

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

N° attribué par la bibliothèque

--	--	--	--	--	--	--	--	--	--

THESE

pour obtenir le grade de

DOCTEUR DE L'INPG

Spécialité : “mathématiques appliquées”

préparée au laboratoire de modélisation et calcul (LMC)
dans le cadre de l'École Doctorale “mathématiques, sciences et technologies de
l'information, informatique”

présentée et soutenue publiquement

par

Laurent TOURNIER

le 30 novembre 2005

Titre :

**Etude et modélisation mathématique de réseaux
de régulation génétique et métabolique**

Directeur de thèse : Jean DELLA DORA

JURY

M. Anestis ANTONIADIS	Président du jury
M. Léon BRENIG	Rapporteur
M. Albert SORRIBAS	Rapporteur
M. Arnaud TONNELIER	Examineur
M. Jean DELLA DORA	Directeur de thèse

Table des matières

I	Vers une modélisation de la régulation génétique	11
1	Introduction à la notion de régulation génétique	13
1.1	Fondements biologiques du problème	14
1.1.1	Le dogme central de la biologie cellulaire	15
1.1.2	Equations de base de la cinétique enzymatique	18
1.1.3	Coopérativité et allostérie	23
1.2	Les réseaux de régulation génétique	29
1.2.1	Production d'énergie chez la bactérie e-coli	29
1.2.2	Fonctionnement de l'opéron lactose	30
1.2.3	Vers une modélisation de la régulation génétique	35
1.3	Différentes classes de modèles de systèmes biologiques	36
1.3.1	Les classes de systèmes dynamiques continus	37
1.3.2	La modélisation discrète	38
1.3.3	L'approche hybride	39
2	Modèle hybride de l'opéron lactose	41
2.1	L'outil des systèmes hybrides	41
2.1.1	Rappel de la définition d'un système hybride	41
2.1.2	Intérêt des systèmes hybrides pour la modélisation en biologie	42
2.1.3	Pouvoir algorithmique des systèmes hybrides : le calcul hybride	44
2.2	Modèle hybride de l'opéron lactose	47
2.2.1	Etablissement du modèle	47
2.2.2	Etude des équilibres	52
2.2.3	Interprétation biologique du modèle hybride	55
2.3	Calcul de trajectoires	56
2.3.1	Hybridisation et calcul des temps de sortie	56
2.3.2	Exemples d'exécutions du système	58
2.3.3	Limites du modèle	58
II	Réseaux de régulation et Systèmes dynamiques discrets	63
3	Les réseaux d'automates booléens	65
3.1	Rappels et définitions	66
3.1.1	Rappels sur les itérations sur un ensemble dénombrable	66
3.1.2	Vocabulaire de base de la théorie des systèmes dynamiques discrets	68
3.1.3	Le modèle Réseau d'Automates sur un Graphe	70

3.1.4	Exemple de graphe d'automates : les réseaux d'automates cellulaires . . .	72
3.2	Les réseaux d'automates booléens	76
3.2.1	Structure d'un réseau booléen	76
3.2.2	Dynamique d'un réseau booléen	79
3.2.3	Les modes opératoires parallèle, série et mixte	81
3.2.4	Application à la biologie	85
3.3	Etude de la dynamique des réseaux booléens	89
3.3.1	Outil métrique et dérivée discrète	89
3.3.2	Réseaux booléens aléatoires	91
3.3.3	Liens entre graphe de transition et graphe d'interaction	92
4	Inférence d'un réseau booléen	95
4.1	Présentation du problème	96
4.1.1	L'inférence de réseaux booléens	96
4.1.2	Connectivité réelle d'une fonction booléenne	97
4.1.3	Théorème d'inférence de réseau dans le cas synchrone	98
4.2	Existence d'une solution au problème d'identification	99
4.2.1	Entropie de Shannon et fonctions booléennes	99
4.2.2	Algorithme d'identification	104
4.2.3	Discussion sur la complexité	108
4.3	Le problème de l'identification partielle	111
4.3.1	Présentation du problème	111
4.3.2	Algorithme d'identification partielle	117
4.3.3	Résultats	124
4.3.4	Résolution du problème de consistance minimale à support fixé	126
III	Etude de systèmes dynamiques comportant des lois de puissances	133
5	Approximation de systèmes par des lois de puissance	135
5.1	La classe des S-systèmes	136
5.1.1	Définition	136
5.1.2	Discussion sur le modèle "S-systèmes"	137
5.1.3	Etude des points d'équilibres d'un S-système	140
5.2	S-approximation et recherche d'équilibres positifs	143
5.2.1	Approximation monomiale d'un champ de vecteurs positifs	143
5.2.2	Algorithme de recherche des points d'équilibres positifs	146
5.2.3	Analyse de stabilité de l'approximation par S-système	152
5.3	Résultats	154
5.3.1	Exemple en dimension 2 : bascule biologique	154
5.3.2	Un exemple en dimension 3 : comparaison avec la linéarisation	156
5.3.3	Discussion sur la vitesse de convergence de l'algorithme 4	158
6	Etude d'un réseau métabolique chez <i>Arabidopsis thaliana</i>	163
6.1	Présentation du modèle	164
6.1.1	La plante <i>Arabidopsis thaliana</i>	164
6.1.2	Réseau métabolique responsable de la production d'acides aminés	164
6.1.3	Le système TILV	166

6.1.4	Les équations de la dynamique	167
6.2	Etude du système dynamique TILV	169
6.2.1	Approche formelle	169
6.2.2	Approche par S-systèmes	169
Annexes		175
A Quelques propriétés mathématiques des fonctions de Hill		175
B Implémentation de l'identification de réseaux		179
Bibliographie		189

Notations employées

- \mathbb{N} désigne l'ensemble des entiers naturels.
- \mathbb{Z} désigne l'ensemble des entiers relatifs.
- \mathbb{R} désigne l'ensemble des nombres réels. On notera respectivement \mathbb{R}_+ et \mathbb{R}_+^* l'ensemble des réels positifs et des réels positifs stricts.
- \mathbb{R}^n et $(\mathbb{R}_+^*)^n$ désignent l'ensemble des vecteurs de dimension n dont les coordonnées sont réelles (respectivement strictement positives).
- Si Ω est une partie de \mathbb{R}^n , $\partial\Omega$ désigne sa frontière.
- $\mathfrak{M}_n(\mathbb{R})$ désigne l'ensemble des matrices carrées de dimension n dont les coefficients sont réels.
- $\mathfrak{M}_n(\{0,1\})$ désigne l'ensemble des matrices carrées de dimension n dont les coefficients sont booléens.
- A étant une matrice, on note A^T la matrice transposée de A .
- E étant un ensemble dénombrable, $\text{card}(E)$ désigne son cardinal et $\mathcal{P}(E)$ désigne l'ensemble des parties de E .
- E et F étant deux ensembles quelconques, $E \setminus F$ désigne la différence entre E et F , c'est-à-dire l'ensemble des éléments de E qui n'appartiennent pas à F .
- E et F étant deux ensembles quelconques, l'ensemble des applications de E dans F sera noté F^E ou $E \rightarrow F$.
- L'application qui à tout élément de l'ensemble E associe lui-même est notée id_E .

Pour les calculs de complexité ou de convergence, nous serons amenés à utiliser les notations de Landau : o et O .

Lors de l'expression d'équations différentielles, la dérivée d'une variable x par rapport à la variable temporelle t sera notée $\dot{x}(t)$ ou abusivement \dot{x} .

Les expressions booléennes utilisent les symboles classiques :

- \vee : “ou” logique,
- \wedge : “et” logique,
- \neg : “non” logique.

Introduction

La compréhension des systèmes issus du vivant constitue l'un des défis majeurs de ce début de siècle. Devant l'ampleur et la complexité de la tâche, toutes les disciplines scientifiques sont concernées. La biologie, bien sûr, et notamment la biologie cellulaire née au cours du vingtième siècle, sont au cœur de la question. La technologie avançant sans cesse de manière plus rapide, nous disposons actuellement de bases de données gigantesques encore inimaginables quelques décennies auparavant. Avec l'amélioration des techniques d'expérimentations, il est aujourd'hui possible de pénétrer au cœur d'une cellule, qui est l'entité biologique de base de tout être vivant, d'examiner son patrimoine génétique, de mesurer certains de ses constituants principaux et aussi et surtout de commencer à entrevoir l'immense complexité des interactions entre ces constituants, dont l'organisation est à la base même de la vie. Si l'on commence aujourd'hui à comprendre comment le patrimoine génétique d'une cellule peut influencer sur son métabolisme, comment appréhender cette influence lorsque la cellule comporte plusieurs milliers, voire dizaines de milliers de gènes ? Comment alors ajouter à cela l'extraordinaire capacité des êtres vivants pour interagir avec leur environnement ainsi que leur capacité à évoluer au cours des générations ?

Les biologistes disposent aujourd'hui d'outils de modélisation puissants, qui permettent de simuler avec beaucoup de précision des systèmes de plus en plus grands. Néanmoins, étant donnée la masse de connaissances croissante dont on dispose aujourd'hui, la dimension et la complexité de nombreux systèmes réels augmentent jusqu'à atteindre les limites des outils les plus performants. L'approche selon laquelle on cherche à isoler un système particulier et à en faire une analyse la plus détaillée possible constitue certes une avancée importante vers la compréhension des mécanismes cellulaires, mais elle n'est plus suffisante. Une autre approche, propre aux mathématiques, consiste à tenter d'abstraire les informations de plus en plus précises apportées par la première approche afin de créer des classes de modèles, plus ou moins éloignées de la réalité, mais dont le traitement mathématique pourrait permettre à terme de répondre à des questions globales, parfois éloignées des systèmes initiaux. La création de ces modèles nécessite évidemment de fortes hypothèses simplificatrices. Il serait illusoire de penser que seules des hypothèses mathématiques, liées à la forme des équations que l'on veut manipuler sont nécessaires. En effet, la construction de modèles de systèmes biologiques ne saurait se faire sans une bonne connaissance des phénomènes que nous voulons représenter et surtout sans la connaissance des contraintes imposées par la biologie. Cette construction n'est donc pas une fin en soi, mais découle d'un va-et-vient entre abstraction et expérimentation.

Cette thèse s'inscrit dans cette démarche, avec bien entendu l'humilité qu'impose toute tentative de modélisation de phénomènes aussi complexes. Plus précisément, nous nous intéressons dans ce mémoire à différentes modélisations du phénomène, découvert au milieu du vingtième siècle, de *régulation génétique*. Les modèles que nous étudions sont des classes de systèmes dynamiques déterministes et de systèmes hybrides. La littérature dans ce domaine est riche et

variée et concerne aussi bien les mathématiques appliquées que l'informatique.

La première partie de cette thèse est consacrée à une introduction de la notion de régulation génétique. N'étant pas biologiste, je me contenterai d'en faire une description forcément partielle, en insistant essentiellement sur les points qui me semblent importants dans l'optique de modélisation du phénomène. Le chapitre 1 contient une description d'un réseau de régulation particulier : l'opéron lactose. Il s'agit d'un exemple historique puisque c'est le premier réseau de régulation génétique découvert, par les chercheurs F. Jacob, A. Lwoff et J. Monod dans les années 60. Nous ferons dans ce chapitre un rapide état de l'art sur les modèles de réseaux de régulation génétique existants. Le chapitre 2 poursuit l'étude de l'opéron lactose, en proposant un modèle de cette structure sous la forme d'un système hybride. Nous verrons à cet effet que la théorie relativement récente des systèmes hybrides constitue un outil tout-à-fait intéressant de modélisation pour de tels systèmes.

Dans la seconde partie, nous proposons une étude d'une modélisation beaucoup employée au cours des dernières décennies : l'approche discrète. Cette approche consiste à simplifier de façon drastique les systèmes, afin de les représenter par des réseaux d'automates. Si cette simplification nous éloigne de la réalité biologique, elle nous permet en revanche de répondre à des questions spécifiques, que les autres approches peinent à aborder. Le chapitre 3 pose les bases mathématiques de notre étude, en reprenant la définition très générale de graphe d'automates. Nous verrons notamment dans ce chapitre que ces systèmes peuvent avoir des dynamiques qui sont loin d'être simples. Nous nous intéresserons dans le chapitre 4 au problème de l'inférence de réseaux, qui fait partie des questions globales posées par la biologie (notamment avec la récente technologie des biopuces à ADN) et qui peut également s'employer dans d'autres disciplines. Nous reprenons dans ce chapitre un algorithme existant dans la littérature sous le nom de REVEAL. Nous proposons une preuve de cet algorithme, ainsi que le calcul de sa complexité, dans le cas complet et dans le cas partiel. Dans ce dernier cas, nous faisons une étude probabiliste qui nous permet de relier la taille des données avec l'efficacité de l'inférence. L'implémentation de cet algorithme nous permet également de proposer un ensemble de résultats numériques étayant notre étude théorique.

Enfin, la troisième et dernière partie de ce mémoire est consacrée à l'étude d'une classe de systèmes continus souvent employés pour la modélisation : les S-systèmes. Ces systèmes font partie d'une classe plus générale : les systèmes quasi-monomiaux. L'étude des S-systèmes a été menée à la suite d'une collaboration avec G. Curien, biologiste au CEA, qui nous a proposé un travail sur un réseau métabolique chez *Arabidopsis thaliana*. Dans le chapitre 5, consacré à l'étude des S-systèmes, nous proposons un algorithme (présenté durant l'été 2005 à la conférence de calcul formel ISSAC [79]) dont le but est de trouver les points d'équilibre positifs d'un système d'équations différentielles quelconques. Ce faisant, nous trouvons une approximation sous forme de S-système du système initial autour de cet équilibre. Une étude de cette approximation est menée. Enfin, le chapitre 6 présente le travail encore en cours de l'étude du réseau TILV, qui est un réseau métabolique responsable de la synthèse de certains acides aminés chez *Arabidopsis*.

Première partie

Vers une modélisation de la
régulation génétique

Chapitre 1

Introduction à la notion de régulation génétique

Le vingtième siècle a connu nombre de découvertes qui ont fait des sciences du vivant l'un des domaines les plus productifs en termes de recherche scientifique et technologique. Depuis 1953 et le premier modèle de la structure de la molécule d'acide désoxyribo-nucléique (ADN) par Watson et Crick, le chemin parcouru est immense. Moins de 40 années plus tard, en 1990, débutait le projet mondial HGP¹ de séquençage du génome humain. Achievé en 2003, ce projet, appuyé sur quatre décennies de progrès technologiques, a permis de constituer une base de données gigantesque référençant l'ensemble des gènes de l'espèce humaine. On trouve un autre exemple de l'avancée technique en biologie dans la récente technologie des *biopuces à ADN* qui permettent de visualiser à un instant donné le niveau d'expression d'un ensemble de plusieurs milliers de gènes. Dans le cas d'organismes simples (tels que la levure *saccharomyces cerevisiae*), l'ensemble du génome peut même être représenté. L'utilisation de ces biopuces a notamment pour but de générer de multiples bases de données dont le traitement statistique permettra à terme de nombreuses applications dans des domaines aussi variés que l'agro-alimentaire, la pharmacologie et la médecine.

Grâce aux nombreux travaux entrepris depuis environ un demi-siècle, la somme des connaissances en biologie cellulaire s'est donc vue accroître de manière phénoménale. Parallèlement à ces recherches, de plus en plus de biologistes, physiciens et mathématiciens appliqués proposent des modèles dont le but est d'intégrer, d'*organiser* autant que possible la masse de connaissances et de données accumulées. Dans cette optique, la démarche propre aux mathématiques appliquées consiste à comparer ces modèles, les critiquer et le plus souvent à les simplifier afin d'apporter des outils abstraits ayant pour but d'améliorer la compréhension des phénomènes biologiques représentés. L'un des exemples les plus frappants de cette démarche est sans doute la théorie des réseaux neuronaux. Cet exemple démontre que dès les années 40 (et plus précisément en 1943 avec MacCulloch et Pitts [47]), de nouveaux types de modèles apparaissent, se basant à la fois sur la biologie et l'informatique. Les deux biophysiciens MacCulloch et Pitts proposent une abstraction du neurone physiologique sous la forme d'un automate à seuils. Ce neurone formel, issu d'une simplification de son homologue biologique, ouvre la voie à une nouvelle branche de recherche dont le succès n'est aujourd'hui plus à démontrer. Avec les travaux de Hebb ou

¹HGP : Human Genome Project. Projet de recherche à l'échelle mondiale qui se fixe pour but d'identifier l'ensemble des quelques vingt à vingt-cinq mille gènes présents chez l'homme. Voir http://www.ornl.gov/sci/techresources/Human_Genome/home.shtml

plus récemment de Hopfield [33], les *réseaux de neurones artificiels* sont devenus un outil aux applications multiples (citons entre autres l'*apprentissage* qui permet notamment de fournir un outil puissant de classification de données) débordant de beaucoup la seule application à la biologie.

Aujourd'hui, l'intérêt pour les sciences du vivant est toujours croissant en mathématiques appliquées. On pourra par exemple se référer à l'ouvrage de J.D. Murray [54] pour nous rendre compte de la quantité d'exemples biologiques donnant lieu à une étude mathématique pertinente. Néanmoins, de nombreuses zones d'ombre demeurent, particulièrement dans la modélisation des mécanismes globaux ayant lieu dans la cellule, qui est l'entité biologique de référence. La complexité et le nombre des processus biochimiques en jeu sont tels que les modèles actuels peinent à rendre compte de la grande diversité des comportements dynamiques exhibés par l'ensemble des êtres vivants. Dans un futur plus ou moins proche, l'un des principaux défis lancés aux différentes disciplines scientifiques reste l'établissement de modèles abstraits permettant d'appréhender de façon toujours plus pertinente l'organisation globale des différentes tâches cellulaires qui sont à la base de la vie.

Dans ce mémoire, nous nous concentrons plus particulièrement sur la question de la régulation génétique, dont les bases sont posées dans les années 60 par les chercheurs A. Lwoff, F. Jacob et J. Monod qui pour la première fois décrivent un ensemble de faits biologiques permettant d'imaginer une structure de régulation de l'expression de certains gènes de la bactérie *Escherichia coli* [35]. Cette structure, portant le nom d'opéron lactose valut à ses découvreurs le prix Nobel de médecine en 1965. Le concept d'opéron a depuis été affiné, d'autres structures du même type ayant été isolées. Toutefois même si la notion de *régulation génétique* reste un champ de recherche classique en biologie cellulaire, la modélisation de ce phénomène demeure un sérieux et épineux problème. Dans la suite de ce chapitre, nous donnons une description (non exhaustive, cela va de soi) des réseaux de régulation génétique. Nous proposons également un ensemble de remarques montrant la grande difficulté de toute tentative de formalisation dans ce domaine. Nous évoquerons par la suite quelques unes des différentes approches entreprises au cours des cinquante dernières années (et toujours d'actualité) visant à définir des *classes* de systèmes mathématiques exhibant des comportements censés "s'approcher" des comportements réels.

Dans ce chapitre, nous allons décrire le phénomène de régulation génétique tel qu'il est connu aujourd'hui. Bien entendu, nous ne prétendons aucunement à l'exhaustivité de notre description. Pour une étude plus complète des mécanismes biologiques en jeu, nous pourrions nous référer à des ouvrages spécialisés comme par exemple [45]. Nous nous attacherons essentiellement à tirer de nos différentes lectures un ensemble de faits biologiques utiles du point de vue de la modélisation. Nous tenterons également d'expliquer l'idée de *programme génétique* qui nous semble l'un des problèmes les plus intéressants, et également les plus complexes posés par la biologie cellulaire aujourd'hui.

1.1 Fondements biologiques du problème

Nous devons avant tout comprendre le fonctionnement de base de la cellule et le rôle que jouent les gènes dans ce fonctionnement. Pour cela, nous allons décrire les mécanismes fondamentaux de la biologie cellulaire qui sont à la base de l'expression des gènes. Nous verrons ensuite quelques éléments de compréhension du métabolisme d'une cellule. Ceci nous permettra d'expliquer, à partir d'exemples réels, la notion de régulation génétique.

1.1.1 Le dogme central de la biologie cellulaire

Avant de définir ce qu'est un gène et comment il intervient dans la vie cellulaire, nous rappelons brièvement ici la structure de la molécule d'ADN. Le premier modèle de l'acide désoxyribonucléique (ou ADN) a été proposé par Wilkins, Watson et Crick qui, en 1953, rassemblent toutes les données existantes sur l'ADN et proposent une structure en double brin, organisée en hélice (voir figure 1.1). Si on l'observe de plus près, un brin d'ADN est constitué d'une chaîne de structures moléculaires appelées nucléotides. Le composant principal des nucléotides est une base azotée. Il existe quatre bases différentes dans la nature : l'adénine (A), la thymine (T), la cytosine (C) et la guanine (G). Les deux brins sont liés entre eux au niveau de chacune des bases par des liaisons faibles (liaisons hydrogène), chaque base se liant avec celle qui lui correspond en suivant une règle précise de complémentarité : A se lie avec T et C avec G . La figure 1.1 montre une portion de molécule d'ADN schématisée dans laquelle on peut observer la règle de complémentarité.

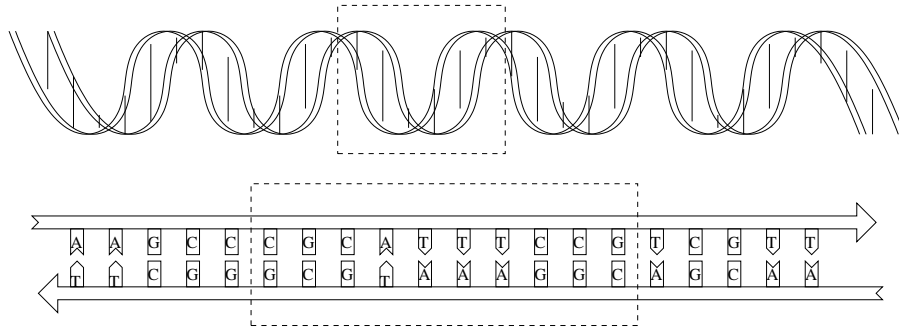


FIG. 1.1: Représentations schématisées d'un segment de molécule d'ADN. En haut : structure en double hélice. En bas : exemple de séquence de bases suivant la règle de complémentarité A-T C-G. Un pas d'hélice correspond environ à une dizaine de paires de bases.

Dans les cellules *eucaryotes*, les molécules d'ADN se trouvent dans un compartiment spécifique de la cellule appelé noyau. Elles sont la plupart du temps compactées et prennent la forme de "batonnets" appelés *chromosomes*. Leur structure en séquence de paires de bases permet un stockage d'information. La taille d'une molécule d'ADN, c'est-à-dire le nombre de paires de bases qu'elle contient, est très variable selon les espèces. Ainsi, le génome humain compte quelques trois milliards de bases alors que celui d'une bactérie ne se compte qu'en millions. D'un point de vue formel, une molécule d'ADN est donc un enchaînement fini de *lettres* sur un alphabet comportant seulement quatre éléments : $\mathcal{A} = \{A, T, C, G\}$. Ce que l'on nomme *gène* est une portion généralement continue de cette longue séquence, comportant une *unité* d'information. Les gènes peuvent donc être vus comme des *mots* sur \mathcal{A} . Chacun de ces mots a pour fonction de stocker et de transmettre aux cellules descendantes le code de fabrication d'un type de *protéine*. Les protéines sont des macromolécules organiques aux fonctions très diverses. Afin d'accomplir les tâches nécessaires à sa survie, une cellule a en permanence besoin de protéines, chacune remplissant un rôle spécifique. Par exemple, les enzymes, qui forment une sous-classe des protéines, sont des catalyseurs biochimiques, c'est-à-dire qu'ils permettent

d'accélérer certaines réactions chimiques au sein du milieu intracellulaire. Les protéines sont des complexes formés d'une chaîne d'éléments appelés acides aminés. On dénombre à environ une vingtaine les acides aminés constituant l'ensemble des protéines de la biosphère.

Nous avons dit que chaque gène du génome contient l'information nécessaire à la synthèse d'un type de protéine. Cette synthèse implique un ensemble de processus biochimiques constituant le *dogme central* de la biologie cellulaire. Le passage du gène à la protéine se fait en deux étapes. Tout d'abord, grâce à une enzyme particulière, l'ARN-polymérase, la séquence du gène est recopiée sur une molécule nommée ARN (acide ribonucléique) dont la structure en séquence est identique à celle de l'ADN, si ce n'est qu'elle n'est composée que d'un seul brin. Cette phase s'appelle la *transcription*. Le terme "recopier" cache un mécanisme biochimique complexe au cours duquel la polymérase sépare localement les deux brins d'ADN au niveau des liaisons hydrogènes entre les paires de bases situées au début du gène, puis parcourt l'un des deux brins (appelé brin codant) en fabriquant au fur et à mesure son complémentaire en suivant la règle de complémentarité des bases². Une fois arrivée à la fin du gène, la polymérase referme la molécule d'ADN et libère l'ARN transcrite (voir figure 1.2). La molécule d'ARN ainsi constituée porte le nom d'ARN messenger (ARN-m) car dans les cellules eucaryotes, elle se déplace du noyau vers le cytoplasme, alors que les molécules d'ADN restent en permanence dans le noyau.

D'un point de vue formel, l'étape de transcription est simple à comprendre. Nous avons dit qu'un gène peut être vu comme un mot g sur l'alphabet $\mathcal{A} = \{A, T, C, G\}$ (en ne considérant que le brin codant, l'autre étant son complémentaire). La séquence de nucléotides de l'ARN messenger est simplement le complémentaire de ce mot, dans lequel la thymine T est remplacée par l'uracile U . l'ARN-m est donc la transcription du mot g sur l'alphabet $\mathcal{A}' = \{A, U, C, G\}$. Il n'y a donc pas d'interprétation de l'information contenue dans g mais juste un changement de support chimique.

