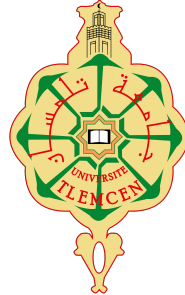


RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE



UNIVERSITÉ ABOU BEKR BELKAID DE TLEMCE
FACULTÉ DES SCIENCES
DÉPARTEMENT DE MATHÉMATIQUES

MÉMOIRE DE MASTER

Domaine : Mathématiques et Informatique
Filière : Mathématiques
Spécialité : Probabilités Approfondies et Statistiques

Présenté Par :
Mr NEKKACHE Abdellatif

ÉTUDE DU MODÈLE DU CHEMOSTAT À CROISSANCE NON
MONOTONE AVEC PERTURBATIONS ALÉATOIRES SUR LE TAUX
DE DILUTION.

Soutenu le 23 / 06 / 2025 devant le jury composé de :

M. Yadi Karim	Pr. Université Abou Bekr Belkaid, Tlemcen	Président
Mme Benyahia Wahiba	M.C.A. Université Abou Bekr Belkaid, Tlemcen	Examinatrice
Mme Bensmain Nawel	M.C.B. Université Abou Bekr Belkaid, Tlemcen	Examinatrice
Mme Dali Youcef Manel	M.C. École d'ingénieurs ECAM EPMI, Cergy	Encadrante
Mme Korso Féciane Malika	M.C.A. Université Abou Bekr Belkaid, Tlemcen	Co-Encadrante
M. Catellier Rémi	M.C. Université de Côte d'Azur, Nice	Co-Encadrant

Année Universitaire : 2024 - 2025

Remerciements

Avant toute chose, je rends grâce à DIEU, le Tout-Puissant, pour la patience, la force et la persévérance qu'Il m'a accordées tout au long de la réalisation de ce travail, me permettant ainsi d'atteindre mes objectifs.

Je tiens ensuite à exprimer ma profonde gratitude aux membres du jury, Madame BENYAHIA Wahiba, Madame BENSMAIN Nawel et Monsieur YADI Karim, pour l'intérêt qu'ils ont porté à ce mémoire et le temps qu'ils ont bien voulu consacrer à son évaluation. Je leur suis très reconnaissant pour leurs remarques et leur disponibilité.

Je souhaite également remercier chaleureusement Madame DALI YUCEF Manel pour ses explications claires, sa patience, et le temps précieux qu'elle m'a accordé pour corriger mes erreurs. Sa contribution a été essentielle à l'élaboration de ce mémoire. Travailler avec elle fut une expérience aussi enrichissante qu'agréable.

Ma reconnaissance s'adresse aussi à Madame KORSO FECIANE Malika, mon enseignante, pour son accompagnement constant. Sa disponibilité, sa bienveillance et ses conseils avisés m'ont grandement soutenu tout au long de ce travail. J'ai eu la chance de bénéficier de son expertise et de son encadrement rigoureux, et je lui en suis profondément reconnaissant.

Je remercie également Monsieur CATELLIER Rémi pour le temps qu'il a consacré à la relecture de mes démonstrations mathématiques. Ses éclaircissements sur certains aspects théoriques complexes ont permis d'apporter davantage de rigueur et de clarté à ce mémoire. Notre collaboration fut très enrichissante.

Enfin, je tiens à remercier chaleureusement ma famille et mes amis pour leur soutien moral indéfectible, leurs encouragements constants, et leur présence bienveillante, qui m'ont permis de garder confiance et motivation tout au long de cette aventure académique.

Résumé

Le chimostat constitue un modèle fondamental en biologie pour décrire la croissance des micro-organismes dans un environnement contrôlé. Ce mémoire est consacré à l'étude du modèle du chimostat avec une fonction de croissance non monotone telle que la fonction de Haldane, soumis à des perturbations aléatoires sur le taux de dilution.

Après une introduction biologique et mathématique du modèle, on présente les outils fondamentaux de la théorie de la stabilité des systèmes dynamiques à deux équations. On commence par une analyse du modèle déterministe en étudiant l'existence et la stabilité des équilibres. Une composante aléatoire est ensuite introduite à travers un bruit additif affectant le taux de dilution, modélisé à l'aide d'un processus d'Ornstein-Uhlenbeck. Ce passage au cadre stochastique permet d'examiner l'influence des perturbations sur la dynamique du système, notamment en ce qui concerne la persistance et l'extinction de la biomasse.

Enfin, des simulations numériques sont proposées afin d'illustrer les comportements dynamiques du système selon différents régimes de perturbation. Les scripts Python utilisés pour la génération des figures ainsi que quelques notions de mathématique fondamentale sont regroupés dans deux annexes à la fin du mémoire.

Abstract

The chemostat constitutes a fundamental model in biology for describing the growth of microorganisms in a controlled environment. This thesis is devoted to the study of the chemostat model with a non-monotonic growth function such as the Haldane function, subject to random disturbances on the dilution rate.

After a biological and mathematical introduction to the model, we present the fundamental tools of the stability theory for two-equation dynamical systems. We begin with an analysis of the deterministic model by studying the existence and stability of equilibria. A random component is then introduced through an additive noise affecting the dilution rate, modeled using an Ornstein-Uhlenbeck process. This transition to the stochastic framework allows us to examine the influence of disturbances on the system dynamics, particularly with regard to the persistence and extinction of biomass.

Finally, numerical simulations are proposed to illustrate the dynamic behaviors of the system under different disturbance regimes. The Python scripts used to generate the figures, as well as some fundamental mathematical concepts, are grouped in two appendices at the end of the thesis.

Glossaire

Acronymes

GAS : Globalement Asymptotiquement Stable.

LES : Localement Exponentiellement Stable.

I : Instable.

O-U : Ornstein-Uhlenbeck.

EDS : Équation Différentielle Stochastique

M.B.B : Mouvement Brownien Bilatéral.

p.s : Presque sûrement

Symboles mathématiques

$\approx$: Signe d'approximation.

$\sim$: Suit la même loi.

$\stackrel{\mathcal{L}}{=}$: Egalité en loi.

Table des figures

1.1	Représentation simplifiée d'un chémostat.	3
1.2	Allure de la fonction de Haldane.	5
2.1	Comportement local autour d'un point d'équilibre stable	10
2.2	Comportement local autour d'un point d'équilibre instable	10
2.3	Comportement local autour d'un point d'équilibre : shunt positif	11
2.4	Comportement local autour d'un point d'équilibre : shunt négatif	11
2.5	L'allure locale d'un noeud instable , $\lambda_1 > 0, \lambda_2 > 0$	17
2.6	L'allure locale d'un noeud stable , $\lambda_1 < 0, \lambda_2 < 0$	17
2.7	L'allure locale d'un point selle , $\lambda_1 > 0, \lambda_2 < 0$	18
2.8	L'allure locale d'une étoile instable , $\lambda > 0$	18
2.9	L'allure locale d'une étoile stable , $\lambda < 0$	19
2.10	L'allure locale d'un noeud dégénéré instable , $\lambda > 0$	19
2.11	L'allure locale d'un noeud dégénéré stable , $\lambda < 0$	20
2.12	L'allure locale d'un foyer instable , $\alpha > 0, \beta > 0$	20
2.13	L'allure locale d'un foyer stable , $\alpha < 0, \beta < 0$	21
2.14	L'allure locale d'un centre , $\alpha = 0, \beta > 0$	21
2.15	Classification des portraits de phase du système $\dot{X} = AX$ en fonction de la trace et du déterminant.	22
3.1	Configuration d'un chémostat.	26
3.2	Solution de l'équation $f(s) = D$ avec f une fonction de type Haldane.	26
3.3	Diagramme opératoire d'un chémostat avec une fonction de croissance non-monotone.	30
3.4	Stabilité des équilibres.	31
5.1	Simulation de l'extinction de la biomasse.	51
5.2	Fonction de croissance f avec repères des valeurs D, D_l , et D_r dans le cas d'extinction de la biomasse. Paramètres choisis : $s_{in} = 14, s_m = 5.4772, D = 1.4, D_l = 1.15, D_r = 1.65, f(s_{in}) = 0.7943, f(s_m) = 1.0699$	52
5.3	Simulation de la faible persistance de la biomasse.	52
5.4	Fonction de croissance f avec repères des valeurs D, D_l , et D_r dans le cas de la persistance faible de la biomasse. Paramètres choisis : $s_{in} = 16.7, s_m = 7, D = 1.7, D_l = 1.45, D_r = 1.95, f(s_{in}) = 1.8397, f(s_m) = 2.3333$	53

Table des matières

Glossaire

1	Introduction	3
1.1	Fonctions de croissance	4
1.2	Le chémostat en biologie	5
1.3	Modèle du chémostat avec perturbations aléatoires	6
2	Quelques notions fondamentales dans la théorie de stabilité dans un modèle déterministe	9
2.1	Équation différentielle ordinaire scalaire (EDO)	9
2.2	Étude de deux équations différentielles ordinaires	11
2.3	Linéarisation au voisinage d'un point d'équilibre	12
2.4	Solution d'un système dynamiques linéaire	13
2.5	Typologie des systèmes planaires linéaires	16
2.6	Typologie des solutions des systèmes linéaires dans le plan	21
2.7	Stabilité asymptotique et stabilité exponentielle.	22
3	Etude du modèle mathématique déterministe du chémostat simple	25
3.1	Présentation du modèle déterministe	25
3.2	Classification des équilibres	28
4	Étude du modèle stochastique du chémostat simple	33
4.1	Processus d'O-U	33
4.1.1	Solution stationnaire	34
4.1.2	Changement de variable et flux de Wiener	36
4.2	Introduction des perturbations dans le modèle	42
4.2.1	Contrôle des perturbations	42
4.2.2	Système dynamique aléatoire (perturbé)	43
4.3	Persistence dans le chémostat perturbé	43
4.4	Propriétés des solutions de la dynamique aléatoire	43
4.5	Condition de persistance	47
5	Simulations numériques	51
5.1	Extinction de la biomasse	51
5.2	Faible persistance	52
A	Notions mathématiques utiles	59
B	Codes des figures sous Python	61

Chapitre 1

Introduction

Le chémostat est un dispositif expérimental fondamental en biologie et en biotechnologie, conçu pour cultiver des microorganismes (bactéries, levures, etc.) dans des conditions contrôlées. Inventé dans les années 1950 simultanément par le biologiste Monod [27, 28], qui a nommé son invention "bactogène", ainsi que par le physicien Novick et le chimiste Szilard [30], qui ont introduit le terme "chémostat". C'est un bioréacteur à alimentation continue conçu sous forme d'un réservoir dont le contenu est un milieu homogène où la température, le potentiel d'hydrogène et le volume total sont constants. Il permet d'étudier la dynamique des populations microbiennes en régime stationnaire, où l'apport continu de milieu frais et l'élimination simultanée de biomasse et de substrat maintiennent un équilibre écologique artificiel. Ce système modélise également des écosystèmes naturels, comme les lacs, où les fluctuations environnementales influencent la survie des espèces.

Rappelons rapidement en quoi consiste le dispositif du chémostat. Il est composé de trois compartiments : compartiment de nutriments, le compartiment de culture et le compartiment de collecte, qui sont interconnectés par des pompes, voir Figure 1.1. Le substrat est stocké dans le réservoir de nutriments et fourni à un débit donné au réservoir de culture, où les interactions entre le nutriment et la biomasse microbienne ont lieu. Le milieu du réservoir de culture est également prélevé vers le réservoir de collecte avec le même débit, pour maintenir le volume dans le réservoir de culture constant.

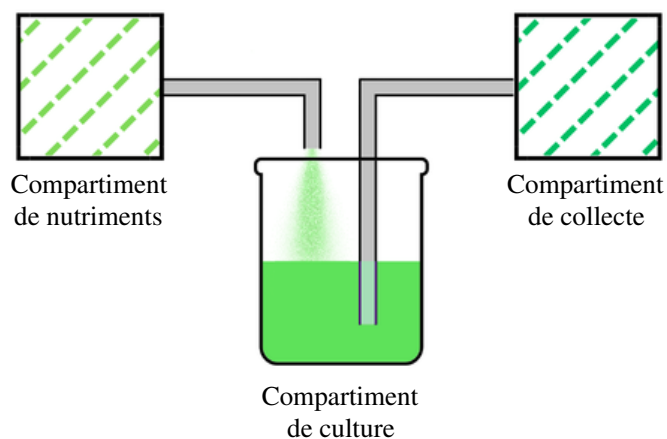


FIGURE 1.1 – Représentation simplifiée d'un chémostat.

Le modèle mathématique du chémostat est donné par le système dynamique suivant :

$$\begin{cases} \frac{ds}{dt} = D(s_{in} - s) - f(s)\frac{X}{Y} \\ \frac{dX}{dt} = -DX + f(s)X, \end{cases} \quad (1.1)$$

où

- s désigne la concentration de substrat,
- X désigne la concentration de biomasse,
- Y est le taux de conversion du substrat en biomasse,
- s_{in} est la concentration de substrat en entrée,
- D est le taux de dilution avec $D = Q/V$ où Q est le flux total constant et V est le volume total constant,
- $f(\cdot)$ est la fonction de croissance spécifique décrivant la cinétique de la consommation de substrat par l'espèce bactérienne.

1.1 Fonctions de croissance

Les fonctions de croissance sont influencées par divers facteurs environnementaux tels que la concentration en substrat et en biomasse, le pH, la température, l'oxygène dissous, la lumière et certains inhibiteurs. Parmi les fonctions de croissance les plus utilisées en modélisation mathématique, nous citons :

La fonction de Monod [26], les fonctions de Hill [29], la fonction de Haldane [19] et la fonction de Contois [11].

Ces quatre fonctions de croissance dépendent soit uniquement du substrat, soit à la fois du substrat et de la biomasse. Elles peuvent dépendre également d'autres paramètres physico-chimiques ou biologiques.

Dans ce travail, nous considérons la fonction de croissance de Haldane. Cette fonction est très utilisée parce qu'elle combine réalisme biologique, flexibilité mathématique et applicabilité pratique. Elle est particulièrement utile pour modéliser des systèmes où le substrat peut devenir inhibiteur, ce qui est fréquent dans de nombreux processus industriels, environnementaux et biologiques. La fonction de Haldane est définie par :

$$f(s) = m_0 \frac{s}{K_s + s + \frac{s^2}{K_I}}, \quad s \geq 0, \quad (1.2)$$

où

- K_s est la constante d'affinité,
- K_I un paramètre modélisant l'inhibition de la croissance sous de fortes concentrations de substrat,

- $m_0 = m \left(1 + \sqrt{\frac{K_s}{K_I}} \right)$, avec

$$m = \sup_{s>0} f(s), \quad (1.3)$$

-

$$s_m := \arg \max f(s) = \sqrt{K_s K_I}. \quad (1.4)$$

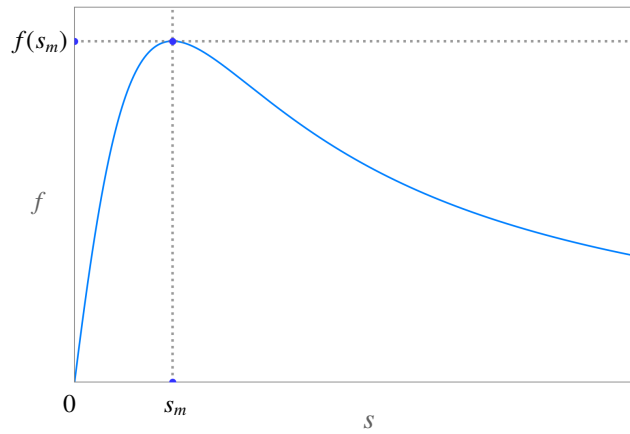


FIGURE 1.2 – Allure de la fonction de Haldane.

1.2 Le chémostat en biologie

Le chémostat, initialement conçu pour la culture de bactéries, est aujourd'hui un outil indispensable dans divers domaines industriels et scientifiques. Sa capacité à reproduire des conditions expérimentales contrôlées en fait un instrument privilégié pour optimiser les processus biologiques, notamment dans le traitement des eaux usées, la production de biogaz et la fabrication de produits pharmaceutiques.

Dans le domaine du traitement des eaux usées, le chémostat joue un rôle crucial en réduisant les concentrations de nutriments toxiques, contribuant ainsi à la préservation de l'environnement et à la santé publique. En permettant une gestion optimisée des ressources en eau, il favorise l'utilisation des eaux traitées dans des secteurs tels que l'agriculture et l'industrie, tout en préservant les réserves d'eau potable naturelles. Plus généralement, l'utilisation du chémostat permet de maintenir des conditions de culture stables.

L'efficacité du chémostat est évaluée à travers plusieurs critères de performance, dont :

- **La concentration de substrat en sortie** : Indicateur clé de l'efficacité de la biodégradation, elle reflète la réduction des polluants dans les eaux traitées, [13].
- **Le temps de séjour** : Paramètre essentiel pour déterminer la durée nécessaire au renouvellement du milieu dans le réacteur, influençant directement l'efficacité du traitement, [20].
- **Le débit de biogaz** : Mesure de la production de gaz résultant de la décomposition de matières organiques, directement liée à l'activité microbienne, [34].
- **La productivité de la biomasse** : Indicateur de la quantité de micro-organismes produits par unité de temps, essentiel pour évaluer la performance des systèmes de culture continue, [14, 15].

Ces critères font l'objet de nombreuses études, tant biologiques [3, 16] que mathématiques [12, 31, 32], visant à optimiser les processus industriels et à améliorer les performances du chémostat. En conclusion, le chémostat est un outil polyvalent et essentiel dans la recherche biologique et les applications industrielles. Sa capacité à modéliser des écosystèmes microbiens en conditions contrôlées en fait un instrument incontournable pour relever les défis environnementaux et industriels actuels, tout en ouvrant la voie à des innovations technologiques et scientifiques majeures. Pour plus de détails, le lecteur peut se référer principalement aux chapitres 1 et 2 du livre [21].

1.3 Modèle du chémostat avec perturbations aléatoires

Plusieurs approches permettent d'intégrer la stochasticité dans un modèle déterministe. Considérer des perturbations sur la fonction de croissance semble particulièrement pertinent lorsque le nombre de la biomasse est faible, avec un risque d'extinction en un temps fini. Néanmoins, les extinctions soudaines dans les cultures continues qui sont bien supervisées autour d'un régime nominal (où les paramètres sont maintenus constants et optimaux pour la croissance bactérienne) sont assez rares dans la pratique. D'un autre côté, les fluctuations du flux d'entrée qui apporte en permanence du substrat à la biomasse dans les cultures continues sont très susceptibles d'être observées. D'un point de vue biologique, le fait d'introduire un terme bruité dans le flux d'entrée de notre modèle de chémostat est un problème très intéressant rencontré en laboratoire puisque, par exemple, il reflète la présence de particules de saleté à l'intérieur des pompes ou de bouchons temporaires à l'entrée ou à la sortie du chémostat. De plus, il est bien connu que les flux continus sont souvent soumis à des fluctuations aléatoires dans le temps.

Dans cette partie, on va se concentrer sur les fluctuations aléatoires du flux d'entrée en tenant compte du fait que ce paramètre doit rester positif. Comme le volume V est constant, il est équivalent de considérer des perturbations sur le taux de dilution $D = Q/V$ plutôt que sur le flux d'entrée Q . Une option intuitive s'embble d'utiliser un processus de Wiener standard de sorte que nous remplaçons D par le terme stochastique $D + \alpha dW(\omega)$ dans le système déterministe (1.1), où $W(\omega)$ désigne une réalisation du processus de Wiener standard et $\alpha > 0$ représente l'intensité du bruit. Cependant, cette approche commune présente certains inconvénients, par exemple, la persistance des micro-organismes ne peut jamais être garantie et certaines variables d'état peuvent prendre des valeurs négatives. Par conséquent, ce processus stochastique ne semble pas être la meilleure façon de représenter les perturbations du flux d'entrée dans le modèle de chémostat, car il produit des situations irréalistes du point de vue biologique.

Une méthode plus appropriée consiste à utiliser un processus d'Ornstein-Uhlenbeck (O-U) (défini explicitement dans le chapitre 4). Ce processus est bien adapté pour modéliser les fluctuations du taux de dilution car il correspond beaucoup mieux aux situations réelles que nous souhaitons modéliser, en effet, il présente une dynamique de réversion à la moyenne, ce qui reflète le fait que les perturbations du système ont tendance à rester centrées autour d'une valeur nominale, et il possède des propriétés mathématiques fondamentales qui constitueront la base de nos développements calculatoires. On combinera le processus O-U à une fonction bornée croissante, ce qui produit des perturbations bornées et garantit que le taux de dilution reste dans des valeurs admissibles, et évite des scénarios irréalistes (débits négatifs ou trop grands). Ainsi, on va remplacer D par le terme aléatoire $D + \Phi(z^*(\theta_t W(\omega)))$, où $z^*(\theta_t W(\omega))$ désigne une réalisation du processus d'O-U approprié qui sera soigneusement introduit plus tard, et Φ est définie comme une fonction bornée par

$$\Phi(z) = \frac{2d}{\pi} \arctan(z), \quad (1.5)$$

où $d > 0$. Cette fonction est impaire, croissante et tend vers d lorsque z tend vers $+\infty$. En fait, toute fonction possédant ces propriétés peut être considérée comme une alternative valable, (pour d'autres choix possibles de fonctions, voir [2, 8]).

Ainsi, lorsque les expérimentateurs fournissent un intervalle $[D_l, D_r] \subset \mathbb{R}$, avec $0 < D_l < D < D_r < \infty$ (généralement déduit d'observations), on pose $d = \min(D_r - D, D - D_l)$. Par conséquent, le flux d'entrée perturbé reste borné à tout instant et pour toute réalisation du bruit, c'est-à-dire

$$D_l \leq D + \Phi(z^*(\theta_t W(\omega))) \leq D_r \quad \forall t \in \mathbb{R}.$$

De cette manière, le modèle aléatoire résultant est donné par le système d'équations différentielles

suivant

$$\begin{cases} \frac{ds}{dt} &= (D + \Phi(z^*(\theta_t W(\omega))))(s_{in} - s) - f(s)\frac{X}{Y} \\ \frac{dX}{dt} &= -(D + \Phi(z^*(\theta_t W(\omega))))X + f(s)X. \end{cases}$$

Dans un contexte industriel, il est fréquent d'observer de fortes concentrations du substrat en entrée s_{in} . Cependant, des concentrations trop élevées peuvent inhiber la croissance des bactéries. Afin de modéliser ce phénomène, on utilise la fonction de croissance de Haldane (1.2), qui prend en compte l'effet inhibiteur des grandes concentrations de substrat.

D'un point de vue dynamique, la croissance bactérienne dépend du taux de dilution D . Si D excède un seuil critique, le système peut évoluer vers un état de « lessivage » caractérisé par la disparition de la biomasse. Pour garantir la persistance de la biomasse, il est nécessaire que le taux de dilution respecte la condition suivante

$$D < \min(f(s_{in}), f(s_m)).$$

Cette condition couvre tous les cas possibles, notamment lorsque $s_{in} < s_m$. Dans les situations où $s_{in} > s_m$, cette condition se réduit naturellement à $D < f(s_{in})$.

Le recours au processus d'O-U comme modèle de bruit constitue une approche originale particulièrement intéressante, car elle permet de garantir l'existence d'ensembles compacts attractifs positifs, qui assure la persistance uniforme de l'espèce, c'est à dire qu'il existe un $\eta > 0$ tel que pour toute concentration initiale non nulle de la biomasse $X(0)$ chaque réalisation satisfait

$$\liminf_{t \rightarrow +\infty} X(t) \geq \eta > 0.$$

Dans ce travail, on analysera les modèles de chémostat perturbés par le processus d'O-U. On prouvera l'existence et l'unicité d'une solution globale, et on énoncera également certains résultats concernant l'existence d'ensembles compacts absorbants et attractifs. Enfin, on présentera des simulations numériques qui appuieront les résultats précédemment prouvés.

Chapitre 2

Quelques notions fondamentales dans la théorie de stabilité dans un modèle déterministe

Dans tout le chapitre, on se réfère aux annexes du livre [21]. Toutefois, on a étayé certains résultats par des démonstrations détaillées ou par des simulations programmées sous Python, voir Annexe B.

2.1 Équation différentielle ordinaire scalaire (EDO)

Soit $x(t)$ une variable réelle dépendant d'une variable réelle t , où t représente généralement le temps.

Une équation différentielle du premier ordre s'écrit sous la forme générale suivante :

$$\frac{dx}{dt} = f(t, x). \quad (2.1)$$

Si la fonction f dépend du temps t , alors l'équation différentielle est dite non autonome. Sinon, elle est dite autonome.

Dans la suite, on se penche sur des systèmes dynamiques autonomes.

Soit le système différentiel autonome défini par

$$\begin{cases} \frac{dx}{dt} = f(x) \\ x(0) = x_0. \end{cases} \quad (2.2)$$

Théorème 2.1 (Cauchy-Lipschitz pour une EDO (version autonome)).

Soit f une fonction Lipschitzienne sur un ouvert $\mathcal{U} \subset \mathbb{R}$ contenant le point x_0 , c'est-à-dire qu'il existe une constante $L > 0$ telle que pour tous x_1, x_2 dans $\mathcal{U}$,

$$|f(x_1) - f(x_2)| \leq L|x_1 - x_2|.$$

Alors il existe un intervalle $I = [-\alpha, \alpha]$ (avec $\alpha > 0$) dans lequel le système (2.2) admet une solution unique $x(\cdot)$.

Définition 2.1. Un point d'équilibre x^* du système (2.2) est un point où la vitesse s'annule, c'est-à-dire $\frac{dx}{dt} = 0$. Il vérifie donc $f(x^*) = 0$.

Le comportement des solutions autour des points d'équilibre peut être classé en différents types.

On considère un point $x(\cdot)$ voisin d'un point d'équilibre x^* . Définissons une nouvelle variable $u(t) = x(t) - x^*$. Alors on a

$$\dot{u} = \frac{du}{dt} = \frac{dx}{dt} - \frac{dx^*}{dt} = \frac{dx}{dt} = f(x) = f(x^* + u).$$

Par un développement limité d'ordre 1 au voisinage de x^* , on obtient

$$\dot{u} = f(x^* + u) = f(x^*) + f'(x^*)u + o(u^2).$$

Pour u suffisamment petit, on peut négliger le terme de reste. L'équation différentielle ordinaire devient alors

$$\dot{u} = f'(x^*)u.$$

Puis en posant $f'(x^*) = \lambda^* = \frac{df}{dx}(x^*)$, on a

$$\dot{u} = \lambda^* u,$$

dont la solution est donnée par

$$u(t) = u_0 e^{\lambda^* t}.$$

1. Si $\lambda^* < 0$, alors $u(t) \xrightarrow[t \rightarrow \infty]{} 0$ et par conséquent $x(t) \xrightarrow[t \rightarrow \infty]{} x^*$, c'est la définition de la stabilité d'un point d'équilibre.

Les solutions proches du point d'équilibre convergent vers celui-ci lorsque $t \rightarrow \infty$, le point d'équilibre x^* est dit stable (ou attractif).

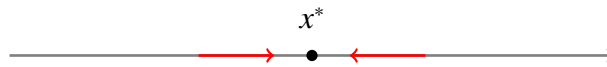


FIGURE 2.1 – Comportement local autour d'un point d'équilibre **stable**.

2. Si $\lambda^* > 0$, alors $u(t) \xrightarrow[t \rightarrow \infty]{} \pm\infty$ selon le signe de u_0 , et par conséquent $x(t)$ s'éloigne de x^* , c'est la définition de l'instabilité d'un point d'équilibre.

