varie continuellement avec le temps, ce qui rend le réseau continu meilleur que le réseau discret en terme de parallélisme et de performances en temps réel. Plus proche du mécanisme de travail du réseau de neurones biologiques. Le réseau de Hopfield continu peut être décrit par une équation différentielle à coefficients constants (la dynamique du réseau à temps continu).

La fonction d'énergie est définie par :

$$E(\mathbf{x}) = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N w_{ij} x_i x_j - \sum_{i=1}^N I_i x_i + \frac{\alpha}{\gamma} \sum_{i=1}^N \int_0^{x_i} g^{-1}(x) dx \quad (2.5)$$

où α est un réel positif et g est la fonction d'activation des neurones ; En général, le dernier terme de cette énergie est négligeable comparé aux précédents dès lors que la pente γ à l'origine des neurones est grande, ou encore que α est petit.

L'équation d'évolution d'un neurone continu i est donnée par l'équation différentielle suivante :

$$\frac{du_i}{dt} = \mu_i \left[-\alpha_i u_i - \frac{\partial E(\mathbf{x})}{\partial x_i} \right]$$

et la sortie est

$$x_i = \tanh(\gamma u_i) = \tanh\left(\frac{u_i}{T}\right).$$

Dans ces équations, $\mu_i = 1/\tau_i$ est un réel positif paramétrant la vitesse de convergence, α_i est un réel positif, T est la température (inverse de la pente à l'origine de la fonction d'activation du neurone) et $E(\mathbf{x})$ est la fonction d'énergie du problème, qui n'est pas nécessairement quadratique.

La dérivée de la fonction énergie E par rapport au temps peut s'écrire, d'après les équations de mouvement des neurones que l'on vient de présenter :

$$\begin{aligned} \frac{dE}{dt} &= \sum_{i=1}^N \frac{\partial E}{\partial x_i} \frac{dx_i}{dt} \\ &= -\sum_{i=1}^N \tau_i \frac{dx_i}{dt} \frac{du_i}{dt} - \sum_{i=1}^N \alpha_i u_i \frac{dx_i}{dt} \\ &= -\sum_{i=1}^N \tau_i \frac{dx_i}{du_i} \left(\frac{du_i}{dt} \right)^2 - \sum_{i=1}^N \alpha_i u_i \frac{dx_i}{dt} \end{aligned}$$

On voit clairement que le premier terme de cette équation est toujours négatif ou nul car $\tau_i > 0$ et $dx_i/du_i > 0$. En revanche, le second terme peut être positif, négatif ou nul ; en conséquence, la dérivée temporelle de la fonction d'énergie peut être positive [47]. Pour éviter cela, on utilise fréquemment une version différente des équations de mouvement, qui est la suivante :

$$\frac{du_i}{dt} = -\mu_i \frac{\partial E(\mathbf{x})}{\partial x_i} \quad i = 1, \dots, N$$

Du point de vue des sorties des neurones, on obtient, à partir de ces équations de mouvement, les équations suivantes :

$$\begin{aligned} \frac{dx_i}{dt} &= \frac{dx_i}{du_i} \frac{du_i}{dt} \\ &= \frac{1}{T} (1 - x_i^2) \frac{du_i}{dt} \\ &= -\frac{\mu_i}{T} (1 - x_i^2) \frac{\partial E(\mathbf{x})}{\partial x_i} \quad i = 1, \dots, N \end{aligned}$$

Les propriétés dynamiques d'un réseau de Hopfield analogique sont donc régies par un système d'équations différentielles non linéaires.

Avec cette modification, on peut montrer facilement que tout changement d'état (de sortie) d'un neurone se traduit sur l'énergie du réseau par une diminution ou un maintien à sa valeur. En effet :

$$\begin{aligned}
\frac{dE}{dt} &= \sum_{i=1}^N \frac{\partial E}{\partial x_i} \frac{dx_i}{dt} \\
&= \sum_{i=1}^N \left[-\tau_i \frac{du_i}{dt} \right] \frac{dx_i}{dt} \\
&= - \sum_{i=1}^N \tau_i \frac{dx_i}{du_i} \left(\frac{du_i}{dt} \right)^2
\end{aligned}$$

Donc,

$$\frac{dE}{dt} \leq 0$$

En d'autres termes, le système d'équations d'évolution contraint la fonction d'énergie E à décroître d'une manière monotone vers un minimum local. Plus précisément, ce système a la propriété suivante : à partir d'un point initial x à l'intérieur de l'hypercube $\{-1, 1\}^N$, le système dynamique converge vers un minimum local de la fonction d'énergie $E(\mathbf{x})$, situé soit sur un sommet, soit sur la surface de l'hypercube. Quand l'attracteur est sur la surface, on peut choisir comme solution tout sommet de la surface car l'énergie est constante sur cette surface. Ainsi, chaque solution vers laquelle converge le réseau de neurones est localement optimale.

En pratique, avec une réalisation électronique d'un tel réseau, les temps de convergence sont de l'ordre de quelques nanosecondes ou microsecondes. Cela autorise en général plusieurs résolutions du problème en partant de différents points initiaux. Cette stratégie peut parfois se montrer efficace, mais en général les minima attracteurs ne sont pas de qualité suffisante.

Pour traiter des problèmes d'optimisation combinatoire avec plus d'efficacité, on a souvent intérêt à utiliser des neurones continus avec une pente variable. En effet, en évoluant à l'intérieur de l'hypercube des solutions, et non uniquement sur les sommets comme le ferait un réseau de neurones binaires, les minima locaux sous-optimaux de haute énergie sont mieux évités au cours de la convergence. En outre, on peut montrer qu'un réseau continu est bien plus rapide et fiable qu'un réseau de neurones binaires avec une mise à jour asynchrone des neurones [49].

Minimiser la fonction $E(\mathbf{x})$ avec \mathbf{x} continu réduit considérablement la probabilité d'être piégé dans un minimum local, car les vallées de l'espace des solutions sont plus larges que dans le cas binaire. Hélas, il est rare que les neurones analogiques, en évoluant à l'intérieur de l'hypercube, convergent finalement vers un sommet de ce dernier. Pour forcer les neurones analogiques à prendre finalement des valeurs binaires, codant une solution du problème, il existe plusieurs solutions. L'une d'entre elles consiste à faire croître les pentes à l'origine des fonctions de transfert des neurones analogiques au cours du processus d'optimisation. Une autre solution, qu'il est possible de mettre en œuvre conjointement à la solution précédente, est d'ajouter à la fonction d'énergie un terme de pénalité de la forme :

$$E_x = \sum_{j=1}^N (1 - x_j^2)$$

ce terme est nul si les sorties des neurones sont discrètes à valeurs dans $\{-1, 1\}$.

En général, le modèle continu est supérieur au modèle discret en termes du problème minimum local, en raison de sa surface lisse de l'énergie. De plus, le réseau continu avec un gain variable permet d'éviter les mauvais optimums locaux. Il est plus rapide et réel en comparaison avec un réseau discret. Par conséquent, le modèle continu a dominé les techniques de résolution de problèmes d'optimisation, en particulier pour les problèmes combinatoires.

2.8 Principes de fonctionnement du réseau de Hopfield pour l'optimisation

Pour l'optimisation, on utilise le réseau de la manière suivante : à partir d'un état initial, on laisse le réseau évoluer librement jusqu'à un attracteur, qui est généralement, pour les problèmes d'optimisation, un état indépendant du temps (un « point fixe » de la dynamique). On dit alors que le réseau a convergé : la convergence est atteinte lorsque les sorties des neurones n'évoluent plus. Pour plus de détails sur les propriétés de convergence, on se référera aux travaux de Goles. La dynamique du réseau est généralement asynchrone : entre deux instants de temps, un seul neurone, choisi aléatoirement, est mis à jour ; autrement dit, son potentiel est calculé, et sa sortie réévaluée en conséquence.

Quand ces réseaux sont utilisés pour résoudre des problèmes d'optimisation, les poids des connexions sont déterminés analytiquement à partir de la formulation du problème ; en général, cela est fait directement à partir de la fonction énergie associée au problème, comme nous le verrons dans les exemples que nous présenterons. De plus, les sorties des neurones, dans l'attracteur vers lequel converge le réseau, codent une solution au problème d'optimisation.