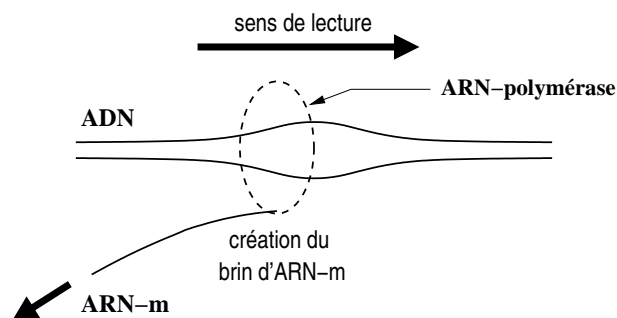


FIG. 1.2: Représentation schématique de l'étape de transcription : création de l'ARN messenger à partir du brin codant de la molécule d'ADN.

En revanche, une fois l'ARN messenger produit, la synthèse de la protéine nécessite de *traduire* cette information en une séquence d'acides aminés, c'est-à-dire dans un alphabet d'une vingtaine de lettres. Un code est alors utilisé, c'est-à-dire une loi de passage d'un alphabet à

²dans l'ARN, la thymine est remplacée par un autre nucléotide, l'uracile. Ce changement est sans importance d'un point de vue formel car l'uracile U présente la même complémentarité à l'adénine que la thymine.

l'autre. Un rapide calcul montre qu'au minimum trois bases sont nécessaires pour coder un acide aminé. Ces triplets de bases portent le nom de codons. Il y a $(\text{card } \mathcal{A})^3 = 64$ codons possibles. On sait aujourd'hui que le code est redondant, c'est-à-dire qu'un même acide aminé peut être codé par plus d'un codon. On sait également que parmi les 64 codons, trois ne codent pour aucun acide aminé et ont un rôle particulier : il s'agit de signaux dont le but est de stopper l'étape de traduction, on les appelle des codons "stop". L'étape de traduction se déroule dans le cytoplasme cellulaire (c'est-à-dire pour les cellules eucaryotes à l'extérieur du noyau) par l'intermédiaire d'agents appelés ribosomes. Ces agents agissent comme des "têtes de lecture" en parcourant la molécule d'ARN messager et en associant à chaque codon l'acide aminé pour lequel il code. De proche en proche, la séquence d'acides aminés (appelée chaîne polypeptidique) se construit (voir figure 1.3). Une fois terminée, elle se détachera du ribosome et pourra alors prendre une configuration spatiale qui lui est propre³ et devenir ainsi une protéine.

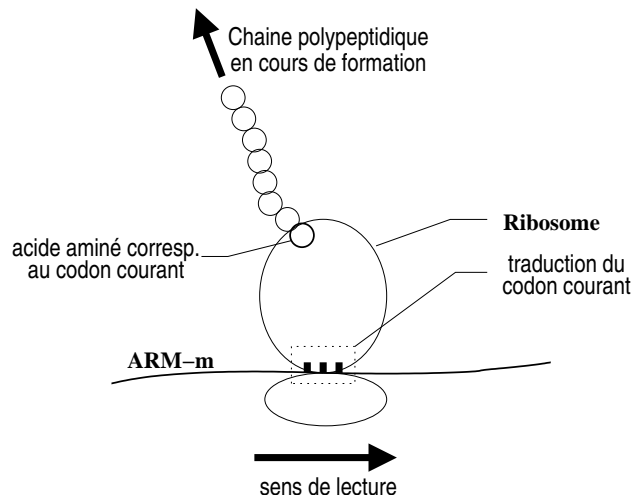


FIG. 1.3: Représentation schématique de l'étape de traduction : le brin d'ARN-m est lu codon par codon et la chaîne polypeptidique se construit au fur et à mesure.

A ce stade, nous savons donc que chaque cellule contient un patrimoine génétique contenu sur une ou plusieurs molécules d'ADN. Ce patrimoine est composé de gènes qui sont des séquences de paires de nucléotides complémentaires. Chaque gène contient dans sa séquence le code de fabrication d'un type de protéine spécifique. Cette rapide description permet de nous familiariser avec les mécanismes biologiques mis en œuvre par la cellule pour *exprimer* son patrimoine génétique, c'est-à-dire pour synthétiser les protéines à partir de l'information contenue dans les gènes. Notre description reste pour le moment statique : nous comprenons comment la cellule procède, il nous reste à déterminer dans quelle mesure elle commande ce processus. En effet, les protéines sont des molécules qui ont un temps de vie limité dans le temps, elles doivent donc être synthétisées de manière permanente si nous voulons les maintenir à un niveau constant. Étant donné le grand nombre de gènes d'une cellule, il semble clair que tous

³La formation spatiale de la protéine à partir de la chaîne polypeptidique est un phénomène extrêmement complexe que nous ne décrivons pas dans ce mémoire.

ne sont pas exprimés en permanence. Cette idée est confortée par le fait que chaque type de protéine a un (ou parfois plusieurs) rôle qui lui est bien spécifique. Au cours des différentes phases de la vie cellulaire, l'expression d'un gène peut donc tour à tour être inutile ou bien capitale pour la survie de la cellule. Chez des organismes pluricellulaires dans lesquels les cellules sont différenciées, comme chez l'homme, toutes les cellules d'un même individu contiennent exactement le même bagage génétique. Or les gènes spécifiques à un certain type de cellule (un neurone, une cellule de la peau ou encore une cellule du foie par exemple) n'ont à priori aucune raison d'être exprimés dans un autre type de cellules. Tous ces arguments tendent à infléchir l'idée que l'expression des gènes doit être *contrôlée*, en d'autres termes qu'il doit exister dans chaque cellule un ou plusieurs mécanismes permettant de déterminer à chaque instant quels sont les gènes dont l'expression est inutile et au contraire quels sont ceux dont l'expression est importante ou même vitale.

Afin d'aller plus avant dans la notion de régulation génétique et avant de voir sur des exemples comment se traduit cette régulation à l'échelle de la cellule, nous proposons dans ce qui suit une brève description de certaines propriétés du métabolisme d'une cellule.

1.1.2 Equations de base de la cinétique enzymatique

Dans notre description, nous avons souvent évoqué la présence de processus biochimiques ayant lieu dans la cellule. Nous avons notamment évoqué l'action de catalyseur biochimique des *enzymes*. La cellule est de fait le siège de nombreuses réactions chimiques qui ont lieu simultanément, et beaucoup de ces réactions font intervenir des enzymes. Il est donc essentiel d'avoir une bonne compréhension du phénomène de catalyse enzymatique avant de poursuivre l'étude de la régulation génétique.

Exprimer la dynamique de ces réactions est un problème difficile. Nous disposons aujourd'hui de *modèles* qui fournissent une simplification de leur dynamique réelle. Cette simplification est, comme nous l'avons dit en introduction, une étape obligée vers la compréhension du phénomène. Nous proposons à présent l'étude d'une réaction enzymatique simple, impliquant un seul enzyme (nous nous référons notamment à [54]).

Cette étude a été menée pour la première fois par Michaelis et Menten en 1913. Nous considérons trois espèces en présence : le substrat noté S , l'enzyme E et le produit P . Le schéma proposé par Michaelis et Menten est le suivant (voir [54]) :



Le substrat se lie à l'enzyme pour former un complexe intermédiaire noté (SE) dans une réaction réversible. Ce complexe (SE) est ensuite transformé en produit, en libérant l'enzyme. Les quantités k_1 , k_{-1} et k_2 sont des constantes qui ne dépendent que du type de réaction et des conditions de température et de pression.

En notant respectivement s , e , c et p les concentrations en substrat, en enzyme, en complexe et en produit, la loi d'action de masse nous donne le système différentiel suivant :

$$\begin{cases} \dot{s}(t) &= -k_1 e(t)s(t) + k_{-1}c(t) \\ \dot{e}(t) &= -k_1 e(t)s(t) + (k_{-1} + k_2)c(t) \\ \dot{c}(t) &= k_1 e(t)s(t) - (k_{-1} + k_2)c(t) \\ \dot{p}(t) &= k_2 c(t) \end{cases} \quad (1.2)$$

auquel on associe la condition initiale :

$$s(0) = s_0 \in \mathbb{R}_+^*, \quad e(0) = e_0 \in \mathbb{R}_+^*, \quad c(0) = 0, \quad p(0) = 0$$

Dans le système (1.2), la variable p est entièrement déterminée lorsque l'on connaît la variable c :

$$\forall t > 0, \quad p(t) = k_2 \int_0^t c(t') dt' \quad (1.3)$$

De plus, en additionnant la deuxième et la troisième équation, on obtient :

$$\forall t \geq 0, \quad \dot{e}(t) + \dot{c}(t) = 0 \quad (1.4)$$

ce qui entraîne que pour tout t , $e(t) = e_0 - c(t)$.

Ceci implique que la quantité d'enzyme (sous sa forme libre et sous la forme du complexe enzyme-substrat) reste constante au cours du temps. Dans ce modèle, l'enzyme n'est donc ni produit, ni dégradé⁴ par un processus annexe. Grâce à (1.3) et (1.4), la résolution du système (1.2) se ramène donc à la résolution du système bidimensionnel suivant (voir [54]) :

$$\begin{cases} \dot{s}(t) &= -k_1 e_0 s(t) + (k_{-1} + k_1 s(t)) c(t) \\ \dot{c}(t) &= k_1 e_0 s(t) - (k_1 s(t) + k_{-1} + k_2) c(t) \end{cases} \quad (1.5)$$

En appliquant à ce système l'adimensionnalisation suivante :

$$\tau = k_1 e_0 t, \quad u(\tau) = \frac{s(t)}{s_0}, \quad v(\tau) = \frac{c(t)}{e_0}$$

puis en posant $\lambda = \frac{k_2}{k_1 s_0}$, $K = \frac{k_{-1} + k_2}{k_1 s_0}$ et $\varepsilon = \frac{e_0}{s_0}$, le système (1.5) se réécrit :

$$\begin{cases} \frac{du}{d\tau} &= -u + (u + K - \lambda)v \\ \varepsilon \frac{dv}{d\tau} &= u - (u + K)v \end{cases} \quad (1.6)$$

avec les conditions initiales suivantes :

$$u(0) = 1, \quad v(0) = 0. \quad (1.7)$$

La résolution du système (1.6) pose deux difficultés mathématiques majeures. Il comporte d'une part des termes quadratiques qui empêchent une résolution analytique directe. La seconde difficulté réside dans le terme $\varepsilon \frac{dv}{d\tau}$ de la deuxième équation qui induit, lorsque ε prend des valeurs proches de 0 un phénomène de *couche limite*. Ce phénomène provient du fait que pour de petites⁵ valeurs de ε , la variable v qui représente la concentration en complexe a, au voisinage de $\tau = 0$, une dynamique *rapide* relativement à celle du substrat u (voir l'allure des courbes sur la figure 1.4).

Pour résoudre le système (1.6), la technique, classique en théorie des perturbations singulières (voir [32]), est de considérer des développements asymptotiques de u et de v sous la forme :

$$\begin{cases} u(\tau, \varepsilon) &= \sum_{n=0}^{+\infty} \varepsilon^n u_n(\tau) \\ v(\tau, \varepsilon) &= \sum_{n=0}^{+\infty} \varepsilon^n v_n(\tau) \end{cases}$$

⁴Cette hypothèse sera discutée par la suite

⁵Le paramètre $\varepsilon = \frac{e_0}{s_0}$ peut raisonnablement être considéré comme très faible (de l'ordre de 10^{-2} par exemple), car l'enzyme n'est pas *consommé* au cours de la réaction (la cellule a donc besoin de peu de molécules d'enzyme relativement aux molécules de substrat).

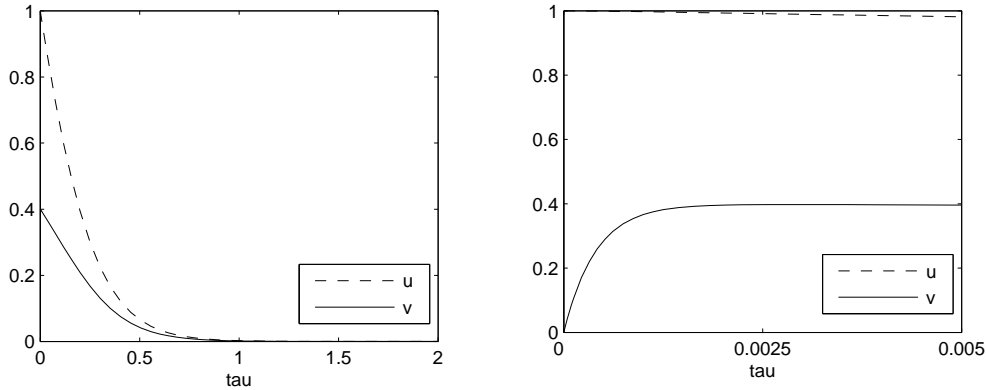


FIG. 1.4: Allure des courbes solutions du système (1.6)-(1.7) avec $K = 1.5$, $\lambda = 10$ et $\varepsilon = 10^{-3}$. Sur les courbes de gauche, on observe que presque instantanément, le complexe v croît jusqu'à une valeur proche de 0.4 (c'est la dynamique rapide). A droite, un zoom sur ces courbes au voisinage de $\tau = 0$ permet de mieux visualiser le phénomène de couche limite. Ces courbes sont obtenues par un solveur numérique classique (*ode15s* de *Matlab*).

En injectant ces expressions dans (1.6) et en ne retenant que les termes u_0 et v_0 d'ordre 0 en ε , nous obtenons le système :

$$\begin{cases} \frac{du_0}{d\tau} &= -u_0 + (u_0 + K - \lambda)v_0 \\ 0 &= u_0 - (u_0 + K)v_0 \end{cases} \quad (1.8)$$

Ne retenir que les termes d'ordre 0 en ε revient à considérer que ε est nul. Ceci explique pourquoi la seconde équation n'est plus différentielle mais algébrique. Le système (1.8) est donc algébro-différentiel : l'équation algébrique est l'équation d'une courbe \mathcal{C} dans l'espace des phase (u_0, v_0) et l'équation différentielle scalaire nous donne la dynamique de $u_0(\tau)$ sur \mathcal{C} . Le problème de cette approximation est qu'elle ne vérifie pas la condition initiale $u_0(0) = 1$ et $v_0(0) = 0$. En fait, cette approximation revient à ignorer la dynamique rapide de v au voisinage de $\tau = 0$ que nous avons évoquée plus haut. Elle porte le nom de solution *extérieure* car elle fournit une bonne approximation de la solution dès que τ prend des valeurs suffisamment éloignées de l'origine.

Remarquons ici qu'il est possible de trouver une approximation *intérieure* qui approche la solution au voisinage de l'origine. Sans entrer dans le détail technique, signalons simplement que par un changement de variable temporelle puis en utilisant les développements de u et v aux ordres 0 et 1, nous pouvons trouver l'expression d'une approximation des solutions valable lorsque τ est proche de 0. Vient ensuite une étape de *raccordement* des approximations extérieure et intérieure afin de donner une approximation de la solution globale (i.e. pour tout τ). Le lecteur trouvera dans [54] une étude détaillée de cette résolution.

L'intérêt de cette étude est évident d'un point de vue mathématique : il s'agit de donner une résolution du système initial (1.2) la plus complète possible. Néanmoins, du point de vue de la modélisation biologique, la solution globale constituée du raccordement entre approximations extérieure et intérieure reste un objet mathématique relativement complexe dont l'utilisation

peut, suivant l'application choisie et le phénomène étudié, rendre le modèle obscur et plus difficile à manipuler. Le plus souvent, la modélisation d'une application biologique demande de faire à plusieurs niveaux des hypothèses simplificatrices. Ces hypothèses doivent bien entendu être clairement exprimées, de manière à pouvoir être éventuellement relaxées par la suite. Il s'en suit un va-et-vient constant entre la définition du modèle, sa simulation et l'expérimentation biologique. La démarche de l'établissement du modèle n'est donc pas une fin en soi mais, plus humblement, une construction permanente, basée sur une succession d'hypothèses réductrices, dont le but est de permettre d'avancer progressivement vers la compréhension du phénomène.

En ce qui concerne la réaction enzymatique (1.1), l'hypothèse proposée par Michaelis et Menten en 1913 reste la plus couramment utilisée en biologie pour la plupart des systèmes faisant intervenir le phénomène de catalyse enzymatique. Cette hypothèse consiste à considérer l'approximation extérieure comme suffisamment pertinente pour décrire la dynamique de (1.1). En d'autres termes, cela revient à négliger la dynamique rapide de la fonction $v(\tau)$ au voisinage de l'origine. Il s'agit de l'hypothèse de *quasi-stationnarité*. En remplaçant dans (1.8) u_0 et v_0 par u et v , nous obtenons :

$$\begin{cases} v &= \frac{u}{u + K} \\ \frac{du}{d\tau} &= -\frac{\lambda u}{u + K} \end{cases} \quad (1.9)$$

La seconde équation est une équation différentielle scalaire :

$$\frac{du}{d\tau} = -\frac{\lambda u}{u + K}$$

à laquelle on associe la condition initiale : $u(0) = 1$. La résolution de cette équation est possible de manière implicite (intégrale première) :

$$K \ln u(\tau) + u(\tau) = 1 - \lambda\tau$$

En redimensionnalisant le système (1.9), nous obtenons :

$$\begin{aligned} \frac{1}{k_1 s_0 e_0} \frac{ds}{dt} &= -\frac{k_2 s(t)}{k_1 s_0^2 \left(\frac{k_{-1} + k_2}{k_1 s_0} + \frac{s(t)}{s_0} \right)} \\ \frac{ds}{dt} &= -\frac{k_2 e_0 s(t)}{\frac{k_{-1} + k_2}{k_1} + s(t)} \end{aligned}$$

D'où, en posant $V_m = k_2 e_0$ et $K_M = \frac{k_{-1} + k_2}{k_1}$:

$$\boxed{\mathcal{V}_r = -\frac{ds}{dt} = \frac{V_m s}{K_M + s}} \quad (1.10)$$

Cette formule donne la vitesse \mathcal{V}_r de la réaction (1.1) en fonction de la concentration en substrat s . Elle est appelée équation quasi-stationnaire de Michaelis-Menten. Dans cette équation, deux termes spécifiques à l'enzyme E apparaissent. Le terme V_m qui dépend de la concentration initiale e_0 est appelé *vitesse maximale* car il correspond à la vitesse limite de la réaction lorsque la concentration en substrat s tend vers $+\infty$, et la constante K_M (dite de *Michaelis*)

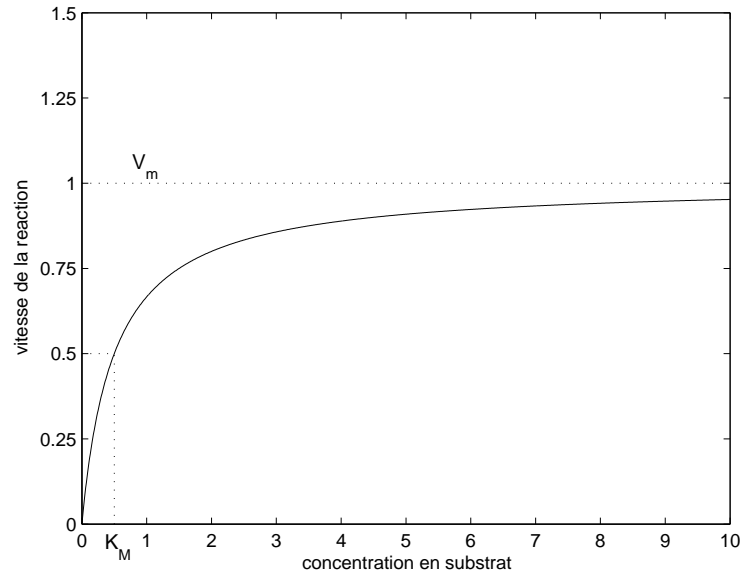


FIG. 1.5: Vitesse de la réaction de catalyse enzymatique en fonction de la concentration en substrat sous l'hypothèse de quasi-stationnarité de Michaelis-Menten (valeurs numériques : $V_m = 1$ et $K_M = 0.5$). Les constantes V_m et K_M peuvent se lire directement sur la courbe : V_m est l'asymptote horizontale (vélocité maximale) et K_M est égale à la concentration en substrat lorsque la vitesse est égale à la moitié de V_m .

qui ne dépend que du type de l'enzyme et bien sûr des conditions de température et de pression. La figure 1.5 montre l'allure de \mathcal{V}_r en fonction de s .

Dans la suite de ce rapport, nous serons amenés à reconsidérer l'hypothèse faite plus haut selon laquelle la quantité d'enzyme reste constante au cours du temps (équation (1.4)). Lors d'une expérience *in vitro*, une telle hypothèse semble raisonnable et tout-à-fait justifiable. Lorsque l'on considère une cellule, et d'autant plus si l'on inclut dans notre modèle la production de l'enzyme par la cellule, cette hypothèse devient discutable. Voyons quelles modifications entraînent la relaxation de cette hypothèse. Lors des précédents calculs, la relation (1.4) imposait :

$$\forall t \geq 0, \quad e(t) = e_0 - c(t)$$

Supposons que nous ayons à la place :

$$\forall t \geq 0, \quad e(t) = \alpha(t) - c(t)$$

où α est une fonction connue du temps qui vérifie : $\alpha(0) = e_0$. Cela revient à rajouter un terme à la deuxième équation du système (1.2) :

$$\dot{e}(t) = \dot{\alpha}(t) - k_1 e(t) s(t) + (k_{-1} + k_2) c(t)$$

Le système (1.5) s'écrit alors :

$$\begin{cases} \dot{s}(t) &= -k_1 \alpha(t) s(t) + (k_{-1} + k_1 s(t)) c(t) \\ \dot{c}(t) &= k_1 \alpha(t) s(t) - (k_1 s(t) + k_{-1} + k_2) c(t) \end{cases}$$

Lors de l'adimensionnalisation de ce système, nous rajoutons la relation :

$$w(\tau) = \frac{\alpha(t)}{e_0}$$

alors, par des calculs similaires aux précédents, nous nous ramenons au système adimensionné suivant :

$$\begin{cases} \frac{du}{d\tau} = -wu + (u + K - \lambda)v \\ \varepsilon \frac{dv}{d\tau} = wu - (u + K)v \end{cases}$$

Nous considérons alors l'hypothèse de quasi-stationnarité $\varepsilon = 0$, qui nous conduit à :

$$\begin{cases} v = \frac{wu}{u + K} \\ \frac{du}{d\tau} = -\frac{\lambda wu}{u + K} \end{cases}$$

Comme précédemment, l'hypothèse de quasi-stationnarité induit la perte d'un ordre pour l'équation de la dynamique rapide, qui ainsi devient une équation algébrique. Après redimensionnalisation, nous trouvons la vitesse de la réaction incluant le terme α :

$$\boxed{\mathcal{V}_r^\alpha = \frac{k_2 \alpha s}{K_M + s}} \quad (1.11)$$

En conclusion, si nous voulons inclure dans notre modèle un processus indépendant de production ou de dégradation de l'enzyme E , alors, sous l'hypothèse de quasi-stationnarité, il nous suffit de considérer l'équation de Michaelis (1.10) en remplaçant la constante V_m par le terme $k_2 \alpha$ où $\frac{d\alpha}{dt}$ est le terme de production ou de dégradation voulu.

1.1.3 Coopérativité et allostérie

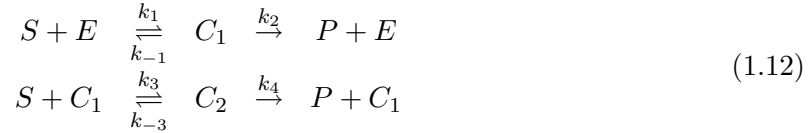
L'équation de Michaelis (1.10) (ou éventuellement sa variante (1.11)) est très utile dans la modélisation en biologie. Elle est d'ailleurs largement utilisée car elle donne une bonne approximation de la dynamique d'une catalyse enzymatique simple. Toutefois, elle est insuffisante pour de nombreux enzymes présents dans la nature qui ont certaines propriétés particulières ayant un effet sur leur dynamique. Il s'agit des propriétés de *coopérativité* et d'*allostérie*.

Dans le modèle précédent, une molécule d'enzyme se lie avec une molécule de substrat pour donner une molécule de produit. Plus précisément, la molécule de substrat se lie avec l'enzyme au niveau d'un site particulier, doté d'une affinité chimique avec le substrat. Rappelons qu'un enzyme est une protéine, c'est-à-dire une séquence d'acides aminés, qui prend lors de sa formation une configuration spatiale particulière, propre au type de l'enzyme. Grâce à cette structure, chaque type d'enzyme possède un ou plusieurs sites particuliers présentant une affinité avec un certain type de substrat. Dans le modèle précédent, nous avons supposé que l'enzyme E ne possédait qu'un seul site de liaison. Une fois qu'une molécule d'enzyme rencontre une molécule de substrat, ce substrat vient se lier à l'enzyme au niveau de ce site pour former le complexe (SE). L'enzyme devient alors temporairement indisponible pour d'éventuelles autres molécules de substrat. Cette hypothèse n'est pas toujours vérifiée dans la nature. En effet, certains types d'enzymes possèdent, de par leur structure, *plusieurs* sites de liaison avec

leur substrat (voir [54]). Ainsi, l'hémoglobine, qui est une protéine⁶ permettant le transport de l'oxygène dans le sang possède quatre sites de liaison présentant une affinité avec le dioxygène O_2 . Lorsqu'une molécule de dioxygène se lie à une molécule d'hémoglobine au niveau de l'un de ces sites, le complexe formé par ces deux molécules est encore disponible pour une autre liaison. On parle dans ce cas de liaison *coopérative*.