Les solutions proches du point d'équilibre s'en éloignent lorsque $t \rightarrow \infty$, le point d'équilibre x^* est dit instable.

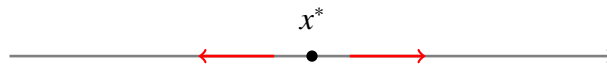


FIGURE 2.2 – Comportement local autour d'un point d'équilibre **instable**.

3. Si $\lambda^* = 0$, la linéarisation n'apporte pas d'informations. Le point d'équilibre peut être stable, instable ou encore shunt positif (respectivement shunt négatif), si la vitesse est positive (respectivement négative) à droite et à gauche du point d'équilibre.

Une autre méthode consiste à utiliser un développement de Taylor d'ordre supérieur puisque la linéarisation ne suffit pas.

On obtient

$$\dot{u} = f(x^* + u) = f(x^*) + f'(x^*)u + \frac{f''(x^*)}{2}u^2 + o(u^3).$$

Pour u suffisamment petit, l'équation différentielle ordinaire devient

$$\dot{u} = \frac{f''(x^*)}{2}u^2.$$

Le comportement des solutions dépend du signe de $f''(x^*)$.

En d'autres termes, si $f''(x^*) > 0$, alors on a un shunt positif et si $f''(x^*) < 0$, alors on a un shunt négatif, voir Figures 2.3 et 2.4.

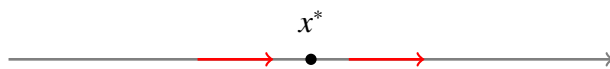


FIGURE 2.3 – Comportement local autour d'un point d'équilibre : **shunt positif**.

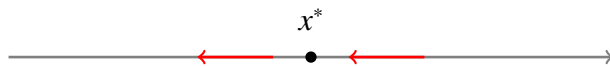


FIGURE 2.4 – Comportement local autour d'un point d'équilibre : **shunt négatif**.

Si $f''(x^*) = 0$, il faut pousser le développement de Taylor à un ordre supérieur pour déterminer le comportement.

2.2 Étude de deux équations différentielles ordinaires

La forme générale d'un système de deux équations différentielles ordinaires autonome est la suivante :

$$\begin{cases} \frac{dx}{dt} = f(x, y) \\ \frac{dy}{dt} = g(x, y). \end{cases} \quad (2.3)$$

Théorème 2.2 (Cauchy-Lipschitz pour deux EDO (version autonome)).

Soient f et g deux fonctions Lipschitziennes sur un ouvert $\mathcal{U} \subset \mathbb{R}^2$ contenant le point (x_0, y_0) , c'est-à-dire qu'il existe deux constantes L_1 et L_2 positives telles que pour tous $(x_1, y_1), (x_2, y_2)$ dans $\mathcal{U}$,

$$|f(x_1, y_1) - f(x_2, y_2)| \leq L_1 (|x_1 - x_2| + |y_1 - y_2|),$$

$$|g(x_1, y_1) - g(x_2, y_2)| \leq L_2 (|x_1 - x_2| + |y_1 - y_2|).$$

Alors, il existe un intervalle $I = [-\alpha, \alpha]$ (avec $\alpha > 0$) dans lequel le système (2.3) muni d'une condition initiale $(x(0), y(0)) = (x_0, y_0)$, admet une solution unique $(x(t), y(t))$.

Définition 2.2. Un point d'équilibre (x^*, y^*) du système (2.3) est un point où la vitesse s'annule $\left(\frac{dx}{dt}, \frac{dy}{dt}\right) = (0, 0)$. Il vérifie donc

$$\begin{cases} f(x^*, y^*) = 0 \\ g(x^*, y^*) = 0. \end{cases}$$

Définition 2.3. Le plan de phase est l'espace bidimensionnel des variables (x, y) , où chaque point représente un état instantané du système. Les trajectoires complètes dans ce plan sont appelées orbites et montrent comment un état évolue au fil du temps.

Définition 2.4. Pour l'équation différentielle $\frac{dx}{dt} = f(x)$, une isocline est une courbe dans le plan de phase où la pente $\frac{dx}{dt}$ des solutions est constante.

Pour le système dynamique (2.3), une isocline est une courbe où le rapport $\frac{dy}{dx} = \frac{g(x,y)}{f(x,y)}$ est constant. Les isoclines aident à esquisser les champs de direction des trajectoires sans résoudre explicitement le système dynamique. Elles permettent de visualiser comment les solutions évoluent dans le plan. En traçant plusieurs isoclines pour différentes constantes, on obtient une idée de la dynamique du système.

Définition 2.5. Les isoclines nulles sont les points du plan où l'une des composantes du vecteur de vitesse s'annule.

En dimension deux, on a deux isoclines,

$$\begin{aligned}\frac{dx}{dt} &= 0 \Leftrightarrow f(x,y) = 0 \quad (\text{isocline verticale}), \\ \frac{dy}{dt} &= 0 \Leftrightarrow g(x,y) = 0 \quad (\text{isocline horizontale}).\end{aligned}$$

Remarque 2.1. Le plan de phase est représenté à partir des isoclines nulles. L'intersection d'une isocline verticale et d'une isocline horizontale est un point d'équilibre.

2.3 Linéarisation au voisinage d'un point d'équilibre

Soit (x^*, y^*) un point d'équilibre du système (2.3). Considérons $(u(t), v(t))$ un point au voisinage de (x^*, y^*) tel que

$$\begin{aligned}\begin{cases} u(t) &= x(t) - x^* \\ v(t) &= y(t) - y^*. \end{cases} \\ \text{Ainsi, on a } \begin{cases} \dot{u} &= \frac{du}{dt} = \frac{dx}{dt} - \frac{dx^*}{dt} = \frac{dx}{dt} = f(x,y) = f(x^* + u, y^* + v) \\ \dot{v} &= \frac{dv}{dt} = \frac{dy}{dt} - \frac{dy^*}{dt} = \frac{dy}{dt} = g(x,y) = g(x^* + u, y^* + v). \end{cases}\end{aligned}$$

Par un développement limité d'ordre 1 au voisinage de (x^*, y^*) , on obtient

$$\begin{cases} \dot{u} &= f(x^* + u, y^* + v) = f(x^*, y^*) + u \frac{df}{dx}(x^*, y^*) + v \frac{df}{dy}(x^*, y^*) + o_1(\|(u, v)\|) \\ \dot{v} &= g(x^* + u, y^* + v) = g(x^*, y^*) + u \frac{dg}{dx}(x^*, y^*) + v \frac{dg}{dy}(x^*, y^*) + o_2(\|(u, v)\|). \end{cases}$$

Ainsi, pour des u, v suffisamment petits, on peut négliger les termes de reste, et obtenir le système linéarisé

$$\begin{cases} \dot{u} &= u \frac{df}{dx}(x^*, y^*) + v \frac{df}{dy}(x^*, y^*) \\ \dot{v} &= u \frac{dg}{dx}(x^*, y^*) + v \frac{dg}{dy}(x^*, y^*). \end{cases}$$

Ce système peut être réécrit sous la forme matricielle suivante

$$\underbrace{\begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix}}_{\dot{X}} = \underbrace{\begin{pmatrix} \frac{df}{dx}(x^*, y^*) & \frac{df}{dy}(x^*, y^*) \\ \frac{dg}{dx}(x^*, y^*) & \frac{dg}{dy}(x^*, y^*) \end{pmatrix}}_A \underbrace{\begin{pmatrix} u \\ v \end{pmatrix}}_X.$$

2.4 Solution d'un système dynamiques linéaire

On considère le système linéaire

$$\begin{cases} \dot{x} = a_{11}x + a_{12}y \\ \dot{y} = a_{21}x + a_{22}y \end{cases} \iff \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = A \begin{pmatrix} x \\ y \end{pmatrix} \quad \text{avec} \quad A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}. \quad (2.4)$$

Proposition 2.1. *Pour toute matrice $A \in \mathcal{M}_2(\mathbb{R})$, il existe une matrice réelle inversible P telle que $J = P^{-1}AP$ admet l'une des formes canoniques ci-dessous*

$$\begin{aligned} (a) & \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \quad \lambda_1 \neq \lambda_2, \quad (b) \begin{pmatrix} \lambda_0 & 0 \\ 0 & \lambda_0 \end{pmatrix} \\ (c) & \begin{pmatrix} \lambda_0 & 1 \\ 0 & \lambda_0 \end{pmatrix}, \quad (d) \begin{pmatrix} \alpha & -\beta \\ \beta & \alpha \end{pmatrix} \quad \beta > 0. \end{aligned}$$

Les scalaires $\lambda_0, \lambda_1, \lambda_2, \alpha, \beta$ dépendent des valeurs propres de A .

La matrice J est la forme de Jordan de A , et on dit que A et J sont semblables.

On donne la solution du système (2.4) pour chaque cas.

a) **Cas de deux valeurs propres réelles distinctes λ_1, λ_2 .**

$$J = P^{-1}AP = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}.$$

La matrice P est la matrice de passage formée par les vecteurs propres v_1 et v_2 (associés respectivement aux valeurs propres λ_1 et λ_2) disposés en colonnes, c'est-à-dire

$$P = (v_1 \mid v_2) = \begin{pmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{pmatrix}.$$

On effectue le changement de variables

$$\begin{pmatrix} u \\ v \end{pmatrix} = P^{-1} \begin{pmatrix} x \\ y \end{pmatrix}, \quad (2.5)$$

ceci permet de transformer le système en un système diagonal (ou de Jordan), dont la résolution est explicite.

Remarque 2.2. *Le changement de variable induit la condition initiale en $t = 0$*

$$\begin{pmatrix} u_0 \\ v_0 \end{pmatrix} = \begin{pmatrix} u(0) \\ v(0) \end{pmatrix} = P^{-1} \begin{pmatrix} x(0) \\ y(0) \end{pmatrix}.$$

La dérivation de (2.5) donne

$$\frac{d}{dt} \begin{pmatrix} u \\ v \end{pmatrix} = P^{-1} \frac{d}{dt} \begin{pmatrix} x \\ y \end{pmatrix} = P^{-1} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = P^{-1} A \begin{pmatrix} x \\ y \end{pmatrix} = P^{-1} A P \begin{pmatrix} u \\ v \end{pmatrix} = J \begin{pmatrix} u \\ v \end{pmatrix},$$

donc le système se ramène à

$$\begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}.$$

Ce système admet pour solution

$$\begin{cases} u(t) = u_0 e^{\lambda_1 t} \\ v(t) = v_0 e^{\lambda_2 t}, \end{cases} \quad (2.6)$$

ce qui implique

$$\begin{pmatrix} x \\ y \end{pmatrix} = P \begin{pmatrix} u \\ v \end{pmatrix} = P \begin{pmatrix} u_0 e^{\lambda_1 t} \\ v_0 e^{\lambda_2 t} \end{pmatrix} = \begin{pmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{pmatrix} \begin{pmatrix} u_0 e^{\lambda_1 t} \\ v_0 e^{\lambda_2 t} \end{pmatrix}.$$

On aboutit à la solution suivante

$$\begin{cases} x(t) = v_{11}u_0 e^{\lambda_1 t} + v_{12}v_0 e^{\lambda_2 t} \\ y(t) = v_{21}u_0 e^{\lambda_1 t} + v_{22}v_0 e^{\lambda_2 t} \end{cases} \iff \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = u_0 \begin{pmatrix} v_{11} \\ v_{21} \end{pmatrix} e^{\lambda_1 t} + v_0 \begin{pmatrix} v_{12} \\ v_{22} \end{pmatrix} e^{\lambda_2 t}.$$

b) Cas d'une valeur propre réelle double $\lambda_1 = \lambda_2 = \lambda$.

Lorsque la matrice A admet une valeur propre double λ , sa forme de Jordan dépend de la dimension du sous-espace propre associé. Deux cas se présentent

i.) Matrice A diagonalisable

Si A a deux vecteurs propres linéairement indépendants (le sous-espace propre est de dimension deux), alors A est diagonalisable. Sa forme de Jordan est simplement une matrice diagonale

$$J = P^{-1}AP = \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix} \quad \text{avec} \quad P = (v_1 \mid v_2) = \begin{pmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{pmatrix},$$

où v_1 et v_2 sont les deux vecteurs propres associés à λ .

Dans ce cas, la méthode de résolution du système différentiel (2.4) est analogue au cas à valeurs propres distinctes, avec la particularité que $\lambda_1 = \lambda_2$.

On aboutit à la solution suivante

$$\begin{cases} x(t) = (v_{11}u_0 + v_{12}v_0)e^{\lambda t} \\ y(t) = (v_{21}u_0 + v_{22}v_0)e^{\lambda t} \end{cases} \iff \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = u_0 \begin{pmatrix} v_{11} \\ v_{21} \end{pmatrix} e^{\lambda t} + v_0 \begin{pmatrix} v_{12} \\ v_{22} \end{pmatrix} e^{\lambda t}.$$

ii.) Matrice A non diagonalisable

Si A n'a qu'un seul vecteur propre (sous-espace propre de dimension 1), elle n'est pas diagonalisable. Sa forme de Jordan comprend un bloc de Jordan avec un 1 sur la sur-diagonale,

$$J = P^{-1}AP = \begin{pmatrix} \lambda & 1 \\ 0 & \lambda \end{pmatrix}.$$

La matrice de passage P est définie par

$$P = \begin{pmatrix} v_1 & m_1 \\ v_2 & m_2 \end{pmatrix},$$

où $\begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$ est un vecteur propre associé à λ , et $\begin{pmatrix} m_1 \\ m_2 \end{pmatrix}$ est un vecteur linéairement indépendant de $\begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$.

En suivant une démarche analogue à celle du cas des valeurs propres distinctes, on aboutit au système suivant

$$\begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} \lambda & 1 \\ 0 & \lambda \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}.$$

Il s'ensuit que

$$\begin{cases} \dot{u}(t) = \lambda u(t) + v(t) \\ \dot{v}(t) = \lambda v(t) \end{cases} \iff \begin{cases} \dot{u}(t) = \lambda u(t) + v_0 e^{\lambda t} \\ v(t) = v_0 e^{\lambda t} \end{cases} \iff \begin{cases} \dot{u}(t) - \lambda u(t) = v_0 e^{\lambda t} \\ v(t) = v_0 e^{\lambda t} \end{cases}.$$

On pose $z(t) = u(t)e^{-\lambda t}$ alors

$$\dot{z}(t) = \dot{u}(t)e^{-\lambda t} - u(t)\lambda e^{-\lambda t} = e^{-\lambda t}(\dot{u}(t) - \lambda u(t)) = e^{-\lambda t}v_0 e^{\lambda t} = v_0.$$

On en déduit

$$z(t) = v_0 t + u_0 \quad \text{et} \quad u(t) = e^{\lambda t} (v_0 t + u_0).$$

La solution dans la base initiale s'écrit alors

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = P \begin{pmatrix} u(t) \\ v(t) \end{pmatrix} = \begin{pmatrix} v_1 & m_1 \\ v_2 & m_2 \end{pmatrix} \begin{pmatrix} (v_0 t + u_0)e^{\lambda t} \\ v_0 e^{\lambda t} \end{pmatrix}.$$

Ce qui conduit au système

$$\begin{cases} x(t) = v_1(v_0 t + u_0)e^{\lambda t} + m_1 v_0 e^{\lambda t} \\ y(t) = v_2(v_0 t + u_0)e^{\lambda t} + m_2 v_0 e^{\lambda t}. \end{cases}$$

iii.) **Cas de deux valeurs propres complexes conjuguées $\lambda_{1,2} = \alpha \pm i\beta$.**

Dans ce cas, les vecteurs propres associés aux valeurs propres sont également complexes conjugués et s'écrivent sous la forme $v = a \pm ib$, où $a = (a_1, a_2)$ et $b = (b_1, b_2)$ sont des vecteurs réels de $\mathbb{R}^2$. La matrice A se décompose alors sous la forme de Jordan réelle suivante

$$J = P^{-1}AP = \begin{pmatrix} \alpha & -\beta \\ \beta & \alpha \end{pmatrix}.$$

La matrice P est construite à partir des vecteurs b et a disposés en colonnes. Plus précisément

$$P = (b \mid a) = \begin{pmatrix} b_1 & a_1 \\ b_2 & a_2 \end{pmatrix}.$$

En suivant une démarche analogue à celle du cas des valeurs propres réelles distinctes, on obtient le système différentiel suivant

$$\begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} \alpha & -\beta \\ \beta & \alpha \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} \iff \begin{cases} \dot{u} = \alpha u - \beta v \\ \dot{v} = \beta u + \alpha v. \end{cases}$$

On passe aux coordonnées polaires

$$\begin{cases} u = r \cos \theta \\ v = r \sin \theta \end{cases} \iff \begin{cases} r = \sqrt{u^2 + v^2} \\ \tan \theta = \frac{v}{u} \end{cases} \iff \begin{cases} r^2 = u^2 + v^2 \\ \tan \theta = \frac{v}{u}. \end{cases}$$

On calcule les dérivées et on obtient

$$\begin{aligned} 2r\dot{r} = 2u\dot{u} + 2v\dot{v} &\iff \dot{r} = \frac{u\dot{u} + v\dot{v}}{r} \\ &= \frac{u(\alpha u - \beta v) + v(\beta u + \alpha v)}{r} \\ &= \frac{\alpha(u^2 + v^2)}{r} \\ &= \frac{\alpha r^2}{r} \\ &= \alpha r. \end{aligned}$$

$$\begin{aligned}
\left(\frac{1}{\cos^2 \theta}\right) \dot{\theta} &= \frac{\dot{v}u - \dot{u}v}{u^2} \iff \dot{\theta} = \frac{(\beta u + \alpha v)u - (\alpha u - \beta v)v}{u^2} \cos^2 \theta \\
&= \frac{(\beta u + \alpha v)u - (\alpha u - \beta v)v}{u^2} \cos^2 \theta \\
&= \frac{\beta(u^2 + v^2)}{u^2} \cos^2 \theta = \frac{\beta r^2}{u^2} \cos^2 \theta \\
&= \frac{\beta r^2}{r^2 \cos^2 \theta} \cos^2 \theta \\
&= \beta.
\end{aligned}$$

Ainsi, en notant

$$\begin{pmatrix} r_0 \\ \theta_0 \end{pmatrix} = \begin{pmatrix} r(0) \\ \theta(0) \end{pmatrix} = \begin{pmatrix} \sqrt{u^2(0) + v^2(0)} \\ \arctan\left(\frac{v(0)}{u(0)}\right) \end{pmatrix},$$

on obtient

$$\begin{cases} \dot{r} = \alpha r \\ \dot{\theta} = \beta \end{cases} \implies \begin{cases} r(t) = r_0 e^{\alpha t} \\ \theta(t) = \beta t + \theta_0 \end{cases} \implies \begin{cases} u(t) = r_0 e^{\alpha t} \cos(\beta t + \theta_0) \\ v(t) = r_0 e^{\alpha t} \sin(\beta t + \theta_0). \end{cases} \quad (2.7)$$

La solution dans la base initiale s'écrit alors

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = P \begin{pmatrix} u(t) \\ v(t) \end{pmatrix} = \begin{pmatrix} b_1 & a_1 \\ b_2 & a_2 \end{pmatrix} \begin{pmatrix} r_0 e^{\alpha t} \cos(\beta t + \theta_0) \\ r_0 e^{\alpha t} \sin(\beta t + \theta_0) \end{pmatrix}.$$

Ce qui conduit au système

$$\begin{cases} x(t) = r_0 e^{\alpha t} (b_1 \cos(\beta t + \theta_0) + a_1 \sin(\beta t + \theta_0)) \\ y(t) = r_0 e^{\alpha t} (b_2 \cos(\beta t + \theta_0) + a_2 \sin(\beta t + \theta_0)). \end{cases}$$

2.5 Typologie des systèmes planaires linéaires

Soit le système

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \underbrace{\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}}_A \begin{pmatrix} x \\ y \end{pmatrix}.$$

On se limite au cas où la matrice A est inversible ($\det(A) \neq 0$). Dans ce cas, l'unique solution de $AX = 0$ est $X = 0$, et le système admet un seul point d'équilibre à l'origine du plan de phase.

1. Cas de deux valeurs propres réelles distinctes $\lambda_1 \neq \lambda_2$.

La matrice A se réduit à sa forme de Jordan

$$J = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix},$$

et la solution dans la base de Jordan est donnée alors par (2.6).

Ainsi, l'allure des trajectoires va dépendre du signe de λ_1, λ_2 . Trois cas peuvent être distingués.

1) $\lambda_1 > 0, \lambda_2 > 0$,

il s'agit d'un noeud instable (ou source). Toute trajectoire s'éloigne du point d'équilibre $(0, 0)$ quand $t \rightarrow +\infty$.

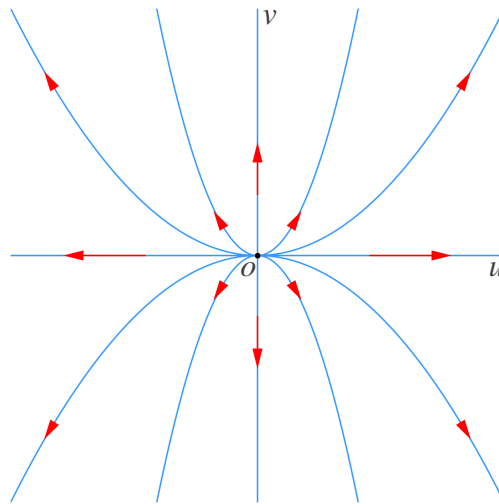


FIGURE 2.5 – L'allure locale d'un **noeud instable**, $\lambda_1 > 0$, $\lambda_2 > 0$.

- 2) $\lambda_1 < 0$, $\lambda_2 < 0$,
il s'agit d'un noeud stable (ou puits). Toute trajectoire tend vers le point d'équilibre c'est-à-dire quand $t \rightarrow +\infty$, les solutions $(u(t), v(t)) \rightarrow (0, 0)$.

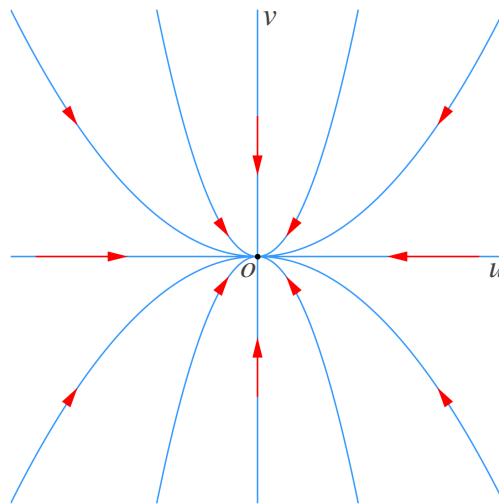


FIGURE 2.6 – L'allure locale d'un **noeud stable**, $\lambda_1 < 0$, $\lambda_2 < 0$.

- 3) λ_1 et λ_2 sont de signes contraires,
il s'agit d'un point selle (ou point col). Un point selle est toujours instable, donc la trajectoire s'éloigne du point d'équilibre $(0, 0)$. Autrement dit, quand $t \rightarrow +\infty$, une des deux composantes $u(t)$, $v(t)$ tend vers $\pm\infty$ et l'autre vers 0, voir Figure 2.7.

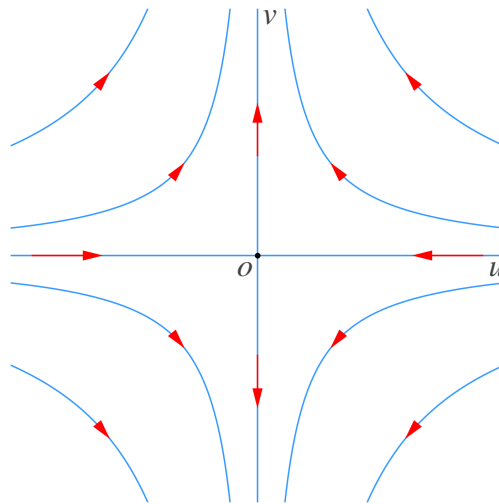


FIGURE 2.7 – L'allure locale d'un **point selle**, $\lambda_1 > 0$, $\lambda_2 < 0$.

2. Cas d'une valeur propre réelle double λ .

Dans le cas où le sous-espace propre est de dimension deux, la matrice A se réduit à la forme de Jordan suivante

$$J = \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}.$$

Deux cas possibles peuvent être distingués.

- 1) $\lambda > 0$, il s'agit d'une étoile instable.

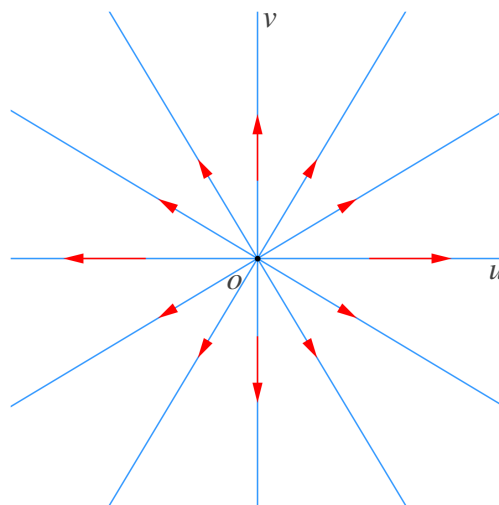


FIGURE 2.8 – L'allure locale d'une **étoile instable**, $\lambda > 0$.

- 2) $\lambda < 0$, il s'agit d'une étoile stable.

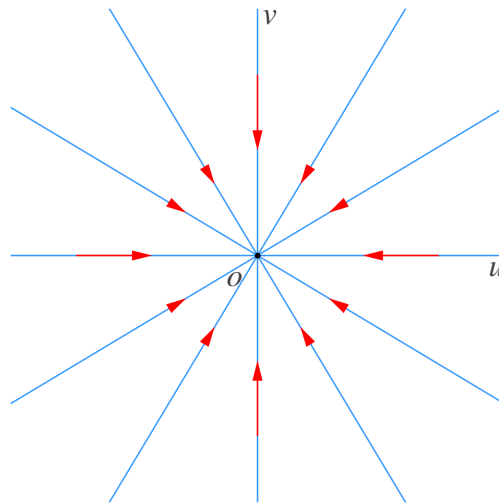


FIGURE 2.9 – L'allure locale d'une **étoile stable**, $\lambda < 0$.

Dans le cas où le sous-espace propre est de dimension un, la matrice A se réduit à la forme de Jordan suivante

$$J = \begin{pmatrix} \lambda & 1 \\ 0 & \lambda \end{pmatrix}.$$

Deux cas possibles peuvent être distingués.

- 1) $\lambda > 0$, il s'agit d'un noeud dégénéré instable.

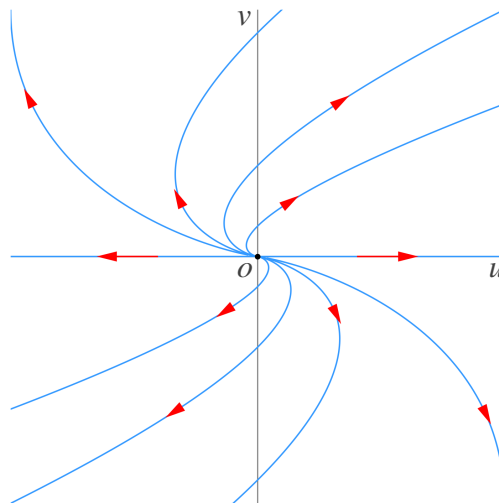


FIGURE 2.10 – L'allure locale d'un **noeud dégénéré instable**, $\lambda > 0$.