2.9 Application des réseaux de Hopfield à l'optimisation

Ce sont des propriétés de convergence des réseaux de Hopfield, décrites ci-avant, qu'a découlé l'application de ce type de réseaux à la résolution de problèmes d'optimisation. Celle-ci comprend quatre étapes :

1. Déterminer un codage du problème. Il s'agit, dans un premier temps, de trouver une représentation du problème d'optimisation telle qu'une solution, c'est-à-dire une instantiation des variables du problème, soit représentée par l'état des sorties des neurones à la convergence.
2. Exprimer la fonction de cout et les contraintes sous la forme d'une énergie d'un réseau de Hopfield. On cherchera à exprimer la fonction de cout comme une fonction quadratique. Quant aux contraintes à satisfaire, elles sont de deux types : celles qui sont définies par le problème d'optimisation, et celles qui résultent du codage du problème choisi à l'étape précédente. Parfois, il n'est pas possible de trouver une énergie quadratique. Il faut alors essayer d'exprimer le problème comme celui de la minimisation d'une fonction F , dérivable par rapport aux sorties des neurones.
3. Déterminer les équations d'évolution de chaque neurone.
4. Démarrer l'exécution du réseau de neurones. Généralement, quand on ne dispose d'aucune connaissance a priori sur la localisation de la solution recherchée, les états des neurones sont initialisés aléatoirement. La dynamique du réseau de neurones les fera converger vers un minimum local de l'énergie.

Ce type de méthodologie a été appliqué pour résoudre une grande variété de problèmes combinatoires, qui se ramènent en général à des problèmes de théorie des graphes.

2.10 Limitations des réseaux de Hopfield

L'utilisation des réseaux de Hopfield pour résoudre des problèmes d'optimisation pose un certain nombre de problèmes.

La principale difficulté, pour l'optimisation d'une fonction d'énergie, vient du fait que la dynamique du réseau conduit fréquemment vers un minimum local de l'énergie, qui n'est pas nécessairement proche de l'optimum, ou bien qui ne correspond pas à une solution acceptable. Et ce, parce que les contraintes sont combinées à la fonction de coût dans l'énergie du réseau.

De plus, le réseau de Hopfield ne fait aucune distinction entre les contraintes strictes et les contraintes de préférence, autrement que par des pondérations différentes dans la fonction d'énergie.

Enfin, les valeurs des paramètres qui pondèrent les divers termes de l'énergie du réseau influent sur le nombre d'itérations nécessaires pour que le réseau converge. Ces difficultés ont été à l'origine de nombreux travaux qui ont cherché à en limiter les effets.

2.11 Conclusion

Nous avons vu qu'un réseau de neurone de Hopfield est un système dynamique et que son évolution peut être agencée de plusieurs façons. Le réseau reçoit des entrées et propage son activation pendant un nombre fixe de cycle d'activation. L'activation initiale des neurones du réseau est fixée par l'utilisateur. On laisse ensuite le réseau évoluer jusqu'à un état stable. Cet état est considéré comme la réponse du réseau.

Deux principales contraintes du modèle de Hopfield sont ses liaisons symétriques et la connectivité totale.

Les réseaux de Hopfield ont prouvé leur efficacité dans la résolution de problèmes d'optimisation. Toutes ces applications nécessitent un codage du problème sous la forme d'un réseau de neurones artificiels décrit au moyen d'une forme quadratique, appelée énergie du problème, et dont le minimum correspond à une solution optimale du problème. Une fois cette énergie définie, il est nécessaire de choisir la méthodologie neuronale à mettre en œuvre pour sa minimisation.

Dans le reste de notre travail, nous présentons un exemple d'application de réseau de Hopfield qui est le problème du voyageur de commerce .

Chapitre 3

Application : Le problème du voyageur de commerce

3.1 Introduction

Le problème du voyageur de commerce, étudié depuis le 19-ème siècle, est l'un des plus connus dans le domaine de la recherche opérationnelle. Jouez à trouver le meilleur parcours possible... et découvrez différentes méthodes informatiques proposées pour résoudre ce problème.

C'est déjà sous forme de jeu que William Rowan Hamilton a posé pour la première fois ce problème, dès 1859. Sous sa forme la plus classique, son énoncé est le suivant : « Un voyageur de commerce doit visiter une et une seule fois un nombre fini de villes et revenir à son point d'origine. Trouvez l'ordre de visite des villes qui minimise la distance totale parcourue par le voyageur ». Ce problème d'optimisation combinatoire appartient à la classe des problèmes NP-Complets.

Les domaines d'application sont nombreux : problèmes de logistique, de transport aussi bien de marchandises que de personnes, et plus largement toutes sortes de problèmes d'ordonancement de plus certains problèmes rencontrés dans l'industrie se modélisent sous la forme d'un problème du voyageur de commerce, comme l'optimisation de trajectoires des machines etc.

Pour illustrer les propos du chapitre 2 à l'aide d'un exemple concret, nous allons détailler la résolution d'un problème du voyageur de commerce, proposée par Hopfield et Tank en 1985 [26].

La première étape consiste à reformuler le problème d'un point de vue mathématique, afin qu'il soit possible de le résoudre par un réseau de neurones.

3.2 Codage du problème

Variables de décision :

Étant donné N villes, soit

$$v_{x,i} = \begin{cases} 1 & \text{si la ville } x \text{ est visitée à l'ordre } i \\ 0 & \text{sinon} \end{cases} \quad \forall x, i \in \{1, 2, \dots, N\}$$

Donc une solution au problème, c'est-à-dire une tournée entre les différentes villes, peut être codée comme une matrice de permutation. Autrement dit, une tournée est représentée par une matrice $N \times N$ de N^2 éléments. Dans cette matrice, une ligne représente une ville, et une colonne un numéro de passage au cours d'une tournée. Une tournée acceptable est représentée par une matrice de permutation, ayant donc exactement N « 1 » et $N^2 - N$ « 0 ».

Le tableau 3.1 présente un exemple de matrice qui code une solution acceptable d'un problème à 5 villes :

Remarque 3.1. Une matrice de permutation est une matrice carrée qui vérifie les propriétés suivantes :

- les coefficients sont 0 ou 1 ;
- il y a un et un seul 1 par ligne ;
- il y a un et un seul 1 par colonne.

	1	2	3	4	5
Ville A	0	0	1	0	0
Ville B	1	0	0	0	0
Ville C	0	1	0	0	0
Ville D	0	0	0	1	0
Ville E	0	0	0	0	1

FIGURE 3.1 – Solution acceptable du problème du voyageur de commerce pour cinq villes

La solution ainsi codée correspond à une tournée où les villes A, B, C, D et E sont visitées dans l'ordre suivant : B-C-A-D-E-B.

À chaque coefficient de la matrice est associé un neurone. Ce codage nécessite donc $n = N^2$ neurones et N^4 connexions entre les neurones. On notera $v_{x,i}$ la sortie du neurone (x, i) correspondant à la x -ème ligne et à la i -ème colonne, et $u_{x,i}$ son potentiel. Par ailleurs, on connaît a priori les distances deux à deux entre les villes : d_{xy} représente ainsi la distance entre la ville x et la ville y .

Fonction de coût :

La fonction de coût à minimiser s'exprime comme une mesure de la longueur du trajet parcouru lors de la tournée. Elle peut s'exprimer en fonction des sorties des N neurones qui valent 1 dans la matrice de permutation, de la manière suivante :

$$F_C = \frac{1}{2} \sum_{x=1}^N \sum_{i=0}^N \sum_{\substack{y=1 \\ y \neq x}}^N d_{x,y} v_{x,i} (v_{y,i-1} + v_{y,i+1})$$

où, par définition, $v_{y,0} = v_{y,N}$ et $v_{y,N+1} = v_{y,1}$.

Le terme dans les sommes multiples est non nul quand la ville x est en position i , et ce terme est égal à la distance du trajet entre cette ville et celle qui la suit. La sortie du neurone en position (x, i) est ainsi multipliée par les sorties des neurones de la colonne $i + 1$. Ainsi, chaque neurone est connecté à $2N$ neurones dans le réseau. Le codage de cette fonction de coût ne nécessite donc pas N^4 connexions, mais N^3 .

Contraintes :

Pour garantir que, à la convergence, la matrice des sorties des neurones est bien une matrice de permutation, ce qui assure la validité de la solution, des contraintes supplémentaires sont définies. On associe à leur éventuelle violation une fonction de pénalité.