Un autre phénomène voisin de celui-ci est connu sous le nom d'*allostérie*. Une enzyme allostérique possède également plusieurs sites de liaison, mais la différence est que chaque type de liaison concerne un substrat différent. On parlera d'*effet allostérique* si la liaison d'un substrat S_1 sur un site Σ_1 modifie l'affinité d'un autre site Σ_2 pour son substrat S_2 . S_1 est appelé *effecteur* de Σ_2 , car il a sur ce site un effet indirect : il peut activer ou au contraire inhiber la liaison du deuxième substrat S_2 avec le site Σ_2 . Si la liaison S_1 - Σ_1 a pour effet d'augmenter l'affinité entre S_2 et Σ_2 , alors S_1 est appelé *activateur* de l'enzyme. Si au contraire elle a pour effet de diminuer cette affinité, S_1 est alors appelé *inhibiteur* de l'enzyme. L'allostérie est un phénomène riche et complexe, que nous ne détaillerons pas plus ici.

En revanche, nous proposons l'étude d'un enzyme coopératif ayant deux sites de liaison avec le même substrat. Considérons le modèle suivant (voir [54]) :



Dans ces équations, S dénote le substrat, E l'enzyme et P le produit. La lettre C_1 désigne le complexe formé par une molécule d'enzyme (libre) et une molécule de substrat. La lettre C_2 , quant à elle, désigne le complexe formé par une molécule d'enzyme et deux molécules de substrat (ou, de manière équivalente, par une molécule de complexe C_1 et une molécule de substrat). Comme dans la partie précédente, nous appliquons à ce système la loi d'action de masse, qui nous conduit au système différentiel suivant :

$$\begin{cases} \dot{s}(t) &= -k_1 e(t)s(t) + (k_{-1} - k_3 s(t))c_1(t) + k_{-3}c_2(t) \\ \dot{e}(t) &= -k_1 e(t)s(t) + (k_{-1} + k_2)c_1(t) \\ \dot{c}_1(t) &= k_1 e(t)s(t) - (k_{-1} + k_3 s(t) + k_2)c_1(t) + (k_{-3} + k_4)c_2(t) \\ \dot{c}_2(t) &= k_3 s(t)c_1(t) - (k_{-3} + k_4)c_2(t) \\ \dot{p}(t) &= k_2 c_1(t) + k_4 c_2(t) \end{cases} \quad (1.13)$$

muni de la condition initiale :

$$s(0) = s_0 \in \mathbb{R}_+^*, \quad e(0) = e_0 \in \mathbb{R}_+^*, \quad c_1(0) = 0, \quad c_2(0) = 0, \quad p(0) = 0$$

Comme dans la partie précédente, la concentration en produit $p(t)$ peut se déduire directement des concentrations en complexes $c_1(t)$ et $c_2(t)$:

$$\forall t > 0, \quad p(t) = k_2 \int_0^t c_1(t') dt' + k_4 \int_0^t c_2(t') dt'$$

De plus, en additionnant la deuxième, la troisième et la quatrième équation du système (1.13), nous obtenons une équation de conservation de la quantité d'enzyme :

$$\forall t \geq 0, \quad \dot{e}(t) + \dot{c}_1(t) + \dot{c}_2(t) = 0$$

⁶Même si l'hémoglobine n'est pas à proprement parler un enzyme, son action de transporteur d'oxygène sera considérée, à notre niveau, de la même manière qu'une catalyse enzymatique.

qui entraîne pour tout t , $e(t) = e_0 - c_1(t) - c_2(t)$. La variable e est donc entièrement déterminée par les variables c_1 et c_2 . Nous nous ramenons donc au système suivant :

$$\begin{cases} \dot{s}(t) &= -k_1 e_0 s(t) + (k_{-1} + k_1 s(t) - k_3 s(t)) c_1(t) + (k_1 s(t) + k_{-3}) c_2(t) \\ \dot{c}_1(t) &= k_1 e_0 s(t) - (k_1 s(t) + k_3 s(t) + k_{-1} + k_2) c_1(t) + (k_{-3} + k_4 - k_1 s(t)) c_2(t) \\ \dot{c}_2(t) &= k_3 s(t) c_1(t) - (k_{-3} + k_4) c_2(t) \end{cases}$$

Nous appliquons alors l'adimensionnalisation suivante :

$$\tau = k_1 e_0 t, \quad u(\tau) = \frac{s(t)}{s_0}, \quad v_1(\tau) = \frac{c_1(t)}{e_0}, \quad v_2(\tau) = \frac{c_2(t)}{e_0}$$

et nous posons les constantes :

$$a_1 = \frac{k_{-1}}{k_1 s_0}, \quad a_2 = \frac{k_2}{k_1 s_0}, \quad a_3 = \frac{k_3}{k_1}, \quad a_4 = \frac{k_{-3}}{k_1 s_0}, \quad a_5 = \frac{k_4}{k_1 s_0}, \quad \varepsilon = \frac{e_0}{s_0}$$

Nous obtenons alors le système adimensionnalisé équivalent du système (1.6) :

$$\begin{cases} \frac{du}{d\tau} &= -u + (u + a_1 - a_3 u) v_1 + (u + a_4) v_2 \\ \varepsilon \frac{dv_1}{d\tau} &= u - (u + a_1 + a_2 + a_3 u) v_1 + (a_4 + a_5 - u) v_2 \\ \varepsilon \frac{dv_2}{d\tau} &= a_3 u v_1 - (a_4 + a_5) v_2 \end{cases}$$

muni de la condition initiale : $u(0) = 1, v_1(0) = v_2(0) = 0$.

Comme précédemment, les variables v_1 et v_2 ont une dynamique rapide au voisinage de l'origine lorsque ε est proche de zéro. Nous pouvons utiliser les mêmes méthodes de résolution que dans la partie précédente, de manière à trouver une approximation globale de la solution de ce système, à l'intérieur et à l'extérieur de la couche limite. Néanmoins, comme nous l'avons vu, cette approche n'est pas très utile du point de vue de la modélisation. Nous allons donc directement nous intéresser à l'hypothèse de quasi-stationnarité, c'est-à-dire au cas où $\varepsilon = 0$. Les deux dernières équations deviennent alors des équations algébriques, et déterminent une surface \mathcal{S} dans l'espace (u, v_1, v_2) . Après calculs, cette surface est donnée par :

$$\mathcal{S} \begin{cases} v_1 &= \frac{u}{u + a_1 + a_2 + \frac{a_3 u^2}{a_4 + a_5}} \\ v_2 &= \frac{a_3 u^2}{(a_4 + a_5)(u + a_1 + a_2) + a_3 u^2} \end{cases}$$

Sur cette surface, l'équation différentielle scalaire devient :

$$\frac{du}{d\tau} = -u \frac{a_2 + \frac{a_3 a_5 u}{a_4 + a_5}}{u + a_1 + a_2 + \frac{a_3 u^2}{a_4 + a_5}}$$

Lors de la redimensionnalisation, nous posons les deux constantes :

$$K_M = \frac{k_{-1} + k_2}{k_1}, \quad K'_M = \frac{k_{-3} + k_4}{k_3} \quad (1.14)$$

Nous obtenons alors l'équation de quasi-stationnarité :

$$\mathcal{V}_r = e_0 \frac{K'_M k_2 s + k_4 s^2}{K_M K'_M + K'_M s + s^2} \quad (1.15)$$

Nous nous intéressons à présent à l'allure de la courbe de la vitesse \mathcal{V}_r en fonction de la concentration en substrat s . Il nous faut distinguer deux cas, selon la position relative des constantes k_2 et k_4 . Nous considérons la fonction d'une variable réelle :

$$f(s) = \frac{K'_M k_2 s + k_4 s^2}{K_M K'_M + K'_M s + s^2}$$

La fonction f vérifie :

$$f(0) = 0 \quad \text{et} \quad \lim_{s \rightarrow +\infty} f(s) = k_4$$

Afin d'étudier le comportement de f sur \mathbb{R}_+ , nous calculons sa dérivée :

$$f'(s) = \frac{As^2 + Bs + C}{(K_M K'_M + K'_M s + s^2)^2}$$

avec :

$$A = K'_M(k_4 - k_2), \quad B = 2K_M K'_M k_4, \quad C = K_M (K'_M)^2 k_2$$

Afin de trouver le signe du polynôme $P(s) = As^2 + Bs + C$, nous discutons selon le signe de $k_2 - k_4$:

Premier cas : si $k_2 > k_4$, alors on a :

$$\begin{cases} A < 0 \\ B > 0 \\ C > 0 \end{cases}$$

En utilisant la règle de Descartes (voir [52] p. 208 et remarque 1.1 ci-dessous), nous déduisons que le polynôme P admet une unique racine réelle positive. Ceci implique que la fonction f est croissante sur $[0, \rho]$ (où ρ est l'unique racine réelle positive de P), puis décroissante sur $[\rho, +\infty[$. Cette courbe présente donc un maximum sur \mathbb{R}_+ . La figure 1.6 montre l'allure de la courbe de (1.15) dans le cas où $k_2 > k_4$.

Deuxième cas : si $k_2 < k_4$, alors on a :

$$\begin{cases} A > 0 \\ B > 0 \\ C > 0 \end{cases}$$

Dans ce cas, pour tout $s \geq 0$, $P(s)$ est positif, ce qui implique que la vitesse de la réaction est croissante en fonction de la quantité en substrat. La figure 1.7 montre l'allure de la courbe. On observe sur cette courbe un point d'inflexion, c'est-à-dire un point où la concavité de la courbe change. Afin de vérifier qu'un tel point existe, il nous faut chercher les zéros de $f''(s)$.

$$f''(s) = \frac{A's^3 + B's^2 + C's + D'}{(K_M K'_M + K'_M s + s^2)^3}$$

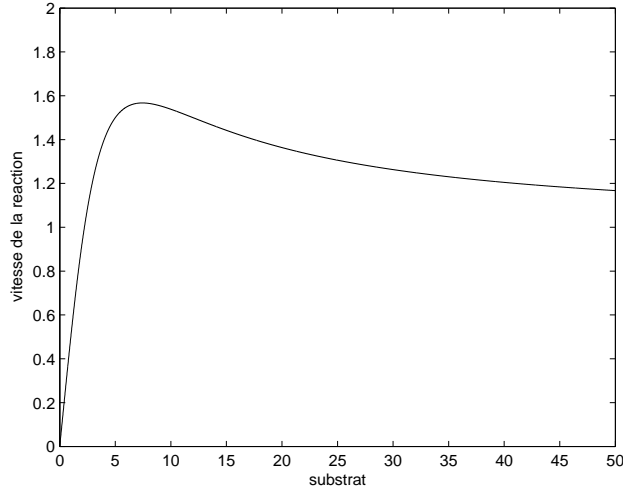


FIG. 1.6: Vitesse de la réaction de catalyse enzymatique coopérative (avec deux sites de liaison) en fonction de la concentration en substrat sous l'hypothèse de quasi-stationnarité de Michaelis-Menten (valeurs numériques : $e_0 = 1$, $K_M = 20$, $K'_M = 1$, $k_2 = 10$ et $k_4 = 1$). Cas 1 : $k_2 > k_4$.

avec :

$$\begin{cases} A' &= 2K'_M(k_2 - k_4) \\ B' &= -6K_M K'_M k_4 \\ C' &= -6K_M (K'_M)^2 k_2 \\ D' &= 2K_M (K'_M)^2 (K_M k_4 - K'_M k_2) \end{cases}$$

Etant donné que $k_2 < k_4$, nous déduisons :

$$\begin{cases} A' < 0 \\ B' < 0 \\ C' < 0 \end{cases}$$

Donc, le polynôme $Q(s) = A's^3 + B's^2 + C's + D'$ admet une racine positive si et seulement si le coefficient D' est positif. La courbe $\mathcal{V}_r(s)$ admet donc un unique point d'inflexion $s_0 > 0$ à la condition nécessaire et suffisante que :

$$K_M k_4 - K'_M k_2 > 0$$

En utilisant (1.14), nous exprimons cette condition sous la forme :

$$\frac{k_{-1} + k_2}{k_1 k_2} > \frac{k_{-3} + k_4}{k_3 k_4} \quad (1.16)$$

Remarque 1.1 Nous avons mentionné plusieurs fois la règle de Descartes, dont nous trouverons un énoncé complet dans [52]. Nous utilisons ici un corollaire évident de cette règle, qui stipule que si la liste des coefficients d'un polynôme à coefficients réels possède un unique changement de signe, alors le polynôme admet une et une seule racine réelle positive.

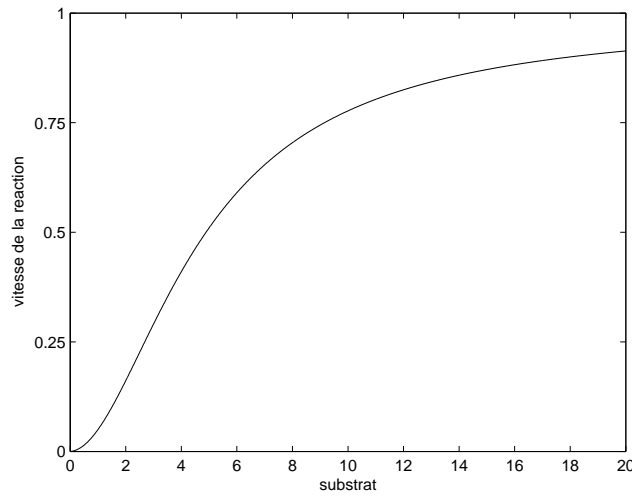
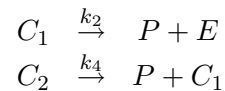


FIG. 1.7: Vitesse de la réaction de catalyse enzymatique coopérative (avec deux sites de liaison) en fonction de la concentration en substrat sous l'hypothèse de quasi-stationnarité de Michaelis-Menten (valeurs numériques : $e_0 = 1$, $K_M = 20$, $K'_M = 1$, $k_2 = 0.1$ et $k_4 = 1$). Cas 2 : $k_2 < k_4$.

Voyons à présent comment s'interprètent ces deux cas d'un point de vue biologique. D'après les équations (1.12), les constantes k_2 et k_4 correspondent respectivement aux taux de transformation des complexes C_1 et C_2 en produit :



Supposer $k_2 > k_4$ revient donc à supposer que la réaction de dégradation de C_1 en produit est plus efficace que la dégradation de C_2 . Ceci aurait tendance à diminuer le rôle coopératif de l'enzyme E . Si nous voulons mettre en avant cette coopérativité, l'hypothèse $k_2 < k_4$ semble donc plus raisonnable.

Intéressons-nous de plus près à la courbe de la figure 1.7. Cette courbe présente plusieurs propriétés tout-à-fait particulières d'un point de vue mathématique :

- On a $f(0) = 0$ et $\lim_{s \rightarrow +\infty} f(s) = k_4 > 0$,
- f est strictement croissante sur \mathbb{R}_+ ,
- On observe un unique point d'inflexion dans \mathbb{R}_+ (si la condition (1.16) est vérifiée).

Ces trois propriétés sont classiques et déterminent une classe de courbes particulières : les *sigmoïdes*. Elles apparaissent dans de nombreux modèles, notamment en biologie. Dans le cas présent, nous avons une expression analytique exacte (équation (1.15)). Ceci est loin d'être toujours le cas. Dans d'autres modèles, nous ne disposons d'aucune expression analytique. Nous savons en revanche, par exemple grâce à des expérimentations, que la dynamique du phénomène étudié a une forme sigmoïdale. Dans ce cas, la démarche classique consiste à proposer un modèle dans lequel on inclut un type de sigmoïde connu, dans le but d'obtenir un type de comportement censé s'approcher du comportement réel. Cette démarche est uniquement *qualitative*, et ne

permet pas à priori d'obtenir une approximation numérique⁷ ou quantitative du phénomène. En biologie, de nombreux processus sont expliqués de manière qualitative. Ils présentent une telle complexité qu'il est à l'heure actuelle impossible d'en tirer un modèle quantitatif. C'est notamment le cas de la régulation génétique.

On peut citer différentes classes de sigmoïdes, néanmoins la plus utilisée en biologie est la fonction de *Hill*. Nous décrivons dans l'annexe A quelques propriétés utiles de cette fonction.

En ce qui concerne le phénomène de coopérativité, l'hypothèse $k_2 < k_4$ présente plus de pertinence. En l'absence de valeurs numériques précises, on considère généralement qu'un enzyme coopératif admet une courbe de vitesse sigmoïdale. Selon l'usage que l'on veut faire de notre modèle, on cherchera parfois à s'approcher au plus près du phénomène biologique étudié. Nous reprenons alors le type de calculs faits plus haut afin de trouver une expression analytique faisant intervenir des paramètres biologiques dépendant directement du phénomène étudié (dans le cas précédent, les paramètres K_M et K'_M se déduisent des constantes k_i des réactions chimiques (1.12)). Dans d'autres cas, par exemple si les valeurs de ces paramètres ne sont pas connues, il nous faut passer par une étape de modélisation, dans laquelle on propose un système présentant des propriétés censées s'approcher des propriétés du système réel. Le plus souvent, pour décrire la dynamique d'un enzyme coopératif ou d'un enzyme allostérique, les modèles proposés utilisent des fonctions de Hill (voir annexe A), car leur expression facilite le traitement mathématique des équations.

1.2 Les réseaux de régulation génétique

Après avoir donné des éléments de compréhension de la catalyse enzymatique, nous proposons dans cette partie une description plus détaillée de la notion de régulation génétique à travers l'exemple de l'opéron lactose, déjà évoqué plus haut. Les premières avancées vers la découverte de ce mécanisme complexe datent de 1961 (voir [35, 36]). Afin d'en comprendre le mécanisme, nous décrivons tout d'abord certains éléments concernant la production d'énergie chez la bactérie *e-coli*. Dans tout ce qui suit, nous nous référons notamment au livre de J. Monod intitulé *Le hasard et la nécessité* [53].

1.2.1 Production d'énergie chez la bactérie e-coli

Sans entrer dans le détail biologique de tous les mécanismes cellulaires, nous expliquons brièvement ici comment la bactérie procède pour produire et stocker de l'énergie. Chaque cellule contient dans son cytoplasme des organites appelés *mitochondries*. C'est à l'intérieur de ces organites que la cellule produit son énergie. Cette énergie est sous forme chimique : il s'agit de molécules d'ATP (adénosine triphosphate) qui contiennent une énergie potentielle. Lorsqu'il y en a besoin, une molécule d'ATP peut se dégrader en ADP (adénosine diphosphate), ce qui libère une certaine quantité d'énergie utilisable directement par la cellule pour toute sorte de fonctions⁸. L'ADP retourne alors dans la mitochondrie pour être à nouveau transformée en ATP et ainsi de suite. Il est clair que si la dégradation de l'ATP dégage de l'énergie, sa fabrication en nécessite (c'est une réaction endergonique). Cette énergie est récupérée par la dégradation d'un certain type de métabolite : le glucose. Ce sucre est la biomolécule la plus répandue dans

⁷Au sens des numériciens ; c'est-à-dire que l'on ne peut pas à priori espérer une solution approchée qui converge vers la solution réelle.

⁸par exemple les mécanismes de transcription et de traduction décrits dans la partie précédente nécessitent une grande quantité d'énergie pour séparer localement les brins d'ADN, pour produire l'ARN messager etc. . .

la biosphère, et elle est la source première d'énergie pour l'ensemble des cellules. Le mécanisme permettant la production d'énergie à partir du glucose est appelé *glycolyse*.

Dans un organisme pluricellulaire évolué comme l'être humain, l'accès au glucose par les cellules est géré par des mécanismes complexes, que nous n'aborderons pas ici. Pour un organisme unicellulaire comme *e-coli*, l'absorption de glucose est une phase critique puisque la bactérie doit pouvoir s'approvisionner toute seule. Cet approvisionnement se fait par un transport des molécules de glucose extérieures à travers la membrane. Ce transport s'effectue par l'intermédiaire de complexes biochimiques nommés PTS (phosphotransferase system, voir [86]). Nous n'irons pas plus avant dans la description de ce processus. Ce qui nous intéresse ici est le cas où le glucose extérieur vient à manquer. Dans ces conditions, la bactérie peut continuer à dégrader les molécules de glucose présentes dans le milieu intracellulaire, mais elle doit trouver une solution pour le futur. Nous verrons dans la partie suivante que c'est à ce niveau qu'intervient le patrimoine génétique : il lui permet d'abord de détecter l'absence de glucose dans le milieu extracellulaire, puis met en place une solution de remplacement, c'est-à-dire l'utilisation d'une source d'énergie secondaire.

Plusieurs possibilités s'offrent à notre bactérie, elle peut par exemple choisir de dégrader un autre glucide comme le fructose ou le lactose. On parle alors de croissance diauxique, la cellule est en présence de deux sources d'énergie glucidiques : elle décide dans ce cas d'absorber d'abord le glucose ; puis, une fois cette première substance épuisée, elle utilise la deuxième substance (par exemple le lactose). Pourquoi cet ordre ? Ce n'est certes pas par hasard, mais pour des raisons d'économies. On a observé en effet une croissance plus rapide de populations d'*e-colis* nourries à partir d'hexoses (hexoses : sucres à six carbones, dont fait parti le glucose). Ceci vient du fait que les autres types de sucres, comme le lactose (qui est un pentose à cinq carbones), doivent subir un traitement préalable avant d'être dégradés en vue de produire de l'énergie. Pour le lactose, ce traitement consiste en deux étapes essentielles : sa pénétration à l'intérieur de la cellule et sa métabolisation (nous décrirons par la suite plus en détail ces deux phases). Ces deux étapes nécessitent des enzymes spécifiques, qui doivent être synthétisées avant toute utilisation possible du lactose dans la bactérie. L'alimentation en lactose est donc plus coûteuse et c'est pourquoi la bactérie choisit une croissance en deux temps : d'abord utilisation du glucose jusqu'à épuisement ; puis, démarrage de la synthèse des enzymes lactiques en vue de l'utilisation du lactose.

A travers cet exemple, nous comprenons mieux la notion de contrôle que nous avons évoquée plus haut : dans des conditions où le glucose est disponible en quantité suffisante, les enzymes lactiques ne sont pas produits ; en revanche, une fois que cette source de glucose disparaît, la bactérie démarre la transcription des gènes codant pour ces enzymes. La transcription des gènes est donc bien contrôlée dynamiquement, et ce contrôle doit permettre à l'organisme de se développer en suivant une *organisation* à la fois robuste et adaptative. Nous allons étudier à présent en détail la mise en œuvre de ce contrôle dans le cas précis du traitement du lactose.

1.2.2 Fonctionnement de l'opéron lactose

Nous nous intéressons ici à la structure de régulation, nommée *opéron lactose* qui permet à la bactérie *e-coli* de commander la synthèse des gènes codant pour les enzymes traitant le lactose. L'opéron lactose est cité en exemple dans beaucoup d'ouvrages de modélisation (consulter par exemple [43] et [76]) car il s'agit du cœur historique de la théorie des réseaux de régulation génétiques. Depuis sa découverte dans les années 60, d'autres opérons ont été étudiés (c'est le cas par exemple de l'opéron tryptophane, [66]).

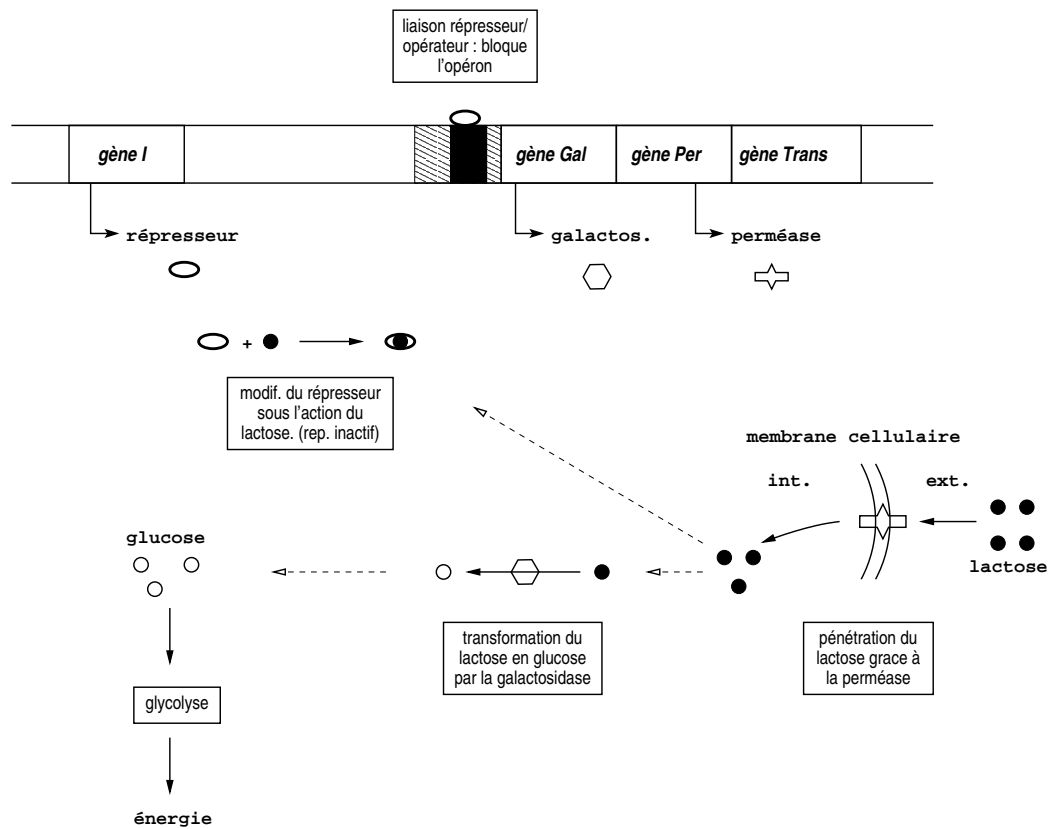


FIG. 1.8: Schéma de l'opéron lactose. On retrouve dans ce schéma les phénomènes liés à la boucle négative de l'opéron. Remarques : le promoteur du *gène I* n'est pas représenté, de même que la modification du lactose en allo-lactose par la galactosidase.