- 2) $\lambda < 0$, il s'agit d'un noeud dégénéré stable.

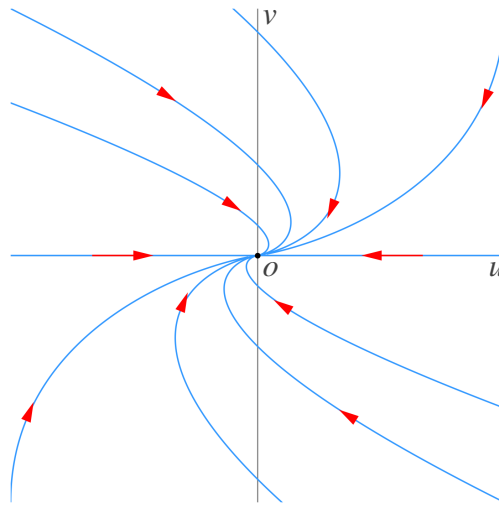


FIGURE 2.11 – L'allure locale d'un **noeud dégénéré stable**, $\lambda < 0$.

3. Cas de deux valeurs propres complexes conjuguées $\lambda_{1,2} = \alpha \pm i\beta$.

La matrice A se réduit à sa forme de Jordan

$$J = \begin{pmatrix} \alpha & -\beta \\ \beta & \alpha \end{pmatrix}.$$

et la solution dans la base de Jordan est donnée alors par (2.7).

Ainsi, l'allure des trajectoires correspond à des spirales qui s'éloignent ou se rapprochent de $(0, 0)$ selon le signe de α (sauf pour $\alpha = 0$), et dont le sens de rotation est donné par le signe de β . Les portraits de phase possibles sont les suivants

- 1) $\alpha > 0$, il s'agit d'un foyer instable.

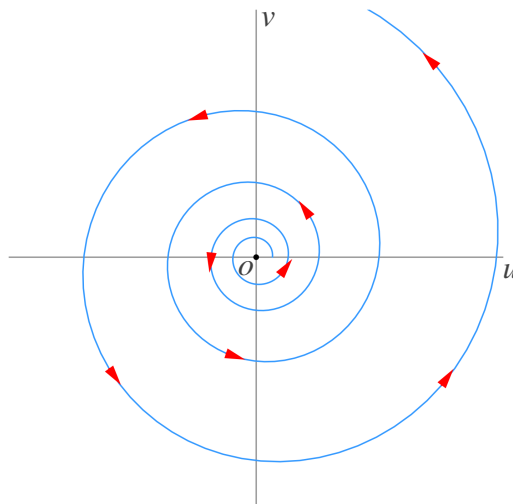


FIGURE 2.12 – L'allure locale d'un **foyer instable**, $\alpha > 0$, $\beta > 0$.

- 2) $\alpha < 0$, il s'agit d'un foyer stable.

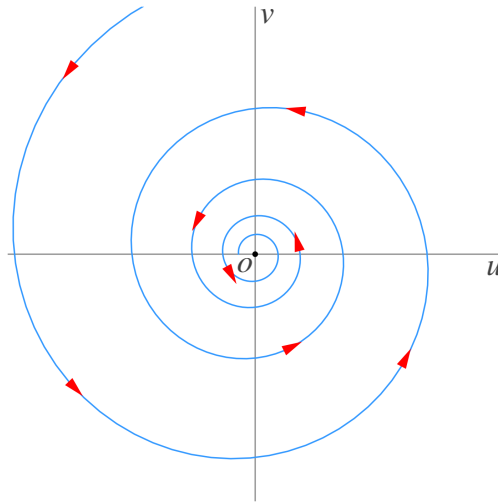


FIGURE 2.13 – L'allure locale d'un **foyer stable**, $\alpha < 0, \beta < 0$

3) $\alpha = 0$, il s'agit d'un point centre.

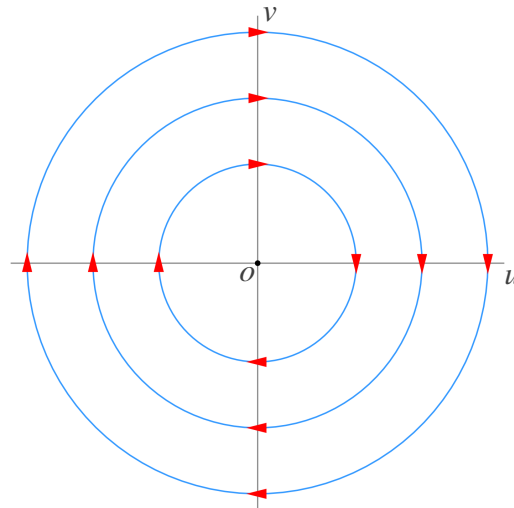


FIGURE 2.14 – L'allure locale d'un **centre**, $\alpha = 0, \beta > 0$.

2.6 Typologie des solutions des systèmes linéaires dans le plan

La classification des solutions des systèmes linéaires planaires, basée sur la nature des valeurs propres de la matrice du système, peut également être visualisée dans le plan (trace, déterminant).

Les valeurs propres λ_1, λ_2 de la matrice A sont solutions de l'équation caractéristique

$$\lambda^2 - \text{tr}(A)\lambda + \det(A) = 0 \quad \text{avec} \quad \begin{cases} \text{tr}(A) &= \lambda_1 + \lambda_2 \\ \det(A) &= \lambda_1 \lambda_2. \end{cases}$$

La nature des valeurs propres dépend du signe du discriminant $\Delta = (\text{tr}(A))^2 - 4\det(A)$. Dans le plan (tr, det) (voir Figure 2.15), la condition $\Delta = 0$ définit une parabole d'équation

$$\det(A) = \frac{1}{4}(\text{tr}(A))^2,$$

passant par l'origine. Cette parabole divise le plan en deux régions clés

- **Au-dessus** ($\Delta < 0$) : On retrouve les foyers et les centres.

- **En-dessous** ($\Delta > 0$) : On retrouve les noeuds et les points-selle.

1. Cas $\Delta = 0$

On a une valeur propre double ($\lambda_1 = \lambda_2 = \lambda_0$), on obtient les relations suivantes

$$\det(A) = \lambda_0^2 > 0 \quad \text{et} \quad \text{tr}(A) = 2\lambda_0.$$

La nature du portrait de phase dépend alors du signe de la trace

- Si $\text{tr}(A) > 0$ ($\lambda_0 > 0$) alors l'origine est une étoile instable ou un noeud dégénéré instable.
- Si $\text{tr}(A) < 0$ ($\lambda_0 < 0$) alors l'origine est une étoile stable ou un noeud dégénéré stable.

2. Cas $\Delta > 0$

On a deux valeurs propres réelles distinctes, correspondant à la région en-dessous de la parabole et on obtient les relations suivantes

- Si $\det(A) < 0$ (λ_1 et λ_2 sont de signes opposés) alors l'origine est un point selle.
- Si $\det(A) > 0$ et $\text{tr}(A) > 0$ ($\lambda_1, \lambda_2 > 0$) alors l'origine est un noeud instable.
- Si $\det(A) > 0$ et $\text{tr}(A) < 0$ ($\lambda_1, \lambda_2 < 0$) alors l'origine est un noeud stable.

3. Cas $\Delta < 0$

On a deux valeurs propres complexes conjuguées ($\lambda_{1,2} = \alpha \pm i\beta$), qui correspondent à la zone située au-dessus de la parabole. On a alors les relations suivantes

$$\det(A) = \alpha^2 + \beta^2 > 0 \quad \text{et} \quad \text{tr}(A) = 2\alpha.$$

Cette région se subdivise en trois cas

- Si $\text{tr}(A) < 0$ ($\alpha < 0$) alors l'origine est un foyer stable.
- Si $\text{tr}(A) > 0$ ($\alpha > 0$) alors l'origine est un foyer instable.
- Si $\text{tr}(A) = 0$ ($\alpha = 0$) alors l'origine est un centre.

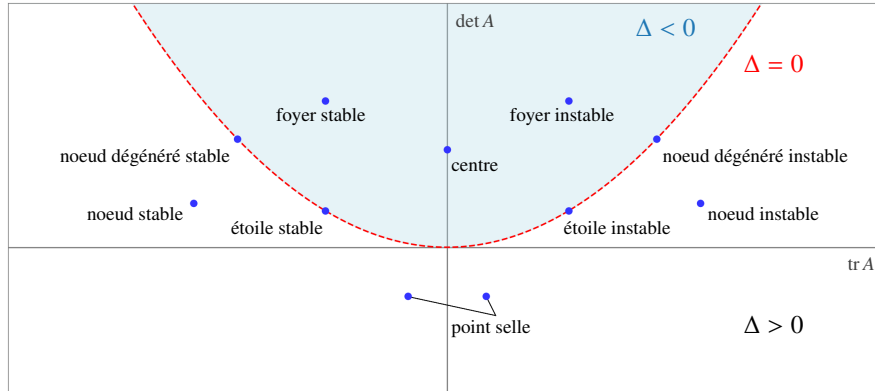


FIGURE 2.15 – Classification des portraits de phase du système $\dot{X} = AX$ en fonction de la trace et du déterminant.

2.7 Stabilité asymptotique et stabilité exponentielle.

Soit $E^* = (x^*, y^*)$ un point d'équilibre du système (2.3) de condition initial $X_0 = (x_0, y_0)$. Sa linéarisation conduit au système

$$\begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = J_{E^*} \begin{pmatrix} u \\ v \end{pmatrix},$$

où J_{E^*} désigne la matrice jacobienne du système évaluée en E^* .

Définition 2.6. L'équilibre E^* est dit stable si la solution $X(t)$ du système vérifiant $X(0) = X_0$ (assez proche de E^*), satisfait

$$\forall \varepsilon > 0, \exists \delta > 0 \text{ tel que } \|X_0 - E^*\| < \delta \implies \|X(t) - E^*\| < \varepsilon, \text{ pour tout } t \geq 0.$$

Définition 2.7. L'équilibre E^* est dit localement asymptotiquement stable (LAS) s'il est stable et

$$\exists \delta > 0, \|X_0 - E^*\| < \delta \implies \lim_{t \rightarrow +\infty} X(t) = E^*.$$

Définition 2.8. L'équilibre E^* est dit localement exponentiellement stable (LES) s'il existe des constantes C et μ strictement positives, telles que pour toute solution $X(t)$ vérifiant $X(0) = X_0$ (assez proche de E^*), on ait

$$\|X(t) - E^*\| \leq Ce^{-\mu t} \|X_0 - E^*\| \text{ pour tout } t \geq 0.$$

Remarque 2.3. Un équilibre est localement asymptotiquement stable si

- Stabilité : toute trajectoire qui commence suffisamment près de l'équilibre y reste proche.
- Attractivité : toute trajectoire qui commence suffisamment près de l'équilibre converge vers celui-ci quand $t \rightarrow \infty$.

Mais on ne précise pas la vitesse de convergence. Cette vitesse peut-être lente, rapide, variable ...

Un équilibre est localement exponentiellement stable si

- Il est asymptotiquement stable (donc stable et attractif).
- Et en plus, il existe des constantes positives C et μ (avec $\mu = \min_i |\operatorname{Re}(\lambda_i)|$ et λ_i les valeurs propres du système) telles que

$$\|X(t)\| \leq C\|X(0)\|e^{-\mu t}$$

pour tout $X(0)$ assez proche de l'équilibre.

La vitesse de convergence est alors bornée de manière exponentielle (c'est plus fort!).

Chapitre 3

Etude du modèle mathématique déterministe du chimostat simple

Ce chapitre repose sur le livre [21] et sur la thèse [36] pour l'étude du modèle déterministe du chimostat.

3.1 Présentation du modèle déterministe

Nous considérons un chimostat de volume constant total V . On note respectivement par s et X les concentrations de substrat et de biomasse. Le flux est constant et noté Q . La concentration de substrat en entrée est désignée par s_{in} . L'évolution par rapport au temps des concentrations de substrat et de biomasse est modélisée par le système d'équations différentielles ordinaires suivant :

$$\begin{cases} \frac{ds}{dt} = D(s_{in} - s) - f(s)\frac{X}{Y} \\ \frac{dX}{dt} = -DX + f(s)X, \end{cases} \quad (3.1)$$

où Y représente le taux de conversion du substrat en biomasse et $f(\cdot)$ est la fonction de croissance de la biomasse vérifiant l'hypothèse suivante :

Hypothèse 3.1. La fonction f est de classe C^1 , issue de 0 ($f(0) = 0$) et $\lim_{s \rightarrow +\infty} f(s) = 0$. De plus, il existe $s_m \in]0, +\infty[$ tel que

$$\forall s \in]0, +\infty[- \{s_m\}, \quad f'(s)(s_m - s) > 0.$$

D'autre part, D le taux de dilution est défini par $D := Q/V$. Sans perte de généralité, on suppose que $Y = 1$ dans les équations (3.1) en utilisant le changement de variable $x = X/Y$ et en remplaçant dans les équations, il s'ensuit

$$\begin{cases} \frac{ds}{dt} = D(s_{in} - s) - f(s)x \\ \frac{dx}{dt} = \frac{1}{Y}(-D + f(s))Yx. \end{cases}$$

Ainsi, le système (3.1) devient

$$\begin{cases} \frac{ds}{dt} = D(s_{in} - s) - f(s)x \\ \frac{dx}{dt} = -Dx + f(s)x, \end{cases} \quad (3.2)$$

et est représenté dans la Figure 3.1.

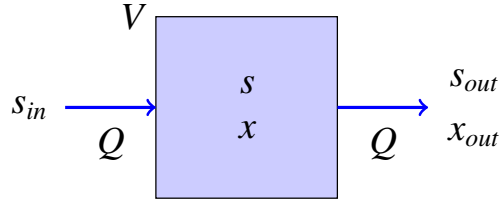


FIGURE 3.1 – Configuration d'un chémostat.

Dans la Figure 3.1, s_{out} et x_{out} représentent respectivement la concentration de substrat et de biomasse sortant du réservoir.

Remarque 3.1. *L'Hypothèse 3.1 est satisfaite par une fonction de type Haldane définie par (1.2).*

Pour ce type de fonction, lorsque $D < f(s_m)$, l'équation $f(s) = D$ admet deux solutions $\lambda_1(D)$ et $\lambda_2(D)$, telles que $\lambda_1 < \lambda_2$, comme illustré dans la Figure 3.2.

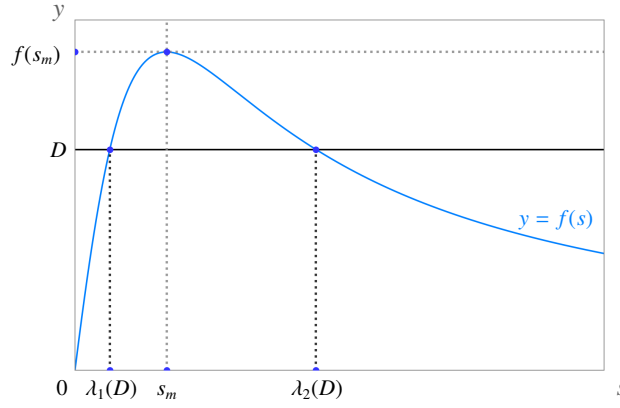


FIGURE 3.2 – Solution de l'équation $f(s) = D$ avec f une fonction de type Haldane.

Définition 3.1. *Dans un cadre général, le seuil de croissance "break-even concentration" $\lambda : D \in [0, m := f(s_m)[\mapsto \lambda(D) \in [0, +\infty[$ est défini par*

$$\lambda(D) := \begin{cases} \sup \{s \in [0, s_m], f(s) < D\} \\ \min \{s \in [0, s_m], f(s) < D\}. \end{cases}$$

Le seuil de croissance d'une fonction vérifiant l'Hypothèse 3.1 est défini par

$$\lambda(D) := f^{-1}(D) = \begin{cases} \lambda_1(D) & \text{si } 0 \leq D \leq s_m \\ \lambda_2(D) & \text{si } s_m \leq D < m. \end{cases} \quad (3.3)$$

où m et s_m sont donnés dans (1.3) et (1.4).

Lemme 3.1. *Les solutions $(s(t), x(t))$ du système (3.2) avec une condition initiale positive existent pour tout $t \geq 0$, sont uniques, positives, bornées et $\lim_{t \rightarrow +\infty} (s(t) + x(t)) = s_{in}$.*

Démonstration.

Sous l'Hypothèse 3.1, la fonction f est de classe C^1 (c'est-à-dire qu'elle admet des dérivées continues). Par conséquent, le second membre du système (3.2) est de classe C^1 , car il est composé de

produits et de sommes de fonctions C^1 . Ainsi, le second membre est une fonction localement lipschitzienne et par suite le Théorème d'existence et d'unicité de solutions 2.2 s'applique. Par conséquent, il existe une unique solution locale pour toute condition initiale donnée.

Isoclines verticales :

$$\frac{ds}{dt} = 0 \iff D(s_{in} - s) - f(s)x = 0 \implies D(s_{in} - s) = f(s)x.$$

Isoclines horizontales :

$$\frac{dx}{dt} = 0 \iff -Dx + f(s)x = 0 \implies x = 0 \text{ ou } s = \lambda_1(D) \text{ ou } s = \lambda_2(D).$$

Lorsque $s = 0$, on a $\frac{ds}{dt} = Ds_{in}$ qui est strictement positif. Cela implique qu'aucune trajectoire ne peut quitter le quadrant positif $\mathbb{R}_+^2$ en traversant l'axe horizontal $x = 0$ qui est lui-même une trajectoire du système. En vertu de l'unicité des solutions, deux trajectoires ne peuvent pas se croiser. Par conséquent, aucune trajectoire ne peut quitter le quadrant positif en traversant l'axe horizontal $x = 0$. Ainsi, pour tout $t \geq 0$, les solutions $(s(t), x(t))$ restent dans le quadrant positif, c'est-à-dire $(s(t), x(t)) \geq (0, 0)$.

Considérons la fonction $z(t) = s(t) + x(t)$. En dérivant $z(t)$ par rapport au temps, on obtient :

$$\frac{dz}{dt} = \frac{ds}{dt} + \frac{dx}{dt} = D(s_{in} - s) - f(s)x + (-Dx + f(s)x) = D(s_{in} - z).$$

On résout cette équation par séparation des variables

$$\begin{aligned} \frac{dz}{s_{in} - z} &= D dt \\ \iff \int_{z(0)}^{z(t)} \frac{dz}{s_{in} - z} &= \int_0^t D dt \\ \iff [-\ln |s_{in} - z|]_{z(0)}^{z(t)} &= Dt \\ \iff -\ln(s_{in} - z(t)) + \ln(s_{in} - z(0)) &= Dt \quad (s_{in} > z) \\ \iff \ln\left(\frac{s_{in} - z(0)}{s_{in} - z(t)}\right) &= Dt \\ \iff \frac{s_{in} - z(0)}{s_{in} - z(t)} &= e^{Dt} \\ \iff s_{in} - z(t) &= (s_{in} - z(0))e^{-Dt} \\ \iff z(t) &= s_{in} + (z(0) - s_{in})e^{-Dt}. \end{aligned}$$

Finalement, on obtient la solution

$$z(t) = s_{in} + (s(0) + x(0) - s_{in})e^{-Dt}.$$

Ainsi, lorsque $t \rightarrow +\infty$, le terme e^{-Dt} tend vers 0, et on a

$$\lim_{t \rightarrow +\infty} z(t) = \lim_{t \rightarrow +\infty} (s(t) + x(t)) = s_{in}.$$

Par ailleurs, puisque $s(t) \geq 0$ et $x(t) \geq 0$ pour tout $t \geq 0$, il en découle que $z(t) \geq 0$ pour tout $t \geq 0$. De plus, comme $z(t)$ est une fonction bornée (car elle converge vers s_{in}), les fonctions $s(t)$ et $x(t)$ sont également bornées. Cela garantit que les solutions ne divergent pas et restent dans un domaine borné pour tout $t \geq 0$. $\square$

3.2 Classification des équilibres

Le point (s^*, x^*) est un équilibre du système (3.2) au sens de la définition 2.2, c'est-à-dire qu'il vérifie

$$\begin{cases} D(s_{in} - s^*) - f(s^*)x^* &= 0 \\ -Dx^* + f(s^*)x^* &= 0. \end{cases} \quad (3.4)$$

Dans l'étude des systèmes différentiels, la notion d'équilibre joue un rôle central : il s'agit d'un état stationnaire où le système n'évolue pas dans le temps, offrant ainsi un point de référence crucial pour analyser la stabilité et le comportement global des solutions.

Théorème 3.1. *Supposons que l'Hypothèse 3.1 est vérifiée. Les équilibres de (3.2) sont :*

1. *L'équilibre de lessivage $e_0 = (s_{in}, 0)$ qui existe toujours.
Il est LES si et seulement si $D > f(s_{in})$.*
2. *Les équilibres de persistance de l'espèce :*

- *L'équilibre $e_1^1 = (s_1^{1*}, x_1^{1*})$ où*

$$s_1^{1*} := \lambda_1(D) \text{ et } x_1^{1*} := s_{in} - \lambda_1(D). \quad (3.5)$$

Cet équilibre existe et il est strictement positif si et seulement si $\lambda_1(D) < s_{in}$. Il est LES dès qu'il existe et qu'il est strictement positif.

- *L'équilibre $e_1^2 = (s_1^{2*}, x_1^{2*})$ où*

$$s_1^{2*} := \lambda_2(D) \text{ et } x_1^{2*} := s_{in} - \lambda_2(D). \quad (3.6)$$

Cet équilibre existe et il est strictement positif si et seulement si $\lambda_2(D) < s_{in}$. Il est instable dès qu'il existe.

Démonstration.

1) Équilibres du système

Les points d'équilibre du système correspondent aux intersections des isoclines préalablement établies.

◦ *Isocline horizontale*

$$\begin{aligned} \frac{dx}{dt} = 0 &\implies x(-D + f(s)) = 0 \\ &\implies x = 0 \quad \text{ou} \quad f(s) = D \\ &\implies x = 0 \quad \text{ou} \quad s = \lambda_1(D) \quad \text{ou} \quad s = \lambda_2(D). \end{aligned}$$

a) Si $x = 0$ (lessivage), alors

◦ *Isocline verticale*

$$\frac{ds}{dt} = D(s_{in} - s) = 0 \implies s = s_{in}.$$

Ceci donne l'équilibre de lessivage $e_0 = (s_{in}, 0)$.

b) Si $s = \lambda_1(D)$, alors

◦ *Isocline verticale*

$$\frac{ds}{dt} = D(s_{in} - \lambda_1(D)) - Dx = 0 \implies x = s_{in} - \lambda_1(D).$$

Ce qui donne l'équilibre $e_1^1 = (\lambda_1(D), s_{in} - \lambda_1(D))$ qui existe si et seulement si $s_{in} > \lambda_1(D)$ (la biomasse est toujours positive).

- c) Si $s = \lambda_2(D)$, alors
 ◦ *Isocline verticale*

$$\frac{ds}{dt} = D(s_{in} - \lambda_2(D)) - Dx = 0 \implies x = s_{in} - \lambda_2(D).$$

Ce qui donne l'équilibre $e_1^2 = (\lambda_2(D), s_{in} - \lambda_2(D))$ qui existe si et seulement si $s_{in} > \lambda_2(D)$.

2) Stabilité des équilibres

La matrice jacobienne du système est

$$J_{(s,x)} = \begin{pmatrix} -D - f'(s)x & -f(s) \\ f'(s)x & -D + f(s) \end{pmatrix}. \quad (3.7)$$

- a) Analyse de e_0

$$J_{e_0} = \begin{pmatrix} -D & -f(s_{in}) \\ 0 & -D + f(s_{in}) \end{pmatrix}.$$

Les valeurs propres sont

$$\begin{aligned} \mu_1 &= -D < 0 \\ \mu_2 &= -D + f(s_{in}). \end{aligned}$$

Ainsi

- e_0 est LES (noeud stable) si et seulement si $D > f(s_{in})$,
 (c'est-à-dire $\lambda_1(D) > s_{in}$ ou $\lambda_2(D) < s_{in}$).
- e_0 est instable (point selle) si et seulement si $D \leq f(s_{in})$,
 (c'est-à-dire $\lambda_1(D) \leq s_{in}$ ou $\lambda_2(D) \geq s_{in}$).

- b) Analyse de e_1^1

$$J_{e_1^1} = \begin{pmatrix} -D - f'(\lambda_1(D))(s_{in} - \lambda_1(D)) & -D \\ f'(\lambda_1(D))(s_{in} - \lambda_1(D)) & 0 \end{pmatrix}.$$

Le déterminant et la trace de la matrice jacobienne sont

$$\begin{aligned} \det(J_{e_1^1}) &= Df'(\lambda_1(D))(s_{in} - \lambda_1(D)) > 0 \\ \text{tr}(J_{e_1^1}) &= -D - f'(\lambda_1(D))(s_{in} - \lambda_1(D)) < 0. \end{aligned}$$

Donc e_1^1 est toujours LES (noeud stable) lorsqu'il existe (c'est-à-dire lorsque $s_{in} > \lambda_1(D)$).

- c) Analyse de e_1^2

$$J_{e_1^2} = \begin{pmatrix} -D - f'(\lambda_2(D))(s_{in} - \lambda_2(D)) & -D \\ f'(\lambda_2(D))(s_{in} - \lambda_2(D)) & 0 \end{pmatrix}.$$

Le déterminant est

$$\det(J_{e_1^2}) = Df'(\lambda_2(D))(s_{in} - \lambda_2(D)).$$

Sachant que par hypothèse $f'(\lambda_2(D)) < 0$ et $s_{in} > \lambda_2(D)$, le déterminant est négatif. Par conséquent, e_1^2 est un équilibre instable (point selle) dès qu'il existe (c'est-à-dire si $s_{in} > \lambda_2(D)$).

□

Le théorème précédent se résume comme suit. Notons φ la courbe définie par

$$\begin{aligned} \varphi &:= \{ (s_{in}, D) : D = f(s_{in}) \} \\ &:= \{ (s_{in}, D) : \lambda_1(D) = s_{in} \text{ si } s_{in} \leq s_m \text{ et } \lambda_2(D) = s_{in} \text{ si } s_{in} \geq s_m \}. \end{aligned} \quad (3.8)$$

Cette courbe sépare le plan opératoire en trois régions, voir Figure 3.3, désignées par I_0 , I_1 et I_2 , qui sont définies de la manière suivante.

$$\begin{aligned}
I_0 &= \{ (s_{in}, D) : 0 < s_{in} \leq \lambda_1(D) \}, \\
I_1 &= \{ (s_{in}, D) : \lambda_1(D) < s_{in} \leq \lambda_2(D) \}, \\
I_2 &= \{ (s_{in}, D) : \lambda_2(D) \leq s_{in} \}.
\end{aligned} \tag{3.9}$$

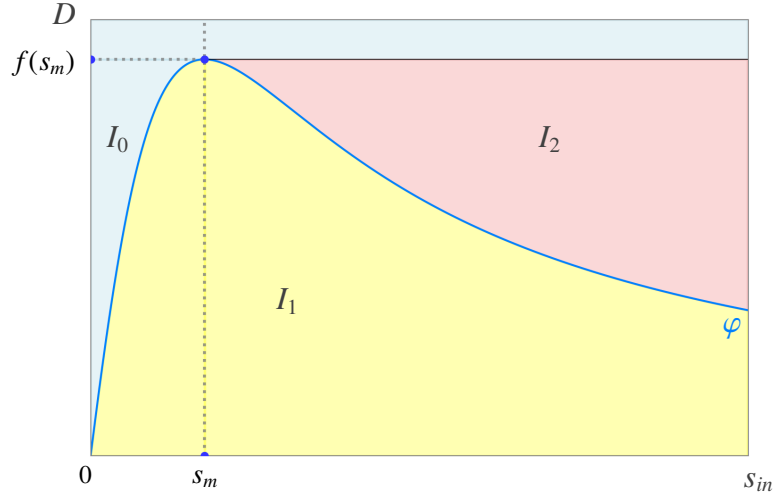


FIGURE 3.3 – Diagramme opératoire d'un chémostat avec une fonction de croissance non-monotone.