Pour que le tour soit acceptable, chaque ville doit être visitée exactement une fois. Cela impose que, dans chaque ligne, il n'existe pas deux neurones dont la sortie est à 1. Pour cela, on définit la fonction énergie suivante, qui est non nulle si au moins deux neurones ont une sortie à 1 dans une ligne :

$$F_1 = \frac{1}{2} \sum_{x=1}^N \sum_{i=0}^N \sum_{\substack{j=1 \\ j \neq i}}^N v_{x,i} v_{x,j}$$

De même, à chaque étape, le voyageur de commerce ne peut pas être dans plus d'une ville. En conséquence, dans chaque colonne, il n'existe pas deux neurones dont la sortie vaut 1 à la convergence du réseau. On définit donc la fonction de pénalisation suivante, qui est non nulle si au moins deux neurones ont une sortie à 1 dans une colonne :

$$F_2 = \frac{1}{2} \sum_{x=1}^N \sum_{i=0}^N \sum_{\substack{y=1 \\ y \neq x}}^N v_{x,i} v_{y,i}$$

Les deux contraintes définies ci-avant ne sont pas encore suffisantes pour garantir la validité d'une solution. En effet, sans nouvelle contrainte, le processus de minimisation ferait naturellement converger toutes les sorties des neurones vers 0, ce qui n'aurait aucun sens. La validité d'une solution lui impose qu'il y ait exactement N neurones dont la sortie soit égale à 1.

Une troisième fonction de contraintes est donc définie :

$$F_3 = \frac{1}{2} \left[\sum_{x=1}^N \sum_{i=0}^N v_{x,i} - N \right]^2$$

Alors, on obtient le modèle suivant :

$$\text{Min } F_C = \sum_{x=1}^N \sum_{i=0}^N \sum_{\substack{y=1 \\ y \neq x}}^N d_{x,y} v_{x,i} (v_{y,i-1} + v_{y,i+1})$$

s/c :

$$\sum_{x=1}^N \sum_{i=0}^N \sum_{\substack{j=1 \\ j \neq i}}^N v_{x,i} v_{x,j} = 0$$

$$\sum_{x=1}^N \sum_{i=0}^N \sum_{\substack{y=1 \\ y \neq x}}^N v_{x,i} v_{y,i} = 0$$

$$\left[\sum_{x=1}^N \sum_{i=0}^N v_{x,i} - N \right]^2 = 0$$

$$v_{x,i} \in \{0, 1\} \quad \forall x, i \in \{1, 2, \dots, N\}$$

La dynamique du réseau de Hopfield est décrite par l'équation différentielle :

$$\frac{d\mathbf{u}}{dt} = -\frac{\mathbf{u}}{\tau} + \mathbf{T}\mathbf{v} + \mathbf{I} \quad (3.1)$$

et la fonction de sortie $v_{x,i} = g(u_{x,i})$ est une tangente hyperbolique définie par :

$$g(u_{x,i}) = \frac{1}{2} \left(1 + \tanh \left(\frac{u_{x,i}}{u_0} \right) \right) \quad u_0 > 0 \quad (3.2)$$

où la matrice T représente les poids des connexions synaptiques entre neurones.

La résolution d'un problème combinatoire via CHN nécessite la construction d'une fonction énergétique associée au problème.

3.3 La fonction d'énergie

L'énergie totale du réseau de Hopfield est la somme pondérée des fonctions définies ci-avant :

$$\begin{aligned} E(V) = & \frac{A}{2} \sum_x \sum_i \sum_{j \neq i} v_{x,i} v_{x,j} \\ & + \frac{B}{2} \sum_i \sum_x \sum_{y \neq x} v_{x,i} v_{y,i} \\ & + \frac{C}{2} \left(\sum_x \sum_i v_{x,i} - N \right)^2 \\ & + \frac{D}{2} \sum_x \sum_{y \neq x} \sum_i d_{x,y} v_{x,i} (v_{y,i+1} + v_{y,i-1}) \end{aligned}$$

Les constantes A, B, C et D doivent être ajustées en fonction des poids relatifs des différentes contraintes.

On peut mettre la fonction d'énergie du problème sous la forme d'une fonction quadratique du type suivant, qui est l'énergie d'un réseau de Hopfield :

$$E(V) = -\frac{1}{2} \sum_x \sum_i \sum_y \sum_j v_{x,i} T_{xi,yj} v_{y,j} - \sum_x \sum_i I_{x,i} v_{x,i} \quad (3.3)$$

Dans cette équation, on détermine les poids $T_{xi,yj}$ à partir de la forme analytique du coût et des contraintes :

$$T_{xi,yj} = - (A \delta_{x,y} (1 - \delta_{i,j}) + B (1 - \delta_{xy}) \delta_{i,j} + C + D (\delta_{i,j-1} + \delta_{i,j+1}) d_{x,y}) \quad (3.4)$$

$$I_{x,i} = CN \quad (3.5)$$

où δ est l'opérateur de Kronecker :

$$\delta_{n,m} = \begin{cases} 1 & \text{si } n = m \\ 0 & \text{sinon} \end{cases} \quad \forall n, m$$

3.4 Réglage des paramètres

Dans ce qui suit, soient

$$H = \{V \in [0, 1]^{N \times N}\} \quad \text{et} \quad H_C = \{V \in \{0, 1\}^{N \times N}\}$$

sont, respectivement, l'hypercube de \mathbb{R}^n , où $n = N \times N$ et son ensemble de sommets. Étant donné que $V \in H$ fixe, soient

$$S_i = \sum_x v_{x,i} \quad S_x = \sum_i v_{x,i}$$

$$S = \sum_x \sum_i v_{x,i} = \sum_x S_x = \sum_i S_i$$

sont la somme des lignes, des colonnes et des lignes et des colonnes. L'ensemble des tournées pour le TSP est :

$$H_T = \{V \in H_C / S_x = 1, S_i = 1 \quad \forall x, i \in \{1, \dots, N\}\} \quad (3.6)$$

La convergence du réseau de neurones vers un point $V \in [0, 1]^n$ est liée à la dérivée partielle de sa fonction énergétique $E(V)$ par rapport à $v_{x,i}$ pour tous $x, i \in \{1, \dots, N\}$, qui sera notée comme :

$$E_{x,i}(V) = \frac{\partial E(V)}{\partial v_{x,i}}$$

En tenant compte de l'équation (3.3), $E_{x,i}(V)$ peuvent être calculés comme :

$$E_{x,i}(V) = A(S_x - v_{x,i}) + B(S_i - v_{x,i}) + C(S - N) + D \sum_{y \neq x} d_{x,y} (v_{y,i-1} + v_{y,i+1}) \quad (3.7)$$

Un point $V \in H$, sera un point d'équilibre pour le CHN si et seulement si, les trois relations suivantes sont vérifiées [34] :

$$E_{x,i}(V) \geq 0 \quad \forall (x, i) \in \{1, \dots, N\}^2 / v_{x,i} = 0$$

$$E_{x,i}(V) \leq 0 \quad \forall (x, i) \in \{1, \dots, N\}^2 / v_{x,i} = 1$$

$$E_{x,i}(V) = 0 \quad \forall (x, i) \in \{1, \dots, N\}^2 / v_{x,i} \in (0, 1)$$

Nous introduisons les notations suivantes :

$$\underline{E}^0(V) = \min_{(x,i)/v_{x,i}=0} E_{x,i}(V), \quad \bar{E}^1(V) = \max_{(x,i)/v_{x,i}=1} E_{x,i}(V)$$

On peut conclure que tout point $V \in H$ sera un point d'équilibre pour CHN si et seulement si les trois relations suivantes :

$$\underline{E}^0(V) \geq 0 \quad (3.8)$$

$$\bar{E}^1(V) \leq 0 \quad (3.9)$$

$$E_{x,i}(V) = 0 \quad \forall (x, i) \in \{1, \dots, N\}^2 / v_{x,i} \in (0, 1) \quad (3.10)$$

sont vérifiés.

La convergence des solutions admissibles :

Tout point $V \in H_T$ sera un point d'équilibre si et seulement si les relations (3.8) et (3.9) sont vérifiés.

Deux cas exhaustifs et exclusifs doivent être considérés :

Cas 1 : Soit $v_{x,i} = 1$ un élément de V , en tenant compte des équations (3.3) et (3.6), alors il existe $y, z \neq x$ tel que $v_{y,i-1} = v_{z,i+1} = 1$ et, par conséquent

$$\begin{aligned} E_{x,i}(V) &= A(S_x - 1) + B(S_i - 1) + C(S - N) + D(d_{y,x} + d_{x,z}) \\ &= A(1 - 1) + B(1 - 1) + C(N - N) + D(d_{y,x} + d_{x,z}) \\ &= D(d_{x,y} + d_{x,z}) > 0 \end{aligned}$$

De cette façon, toute tournée $V \in H_T$ ne pouvait pas être un point d'équilibre pour le CHN.