Un opéron est un groupe de gènes structurels, c'est-à-dire de gènes codant pour des protéines ayant un rôle spécifique dans le développement de la cellule, par opposition aux protéines régulatrices, dont la fonction est de réguler l'expression des autres gènes. Dans le cas de l'opéron lactose, il s'agit de trois gènes adjacents le long de la molécule d'ADN d'e-coli : les gènes *gal*, *per* et *trans* (voir figure 1.8). Ces gènes codent pour trois enzymes qui sont respectivement la β -galactosidase, la perméase et la transacétylase. Si la fonction du troisième enzyme est encore mal connue⁹, la fonction de la galactosidase et de la perméase est, comme nous l'avons déjà dit, de permettre à la cellule de s'alimenter en lactose. La perméase est un enzyme transporteur : elle permet de faire pénétrer les molécules de lactose du milieu extracellulaire qui sont à proximité de la membrane à l'intérieur de la cellule. La β -galactosidase, quant à elle, est un enzyme qui favorise la transformation du lactose intracellulaire en glucose, selon la réaction chimique suivante :



Elle permet également de modifier certaines molécules de lactose en allo-lactose qui est une deuxième forme du lactose (nous verrons l'utilité de cette réaction dans la suite de notre description).

Si la bactérie veut pouvoir utiliser le lactose comme source d'énergie, elle a donc forcément besoin de ces deux enzymes. Or, on observe que dans une population d'e-colis évoluant dans un milieu dépourvu de lactose, le taux de transcription des gènes associés est extrêmement faible : environ une molécule est synthétisée toute les cinq générations cellulaires en moyenne. Lorsque du lactose est introduit dans le milieu, on note une brusque augmentation de ce taux : en l'espace de deux à trois minutes, en l'absence de glucose dans le milieu, il est multiplié par 1000. Il revient à sa valeur initiale quelques minutes après avoir cessé l'addition de lactose (chiffres tirés de [53]). Ces chiffres montrent de façon remarquable que la cellule dispose d'un moyen performant de régulation qui lui permet d'agir rapidement sur le taux de transcription des gènes de l'opéron.

Comme nous l'avons expliqué précédemment, la bactérie veut pouvoir choisir quel type de sucre elle va utiliser. Tant qu'elle est en présence de glucose, la synthèse des enzymes doit être freinée puisque la cellule préfère utiliser directement le glucose. Leur taux de transcription suit donc une loi qui dépend de la quantité de lactose *et* de la quantité de glucose :

- Si le milieu extracellulaire est dépourvu de lactose, les enzymes n'ont aucune utilité : la transcription des deux gènes est *bloquée* (on dit aussi *réprimée*).
- Si le milieu est pourvu en lactose, alors la bactérie a le choix d'utiliser ou non ce lactose. Si l'on suit ce que nous avons dit plus haut, e-coli va choisir de s'alimenter en lactose seulement si l'alimentation en glucose n'est plus possible :
 - Si la bactérie dispose de glucose, alors les gènes seront peu exprimés (c'est la phase d'*induction modérée*),
 - dans le cas contraire, la bactérie accélèrera le processus de synthèse des deux enzymes (phase d'*induction*).

Cette description montre le *calcul* effectué par la cellule. Il dépend de deux entrées qui sont les concentrations en glucose et en lactose. Voyons à présent comment ce calcul s'effectue biologiquement.

⁹ ceci fait partie des limites inhérentes à la modélisation en biologie.

a. La boucle de régulation négative

Avant d'expliquer le fonctionnement de la régulation négative de l'opéron lactose, nous revenons sur le processus de transcription décrit dans la première partie. Nous avons dit que ce processus se fait grâce à un enzyme, l'ARN-polymérase, qui détecte le début du gène et ouvre localement la molécule d'ADN (c'est-à-dire sépare les deux brins aux niveaux des liaisons hydrogènes entre les bases complémentaires). La détection du début d'un gène est une phase capitale. Elle se fait grâce à un site particulier, appelé *promoteur*, qui consiste en une séquence spécifique de paires de bases qui est située avant le début de la séquence propre du gène. Un promoteur est de taille variable, de l'ordre en général de quelques dizaines de paires de bases. La spécificité de la séquence du promoteur est de présenter une affinité plus ou moins forte avec la polymérase, permettant ainsi de la fixer au début du gène, de manière à commencer la transcription. De manière analogue, il existe un site appelé *terminateur* situé juste après le gène, qui donne à la polymérase le signal de fin de transcription : lorsqu'elle rencontre le terminateur, la liaison de la polymérase avec l'ADN devient instable, les deux molécules sont alors séparées et la molécule d'ARN-m construite est libérée. Dans le cas de l'opéron lactose, les trois gènes structurels étant adjacents, il n'y a qu'un seul promoteur, situé avant le gène *gal* (voir figure 1.8) et un seul site terminateur, après le gène *trans*. La phase de transcription est donc unique pour les trois gènes : une seule molécule d'ARN est construite ; puis elle est séparée pour permettre de produire les trois protéines. Il est intéressant de noter l'utilité pour la cellule d'une telle structure : puisque les enzymes participent à la même tâche cellulaire (l'alimentation en lactose), il est beaucoup plus efficace de transcrire les trois gènes *simultanément*. La transcription de l'opéron est donc *coordonnée*.

Nous avons dit que lorsque la cellule ne dispose pas de lactose, l'opéron doit être réprimé, c'est-à-dire que la transcription des gènes ne doit pas être mise en route. Ceci se produit grâce à une protéine spécifique, souvent notée *lacl*. Cette protéine est codée par un gène en amont, le gène *l*, comme le montre la figure 1.8. Comme la polymérase, cette protéine présente une affinité particulière avec une séquence de nucléotides, celle-ci placée entre le promoteur et le début du gène *gal*. Ce site est nommé *opérateur*. La liaison de l'opérateur avec *lacl* est une liaison relativement forte, de sorte qu'elle empêche la polymérase d'initier la transcription de l'opéron : elle ne peut plus se fixer à son promoteur (voir figure 1.8). Le gène *l* n'est quant à lui pas régulé, c'est-à-dire qu'il est continuellement transcrit (on parle de transcription constitutive), à un taux estimé à une valeur faible. Ceci est une clé pour comprendre le fonctionnement l'opéron.

Lorsque la bactérie se trouve dans un milieu riche en lactose, on observe une *dérépression* de l'opéron. Ceci est dû à une autre particularité de la protéine *lacl* : cette protéine présente, outre son affinité avec l'opérateur, une affinité avec l'allo-lactose, qui est une forme dérivée du lactose (produite grâce à une modification du lactose par la *galactosidase*, cf. plus haut). Le répresseur a donc deux interactions possibles, toutes deux covalentes et réversibles (voir [53]). Il inactive la transcription en se liant à l'opérateur, et cette inhibition se trouve à son tour inactivée lorsqu'il se lie avec le lactose. Ce double effet négatif produit un effet positif appelé *induction* de l'opéron. Le rôle de l'inducteur est joué par le lactose lui-même. Ce mécanisme est représenté sur la figure 1.8.

Remarquons que le lactose, pour inactiver le répresseur, doit pénétrer dans la cellule et être préalablement modifié en allo-lactose. Pour cela, il doit avoir rencontré successivement une molécule de perméase (pour entrer dans la cellule) et une molécule de *galactosidase* (pour être modifié). Or, ces enzymes sont présentes en très faibles quantités avant induction (rappelons

qu'une molécule est produite toute les cinq générations environ). On peut donc raisonnablement supposer que l'allo-lactose est présent en très faible quantité. Il faut donc, pour que l'induction ait lieu, que cette quantité suffise à désactiver le répresseur. Le fait que les molécules de répresseur soient présentes en permanence dans la cellule en nombre très faible est donc capital pour permettre la dérégulation de l'opéron. Il est donc tout-à-fait cohérent de supposer que quelques molécules de lactose intérieur suffisent pour déclencher la **bascule génétique** qui permet d'induire l'opéron.

b. La boucle de régulation positive

Le processus de régulation décrit plus haut porte le nom de rétroaction négative. L'adjectif "négatif" vient du fait que le rôle de la molécule régulatrice est un rôle inhibiteur. Ce processus permet à la cellule d'adapter la transcription des enzymes lactiques à la présence ou à l'absence de lactose. On a découvert depuis un deuxième mécanisme de régulation de l'opéron, appelé boucle *positive*, qui concerne cette fois la quantité de glucose à l'intérieur de la cellule.

D'après le phénomène de croissance diauxique que nous avons défini plus haut, une cellule en présence de deux sources de carbone, dans notre cas le glucose et le lactose, choisit de consommer préférentiellement la source plus facilement utilisable (le glucose), puis, une fois cette source épuisée, de consommer la deuxième, qui lui demande plus d'efforts (le lactose). En présence des deux sucres, il y a donc freinage de la transcription de l'opéron. C'est ce processus de régulation que nous avons appelé boucle positive. Grâce à des mutations, on a pu isoler deux gènes indirectement responsables de cette régulation. Le premier participe à l'élaboration d'AMPc (adénosine monophosphate cyclique) et le deuxième synthétise la molécule CAP (pour catabolite activator protein). Ces deux substances semblent donc participer à l'activation de l'opéron. Or, on sait que la présence de glucose dans la cellule a tendance à freiner la production d'AMPc à partir d'ATP et à maintenir la quantité d'AMPc à une valeur très faible. Il résulte de ceci qu'une chute de la concentration de glucose dans la cellule entraîne une augmentation d'AMPc. Le mécanisme sous-jacent de la régulation positive de l'opéron est aujourd'hui connu : l'AMPc se lie à la protéine CAP pour former un complexe (AMPCAP) qui joue le rôle d'*activateur* pour l'opéron. Cet activateur a la propriété de se fixer sur le promoteur de l'opéron et de grandement favoriser son affinité avec la polymérase. Lorsque le complexe est fixé au promoteur, la transcription des enzymes s'en trouve donc accélérée.

Nous avons dit que la présence de glucose a un rôle inhibiteur sur la transcription de l'opéron. Ce phénomène porte un nom : il s'agit de la *répression catabolite*. Ce n'est pas un phénomène spécifique à l'opéron lactose, puisqu'on a découvert qu'il intervenait dans d'autres opérons commandant l'alimentation en d'autres types de glucides (l'opéron galactose, arabinose, etc.). Il permet à la bactérie de privilégier le glucose par rapport à n'importe quel autre source d'énergie. En outre, il lui permet, dans le cas où la quantité de glucose vient à diminuer, de déclencher de manière rapide un opéron mettant en place un mécanisme d'alimentation secondaire. Dans le cas du lactose, la répression catabolite est accentuée par un deuxième phénomène : l'exclusion de l'inducteur (voir [86]). En effet, sous l'influence du glucose, la molécule de perméase est légèrement modifiée dans sa configuration spatiale habituelle, perdant ainsi la capacité de faire rentrer efficacement le lactose dans la cellule. La régulation se fait donc à deux niveaux : d'une part, la présence de glucose empêche la formation d'AMPc et donc la formation du complexe activateur¹⁰, et d'autre part, ce même glucose freine l'entrée du

¹⁰Si l'on connaît l'influence du glucose sur l'AMP, on sait en revanche peu de choses sur celle de la protéine CAP.

lactose dans la cellule, empêchant ainsi l'induction de l'opéron. Grâce à ces deux phénomènes, le glucose joue donc un rôle inhibiteur à deux niveaux sur la transcription des gènes *gal*, *per* et *trans*.

1.2.3 Vers une modélisation de la régulation génétique

Grâce aux travaux sur l'opéron lactose, l'étude de la régulation génétique est devenue aujourd'hui un domaine de recherche actif. D'autres opérons ont été mis en évidence, pour le traitement d'autres sucres (galactose, arabinose, fructose etc.) mais également pour réguler la concentration d'autres substances (l'opéron tryptophane en est un exemple). D'autres réseaux de régulation génétiques ont été isolés, nous citerons notamment le cas du λ -phage qui est un parfait exemple de bascule génétique (voir [61]). Le cas du système lactose reste néanmoins très intéressant puisqu'il contient un exemple de régulation négative et de régulation positive. Reprenant les arguments de J. Monod [53], nous résumons ici les caractéristiques qui nous semblent intéressantes pour définir la notion de système de régulation génétique.

- Il existe sur la molécule d'ADN des séquences de nucléotides qui ne font partie d'aucun gène. Cette partie non codante est même prédominante, et l'on n'a pas encore élucidé tous les rôles possibles qu'elle peut avoir dans le développement d'une cellule ou dans le principe plus général d'évolution. On a néanmoins découvert que certains de ces sites (les promoteurs et les opérateurs) ont, par leur séquence particulière, un rôle régulateur, bloquant ou activant la transcription de certains gènes.
- Le répresseur, qui fait partie de la classe des protéines, n'a aucune activité spécifique directe. Il est un simple élément *régulateur*, qui permet par ses affinités chimiques avec le promoteur et l'allo-lactose de respectivement réprimer ou activer l'opéron. Faisons l'analogie entre l'opéron et un circuit électrique : grâce à ces deux états chimiques mutuellement exclusifs, le comportement du répresseur peut être interprété comme celui d'un interrupteur ou d'une bascule. D'autres protéines du même type ont été découvertes, on les nomme *protéines régulatrices* par opposition aux protéines constitutives ou structurelles qui, par leurs fonctions spécifiques, interviennent directement dans le métabolisme cellulaire (par exemple, les enzymes).
- Le fait que le lactose soit lui-même l'inducteur de l'opéron est dû uniquement à ses propriétés chimiques de liaison avec le répresseur. Son action d'inducteur est donc une action indirecte, liée aux connexions chimiques entre les éléments du système. Dans le même ordre d'idées, le fait que la galactosidase et la perméase voit leur production induite par leur propre substrat est une relation utile du point de vue métabolique, mais chimiquement arbitraire (J. Monod parle de relation *gratuite*).

Ainsi, l'étude des opérons bactériens a permis de mettre en évidence une *structure logique* sous-jacente dans la bactérie, dont les éléments interagissants sont certaines espèces présentes dans la cellule et les connexions sont des interactions chimiques (éventuellement gratuites) entre ces espèces. Notons que les espèces chimiques dont on parle sont très diverses : il peut s'agir de protéines (cas du répresseur) mais également de sites sur la molécule d'ADN (l'opérateur et le promoteur) ou encore de métabolites (le lactose et le glucose). L'une des premières difficultés d'une modélisation du phénomène résidera donc dans le choix des variables : comment

représenter mathématiquement les quantités en les différentes espèces ? Pour les protéines, ou les métabolites, nous pouvons utiliser la concentration chimique. Un problème se posera inévitablement pour modéliser l'induction de l'opéron. Etant donné qu'à chaque instant il y a peu de molécules de répresseur, la quantité d'allo-lactose nécessaire pour libérer la liaison répresseur-opérateur ne s'exprime donc plus en concentration mais en nombre de molécules. Nous nous heurtons là à un sérieux problème de modélisation.

On notera également que les réseaux génétiques que nous avons cités jusqu'ici concernent des bactéries (le plus souvent *Escherichia coli*) qui sont des organismes monocellulaires. Dans des organismes pluricellulaires plus évolués, beaucoup de systèmes de régulation sont plus complexes encore car ils font intervenir une organisation plus spécialisée. Ainsi chez l'homme, le foie participe directement à l'alimentation en glucose puisqu'il stocke celui-ci sous la forme de macromolécules de glycogène. Ceci ne met pas en cause l'existence d'opérons dans ce type de cellules, mais leur étude est beaucoup plus complexe car les phénomènes de régulation à l'échelle de la cellule sont couplés avec des régulations entre les différents organes, à l'échelle de l'organisme. En outre, la coordination entre les différents tissus ou organes échappe, le plus souvent, à la description moléculaire. Nous nous contenterons donc de considérer la régulation à l'échelle d'un organisme monocellulaire. Pour avoir une idée de la complexité de la tâche, nous donnons quelques chiffres concernant la bactérie *e-coli* (on se référera notamment à la très complète base de donnée *ecocyc*¹¹) : le nombre de gènes répertoriés est de l'ordre de 4500, le nombre de promoteurs de 900 et le nombre de voies métaboliques¹² connues d'environ 200.

1.3 Différentes classes de modèles de systèmes biologiques

Ce que l'on peut conclure de cette description des réseaux génétiques, c'est avant tout la grande complexité des phénomènes sous-jacents, et de fait la difficulté de toute entreprise de modélisation. En ce qui concerne la catalyse enzymatique, nous avons vu que nous disposons de modèles communément admis et utilisés depuis plusieurs dizaines d'années. Néanmoins, l'étude de voies métaboliques, c'est-à-dire de chaînes de réactions biochimiques incluant des catalyses enzymatiques, demeure difficile lorsque le nombre de réactions augmente (ce qui est souvent le cas, même pour les organismes les plus simples). La régulation génétique à proprement parler est quant à elle un domaine de recherche récent. Il existe toutefois un grand nombre de modèles, dont la variété tend à confirmer la richesse de ces systèmes.

La compréhension des systèmes biologiques et de leur dynamique est donc un champ d'investigation extrêmement riche (on pourra par exemple consulter la bibliographie très fournie de [16]). Nous nous proposons dans cette partie de donner un aperçu des différents types de modélisation qui existent à l'heure actuelle pour représenter ces systèmes. Nous les classons en deux catégories principales : les approches continues et les approches discrètes. Nous verrons également par la suite une approche hybride, dont le but est de tirer avantage de ces deux types de modélisations.

Remarquons ici que l'ensemble des systèmes dynamiques que nous étudions dans ce mémoire est constitué de systèmes déterministes. Il existe également une approche stochastique de la modélisation de systèmes biologiques ; nous ne la considérerons pas dans ce mémoire.

¹¹<http://ecocyc.org>

¹²Nous entendons par voie métabolique une chaîne de réactions biochimiques, plus ou moins complexe, réalisant une *fonction* cellulaire.

1.3.1 Les classes de systèmes dynamiques continus

L'article [86], dont le but est de proposer un système complet d'équations différentielles modélisant le fonctionnement de l'opéron lactose montre à quel point l'entreprise de modélisation quantitative des réseaux génétiques s'avère difficile. Si nous voulons construire un système donnant lieu à une approximation numérique des courbes expérimentales, la tâche est ardue et, dans la plupart des cas impossible, tant il est encore aujourd'hui difficile d'obtenir les données numériques suffisantes. Ce type d'étude est en outre difficilement généralisable car il s'appuie sur les connaissances d'un système précis, avec ses propres particularités.

Comme nous l'avons déjà évoqué, la démarche propre aux mathématiques appliquées se situe davantage dans l'étude de *classes* de systèmes, dont la définition mathématique permet une plus ou moins grande généralité. Par définition, l'analyse de ces classes de systèmes ne donne en général qu'une approximation qualitative des comportements dynamiques des systèmes réels. Le but est donc de trouver un cadre suffisamment large, qui soit à la fois manipulable et qui présente (du moins on peut l'espérer) une gamme de comportements variés.

Ainsi, de nombreux articles et ouvrages sont consacrés à l'analyse de systèmes dynamiques incluant des sigmoïdes. En guise d'exemple, nous citons les travaux de Plahte *et al.* [51, 57, 59] qui considèrent dans [59] des systèmes donnés par des équations différentielles ordinaires du type :

$$\dot{x}_i = F_i(\mathbf{x}, \mathbf{Z}), \quad i = 1 \dots n \quad (1.17)$$

où les F_i sont supposées analytiques (généralement $F_i(\mathbf{x}, \mathbf{Z})$ est de la forme $a_i(\mathbf{Z}) - b_i x_i$), $\mathbf{x}(t)$ est un vecteur de \mathbb{R}_+^n et $Z_i = S_i(x_j, \theta, q)$ est une fonction de forme sigmoïdale vérifiant les propriétés suivantes :

- (i) $S(x, \theta, q)$ est strictement croissante (pour $x \in \mathbb{R}_+$), vaut 0 en 0 et tend vers 1 en $+\infty$,
- (ii) Lorsque $q \rightarrow 0$,

$$(\forall x < \theta, \quad S(x, \theta, q) \rightarrow 0) \quad \text{et} \quad (\forall x > \theta, \quad S(x, \theta, q) \rightarrow 1)$$

- (iii) La dérivée partielle $\frac{\partial S}{\partial x}$ dépend explicitement de $\frac{1}{q}$.

Ces propriétés sont toutes vérifiées par les fonctions de Hill : $H_{\frac{1}{q}, \theta}$ (voir annexe A). Les auteurs rajoutent encore d'autres propriétés qui ont pour but de simplifier les calculs ultérieurs (voir [59]). D'autres fonctions sont également proposées par ces auteurs (voir par exemple [57]) comme approximation des sigmoïdes : il s'agit des *logoïdes*. Une logoïde est une fonction qui est constante sur $[0, i_1]$ et vaut 0, elle est constante sur $[i_2, +\infty[$ et vaut 1. Dans l'intervalle $I = [i_1, i_2]$ (qui contient le seuil θ), la logoïde croît de façon continue de 0 à 1 (le cas le plus simple est de considérer une croissance linéaire sur I).

Ces deux types de fonctions, logoïdes et sigmoïdes admettent toutes deux comme cas limite la fonction de Heaviside centrée en θ (voir annexe A), autrement appelée la fonction *step* (pour les sigmoïdes, il suffit de faire tendre q vers 0, et pour les logoïdes, on obtient une fonction de Heaviside lorsque l'intervalle I est réduit au singleton $\{\theta\}$). Introduire des fonctions de Heaviside à la place de la variable \mathbf{Z} induit des discontinuités dans le second membre. On obtient alors le système :

$$\dot{x}_i = a_i(\mathbf{Z}) - b_i x_i, \quad i = 1 \dots n$$

qui, si l'on suppose que les a_i sont constantes par morceaux, est un système linéaire par morceaux. Ces systèmes, proposés par L. Glass ont été largement étudiés (voir par exemple [22]).

L'intérêt de tels systèmes réside notamment dans le fait que sur chaque morceau (qui est une boîte rectangulaire de $(\mathbb{R}_+^*)^n$), la linéarité des équations permet une résolution exacte, et ce, quelle que soit la dimension n .

Remarque 1.2 *Les systèmes linéaires par morceaux généraux font aujourd'hui l'objet de recherches intéressantes, car ils permettent notamment de fournir un outil de calcul puissant (le calcul hybride) permettant d'approcher des systèmes dynamiques non linéaires. On consultera à cet égard la thèse d'A. Girard [27]. Remarquons néanmoins que dans ce cadre, les systèmes considérés ne sont à priori pas définis de la même manière, en particulier les "boîtes" ne sont pas des pavés mais des simplexes.*

Une autre voie intéressante dans la modélisation de systèmes dynamiques issus de la biologie consiste en l'analyse des *S-systèmes* (voir notamment [80]). Nous reviendrons dans la troisième partie de ce mémoire sur cette classe de systèmes, en proposant notamment un algorithme permettant de rechercher les points d'équilibres de systèmes biologiques en utilisant la théorie des S-systèmes. Nous verrons qu'il est également possible de construire une approximation locale de n'importe quel système dynamique sous la forme d'un système comportant des lois de puissance. Cette approximation offre une alternative à une linéarisation classique, et peut s'avérer plus utile lors de la modélisation de systèmes biologiques.

1.3.2 La modélisation discrète

Si nous considérons l'équation (1.17) dans laquelle les termes Z_i sont composés de fonctions steps, nous avons remarqué que le système dynamique est alors affine par morceaux sur des boîtes de $(\mathbb{R}_+^*)^n$. En discrétisant le temps, nous obtenons alors un système purement discret :

$$y_i^{t+1} = \tilde{F}_i(y^t)$$

dans lequel les niveaux d'expression des gènes y_i sont à valeurs dans un ensemble fini et les fonctions \tilde{F}_i sont des applications discrètes. Il existe plusieurs formalismes dans la littérature dont le but est d'étudier la dynamique de tels systèmes appliqués à des systèmes biologiques (voir notamment [43, 76]). Dans la partie II, nous nous intéresserons à ces systèmes discrets, et nous verrons qu'ils rentrent dans le cadre général des réseaux d'automates sur un graphe à degré borné (on trouvera dans cette partie davantage de références bibliographiques).

L'un des principaux avantages de ces modèles discrets réside dans le fait qu'ils permettent de traiter des réseaux de dimension élevée. Le problème de la dimension est un problème important dans l'étude de systèmes dynamiques. Il l'est particulièrement en ce qui concerne l'application en biologie car l'étude de réseaux particuliers montre qu'il est indispensable de pouvoir manipuler des "gros" réseaux (c'est-à-dire d'une dimension supérieure à 10, voire, et c'est le cas pour les systèmes discrets, supérieure à 100 ou 1000). Le choix du modèle employé dépend donc très souvent de la dimension des réseaux que l'on veut modéliser. Plus cette dimension est importante, plus la modélisation appelle une simplification des équations et des variables du système.