Le comportement du système (3.2) dans chaque région est décrit dans le Tableau 3.1.

Régions	I_0	I_1	I_2
e_0	<i>LES</i>	<i>I</i>	<i>LES</i>
e_1	<i>N'existe pas</i>	<i>LES</i>	<i>LES</i>
e_2	<i>N'existe pas</i>	<i>N'existe pas</i>	<i>I</i>

TABLE 3.1 – Stabilité des équilibres dans les différentes régions du diagramme opératoire.

En résumé, quand $s_{in} < \lambda_1(D) < \lambda_2(D)$, pour toute condition initiale positive, toute solution du système (3.2) converge vers l'équilibre de lessivage $e_0 = (s_{in}, 0)$ et on se situe dans la région bleue de la Figure 3.3 et a) de la Figure 3.4.

Si $\lambda_1(D) < s_{in} < \lambda_2(D)$, pour toute condition initiale positive, toute solution du système (3.2) converge vers l'équilibre positif $e_1^1 = (\lambda_1(D), s_{in} - \lambda_1(D))$ et on se situe dans la région jaune de la Figure 3.3 et dans le cas b) de la Figure 3.4.

Enfin, si $\lambda_1(D) < \lambda_2(D) < s_{in}$, pour toute condition initiale positive, toute solution du système (3.2) peut converger soit vers l'équilibre positif $e_1^1 = (\lambda_1(D), s_{in} - \lambda_1(D))$ ou vers l'équilibre de lessivage $e_0 = (s_{in}, 0)$ et on se situe dans la région rouge de la Figure 3.3 et dans le cas c) de la Figure 3.4. C'est une région dite de bistabilité. La convergence vers l'un des deux équilibres dépend si la condition initiale se situe dans le bassin d'attraction de e_0 ou de e_1 .

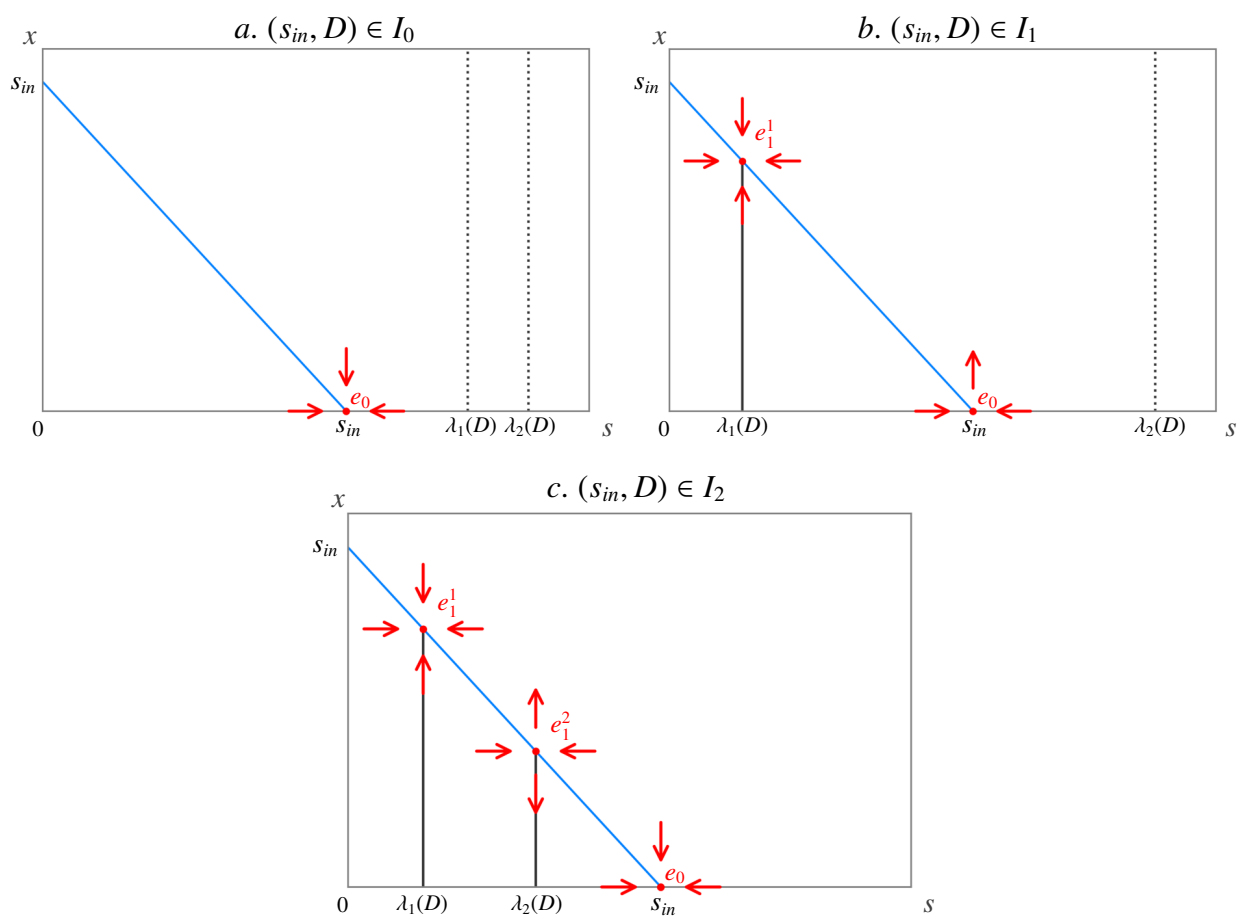


FIGURE 3.4 – Stabilité des équilibres.

Chapitre 4

Étude du modèle stochastique du chémostat simple

Ce chapitre s'appuie principalement sur les travaux de Rapaport et al. [5] concernant le modèle du chémostat avec croissance non monotone sous perturbations aléatoires du taux de dilution. Nous avons clarifié certaines démonstrations et proposé des simulations numériques en Python afin d'illustrer les dynamiques du modèle, voir Annexe B.

Dans ce chapitre, on enrichit le modèle déterministe en y intégrant des perturbations stochastiques adéquates, afin de capturer les effets des fluctuations sur la dynamique du chémostat. On étudiera comment ces perturbations influencent la stabilité du système, la persistance des populations microbiennes et les éventuels états d'équilibre. Cette approche stochastique permet de mieux comprendre le comportement du chémostat dans un environnement réaliste où l'aléatoire joue un rôle significatif.

Il existe plusieurs processus stochastiques pour modéliser des systèmes dynamiques soumis à des fluctuations aléatoires (voir [22, 35]). Parmi eux, le processus d'Ornstein-Uhlenbeck (O-U) se distingue particulièrement par sa capacité à décrire des systèmes présentant des fluctuations aléatoires qui correspondent mieux aux situations réelles qu'on souhaite modéliser et possède des propriétés mathématiques fondamentales particulièrement utiles tel la propriété de réversion à la moyenne qui est un concept statistique désignant la tendance d'une variable à revenir vers sa moyenne au fil du temps. Dans le contexte des systèmes dynamiques, cela signifie que lorsque la variable s'écarte de sa moyenne, il est probable qu'elle finisse par revenir vers cette moyenne. La réversion à la moyenne implique donc une sorte d'équilibre naturel où les valeurs extrêmes sont souvent suivies de valeurs plus normales.

Contrairement au mouvement brownien standard, qui est non borné et peut générer des fluctuations arbitrairement grandes, le processus O-U, combiné à une fonction bornée telle que Φ définie dans l'introduction, voir (1.5), permet de saturer les fluctuations dans un intervalle fixé, ce qui est plus fidèle aux contraintes physiques et biologiques du système. Cette modélisation garantit que les taux de dilution perturbés restent dans des bornes réalistes, évitant ainsi des valeurs non viables ou biologiquement absurdes. C'est donc un choix judicieux pour analyser la dynamique du chémostat en présence de perturbations aléatoires.

Dans ce chapitre, on se place sur un espace de probabilité $(\Omega, \mathcal{F}, \mathbb{P})$.

4.1 Processus d'O-U

Définition 4.1. *L'équation de Langevin est définie par*

$$dX_t = -\beta X_t dt + \nu dW_t, \quad t \geq 0. \quad (4.1)$$

où $(W_t)_{t \geq 0}$ est un mouvement brownien standard, $\beta > 0$ est appelé taux de retour à la moyenne, qui représente la force avec laquelle le processus est attiré par la moyenne, et $\nu > 0$ est la volatilité qui représente la variation ou la taille du bruit indépendamment de la quantité de bruit.

Proposition 4.1. *L'unique solution générale de l'équation de Langevin (4.1) issue de X_0 est le processus d'O-U, défini par*

$$X_t = X_0 e^{-\beta t} + \nu \int_0^t e^{-\beta(t-s)} dW_s. \quad (4.2)$$

Démonstration.

L'équation de Langevin (4.1) est une EDS linéaire. Pour la résoudre on utilise la méthode du facteur intégrant avec un traitement stochastique. On a l'équation

$$dX_t + \beta X_t dt = \nu dW_t.$$

On multiplie l'équation par le facteur intégrant $\rho(t) = e^{\beta t}$, on obtient

$$e^{\beta t} dX_t + \beta e^{\beta t} X_t dt = \nu e^{\beta t} dW_t.$$

On remarque que le membre de gauche est la différentielle de $e^{\beta t} X_t$. En effet, pour $f(t, x) = e^{\beta t} x$, le lemme d'Itô A.1 donne

$$\begin{aligned} df(t, X_t) = d(e^{\beta t} X_t) &= \frac{\partial f}{\partial x}(t, X_t) dX_t + \frac{\partial f}{\partial t}(t, X_t) dt + \frac{1}{2} \frac{\partial^2 f}{\partial x^2}(t, X_t) d\langle X \rangle_t, \\ &= e^{\beta t} dX_t + \beta e^{\beta t} X_t dt + 0 d\langle X \rangle_t, \\ &= e^{\beta t} dX_t + \beta e^{\beta t} X_t dt. \end{aligned}$$

Donc on a

$$d(e^{\beta t} X_t) = \nu e^{\beta t} dW_t.$$

Par intégration des deux membres de l'équation entre 0 et t , on obtient

$$e^{\beta t} X_t - X_0 = \nu \int_0^t e^{\beta s} dW_s.$$

D'où la solution

$$X_t = X_0 e^{-\beta t} + \nu \int_0^t e^{-\beta(t-s)} dW_s.$$

□

4.1.1 Solution stationnaire

La solution (4.2), dans laquelle le système est supposé démarrer à l'instant $t = 0$, peut être interprétée comme une version tronquée de la solution stationnaire qui a évolué depuis un passé infini. On s'intéresse ici au cas où le système est en évolution depuis "toujours". Il s'agit d'effectuer une extension naturelle du domaine temporel de l'EDS. L'équation est désormais définie pour tout $t \in \mathbb{R}$.

Pour introduire rigoureusement la notion de solution stationnaire, commençons par rappeler la définition suivante :

Définition 4.2. (voir [33])

Soient $(W_t^+)_{t \geq 0}$ et $(W_t^-)_{t \geq 0}$ deux mouvements browniens standards indépendants. Le mouvement brownien bilatéral (M.B.B) $(W_t)_{t \in \mathbb{R}}$ est défini comme suit

$$W_t = \begin{cases} W_t^+ & \text{si } t \geq 0 \\ W_{-t}^- & \text{si } t < 0. \end{cases}$$

Remarque 4.1. Le M.B.B est le prolongement par continuité (naturel) du mouvement brownien standard à $\mathbb{R}$. Il vérifie aussi les mêmes propriétés que le mouvement brownien, c'est à dire

- La trajectoire $t \mapsto W_t$ est presque sûrement continue, et l'on a $W_0 = 0$ presque sûrement.
- Pour tout $t, s \in \mathbb{R}$, tel que $s \leq t$, $W_t - W_s \sim \mathcal{N}(0, |t - s|)$.
- Pour tout $n \geq 1$, et pour tout $t_0 \leq t_1 \leq \dots \leq t_n$, $(W_{t_1} - W_{t_0}, \dots, W_{t_n} - W_{t_{n-1}})$ est un vecteur de variables aléatoires gaussiennes indépendantes.

Reformulation du M.B.B dans un espace fonctionnel

On considère $\Omega^0 = C(\mathbb{R}, \mathbb{R})$. Chaque élément $\omega \in \Omega^0$ représente une trajectoire possible du M.B.B, et pour $t \in \mathbb{R}$, e_t est l'application d'évaluation en t , définie comme suit

$$\begin{aligned} e_t : \Omega_0 &\longrightarrow \mathbb{R} \\ \omega &\longmapsto e_t(\omega) = \omega(t). \end{aligned}$$

On munit l'espace Ω_0 de la tribu $\mathcal{F}^0$ définie au moyen de la tribu borélienne $\mathcal{B}(\mathbb{R})$ par

$$\mathcal{F}^0 = \bigcap_{t \in \mathbb{R}} e_t^{-1}(\mathcal{B}(\mathbb{R})).$$

Autrement dit, $\mathcal{F}^0$ est la tribu engendrée par les cylindres. On définit un cylindre comme suit. Pour tout $n \geq 1$, $t_1 < \dots < t_n \in \mathbb{R}$, $A_1, \dots, A_n \in \mathcal{B}(\mathbb{R})$,

$$C_{t_1, \dots, t_n; A_1, \dots, A_n} = \{\omega \in C(\mathbb{R}, \mathbb{R}) : \forall k \in \{1, \dots, n\}, e_{t_k}(\omega) \in A_k\}.$$

On peut réécrire le M.B.B $(W_t)_t$ dans l'espace canonique $(\Omega^0, \mathcal{F}^0, \mathbb{P}_W)$ en s'appuyant sur sa précédente formulation dans l'espace $(\Omega, \mathcal{F}, \mathbb{P})$ grâce à la mesure de Wiener. La mesure de Wiener $\mathbb{P}_W$ est définie sur les cylindres :

$$\mathbb{P}_W(C_{t_1, \dots, t_n; A_1, \dots, A_n}) := \mathbb{P}(\forall k \in \{1, \dots, n\}, W_{t_k} \in A_k).$$

On introduit le processus canonique W (M.B.B) défini sur l'espace canonique $(\Omega^0, \mathcal{F}^0, \mathbb{P}_W)$ par

$$\begin{aligned} W : \mathbb{R} \times \Omega_0 &\longrightarrow \mathbb{R} \\ (t, \omega) &\longmapsto W_t(\omega) = \omega(t). \end{aligned}$$

Remarque 4.2. Lorsque la dépendance en ω n'est pas importante, il est de tradition de ne pas la noter. Dans la suite on pourra noter si nécessaire, simplement ω une réalisation (trajectoire) ou élément de Ω_0 pour un ω fixé (élément de Ω).

Reprenons la solution classique de l'équation de Langevin

$$X_t(\omega) = X_0 e^{-\beta t} + \nu \int_0^t e^{-\beta(t-s)} d\omega(s).$$

Pour bien définir la solution stationnaire, il est nécessaire que la partie aléatoire (le terme intégral) converge en loi quand $t \rightarrow +\infty$, et que le terme $X_0 e^{-\beta t}$ tende vers zéro, sans quoi il compromettrait la stationnarité. Dans le cas du processus d'O-U, ces conditions sont vérifiées. En effet,

$$\mathbb{E}\left(\nu \int_0^t e^{-\beta(t-s)} d\omega(s)\right) = 0.$$

$$\text{Var}\left(\nu \int_0^t e^{-\beta(t-s)} d\omega(s)\right) = \mathbb{E}\left(\nu \int_0^t e^{-\beta(t-s)} d\omega(s)\right)^2 \stackrel{\text{isométrie}}{=} \nu^2 \int_0^t e^{-2\beta(t-s)} ds = \nu^2 \frac{1 - e^{-2\beta t}}{2\beta} \xrightarrow{t \rightarrow \infty} \frac{\nu^2}{2\beta}.$$

Ainsi

$$\lim_{t \rightarrow +\infty} X_t \sim \mathcal{N}\left(0, \frac{\nu^2}{2\beta}\right).$$

Au lieu d'initialiser le processus à l'instant $t = 0$, on suppose que le système était actif depuis un passé infini, c'est-à-dire que l'équation de Langevin devient

$$dX_t = -\beta X_t dt + \nu d\omega(t), \quad t \in \mathbb{R},$$

où ω est une réalisation du M.B.B défini dans 4.2.

La solution classique (4.2) révèle que plus t augmente, plus le passé est « oublié ». Pour que le système soit stationnaire dès l'instant t , il est donc nécessaire de supposer une exposition au bruit depuis un passé infini. La solution stationnaire doit par conséquent s'exprimer sous la forme

$$X_t(\omega) = \nu \int_{-\infty}^t e^{-\beta(t-s)} d\omega(s).$$

4.1.2 Changement de variable et flux de Wiener

En effectuant le changement de variable $u = s - t$, l'intégrale précédente peut s'écrire

$$X_t(\omega) = \nu \int_{-\infty}^0 e^{\beta u} d\omega(u + t). \quad (4.3)$$

On introduit la définition du flux de décalage de Wiener.

Définition 4.3. *Le flux de décalage de Wiener θ_t est l'opération qui décale les trajectoires browniennes d'un temps $t \geq 0$, défini par*

$$\theta_t : \omega(u) \mapsto (\theta_t \omega)_u = \omega(u + t) - \omega(t).$$

La différentielle stochastique de $(\theta_t \omega)_u$ par rapport à u s'écrit :

$$d(\theta_t \omega)_u = d\omega(u + t) \quad (\text{car } \omega(t) \text{ est constant en } u).$$

Ainsi, l'intégrale (4.3) devient

$$X_t(\omega) = \nu \int_{-\infty}^0 e^{\beta u} d(\theta_t \omega)_u.$$

Par une intégration par parties on a alors

$$\int_{-\infty}^0 e^{\beta u} d(\theta_t \omega)_u = \underbrace{\left[e^{\beta u} (\theta_t \omega)_u \right]_{-\infty}^0}_{=0 \text{ (p.s)}} - \beta \int_{-\infty}^0 e^{\beta u} (\theta_t \omega)_u du.$$

On en déduit que (4.3) se réécrit

$$X_t(\omega) = -\beta \nu \int_{-\infty}^0 e^{\beta u} (\theta_t \omega)_u du. \quad (4.4)$$

Cette représentation montre la structure de réversion à la moyenne du processus d'O-U ainsi défini et son écriture en fonction du flux de Wiener. Ce développement conduit à la définition opérationnelle d'un processus d'O-U équivalente à la définition donnée dans (4.2) et qui sera utilisée dans la suite.

Définition 4.4. Pour les détails sur le processus d'O-U, on peut se référer aux sources [6, 4, 7].

Pour $\beta, \nu > 0$, le processus d'O-U est défini au moyen d'un M.B.B par l'équation suivante

$$z(t, \omega) := z^*(\theta_t \omega) = -\beta \nu \int_{-\infty}^0 e^{\beta s} (\omega(t + s) - \omega(t)) ds = -\beta \nu \int_{-\infty}^0 e^{\beta s} (\theta_t \omega)_s ds, \quad t \in \mathbb{R}, \quad \omega \in \Omega. \quad (4.5)$$

où :

- $z(t, \omega)$ est la valeur du processus au temps t pour une réalisation particulière.
- ω est un M.B.B (voir 4.2) défini dans un espace probabilisé $(\Omega, \mathcal{F}, \mathbb{P})$.
- β est la constante de réversion à la moyenne, qui détermine la force avec laquelle le processus est attiré vers sa moyenne à long terme. ν est la constante de volatilité, qui mesure l'amplitude des fluctuations aléatoires.
- θ_t est le flux de décalage de Wiener (défini en 4.3), qui décrit comment le processus de Wiener est décalé dans le temps.

Ce processus satisfait l'équation de Langevin

$$dX_t = -\beta X_t dt + \nu d\omega(t) \quad , \quad t \in \mathbb{R}. \quad (4.6)$$

Typiquement, le processus d'O-U (4.5) peut modéliser la position d'une particule en prenant en compte sa friction dans un fluide (ce qui est la principale différence avec le processus de Wiener standard). En effet, il peut être considéré comme une généralisation du processus de Wiener standard et établit un lien entre le processus de Wiener standard ($\beta = 0$, $\nu = 1$) et l'absence totale de bruit ($\beta = 1$, $\nu = 0$).

Proposition 4.2.

Le processus d'O-U (4.5) est un processus Gaussien, stationnaire, de loi $\mathcal{N}\left(0, \frac{\nu^2}{2\beta}\right)$, et de covariance

$$\mathbb{E}[z^*(\theta_t \omega) z^*(\theta_s \omega)] = \frac{\nu^2}{2\beta} e^{-\beta|t-s|}.$$

Démonstration.

Grâce à la définition du M.B.B, on note que si $r_1 \leq r_2 \leq 0$, on a

$$\begin{aligned} \mathbb{E}[(\omega(t+r_1) - \omega(t))(\omega(t+r_2) - \omega(t))] &= \mathbb{E}[(\omega(t+r_1) - \omega(t+r_2))(\omega(t+r_2) - \omega(t))] \\ &\quad + \mathbb{E}[(\omega(t+r_2) - \omega(t))^2] \\ &= |r_1| \wedge |r_2| = |r_2|. \end{aligned}$$

On en déduit

$$\begin{aligned} \mathbb{E}[z^*(\theta_t \omega)^2] &= \mathbb{E}\left[\beta^2 \nu^2 \int_{-\infty}^0 e^{\beta r_1} (\omega(t+r_1) - \omega(t)) dr_1 \int_{-\infty}^0 e^{\beta r_2} (\omega(t+r_2) - \omega(t)) dr_2\right] \\ &= \beta^2 \nu^2 \int_{-\infty}^0 \int_{-\infty}^0 e^{\beta(r_1+r_2)} \mathbb{E}[(\omega(t+r_1) - \omega(t))(\omega(t+r_2) - \omega(t))] dr_1 dr_2 \\ &= \beta^2 \nu^2 \int_{-\infty}^0 \int_{-\infty}^0 e^{\beta(r_1+r_2)} |r_1| \wedge |r_2| dr_1 dr_2 \\ &= 2\beta^2 \nu^2 \int_0^{+\infty} \int_0^{r_2} r_1 e^{-\beta(r_1+r_2)} dr_1 dr_2 \\ &= \nu^2 \int_0^{+\infty} r_1 e^{-\beta(r_1+r_2)} dr_1 \\ &= \nu^2 \int_0^{+\infty} 2\beta r_1 e^{-2\beta r_1} dr_1 = \frac{\nu^2}{2\beta}. \end{aligned}$$

De plus, pour $s \leq t$, si $s \leq t+r_1$ on a

$$\mathbb{E}[(\omega(t+r_1) - \omega(t))(\omega(s+r_2) - \omega(s))] = 0,$$

et si $s+r_2 \leq t+r_1 \leq s \leq t$ on obtient

$$\begin{aligned} \mathbb{E}[(\omega(t+r_1) - \omega(t))(\omega(s+r_2) - \omega(s))] &= \mathbb{E}[(\omega(t+r_1) - \omega(s)) + (\omega(s) - \omega(t))] \\ &\quad ((\omega(s+r_2) - \omega(t+r_1)) + (\omega(t+r_1) - \omega(s))) \\ &= |t+r_1-s| = |r_1+(t-s)|. \end{aligned}$$

Enfin si $t + r_1 \leq s + r_2 \leq s \leq t$, de la même manière on a

$$\mathbb{E}[(\omega(t + r_1) - \omega(t))(\omega(s + r_2) - \omega(s))] = |r_2|.$$

Ainsi, pour $s \leq t$ la covariance du processus d'O-U s'écrit

$$\begin{aligned} \mathbb{E}[z^*(\theta_t \omega) z^*(\theta_s \omega)] &= \mathbb{E} \left[v^2 \beta^2 \int_{-\infty}^0 e^{\beta r_1} (\omega(t + r_1) - \omega(t)) dr_1 \int_{-\infty}^0 e^{\beta r_2} (\omega(s + r_2) - \omega(s)) dr_2 \right] \\ &= v^2 \beta^2 \int_{-\infty}^0 \int_{-\infty}^0 e^{\beta(r_1 + r_2)} \mathbb{E}[(\omega(t + r_1) - \omega(t))(\omega(s + r_2) - \omega(s))] dr_1 dr_2 \\ &= v^2 \beta^2 \int_{-\infty}^0 \int_{-\infty}^{r_2 - (t-s)} e^{\beta(r_1 + r_2)} |r_2| dr_1 dr_2 + v^2 \beta^2 \int_{-\infty}^0 \int_{r_2 - (t-s)}^{- (t-s)} e^{\beta(r_1 + r_2)} | - r_1 - (t - s) | dr_1 dr_2 \\ &= v^2 \beta^2 \int_0^{+\infty} \int_{r_2 + (t-s)}^{+\infty} e^{-\beta(r_1 + r_2)} r_2 dr_1 dr_2 + v^2 \beta^2 \int_0^{+\infty} \int_{(t-s)}^{r_2 + (t-s)} e^{-\beta(r_1 + r_2)} (r_1 - (t - s)) dr_1 dr_2 \\ &= v^2 \beta \int_0^{+\infty} r_2 e^{-\beta(2r_2 + t-s)} dr_2 + v^2 \beta^2 \int_0^{+\infty} -\frac{r_2}{\beta} e^{-\beta(2r_2 + t-s)} - \frac{1}{\beta^2} e^{-\beta(2r_2 + t-s)} + \frac{1}{\beta^2} e^{-\beta(r_2 + t-s)} dr_2 \\ &= -\frac{v^2}{4\beta} e^{-\beta(t-s)} + \frac{v^2}{4} e^{-\beta(t-s)} - \frac{v^2}{2\beta} e^{-\beta(t-s)} + \frac{v^2}{\beta} e^{-\beta(t-s)} \\ &= \frac{v^2}{2\beta} e^{-\beta(t-s)}. \end{aligned}$$

Donc, pour $s, t \in \mathbb{R}$ on a

$$\mathbb{E}[z^*(\theta_t \omega) z^*(\theta_s \omega)] = \frac{v^2}{2\beta} e^{-\beta|t-s|}.$$

□

Introduisons à présent la notion de processus stochastique tempéré. Cette définition permet de restreindre notre attention à des trajectoires dont la croissance, à l'infini, reste suffisamment modérée pour garantir l'existence de certaines propriétés.

Définition 4.5. Un processus stochastique $Y = (Y_t)_t$ est dit tempéré si pour tout $\varepsilon > 0$ on a

$$\lim_{|t| \rightarrow \infty} e^{-\varepsilon|t|} |Y_t| = 0.$$

Proposition 4.3. Un processus stochastique Y est dit tempéré si et seulement si

$$\lim_{|t| \rightarrow +\infty} \frac{\ln_+ |Y_t|}{|t|} \rightarrow 0,$$

avec $\ln_+(x) := \max(0, \ln(x))$.