Ce problème peut être évité en introduisant un nouveau paramètre positif N' modifiant chaque seuil du réseau. En effet, l'équation (3.5) sera remplacé par :

$$I_{x,i} = CN' \quad \forall x, i \quad (3.11)$$

Pour une valeur $N' > N$ suffisamment grande, l'inégalité

$$E_{x,i}(V) = C(N - N') + D(d_{x,y} + d_{x,z}) \leq 0 \quad \forall x, y, z \quad \forall i \quad (3.12)$$

peut être toujours assuré et l'équation (4.9) est vérifiée.

Remarque 3.2. Désormais, avec le seuil modifié l'équation (3.11), la dérivée partielle sera calculée comme suit :

$$E_{x,i}(V) = A(S_x - v_{x,i}) + B(S_i - v_{x,i}) + C(S - N') + D \sum_{y \neq x} d_{x,y} (v_{y,i-1} + v_{y,i+1}) \quad (3.13)$$

Cas 2 : Soit $v_{x,i} = 0$ un élément de tour $V \in H_T$, Alors, par l'équation (3.12)

$$E_{x,i}(V) = A + B + C(N - N') + D \sum_{y \neq x} (d_{x,y} v_{y,i-1} + d_{x,y} v_{y,i+1})$$

Choisir une paire (x, i) de telle sorte que $v_{x,i-1} = 1$, donc $v_{x,i} = 0$ et $v_{y,i-1} = 0 \quad \forall y \neq x$:

$$A + B + C(N - N') + Dd_{x,z} \geq 0, \quad \forall z \neq x \quad (3.14)$$

En conséquence, l'équation (3.8) est vérifiée.

Soit d_U et d_L sont respectivement les bornes supérieure et inférieure pour les distances $d_{x,y}$:

$$d_L \leq d_{x,y} \leq d_U \quad \forall x, y \in \{1, \dots, N\} \quad (3.15)$$

Ensuite, basé sur les équations (3.12) et (3.14), le théorème suivant a été prouvé.

Théorème 3.1. [34]. *Si les coefficients de la fonction d'énergie de CHN vérifient :*

$$A + B + C(N - N') + Dd_L \geq 0 \quad (3.16)$$

$$C(N - N') + 2Dd_U \leq 0 \quad (3.17)$$

Alors, tout tour $V \in H_T$ sera un point d'équilibre pour CHN.

L'équation (3.17) souligne l'importance du paramètre, introduit par Hopfield.

Exemple 3.1. *Considérons le TSP défini par six villes, la matrice de distance est*

$$(d_{x,y}) = \begin{pmatrix} 0 & 2 & 2\sqrt{3} & 4 & 2\sqrt{3} & 2 \\ 2 & 0 & 2 & 2\sqrt{3} & 4 & 2\sqrt{3} \\ 2\sqrt{3} & 2 & 0 & 2 & 2\sqrt{3} & 4 \\ 4 & 2\sqrt{3} & 2 & 0 & 2 & 2\sqrt{3} \\ 2\sqrt{3} & 4 & 2\sqrt{3} & 2 & 0 & 2 \\ 2 & 2\sqrt{3} & 4 & 2\sqrt{3} & 2 & 0 \end{pmatrix}$$

À partir de la matrice de distance, les bornes suivantes peuvent être obtenues :

$$d_L = 2 \quad d_U = 4$$

C'est facile pour vérifier que les paramètres

$$N' = 8.1 \quad A = 1.7 \quad B = 2.2 \quad C = 1 \quad D = 0.25$$

vérifient les équations. (3.16) et (3.17) :

$$8.1 = N' > N = 6 \text{ et } 2Dd_U = 2 \leq C(N' - N) = 2.1 \leq A + B + Dd_L = 4.4$$

Non convergence des solutions non admissibles : Cas $V \in H_C - H_T$:

D'autre part, Toute tournée non admissible $V \in H - H_T$ ne peut pas être un point d'équilibre pour le CHN.

Deux cas doivent être distingués pour toute tournée non admissible $V \in H - H_T$: au ensemble de sommets de H , $V \in H_C$, ou hors de cet ensemble :

$$H - H_T = (H_C - H_T) \cup (H - H_C)$$

Ces sous-ensembles de H sont représentés sur la figure 3.2.

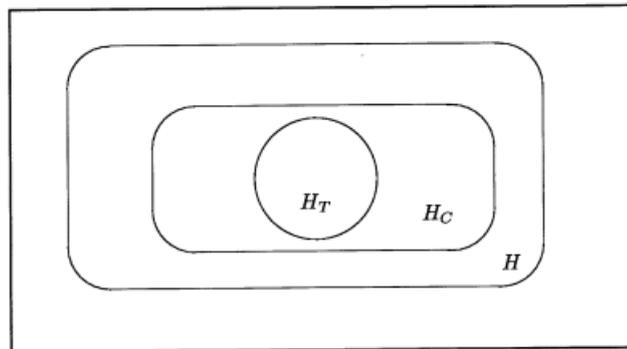


FIGURE 3.2 – Les sous-ensembles de H .

Le cas $V \in H_C - H_T$:

En tenant compte des équations (3.8) et (3.9), tout point $V \in H_C$ vérifié :

$$\underline{E}^0(V) < 0 \tag{3.18}$$

ou

$$\bar{E}^1(V) > 0 \quad (3.19)$$

ne peut pas être un point d'équilibre. Afin d'obtenir certaines de ces conditions suffisantes, une classe de solutions adjacentes d'une solution invalide $V \in H_C - H_T$ sera déterminée. Ces solutions sont déduites en substituant zéro à un ou un à zéro.

Définition 3.1. *Étant donné*

$$V = \begin{pmatrix} v_{1,1} & \cdots & v_{1,i} & \cdots & v_{1,N} \\ \vdots & \vdots & \vdots & & \\ v_{x,1} & \cdots & v_{x,i} & \cdots & v_{x,N} \\ \vdots & \vdots & \vdots & & \\ v_{N,1} & \cdots & v_{N,i} & \cdots & v_{N,N} \end{pmatrix} \in H$$

un point adjacent $A_{x,i}(V)$ peut être défini pour toute paire $(x, i) \in \{1, \dots, N\}^2$:

$$A_{x,i}(V) \equiv \begin{pmatrix} v_{1,1} & \cdots & v_{1,i} & \cdots & v_{1,N} \\ \vdots & \vdots & \vdots & & \\ v_{x,1} & \cdots & 1 - v_{x,i} & \cdots & v_{x,N} \\ \vdots & \vdots & \vdots & & \\ v_{N,1} & \cdots & v_{N,i} & \cdots & v_{N,N} \end{pmatrix} \in H$$

Évidemment, pour toute paire $(x, i) \in \{1, \dots, N\}^2$, il est vérifié que :

$$V \in H_C \Leftrightarrow A_{x,i}(V) \in H_C$$

Étant donné $V \in H_C$, deux sous-ensembles de H_C peut être défini en changeant un seul élément $v_{x,i}$ par son complément $1 - v_{x,i}$:

$$A^+(V) = \{A_{x,i}(V) \in H_C / v_{x,i} = 0, x, i \in \{1, \dots, N\}\}$$

$$A^-(V) = \{A_{x,i}(V) \in H_C / v_{x,i} = 1, x, i \in \{1, \dots, N\}\}$$

La proposition suivante donne des conditions suffisantes pour éviter l'équilibre des tournées non admissibles $V \in H_C - H_T$.

Propositions 3.1. [34]. *Si les paramètres A, B, C, D sont non négatives et la fonction d'énergie pour le CHN vérifie :*

$$\max_{\{V \in H_C - H_T / S = N - 1\}} \underline{E}^0(V) < 0 < \min_{\{V \in H_C - H_T / S = N\}} \bar{E}^1(V) \quad (3.20)$$

ou

$$\max_{\{V \in H_C - H_T / S = N\}} \underline{E}^0(V) < 0 < \min_{\{V \in H_C - H_T / S = N + 1\}} \bar{E}^1(V) \quad (3.21)$$

Alors, toute tournée non admissible $V \in H_C - H_T$ ne peut pas être un point d'équilibre.