Les systèmes discrets, bien qu'éloignés des systèmes biologiques initiaux, offrent de nombreuses applications. Ils présentent un intérêt qui dépasse le cadre de la régulation génétique. Nous verrons notamment dans le chapitre 4 une application intéressante qui consiste à inférer un tel système à partir d'une partie de son portrait de phase.

1.3.3 L'approche hybride

Dans le chapitre qui suit, nous proposons une méthode basée sur la théorie des systèmes hybrides. Ces systèmes, basés à la fois sur un automate discret et sur des équations différentielles offrent un outil de modélisation puissant, bien adapté aux systèmes biologiques. Nous citons ici trois articles [4, 8, 50] qui, chacun à leur manière, utilisent le formalisme des systèmes hybrides pour modéliser des réseaux de régulation génétique. Nous renvoyons au chapitre suivant pour de plus amples références sur les systèmes hybrides.

Chapitre 2

Modèle hybride de l'opéron lactose

Nous proposons dans ce chapitre, après une brève introduction au concept de *système hybride* (pour une description plus détaillée, on pourra par exemple se référer à [21, 27]) d'utiliser l'outil des systèmes hybrides pour proposer un modèle du fonctionnement de l'opéron lactose (voir chapitre précédent). Après l'établissement du modèle hybride, nous verrons notamment comment utiliser le *calcul hybride* afin de simuler un tel système, et d'obtenir des trajectoires.

2.1 L'outil des systèmes hybrides

Les systèmes hybrides constituent un outil mathématique puissant de modélisation de phénomènes naturels ou industriels complexes. Ils permettent notamment de rendre compte du comportement de systèmes dont l'évolution au cours du temps est continue et qui sont parfois soumis à de brusques variations de certaines variables. Dans un système hybride, ces variations rapides, ou *bascules*, sont représentées comme des événements discrets, et on utilise le plus souvent le terme anglais *switch* pour les nommer. L'adjonction de ces événements discrets à un système classique d'équations différentielles ordinaires est à la base du concept de système hybride.

2.1.1 Rappel de la définition d'un système hybride

Il existe dans la littérature plusieurs définitions d'un système hybride. Nous reprenons ici la définition proposée dans [27] :

Définition 2.1 *Un système hybride est un sextuplet $\mathcal{H} = (\mathcal{Q}, \mathcal{E}, \mathcal{D}, \mathcal{F}, \mathcal{G}, \mathcal{R})$, dans lequel :*

- \mathcal{Q} est un ensemble dénombrable. Il est appelé ensemble des états ou modes.
- $\mathcal{E} \subset \mathcal{Q} \times \mathcal{Q}$ est l'ensemble des arêtes ou transitions.
- $\mathcal{D} = \{D_q \mid q \in \mathcal{Q}\}$ est une collection d'ensembles $D_q \subset \mathbb{R}^n$ d'intérieurs non vides appelés domaines.
- $\mathcal{F} = \{f_q \mid q \in \mathcal{Q}\}$ est une collection de champs de vecteurs $f_q : D_q \rightarrow \mathbb{R}^n$.
- $\mathcal{G} = \{G_e \mid e \in \mathcal{E}\}$ est une collection d'ensembles nommés gardes. On a :
 $\forall e = (q, q') \in \mathcal{E}, \quad G_e \subset D_q$.
- $\mathcal{R} = \{R_e \mid e \in \mathcal{E}\}$ est une collection de fonctions telles que si $e = (q, q') \in \mathcal{E}$, alors $R_e : G_e \rightarrow \mathcal{P}(D_{q'})$. Ces fonctions sont appelées fonctions reset. On supposera que pour tout $x \in G_e$, $R_e(x)$ est non vide.

Cette définition étant établie, il nous faut à présent décrire la dynamique d'un tel système. La base du concept de système hybride se situe dans la présence de deux composantes dynamiques, l'une discrète, l'autre continue. Ces deux composantes sont interconnectées et cette interaction permet de définir le fonctionnement du système. Considérons un système hybride $\mathcal{H} = (\mathcal{Q}, \mathcal{E}, \mathcal{D}, \mathcal{F}, \mathcal{G}, \mathcal{R})$. Le couple $(\mathcal{Q}, \mathcal{E})$ peut être vu comme un graphe dont l'ensemble des sommets est l'ensemble des états \mathcal{Q} . Ce graphe constitue la composante discrète de \mathcal{H} . D'autre part, l'ensemble $\mathcal{D} \times \mathcal{F}$ forme une collection de systèmes dynamiques continus (D_q, f_q) indicés par l'ensemble des états \mathcal{Q} . Pour chaque $q \in \mathcal{Q}$, le couple (D_q, f_q) détermine un système dynamique sur le domaine D_q , sous-espace de \mathbb{R}^n . La dynamique du système à un instant donné $t \in \mathbb{R}$ dépend de l'état courant q du système. A chaque instant $t \in \mathbb{R}$ tel que $q(t) = q$, le système suit la dynamique régie par le système différentiel :

$$(S_q) \begin{cases} \frac{dx}{dt}(t) = f_q(x(t)) \\ x(t) \in D_q \end{cases}$$

Afin de définir de manière adéquate la dynamique globale, il faut coupler ces deux composantes, discrète et continue. Ce couplage se fait grâce aux gardes et aux fonctions reset. Supposons qu'à l'instant $t = 0$, le système soit dans l'état discret $q_0 \in \mathcal{Q}$. La donnée d'une condition initiale $x(0) = x_0 \in D_{q_0}$ permet de faire évoluer $x(t)$ selon le système continu (S_{q_0}) . La dynamique est donc déterminée par le mode discret q_0 dans lequel le système se trouve. Si, pour un temps $t_1 \in \mathbb{R}_+$ le vecteur $x(t_1)$ se trouve dans une garde G_{e_1} avec $e_1 = (q_0, q_1)$, nous pouvons alors décider de basculer dans l'état q_1 . Cette bascule correspond dans la composante discrète à la transition de l'état discret q_0 vers l'état discret q_1 . Nous choisissons alors une nouvelle condition initiale x_1 dans l'ensemble $R_{e_1}(x(t_1))$ (que nous avons supposé non vide). Nous sommes alors en présence d'un nouveau système dynamique (S_{q_1}) muni de la condition initiale $x(t_1) = x_1$. Cette description justifie l'appellation *hybride* de notre système : la variable continue $x(t)$ suit à chaque instant un système d'équations différentielles ordinaires (S_q) qui dépend du mode discret courant. Rétroactivement, les transitions discrètes entre les différents états se font en fonction de l'évolution continue de cette variable.

Remarque 2.1 *Cette description reste théorique et masque plusieurs problèmes pratiques difficiles. Ainsi, lors de la définition des systèmes différentiels ordinaires (S_q) , nous avons supposé que la variable $x(t)$ restait dans le domaine D_q . Si cette variable est amenée à sortir de ce domaine sans rencontrer de garde, alors aucune transition discrète n'est possible et le système se retrouve bloqué. Cet exemple montre à quel point la dynamique d'un système hybride peut être un problème complexe. Ce n'est pas l'objet de ce mémoire de rentrer dans le détail du fonctionnement général d'un système hybride. Nous nous reportons pour cela à [27] qui pose certaines bases d'une telle étude. Nous nous contenterons simplement de considérer un exemple particulier, en nous efforçant de rendre compte de sa dynamique spécifique.*

Etant donné qu'un système hybride comporte une composante discrète, il est souvent d'usage de le représenter par un *automate hybride* afin de visualiser les états discrets \mathcal{Q} et les transitions discrètes \mathcal{E} .

2.1.2 Intérêt des systèmes hybrides pour la modélisation en biologie

L'outil des systèmes hybrides est, comme nous l'avons déjà dit, très utile pour représenter des systèmes dynamiques complexes dans lesquels certaines variables physiques subissent des

variations rapides. Leur intérêt réside aussi et surtout dans le fait qu'à chaque état discret est associé un champ de vecteurs particulier, permettant ainsi de représenter un système dynamique dont les équations changent en fonction de la variable continue x . Grâce à ceci, ces systèmes ont de multiples applications en sciences physiques, que ce soit en mécanique (voir à cet effet l'exemple de la balle rebondissante [27]), en électricité (problème de l'oscillateur à valve électromagnétique, [27, 60]) etc.

Si la théorie des systèmes hybrides a de nombreux champs applicatifs en sciences physiques, elle fournit également un outil de modélisation intéressant en biologie. En effet, nous avons évoqué dans le chapitre précédent la notion de bascule biologique qui s'opère dans une cellule lorsque l'expression de certains gènes devient active ou inactive. Considérons par exemple le cas de la concentration en galactosidase $x(t)$ et en perméase $y(t)$ dans une population d'e-coli. Dans le cas où les bactéries peuvent se nourrir normalement, l'opéron lactose est réprimé et les concentrations x et y restent donc très faibles. Tant que la population peut continuer à se nourrir en glucose, ces deux valeurs restent dans un état d'équilibre. Si les conditions viennent à changer (suppression de la source en glucose et apport d'une source de lactose), on observe un brusque changement dans la dynamique de $x(t)$ et de $y(t)$ qui reflètent la dérégulation (ou induction) de l'opéron. Il semble donc clair que ces deux variables suivent (au moins) deux modes dynamiques différents, et que la transition d'un mode à un autre se fait *rapidement* (de l'ordre de quelques minutes selon [53]).

Cet exemple montre que les ordres de grandeur ne sont pas les mêmes en biologie et dans d'autres domaines. Si nous considérons l'exemple de la balle bondissante (cf. [27]), les transitions discrètes modélisent la fraction de seconde qu'il faut à la balle pour toucher le sol et rebondir. Modéliser ce petit laps de temps par une transition instantanée paraît donc être une hypothèse raisonnable. Dans le cas d'une cellule, ou d'une population de cellule, les transitions ont lieu en quelques minutes. Les considérer comme instantanées est donc une hypothèse forte, qui montre les limites de la modélisation de systèmes biologiques.

Nous proposons ici un autre exemple de réseau génétique montrant l'intérêt d'une modélisation hybride, il s'agit de l'infection par un virus parasite de la bactérie e-coli : le phage λ . Pour une description détaillée, on se référera notamment à [61]. L'enveloppe du phage est constituée de deux parties comme le montre la figure 2.1. Il contient un unique chromosome qui code, entre autres, pour les quinze protéines qui constituent son enveloppe, et qui se situe dans la tête du virus. Comme tout parasite, le phage est obligé d'infecter un hôte pour assurer sa reproduction. Il le fait en injectant dans une bactérie son matériel génétique, constitué de son chromosome λ . Après l'infection, la cellule hôte a alors deux comportements possibles. Elle peut rentrer dans un cycle appelé *cycle lytique*, c'est-à-dire que le chromosome λ est répliqué de façon intensive de manière à ce que la bactérie synthétise en grandes quantités les protéines constituant l'enveloppe du phage. Pendant une durée d'environ 45 minutes, une centaine de nouveaux virus sont ainsi fabriqués à l'intérieur de la cellule. Au bout de cette période, la cellule meurt et sa membrane se perce pour libérer les nouveaux virus qui vont ainsi chercher à infecter de nouveaux hôtes. La deuxième possibilité pour la bactérie est de rentrer dans le *cycle lysogénique* : tous les gènes du chromosome λ sont inactivés à l'exception d'un seul, dont la fonction est de synthétiser une protéine qui sert de répresseur pour l'expression des autres gènes. Le chromosome λ fait alors partie intégrante du matériel génétique de la cellule et reste sous une forme latente. La bactérie est alors appelée *lysogène* et se trouve immunisée contre toute nouvelle infection par un autre phage. Elle continue à vivre tout-à-fait normalement. En revanche, un lysogène soumis à une irradiation voit sa concentration en répresseur chuter ce qui a pour effet de perturber cet état d'équilibre de la cellule, qui rentre alors dans le cycle

lytique. Le virus peut ainsi propager une nouvelle infection dans son milieu.

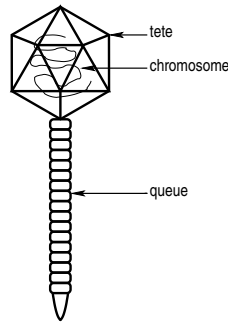


FIG. 2.1: le λ -phage est constitué de deux parties (tête et queue) et contient un seul chromosome.

Là encore, nous voyons un phénomène de bascule génétique. Selon la concentration d'une espèce critique (ici, comme dans le cas de l'opéron lactose il s'agit d'un répresseur), la bactérie se trouve dans deux états dynamiques distincts. Comme dans le cas de l'opéron, une fois le stimulus extérieur déclenché (chute de la source de glucose et introduction de lactose pour l'opéron, irradiation pour le phage λ), il suffit de quelques minutes pour voir la cellule modifier son métabolisme.

2.1.3 Pouvoir algorithmique des systèmes hybrides : le calcul hybride

Avant de rentrer dans la description proprement dite de notre modèle, nous décrivons dans ce paragraphe un autre intérêt des systèmes hybrides, qui réside dans la théorie du *calcul hybride*. Cette théorie, dont les bases sont posées dans [21] et développées dans [27] est fondée sur l'approximation de systèmes dynamiques non linéaires par un système linéaire par morceaux.

Considérons un système de n équations différentielles ordinaires autonomes sur un domaine D de \mathbb{R}^n :

$$\begin{cases} \frac{dx}{dt}(t) = f(x(t)) \\ x(0) = x_0 \in D \end{cases} \quad (2.1)$$

On supposera ici que le champ de vecteur $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ est de classe \mathcal{C}^1 sur D . Lorsque f est non linéaire, la résolution analytique de (2.1) est souvent impossible. La simulation des trajectoires d'un tel système demande donc une approximation numérique.

Les méthodes classiques de résolution numérique d'équations différentielles ordinaires non linéaires (du type Euler ou Runge-Kutta) font appel à une discrétisation temporelle. Dans la méthode d'Euler par exemple, on considère un pas de temps $\Delta t > 0$ et on pose, pour tout $p \in \mathbb{N}$, $t_p = p\Delta t$. En supposant alors le pas de temps suffisamment petit, on fait l'approximation suivante :

$$\frac{dx}{dt}(t_p) \approx \frac{x_{p+1} - x_p}{\Delta t}$$

qui nous mène à la formule de récurrence :

$$\begin{cases} x_{p+1} = x_p + \Delta t f(x_p) \\ x_0 \in U \end{cases}$$

Cette formule nous donne une discrétisation des trajectoires solutions de (2.1). Afin de prouver que ces trajectoires discrètes approchent correctement les vraies solutions, il reste à déterminer la convergence de la méthode. Dans le cas précédent (méthode d'Euler explicite), on peut montrer que la méthode est convergente (c'est-à-dire que les trajectoires approchées tendent bien vers les trajectoires réelles lorsque le pas de temps Δt tend vers 0) et que l'ordre de convergence est de 1 (voir par exemple [18]), c'est-à-dire que l'on a, pour tout $P \in \mathbb{N}$:

$$\max_{0 \leq p \leq P} |x_p - x(t_p)| = O_{\Delta t \rightarrow 0}(\Delta t)$$

Cette méthode est bien sûr rudimentaire et il existe de nombreuses méthodes présentant de bien meilleurs ordres de convergence. On peut citer notamment la méthode classique de Runge-Kutta *RK45* d'ordre 4 (voir [18]).

La méthodologie du calcul hybride se base sur un point de vue radicalement différent. La discrétisation est spatiale, c'est-à-dire qu'on ne discrétise plus la variable temporelle t mais la variable d'état x . Dans le cas scalaire ($n = 1$), cela consiste à définir un pas d'espace $\Delta x > 0$. On ne cherche plus à donner une approximation de la dérivée $\frac{dx}{dt}$ mais à donner une approximation du second membre $f(x)$.

Dans les méthodes numériques classiques, la discrétisation de la dérivée donne lieu à une équation récurrente qui approche l'équation différentielle. Dans le calcul hybride, nous décomposons l'espace des phases en une partition de segments $I_p = [p\Delta x, (p+1)\Delta x]$ sur chacun desquels nous sommes amenés à résoudre une équation différentielle approchée :

$$\frac{dx}{dt} = f_p(x) \quad (2.2)$$

Le but du calcul hybride n'est donc pas de transformer une équation différentielle en une équation récurrente, mais de proposer une simplification de cette équation différentielle en l'approximant sur chaque segment I_p par une équation différentielle dont la résolution formelle est possible explicitement. Les approximants f_p doivent donc être à la fois suffisamment généraux pour pouvoir approcher correctement la fonction de départ, et aussi suffisamment simples pour permettre une résolution exacte des équations différentielles (2.2). Les fonctions affines réalisent ce compromis, et ce, quelle que soit la dimension n du système.

Considérons le système différentiel affine suivant :

$$\begin{cases} \frac{dx}{dt} = Ax(t) + b \\ x(t_0) = x_0 \end{cases} \quad (2.3)$$

où A est une matrice réelle carrée de dimension n et b un vecteur réel de dimension n . La résolution de ce système nous donne :

$$x(t) = e^{A(t-t_0)}x_0 + e^{At} \int_{t_0}^t e^{-Au}b \, du$$

ou alors, dans le cas où A est une matrice inversible :

$$x(t) = e^{A(t-t_0)}(x_0 + A^{-1}b) - A^{-1}b$$

Il nous reste à déterminer comment discrétiser le domaine D de départ lorsque la dimension n est strictement supérieure à 1. Selon [27], la discrétisation de D doit se faire en n -simplexes de \mathbb{R}^n , dont nous rappelons ici la définition (voir par exemple [89]) :

Définition 2.2 Soient d et n deux entiers naturels tels que $n \geq d$. Un d -simplexe de \mathbb{R}^n est l'enveloppe convexe de n'importe quel ensemble de $d+1$ points affinement indépendants. C'est-à-dire que si S est un d -simplexe de \mathbb{R}^n , alors il existe un ensemble fini $\{x_1, \dots, x_{d+1}\}$ de points affinement indépendants de \mathbb{R}^n tel que :

$$S = \left\{ \sum_{i=1}^{d+1} \lambda_i x_i \mid (\forall i = 1 \dots d+1, \lambda_i \geq 0) \wedge \left(\sum_{i=1}^{d+1} \lambda_i = 1 \right) \right\}$$

En dimension 2, les 2-simplexes sont les triangles, en dimension 3, les 3-simplexes sont les tétraèdres, etc. Grâce à un maillage simplicial de D (voir définition dans [27]), nous pouvons, sur chaque simplexe du maillage définir une unique approximation affine de f qui interpole f aux sommets du simplexe. Grâce à cette interpolation, nous pouvons assurer que le champ de vecteurs affine par morceaux ainsi construit est continu et lipschitzien sur l'ensemble du domaine (voir [27]).

Nous supposons par commodité que D est un pavé de \mathbb{R}^n (ce sera le cas dans notre exemple). Nous supposons que $(D_i)_{i \in I}$ est une famille de n -simplexes de \mathbb{R}^n constituant un maillage simplicial de D . Nous définissons la *taille* des simplexes D_i par :

$$h_i \stackrel{\text{def}}{=} \sup_{x, y \in D_i} \|x - y\|_\infty$$

où la norme $\|\cdot\|_\infty$ est la norme vectorielle classique :

$$\forall x \in \mathbb{R}^n, \quad \|x\|_\infty = \max_{j \in \{1, \dots, n\}} |x_j|$$

Nous définissons ensuite le pas du maillage par :

$$h \stackrel{\text{def}}{=} \sup_{i \in I} h_i$$

On introduit alors l'approximation affine par morceaux f_h du champ non linéaire f , définie par interpolation de f aux sommets du maillage $(D_i)_{i \in I}$. L'approximation affine par morceaux du système différentiel autonome (2.1) est donc le système différentiel suivant :

$$\begin{cases} \frac{dx_h}{dt}(t) = f_h(x_h(t)) \\ x_h(0) = x_0 \in D \end{cases} \quad (2.4)$$

Remarque 2.2 Dans chaque simplexe D_i (pour $i \in I$), le champ de vecteurs f_h est affine, on pose :

$$\forall x \in D_i, \quad f_h(x) \stackrel{\text{def}}{=} A_i x + b_i$$

Nous notons S_i^j , pour j allant de 1 à $n+1$, les sommets du simplexe D_i . La matrice A_i et le vecteur b_i sont alors calculés grâce aux contraintes d'interpolation suivantes aux sommets de D_i :

$$\forall j \in \{1, \dots, n+1\}, \quad f(S_i^j) = A_i S_i^j + b_i$$

(Pour le détail du calcul de A_i et b_i , nous renvoyons à [27]).

Le théorème qui suit (tiré de [27]) justifie l'intérêt de la méthode d'approximation affine par morceaux, en donnant un résultat de convergence :

Théorème 2.1 *Soient $x(t)$ et $x_h(t)$ les solutions respectives des problèmes de Cauchy (2.1) et (2.4). On suppose que f est L -lipschitzienne. Pour tout t où $x(t)$ et $x_h(t)$ sont définies, on a :*

$$\|x(t) - x_h(t)\| \leq \frac{\varepsilon(h)}{L}(e^{L|t|} - 1)$$

et

$$\left\| \frac{dx}{dt}(t) - \frac{dx_h}{dt}(t) \right\| \leq \varepsilon(h) e^{L|t|}$$

où $\varepsilon(h)$ tend vers zéro lorsque h tend vers zéro.

Dans la suite de ce chapitre, nous proposons un modèle du fonctionnement de l'opéron lactose grâce à un système hybride. Nous utilisons ensuite le calcul hybride pour trouver des trajectoires de ce système.

2.2 Modèle hybride de l'opéron lactose

Avant de définir le système hybride, nous reprenons dans la suite les divers éléments issus de la description biologique de l'opéron lactose faite dans le chapitre précédent (voir notamment le schéma 1.8).

2.2.1 Etablissement du modèle

Choix des variables

Nous cherchons à représenter l'évolution temporelle de plusieurs variables intervenant dans le fonctionnement de l'opéron lactose. Tout d'abord, nous désignons par $x(t)$ et $y(t)$ les concentrations au temps t de β -galactosidase et de perméase. Les gènes codant pour ces deux enzymes étant contigus le long de la molécule d'ADN, nous n'utilisons qu'une seule variable, notée $a(t)$, pour représenter la concentration en ARN codant pour ces enzymes. D'autre part, nous désignons par $l(t)$ et $g(t)$ les concentrations respectives de lactose et de glucose à l'intérieur de la cellule.

Nous proposons donc un système contenant 5 variables : a , x , y , l et g . Nous ajoutons à ces variables deux fonctions $L(t)$ et $G(t)$ qui représentent respectivement les quantités de lactose et de glucose dans le milieu, c'est-à-dire à l'extérieur de la cellule. Nous supposons ces fonctions données et nous les appellerons par la suite *entrées* du système. Recensons à présent les différentes interactions entre ces variables.

Les états discrets

Nous avons choisi dans notre modèle de ne pas considérer les quantités de répresseur et d'activateur de l'opéron. Nous faisons directement dépendre l'état de l'opéron des concentrations intérieures de glucose et de lactose. En suivant la description que nous avons faite dans le chapitre précédent, nous définissons trois modes distincts :

- en absence de lactose, l'opéron est *bloqué*, ce qui signifie que les enzymes ont un taux de production très faible.
- en présence de lactose et en présence de glucose, l'opéron est *induit* : le taux de production des enzymes est faible.
- en présence de lactose et en absence de glucose dans la cellule, l'opéron est *activé*, ce qui signifie que le taux de production est fort.

Les concentrations des deux sucres, $l(t)$ et $g(t)$ restent bien sûr positives et bornées au cours du temps. On suppose qu'il existe deux constantes max_l et max_g telles que :

$$\forall t \geq 0, \quad \begin{cases} 0 \leq l(t) \leq max_l \\ 0 \leq g(t) \leq max_g \end{cases}$$

(De même nous supposons que les variables a , x et y sont bornées par trois constantes max_a , max_x et max_y).

Afin de modéliser l'absence et la présence des deux sucres, nous supposons l'existence de deux valeurs seuils, θ_l et θ_g telles que :

$$\begin{cases} 0 < \theta_l < max_l \\ 0 < \theta_g < max_g \end{cases}$$

Nous dirons donc qu'il y a présence (respectivement absence) de lactose dans la cellule au temps t si $l(t) > \theta_l$ (resp. $l(t) < \theta_l$) (de même pour le glucose). La projection de l'espace des phases dans le plan (l, g) est donc partitionnée en rectangles (voir figure suivante).

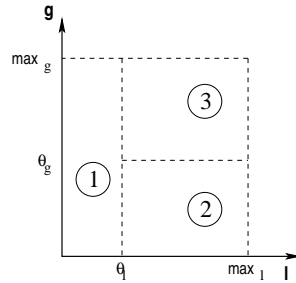


FIG. 2.2: Partition du plan de phase (l, g) . Les trois états du système hybride sont représentés : 1 désigne l'opéron bloqué, 2 l'opéron induit et activé et 3 l'opéron induit non activé.

Nous avons partitionné notre espace de phase en trois domaines qui sont des pavés de \mathbb{R}^5 :

$$\begin{cases} D_1 \stackrel{\text{def}}{=} [0, max_a] \times [0, max_x] \times [0, max_y] \times [0, \theta_l] \times [0, max_g] \\ D_2 \stackrel{\text{def}}{=} [0, max_a] \times [0, max_x] \times [0, max_y] \times [\theta_l, max_l] \times [0, \theta_g] \\ D_3 \stackrel{\text{def}}{=} [0, max_a] \times [0, max_x] \times [0, max_y] \times [\theta_l, max_l] \times [\theta_g, max_g] \end{cases}$$

Chacun de ces pavés correspond à un état de notre système hybride : l'état associé au domaine D_i sera noté q_i (i allant de 1 à 3).