Démonstration. Supposons que

$$\forall \varepsilon > 0, \quad \lim_{|t| \rightarrow \infty} e^{-\varepsilon|t|} |Y_t| = 0.$$

Par hypothèse, on a

$$\forall \varepsilon > 0, \quad \forall \delta > 0, \quad \exists T > 0 \quad \text{tel que} \quad \forall |t| > T, \quad e^{-\varepsilon|t|} |Y_t| < \delta.$$

On pose $\varepsilon = \delta$, on obtient

$$\forall \delta > 0, \quad \exists T > 0 \quad \text{tel que} \quad \forall |t| > T, \quad e^{-\delta|t|} |Y_t| < \delta.$$

Donc

$$|Y_t| < \delta e^{\delta|t|}.$$

On applique le logarithme positif aux deux membres

$$\ln_+ |Y_t| < \delta |t| \ln_+(\delta).$$

Donc

$$\frac{\ln_+ |Y_t|}{|t|} < \delta \ln_+(\delta).$$

Mais cela est vrai pour tout $\delta > 0$, d'où

$$0 \leq \lim_{|t| \rightarrow \infty} \frac{\ln_+ |Y_t|}{|t|} \leq \lim_{\delta \rightarrow 0^+} \delta \ln_+(\delta) = 0.$$

Ce qui implique que

$$\lim_{|t| \rightarrow \infty} \frac{\ln_+ |Y_t|}{|t|} = 0.$$

Supposons maintenant que

$$\lim_{|t| \rightarrow \infty} \frac{\ln_+ |Y_t|}{|t|} = 0$$

Soit $\varepsilon > 0$. Alors il existe $T > 0$ tel que pour tout $|t| > T$

$$\frac{\ln_+ |Y_t|}{|t|} < \frac{\varepsilon}{2} \implies |Y_t| \leq e^{\frac{\varepsilon}{2}|t|}.$$

Et par suite il en découle

$$e^{-\varepsilon|t|} |Y_t| \leq e^{-\varepsilon|t|} e^{\frac{\varepsilon}{2}|t|} = e^{-\frac{\varepsilon}{2}|t|} \xrightarrow{|t| \rightarrow \infty} 0.$$

□

On considère dans la suite la constante de réversion à la moyenne β et la constante de volatilité ν comme données, ces dernières étant généralement estimées à partir d'observations. On note $z^*(\theta, \omega)$ le processus d'Ornstein-Uhlenbeck défini précédemment. On en rappelle certaines propriétés dans la proposition suivante.

On introduit au préalable le lemme et les théorèmes suivants, dont on fera usage par la suite.

Lemme 4.1 (Lemme de Fernique [1]). *Soit X une variable aléatoire gaussienne réelle centrée sur un espace de probabilité $(\Omega, \mathcal{F}, \mathbb{P})$. Alors, il existe une constante $\alpha > 0$ telle que*

$$\mathbb{E} \left[\exp(\alpha X^2) \right] < \infty.$$

Autrement dit, X admet un moment exponentiel quadratique fini.

Théorème 4.1 (Théorème ergodique de Birkhoff en temps continu [24]).

Soit $(\Omega, \mathcal{F}, \mathbb{P})$ un espace de probabilité, et soit $(\theta_t)_{t \in \mathbb{R}}$ un flux mesurable préservant la mesure, c'est-à-dire que pour tout $t \in \mathbb{R}$, θ_t est mesurable et vérifie :

$$\mathbb{P} \circ \theta_t^{-1} = \mathbb{P}.$$

On suppose de plus que le flux est ergodique, c'est-à-dire que toute fonction $f \in L^1(\mathbb{P})$ invariante (i.e., $f \circ \theta_t = f$ $\mathbb{P}$ -p.s. pour tout t) est constante $\mathbb{P}$ -p.s.

Alors, pour toute fonction $f \in L^1(\Omega, \mathcal{F}, \mathbb{P})$, on a $\mathbb{P}$ -p.s :

$$\lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T f(\theta_s \omega) ds = \mathbb{E}[f].$$

Proposition 4.4. (voir [1, 9]), Il existe un ensemble $\widetilde{\Omega} \in \mathcal{F}$, θ_t -invariant, de mesure 1 et tel que : pour $\omega \in \widetilde{\Omega}$ et $\beta, \nu > 0$, nous avons

- (i) La variable aléatoire $|z^*(\omega)|$ est tempérée par rapport à $\{\theta_t\}_{t \in \mathbb{R}}$ (voir 4.5), c'est-à-dire qu'il existe $\kappa > 0$ tel que presque sûrement,

$$\lim_{|t| \rightarrow \infty} e^{-\kappa|t|} |z^*(\theta_t \omega)| = 0.$$

- (ii) Cette application est une solution stationnaire de (4.6) avec des trajectoires continues

$$(t, \omega) \mapsto z^*(\theta_t \omega) = -\beta \nu \int_{-\infty}^0 e^{\beta s} (\theta_t \omega)_s \, ds.$$

- (iii) Pour tout $\omega \in \widetilde{\Omega}$ on a

$$\lim_{t \rightarrow \pm\infty} \frac{|z^*(\theta_t \omega)|}{t} = 0 \quad , \quad \lim_{t \rightarrow \pm\infty} \frac{1}{t} \int_0^t z^*(\theta_s \omega) \, ds = 0,$$

$$\lim_{t \rightarrow \pm\infty} \frac{1}{t} \int_0^t |z^*(\theta_s \omega)| \, ds = \mathbb{E}[|z^*|] < \infty.$$

Démonstration.

1. On va montrer que le processus $z^*(\theta_t \omega)$ est tempéré au sens suivant

$$\exists \kappa > 0, \quad \lim_{|t| \rightarrow \infty} e^{-\kappa|t|} |z^*(\theta_t \omega)| = 0.$$

Pour cela, on utilise la Proposition 4.3, or, $z^*(\theta_t \omega)$ est un processus stationnaire gaussien de moyenne nulle et de variance constante $\sigma^2 = \nu^2/2\beta$. En particulier, pour tout $t \in \mathbb{R}$, $z^*(\theta_t \omega) \sim \mathcal{N}(0, \sigma^2)$. Par le lemme de Fernique (voir 4.1), il existe une constante $a > 0$ telle que

$$\mathbb{E} \left[\exp \left(a |z^*(\theta_t \omega)|^2 \right) \right] < \infty, \quad \text{pour tout } t \in \mathbb{R}.$$

Soit $\varepsilon > 0$, et définissons les événements

$$A_n := \{\omega : |z^*(\theta_n \omega)| > e^{\varepsilon n}\}.$$

Alors, par inégalité de Markov et Fernique, il existe une constante $C > 0$ telle que

$$\mathbb{P}(A_n) = \mathbb{P}(|z^*(\theta_n \omega)| > e^{\varepsilon n}) \leq \mathbb{E} \left[e^{a|z^*(\theta_n \omega)|^2} \right] \cdot e^{-ae^{2\varepsilon n}} \leq C e^{-ae^{2\varepsilon n}}.$$

La série $\sum_{n=1}^{\infty} \mathbb{P}(A_n)$ converge donc. D'après le lemme de Borel–Cantelli (voir A.2), il vient

$$\mathbb{P}(|z^*(\theta_n \omega)| > e^{\varepsilon n} \text{ pour une infinité de } n) = 0.$$

Autrement dit, presque sûrement, il existe $N(\omega)$ tel que pour tout $n \geq N(\omega)$, $|z^*(\theta_n \omega)| \leq e^{\varepsilon n}$, donc

$$\frac{\ln^+ |z^*(\theta_n \omega)|}{n} \leq \varepsilon, \quad \text{pour } n \gg 1.$$

Comme $\varepsilon > 0$ est arbitraire, on en déduit

$$\lim_{n \rightarrow \infty} \frac{\ln^+ |z^*(\theta_n \omega)|}{n} = 0 \quad \text{presque sûrement.}$$

Enfin, pour passer de $n \in \mathbb{N}$ à $t \in \mathbb{R}$, on utilise la continuité presque sûre des trajectoires de $z^*(\theta_t \omega)$: pour tout $t \in \mathbb{R}$, il existe un entier n tel que $|t - n| \leq 1$, donc

$$|z^*(\theta_t \omega)| \leq |z^*(\theta_n \omega)| + \sup_{s \in [n, n+1]} |z^*(\theta_s \omega) - z^*(\theta_n \omega)|,$$

et comme les trajectoires sont continues et gaussiennes, le second terme est contrôlé. On obtient ainsi

$$\lim_{|t| \rightarrow \infty} \frac{\ln^+ |z^*(\theta_t \omega)|}{|t|} = 0 \quad \text{presque sûrement,}$$

ce qui montre que $z^*(\theta_t \omega)$ est tempéré, et conclut la démonstration du point (i).

2. On sait que pour tout $t \in \mathbb{R}$, la variable $z^*(\theta_t \omega)$ est gaussienne centrée de même loi que $z^*(\omega)$ et de variance finie. Il existe donc $K, \lambda > 0$ tels que

$$\forall x > 0, \quad \mathbb{P}(|z^*(\theta_t \omega)| > x) = \mathbb{P}(|z^*(\omega)| > x) \leq K e^{-\lambda x^2}.$$

Fixons un $\varepsilon > 0$ et pour chaque entier $n \geq 1$, on pose

$$A_n := \{\omega : |z^*(\theta_n \omega)| > \varepsilon n\}.$$

Alors

$$\mathbb{P}(A_n) = \mathbb{P}(|z^*(\omega)| > \varepsilon n) \leq K e^{-\lambda (\varepsilon n)^2}$$

et la série

$$\sum_{n=1}^{\infty} \mathbb{P}(A_n) \leq \sum_{n=1}^{\infty} K e^{-\lambda \varepsilon^2 n^2}$$

converge. Par le lemme de Borel-Cantelli A.2, presque sûrement seuls un nombre fini de A_n se produisent. Il existe donc, pour presque tout ω , un rang $N(\omega)$ tel que

$$\forall n > N(\omega), \quad |z^*(\theta_n \omega)| \leq \varepsilon n.$$

D'où

$$\limsup_{n \rightarrow \infty} \frac{|z^*(\theta_n \omega)|}{n} \leq \varepsilon,$$

et comme $\varepsilon > 0$ est arbitraire,

$$\lim_{n \rightarrow \infty} \frac{|z^*(\theta_n \omega)|}{n} = 0.$$

Pour passer de l'entier n au réel $t \geq 0$, on écrit, en posant $n = \lfloor t \rfloor$,

$$\frac{|z^*(\theta_t \omega)|}{t} \leq \frac{|z^*(\theta_n \omega)|}{n} \frac{n}{t} + \frac{\sup_{s \in [n, n+1]} |z^*(\theta_s \omega) - z^*(\theta_n \omega)|}{t}.$$

Le premier terme tend vers 0 quand $t \rightarrow +\infty$, et le second aussi (continuité de $z^*(\theta_s W(\omega))$ sur chaque compact). On obtient donc

$$\lim_{t \rightarrow +\infty} \frac{|z^*(\theta_t \omega)|}{t} = 0.$$

Le même raisonnement appliqué à $t \rightarrow -\infty$ (en remplaçant n par $-n$ et en utilisant l'inversibilité de θ_t) donne enfin

$$\lim_{t \rightarrow \pm\infty} \frac{|z^*(\theta_t \omega)|}{|t|} = 0,$$

ce qui conclut la preuve.

On a

$$\frac{1}{t} \int_0^t z^*(\theta_s \omega) \, ds.$$

Le processus $z^*(\theta_s \omega)$ est stationnaire (par construction) et mesurable. De plus, c'est une variable aléatoire gaussienne centrée, donc intégrable :

$$\mathbb{E}[z^*(\omega)] = 0, \quad \text{et} \quad \mathbb{E}[|z^*(\omega)|] < \infty.$$

Le flux $(\theta_t)_{t \in \mathbb{R}}$ est ergodique sur l'espace de Wiener. Par le théorème ergodique de Birkhoff appliqué à la fonction $f(\omega) = z^*(\omega)$, on a, presque sûrement :

$$\lim_{t \rightarrow +\infty} \frac{1}{t} \int_0^t z^*(\theta_s \omega) \, ds = \mathbb{E}[z^*(\omega)] = 0.$$

De même, en posant $s' = -s$ et en inversant l'intégration, on obtient :

$$\frac{1}{t} \int_0^t z^*(\theta_s \omega) ds = -\frac{1}{|t|} \int_t^0 z^*(\theta_s \omega) ds \quad \text{pour } t < 0,$$

et le même théorème donne :

$$\lim_{t \rightarrow -\infty} \frac{1}{t} \int_0^t z^*(\theta_s \omega) ds = 0.$$

Donc, pour presque tout $\omega \in \Omega$,

$$\lim_{t \rightarrow \pm\infty} \frac{1}{t} \int_0^t z^*(\theta_s \omega) ds = 0.$$

Pour l'égalité suivante, le processus d'Ornstein-Uhlenbeck a la covariance qui décroît exponentiellement vite

$$\text{Cov}(X_0, X_t) = \frac{\nu^2}{2\beta} e^{-\beta|t|},$$

et par conséquent,

$$\int_0^\infty |\text{Cov}(X_0, X_t)| dt < \infty.$$

Cela suffit à garantir l'ergodicité du processus. On en déduit que

$$\lim_{t \rightarrow \pm\infty} \frac{1}{t} \int_0^t z^*(\theta_s \omega) ds = 0.$$

Pour l'égalité suivante, on considère la fonction $f(x) = |x|$, qui est mesurable. On peut alors appliquer le théorème ergodique de Birkhoff (voir 4.1). On obtient

$$\lim_{t \rightarrow \pm\infty} \frac{1}{t} \int_0^t |z^*(\theta_s \omega)| ds = \mathbb{E}[|z^*|].$$

Or, on sait que $z^* \sim \mathcal{N}\left(0, \frac{\nu^2}{2\beta}\right)$, donc

$$\mathbb{E}[|z^*|] = \sqrt{\frac{\nu^2}{\beta\pi}} < \infty.$$

Ce qui conclut la démonstration. □

4.2 Introduction des perturbations dans le modèle

Comme on l'a expliqué dans l'introduction, les perturbations aléatoires opérées dans le modèle déterministe du chémostat concernent le taux de dilution $D = Q/V$, (V étant constant). L'idée est de remplacer D dans le modèle déterministe par $D + \Phi(z^*(\theta_t \omega))$. On rappelle que $z^*(\theta_t \omega)$ est un processus d'O-U (4.5) et Φ est la fonction de saturation définie par (1.5).

4.2.1 Contrôle des perturbations

Soit $[D_l, D_r] \subset \mathbb{R}^+$ (avec $0 < D_l < D < D_r$) l'intervalle expérimental. On choisit la constante d intervenant dans la définition de la fonction Φ tel que $d = \min(D_r - D, D - D_l)$, pour garantir que le taux perturbé reste dans l'intervalle physique

$$D_l \leq D + \Phi(z^*(\theta_t \omega)) \leq D_r \quad \forall t \in \mathbb{R}. \quad (4.7)$$

Dans la section suivante le modèle du chémostat stochastique est introduit.

4.2.2 Système dynamique aléatoire (perturbé)

Le modèle stochastique correspondant aux perturbations introduites plus haut s'écrit alors

$$\begin{cases} \frac{ds}{dt} = (D + \Phi(z^*(\theta_t\omega))(s_{in} - s) - f(s))\frac{X}{Y} \\ \frac{dX}{dt} = -(D + \Phi(z^*(\theta_t\omega)))X + f(s)X. \end{cases}$$

En appliquant un changement de variable similaire au cas déterministe (voir le passage du système (3.1) au système (3.2)), la dynamique stochastique se simplifie sous la forme suivante

$$\begin{cases} \frac{ds}{dt} = (D + \Phi(z^*(\theta_t\omega))(s_{in} - s) - f(s))x \\ \frac{dx}{dt} = -(D + \Phi(z^*(\theta_t\omega)))x + f(s)x. \end{cases} \quad (4.8)$$

4.3 Persistance dans le chémostat perturbé

Dans la suite on se focalise sur le système ainsi obtenu (4.8). On rappelle les définitions de persistance uniforme dans le cadre stochastique considérées (voir [18]).

Définition 4.6. La dynamique du système (4.8) est fortement uniformément persistante s'il existe $\varepsilon > 0$ tel que pour toute condition initiale $(s(0), x(0))$ dans $\mathbb{R}_+ \times \mathbb{R}_+^*$ et toute réalisation de $D + \Phi(z^*(\theta_y\omega))$, la solution vérifie

$$\liminf_{t \rightarrow +\infty} x(t) > \varepsilon.$$

Définition 4.7. La dynamique du système (4.8) est faiblement uniformément persistante s'il existe $\varepsilon > 0$ tel que pour toute condition initiale $(s(0), x(0))$ dans $\mathbb{R}_+ \times \mathbb{R}_+^*$ et toute réalisation de $D + \Phi(z^*(\theta_y\omega))$, la solution vérifie

$$\limsup_{t \rightarrow +\infty} x(t) > \varepsilon.$$

4.4 Propriétés des solutions de la dynamique aléatoire

Dans cette section, on étudie le modèle stochastique du chémostat (4.8). On démontre l'existence et l'unicité d'une solution globale strictement positive, et on établit des résultats concernant l'existence d'ensembles absorbants et attracteurs (voir définition 4.8), qui présentent la particularité d'être déterministes (c'est-à-dire indépendants des trajectoires du processus aléatoire). On identifie également des conditions initiales sous lesquelles l'extinction de la biomasse devient inévitable.

Dans ce qui suit, on désigne par $\mathcal{X} := \{(s, x) \in \mathbb{R}^2; s, x \geq 0\}$ le quart du plan positif ($\mathbb{R}_+^2$).

Théorème 4.2. Pour toute condition initiale $u(0) \in \mathcal{X}$, le système (4.8) admet une solution globale unique

$$u(t; 0, \omega, u(0)) = (s(t; 0, \omega, u(0)), x(t; 0, \omega, u(0))) \in C^1([0, +\infty[, \mathcal{X}).$$

De plus, cette solution reste dans $\mathcal{X}$ pour tout $t > 0$.

Démonstration. On réécrit le système (4.8) sous la forme

$$\frac{du}{dt} = L(\theta_t\omega)u + F(u, \theta_t\omega), \quad (4.9)$$

où

$$u = \begin{pmatrix} s \\ x \end{pmatrix}, \quad L(\theta_t\omega) = \begin{pmatrix} -(D + \Phi(z^*(\theta_t\omega))) & 0 \\ 0 & -(D + \Phi(z^*(\theta_t\omega))) \end{pmatrix},$$

et

$$\begin{aligned} F : \mathcal{X} \times [0, +\infty[&\longrightarrow \mathcal{X} \\ (u, t) &\longmapsto F(u, \theta_t \omega), \end{aligned}$$

avec

$$F(u, \theta_t \omega) = \begin{pmatrix} (D + \Phi(z^*(\theta_t \omega)))s_{in} - f(s)x \\ f(s)x \end{pmatrix},$$

et f est la fonction de croissance de Haldane (1.2).

On a $F(\cdot, \theta_t \omega) \in C^1(\mathcal{X} \times [0, +\infty); \mathcal{X})$, ce qui implique que F est localement lipschitzienne par rapport à $u \in \mathcal{X}$. Par conséquent, pour chaque réalisation du bruit, l'équation (4.9) est une équation différentielle non autonome dont le membre de droite est localement lipschitzien en u et continu en t (on rappelle que $z^*(\theta_t \omega)$ est continu en t et que Φ est également continue). D'après le Théorème A.1, le système (4.8) admet donc une solution locale unique.

Tout d'abord on montre que $u = (s, x)$, solution de (4.8) reste dans $\mathcal{X}$ pour $u(0) = (s(0), x(0)) \in \mathcal{X}$. Pour cela, on observe d'abord que $x(t) = 0$ est une solution de

$$\frac{dx}{dt} = -(D + \Phi(z^*(\theta_t \omega)))x + f(s)x.$$

Par unicité de la solution du problème de Cauchy, on déduit que pour toute condition initiale non nulle, on a $x(t) \neq 0$ pour tout t . De plus, grâce à (4.7), on a

$$\left. \frac{ds}{dt} \right|_{s=0} = (D + \Phi(z^*(\theta_t \omega)))s_{in} \geq D_l s_{in} > 0,$$

ce qui prouve que l'axe $s = 0$ est répulsif dans $\mathcal{X}$. Cela démontre la positivité de la solution du système (4.8).

On montre à présent que les concentrations en substrat et en biomasse restent bornées pour tout t . Pour cela, on définit $v(t) := s(t) + x(t)$. La variable v est alors solution de l'équation différentielle suivante

$$\frac{dv}{dt} = (D + \Phi(z^*(\theta_t \omega)))(s_{in} - v)$$

et, grâce à (4.7), on a

$$\frac{dv}{dt} \leq D_r s_{in} - D_l v.$$

Par le principe de comparaison (voir A.3), on obtient

$$v(t; 0, \omega, v(0)) \leq v(0)e^{-D_l t} + \frac{D_r}{D_l} s_{in} (1 - e^{-D_l t}), \quad t \geq 0. \quad (4.10)$$

En effet, on considère la fonction suivante

$$\begin{aligned} h : [0, T] \times \mathbb{R} &\longrightarrow \mathbb{R} \\ (t, u) &\longmapsto h(t, u(t)) := D_r s_{in} - D_l u. \end{aligned}$$

h est une fonction continue, localement lipschitzienne par rapport à la seconde variable, uniformément en t .

On considère alors le problème de Cauchy suivant

$$\frac{du}{dt} = h(t, u(t)) = D_r s_{in} - D_l u, \quad u(0) = v(0).$$

On réécrit l'équation différentielle sous forme linéaire standard

$$\frac{du}{dt} + D_l u = D_r s_{in}.$$

On calcule le facteur intégrant

$$\rho(t) = e^{\int D_l dt} = e^{D_l t}.$$

On multiplie chaque terme de l'équation différentielle par $e^{D_l t}$

$$e^{D_l t} \frac{du}{dt} + D_l e^{D_l t} u = D_r s_{in} e^{D_l t}.$$

Le membre de gauche est la dérivée du produit

$$\frac{d}{dt} (u(t) e^{D_l t}) = D_r s_{in} e^{D_l t}.$$

On intègre les deux côtés de l'équation

$$\int \frac{d}{dt} (u(t) e^{D_l t}) dt = \int D_r s_{in} e^{D_l t} dt,$$

ce qui donne

$$u(t) e^{D_l t} = \frac{D_r s_{in}}{D_l} e^{D_l t} + C. \quad \text{avec } C \in \mathbb{R}.$$

On divise par $e^{D_l t}$ pour isoler $u(t)$

$$u(t) = \frac{D_r s_{in}}{D_l} + C e^{-D_l t}.$$

On utilise la condition initiale $u(0) = v(0)$ pour déterminer C

$$u(0) = \frac{D_r s_{in}}{D_l} + C \quad \Rightarrow \quad C = u(0) - \frac{D_r s_{in}}{D_l} = v(0) - \frac{D_r s_{in}}{D_l}.$$

On obtient la solution générale du problème de Cauchy

$$u(t) = \frac{D_r s_{in}}{D_l} + \left(v(0) - \frac{D_r s_{in}}{D_l} \right) e^{-D_l t}.$$

On obtient ainsi la majoration souhaitée grâce au principe de comparaison.

Ainsi, d'après (4.10), v est bornée en temps positif, et puisque $v(t) = s(t) + x(t) \geq 0$, on en déduit que les solutions s et x sont également bornées pour $t \geq 0$. La solution unique du système (4.8) est donc définie pour tout $t \geq 0$. $\square$

Pour la suite, il est nécessaire de définir les ensembles attracteurs et absorbants.

Définition 4.8. *Ensemble absorbant et ensemble attracteur*

Soit $\psi(t, s_0, x_0)$ une solution du système dynamique (4.8) avec la condition initiale $(s_0, x_0) \in \mathcal{X}$.

1. Un ensemble $B \subset \mathcal{X}$ est dit absorbant si pour tout compact $K \subset \mathcal{X}$, il existe un temps $T_K \geq 0$ tel que

$$\forall (s_0, x_0) \in K, \forall t \geq T_K \quad ; \quad \psi(t, s_0, x_0) \in B.$$

Autrement dit, toute trajectoire issue d'un ensemble compact K entre dans B après un certain temps T_K et y reste pour tous les temps ultérieurs.

2. Un ensemble $\mathcal{A} \subset \mathcal{X}$ est dit attracteur s'il vérifie les deux propriétés suivantes

— *Invariance*

$$\forall (s_0, x_0) \in \mathcal{A}, \forall t \geq 0, \quad \psi(t, s_0, x_0) \in \mathcal{A}.$$

— *Attractivité*

Pour tout compact $K \subset \mathcal{X}$, on a

$$\forall \varepsilon > 0, \exists T_\varepsilon \in \mathbb{R} \text{ tel que } \forall t > T_\varepsilon \sup_{(s_0, x_0) \in K} \text{dist}(\psi(t, s_0, x_0), A) < \varepsilon,$$

où

$$\text{dist}(\psi(t, s_0, x_0), A) := \inf_{(\hat{s}, \hat{x}) \in \mathcal{A}} \|\psi(t, s_0, x_0) - (\hat{s}, \hat{x})\|.$$

Les théorèmes suivants établissent l'existence d'ensembles déterministes attracteurs et absorbants pour les solutions du modèle aléatoire du chémostat (4.8).

Théorème 4.3. *Le système (4.8) possède un ensemble attractif déterministe*

$$\mathcal{A} := \{(s, x) \in \mathcal{X} : s + x = s_{in}\}. \quad (4.11)$$

Démonstration. Soit la variable $q(t) := s(t) - s_{in} + x(t)$. Alors q satisfait l'équation différentielle suivante

$$\frac{dq}{dt} = -(D + \Phi(z^*(\theta_t \omega)))q,$$

par séparation des variables, on obtient la solution

$$q(t; 0, \omega, q(0)) = q(0) \exp\left(-\int_0^t (D + \Phi(z^*(\theta_s \omega))) ds\right).$$

D'après (4.7), on a

$$q(0)e^{-D_1 t} \leq q(t; 0, \omega, q(0)) \leq q(0)e^{-D_l t},$$

et en prenant la limite lorsque t tend vers l'infini, on obtient

$$\lim_{t \rightarrow +\infty} q(t; 0, \omega, q(0)) = 0,$$

ce qui signifie que, pour tout $\varepsilon > 0$, il existe $T(\omega, \varepsilon) > 0$ tel que

$$q(t; 0, \omega, q(0)) \in [-\varepsilon, \varepsilon], \quad t \geq T(\omega, \varepsilon).$$

Ceci prouve que

$$\mathcal{B}_\varepsilon := \{(s, x) \in \mathcal{X} : s_{in} - \varepsilon \leq s + x \leq s_{in} + \varepsilon\}$$

est un ensemble absorbant déterministe pour le système (4.8). Comme $\varepsilon > 0$ est arbitraire, l'ensemble limite est bien $\mathcal{A}$.

On remarque que l'ensemble attractif (4.11) obtenu dans le Théorème 4.3 ne dépend pas de l'événement $\omega \in \Omega$. Ceci constitue une autre particularité issue de cette modélisation des réalisations aléatoires bornées. $\square$

A présent, l'objectif est d'établir des conditions sur les paramètres du système (4.8) conduisant à l'extinction de l'espèce.

Théorème 4.4. *On suppose qu'on est dans le cas où $D_l > f(s_m)$, où s_m est défini dans (1.4). Alors le singleton*

$$\mathcal{A}_e := \{(s_{in}, 0)\} \subset \mathcal{A},$$

est un ensemble attractif.