Théorème 3.2. [34]. *Si les coefficients de la fonction d'énergie pour le CHN sont non négatifs et vérifiés :*

$$C(N - N') + \min \{B, A + Dd_L, (N - 1)A\} > 0 \quad (3.22)$$

$$3Dd_U - C + C(N - N') < 0 \quad (3.23)$$

Alors, toute tournée non admissible $V \in H_C - H_T$ ne peut pas être un point d'équilibre.

Non convergence des solutions non admissibles : Cas $V \in H - H_C$:

Le théorème 3.2 donne des conditions suffisantes pour que le CHN ne se laissera pas piéger par une solution non admissible $V \in H_C - H_T$. Cependant, il peut y avoir des points $V \notin H_C$ (visites non valables par conséquent) qui sont également des points d'équilibre. Voir l'exemple ci-dessous. Dans l'exemple 3.1, les paramètres suivants :

$$N' = 8.1 \quad A = 1.7 \quad B = 2.2 \quad C = 1 \quad D = 0.25$$

vérifient les inégalités (3.22) et (3.23).

Cependant, le point

$$V^e = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0.9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \in H - H_C$$

ne peut pas être un tour et aussi un point d'équilibre car elle vérifie les équations (3.8), (3.9) et (3.10).

1. $E_{x,i}(V^e) > 0 \quad \forall (x,i) \in \{1, \dots, N\}^2$ de telle sorte que $v_{x,i}^e = 0$
Soit $(x,i) = (2,4)$, par exemple, l'un de ces éléments de la matrice V^e , qui vérifie $v_{2,4}^e = 0$;
alors par l'équation (3.13) :
$$E_{2,4}(V^e) = (1.7)(0) + (2.2)(1) + (1)(5.9 - 8.1) + 0.25((2\sqrt{3})(0.9) + (2\sqrt{3})(1)) = 1.645 > 0$$

Soit $(x,i) = (2,3)$, par exemple, l'un de ces éléments de la matrice V^e , qui vérifie $v_{2,3}^e = 0$;
alors par l'équation (3.13) :
$$E_{3,1}(V^e) = (1.7)(0) + (2.2)(0) + (1)(5.9 - 8.1) + 0.25((2\sqrt{3})(1) + (2)(1)) = -0.834 < 0$$
2. $E_{x,i}(V^e) < 0 \quad \forall (x,i) \in \{1, \dots, N\}^2$ de telle sorte que $v_{x,i}^e = 1$
Soit $(x,i) = (3,1)$, par exemple, l'un de ces éléments de la matrice V^e , qui vérifie $v_{3,1}^e = 1$;
alors par l'équation (3.13) :
$$E_{3,1}(V^e) = (1.7)(0) + (2.2)(0) + (1)(5.9 - 8.1) + 0.25((2\sqrt{3})(1) + (2)(1)) = -0.834 < 0$$

Soit $(x,i) = (4,2)$, par exemple, l'un de ces éléments de la matrice V^e , qui vérifie $v_{4,2}^e = 1$;
alors par l'équation (3.13) :
$$E_{4,2}(V^e) = (1.7)(0.9) + (2.2)(0) + (1)(5.9 - 8.1) + 0.25((0)(0) + (2)(1)) = -0.17 < 0$$
3. $E_{x,i}(V^e) = 0 \quad \forall (x,i) \in \{1, \dots, N\}^2$ de telle sorte que $0 < v_{x,i}^e < 1$

L'unique élément de la matrice V^e qui vérifie ces conditions est $(x,i) = (4,3)$, car $v_{4,3}^e = 0.9$; de manière analogue :

$$E_{4,3}(V^e) = (1.7)(1) + (2.2)(0) + (1)(5.9 - 8.1) + 0.25(2)(1) = 0$$

Ce point V^e est un point intérieur du cube de Hamming et c'est aussi un point de stabilité. Ces points nécessitent une analyse supplémentaire afin d'éviter leur convergence dans le CHN. Certains auteurs, (Sungkwon Park, 1989), ont analysé la stabilité des points intérieurs. Cependant, ils ont supposé que tous les éléments de la diagonale de la matrice de connexion sont nuls, i.e. $T_{i,i} = 0 \quad \forall i \in \{1, \dots, n\}$.

Évidemment, ce n'est pas le cas pour le TSP. En effet,

$$T_{xi,xi} = -C < 0 \quad \forall (x, i) \in \{1, \dots, N\}^2$$

Nous devons nous rappeler que tout point $V \in H$ de l'hypercube \mathbb{R}^n est un point intérieur $V \in H - H_C$ si et seulement s'il existe un composant $v_{x,i} \in (0, 1)$. Afin de le analyser, nous pouvons distinguer le concept suivant :

Définition 3.2. *Un point intérieur $V \in H - H_C$ sera un point d'arête si*

$$\left| \left\{ (x, i) \in \{1, \dots, N\}^2 / v_{x,i} \in (0, 1) \right\} \right| = 1$$

Tout point intérieur $V \in H - H_C$ not étant un point d'arête est un point de selle par rapport à la fonction d'énergie et, par conséquent, a une probabilité nulle d'être le point de convergence du CHN. Sinon, des conditions nécessaires et suffisantes de la stabilité des points d'arêtes seront introduites.

Le point stable précédent V^e de l'exemple 4.1 est un point d'arête.

Pour que tout soit clair, la matrice de connexion du réseau T en évitant la dépendance explicite de ses éléments par rapport au paramètres A, B, C et D sera utilisé.

Un lemme technique 3.1 est nécessaire.

Lemme 3.1. [34]. *Étant donné $V, V' \in H$. Alors :*

$$E_{x,i}(V') = E_{x,i}(V) + \sum_{y,j} (v_{y,j} - v'_{y,j}) T_{xi,yj} \quad \forall (x, i) \in \{1, \dots, N\}^2$$

Propositions 3.2. [34]. *Soit $V \in H - H_C$ un point d'arête avec $v_{x^0, i^0} \in (0, 1)$. Ce point V sera un point d'équilibre pour le CHN avec $T_{xi,xi} < 0 \quad \forall (x, i) \in \{1, \dots, N\}^2$ si et seulement si le point de sommet $V' \in H_C$ défini par :*

$$v'_{x,i} = \begin{cases} v_{x,i} & \text{if } (x, i) \neq (x^0, i^0) \\ 1 & \text{if } (x, i) = (x^0, i^0) \end{cases}$$

vérifie toutes ces conditions :

$$0 < E_{x^0, i^0}(V') < -T_{x^0 i^0, x^0 i^0} \quad (3.24)$$

$$E_{x,i}(V') \geq \frac{E_{x^0, i^0}(V')}{T_{x^0 i^0, x^0 i^0}} T_{xi, x^0 i^0} \quad \forall (x, i) / v'_{x,i} = 0 \quad (3.25)$$

$$E_{x,i}(V') \leq \frac{E_{x^0, i^0}(V')}{T_{x^0 i^0, x^0 i^0}} T_{xi, x^0 i^0} \quad \forall (x, i) / v'_{x,i} = 1 \quad (3.26)$$

Les conditions nécessaires et suffisantes données dans la Proposition 3.2 sont basées sur le point de sommet $V' \in H_C$ obtenu lorsque la valeur v_{x^0, i^0} est augmentée. De manière analogue, si cette valeur est diminuée, un autre point de sommet peut être obtenu et d'autres conditions sont déduites. C'est la proposition suivante.

Propositions 3.3. [34]. *Soit $V \in H - H_C$ un point d'arête avec $v_{x^0, i^0} \in (0, 1)$. Alors, ce point V sera un point d'équilibre pour le CHN avec $T_{xi, xi} < 0 \quad \forall (x, i) \in \{1, \dots, N\}^2$ si et seulement si le point de sommet V' dans H_C défini par :*

$$v'_{x,i} = \begin{cases} v_{x,i} & \text{si } (x, i) \neq (x^0, i^0) \\ 0 & \text{si } (x, i) = (x^0, i^0) \end{cases}$$

vérifie toutes ces conditions :

$$T_{x^0 i^0, x^0 i^0} < E_{x^0, i^0}(V') < 0 \quad (3.27)$$

$$E_{x,i}(V') \geq \frac{E_{x^0, i^0}(V')}{T_{x^0 i^0, x^0 i^0}} T_{xi, x^0 i^0} \quad \forall (x, i)/v'_{x,i} = 0 \quad (3.28)$$

$$E_{x,i}(V') \leq \frac{E_{x^0, i^0}(V')}{T_{x^0 i^0, x^0 i^0}} T_{xi, x^0 i^0} \quad \forall (x, i)/v'_{x,i} = 1 \quad (3.29)$$

Sur la base des propositions 3.2 et 3.3, en évitant ces conditions pour les points de sommets adjacents, la stabilité de tout point d'arête sera évitée si

$$E_{x,i}(V') \notin (0, -T_{xi, xi}) = (0, C) \quad \forall (x, i)/v'_{x,i} = 1 \quad \forall V' \in H_C$$

ou

$$E_{x,i}(V') \notin (T_{xi, xi}, 0) = (-C, 0) \quad \forall (x, i)/v'_{x,i} = 0, \quad \forall V' \in H_C$$

sont vérifiés.