Les équations différentielles

L'étape suivante dans la construction du système consiste à donner les champs de vecteurs associés à chacun des états q_i . En notant $X(t)$ le vecteur $(a(t), x(t), y(t), l(t), g(t))$, nous cherchons les fonctions $F_i : \mathbb{R}^5 \rightarrow \mathbb{R}^5$ telles que, si $X(t)$ appartient à D_i , alors X suit la dynamique donnée par l'équation différentielle :

$$\dot{X}(t) = F_i(X(t))$$

En ce qui concerne la variable a représentant la concentration en molécule d'ARN, nous considérons une évolution linéaire par morceaux. Cette hypothèse est bien entendu simplificatrice mais est communément admise dans beaucoup d'ouvrages et articles. Elle est à la base des systèmes de Glass (voir [22], voir aussi [17] et [23]). Elle est également utilisée dans des articles de modélisation biologique (voir [86]). Nous supposons donc que l'équation différentielle régissant l'évolution de la variable $a(t)$ est de la forme :

$$\dot{a}(t) = \gamma(l(t), g(t)) - \nu_a a(t) \quad (2.5)$$

où $\gamma(l, g)$ est une fonction constante par morceaux :

$$\gamma(l, g) = \begin{cases} \varepsilon, & \text{si } l < \theta_l & \text{(état } q_1) \\ p, & \text{si } l > \theta_l \text{ et } g > \theta_g & \text{(état } q_3) \\ 50p, & \text{si } l > \theta_l \text{ et } g < \theta_g & \text{(état } q_2) \end{cases}$$

(Pour les valeurs des différentes constantes, on se référera au tableau 2.1).

Le terme $\gamma(l, g)$ est le taux de production de l'ARN. Il dépend bien sûr de l'état de l'opéron, c'est-à-dire de la présence ou de l'absence de lactose et de glucose à l'intérieur de la cellule. Dans chacun des domaines D_i , l'équation (2.5) est une équation affine.

Pour l'évolution des concentrations en enzymes, nous utilisons à nouveau une croissance affine :

$$\begin{cases} \dot{x}(t) = \alpha a(t) - \nu_{\text{enz}} x(t) \\ \dot{y}(t) = \beta a(t) - \nu_{\text{enz}} y(t) \end{cases} \quad (2.6)$$

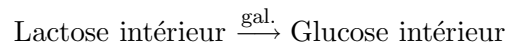
Ces équations modélisent l'étape de traduction de l'ARN en enzymes. La constante ν_{enz} représente le taux de dégradation moyen des protéines au sein de la cellule.

En ce qui concerne les deux dernières équations, qui représentent l'évolution des concentrations en lactose et glucose à l'intérieur de la cellule, nous allons utiliser la fonction de Michaelis pour modéliser le phénomène de catalyse enzymatique (voir l'expression (1.11) du chapitre précédent). En nous inspirant de [86], nous proposons les équations suivantes :

$$\begin{cases} \dot{l}(t) = V_{\text{transport},l}(t) - V_{\text{catalyse},l}(t) \\ \dot{g}(t) = V_{\text{transport},g}(t) + V_{\text{catalyse},l}(t) - V_{\text{dégradation},g}(t) \end{cases}$$

– Catalyse du lactose :

La dégradation du lactose est catalysée par la β -galactosidase. Ceci est représenté par l'équation chimique :

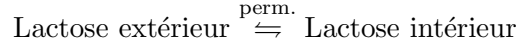


La vitesse de cette réaction est donnée par :

$$V_{\text{catalyse},l}(t) = \frac{k_1 x(t) l(t)}{K_1 + l(t)}$$

– Transport du lactose :

Le transport du lactose à l'intérieur de la cellule est catalysé par la perméase :



Comme dans [86] nous supposons que cette réaction est une réaction réversible. Nous obtenons alors une expression du terme $V_{\text{transport},l}$:

$$V_{\text{transport},l}(t) = \frac{k_2 y(t)L(t)}{K_2 + L(t)} - \frac{k_3 l(t)y(t)}{K_3 + l(t)}$$

Nous considérons en outre un phénomène que nous n'avons pas évoqué jusqu'ici : l'*exclusion de l'inducteur*. On a observé en effet que lorsque du glucose est présent à l'extérieur de la cellule, ceci a pour effet de diminuer l'action de la perméase. Ceci permet à la cellule de diminuer la pénétration du lactose tant qu'elle a à sa disposition une source d'énergie plus simple à utiliser. Nous représentons ce mécanisme par un terme multiplicatif dans l'expression précédente (voir [86]). La vitesse de transport du lactose devient :

$$V_{\text{transport},l}(t) = \frac{k_2 y(t)L(t)}{K_2 + L(t)} \underbrace{\frac{K_{\text{excl}}}{K_{\text{excl}} + G(t)}}_{\text{excl. de l'induc.}} - \frac{k_3 l(t)y(t)}{K_3 + l(t)}$$

– Transport du glucose :

Le transport du glucose à l'intérieur de la cellule est assuré par un mécanisme spécial, que nous représentons ici par le terme¹ :

$$V_{\text{transport},g}(t) = \frac{k_4 G(t)}{K_4 + G(t)}$$

– Dégénération du glucose :

La dégradation du glucose en vue de fournir à la cellule son énergie est un processus complexe. Nous faisons l'hypothèse ici que le terme $V_{\text{dégradation},g}$ est de la forme :

$$V_{\text{dégradation},g}(t) = \frac{\mu g(t)}{K + g(t)}$$

En récapitulant tout ce qui vient d'être dit, nous pouvons donner les équations différentielles régissant l'évolution des variables l et g :

$$\begin{cases} \dot{l}(t) &= \frac{k_2 y(t)L(t)}{K_2 + L(t)} \frac{K_{\text{excl}}}{K_{\text{excl}} + G(t)} - \frac{k_3 l(t)y(t)}{K_3 + l(t)} - \frac{k_1 x(t)l(t)}{K_1 + l(t)} \\ \dot{g}(t) &= \frac{k_4 G(t)}{K_4 + G(t)} + \frac{k_1 x(t)l(t)}{K_1 + l(t)} - \frac{\mu g(t)}{K + g(t)} \end{cases} \quad (2.7)$$

Enfin, nous résumons dans le tableau suivant les valeurs (et unités) des différentes constantes chimiques dont nous avons besoin. Ces valeurs viennent pour la plupart de l'article [86], certaines sont directement empruntées à cet article, tandis que d'autres sont estimées ou calculées à partir des simulations proposées à la fin de l'article.

¹Nous avons déjà évoqué au chapitre précédent que l'entrée du glucose dans la cellule était dû à la présence de complexes biochimiques nommés PTS (PhosphoTransférase System). Nous renvoyons à [86] pour des références précises concernant ce processus.

Symbole	Valeur	Unité
ε	$1,9905 \cdot 10^{-13}$	$\text{mol min}^{-1} g^{-1}$
p	$1,9905 \cdot 10^{-11}$	$\text{mol min}^{-1} g^{-1}$
ν_a	0,693	min^{-1}
ν_{enz}	0,01	min^{-1}
α	9,4	min^{-1}
β	18,8	min^{-1}
k_1	9540	min^{-1}
K_1	$4,6667 \cdot 10^{-7}$	$\text{mol } g^{-1}$
k_2	2148	min^{-1}
K_2	$2,6 \cdot 10^{-4}$	$\text{mol } g^{-1}$
k_3	71,38	min^{-1}
K_3	$4,8733 \cdot 10^{-5}$	$\text{mol } g^{-1}$
k_4	$2 \cdot 10^{-4}$	min^{-1}
K_4	$1,5 \cdot 10^{-5}$	$\text{mol } g^{-1}$
K_{excl}	$2,71 \cdot 10^{-4}$	M
μ	8	min^{-1}
K	10^{-6}	$\text{mol } g^{-1}$

TAB. 2.1: Table des valeurs numériques des différentes constantes chimiques. L'unité M désigne l'unité de concentration en moles par litres. Les masses sont exprimées en gramme de masse sèche (g DCW).

Gardes, transitions et fonctions reset

Jusqu'ici, nous avons défini les ensembles suivants :

- $\mathcal{Q} = \{q_1, q_2, q_3\}$ est l'ensemble des états discrets.
- $\mathcal{D} = \{D_1, D_2, D_3\}$ est la collection des domaines. Pour tout i , le domaine D_i est un sous-ensemble du domaine :

$$D = [0, \max_a] \times [0, \max_x] \times [0, \max_y] \times [0, \max_l] \times [0, \max_g]$$

- $\mathcal{F} = \{F_1, F_2, F_3\}$ est la collection des champs de vecteur. En notant f_i^j la j -ième coordonnée de F_i (pour j allant de 1 à 5), nous avons :

$$f_1^1(a, x, y, l, g) = \varepsilon - \nu_a a, \quad f_2^1(a, x, y, l, g) = 50p - \nu_a a, \quad f_3^1(a, x, y, l, g) = p - \nu_a a$$

$$f_i^2(a, x, y, l, g) = \alpha a - \nu_{\text{enz}} x, \quad i \in \{1, 2, 3\}$$

$$f_i^3(a, x, y, l, g) = \beta a - \nu_{\text{enz}} y, \quad i \in \{1, 2, 3\}$$

$$f_i^4(a, x, y, l, g) = \frac{k_2 y L}{K_2 + L} \frac{K_{\text{excl}}}{K_{\text{excl}} + G} - \frac{k_3 l y}{K_3 + l} - \frac{k_1 x l}{K_1 + l}, \quad i \in \{1, 2, 3\}$$

$$f_i^5(a, x, y, l, g) = \frac{k_4 G}{K_4 + G} + \frac{k_1 x l}{K_1 + l} - \frac{\mu g}{K + g}, \quad i \in \{1, 2, 3\}$$

Afin de définir un système hybride complet, il nous faut encore définir l'ensemble des transitions discrètes, l'ensemble des gardes et l'ensemble des fonctions reset.

Pour l'ensemble des transitions, nous posons :

$$\mathcal{E} = \{(q_1, q_2), (q_2, q_1), (q_1, q_3), (q_3, q_1), (q_2, q_3), (q_3, q_2)\}$$

Les gardes sont données par les frontières des domaines. $\mathcal{G} = \{G_e \mid e \in \mathcal{E}\}$ avec :

$$\begin{cases} G_{q_1, q_2} = G_{q_2, q_1} = \{(a, x, y, l, g) \in D \mid l = \theta_l \text{ et } g < \theta_g\} \\ G_{q_1, q_3} = G_{q_3, q_1} = \{(a, x, y, l, g) \in D \mid l = \theta_l \text{ et } g > \theta_g\} \\ G_{q_2, q_3} = G_{q_3, q_2} = \{(a, x, y, l, g) \in D \mid l > \theta_l \text{ et } g = \theta_g\} \end{cases}$$

Enfin, l'ensemble $\mathcal{R} = \{R_e \mid e \in \mathcal{E}\}$ est donné par :

$$\forall e \in \mathcal{E}, \forall x \in G_e, \quad R_e(x) = \{x\}$$

2.2.2 Etude des équilibres

Dans la partie précédente, nous avons défini les constantes dont nous avons besoin pour exprimer les équations de la dynamique. Ces constantes sont issues de mesures faites par des biologistes ou d'estimations. En revanche, les paramètres max_l , max_g ainsi que les seuils θ_l et θ_g viennent directement du choix de notre modélisation et n'ont à priori aucune réalité biologique. Afin de leur donner une valeur cohérente, nous allons étudier le portrait de phase des systèmes dynamiques :

$$\dot{X} = F_i(X)$$

Nous cherchons pour cela les points d'équilibre, c'est-à-dire les points $X_* = (a_*, x_*, y_*, l_*, g_*)$ appartenant à \mathbb{R}_+^5 tels que $F_i(X_*) = 0$. Pour $i \in \{1, 2, 3\}$, nous sommes donc ramenés à la résolution du système :

$$\begin{cases} f_i^1(a, x, y, l, g) = 0 \\ f_i^2(a, x, y, l, g) = 0 \\ f_i^3(a, x, y, l, g) = 0 \\ f_i^4(a, x, y, l, g) = 0 \\ f_i^5(a, x, y, l, g) = 0 \end{cases}$$

Les trois premières équations étant linéaires, leur résolution est immédiate. Seule la première coordonnée de F_i dépend explicitement de l'état discret q_i du système. Nous trouvons donc trois solutions pour la première équation, chacune correspondant à un état du système hybride :

$$a_*^1 = \frac{\varepsilon}{\nu_a} > 0, \quad a_*^2 = \frac{50p}{\nu_a} > 0, \quad a_*^3 = \frac{p}{\nu_a} > 0$$

Pour la deuxième et la troisième équation, on trouve :

$$x_*^i = \frac{\alpha a_*^i}{\nu_{enz}} > 0 \quad \text{et} \quad y_*^i = \frac{\beta a_*^i}{\nu_{enz}} > 0 \quad i \in \{1, 2, 3\}$$

En injectant ces valeurs dans la quatrième équation, nous nous ramenons à l'équation polynomiale du second degré :

$$P_{L,G}(l) = A(L, G)l^2 + B(L, G)l + C(L) = 0$$

Cette équation dépend des entrées $L(t)$ et $G(t)$, qui correspondent aux concentrations extérieures de lactose et de glucose. Nous allons considérer que ces entrées sont constantes par morceaux, et sont à valeurs dans $\{0, E\}$, où E est un réel positif que nous fixons à $5,515 \cdot 10^{-3}$. Cette valeur est calculée à partir des simulations proposées dans [86]. Cette hypothèse nous permet de considérer quatre cas, selon la valeur prise par le couple (L, G) . En utilisant le logiciel de calcul formel *Maple*, nous trouvons l'expression des coefficients du polynôme $P_{L,G}$ ainsi que leurs signes :

$$\forall L, G \in \{0, E\}, \quad A(L, G) > 0, \quad B(L, G) > 0 \text{ et } C(L) \leq 0$$

Grâce à la règle de Descartes (voir [52] et la remarque 1.1 du chapitre précédent), nous déduisons que quelle que soit la valeur prise par les entrées L et G , le polynôme $P_{L,G}$ admet une unique racine réelle positive ou nulle. Nous notons cette racine $l_*(L, G)$.

Enfin, en utilisant les valeurs numériques du tableau 2.1, nous trouvons que la cinquième équation admet une unique racine positive en la variable g que nous notons $g_*^i(L, G)$. Cette racine dépend à la fois des entrées L et G , et aussi de l'état discret q_i du système hybride.

En conclusion, pour chaque état discret q_i du système, nous avons trouvé un point d'équilibre positif dépendant des valeurs des entrées L et G . Il y a donc quatre points d'équilibre par état, selon que $(L, G) = (0, 0)$, $(0, E)$, $(E, 0)$ ou (E, E) . En utilisant les valeurs numériques données dans le tableau 2.1, nous pouvons calculer les coordonnées de ces points d'équilibre. Nous représentons sur la figure 2.3 les différentes valeurs de $l_*(L, G)$ et de $g_*^i(L, G)$ dans les trois états q_1, q_2, q_3 pour les différentes valeurs du couple (L, G) . En ce qui concerne les valeurs des seuils, nous choisissons (voir figure 2.3) :

$$\begin{cases} \theta_l & = 10^{-7} \\ \theta_g & = 0,2 \cdot 10^{-8} \\ \max_l & = 4 \cdot 10^{-7} \\ \max_g & = 1,2 \cdot 10^{-8} \end{cases}$$

Il nous reste à déterminer la stabilité de ces points d'équilibre. Pour cela, nous allons calculer les valeurs propres de la matrice jacobienne des fonctions F_i . Nous rappelons l'expression de F_i :

$$F_i(a, x, y, l, g) = \begin{pmatrix} \gamma_i - \nu_a a \\ \alpha a - \nu_{\text{enz}} x \\ \beta a - \nu_{\text{enz}} x \\ \frac{k_2 y L}{K_2 + L} \frac{K_{\text{excl}}}{K_{\text{excl}} + G} - \frac{k_3 l y}{K_3 + l} - \frac{k_1 x l}{K_1 + l} \\ \frac{k_4 G}{K_4 + G} + \frac{k_1 x l}{K_1 + l} - \frac{\mu g}{K + g} \end{pmatrix}$$

où γ_i est égal à ε si $i = 1$, à p si $i = 3$ et à $50p$ si $i = 2$.

La jacobienne de F_i au point $X = (a, x, y, l, g)$ est égale à :

$$J_{F_i}(X) = \begin{pmatrix} -\nu_a & 0 & 0 & 0 & 0 \\ \alpha & -\nu_{\text{enz}} & 0 & 0 & 0 \\ \beta & 0 & -\nu_{\text{enz}} & 0 & 0 \\ 0 & -\frac{k_1 l}{K_1 + l} & g_1(l, L, G) & g_2(x, y, l) & 0 \\ 0 & \frac{k_1 l}{K_1 + l} & 0 & \frac{k_1 K_1 x}{(K_1 + l)^2} & -\frac{\mu K}{(K + g)^2} \end{pmatrix}$$

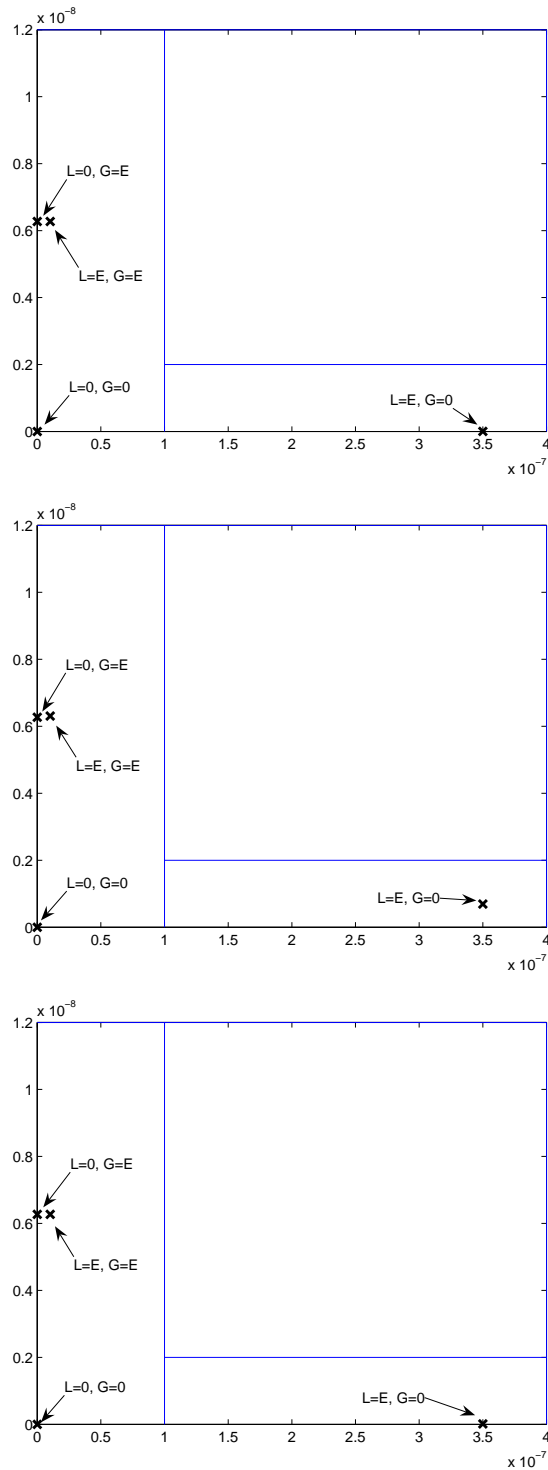


FIG. 2.3: Projection de l'espace des phases sur le plan (l, g) . Les croix désignent les points d'équilibre en fonction des valeurs des entrées L et G . On a placé les points d'équilibres de F_1 dans le premier graphique, ceux de F_2 dans le deuxième et ceux de F_3 dans le troisième. Les valeurs numériques des seuils sont données par : $\theta_l = 10^{-7}$, $\theta_g = 0,2 \cdot 10^{-8}$, $max_l = 4 \cdot 10^{-7}$ et $max_g = 1,2 \cdot 10^{-8}$.

où les fonctions g_1 et g_2 sont égales à :

$$g_1(l, L, G) = \frac{k_2 L K_{\text{excl}} K_3 + k_2 L K_{\text{excl}} l - k_3 L K_{\text{excl}} l - k_3 L G l - k_3 K_2 K_{\text{excl}} l - k_3 K_2 G l}{(L + K_2)(K_{\text{excl}} + G)(K_3 + l)}$$

$$g_2(x, y, l) = -\frac{k_3 K_3 K_1^2 y + 2k_3 K_3 K_1 y l + k_3 K_3 y l^2 + k_1 K_3^2 K_1 x + 2k_1 K_3 K_1 x l + k_1 K_1 x l^2}{(K_3 + l)^2 (K_1 + l)^2}$$

La matrice $J_{F_i}(X)$ étant triangulaire inférieure, ses valeurs propres λ_j , pour $j \in \{1, \dots, 5\}$ sont égales aux termes diagonaux :

$$\begin{cases} \lambda_1 = -\nu_a & < 0 \\ \lambda_2 = -\nu_{\text{enz}} & < 0 \\ \lambda_3 = -\nu_{\text{enz}} & < 0 \\ \lambda_4 = g_2(x, y, l) & < 0, \text{ pour tout } x, y \text{ et } l \\ \lambda_5 = -\frac{\mu K}{(K + g)^2} & < 0, \text{ pour tout } g \end{cases}$$

Nous en déduisons que pour tout $X \in \mathbb{R}_+^5$, les valeurs propres de la matrice $J_{F_i}(X)$ sont réelles et strictement négatives. En particulier, si X_* est un point d'équilibre positif de F_i , alors X_* est un point d'équilibre linéairement stable et donc asymptotiquement stable (nous utilisons la terminologie classique dans l'étude des systèmes dynamiques, voir par exemple [34]).

2.2.3 Interprétation biologique du modèle hybride

Comme l'indique la figure 2.3, la localisation des points d'équilibre du système dépend essentiellement des variables d'entrée L et G . Il y a donc quatre comportements possibles. Etudions plus en détail ces comportements :

- Premier cas : $L = 0$ et $G = 0$,

Dans ce cas, la cellule n'a à sa disposition ni glucose, ni lactose. Comme nous n'avons envisagé dans notre modèle aucune autre solution de remplacement, les variables l et g vont donc converger vers zéro, et ce quel que soit l'état courant du système. Ce cas est donc un cas particulier, il n'a pas vraiment de correspondance dans la nature (car comme nous l'avons dit, la cellule met alors en place d'autres mécanismes que nous n'avons pas représentés) et il est donc inhérent aux choix de modélisation que nous avons faits.

- Deuxième cas : $G = E$,

Lorsque $G = E$, cela signifie que la cellule a à sa disposition du glucose. Comme l'indique la figure 2.3, quel que soit l'apport en lactose, le point d'équilibre se situe dans l'état q_1 , ce qui signifie que l'opéron est bloqué.

Dans le cas où $L = 0$, la valeur d'équilibre du lactose intérieur vaut zéro : il n'y a aucun apport de lactose. En revanche, lorsque $L = E$, alors du lactose pénètre dans la cellule, ce qui a pour effet d'induire l'opéron. Toutefois, le phénomène d'exclusion de l'inducteur que nous avons décrit précédemment engendre une perte de l'efficacité de la perméase, ce qui entraîne un faible apport de lactose. Cet effet s'ajoute à la non activation de l'opéron induite par la présence de glucose dans la cellule. Le lactose à l'équilibre atteint donc une valeur non nulle, mais qui reste faible (bien inférieure au seuil θ_l comme le montre la figure 2.3).

Par conséquent, quelle que soit la valeur de L , l'hypothèse $G = E$ implique automatiquement une chute de la concentration en lactose intérieur et donc le blocage de l'opéron.

- Troisième et dernier cas : $L = E$ et $G = 0$,

Ce cas correspond à la phase qui consiste, pour la cellule, à déclencher l'opéron, de manière à utiliser le lactose pour fournir de l'énergie. Comme le montre la figure 2.3, les points d'équilibre correspondant à ces conditions se situent tous dans l'état q_2 , qui est l'état d'induction activée de l'opéron. Tant que les conditions extérieures restent inchangées, l'opéron restera activé de manière à faire pénétrer le lactose et à en extraire l'énergie nécessaire.

Le système hybride que nous avons défini reprend donc les observations que nous avons faites du comportement de l'opéron. Ce modèle nous permet de reproduire les principales caractéristiques du comportement de l'opéron, telles qu'elles sont connues à l'heure actuelle.

2.3 Calcul de trajectoires

Nous présentons dans cette partie un travail qui est en cours d'élaboration et qui consiste à calculer des trajectoires du système hybride. Pour cela, nous utilisons la technique du calcul hybride que nous avons décrite précédemment. Nous utilisons le code élaboré par A. Rondepierre [64]. Ce programme utilise le logiciel de calcul formel Maple.

2.3.1 Hybridisation et calcul des temps de sortie

La première étape consiste en l'*hybridisation* de notre système. Cette phase consiste à calculer un maillage simplicial des différents domaines D_1 , D_2 et D_3 . On trouvera sur la figure 2.4 la projection sur le plan (l, g) du maillage que nous avons calculé grâce au programme.

Une fois ce maillage défini, nous interpolons le champ de vecteurs aux sommets des simplexes afin de calculer les systèmes linéaires associés à chaque élément du maillage. Une fois ceci effectué, le calcul de l'exécution du système hybride à partir d'une condition initiale $X^0 = (a^0, x^0, y^0, l^0, g^0) \in D$ donnée se déroule de la façon suivante :

1. Nous calculons le simplexe du maillage qui contient le point X^0 . Nous le notons S_0 (nous utiliserons dans la suite le terme de *cellule*).
2. Nous calculons alors la trajectoire du système linéaire :

$$\dot{X} = A_0 X + b_0$$

issue du point initial X^0 , dans la cellule S_0 .