Démonstration. L'équation décrivant l'évolution de la biomasse dans notre système dynamique (4.8), combinée à l'inégalité (4.7) donnent

$$\begin{aligned}\frac{dx}{dt} &= -(D + \Phi(z^*(\theta_t \omega)))x + f(s)x \\ &\leq -(D_l + f(s_m))x.\end{aligned}$$

On applique le principe de comparaison (voir A.3) ainsi qu'une simple séparation de variables pour résoudre l'EDO, ceci conduit à l'inégalité suivante

$$x(t; 0, \omega, x(0)) \leq x(0)e^{-(D_l - f(s_m))t}. \quad (4.12)$$

En prenant $D_l > f(s_m)$ et en faisant tendre t vers $+\infty$, on obtient

$$\lim_{t \rightarrow +\infty} x(t; 0, \omega, x(0)) = 0,$$

pour presque tout $\omega \in \Omega$. □

4.5 Condition de persistance

Dans cette section, on va chercher à établir des conditions permettant d'assurer la persistance de l'espèce au sens défini dans la section 4.3. Cependant, avant cela, on va énoncer une propriété liée au choix de la fonction Φ .

Proposition 4.5. *Considérons la fonction $\Phi(z) = \frac{2d}{\pi} \arctan(z)$, voir (1.5). On a alors*

$$\lim_{t \rightarrow +\infty} \frac{1}{t} \int_0^t \Phi(z^*(\theta_s \omega)) \, ds = 0, \quad \text{presque sûrement dans } \Omega.$$

Démonstration.

On a

$$\int_{\Omega} |\Phi(z^*(\omega))| \, d\mathbb{P}(\omega) = \int_{\Omega} \left| \frac{2d}{\pi} \arctan(z^*(\omega)) \right| \, d\mathbb{P}(\omega) \leq \frac{2d}{\pi} \cdot \frac{\pi}{2} \mathbb{P}(\Omega) = d,$$

Donc $\Phi(z^*(\cdot)) \in L^1(\Omega, \mathcal{F}, \mathbb{P})$. Comme $\mathbb{P}$ est θ -invariante, le théorème ergodique de Birkhoff (voir Théorème 4.1) donne

$$\lim_{t \rightarrow +\infty} \frac{1}{t} \int_0^t \Phi(z^*(\theta_s \omega)) \, ds = \mathbb{E}[\Phi(z^*(\omega))], \quad \text{p.s. dans } \Omega.$$

Il suffit donc de montrer que $\mathbb{E}[\Phi(z^*(\omega))] = 0$. Or,

$$\mathbb{E}[\Phi(z^*(\omega))] = \int_{\mathbb{R}} \Phi(x) f_{OU}(x) \, dx = 0,$$

où f_{OU} est la densité de la variable aléatoire $z^*(\cdot)$, cette densité est paire car z^* est une gaussienne centrée, et Φ est une fonction impaire. □

Remarque 4.3.

La Proposition 4.5 montre que, si on modélise les perturbations aléatoires à l'aide de la fonction Φ appliquée au processus de bruit $z^(\theta_t \omega)$, alors la moyenne temporelle de ces perturbations converge vers zéro presque sûrement. Ce résultat ne dépend pas du choix précis de la fonction Φ , tant qu'elle vérifie qu'elle est impaire et mesurable. Cela signifie qu'on peut choisir différentes formes réalistes de bruit borné dans le modèle, sans perdre cette propriété utile de "moyenne nulle à long terme".*

Théorème 4.5 (Persistance faible).

Si $f(s_{in}) > D$, alors le modèle aléatoire du chémostat (4.8) est faiblement persistant, c'est-à-dire qu'il existe un réel $\varepsilon > 0$ tel que, pour toute condition initiale $x(0) \in \mathcal{X}$ et pour toute réalisation du bruit, on a

$$\limsup_{t \rightarrow +\infty} x(t; 0, \omega, x(0)) \geq \varepsilon.$$

Démonstration. Soit $\varepsilon > 0$ tel que $f(s) > D$ pour tout $s \in [s_{in} - \varepsilon, s_{in} + \varepsilon]$, et on définit

$$\eta := \min \{f(s) - D : s \in [s_{in} - \varepsilon, s_{in} + \varepsilon]\} > 0.$$

D'après le Théorème 4.3, on sait que $s(t; 0, \omega, s(0))$ converge asymptotiquement vers l'ensemble $[0, s_{in}]$. Par conséquent, il existe un instant $T(\omega, \varepsilon) > 0$ tel que

$$s(t; 0, \omega, s(0)) < s_{in} + \varepsilon, \quad \text{pour tout } t > T(\omega, \varepsilon).$$

On considère maintenant, pour $t > T(\omega, \varepsilon)$, les ensembles suivants

$$U(t) := \{\tau \in [T(\omega, \varepsilon), t] : s(\tau; 0, \omega, s(0)) < s_{in} - \varepsilon\}$$

$$V(t) := \{\tau \in [T(\omega, \varepsilon), t] : s(\tau; 0, \omega, s(0)) \in [s_{in} - \varepsilon, s_{in} + \varepsilon]\},$$

et les fonctions associées, à valeurs dans $[0, 1]$

$$u(t) := \frac{\text{mes}(U(t))}{t - T(\omega, \varepsilon)}, \quad v(t) := \frac{\text{mes}(V(t))}{t - T(\omega, \varepsilon)} = 1 - u(t),$$

où $\text{mes}(\cdot)$ est la mesure de Lebesgue.

À partir de la 2^{ème} équation du modèle du chémostat (4.8), on peut écrire l'inégalité suivante pour la dynamique de $x(t)$

$$\dot{x}(t) \geq \begin{cases} -(D + \Phi(z^*(\theta_t \omega)))x(t), & \text{si } t \in U(t) \\ (\eta - \Phi(z^*(\theta_t \omega)))x(t), & \text{si } t \in V(t). \end{cases}$$

en effet

- Sur $U(t)$

$$\dot{x}(t) = -(D + \Phi(z^*(\theta_t \omega)))x + f(s)x \geq -(D + \Phi(z^*(\theta_t \omega)))x,$$

$$\text{car } f(s) \geq 0.$$

- Sur $V(t)$

$$\dot{x}(t) = (f(s) - D - \Phi(z^*(\theta_t \omega)))x \geq (\eta - \Phi(z^*(\theta_t \omega)))x,$$

$$\text{car } f(s) - D \geq \eta.$$

En intégrant entre $T(\omega, \varepsilon)$ et un temps $t > T(\omega, \varepsilon)$, on obtient

$$x(t) \geq x(T(\omega, \varepsilon)) \exp \left((t - T(\omega, \varepsilon)) \left[-D \cdot u(t) + (1 - u(t))\eta - \frac{1}{t - T(\omega, \varepsilon)} \int_{T(\omega, \varepsilon)}^t \Phi(z^*(\theta_r \omega)) dr \right] \right). \quad (4.13)$$

En effet, on a

$$\dot{x}(t) = (\mu(s(t)) - D - \Phi(z^*(\theta_t \omega)))x(t) := G(t)x(t).$$

On utilise l'identité suivante, valable pour toute fonction positive $x(t)$

$$\frac{d}{dt} \ln x(t) = \frac{\dot{x}(t)}{x(t)} = G(t).$$

Donc

$$\ln x(t) - \ln x(T(\omega, \varepsilon)) = \int_{T(\omega, \varepsilon)}^t G(r) dr,$$

autrement dit

$$\ln\left(\frac{x(t)}{x(T(\omega, \varepsilon))}\right) = \int_{T(\omega, \varepsilon)}^t G(r) dr.$$

Comme on l'a vu, on distingue deux cas selon que $r \in U(t)$ ou $r \in V(t)$

$$\int_{T(\omega, \varepsilon)}^t G(r) dr \geq \int_{U(t)} [-D - \Phi(z^*(\theta_r \omega))] dr + \int_{V(t)} [\eta - \Phi(z^*(\theta_r \omega))] dr.$$

Donc

$$\ln\left(\frac{x(t)}{x(T(\omega, \varepsilon))}\right) \geq -D \cdot \text{mes}(U(t)) + \eta \cdot \text{mes}(V(t)) - \int_{T(\omega, \varepsilon)}^t \Phi(z^*(\theta_r \omega)) dr.$$

Ce qui s'écrit aussi

$$\ln\left(\frac{x(t)}{x(T(\omega, \varepsilon))}\right) \geq (t - T(\omega, \varepsilon)) \left[-D \cdot u(t) + \eta \cdot (1 - u(t)) - \frac{1}{t - T(\omega, \varepsilon)} \int_{T(\omega, \varepsilon)}^t \Phi(z^*(\theta_r \omega)) dr \right].$$

Et un passage à l'exponentielle donne le résultat voulu .

Supposons que

$$\lim_{t \rightarrow +\infty} u(t) := \lim_{t \rightarrow +\infty} \frac{\text{mes}(U(t))}{t - T(\omega, \varepsilon)} = 0.$$

D'après la Proposition 4.5, on a

$$\lim_{y \rightarrow +\infty} \frac{1}{y} \int_0^y \Phi(z^*(\theta_s \omega)) ds = 0, \quad \text{presque sûrement dans } \Omega.$$

En posant le changement de variable $y = t - T(\omega, \varepsilon)$, on obtient

$$\lim_{t \rightarrow +\infty} \frac{1}{t - T(\omega, \varepsilon)} \int_0^{t - T(\omega, \varepsilon)} \Phi(z^*(\theta_s \omega)) ds = 0.$$

Puis, en effectuant un second changement de variable $s = r - T(\omega, \varepsilon)$, on a

$$\lim_{t \rightarrow +\infty} \frac{1}{t - T(\omega, \varepsilon)} \int_{T(\omega, \varepsilon)}^t \Phi(z^*(\theta_{r - T(\omega, \varepsilon)} \omega)) dr = 0.$$

Or, la stationnarité du flux $(\theta_t)_{t \in \mathbb{R}}$ implique que

$$z^*(\theta_{r - T(\omega, \varepsilon)} \omega) \stackrel{\mathcal{L}}{=} z^*(\theta_r \omega).$$

Par conséquent, on obtient

$$\lim_{t \rightarrow +\infty} \frac{1}{t - T(\omega, \varepsilon)} \int_{T(\omega, \varepsilon)}^t \Phi(z^*(\theta_r \omega)) dr = 0.$$

Enfin, en multipliant par (-1) et en passant à l'exponentielle, on conclut

$$\lim_{t \rightarrow +\infty} \exp\left(-\frac{1}{t - T(\omega, \varepsilon)} \int_{T(\omega, \varepsilon)}^t \Phi(z^*(\theta_r \omega)) dr\right) = 1.$$

Et donc, l'inégalité (4.13) implique

$$\lim_{t \rightarrow +\infty} x(t) = +\infty,$$

ce qui est une contradiction, car on sait que $x(t)$ est bornée. On en déduit donc que les ensembles $U(t)$ doivent nécessairement vérifier

$$\text{mes}(U(t)) \rightarrow +\infty \quad \text{lorsque } t \rightarrow +\infty,$$

c'est-à-dire :

$$\text{mes}(\{t : s(t) < s_{\text{in}} - \varepsilon\}) = +\infty.$$

Ce qui montre que :

$$\liminf_{t \rightarrow +\infty} s(t) \leq s_{\text{in}} - \varepsilon,$$

ou de manière équivalente :

$$\limsup_{t \rightarrow +\infty} x(t) \geq \varepsilon > 0,$$

puisque $s(t) + x(t)$ converge vers s_{in} pour toute réalisation (voir le Théorème 4.3).

□

Les résultats obtenus dans ce chapitre fournissent des conditions précises pour caractériser le comportement asymptotique du modèle stochastique du chémostat, en particulier les cas d'extinction et de persistance faible de la biomasse. Dans le chapitre suivant, on va illustrer ces résultats théoriques à l'aide de simulations numériques. Ces simulations permettent non seulement de visualiser l'impact des perturbations aléatoires sur le système, mais aussi de mieux comprendre la dynamique à long terme en fonction des paramètres du modèle.

Chapitre 5

Simulations numériques

Dans ce chapitre, on présente des simulations numériques visant à soutenir l'étude théorique dans différents cas. Pour chaque situation, deux figures seront proposées :

- La première illustre les dynamiques du substrat et de la biomasse dans le modèle stochastique du chimostat (4.8), pour différentes réalisations du bruit aléatoire, ainsi que le cas déterministe utilisé à titre de comparaison, représenté en bleu pour le substrat et en rouge pour la biomasse.
- La deuxième figure fournit des informations sur la fonction de croissance et les paramètres impliqués, afin d'illustrer les différents comportements observés : extinction et persistance faible.

5.1 Extinction de la biomasse

On présente ici le cas d'extinction étudié dans le Théorème 4.4. On fixe les conditions initiales à $s(0) = 14$ et $x(0) = 5$. On choisit les paramètres suivants : $s_{in} = 14$, $m_0 = 4$, $k_s = 7.5$, $k_i = 4$, $D = 1.4$, ainsi que l'intervalle d'observation fourni par les expérimentateurs : $[D_l, D_r] = [1.15, 1.65]$, d'où $d = 0.25$ et on prend $\nu = 1$ et $\beta = 1$.

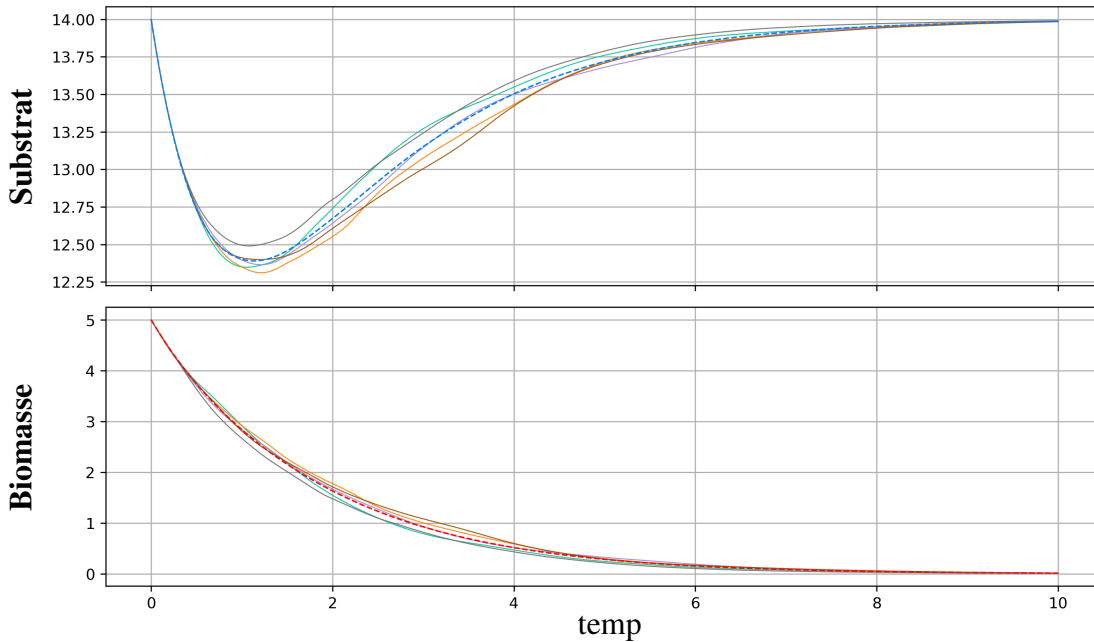


FIGURE 5.1 – Simulation de l'extinction de la biomasse.

Ce résultat est assez intuitif, la condition $D_l > f(s_m)$ signifie que la valeur nominale du taux de dilution D est trop élevée par rapport à la cinétique de croissance. Autrement dit, les micro-organismes sont éliminés plus rapidement qu'ils ne se reproduisent. Ceci conduit à une disparition progressive

de la biomasse même si la biomasse initiale est importante ($x(0) = 5$) et très éloignée de l'état de lessivage ($s_{in}, 0$).

On peut observer dans la Figure 5.2 que $D_l > f(s_m)$. Conformément au Théorème 4.4, cela entraîne la convergence vers l'extinction de l'espèce.

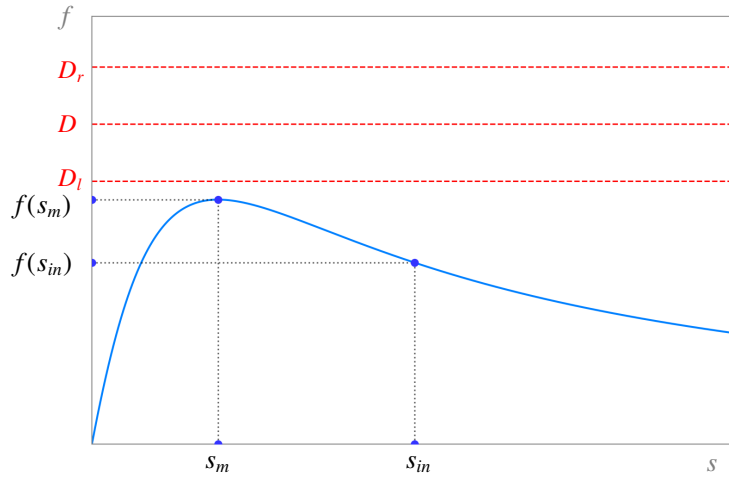


FIGURE 5.2 – Fonction de croissance f avec repères des valeurs D , D_l , et D_r dans le cas d'extinction de la biomasse.

Paramètres choisis : $s_{in} = 14$, $s_m = 5.4772$, $D = 1.4$, $D_l = 1.15$, $D_r = 1.65$,
 $f(s_{in}) = 0.7943$, $f(s_m) = 1.0699$.

5.2 Faible persistance

On passe au cas de la persistance faible étudié dans le Théorème 4.5. On considère les conditions initiales $s(0) = 16.7$ et $x(0) = 0.01$. On choisit les paramètres suivants : $s_{in} = 16.7$, $m_0 = 7$, $k_s = 7$, $k_i = 7$, $D = 1.7$, ainsi que l'intervalle d'observation fourni par les expérimentateurs : $[D_l, D_r] = [1.45, 1.95]$, d'où $d = 0.25$. De même, on prend $\nu = 1$ et $\beta = 1$.

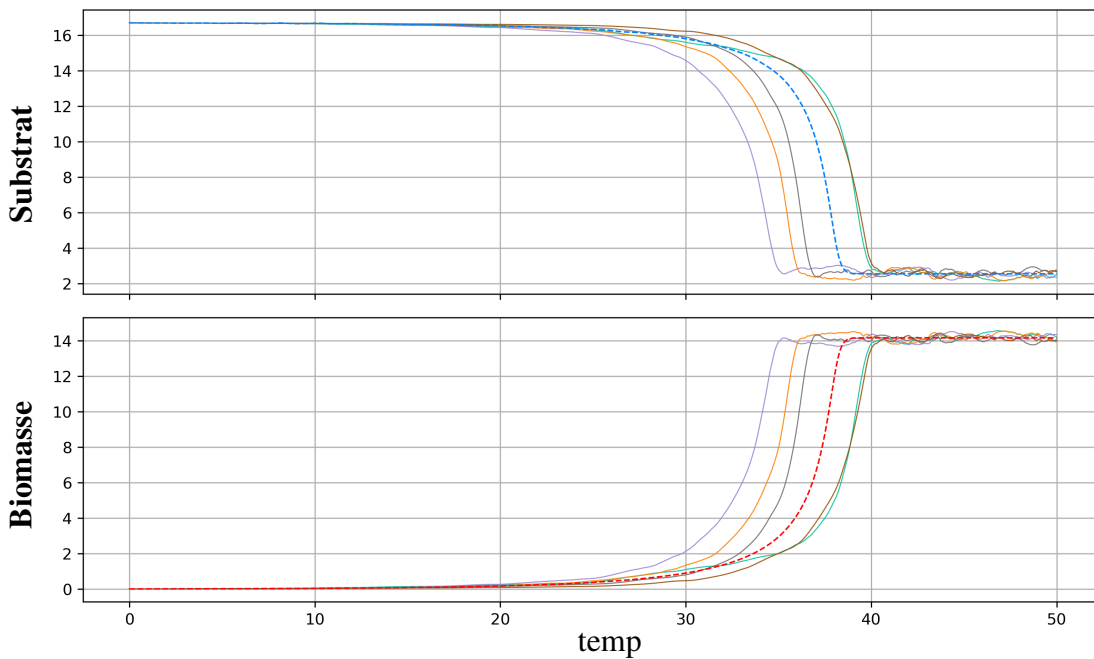


FIGURE 5.3 – Simulation de la faible persistance de la biomasse.

Dans cette simulation, malgré une biomasse initialement très faible, on observe qu'elle ne disparaît pas et reste durablement positive. Ceci illustre la faible persistance du système : la biomasse survit à long terme bien qu'elle puisse rester proche de zéro pendant des périodes prolongées. Ce comportement est cohérent avec la condition $D < f(s_{in})$ qui permet à l'espèce de persister malgré les fluctuations aléatoires du débit d'entrée.

On peut observer dans la Figure 5.4 le cas où $D < f(s_{in})$.

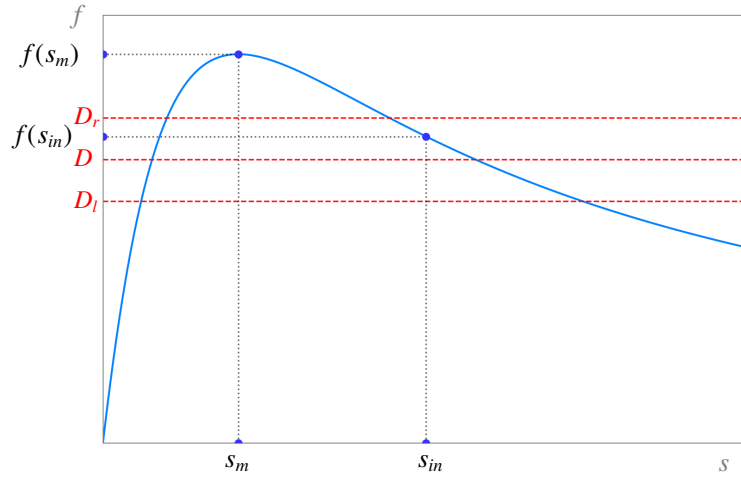


FIGURE 5.4 – Fonction de croissance f avec repères des valeurs D , D_l , et D_r dans le cas de la persistance faible de la biomasse.

Paramètres choisis : $s_{in} = 16.7$, $s_m = 7$, $D = 1.7$, $D_l = 1.45$, $D_r = 1.95$,
 $f(s_{in}) = 1.8397$, $f(s_m) = 2.3333$.

Conclusion

Dans ce mémoire, on s'est intéressé à l'étude du modèle du chimostat avec une fonction de croissance non monotone de type Haldane soumis à des perturbations aléatoires sur le taux de dilution. Ce type de modélisation permet de mieux comprendre l'effet d'un environnement fluctuant sur la dynamique de populations microbiennes, en lien avec des applications concrètes en biotechnologie.

Une première partie a été consacrée à l'analyse du modèle déterministe. Cette étape a permis d'identifier et de caractériser les différents équilibres du système en fonction des paramètres du modèle. Trois principales configurations ont été mises en évidence : extinction systématique, persistance unique, et bistabilité entre extinction et persistance. Ces résultats soulignent l'importance des conditions initiales et des paramètres du système dans la stabilité à long terme de la biomasse. Cette étude préalable s'est révélée essentielle pour mieux comprendre les effets introduits ensuite par l'aléa.

Dans la seconde partie, on a introduit un bruit borné dans le taux de dilution à l'aide d'un processus d'Ornstein-Uhlenbeck. Ce choix permet de modéliser de façon réaliste des perturbations continues mais limitées, telles que celles observées dans des environnements expérimentaux ou industriels. L'étude théorique a alors porté sur deux comportements distincts de la dynamique stochastique : l'extinction et la persistance faible de la biomasse.

On a tout d'abord montré que lorsque le taux moyen de dilution perturbé dépasse la valeur critique de croissance microbienne, l'extinction devient inévitable, quel que soit le taux d'entrée. À l'inverse, lorsque ce taux est en dessous de cette valeur critique, on observe une forme de persistance faible où la biomasse reste positive à long terme malgré des chutes ponctuelles, mais peut s'approcher arbitrairement de zéro selon les réalisations du bruit.

Ces résultats ont été illustrés par des simulations numériques, confirmant les comportements attendus et mettant en lumière la différence entre une persistance sensible au bruit et une extinction systématique. Même si la persistance forte n'a pas été explorée dans ce travail, les résultats obtenus permettent déjà de comprendre les effets qualitatifs des perturbations sur la survie des espèces dans le chimostat.

En conclusion, cette étude met en évidence l'importance de l'analyse conjointe des modèles déterministes et stochastiques. Le modèle déterministe fournit un cadre de référence structurant, tandis que l'introduction du bruit permet d'enrichir l'analyse par des phénomènes plus proches de la réalité expérimentale. Le processus d'Ornstein-Uhlenbeck s'est avéré être un outil efficace pour modéliser ces perturbations, tout en conservant des propriétés mathématiques permettant une analyse rigoureuse du système.

Bibliographie

- [1] L. Arnold. *Random Dynamical Systems*. Springer Berlin Heidelberg, 1998.
- [2] Y. Asai and P. Kloeden. Numerical schemes for random odes via stochastic differential equations. *Commun. Appl. Anal*, 17(3) :521–528, 2013.
- [3] O. Bernard, Z. Hadj-Sadok, D. Dochain, A. Genovesi, and J-P.Steyer. Dynamical model development and parameter identification for an anaerobic wastewater treatment process. *Biotechnology and bioengineering*, 75(4) :424–438, 2001.
- [4] T. Caraballo, R. Colucci, J. López de-la Cruz, and A. Rapaport. A way to model stochastic perturbations in population dynamics models with bounded realizations. *Communications in Nonlinear Science and Numerical Simulation*, 77 :239–257, 2019.
- [5] T. Caraballo, R. Colucci, J. López de-la Cruz, A. Rapaport, et al. *Study of the chemostat model with non-monotonic growth under random disturbances on the removal rate*, volume 17. AIMS Press, 2020.
- [6] T. Caraballo, M. J. Garrido-Atienza, J. López de-la Cruz, and A. Rapaport. Modeling and analysis of random and stochastic input flows in the chemostat model. *Discrete and Continuous Dynamical Systems-Series B*, 24(8) :3591–3614, 2019.
- [7] T. Caraballo and X. Han. *Applied nonautonomous and random dynamical systems : applied dynamical systems*. Springer, 2017.
- [8] T. Caraballo, X. Han, and P. E. Kloeden. Chemostats with random inputs and wall growth. *Mathematical Methods in the Applied Sciences*, 38(16) :3538–3550, 2015.
- [9] T. Caraballo, P. E. Kloeden, and B. Schmalfuss. Exponentially stable stationary solutions for stochastic evolution equations and their perturbation. *Applied Mathematics and Optimization*, 50 :183–207, 2004.
- [10] E. A. Coddington and N. Levinson. *Theory of ordinary differential equations*. McGraw-Hill Book Company, 1955.
- [11] DE Contois. Kinetics of bacterial growth : relationship between population density and specific growth rate of continuous cultures. *Microbiology*, 21(1) :40–50, 1959.
- [12] M. Dali-Youcef, J. Harmand, A. Rapaport, and T. Sari. Some non-intuitive properties of serial chemostats with and without mortality. *IFAC-PapersOnLine*, 55(20) :475–480, 2022.
- [13] M. Dali-Youcef, A. Rapaport, and T. Sari. Study of performance criteria of serial configuration of two chemostats. *Math. Biosci. Eng*, 17(6) :6278–6309, 2020.
- [14] M. Dali-Youcef, A. Rapaport, and T. Sari. Performance study of two serial interconnected chemostats with mortality. *Bulletin of Mathematical Biology*, 84(10) :110, 2022.
- [15] M. Dali-Youcef and T. Sari. The productivity of two serial chemostats. *International Journal of Biomathematics*, 16(06) :2250113, 2023.
- [16] D. Dochain and A. P. Vanrolleghem. *Dynamical modelling & estimation in wastewater treatment processes*. IWA publishing, 2001.
- [17] R. Durrett. *Probability : theory and examples*, volume 49. Cambridge university press, 2019.