Malheureusement, de telles contraintes sont si restrictives qu'elles sont incompatibles avec les conditions des théorèmes 3.1 et 4.1.

Pour surmonter cette difficulté, nous éviterons les points d'arête les plus probablement stables en utilisant $\underline{E}^0(V) \leq -C$ au lieu de l'équation (3.18) et en utilisant $\bar{E}^1(V) \geq C$ au lieu de l'équation (3.19). De cette façon, les inégalités (3.22) et (3.23) du théorème 3.2 seront substituées par :

$$3Dd_U - C + C(N - N') \leq -C \quad (3.30)$$

$$C(N - N') + \min\{B, A + Dd_L, (N - 1)A\} \geq C \quad (3.31)$$

respectivement.

Dans l'exemple 3.1, les valeurs des paramètres du point stable intérieur V^e vérifient l'inégalité (3.23) et ne vérifient pas l'inégalité (3.30) :

$$3Dd_U - C + C(N - N') = -0.1 \in (-C, 0) = (-1, 0)$$

En conséquence des résultats ci-dessus, les conditions suivantes assurent que tout sommet stable $V \in H_C$ sera une solution admissible du TSP en évitant également que tout point intérieur $V \notin H_C$ d'être stable.

D'après le théorème 3.1, les bornes inférieures et supérieures pour $C(N' - N)$ sont :

$$2Dd_U \leq C(N' - N) \leq A + B + Dd_L \quad (3.32)$$

À partir des inégalités (3.30) et (3.31), les bornes suivantes pour $C(N' - N)$ sont obtenus :

$$3Dd_U \leq C(N' - N) \leq \min\{B; A + Dd_L; (N - 1)A\} - C \quad (3.33)$$

Cependant, Les bornes (3.32) sont impliquées par les bornes (3.33).
À partir de ces derniers, le cadre suivant est proposé :

$$\begin{cases} B = 3d_U + C \\ A = (3d_U - d_L D) + C = B - Dd_L \\ N' = N + \frac{3Dd_U}{C} \end{cases} \quad (3.34)$$

Si nous prenons $C' = k \cdot C$ et $D' = k \cdot D$, la nouvelle fonction d'énergie avec le paramétrage ci-dessus est $E' = k \cdot E$.

Par conséquent, si l'un de ces paramètres est fixe, par exemple, $D = \frac{1}{d_U}$ alors il n'y a qu'un seul degré de liberté C .

Remarque 3.3.

Le paramètre D est toujours une échelle de distances de la TSP, elle peut être considérée comme une échelle de paramètres. Par conséquent, le modèle reste valable en cas de changement d'unités de distance.

Dans l'exemple 3.1, si $C = 1$, par l'équation (3.34) :

$$\begin{cases} D = \frac{1}{d_U} = 0.25 \\ B = 3d_U + C = 4 \\ A = B - Dd_L = 3.5 \\ N' = N + \frac{3Dd_U}{C} = 9 \end{cases}$$

• Les étapes suivantes sont suivies pour implémenter ce réseau :

1. Étant donné le nombre de villes N et leurs coordonnées, calculez la matrice distance D
2. Initialiser le réseau et définir la matrice de poids comme indiqué dans l'équation (3.4).
3. Affecter aléatoirement les états d'entrée initiaux au réseau. Après ça, le réseau est laissé à lui-même, et il passe par une séquence d'états, jusqu'à ce qu'il converge vers une solution stable.
4. Calculer l'énergie pour chaque itération. L'énergie doit diminuer d'itération par itération.
5. Répétez la mise à jour de l'activation et de la sortie jusqu'à ce que le réseau converge vers une solution stable.

Cela se fait lorsque la variation de l'énergie entre deux itérations successives est inférieure à une petite valeur seuil (~ 0.000001) ou, lorsque l'énergie commence à augmenter au lieu de diminuer.

3.5 Résultats de simulation

Le réseau neuronal Hopfield est efficace et peut converger vers des états stables en un nombre fini d'itérations. On observe que jusqu'à un problème de 20 villes, le réseau converge bien vers un état stable la plupart du temps avec une solution minimale globale. Cependant, plus le nombre de villes augmente, moins le réseau converge vers un état stable. Les résultats

graphiques sont présentés dans la figure 3.3. Les données d'entrée fournies par l'utilisateur, c'est-à-dire le nombre de villes et leurs distances sous forme de matrice de distances. Ensuite, pour ces villes, la sortie générée est exprimée sous forme de matrice de tournée, de trajet de tournée et de distance totale parcourue.

Les résultats ci-dessous concernent respectivement le nombre d'itérations et le temps nécessaire pour faire converger le réseau en fonction de nombre de villes :

Villes	5	10	15	20	25
Itérations	151	420	562	605	643
Temps (s)	0.2434	0.489	0.5824	0.623	0.7098

Les représentations graphiques de sortie des trajets pour 5, 10, 15, 20 et 25 villes sont présentées à la figure 3.3 ci-dessous. La figure 3.4 illustre la convergence énergétique pour les problèmes de 5, 10, 15 et 20 villes et la figure 3.5 montre le nombre d'itérations nécessaires à la convergence en fonction du nombre de villes et la figure 3.6 illustre le temps nécessaire pour la convergence du réseau en fonction du nombre de villes.

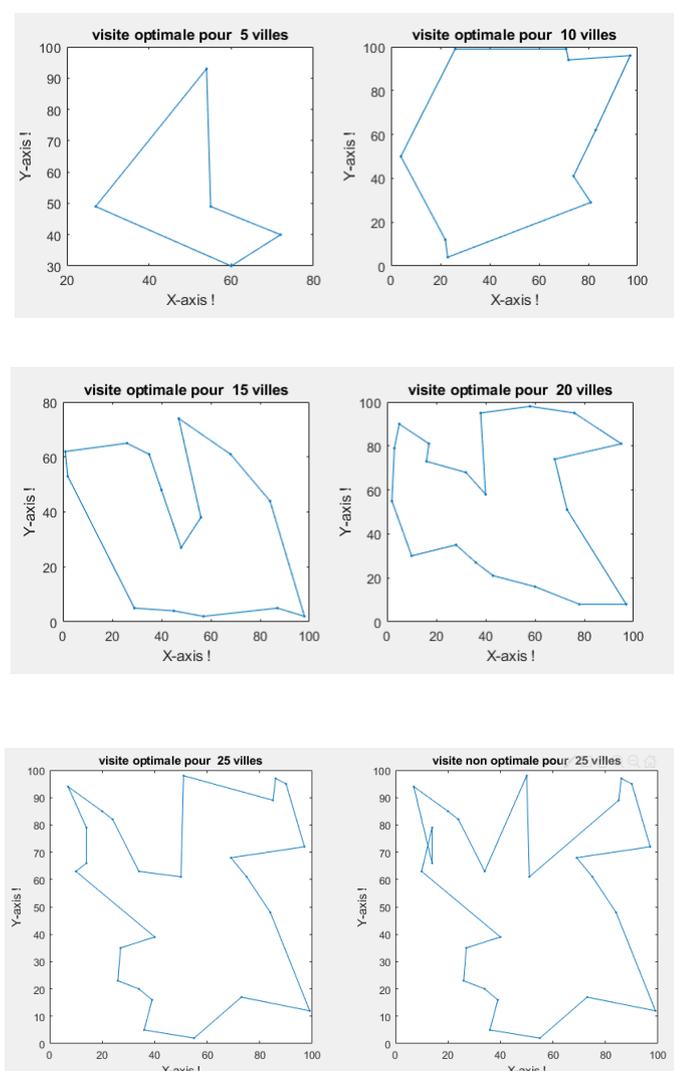


FIGURE 3.3 – Circuit de voyage pour 5, 10, 15, 20 et 25 villes.