3. Nous calculons ensuite le *temps de sortie*, c'est-à-dire le temps que met la trajectoire pour atteindre une frontière de S_0 . Pour cela, nous notons $(\mathcal{F}_0^i)_{i \in \{1, \dots, 6\}}$ les 6 faces du simplexe S_0 . Alors le temps de sortie se calcule par la formule :

$$t_0 = \min_{i \in \{1, \dots, 6\}} \{t \mid X(t) \in \mathcal{F}_0^i\}$$

Si ce temps est infini, cela signifie que la trajectoire ne sort jamais du simplexe et donc qu'elle a rencontré un attracteur.

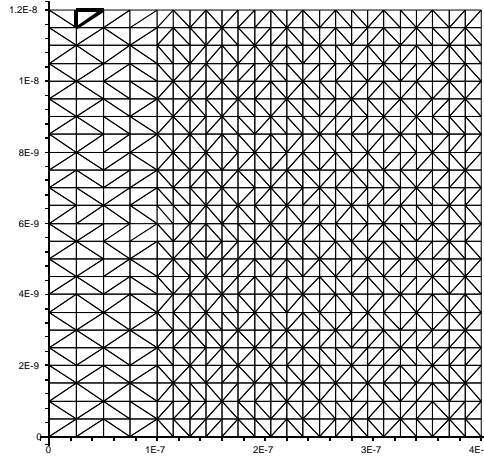


FIG. 2.4: Maillage simplicial du domaine D (il s'agit en fait ici de la projection de ce maillage sur le plan (l, g)).

4. Nous calculons alors le point de sortie : $X^1 = X(t_0)$ ainsi que la cellule de sortie S_1 (i.e. l'unique simplexe adjacent à S_0 ayant pour face la face de sortie). Nous itérons alors ce processus à partir de X^1 et de S_1 .

Dans cet algorithme, la phase critique est la phase du calcul du temps de sortie et de la face de sortie. Etant donnée la forme de $X(t)$ (rappelons qu'il s'agit dans le cas général de la solution d'un système linéaire et donc qu'elle contient des termes exponentiels), le calcul du temps de sortie se fait *numériquement*. Ceci pose un problème dans certains cas. Le problème du calcul des temps de sortie est un problème général inhérent au calcul hybride, qui mérite une attention particulière (voir à ce titre [27]). Nous voulons simplement ici calculer des trajectoires d'un exemple particulier qui est l'opéron lactose, nous n'irons donc pas plus avant dans l'étude de ce problème.

Devant les problèmes numériques rencontrés lors du calcul des trajectoires aux voisinages des frontières des cellules, nous avons décidé d'utiliser un algorithme d'exécution différent. Cet algorithme, au lieu de calculer une exécution continue du système, calcule une exécution discrète, c'est-à-dire une suite de cellules :

0. La toute première étape consiste à calculer, comme dans l'algorithme précédent, la cellule initiale S_0 qui contient le point X^0 .
1. Etant donnée S_0 , nous calculons le barycentre Y^0 de sa projection sur le plan (l, g) :

$$Y_4^0 = \frac{a_1 + a_2 + a_3}{3} \quad \text{et} \quad Y_5^0 = \frac{b_1 + b_2 + b_3}{3}$$

(où les points (a_i, b_i) sont les projetés des sommets du simplexe S_0 sur la quatrième et la cinquième coordonnées)

2. Nous calculons ensuite la face de sortie, si elle existe, de la trajectoire issue du point Y^0 appartenant à la cellule S_0 . Si la face de sortie n'est pas définie, alors l'algorithme se termine.

3. Nous calculons alors la cellule S_1 adjacente de S_0 qui admet pour face la face de sortie. Nous reprenons alors l'algorithme à partir de l'étape 1 avec comme nouvelle cellule la cellule S_1 .

Cet algorithme nous permet donc d'exhiber une trajectoire discrète du système hybride, c'est-à-dire donner un enchaînement de cellules censé reproduire le comportement de l'opéron. Certes, le fait de considérer chaque nouveau point initial comme étant le barycentre d'une cellule, nous fait perdre l'exactitude de cette trajectoire et nous ne pouvons garantir que la suite de cellules trouvée correspond bien à la suite de cellules empruntées par la "vraie" trajectoire continue. Ceci étant, il semble clair que la probabilité que ces deux trajectoires discrètes soient différentes diminue lorsque le pas du maillage s'approche de zéro. Nous avons donc choisi un maillage assez fin (voir figure 2.4) qui contient une vingtaine de points intermédiaires le long des frontières de D .

2.3.2 Exemples d'exécutions du système

Nous proposons trois trajectoires calculées selon l'algorithme précédent (figures 2.5, 2.6 et 2.7). Nous avons choisi pour ces trajectoires trois conditions extérieures différentes. Pour la première, le lactose et le glucose extérieur sont à une valeur haute, ce qui entraîne la trajectoire vers l'état q_1 de blocage de l'opéron. Dans la seconde ils valent tous les deux zéro, ce qui a pour effet de faire chuter les concentrations internes vers zéro. Enfin, la troisième expérience est réalisée en présence de lactose mais sans glucose, ce qui entraîne la trajectoire vers l'état q_2 , c'est-à-dire l'état d'induction activée.

2.3.3 Limites du modèle

Le principal défaut de cette modélisation réside bien sûr dans l'utilisation de l'algorithme calculant les trajectoires discrètes. La résolution des équations différentielles engendrées par la cinétique chimique est complexe, et les ordres de grandeur des différents paramètres induisent des erreurs numériques dont il est difficile de se débarrasser. Comme nous l'avons indiqué, le calcul des points de sortie est encore aujourd'hui un problème épineux, dont la résolution numérique n'est pas garantie dans tous les cas.

Toutefois, la modélisation sous forme de système hybride est prometteuse car elle permet, selon nous, de rendre compte de la complexité du phénomène de régulation génétique. Comme toute approche quantitative, l'adaptation de cette méthode à un autre réseau de régulation demande bien sûr un investissement important. Il faut en effet rassembler le maximum d'informations biologiques *et* numériques sur les systèmes différentiels régissant les dynamiques des réactions biochimiques en jeu. Néanmoins, la partie discrète (l'automate) du système hybride peut être trouvée plus facilement, à partir de la description biologique du fonctionnement du système.

Dans la deuxième partie de ce rapport, nous nous intéressons de plus près à la modélisation discrète, qui constitue depuis une trentaine d'années une approche tout-à-fait pertinente du phénomène de la régulation génétique.

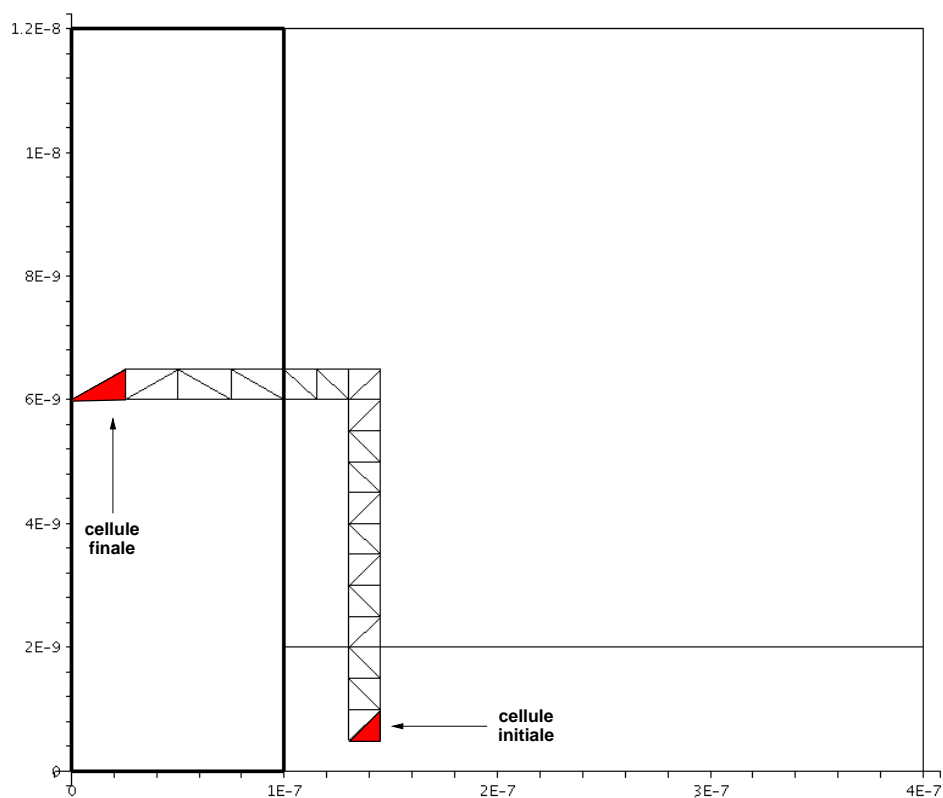


FIG. 2.5: Trajectoire discrète du système avec apport des deux sucres.

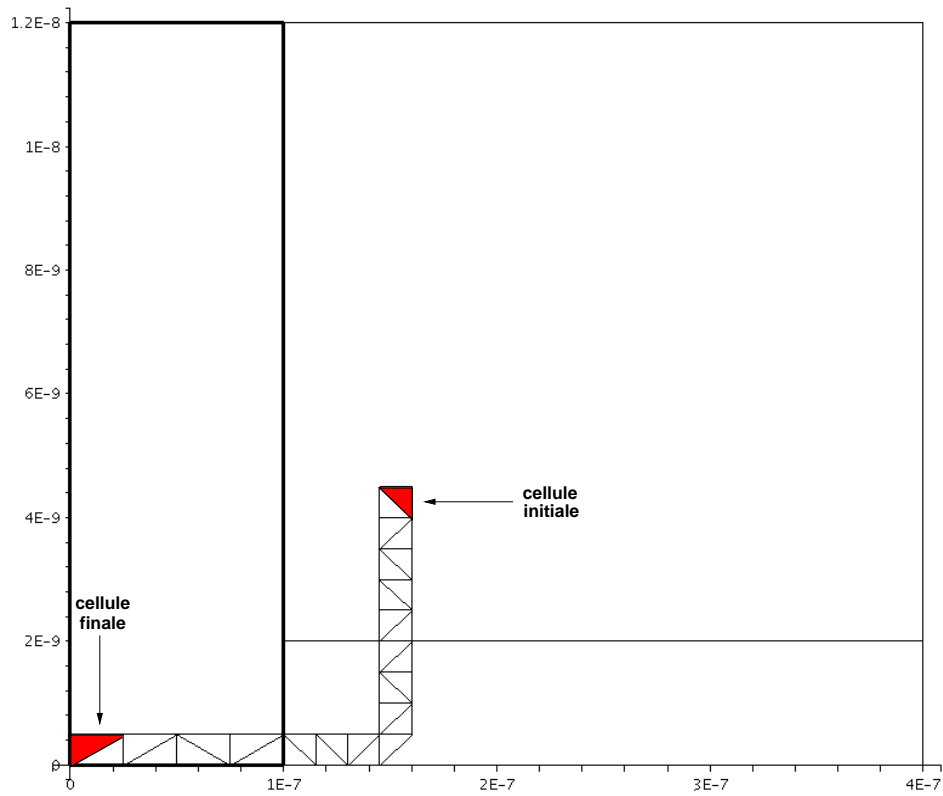


FIG. 2.6: Trajectoire discrète du système en l'absence de glucose et de lactose extérieurs.

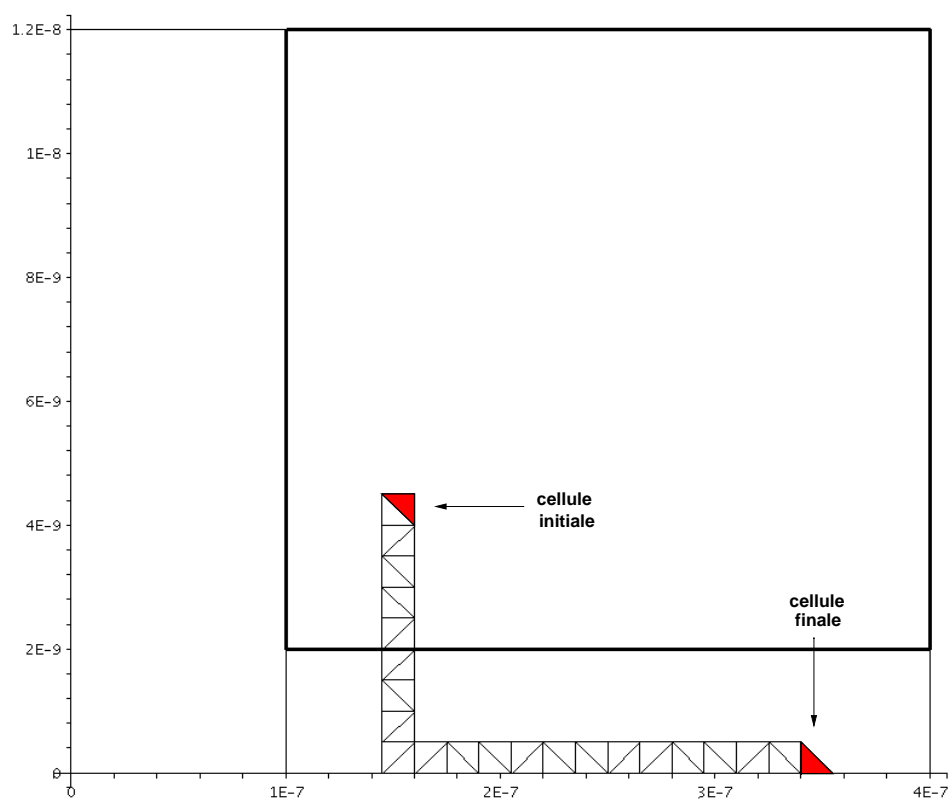


FIG. 2.7: Trajectoire discrète du système en présence de lactose et en absence de glucose (induction activée).

Deuxième partie

Réseaux de régulation et Systèmes
dynamiques discrets

Chapitre 3

Les réseaux d'automates booléens

Dans la démarche de modélisation de processus réels, l'outil mathématique des systèmes dynamiques joue un rôle fondamental. Le plus souvent, la complexité des phénomènes étudiés impose une première étape de simplification durant laquelle on essaie d'isoler un système dont on veut étudier l'ensemble des comportements, et surtout l'évolution de ces comportements au cours du temps. Au cours du vingtième siècle (on peut remonter à Von Neumann [55]) et l'apparition de l'outil informatique, est apparue la méthodologie des systèmes dynamiques discrets, se basant sur l'observation suivante :

Prenons un ensemble fini d'éléments (dits *processeurs*, *automates*, etc.). Associons à chacun de ces éléments un ensemble fini d'*états*. Considérons enfin un temps discret $t \in \mathbb{N}$. A chaque top d'horloge, tout les éléments (ou seulement une partie d'entre eux) changent d'état, chaque élément respectant une règle qui lui est propre, et qui dépend de l'état courant d'un certain nombre d'autres éléments du réseau. Si l'on itère ce processus indéfiniment, on obtient une *trajectoire* du réseau qui dépend uniquement du mode opératoire choisi (quels sont les processeurs qui évoluent à chaque top d'horloge), de la topologie du réseau (pour chaque processeur, quels sont les autres processeurs qui l'influencent, et selon quelle règle) et de la configuration initiale. Un tel système porte le nom de *réseau d'automates*. Au cours du siècle, de nombreux travaux ont porté sur l'étude de la dynamique de tels systèmes. Portant entre autre les noms d'automates cellulaires ([48], [63], voir aussi les travaux de S. Wolfram [84, 85]) ou encore de réseaux neuronaux (Mac Culloch et Pitts 1943, Hopfield 1982), les systèmes dynamiques discrets ont été beaucoup étudiés car ils représentent un outil de modélisation intéressant et surtout facilement manipulable, que ce soit à la main ou par un ordinateur.

Nous nous proposons dans ce chapitre d'apporter une définition formelle de ces systèmes, ainsi que de montrer l'intérêt qu'ils représentent dans la modélisation en biologie. Afin d'étudier la dynamique d'un réseau d'automate discret, nous sommes amenés à définir deux graphes fondamentaux : le graphe d'interaction (ou de connexion) et le graphe de transition (ou d'itération). Le premier définit un réseau discret au sens, évoqué plus haut, de la topologie du réseau : quels sont, pour chaque automate, le ou les automates qui l'influencent, et selon quelles règles. Il s'agit donc d'une définition statique. Le deuxième graphe, quant à lui, nous donne une définition de la dynamique au sens où il nous fournit la fonction de transition d'une configuration du réseau à la suivante. Il s'agit de deux visions différentes du même objet. Il sera intéressant pour nous de montrer les liens qui existent entre les deux afin de donner une description globale de la notion de système dynamique discret. On trouve plusieurs approches intéressantes dans la littérature, notamment dans les ouvrages [62] et [82]. Nous verrons également les différentes définitions de réseaux de régulation génétique discrets qui ont été proposées [43, 76].

3.1 Rappels et définitions

3.1.1 Rappels sur les itérations sur un ensemble dénombrable

Dans la suite de ce chapitre, nous allons être amenés à définir des *réseaux d'automates*. Comme nous allons le voir, étudier la dynamique de tels réseaux revient à itérer une application sur un ensemble dénombrable. Une telle itération constitue un *système dynamique*, qualifié de discret puisque son ensemble d'états est un ensemble dénombrable. Avant de définir la notion de réseau d'automates, nous faisons ici quelques rappels sur la classe des systèmes dynamiques discrets. La notion de système dynamique discret s'inscrit dans le cadre plus général des systèmes dynamiques. La définition qui suit propose une version discrète de la définition d'un système dynamique continu (cf. [31]).

Définition 3.1 *Un système dynamique discret est défini par un couple (V, ϕ) où V est un ensemble dénombrable et ϕ un flot discret sur V , c'est-à-dire une application de $\mathbb{Z} \times V$ dans V vérifiant :*

- $\forall v \in V, \quad \phi(0, v) = v$
- $\forall v \in V, \forall t, s \in \mathbb{Z}, \quad \phi(t, \phi(s, v)) = \phi(t + s, v)$

Remarque 3.1 *On utilise parfois pour le flot une notation indicée : $\phi = (\phi_t)_{t \in \mathbb{Z}}$, où ϕ_t est l'application $\phi(t, \cdot)$. Les deux conditions précédentes signifient que l'application :*

$$\begin{array}{ccc} \mathbb{Z} & \longrightarrow & V^V \\ t & \longmapsto & \phi_t \end{array}$$

est un morphisme de groupe du groupe $(\mathbb{Z}, +)$ vers le groupe (V^V, \circ) .

Cette définition provient de la définition *géométrique* d'un système dynamique continu. Un système dynamique discret est donné par un flot *global* discret sur un ensemble dénombrable (fini ou infini). Etant donné une configuration initiale $v_0 \in V$, au bout d'un temps $t \in \mathbb{N}$, le système a évolué vers une configuration $v(t)$ donnée par :

$$v(t) = \phi(t, v_0)$$

Nous définissons *l'orbite* de $v_0 \in V$ comme étant l'ensemble

$$\mathcal{O}(v_0) = \{\phi(t, v_0) \mid t \in \mathbb{Z}\}$$

Nous sommes parfois amenés à étudier un flot seulement sur des temps positifs. On parle alors de *semi-flot*. Un semi-flot est une application $\phi : \mathbb{N} \times V \rightarrow V$ vérifiant :

- $\forall v \in V, \quad \phi(0, v) = v$
- $\forall v \in V, \forall t, s \in \mathbb{N}, \quad \phi(t, \phi(s, v)) = \phi(t + s, v)$

De la même manière, on appellera *orbite positive* de $v_0 \in V$ l'ensemble

$$\mathcal{O}^+(v_0) = \{\phi(t, v_0) \mid t \in \mathbb{N}\}$$

L'analogie avec les systèmes continus ne s'arrête pas là. En effet, étant donné un système $S = (V, \phi)$ donné par la définition 3.1, nous pouvons exprimer une équation d'évolution discrète (analogue d'une équation différentielle) déterminant l'ensemble des orbites de S .

Définition 3.2 Soit un système dynamique discret $S = (V, \phi)$. On note F l'application de V dans V définie par : $F(v) = \phi(1, v)$ pour $v \in V$ (i.e. $F = \phi_1$).

Alors, on peut associer à S l'équation récurrente suivante, dite équation d'évolution :

$$(E_{v_0}) \begin{cases} x^0 = v_0 \in V \\ x^{t+1} = F(x^t), \quad \text{pour tout } t \in \mathbb{Z} \end{cases} \quad (3.1)$$

où $x = (x^t)_{t \in \mathbb{Z}}$ est une suite d'éléments de V . En d'autres termes, la suite x vérifie :

$$\forall t \in \mathbb{Z}, \quad x^t = \overbrace{F \circ \dots \circ F}^{t \text{ fois}}(v_0) = F^t(v_0)$$

En partant de la définition 3.1, nous sommes arrivés à une définition d'un système dynamique discret sous la forme de l'itération discrète d'une application $F : V \rightarrow V$. Dans l'analogie avec les systèmes continus, l'itération de cette application correspond à une équation différentielle, l'application F jouant le rôle d'un *champ de vecteurs*¹. Nous remarquons, grâce à nos définitions que l'application F ainsi définie est inversible, en effet :

$$F \circ \phi_{-1} = \phi_1 \circ \phi_{-1} = \phi_0 = \text{id}_V$$

donc F^{-1} existe et est égal à ϕ_{-1} .

Intéressons-nous à présent au problème inverse. C'est-à-dire, étant donnée une application sur un ensemble dénombrable $F : V \rightarrow V$, en quel sens peut-on dire que l'itération de F sur V définit un système dynamique discret au sens de la définition 3.1 ? En d'autres termes, nous cherchons à construire à partir de F un flot ϕ . Pour se faire il nous faut étudier deux cas, selon que F est ou non inversible.

Premier cas : F est inversible. Dans ce cas, on peut construire un flot global ϕ tel que les orbites de (V, ϕ) soient les suites des itérées de F .

En effet, en posant $\phi_1 = F$ puis en utilisant les propriétés de morphismes de $t \mapsto \phi_t$, on

construit par induction sur $t \in \mathbb{N}$ la suite $(\phi_t)_{t \in \mathbb{N}}$ (on a : $\phi_t = \overbrace{F \circ \dots \circ F}^{t \text{ fois}} = F^t$).

F étant inversible, en posant $\phi_{-1} = F^{-1}$, on construit par induction sur $-t$ ($t \in \mathbb{N}$) la suite $(\phi_{-t})_{t \in \mathbb{N}}$ (avec $\phi_{-t} = F^{-t}$).

Deuxième cas : F n'est pas inversible. Alors on ne peut pas construire de flot global. En revanche, on peut construire un semi-flot $(\phi_t)_{t \in \mathbb{N}}$ dont les orbites (positives) sont les suites solutions de l'itération de F sur \mathbb{N} (on le voit facilement en reprenant la première étape ci-dessus). Ce semi-flot nous donne les orbites positives du système dynamique (c'est-à-dire pour les temps positifs), mais nous ne pouvons pas "remonter" le temps.

Désormais, nous considérerons des systèmes dynamiques donnés par un couple (V, F) , où V est un ensemble dénombrable et F une application de V dans V . Les **trajectoires** d'un tel système sont donc les solutions de l'itération :

$$\begin{cases} x^0 = v_0 \\ x^{t+1} = F(x^t), \quad \text{pour tout } t \in \mathbb{N} \end{cases} \quad (3.2)$$

¹Pour être tout-à-fait précis, l'analogie discret d'une équation différentielle est $x^{t+1} - x^t = G(x^t)$. L'analogie du champ de vecteur est donc ici l'application $G = F - \text{id}_V$.

Remarque 3.2 Nous avons dit plus haut que F joue le rôle d'un champ de vecteurs. Nous remarquons ici que nous n'avons considéré que des applications autonomes, c'est-à-dire indépendantes du temps t . Dans le cas où l'application dépend aussi du temps (cas non autonome), nous sommes confrontés à l'itération :

$$\begin{cases} x^0 = v_0 \\ x^{t+1} = G(t, x^t), \quad \text{pour tout } t \in \mathbb{Z} \end{cases} \quad (3.3)$$

où l'application G va de $\mathbb{Z} \times V$ dans V .

Nous montrons ici que l'étude d'une telle équation peut toujours se ramener à l'étude d'une équation autonome (3.1) sur un ensemble dénombrable V' . Nous pourrions donc dans la suite considérer des systèmes autonomes sans perte de généralité.

En effet, considérons l'itération non autonome (3.3) et posons $y^t = (x^t, t)$. y^t appartient à l'ensemble $V \times \mathbb{Z}$ qui est dénombrable. En appliquant (3.3), on obtient :

$$y^{t+1} - y^t = \begin{pmatrix} G(t, x^t) - x^t \\ 1 \end{pmatrix} = H(y^t)$$

où l'application

$$\begin{aligned} H : V \times \mathbb{Z} &\longrightarrow V \times \mathbb{Z} \\ (y_1, y_2) &\longmapsto (G(y_2, y_1) - y_1, 1) \end{aligned}$$

ne dépend pas explicitement du temps t .