- [18] H. I. Freedman and P. Moson. Persistence definitions and their connections. *Proceedings of the american mathematical society*, 109(4) :1025–1033, 1990.
- [19] J. B. S. Haldane. *Enzymes*. Longmans, Green and Co., London, 1930.
- [20] J. Harmand. *Contribution à l'analyse et au contrôle des systèmes biologiques : application aux bioprocédés de dépollution*. Thèse de doctorat, Université Montpellier II - Sciences et Techniques du Languedoc, Montpellier, France, 2004.
- [21] J. Harmand, C. Lobry, A. Rapaport, and T. Sari. *The Chemostat : Mathematical Theory of Microorganism Cultures*. John Wiley & Sons, Chemical Engineering Series, 2017.
- [22] J. Harmand, C. Lobry, A. Rapaport, and T. Sari. *The chemostat : Mathematical theory of microorganism cultures*, volume 1. John Wiley & Sons, 2017.
- [23] S. Itô. *Diffusion Equations*, volume 114. American Mathematical Society, 1992.
- [24] O. Kallenberg. *Foundations of modern probability*, volume 2. Springer, 1997.
- [25] I. Karatzas and S. Shreve. *Brownian motion and stochastic calculus*, volume 113. Springer Science & Business Media, 2012.
- [26] L. Michaelis, M. L. Menten L, et al. Die kinetik der invertinwirkung. *Biochem. z.*, 49(333-369) :352, 1913.
- [27] J. Monod. La technique de culture continue, théorie et applications. In *Annales de l'Institut Pasteur*, volume 79, pages 390–410, 1950.
- [28] J. Monod. Recherches sur la croissance des cultures bactériennes. 1958.
- [29] H. Moser. The dynamics of bacterial populations maintained in the chemostat. 1958.
- [30] A. Novick and L. Szilard. Description of the chemostat. *Science*, 112(2920) :715–716, 1950.
- [31] A. Rapaport and I. Haidar. Effects of spatial structure and diffusion on the performances of the chemostat. *arXiv preprint arXiv :1011.3742*, 2010.
- [32] A. Rapaport, I. Haidar, and J. Harmand. Global dynamics of the buffered chemostat for a general class of response functions. *Journal of mathematical biology*, 71(1) :69–98, 2015.
- [33] D. Revuz and M. Yor. *Continuous martingales and Brownian motion*. Springer, 1999.
- [34] T. Sari. Best operating conditions for biogas production in some simple anaerobic digestion models. *Processes*, 10(2) :258, 2022.
- [35] H. L. Smith and P. Waltman. *The theory of the chemostat : dynamics of microbial competition*, volume 13. Cambridge university press, 1995.
- [36] Manel Dali Youcef. *Étude et optimisation du modèle de deux chémostats en série*. PhD thesis, Université de Montpellier, 2022.

Annexe A

Notions mathématiques utiles

Théorème A.1 (Cauchy-Lipschitz pour une EDO).

Soit $f : [a, b] \times \mathbb{R} \rightarrow \mathbb{R}$ une fonction continue. On considère le problème de Cauchy suivant :

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

où $t_0 \in [a, b]$ et $x_0 \in \mathbb{R}$.

Supposons que :

- f est continue sur un voisinage de (t_0, x_0) .
- f est localement lipschitzienne par rapport à la variable x , c'est-à-dire qu'il existe un voisinage $\mathcal{U} \subset [a, b] \times \mathbb{R}$ de (t_0, x_0) et une constante $L > 0$ telle que

$$\forall (t, x_1), (t, x_2) \in \mathcal{U}, \quad |f(t, x_1) - f(t, x_2)| \leq L|x_1 - x_2|.$$

Alors, il existe $\varepsilon > 0$ tel que le problème admet une unique solution

$$x : [t_0 - \varepsilon, t_0 + \varepsilon] \rightarrow \mathbb{R},$$

satisfaisant les conditions initiales.

Définition A.1 (Processus stationnaire). Le processus $(X_t)_{t \in \mathbb{R}}$ est dit stationnaire si

$$\forall t \in \mathbb{R}, \quad X_t \stackrel{\mathcal{L}}{=} X_0.$$

Théorème A.2 (Lemme de Borel-Cantelli [17]). Soit $(\Omega, \mathcal{A}, \mathbb{P})$ un espace probabilisé, $(A_n)_{n \in \mathbb{N}}$ une suite d'événements, et A l'évènement $\limsup_{n \rightarrow +\infty} A_n$, autrement dit

$$A = \{\omega \in \Omega; \omega \text{ appartient à une infinité de } A_n\}.$$

- Si $\sum_{n \in \mathbb{N}} P(A_n) < +\infty$ alors $P(A) = 0$. Autrement dit, avec une probabilité égale à 1, au plus un nombre fini d'événements A_n se réalise.
- Si les événements A_n sont indépendants et si $\sum_{n \in \mathbb{N}} P(A_n) = +\infty$ alors $P(A) = 1$. Autrement dit, avec une probabilité égale à 1, une infinité d'événements A_n se réalise.

Lemme A.1 (Formule d'Itô en dimension 2 [23, 25]). Soit X un processus d'Itô bidimensionnel (c'est-à-dire $X_t = (X_t^1, X_t^2)$) et $f \in C^{1,2}([0, T] \times \mathbb{R}^2)$, alors pour tout $t \in [0, T]$ on a

$$\begin{aligned} f(t, X_t) = & f(0, X_0) + \int_0^t \frac{\partial f}{\partial t}(s, X_s) ds + \sum_{i=1}^2 \int_0^t \frac{\partial f}{\partial x_i}(s, X_s) dX_s^i + \\ & \frac{1}{2} \sum_{i=1}^2 \sum_{j=1}^2 \int_0^t \frac{\partial^2 f}{\partial x_i \partial x_j}(s, X_s) d\langle X^i, X^j \rangle_s. \end{aligned}$$

Définition A.2. Soit $(\Omega, \mathcal{F}, \mathbb{P})$ un espace probabilisé.

Les espaces $L^1(\mathbb{P})$ et $L^1(\Omega, \mathcal{F}, \mathbb{P})$ désignent l'ensemble des fonctions intégrables par rapport à une mesure de probabilité. Ils sont définis comme suit :

$$L^1(\mathbb{P}) = \{f: \Omega \rightarrow \mathbb{R} \mid f \text{ est mesurable et } \mathbb{E}[|f|] < +\infty\}$$

$$L^1(\Omega, \mathcal{F}, \mathbb{P}) = \{f: \Omega \rightarrow \mathbb{R} \mid f \text{ est } \mathcal{F}\text{-mesurable et } \mathbb{E}[|f|] < +\infty\}$$

Théorème A.3 (Principe de comparaison [10]).

Soit $f: [0, T] \times \mathbb{R} \rightarrow \mathbb{R}$ une fonction continue, localement lipschitzienne par rapport à la seconde variable x , uniformément en t sur les compacts. Soient $u, v: [0, T] \rightarrow \mathbb{R}$ deux fonctions de classe C^1 telles que :

$$\frac{du}{dt}(t) = f(t, u(t)),$$

$$\frac{dv}{dt}(t) \leq f(t, v(t)),$$

$$v(0) \leq u(0).$$

Alors, pour tout $t \in [0, T]$, on a :

$$v(t) \leq u(t).$$

Annexe B

Codes des figures sous Python

Python est un langage de programmation simple et puissant, idéal pour tracer des graphes mathématiques grâce à des bibliothèques comme Matplotlib et NumPy. Il permet de visualiser facilement des fonctions, des équations ou des données mathématiques. Python a été employé dans ce mémoire pour produire les visualisations des résultats mathématiques. Les extraits de code inclus reflètent les implémentations essentielles tout en garantissant la reproductibilité des graphiques présentés.

Code Figure 1.2 : Allure de la fonction de Haldane.

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 Sin=10
5 m0=4
6 ks=6
7 ki=1/2
8
9 S=np.arange(0,Sin,0.01)
10 f=lambda S: m0 * (S / (ks + S + S**2 / ki) )
11
12 Sm=np.sqrt(ks*ki)
13 y_max=f(Sm)
14
15 fig, ax = plt.subplots()
16
17 # Tracer la fonction
18 ax.plot(S, f(S), color="#0080FF", linewidth=1)
19
20 # Ajouter le point maximal
21 ax.scatter(Sm, y_max, color="#3333FF", s=10, zorder=3)
22 ax.scatter(Sm,0, color="#3333FF", s=10, zorder=3)
23 ax.scatter(0,y_max, color="#3333FF", s=10, zorder=3)
24
25 # Ajouter des lignes pointillées
26 ax.axvline(Sm, linestyle=":", color="gray", alpha=0.8)
27 ax.axhline(y_max, linestyle=":", color="gray", alpha=0.8)
28
29 # Ajuster les limites pour que le graphe parte du coin (0,0)
30 ax.set_xlim([0, Sin])
31 ax.set_ylim([0, f(S).max() * 1.1])
32
33 # Supprimer les ticks des axes
```

```

34 ax.set_xticks([])
35 ax.set_yticks([])
36
37 ax.spines["top"].set_linewidth(0.5) # Modifier la largeur de l'axe
    superieur
38 ax.spines["right"].set_linewidth(0.5) # Modifier la largeur de l'axe
    droit
39 ax.spines["bottom"].set_linewidth(0.5) # Modifier la largeur de l'axe
    inferieur
40 ax.spines["left"].set_linewidth(0.5) # Modifier la largeur de l'axe
    gauche
41
42 # Mettre en gris les axes X et Y restants
43 ax.spines["bottom"].set_color("gray")
44 ax.spines["left"].set_color("gray")
45 ax.spines["right"].set_color("gray")
46 ax.spines["top"].set_color("gray")
47
48 # Afficher le graphique
49 plt.show()

```

Code Figure 2.5 : L'allure locale d'un noeud instable.

```

1  import numpy as np
2  import matplotlib.pyplot as plt
3
4  lambda_1=1
5  lambda_2=2
6
7  p = [480,620]
8  u_0 = [-3, -0.5, 0.5, 3, -3, -0.5, 0.5, 3]
9  v_0 = [ 3, 0.5, -0.5, -3, -3, -0.5, 0.5, 3]
10
11 def u(i, t): return u_0[i] * np.exp(lambda_1 * t)
12 def v(i, t): return v_0[i] * np.exp(lambda_2 * t)
13
14 plt.figure(figsize=(6, 6))
15
16 t = np.linspace(-5, 5, 1000)
17
18 def fleches(leng, head=False):
19     for i in range(len(u_0)):
20         u_vals = u(i, t)
21         v_vals = v(i, t)
22         plt.plot(u_vals, v_vals, linewidth=1, color="#3399FF", zorder=2)
23
24         for j in p:
25             u_j = u_vals[j]
26             v_j = v_vals[j]
27             dx = leng * lambda_1 * u_j
28             dy = leng * lambda_2 * v_j
29             plt.arrow(u_j, v_j, dx, dy,
30                     shape='full', color='red',
31                     length_includes_head=head,
32                     head_width=0.1, head_length=0.2,

```

```

33                                     zorder=3)
34 fleches(0.001)
35
36 #fleches sur les axes
37 p=[600]
38 u_0=[0.5, -0.5]
39 v_0=[0, 0]
40 fleches(0.7,True)
41
42 p=[480]
43 u_0=[0, 0]
44 v_0=[ 1.1, -1.1]
45 fleches(0.4,True)
46
47 # Axes et point o
48 plt.axhline(0, color='gray', linewidth=0.8, linestyle='--', zorder=1)
49 plt.axvline(0, color='gray', linewidth=0.8, linestyle='--', zorder=1)
50 plt.scatter(0, 0, marker='o', s=8, color='black', zorder=4)
51
52 # Ajuster les limites des axes
53 plt.ylim(-3, 3)
54 plt.xlim(-3, 3)
55 plt.gca().set_aspect('equal')
56 plt.axis('off')
57
58 # Afficher le graphe
59 plt.show()

```

Code Figure 2.6 : L'allure locale d'un noeud stable.

```

1  import numpy as np
2  import matplotlib.pyplot as plt
3
4  lambda_1=-1
5  lambda_2=-2
6
7  p = [380,530]
8  u_0 = [-3, -0.5, 0.5, 3, -3, -0.5, 0.5, 3]
9  v_0 = [ 3, 0.5, -0.5, -3, -3, -0.5, 0.5, 3]
10
11 def u(i, t): return u_0[i] * np.exp(lambda_1 * t)
12 def v(i, t): return v_0[i] * np.exp(lambda_2 * t)
13
14 plt.figure(figsize=(6, 6))
15
16 t = np.linspace(-5, 5, 1000)
17
18 def fleches(leng):
19     for i in range(len(u_0)):
20         u_vals = u(i, t)
21         v_vals = v(i, t)
22         plt.plot(u_vals, v_vals, linewidth=1, color="#3399FF", zorder=2)
23
24     for j in p:
25         u_j = u_vals[j]

```

```

26         v_j = v_vals[j]
27         dx = leng * lambda_1 * u_j
28         dy = leng * lambda_2 * v_j
29         plt.arrow(u_j, v_j, dx, dy,
30                  shape='full', color='red',
31                  length_includes_head=True,
32                  head_width=0.1, head_length=0.2,
33                  zorder=3)
34 fleches(0.001)
35
36 #fleches sur les axes
37 p=[500]
38 u_0=[2.7, -2.7]
39 v_0=[0, 0]
40 fleches(0.4)
41
42 p=[480]
43 u_0=[0, 0]
44 v_0=[ 1.15, -1.15]
45 fleches(0.2)
46
47 # Axes et point o
48 plt.axhline(0, color='gray', linewidth=0.8, linestyle='--', zorder=1)
49 plt.axvline(0, color='gray', linewidth=0.8, linestyle='--', zorder=1)
50 plt.scatter(0, 0, marker='o', s=8, color='black', zorder=4)
51
52 # Ajuster les limites des axes
53 plt.ylim(-3, 3)
54 plt.xlim(-3, 3)
55 plt.gca().set_aspect('equal')
56 plt.axis('off')
57
58 # Afficher le graphe
59 plt.show()

```

Code Figure 2.7 : L'allure locale d'un point selle.

```

1  import numpy as np
2  import matplotlib.pyplot as plt
3
4  lambda_1 = 1
5  lambda_2 = -1
6
7  p = [490, 620]
8  u_0 = [-2, -1, 1, 2, -2, -1, 1, 2]
9  v_0 = [ 2, 1, -1, -2, -2, -1, 1, 2]
10
11 def u(i, t): return u_0[i] * np.exp(lambda_1 * t)
12 def v(i, t): return v_0[i] * np.exp(lambda_2 * t)
13
14 plt.figure(figsize=(6, 6))
15
16 t = np.linspace(-5, 5, 1000)
17
18 def fleches(leng):

```

```

19     for i in range(len(u_0)):
20         u_vals = u(i, t)
21         v_vals = v(i, t)
22         plt.plot(u_vals, v_vals, linewidth=1, color="#3399FF", zorder=2)
23
24         for j in p:
25             u_j = u_vals[j]
26             v_j = v_vals[j]
27             dx = -leng * lambda_1 * u_j
28             dy = -leng * lambda_2 * v_j
29             plt.arrow(u_j, v_j, dx, dy,
30                     shape='full', color='red',
31                     length_includes_head=True,
32                     head_width=0.1, head_length=0.2,
33                     zorder=3)
34 fleches(0.001)
35
36 #fleches sur les axes
37 p=[600]
38 u_0=[1, -1]
39 v_0=[0, 0]
40 fleches(0.3)
41
42 p=[480]
43 u_0=[0, 0]
44 v_0=[ 1, -1]
45 fleches(0.5)
46
47 # Axes et point o
48 plt.axhline(0, color='gray', linewidth=0.8, linestyle='--', zorder=1)
49 plt.axvline(0, color='gray', linewidth=0.8, linestyle='--', zorder=1)
50 plt.scatter(0, 0, marker='o', s=8, color='black', zorder=4)
51
52 # Ajuster les limites des axes
53 plt.ylim(-3, 3)
54 plt.xlim(-3, 3)
55 plt.gca().set_aspect('equal')
56 plt.axis('off')
57
58 # Afficher le graphe
59 plt.show()

```

Code Figure 2.8 : L'allure locale d'une étoile instable.

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 lambda_1=1
5 lambda_2=1
6
7 p = [500]
8 u_0 = [-0.6, 0.6, 1, -1, 0.6, -0.6, -1, 1]
9 v_0 = [ 1, -1, -0.6, 0.6, 1, -1, -0.6, 0.6]
10
11 def u(i, t): return u_0[i] * np.exp(lambda_1 * t)

```

```

12 def v(i, t): return v_0[i] * np.exp(lambda_2 * t)
13
14 plt.figure(figsize=(6, 6))
15
16 t = np.linspace(-5, 5, 1000)
17
18 def fleches(leng, head=False):
19     for i in range(len(u_0)):
20         u_vals = u(i, t)
21         v_vals = v(i, t)
22         plt.plot(u_vals, v_vals, linewidth=1, color="#3399FF", zorder=2)
23
24         for j in p:
25             u_j = u_vals[j]
26             v_j = v_vals[j]
27             dx = leng * lambda_1 * u_j
28             dy = leng * lambda_2 * v_j
29             plt.arrow(u_j, v_j, dx, dy,
30                     shape='full', color='red',
31                     length_includes_head=head,
32                     head_width=0.1, head_length=0.2,
33                     zorder=3)
34 fleches(0.001)
35
36 #fleches sur les axes
37 p=[600]
38 u_0=[0.5, -0.5]
39 v_0=[0, 0]
40 fleches(0.7,True)
41
42 p=[500]
43 u_0=[0, 0]
44 v_0=[ 0.95, -0.95]
45 fleches(0.8,True)
46
47 # Axes et point o
48 plt.axhline(0, color='gray', linewidth=0.8, linestyle='--', zorder=1)
49 plt.axvline(0, color='gray', linewidth=0.8, linestyle='--', zorder=1)
50 plt.scatter(0, 0, marker='o', s=8, color='black', zorder=4)
51
52 # Ajuster les limites des axes
53 plt.ylim(-3, 3)
54 plt.xlim(-3, 3)
55 plt.gca().set_aspect('equal')
56 plt.axis('off')
57
58 # Afficher le graphe
59 plt.show()

```

Code Figure 2.9 : L'allure locale d'une étoile stable.

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 lambda_1=-1

```

```

5 lambda_2=-1
6
7 p = [360,510]
8 u_0 = [-0.6, 0.6, 1, -1, 0.6, -0.6, -1, 1]
9 v_0 = [ 1, -1, -0.6, 0.6, 1, -1, -0.6, 0.6]
10
11 def u(i, t): return u_0[i] * np.exp(lambda_1 * t)
12 def v(i, t): return v_0[i] * np.exp(lambda_2 * t)
13
14 plt.figure(figsize=(6, 6))
15
16 t = np.linspace(-5, 5, 1000)
17
18 def fleches(leng):
19     for i in range(len(u_0)):
20         u_vals = u(i, t)
21         v_vals = v(i, t)
22         plt.plot(u_vals, v_vals, linewidth=1, color="#3399FF", zorder=2)
23
24         for j in p:
25             u_j = u_vals[j]
26             v_j = v_vals[j]
27             dx = leng * lambda_1 * u_j
28             dy = leng * lambda_2 * v_j
29             plt.arrow(u_j, v_j, dx, dy,
30                      shape='full', color='red',
31                      length_includes_head=True,
32                      head_width=0.1, head_length=0.2,
33                      zorder=3)
34 fleches(0.001)
35
36 #fleches sur les axes
37 p=[600]
38 u_0=[6, -6]
39 v_0=[0, 0]
40 fleches(0.4)
41
42 p=[480]
43 u_0=[0, 0]
44 v_0=[ 1.8, -1.8]
45 fleches(0.4)
46
47 # Axes et point o
48 plt.axhline(0, color='gray', linewidth=0.8, linestyle='--', zorder=1)
49 plt.axvline(0, color='gray', linewidth=0.8, linestyle='--', zorder=1)
50 plt.scatter(0, 0, marker='o', s=8, color='black', zorder=4)
51
52 # Ajuster les limites des axes
53 plt.ylim(-3, 3)
54 plt.xlim(-3, 3)
55 plt.gca().set_aspect('equal')
56 plt.axis('off')
57
58 # Afficher le graphe

```

```
59 plt.show()
```

Code Figure 2.10 : L'allure locale d'un noeud dégénéré instable.

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 lambda_1 = 1
5 lambda_2 = 1
6
7 p_indices = [450]
8 u_0 = [-1, -3, 3, 1, -3, -1, 1, 3]
9 v_0 = [1, 3, -3, -1, -3, -1, 1, 3]
10
11 def u(t, u0, v0): return (v0 * t + u0) * np.exp(lambda_1 * t)
12 def v(t, v0): return v0 * np.exp(lambda_2 * t)
13
14 plt.figure(figsize=(6, 6))
15
16 t = np.linspace(-5, 5, 1000)
17
18 def fleches(leng, head=False):
19     for u0_i, v0_i in zip(u_0, v_0):
20         u_vals = u(t, u0_i, v0_i)
21         v_vals = v(t, v0_i)
22         plt.plot(u_vals, v_vals, linewidth=1, color="#3399FF", zorder=2)
23         for j in p_indices:
24             if j < len(t) - 1:
25                 u1, v1 = u_vals[j], v_vals[j]
26                 u2, v2 = u_vals[j + 1], v_vals[j + 1]
27                 du = u2 - u1
28                 dv = v2 - v1
29                 norm = np.sqrt(du**2 + dv**2)
30                 if norm > 0:
31                     plt.arrow(u1, v1, leng * du / norm, leng * dv / norm,
32                               shape='full', color='red',
33                               length_includes_head=True,
34                               head_width=0.1, head_length=0.2,
35                               zorder=3)
36
37 fleches(0.2, head=True)
38
39 #fleches sur l'axe
40 p=[600]
41 u_0=[2, -2]
42 v_0=[0, 0]
43 fleches(0.7, True)
44
45 # Axes et point o
46 plt.axhline(0, color='gray', linewidth=0.8, linestyle='--', zorder=1)
47 plt.axvline(0, color='gray', linewidth=0.8, linestyle='--', zorder=1)
48 plt.scatter(0, 0, marker='o', s=8, color='black', zorder=4)
49
50 # Ajuster les limites des axes
51 plt.ylim(-3, 3)
```

```

52 plt.xlim(-3, 3)
53 plt.gca().set_aspect('equal')
54 plt.axis('off')
55
56 # Afficher le graphe
57 plt.show()

```

Code Figure 2.11 : L'allure locale d'un noeud dégénéré stable.

```

1  import numpy as np
2  import matplotlib.pyplot as plt
3
4  lambda_1 = -1
5  lambda_2 = -1
6
7  p_indices = [540]
8  u_0 = [-1, -3, 3, 1, -3, -1, 1, 3]
9  v_0 = [1, 3, -3, -1, -3, -1, 1, 3]
10
11 def u(t, u0, v0): return (v0 * t + u0) * np.exp(lambda_1 * t)
12 def v(t, v0): return v0 * np.exp(lambda_2 * t)
13
14 plt.figure(figsize=(6, 6))
15
16 t = np.linspace(-5, 5, 1000)
17
18 def fleches(leng, head=False):
19     for u0_i, v0_i in zip(u_0, v_0):
20         u_vals = u(t, u0_i, v0_i)
21         v_vals = v(t, v0_i)
22         plt.plot(u_vals, v_vals, linewidth=1, color="#3399FF", zorder=2)
23         for j in p_indices:
24             if j < len(t) - 1:
25                 u1, v1 = u_vals[j], v_vals[j]
26                 u2, v2 = u_vals[j + 1], v_vals[j + 1]
27                 du = u2 - u1
28                 dv = v2 - v1
29                 norm = np.sqrt(du**2 + dv**2)
30                 if norm > 0:
31                     plt.arrow(u1, v1, leng * du / norm, leng * dv / norm,
32                               shape='full', color='red',
33                               length_includes_head=True,
34                               head_width=0.1, head_length=0.2,
35                               zorder=3)
36
37 fleches(0.2, head=True)
38
39 #fleches sur l'axe
40 p=[300]
41 u_0=[3, -3]
42 v_0=[0, 0]
43 fleches(0.7, True)
44
45 # Axes et point o
46 plt.axhline(0, color='gray', linewidth=0.8, linestyle='-', zorder=1)

```

```

47 plt.axvline(0, color='gray', linewidth=0.8, linestyle='--', zorder=1)
48 plt.scatter(0, 0, marker='o', s=8, color='black', zorder=4)
49
50 # Ajuster les limites des axes
51 plt.ylim(-3, 3)
52 plt.xlim(-3, 3)
53 plt.gca().set_aspect('equal')
54 plt.axis('off')
55
56 # Afficher le graphe
57 plt.show()

```

Code Figure 2.12 : L'allure locale d'un foyer instable.

```

1  import numpy as np
2  import matplotlib.pyplot as plt
3
4  t = np.linspace(0, 2 * np.pi, 1000)
5  r0 = 0.2 # Rayon initial
6  a = 1.6 # Controle l'expansion de la spirale (plus petit = spirale plus
          serree)
7  b = 15 # Frequence angulaire (plus petit = tours moins rapides)
8  b0 = 0 # Phase initiale (pour centrer la spirale)
9
10 # Coefficients harmoniques
11 a1, b1 = 0, 1
12 a2, b2 = 1, 0
13
14 x = lambda t: r0 * np.exp(a*t)*(b1*np.cos(b*t + b0)+ a1*np.sin(b*t + b0))
15 y = lambda t: r0 * np.exp(a*t)*(b2*np.cos(b*t + b0)+ a2*np.sin(b*t + b0))
16
17 plt.figure(figsize=(6, 6))
18
19 plt.plot(x(t), y(t), linewidth=1, color="#3399FF", zorder=2)
20
21 # Ajouter les axes du plan
22 plt.axhline(0, color='gray', linewidth=0.8, linestyle='--', zorder=1) #
    Axe horizontal (y=0)
23 plt.axvline(0, color='gray', linewidth=0.8, linestyle='--', zorder=1) #
    Axe vertical (x=0)
24
25 plt.scatter(0, 0, marker='o', s=8, color='black', zorder=4)
26
27 k=[60,99,140,180,220,240,260,275]
28
29 for i in range(len(k)):
30     plt.arrow(x(t)[k[i]], y(t)[k[i]], x(t)[k[i]+83]*0.001, y(t)[k[i]
31               ]+83]*0.001, shape='full', color='red',
32               head_width=0.1, head_length=0.2, zorder=2)
33
34 plt.ylim(-3, 3)
35 plt.xlim(-3, 3)
36 plt.gca().set_aspect('equal')
37 plt.axis('off')

```

```
38 plt.show()
```

Code Figure 2.13 : L'allure locale d'un foyer stable.