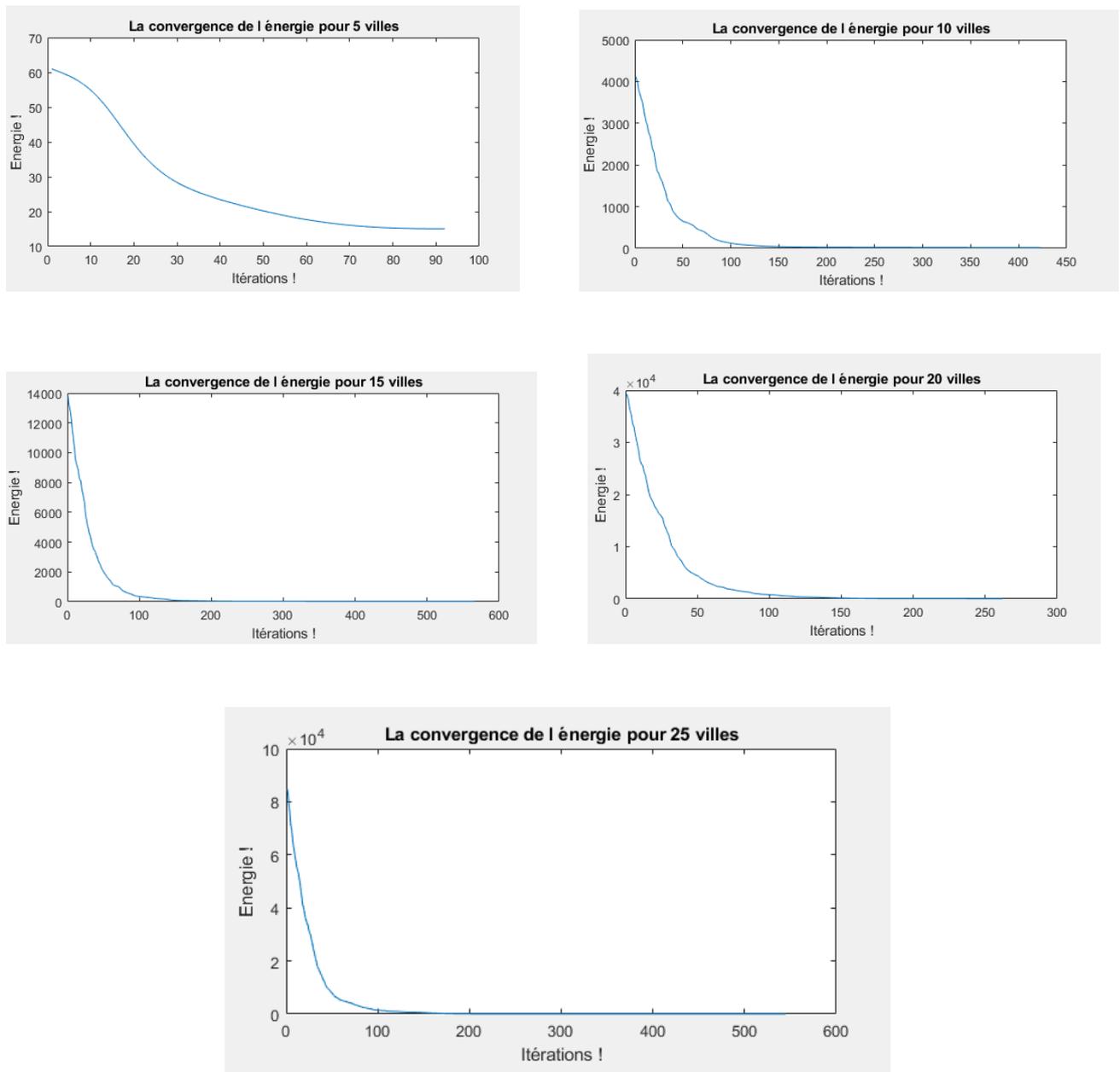


FIGURE 3.4 – Convergence de l'énergie : 5, 10, 15, 20 et 25 villes.

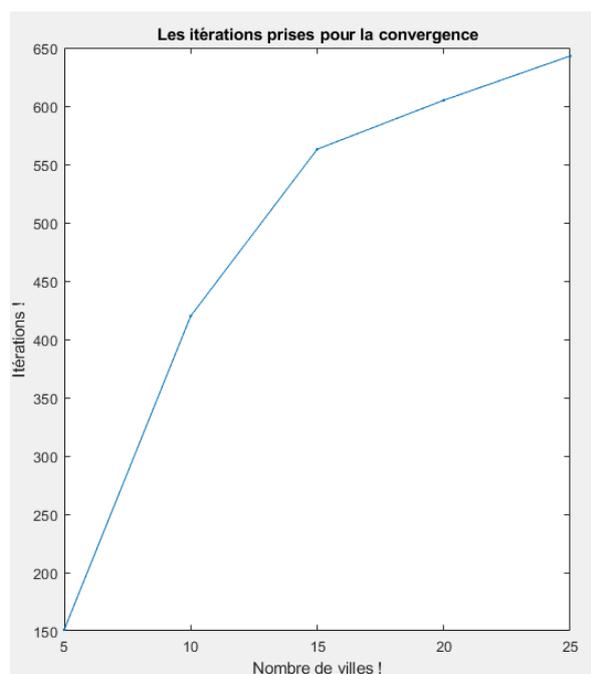


FIGURE 3.5 – Nombre d’itérations nécessaires à la convergence en fonction du nombre de villes.

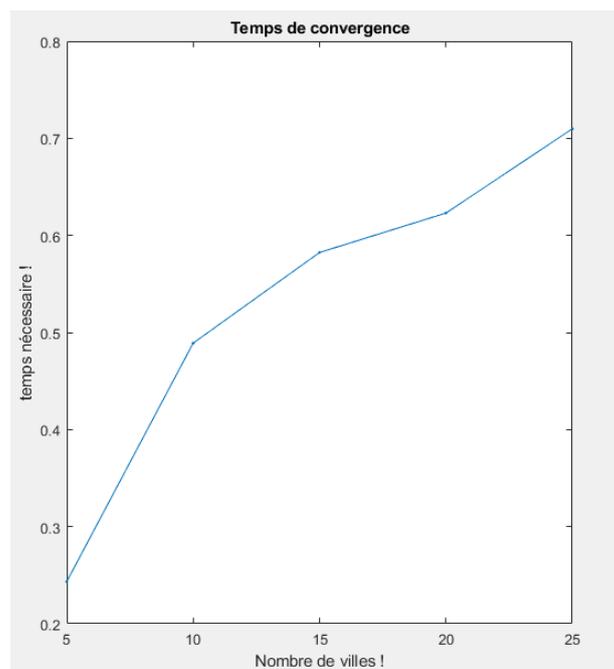


FIGURE 3.6 – Temps nécessaire pour la convergence du réseau en fonction du nombre de villes.

Commentaires sur les résultats :

- Les résultats ci-dessus montrent que lorsque le nombre de villes augmente, le nombre d’itérations nécessaires augmente fortement. L’augmentation n’est pas linéaire.
- Le nombre d’itérations requises pour la convergence n’est pas le même pour toutes les villes. Par exemple, pour 5 villes, le réseau a généralement convergé entre 120 et 170 itérations. mais à quelques occasions, il a nécessité environ 80 itérations, alors que dans quelques cas, il n’a pas convergé du tout ou a nécessité plus de 250 itérations. Ce ceci est dû au fait que l’état initial du réseau est généré de manière aléatoire. Cela peut parfois pas de convergence.
- La plupart du temps, le résultat converge vers un minimum local au lieu d’un minimum global. Pour éviter cela, un biais aléatoire est ajouté aux entrées initiales.
- L’algorithme développé est non déterministe. Il ne garantit donc pas une solution optimale à chaque fois. Bien qu’il donne une solution quasi optimale dans la plupart des cas, il peut ne pas converger et donne une solution correcte. Souvent, lorsque l’énergie du système était calculée, on constatait qu’elle augmentait au lieu de diminuer.
- L’algorithme a donc échoué dans quelques cas. Il s’agit là encore d’une conséquence de l’état initial aléatoire du réseau.
- Dans 93 % cas de tests, l’algorithme a convergé, tandis que dans 7% cas, l’algorithme n’a pas convergé et parfois l’énergie du système a augmenté au lieu de diminuer pendant que le réseau itère vers la convergence.
- La structure du réseau neuronal de Hopfield est très optimale pour la solution de TSP. Il peut être facilement utilisé pour les problèmes d’optimisation comme celui de TSP.
- Il donne des résultats très précis grâce à l’équation énergétique très puissante et complète développée par Hopfield et Tank.

- Cette approche par réseau neuronal est très rapide par rapport aux techniques de programmation standard utilisées pour la résolution des TSP. Avec très peu de changements, cet algorithme peut être modifié pour obtenir la solution approximative de nombreux autres problèmes NP-complets.

Conclusion générale

Nous avons décomposé ce mémoire en trois chapitres :

Dans le premier chapitre, nous avons introduit les définitions des systèmes dynamiques et ses propriétés, également nous avons étudié la stabilité des systèmes dynamiques autonomes.

Puis, dans le deuxième chapitre nous avons vu qu'un réseau de neurones de Hopfield est un système dynamique et que son évolution peut être agencée de plusieurs façons, de plus quand ces réseaux sont utilisés pour résoudre des problèmes d'optimisation les poids des connexions sont déterminés analytiquement à partir de la formulation du problème ; en général, cela est fait directement à partir de la fonction énergie associée au problème, toutes ces applications nécessitent un codage du problème sous la forme d'un réseau de neurones artificiels.

On utilise le réseau de la manière suivante : à partir d'un état initial, on laisse le réseau évoluer librement jusqu'à un attracteur, qui est généralement, pour les problèmes d'optimisation, un état indépendant du temps (c'est-à-dire un point fixe de la dynamique). On dit alors que le réseau converge : la convergence est atteinte lorsque les sorties des neurones n'évoluent plus. L'attracteur vers lequel converge le réseau, codé une solution au problème d'optimisation, cette solution est en général localement optimale.

Ensuite, dans le troisième chapitre nous allons détailler la résolution du problème du voyageur de commerce, proposée par Hopfield et Tank en 1985, par le réseau de Hopfield. Les résultats montrent que lorsque le nombre de villes augmente, le nombre d'itérations nécessaires pour la convergence du réseau augmente fortement, de plus dans la plupart du temps, le réseau converge vers un minimum local au lieu d'un minimum global.

Bibliographie

- [1] **A. Julien-Laferriere** ; Hopfield Network.
- [2] **A.Yacoubi** ; Contribution de l'approche neuromimetique a l'analyse du mouvement dans une sequence d'images (1996).
- [3] **A. EL JAI** et **E. ZERRIK** ; Stabilité des Systèmes Dynamiques (2013).
- [4] **A.Halimi** ; Analyse Interne Des Systèmes Dynamiques (2015).
- [5] **A. Valibouze** ; Réseaux de neurones artificiels (2018-2019).
- [6] **A.Berkani** ; Métaheuristique Hybride Réseaux de Neurones Artificiels-PSO du Recuit Simulé pour la Commande d'un Procédé Industriel Non-linéaire.
- [7] **A.Désilles** ; Introduction à la théorie des systèmes dynamiques à temps discret, 24 septembre 2003.
- [8] **A.Chamekh** ; Optimisation des procédés de mise en forme par les réseaux de neurones artificiels (2010).
- [9] **Behzad Kamgar-Parsi** et **Behrooz Kamgar-Parsi** ; On Problem Solving with Hopfield Neural,Biol. Cybern. 62, 415-423 (1990).
- [10] **B.Bhiri** ; Stabilité et stabilisation en temps fini des systèmes dynamiques (2017).
- [11] **C. Bennani** ; Stabilisation et Estimation de l'état des Systèmes Dynamiques non Linéaires et Applications (2011).
- [12] **C. MAKREM** ; Optimisation de l'ordonnancement par l'approche hybride basée sur les réseaux de neurones ; Montréal, LE 21 Décembre 2004.
- [13] **Charles-Michel MARLE** ; Systèmes Dynamiques. Une introduction ; Ellipses.
- [14] **C.Medjahed** ; Mise en point d'un programme d'optimisation pour le stockage d'énergie thermique, 2012/2013.
- [15] **D. Marx** ; Contribution à l'étude de la stabilité des systèmes électrotechniques. Autre. Institut National Polytechnique de Lorraine, 2009.
- [16] **D.Graupe** ; Principles of Artificial Neural Networks (2nd Edition), University of Illinois, Chicago, USA.
- [17] **E.Daucé** ; adaptation dynamique et apprentissage dans des réseaux de neurones récurrents aléatoires (2000).
- [18] **E.Merid-Casermeiro, G.Galan-Marin et J.Munoz-Perez** ; An Efficient Multivalued Hopfield Network for the Traveling Salesman Problem ; Neural Processing Letters 14 : 203-216, 2001.
- [19] **F.Blayo et M.Verleysen** ; Les réseaux de neurones artificiels (1996).
- [20] **F. Favata et R. Walker** ; A study of the application of Kohonen-type neural networks to the Travelling Salesman Problem ; Biol. Cybern. 64, 463-468 (1991).

- [21] **G. Dreyfus, J.-M. Martinez, M. Samuelides M. B. Gordon, F. Badran et S. Thiria** ; Réseaux de neurones – Méthodologie et applications.
- [22] **G. V. Wilson and G. S. Pawley** ; On the Stability of the Travelling Salesman Problem Algorithm of Hopfield and Tank ; Biol. Cybern. 58, 63-70 (1988).
- [23] **H.Ghaddab** ; Systèmes dynamiques continus (2020).
- [24] **J.Manoel Gomes Da Silva Jr** ; Sur la stabilité locale de systèmes linéaires avec saturation des commandes (1997).
- [25] **J. J. Hopfield**, “Neural networks and physical systems with emergent collective computational abilities.,” Proc. Natl. Acad. Sci., vol. 79, no. 8, pp. 2554–2558, 1982.
- [26] **J. J. Hopfield and D. W. Tank**, “‘Neural’ computation of decisions in optimization problems,” Biol. Cybern., vol. 52, no. 3, pp. 141–152, 1985.
- [27] **J.Y.Haggege** ; Les réseaux de neurones. Réseaux de Hopfield.
- [28] **J.Mandziuk** ; Solving the travelling salesman problem with a Hopfield-type neural network (1996).
- [29] **Jan van den Berg** ; Generalized Hopfield Networks for Constrained Optimization.
- [30] **Jean-Yves Potvin** ; Artificial neural networks for combinatorial optimization.
- [31] **Jean-Yves Potvin** ; The Traveling Salesman Problem : A Neural Network Perspective, Centre de Recherche sur les Transports Université de Montréal.
- [32] **J. J. Hopfield**, “Neurons with graded response have collective computational properties like those of two-state neurons.,” Proc. Natl. Acad. Sci., vol. 81, no. 10, pp. 3088–3092, 1984.
- [33] **M.A.Arbib** ; The Handbook of Brain Theory and Neural Networks, Second Edition.
- [34] **M.Talavan, J.Yanez** ; Parameter setting of the Hopfield network applied to TSP ; Neural Networks 15(2002) 363 – 373.
- [35] **M.EL ALAOUI** ; Optimisation des requêtes dans les bases de données décisionnelles via la programmation par contraintes,(2018).
- [36] **O. Bonnefoy** ; Systèmes Dynamiques et instabilités hydrodynamiques.
- [37] **O.BONNEFOY** ; Systèmes Dynamiques et instabilités hydrodynamiques.
- [38] **Pac Jean-Louis** ; Systèmes dynamiques. cours et exercices corrigés, 2ème édition, Dunod.
- [39] **R.Huet** ; Codage neuronal parcimonieux pour un système de vision ; Université de Bretagne Sud.
- [40] **S. Assia** ; Systèmes dynamiques chaotiques et synchronisation (2020).
- [41] **S.Rezzag** ; Etude et estimation des bornes de systèmes dynamiques chaotiques et hyperchaotiques.
- [42] **S. Masri** ; Systèmes dynamiques discrets, Juin 2019.
- [43] **William J. Wolfe** ; Fuzzy Hopfield-Tank TSP model ; Proc. SPIE 3722, Applications and Science of Computational Intelligence II, (22 March 1999) ;
- [44] **Y. Takefuji** ; Neural network parallel computing (1955).
- [45] **Y. Djeriri** ; Les Réseaux de Neurones Artificiels (2017).
- [46] **Y.Morere** ; Les Réseaux de Neurones Récurrents, Septembre 1998.
- [47] **Y.Takefuji**, Neural Network Parallel Computing, Kluwer Academic Publishers, 1992.
- [48] **A.Cichocki,R.Unberhauen**, Neural Networks for Optimization and Signal Processing, 1993.
- [49] **S.V. B.Ayer**, A theoretical investigation into the performance of the Hopfield model, IEEE Transactions on Neural Networks, vol.1, p. 204-215, juin 1990.