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 t = np.linspace(0, 2 * np.pi, 1000)
5 r0 = 0.2 # Rayon initial
6 a = 1.6 # Controle l'expansion de la spirale (plus petit = spirale plus
    serree)
7 b = 15 # Frequence angulaire (plus petit = tours moins rapides)
8 b0 = 0 # Phase initiale (pour centrer la spirale)
9
10 # Coefficients harmoniques
11 a1, b1 = 0, 1
12 a2, b2 = 1, 0
13
14 x = lambda t: r0 * np.exp(a*t)*(b1*np.cos(b*t + b0)+ a1*np.sin(b*t + b0))
15 y = lambda t: r0 * np.exp(a*t)*(b2*np.cos(b*t + b0)+ a2*np.sin(b*t + b0))
16
17 plt.figure(figsize=(6, 6))
18
19 plt.plot(-x(t), y(t), linewidth=1, color="#3399FF", zorder=2)
20
21 # Ajouter les axes du plan
22 plt.axhline(0, color='gray', linewidth=0.8, linestyle='--', zorder=1) #
    Axe horizontal (y=0)
23 plt.axvline(0, color='gray', linewidth=0.8, linestyle='--', zorder=1) #
    Axe vertical (x=0)
24
25 plt.scatter(0, 0, marker='o', s=8, color='black', zorder=4)
26
27 k=[62,99,140,180,220,240,260,275]
28
29 for i in range(len(k)):
30     plt.arrow(-x(t)[k[i]], y(t)[k[i]], -x(t)[k[i]-85]*0.0000001, y(t)[k[i]
        ]-85]*0.0000001, shape='full', color='red',
31         head_width=0.1, head_length=0.2, zorder=2)
32
33 plt.ylim(-3, 3)
34 plt.xlim(-3, 3)
35 plt.gca().set_aspect('equal')
36 plt.axis('off')
37
38 plt.show()
```

Code Figure 2.14 : L'allure locale d'un centre.

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # Parametres des cercles
5 r1 = 1.2
6 r2 = 2
7 r3 = 2.8
```

```

8 centre_x, centre_y = 0, 0 # Centre des cercles
9 theta = np.linspace(0, 2*np.pi, 100) # 100 points pour un cercle lisse
10
11 # Coordonnees des cercles
12 x1 = centre_x + r1 * np.cos(theta)
13 y1 = centre_y + r1 * np.sin(theta)
14
15 x2 = centre_x + r2 * np.cos(theta)
16 y2 = centre_y + r2 * np.sin(theta)
17
18 x3 = centre_x + r3 * np.cos(theta)
19 y3 = centre_y + r3 * np.sin(theta)
20
21 plt.figure(figsize=(6, 6))
22
23 # Trace des cercles
24 plt.plot(x1, y1, linewidth=1, color="#3399FF", zorder=1)
25 plt.plot(x2, y2, linewidth=1, color="#3399FF", zorder=1)
26 plt.plot(x3, y3, linewidth=1, color="#3399FF", zorder=1)
27
28 # Fleches sur le cercle 1
29 plt.arrow(0, r1, 0.1, 0, shape='full', color='red', length_includes_head=
    True,
30           head_width=0.1, head_length=0.2, zorder=2)
31 plt.arrow(0, -r1, -0.1, 0, shape='full', color='red',
    length_includes_head=True,
32           head_width=0.1, head_length=0.2, zorder=2)
33 plt.arrow(r1, 0, 0, -0.1, shape='full', color='red', length_includes_head
    =True,
34           head_width=0.1, head_length=0.2, zorder=2)
35 plt.arrow(-r1, 0, 0, 0.1, shape='full', color='red', length_includes_head
    =True,
36           head_width=0.1, head_length=0.2, zorder=2)
37
38 # Fleches sur le cercle 2
39 plt.arrow(0, r2, 0.1, 0, shape='full', color='red', length_includes_head=
    True,
40           head_width=0.1, head_length=0.2, zorder=2)
41 plt.arrow(0, -r2, -0.1, 0, shape='full', color='red',
    length_includes_head=True,
42           head_width=0.1, head_length=0.2, zorder=2)
43 plt.arrow(r2, 0, 0, -0.1, shape='full', color='red', length_includes_head
    =True,
44           head_width=0.1, head_length=0.2, zorder=2)
45 plt.arrow(-r2, 0, 0, 0.1, shape='full', color='red', length_includes_head
    =True,
46           head_width=0.1, head_length=0.2, zorder=2)
47
48 # Fleches sur le cercle 3
49 plt.arrow(0, r3, 0.1, 0, shape='full', color='red', length_includes_head=
    True,
50           head_width=0.1, head_length=0.2, zorder=2)
51 plt.arrow(0, -r3, -0.1, 0, shape='full', color='red',
    length_includes_head=True,

```

```

52         head_width=0.1, head_length=0.2, zorder=2)
53 plt.arrow(r3, 0, 0, -0.1, shape='full', color='red', length_includes_head
    =True,
54         head_width=0.1, head_length=0.2, zorder=2)
55 plt.arrow(-r3, 0, 0, 0.1, shape='full', color='red', length_includes_head
    =True,
56         head_width=0.1, head_length=0.2, zorder=2)
57
58 # Ajouter les axes du plan
59 plt.axhline(0, color='gray', linewidth=0.8, linestyle='--', zorder=1) #
    Axe horizontal (y=0)
60 plt.axvline(0, color='gray', linewidth=0.8, linestyle='--', zorder=1) #
    Axe vertical (x=0)
61
62 # origine
63 plt.scatter(0, 0, marker='o', s=8, color='black', zorder=4)
64
65 plt.ylim(-3, 3)
66 plt.xlim(-3, 3)
67 plt.gca().set_aspect('equal')
68 plt.axis('off')
69
70 plt.show()

```

Code Figure 2.15 : Classification des portraits de phase.

```

1  import numpy as np
2  import matplotlib.pyplot as plt
3
4  t = np.linspace(-9, 9, 1000)
5
6  plt.figure(figsize=(10, 8))
7
8  plt.plot(t, 0.12 * t**2, linewidth=1, color="red", zorder=2, linestyle="
    --")
9
10 plt.axhline(0, color='gray', linewidth=0.8, linestyle='--', zorder=1)
11 plt.axvline(0, color='gray', linewidth=0.8, linestyle='--', zorder=1)
12
13 f = lambda t: 0.12 * t**2
14
15 textsize = 11
16 ss = 14
17 col = "black"
18 colp = "#3333FF"
19
20 plt.scatter(0, 2, marker='o', s=ss, color=colp, zorder=4)
21 plt.scatter(2.5, 3, marker='o', s=ss, color=colp, zorder=4)
22 plt.scatter(-2.5, 3, marker='o', s=ss, color=colp, zorder=4)
23 plt.scatter(4.3, f(4.3), marker='o', s=ss, color=colp, zorder=4)
24 plt.scatter(-4.3, f(4.3), marker='o', s=ss, color=colp, zorder=4)
25 plt.scatter(-5.2, 0.9, marker='o', s=ss, color=colp, zorder=4)
26 plt.scatter(5.2, 0.9, marker='o', s=ss, color=colp, zorder=4)
27 plt.scatter(2.5, f(2.5), marker='o', s=ss, color=colp, zorder=4)
28 plt.scatter(-2.5, f(2.5), marker='o', s=ss, color=colp, zorder=4)

```

```

29 plt.scatter(0.8, -1, marker='o', s=ss, color=colp, zorder=4)
30 plt.scatter(-0.8, -1, marker='o', s=ss, color=colp, zorder=4)
31
32
33 d = 0.1
34 c = 1.6
35 plt.arrow(1, -1.4, -1 * d, 2 * d, shape='full', color=col, head_width=0,
           lw=0.5, head_length=0.2, zorder=2)
36 plt.arrow(1, -1.4, -1 * c, 0.216 * c, shape='full', color=col, lw=0.5,
           head_width=0, head_length=0.2, zorder=2)
37
38 plt.fill_between(t, f(t), 5, color="lightblue", alpha=0.3)
39
40 plt.ylim(-3, 5)
41 plt.xlim(-9, 9)
42 plt.gca().set_aspect('equal')
43
44 # Masquer les chiffres des axes
45 plt.xticks([])
46 plt.yticks([])
47
48 # Garder et styliser les bords
49 for spine in plt.gca().spines.values():
50     spine.set_linewidth(0.5)
51     spine.set_color("gray")
52
53 plt.gcf().set_size_inches(10, 8)
54
55 plt.show()

```

Code Figure 3.2 : Solution de l'équation $f(S) = D$ avec f une fonction de type Haldane.

```

1  import numpy as np
2  import matplotlib.pyplot as plt
3
4  Sin=10
5  m0=4
6  ks=6
7  ki=1/2
8  D=0.35
9
10 S=np.arange(0, Sin, 0.01)
11 f=lambda S: m0 * (S / (ks + S + S**2 / ki) )
12
13 Sm=np.sqrt(ks*ki)
14 y_max=f(Sm)
15
16 delta=(D-m0)**2-4*(D/ki)*D*ks
17 lambda_1=( -(D-m0)- np.sqrt(delta) ) / ( 2*(D/ki) )
18 lambda_2=( -(D-m0)+ np.sqrt(delta) )/( 2*(D/ki) )
19
20 fig, ax = plt.subplots()
21
22 # Tracer la fonction
23 ax.plot(S, f(S), color="#0080FF", linewidth=1)

```

```

24
25 ax.plot(S,[D]*len(S),color="black", linewidth=1)
26
27 # Ajouter le point maximal
28 ax.scatter(Sm, y_max, color="#3333FF", s=10, zorder=3)
29 ax.scatter(Sm,0, color="#3333FF", s=10, zorder=3)
30 ax.scatter(0,y_max, color="#3333FF", s=10, zorder=3)
31
32 ax.scatter(lambda_1,D, color="#3333FF", s=10, zorder=3)
33 ax.scatter(lambda_1,0, color="#3333FF", s=10, zorder=3)
34 ax.scatter(lambda_2,D, color="#3333FF", s=10, zorder=3)
35 ax.scatter(lambda_2,0, color="#3333FF", s=10, zorder=3)
36
37 # Ajouter des lignes pointillees
38 ax.axvline(Sm, linestyle=":", color="gray", alpha=0.8)
39 ax.axhline(y_max, linestyle=":", color="gray", alpha=0.8)
40
41 # Ajouter des points
42 ax.plot([lambda_1, lambda_1], [0, D], linestyle=":", color="black", alpha
    =0.8)
43 ax.plot([lambda_2, lambda_2], [0, D], linestyle=":", color="black", alpha
    =0.8)
44
45 # Ajuster les limites pour que le graphe parte du coin (0,0)
46 ax.set_xlim([0, Sin])
47 ax.set_ylim([0, f(S).max() * 1.1])
48
49 # Supprimer les ticks des axes
50 ax.set_xticks([])
51 ax.set_yticks([])
52
53 ax.spines["top"].set_linewidth(0.5) # Modifier la largeur de l'axe
    superieur
54 ax.spines["right"].set_linewidth(0.5) # Modifier la largeur de l'axe
    droit
55 ax.spines["bottom"].set_linewidth(0.5) # Modifier la largeur de l'axe
    inferieur
56 ax.spines["left"].set_linewidth(0.5) # Modifier la largeur de l'axe
    gauche
57
58 # Mettre en gris les axes X et Y restants
59 ax.spines["bottom"].set_color("gray")
60 ax.spines["left"].set_color("gray")
61 ax.spines["right"].set_color("gray")
62 ax.spines["top"].set_color("gray")
63
64 # Afficher le graphique
65 plt.show()

```

Code Figure 3.3 : Diagramme opératoire d'un chémostat.

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 Sin = 10

```

```

5 m0 = 4
6 ks = 6
7 ki = 1 / 2
8
9 S = np.arange(0, Sin, 0.01)
10 f = lambda S: m0 * (S / (ks + S + S**2 / ki))
11
12 Sm = np.sqrt(ks * ki)
13 y_max = f(Sm)
14
15 fig, ax = plt.subplots()
16
17 # Tracer la fonction
18 ax.plot(S, f(S), color="#0080FF", linewidth=1)
19
20 # Ajouter le point maximal
21 ax.scatter(Sm, y_max, color="#3333FF", s=10, zorder=3)
22 ax.scatter(Sm, 0, color="#3333FF", s=10, zorder=3)
23 ax.scatter(0, y_max, color="#3333FF", s=10, zorder=3)
24
25 # Ajouter des lignes pointillees
26 ax.axvline(Sm, linestyle=":", color="gray", alpha=0.8)
27 ax.plot([0, Sm], [y_max, y_max], linestyle=":", color="gray", alpha=0.8)
28
29 # Ajouter une ligne horizontale continue de (S_m, f(S_m)) jusqu'a la fin
    droite du graphique
30 ax.hlines(y_max, Sm, Sin, colors="black", linestyles="-", linewidth=0.4)
31
32 # Remplir les regions colorees
33
34 # Region bleue : Au-dessus de la courbe f(S)
35 ax.fill_between(S[S <= Sm], f(S)[S <= Sm], y_max, color="lightblue",
    alpha=0.3, label="Region_bleue")
36 # Colorier la partie blanche au sommet du graphique en bleu
37 ax.fill_between(S, y_max, f(S).max() * 1.1, color="lightblue", alpha=0.3)
38
39 # Region jaune : Sous la courbe f(s)
40 ax.fill_between(S[S <= Sm], f(S)[S <= Sm], color="yellow", alpha=0.3,
    label="Region_1")
41 ax.fill_between(S[S >= Sm], 0, f(S)[S >= Sm], color="yellow", alpha=0.3,
    label="Region_verte")
42
43 # Region rouge : Au-dessus de la ligne horizontale et au-dessus de la
    courbe f(S)
44 ax.fill_between(S[S >= Sm], y_max, f(S)[S >= Sm], color="lightcoral",
    alpha=0.3, label="Region_rouge")
45
46 # Ajuster les limites pour que le graphe parte du coin (0,0)
47 ax.set_xlim([0, Sin])
48 ax.set_ylim([0, f(S).max() * 1.1])
49
50 # Supprimer les ticks des axes
51 ax.set_xticks([])
52 ax.set_yticks([])

```

```

53
54 ax.spines["top"].set_linewidth(0.5) # Modifier la largeur de l'axe
    superieur
55 ax.spines["right"].set_linewidth(0.5) # Modifier la largeur de l'axe
    droit
56 ax.spines["bottom"].set_linewidth(0.5) # Modifier la largeur de l'axe
    inferieur
57 ax.spines["left"].set_linewidth(0.5) # Modifier la largeur de l'axe
    gauche
58
59 # Mettre en gris les axes X et Y restants
60 ax.spines["bottom"].set_color("gray")
61 ax.spines["left"].set_color("gray")
62 ax.spines["right"].set_color("gray")
63 ax.spines["top"].set_color("gray")
64
65 plt.show()

```

Code Figure 5.1 : Simulation de l'extinction de la biomasse.

```

1  import numpy as np
2  import matplotlib.pyplot as plt
3  from scipy.integrate import solve_ivp
4  from scipy.interpolate import interp1d
5
6  # Parametres
7  nu=1
8  beta=1
9
10 # Generer le processus d'O-U par la methode d'Euler-Maruyama
11 def simulate_OU(N, dt, beta, nu, z0=0):
12     z = np.zeros(N+1)
13     z[0] = z0
14     for n in range(N):
15         dW = np.random.normal(0, np.sqrt(dt)) # increment de Wiener
16         z[n+1] = z[n] - beta * z[n] * dt + nu * dW
17     return z
18
19 # Parametres du systeme
20 s0 = 14
21 x0 = 5
22 s_in = 14
23 m0 = 4
24 ks = 7.5
25 ki = 4
26 D = 1.4
27 d = 0.25
28
29 # Fonction de croissance Haldane et la fonction Phi
30 f = lambda s: m0 * s / (ks + s + (s**2)/ki)
31 Phi = lambda z: (2 * d * np.arctan(z)) / np.pi
32
33 # Temps de simulation
34 dt = 0.01
35 T_total = 10

```

```

36 N = int(T_total / dt)
37 t_grid = np.linspace(0, T_total, N+1)
38 t_span = (0, T_total)
39 t_eval = t_grid
40 y0 = [s0, x0]
41
42 # Nombre de realisations a tracer
43 n_realizations = 5
44
45 # === Trace de plusieurs s(t) perturbés ===
46 fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(10, 6), sharex=True)
47
48 # Resolution de la dynamique NON PERTURBEE (reference en bleu et rouge)
49 def system1(t, y):
50     s, x = y
51     dsdt = D * (s_in - s) - f(s) * x
52     dxdt = -D * x + f(s) * x
53     return [dsdt, dxdt]
54
55 sol1 = solve_ivp(system1, t_span, y0, t_eval=t_eval)
56 ax1.plot(sol1.t, sol1.y[0], color='#0080FF', lw=1, linestyle="--", label=
57     's(t)_sans_bruit', zorder=3)
58 ax2.plot(sol1.t, sol1.y[1], color='#FF0000', lw=1, linestyle="--", label=
59     'x(t)_sans_bruit', zorder=3)
60
61 # Boucle sur différentes realisations
62 colors = ['#01C39C', '#FF8000', '#9F84D4', '#974D03', '#70696D']
63 for i in range(n_realizations):
64     Z = simulate_OU(N, dt, beta, nu)
65     z_func = interp1d(t_grid, Z, kind='linear', fill_value='extrapolate')
66
67     def system2(t, y):
68         s, x = y
69         D_eff = D + Phi(z_func(t))
70         dsdt = D_eff * (s_in - s) - f(s) * x
71         dxdt = -D_eff * x + f(s) * x
72         return [dsdt, dxdt]
73
74     sol2 = solve_ivp(system2, t_span, y0, t_eval=t_eval)
75     ax1.plot(sol2.t, sol2.y[0], color=colors[i], lw=0.7, alpha=0.9, zorder
76         =2)
77     ax2.plot(sol2.t, sol2.y[1], color=colors[i], lw=0.7, alpha=0.9, zorder
78         =2)
79
80 ax1.grid(); ax2.grid()
81 plt.tight_layout()
82 plt.show()

```

Code Figure 5.2 : Fonction de croissance $f(s)$ avec repères des valeurs D , D_l , et D_r dans le cas d'extinction de la biomasse.

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 Sin=14

```

```

5 m0=4
6 ks=7.5
7 ki=4
8 D=1.4
9 D1=1.15
10 Dr=1.65
11
12 S=np.arange(0,Sin*2,0.01)
13 f=lambda S: m0 * (S / (ks + S + S**2 / ki) )
14
15 Sm=np.sqrt(ks*ki)
16 y_max=f(Sm)
17
18 fig, ax = plt.subplots()
19
20 # Tracer la fonction
21 ax.plot(S, f(S), color="#0080FF", linewidth=1)
22
23 ax.plot(S,[D]*len(S),color="red", linewidth=0.75, linestyle="--")
24 ax.plot(S,[D1]*len(S),color="red", linewidth=0.75, linestyle="--")
25 ax.plot(S,[Dr]*len(S),color="red", linewidth=0.75, linestyle="--")
26
27 # Ajouter le point maximal
28 ax.scatter(Sm, y_max, color="#3333FF", s=10, zorder=3)
29 ax.scatter(Sm,0, color="#3333FF", s=10, zorder=3)
30 ax.scatter(0,y_max, color="#3333FF", s=10, zorder=3)
31 ax.scatter(Sin,0, color="#3333FF", s=10, zorder=3)
32 ax.scatter(Sin,f(Sin), color="#3333FF", s=10, zorder=3)
33 ax.scatter(0,f(Sin), color="#3333FF", s=10, zorder=3)
34
35 ax.plot([Sin, Sin], [0, f(Sin)], linestyle=":",lw=0.8, color="black",
        alpha=0.6)
36 ax.plot([Sm, Sm], [0, f(Sm)], linestyle=":",lw=0.8, color="black", alpha
        =0.6)
37 ax.plot([0,Sin], [f(Sin), f(Sin)], linestyle=":",lw=0.8, color="black",
        alpha=0.6)
38 ax.plot([0,Sm], [f(Sm), f(Sm)], linestyle=":",lw=0.8, color="black",
        alpha=0.6)
39
40 # Ajuster les limites pour que le graphe parte du coin (0,0)
41 ax.set_xlim([0, Sin *2])
42 ax.set_ylim([0, f(S).max() * 1.75])
43
44 # Supprimer les ticks des axes
45 ax.set_xticks([])
46 ax.set_yticks([])
47
48 ax.spines["top"].set_linewidth(0.5) # Modifier la largeur de l'axe
    superieur
49 ax.spines["right"].set_linewidth(0.5) # Modifier la largeur de l'axe
    droit
50 ax.spines["bottom"].set_linewidth(0.5) # Modifier la largeur de l'axe
    inferieur
51 ax.spines["left"].set_linewidth(0.5) # Modifier la largeur de l'axe

```

```

    gauche
52
53 # Mettre en gris les axes X et Y restants
54 ax.spines["bottom"].set_color("gray")
55 ax.spines["left"].set_color("gray")
56 ax.spines["right"].set_color("gray")
57 ax.spines["top"].set_color("gray")
58
59 # Afficher le graphique
60 plt.show()

```

Code Figure 5.3 : Simulation de la faible persistance de la biomasse.

```

1  import numpy as np
2  import matplotlib.pyplot as plt
3  from scipy.integrate import solve_ivp
4  from scipy.interpolate import interp1d
5
6  # Parametres
7  nu=1
8  beta=1
9
10 # Generer le processus d'O-U par la methode d'Euler-Maruyama
11 def simulate_OU(N, dt, beta, nu, z0=0):
12     z = np.zeros(N+1)
13     z[0] = z0
14     for n in range(N):
15         dW = np.random.normal(0, np.sqrt(dt)) # increment de Wiener
16         z[n+1] = z[n] - beta * z[n] * dt + nu * dW
17     return z
18
19
20 # Parametres du systeme
21 s0 = 16.7
22 x0 = 0.01
23 s_in = 16.7
24 m0 = 7
25 ks = 7
26 ki = 7
27 D = 1.7
28 d = 0.25
29
30 # Fonction de croissance Haldane et la fonction Phi
31 f = lambda s: m0 * s / (ks + s + (s**2)/ki)
32 Phi = lambda z: (2 * d * np.arctan(z)) / np.pi
33
34 # Temps de simulation
35 dt = 0.01
36 T_total = 50
37 N = int(T_total / dt)
38 t_grid = np.linspace(0, T_total, N+1)
39 t_span = (0, T_total)
40 t_eval = t_grid
41 y0 = [s0, x0]
42

```

```

43 # Nombre de realisations a tracer
44 n_realizations = 5
45
46 # Trace de plusieurs s(t) perturbés
47 fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(10, 6), sharex=True)
48
49 # Resolution de la dynamique NON PERTURBEE (reference en bleu et rouge)
50 def system1(t, y):
51     s, x = y
52     dsdt = D * (s_in - s) - f(s) * x
53     dxdt = -D * x + f(s) * x
54     return [dsdt, dxdt]
55
56 sol1 = solve_ivp(system1, t_span, y0, t_eval=t_eval)
57 ax1.plot(sol1.t, sol1.y[0], color='#0080FF', lw=1, linestyle="--", label=
58     's(t)_sans_bruit', zorder=3)
59 ax2.plot(sol1.t, sol1.y[1], color='#FF0000', lw=1, linestyle="--", label=
60     'x(t)_sans_bruit', zorder=3)
61
62 # Boucle sur différentes realisations
63 colors = ['#01C39C', '#FF8000', '#9F84D4', '#974D03', '#70696D']
64 for i in range(n_realizations):
65     Z = simulate_OU(N, dt, beta, nu)
66     z_func = interp1d(t_grid, Z, kind='linear', fill_value='extrapolate')
67
68     def system2(t, y):
69         s, x = y
70         D_eff = D + Phi(z_func(t))
71         dsdt = D_eff * (s_in - s) - f(s) * x
72         dxdt = -D_eff * x + f(s) * x
73         return [dsdt, dxdt]
74
75     sol2 = solve_ivp(system2, t_span, y0, t_eval=t_eval)
76     ax1.plot(sol2.t, sol2.y[0], color=colors[i], lw=0.7, alpha=0.9, zorder
77         =2)
78     ax2.plot(sol2.t, sol2.y[1], color=colors[i], lw=0.7, alpha=0.9, zorder
79         =2)
80
81 ax1.grid(); ax2.grid()
82 plt.tight_layout()
83 plt.show()

```

Code Figure 5.4 : Fonction de croissance $f(s)$ avec repères des valeurs D , D_l , et D_r dans le cas de la persistance faible de la biomasse.

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 Sin=16.7
5 m0=7
6 ks=7
7 ki=7
8 D=1.7
9 Dl=1.45
10 Dr=1.95

```

```

11
12 S=np.arange(0,Sin*2,0.01)
13 f=lambda S: m0 * (S / (ks + S + S**2 / ki) )
14
15 Sm=np.sqrt(ks*ki)
16 y_max=f(Sm)
17
18 fig, ax = plt.subplots()
19
20 # Tracer la fonction
21 ax.plot(S, f(S), color="#0080FF", linewidth=1)
22
23 ax.plot(S,[D]*len(S),color="red", linewidth=0.75, linestyle="--")
24 ax.plot(S,[Dl]*len(S),color="red", linewidth=0.75, linestyle="--")
25 ax.plot(S,[Dr]*len(S),color="red", linewidth=0.75, linestyle="--")
26
27 # Ajouter les points
28 ax.scatter(Sm, y_max, color="#3333FF", s=10, zorder=3)
29 ax.scatter(Sm,0, color="#3333FF", s=10, zorder=3)
30 ax.scatter(0,y_max, color="#3333FF", s=10, zorder=3)
31 ax.scatter(Sin,0, color="#3333FF", s=10, zorder=3)
32 ax.scatter(Sin,f(Sin), color="#3333FF", s=10, zorder=3)
33 ax.scatter(0,f(Sin), color="#3333FF", s=10, zorder=3)
34
35 ax.plot([Sin, Sin], [0, f(Sin)], linestyle=":",lw=0.8, color="black",
        alpha=0.6)
36 ax.plot([Sm, Sm], [0, f(Sm)], linestyle=":",lw=0.8, color="black", alpha
        =0.6)
37 ax.plot([0,Sin], [f(Sin), f(Sin)], linestyle=":",lw=0.8, color="black",
        alpha=0.6)
38 ax.plot([0,Sm], [f(Sm), f(Sm)], linestyle=":",lw=0.8, color="black",
        alpha=0.6)
39
40
41 # Ajuster les limites pour que le graphe parte du coin (0,0)
42 ax.set_xlim([0, Sin *2])
43 ax.set_ylim([0, f(S).max() * 1.1])
44
45 # Supprimer les ticks des axes
46 ax.set_xticks([])
47 ax.set_yticks([])
48
49 ax.spines["top"].set_linewidth(0.5) # Modifier la largeur de l'axe
    superieur
50 ax.spines["right"].set_linewidth(0.5) # Modifier la largeur de l'axe
    droit
51 ax.spines["bottom"].set_linewidth(0.5) # Modifier la largeur de l'axe
    inferieur
52 ax.spines["left"].set_linewidth(0.5) # Modifier la largeur de l'axe
    gauche
53
54 # Mettre en gris les axes X et Y restants
55 ax.spines["bottom"].set_color("gray")
56 ax.spines["left"].set_color("gray")

```

```
57 ax.spines["right"].set_color("gray")
58 ax.spines["top"].set_color("gray")
59
60 # Afficher le graphique
61 plt.show()
```