En posant $F(y^t) = y^t + H(y^t)$, nous sommes ramenés à une équation autonome du type (3.1) sur $V \times \mathbb{Z}$. Notons néanmoins que l'ensemble d'états $V \times \mathbb{Z}$ est dans ce cas forcément infini (même si l'ensemble V était au départ fini). ■

3.1.2 Vocabulaire de base de la théorie des systèmes dynamiques discrets

Considérons à présent un système autonome $S = (V, F)$. Afin de visualiser la dynamique d'un tel système, nous pouvons donner la table de la fonction F (donnant $F(v)$ pour tout $v \in V$) ou encore son graphe de transition (analogue du *portrait de phase* continu), dont voici la définition :

Définition 3.3 (Graphe de transition) Soit un système dynamique discret $S = (V, F)$. On appelle graphe de transition (ou graphe d'itération) de S le graphe $\mathcal{GT} = (V, E)$ dans lequel, étant donnés u et v deux éléments de V , l'arête reliant u à v , notée (u, v) ou encore $(u \rightarrow v)$ appartient à E si et seulement si $v = F(u)$.

Afin de mieux comprendre cette définition, considérons l'exemple suivant.

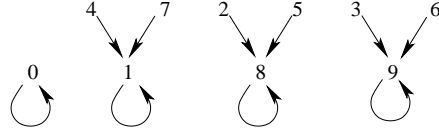
Exemple 3.1

Soit $V = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Etant donné $v \in V$, on obtient $F(v)$ en élevant v au cube puis en additionnant les chiffres du nombre obtenu, éventuellement plusieurs fois, jusqu'à trouver un résultat dans V .

La table de F est la suivante :

v	0	1	2	3	4	5	6	7	8	9
$F(v)$	0	1	8	9	1	8	9	1	8	9

Afin de représenter visuellement la dynamique de ce réseau, on représente son graphe de transition par le schéma suivant :



On observe 4 configurations stables, c'est-à-dire des configurations dans lesquelles le système n'évolue plus au cours du temps. Il s'agit des points fixes de la fonction F (que l'on retrouve dans la table de F).

L'itération (3.2) définie plus haut donne une suite $(x^t) \in V^{\mathbb{N}}$. Nous avons nommé une telle suite orbite positive du système S . Il est connu (voir par exemple [62]) que si V est fini, une telle itération finit toujours par atteindre soit un point fixe de F , soit un cycle, et ce, quelle que soit la condition initiale choisie. Ainsi dans l'exemple précédent, le système présente 4 bassins d'attraction, chacun contenant un unique point fixe de F . Afin de formaliser ce résultat, nous rappelons ici quelques définitions utiles.

Définition 3.4 Soit u, v deux éléments de V . On dira que v est un descendant de u (noté $u \triangleright_F v$) s'il existe $p \in \mathbb{N}$ tel que $v = F^p(u)$.

Il est immédiat que la relation \triangleright_F est un préordre. On peut construire alors la relation d'équivalence suivante :

Définition 3.5 Soit u, v deux éléments de V . u et v seront dits équivalents (noté $u \sim_F v$) s'ils ont un descendant commun, c'est-à-dire si :

$$\exists w \in V, \quad (u \triangleright_F w) \wedge (v \triangleright_F w)$$

Cette relation d'équivalence nous permet de définir formellement la notion déjà évoquée plus haut de bassin d'attraction.

Définition 3.6 Soit un système $S = (V, F)$ et soit \sim_F la relation d'équivalence associée. On nommera bassins d'attraction de S les éléments de l'espace quotient V / \sim_F .

Les bassins d'attraction de S sont évidemment les composantes connexes du graphe d'itération \mathcal{GT} . La proposition suivante nous permet d'exprimer de façon mathématique le type de comportements dynamiques que peuvent exhiber les itérations sur un ensemble fini.

Proposition 3.1 Soit un système dynamique discret $S = (V, F)$, on suppose V fini. Alors S comporte un nombre fini de bassins d'attraction. Chacun de ces bassins contient un unique attracteur, qui est :

- soit un point fixe de F (appelé point d'équilibre), c'est-à-dire un élément ξ de V tel que $F(\xi) = \xi$,
- soit un cycle, c'est-à-dire une partie $\mathcal{C} = \{\xi_1, \dots, \xi_p\}$ de V vérifiant :

$$\begin{cases} \forall i = 1 \dots p-1, & F(\xi_i) = \xi_{i+1} \\ F(\xi_p) = \xi_1 \end{cases}$$

Ce résultat est fondamental pour l'étude des systèmes dynamiques discrets puisqu'il nous permet d'affirmer que même si l'espace des états V est important (si l'on prend comme exemple l'ensemble des mots binaires de taille n , la cardinalité de V est 2^n qui atteint rapidement, lorsque n augmente, des valeurs très grandes), l'itération discrète (3.2) finira toujours par atteindre un attracteur simple, cycle ou point fixe. Toutefois, ce résultat cache derrière son énoncé deux réelles difficultés de ces systèmes. Tout d'abord, étant donné un système de grande taille, l'étude exhaustive du portrait de phase (i.e. du graphe d'itération) est limitée par la puissance de calcul de l'ordinateur sur lequel nous faisons nos simulations. Il nous faut donc, dans la mesure du possible, trouver des outils mathématiques qui nous permettent d'étudier plus précisément de tels réseaux et éventuellement de prévoir certaines propriétés de leur dynamique. Par ailleurs, l'intérêt d'étudier un réseau particulier est parfois assez limité ; nous sommes donc souvent amenés à rechercher des propriétés génériques concernant certaines classes de ces réseaux. Nous verrons dans la suite un certain nombre d'outils nous permettant d'appréhender la dynamique de tels systèmes. Ce préambule établi, nous nous intéressons à présent à la notion de réseau d'automates sur un graphe. Cette notion est fondamentale car elle est à la base de nombre de modèles discrets (réseaux neuronaux, réseaux de Pétri, machines de Turing, etc.).

3.1.3 Le modèle Réseau d'Automates sur un Graphe

Avant de définir la notion de réseau d'automates sur un graphe, faisons un bref rappel sur la notion de *degré* d'un graphe.

Considérons un graphe orienté $G = (X, \mathcal{A})$ dont les sommets appartiennent à un ensemble dénombrable (éventuellement infini) X , et dont les arêtes sont les éléments de l'ensemble $\mathcal{A} \subset X \times X$. L'arête reliant le sommet x au sommet y sera noté $a(x, y)$ où $(x \rightarrow y)$. Etant donné un sommet $x \in X$, on définit les degrés entrant et sortant de x comme suit :

- L'ensemble des prédécesseurs de x est l'ensemble

$$\text{Pred}(x) = \{y \in X \mid a(y, x) \in \mathcal{A}\}$$

Le degré entrant de x , noté $d^-(x)$ est le cardinal de l'ensemble $\text{Pred}(x)$.

- L'ensemble des successeurs de x est l'ensemble

$$\text{Succ}(x) = \{y \in X \mid a(x, y) \in \mathcal{A}\}$$

Le degré sortant de x , noté $d^+(x)$ est le cardinal de l'ensemble $\text{Succ}(x)$.

Dans le cas où X est un ensemble infini, on dira que G est de degré entrant (resp. sortant) borné s'il existe $d \in \mathbb{N}^*$ tel que, pour tout $x \in X$, $d^-(x) \leq d$ (resp. $d^+(x) \leq d$). Plus généralement, on dira que G est de degré borné s'il est de degré entrant borné et de degré sortant borné.

Voici une définition d'un réseau d'automates sur un graphe qui formalise la description que nous avons faite dans l'introduction de ce chapitre (voir notamment [48]).

Définition 3.7 *Un réseau d'automates sur un graphe (ou graphe d'automates) est un quadruplet $R = (G, \mathcal{S}, \mathcal{N}, \delta)$ dans lequel :*

- $G = (X, \mathcal{A})$ est un graphe orienté dont les sommets appartiennent à l'ensemble dénombrable X . Ces sommets sont appelés **automates**. On fait l'hypothèse que G est de degré borné (par une constante $K \in \mathbb{N}^*$ donnée).

- $S = \{S_x\}_{x \in X}$ est une collection d'ensembles finis S_x indicés par l'ensemble X des automates. S_x est l'ensemble fini des **états** que peut prendre l'automate x .
- $\mathcal{N} : X \rightarrow (\{1, \dots, K\} \rightarrow X)$ détermine le **voisinage** du réseau, c'est-à-dire, pour tout automate $x \in X$, l'ensemble ordonné de ces prédécesseurs : $\mathcal{N}(x) (\{1, \dots, K\}) = \text{Pred}(x)$.
- δ est la **fonction de transition locale** du réseau : $\delta = (\delta_x)_{x \in X}$, avec, pour $x \in X$ fixé :

$$\delta_x : S_x \times \left(\prod_{i=1}^{d^-(x)} S_{\mathcal{N}(x)(i)} \right) \longrightarrow S_x$$

Remarque 3.3 Faisons quelques remarques sur cette définition.

- Tout d'abord, il faut noter la différence entre $\text{Pred}(x)$ et $\mathcal{N}(x)$. Le premier est un ensemble non ordonné alors que le second est une fonction qui a pour but d'ordonner les prédécesseurs de l'automate x . C'est pourquoi le voisinage \mathcal{N} entre dans la définition précédente : il n'est pas induit par la donnée du graphe G .
- La fonction δ est une fonction de transition locale. Etant donné un automate $x \in X$ fixé, elle a pour but d'aller chercher l'état associé à chaque voisin de x (ainsi que l'état de x lui-même) et de calculer l'état suivant de x .
- Le graphe G est fondamental dans la définition précédente car il définit la structure d'un réseau d'automates. La définition qui suit insiste sur ce point.

Définition 3.8 Etant donné un réseau d'automates $R = (G, S, \mathcal{N}, \delta)$, on nomme le graphe G **graphe d'interaction** de R (ou encore **graphe de connexion**). Il sera noté dans la suite \mathcal{GI} .

La définition 3.7 est très générale et englobe en fait de nombreux modèles existants dans la littérature. Nous citons ici deux d'entre eux, qui nous semblent importants : les réseaux d'automates cellulaires [48, 63, 84, 85] (que nous verrons dans la partie suivante) et les réseaux de neurones de Hopfield [33] dont voici une brève description.

Exemple 3.2 (réseaux de Hopfield)

Le modèle de Hopfield appartient à la catégorie des réseaux de neurones. Il porte également le nom de réseau d'automates à seuil.

Considérons un graphe $G = (X, \mathcal{A})$ fini (i.e. $\text{card}(X) = n \in \mathbb{N}^*$) et entièrement connecté, c'est-à-dire :

$$\forall x, y \in X, \quad (x \rightarrow y) \in \mathcal{A}$$

Chaque automate (on parlera également de neurone dans ce cas) $x \in X$ peut prendre deux valeurs : 1 (on dira que le neurone est activé) et 0 (non activé). L'ensemble d'états est donc le même pour tous les sommets du graphe :

$$\forall x \in X, \quad S_x = S = \{0, 1\}$$

Numérotions les neurones : $X = \{x_1, \dots, x_n\}$. La fonction de voisinage est donnée par :

$$\forall i, j \in \{1, \dots, n\}, \quad \mathcal{N}(x_i)(j) = x_j$$

Afin de définir la fonction de transition locale, nous considérons une matrice W carrée réelle de dimension n (dite matrice de poids) et un vecteur $\Theta \in \mathbb{R}^n$ (dit vecteur de seuils). On définit

la fonction de transition locale comme le seuillage d'une somme pondérée des entrées :

$$\forall i \in \{1, \dots, n\}, \quad \delta_{x_i}(\text{état}(x_1), \dots, \text{état}(x_n)) = H \left(\sum_{j=1}^n w_{ij} \text{état}(x_j) - \theta_i \right)$$

où $\text{état}(x_k) \in S$ désigne l'état du neurone x_k et H est la fonction de Heaviside :

$$H : \mathbb{R} \longrightarrow S \\ x \longmapsto \begin{cases} 0 & \text{si } x \leq 0 \\ 1 & \text{si } x > 0 \end{cases}$$

Le neurone x_i sera donc activé si la somme pondérée de ses entrées (c'est-à-dire de tous les neurones du réseau) est suffisamment forte pour dépasser le seuil θ_i . Dans le cas contraire, il sera désactivé.

Dans la partie qui suit, nous nous intéressons à une autre classe de graphes d'automates largement étudiée dans la littérature qui sont les automates cellulaires.

3.1.4 Exemple de graphe d'automates : les réseaux d'automates cellulaires

Définition

Nous allons donner ici la définition d'un réseau d'automates cellulaires (voir [63]) en reprenant par étape la définition 3.7. Nous considérons un graphe $G = (X, \mathcal{A})$ infini de degré borné. Le plus souvent, l'ensemble X est la grille \mathbb{Z}^d , avec $d \in \mathbb{N}^*$ (voir figure 3.1). Nous nous plaçons dans ce cas désormais. L'ensemble des connexions \mathcal{A} est donné par exemple par l'expression suivante :

$$\forall x, y \in \mathbb{Z}^d, \quad (x \rightarrow y) \in \mathcal{A} \iff \|x - y\| = 1 \quad (3.4)$$

Nous remarquons grâce à cette définition que le graphe G est *symétrique*, c'est-à-dire que :

$$\forall x, y \in \mathbb{Z}^d, \quad (x \rightarrow y) \in \mathcal{A} \iff (y \rightarrow x) \in \mathcal{A}$$

Deux éléments x et y de \mathbb{Z}^d tels que $(x \rightarrow y) \in \mathcal{A}$ seront appelés *premiers voisins* ou simplement *voisins*. Il y a plusieurs manières de définir cet ensemble de connexions selon la norme $\|\cdot\|$ que

l'on choisit. Si l'on choisit la norme $\|\cdot\|_1$ (définie par $\|x\|_1 = \sum_{i=1}^d |x_i|$), chaque cellule a $2d$ voisins (c'est le cas dans la figure 3.1). Si l'on choisit la norme $\|\cdot\|_\infty$ (définie par $\|x\|_\infty = \max_{i=1 \dots d} |x_i|$), alors chaque cellule a $3^d - 1$ voisins (sur la figure 3.1, il faut rajouter les liaisons diagonales : Nord-Est, Sud-Est, etc.).

Le principe de base des automates cellulaires est l'uniformité du réseau, dans le sens que chaque cellule sur la grille a le même ensemble d'états et la même règle de transition. De plus, l'ensemble des voisins d'un automate se calcule de la même façon pour tous les automates. Grâce à ces hypothèses, nous pouvons formuler la définition suivante :

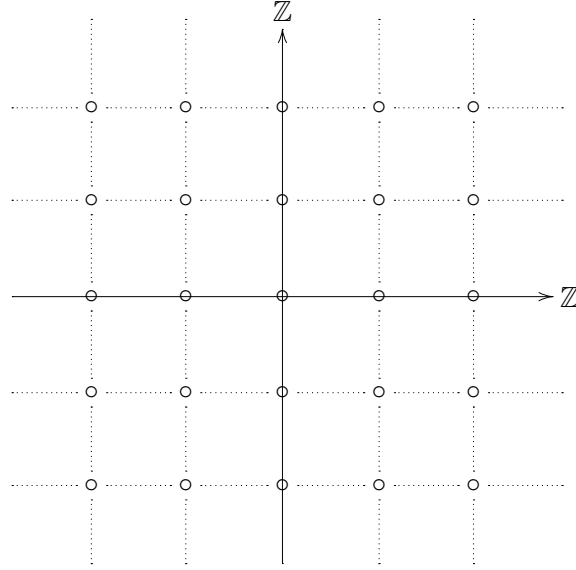


FIG. 3.1: Graphe d'automates cellulaires sur \mathbb{Z}^2 . L'ensemble \mathcal{A} des connexions, représenté par les traits pointillés, est donné par l'expression (3.4) calculée à partir de la norme $\|\cdot\|_1$.

Définition 3.9 Un réseau d'automates cellulaires R sur la grille \mathbb{Z}^d est donné par le quadruplet $(d, S, \mathcal{N}, \delta)$ où :

- $d \in \mathbb{N}^*$ est la dimension de la grille.
- S est l'ensemble (fini) des états.
- Le voisinage \mathcal{N} se réduit à un sous-ensemble $\{n_1, \dots, n_p\}$ **ordonné** et fini de \mathbb{Z}^d .
- δ est la fonction de transition locale du réseau : $\delta : (S)^{K+1} \longrightarrow S$

On appelle configuration du réseau R toute fonction $c : \mathbb{Z}^d \longrightarrow S$. L'ensemble des configurations de R est $C_R = S^{\mathbb{Z}^d}$.

Une fois associé à chaque automate $x \in \mathbb{Z}^d$ un état $s \in S$ (i.e. une fois choisie une *configuration initiale* c), chaque automate calcule simultanément son état suivant s' grâce à la formule :

$$s' = \delta(c(x), c(x + n_1), \dots, c(x + n_p))$$

La configuration suivante c' de c est donc :

$$\forall x \in \mathbb{Z}^d, \quad c'(x) = \delta(c(x), c(x + n_1), \dots, c(x + n_p))$$

Remarque 3.4 Dans le cas des automates cellulaires, nous avons fait l'hypothèse implicite que tous les automates sont mis à jour simultanément. En toute généralité, ceci n'est pas le cas pour un réseau d'automates quelconque. Nous reviendrons sur ce point dans la suite en abordant la dynamique des réseaux booléens.

Automates cellulaires et équations aux dérivées partielles

Les automates cellulaires ont de nombreuses applications dans des domaines variés, que ce soit en mathématiques ou en informatique. Ainsi, la résolution numérique d'une équation aux

dérivées partielles par une discrétisation de l'espace et du temps suivie d'un schéma itératif peut en fait se voir comme la construction et l'itération d'un réseau d'automates cellulaires associé à cette équation.

Pour fixer les idées, considérons une fonction $u = u(t, x, y)$ à valeurs dans \mathbb{R} qui vérifie une équation aux dérivées partielles d'évolution :

$$\begin{cases} \frac{\partial u}{\partial t} + L(u) = f \\ u(0, x, y) = u_0(x, y) \end{cases} \quad (3.5)$$

où L est un opérateur différentiel non linéaire en u , $f = f(t, x, y)$ et $u_0(x, y)$ sont des fonctions données.

Nous considérons à présent une discrétisation de l'EDP (3.5). Pour cela, nous choisissons trois constantes réelles positives Δx , Δy et Δt (pas en espace et en temps) et nous posons :

$$\begin{cases} \forall (i, j) \in \mathbb{Z}^2, & x_i = i\Delta x \text{ et } y_j = j\Delta y \\ \forall n \in \mathbb{N}, & t_n = n\Delta t \end{cases} \quad (3.6)$$

Le but de la méthode est de calculer par un algorithme itératif une approximation de la solution u sur la grille (x_i, y_j) . Nous étudions donc une suite à trois indices : $\left(u_{i,j}^n\right)_{(i,j) \in \mathbb{Z}^2}^{n \in \mathbb{N}}$ telle que $u_{i,j}^n$ "approche" $u(x_i, y_j, t_n)$. A priori, $u_{i,j}^n$ est un réel, nous reviendrons sur ce point précis dans la suite.

Un schéma aux différences finies explicite de l'équation (3.5) selon la discrétisation (3.6) consiste donc à se donner un opérateur H_Δ discret et à résoudre l'équation récurrente suivante :

$$\forall n \in \mathbb{N}, \forall (i, j) \in \mathbb{Z}^2, u_{i,j}^{n+1} = H_\Delta \left(u_{i-k,j-k}^n, u_{i-k+1,j-k}^n, \dots, u_{i+k,j-k}^n, u_{i-k,j-k+1}^n, \dots, u_{i+k,j+k}^n \right)$$

à laquelle on ajoute la condition initiale : $u_{i,j}^0 = u_0(x_i, y_j)$, pour tout $(i, j) \in \mathbb{Z}^2$. H_Δ est une fonction de $(2k+1)^2$ variables, on dira alors que le schéma est à $(2k+1)^2$ points.

La résolution numérique d'une telle équation dépend bien sûr du choix de l'opérateur H_Δ . Une fois cet opérateur judicieusement choisi (voir exemple 3.3 ci-dessous), il faut ensuite montrer la *consistance* d'un tel schéma avec l'EDP, puis sa *convergence* ainsi que sa *stabilité*. Notre objet n'est pas de traiter ici ces questions, relatives à la résolution numérique d'équations aux dérivées partielles. Nous nous contenterons de proposer l'étude d'un exemple particulier afin de préciser les notions que nous venons d'aborder.

Exemple 3.3

Nous considérons une équation de diffusion en deux dimensions sur \mathbb{R}^2 (équation de la chaleur) :

$$\begin{cases} \frac{\partial u}{\partial t} - a\Delta u = 0 \\ u(0, x, y) = u_0(x, y) \end{cases} \quad (3.7)$$

avec a un réel positif. Nous supposons que les pas Δx et Δy sont égaux (grille uniforme). La discrétisation de (3.7) nécessite la discrétisation de trois termes. Nous utilisons pour la dérivée temporelle une approximation d'Euler classique :

$$\frac{\partial u}{\partial t}(x_i, y_j, t_n) \simeq \frac{u(x_i, y_j, t_{n+1}) - u(x_i, y_j, t_n)}{\Delta t}$$

Pour les dérivées spatiales, nous choisissons un schéma centré :

$$\begin{aligned}\frac{\partial^2 u}{\partial x^2}(x_i, y_j, t_n) &\simeq \frac{u(x_{i-1}, y_j, t_n) - 2u(x_i, y_j, t_n) + u(x_{i+1}, y_j, t_n)}{\Delta x^2} \\ \frac{\partial^2 u}{\partial y^2}(x_i, y_j, t_n) &\simeq \frac{u(x_i, y_{j-1}, t_n) - 2u(x_i, y_j, t_n) + u(x_i, y_{j+1}, t_n)}{\Delta x^2}\end{aligned}$$

En injectant ces approximations dans l'équation (3.7), nous obtenons l'équation suivante :

$$\frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} - \frac{a}{\Delta x^2} (u_{i-1,j}^n - 2u_{i,j}^n + u_{i+1,j}^n + u_{i,j-1}^n - 2u_{i,j}^n + u_{i,j+1}^n) = 0$$

d'où, en posant $\lambda = \frac{a\Delta t}{\Delta x^2}$:

$$u_{i,j}^{n+1} = \lambda (u_{i-1,j}^n + u_{i,j-1}^n) + (1 - 4\lambda)u_{i,j}^n + \lambda (u_{i+1,j}^n + u_{i,j+1}^n)$$

Le schéma choisi est donc un schéma à 5 points :

$$H_{\Delta}(u_{-1,0}, u_{0,-1}, u_{0,0}, u_{0,1}, u_{1,0}) = \lambda (u_{-1,0} + u_{0,-1}) + (1 - 4\lambda)u_{0,0} + \lambda (u_{1,0} + u_{0,1})$$

Enfin, la condition initiale est donnée par : $u_{i,j}^0 = u_0(x_i, y_j)$, i et j appartenant à \mathbb{Z} .

Sur cet exemple, nous allons voir qu'à ce schéma aux différences finies explicite de l'équation d'évolution (3.7), nous pouvons associer un réseau d'automates cellulaires défini plus haut (définition 3.9). Pour cela, il nous suffit d'adopter la méthode suivante :

1. La dimension du réseau est égale au nombre de variables spatiales dont dépend l'inconnue u . Dans notre exemple, on construit donc notre réseau sur la grille \mathbb{Z}^2 .
2. Le voisinage dépend des schémas de discrétisation des dérivées que l'on choisit. Le principe général est que, pour calculer une approximation de la dérivée d'une fonction en un point x , on n'a besoin que de la valeur de cette fonction aux points de la grille voisins de x . Dans notre exemple, nous avons choisi un schéma à 5 points, le voisinage peut s'exprimer ainsi : $\mathcal{N} = ((-1, 0), (0, -1), (0, 0), (0, 1), (1, 0))$ (cet ensemble est ordonné selon l'ordre des variables dans l'opérateur H_{Δ} défini plus haut).
3. La fonction de transition δ est égale à H_{Δ} .

Dans notre description, nous avons volontairement omis de préciser l'ensemble des états des automates. Comme nous l'évoquons plus haut, les valeurs $u_{i,j}^n$ sont a priori réelles, ce qui pose un problème car l'ensemble des états d'un automate cellulaire doit être fini. Or, lorsque nous sommes amenés à implémenter un tel schéma sur ordinateur, nous nous apercevons que les valeurs des $u_{i,j}^n$ sont des *nombres flottants*, qui sont en nombre fini (ce nombre dépend bien entendu de la façon de coder ces flottants en machine). On peut donc considérer que l'ensemble des états des automates est en réalité fini puisqu'il est égal à l'ensemble des nombres flottants.

La résolution numérique sur ordinateur d'une équation aux dérivées partielles d'évolution par différences finies peut donc être vue comme l'implémentation d'un réseau d'automates cellulaires sur une grille \mathbb{Z}^d .

Remarque 3.5 Nous avons considéré la solution u de l'équation (3.5) comme étant définie pour $(x, y) \in \mathbb{R}^2$. Le plus souvent, un problème de modélisation par des EDP consiste à trouver des solutions sur un domaine Ω ouvert et borné de \mathbb{R}^2 . L'équation (3.5) sera donc munie de

conditions aux limites sur la frontière $\partial\Omega$. Il n'est alors plus possible de définir notre réseau d'automates cellulaires sur la grille \mathbb{Z}^2 , mais sur une partie de cette grille. Il nous faut dans ce cas modifier notre définition d'un réseau d'automates cellulaires, en le considérant sur un graphe fini.

Dans cette partie, nous avons défini la notion de réseau d'automates sur un graphe ou graphe d'automates. Nous avons vu que cette définition est assez générale puisqu'elle englobe plusieurs types de réseaux discrets, notamment les réseaux neuronaux de Hopfield ou encore les automates cellulaires. Nous allons à présent définir les réseaux d'automates booléens, qui forment une sous-classe de réseaux largement utilisée dans la modélisation de réseaux issus de la biologie.

3.2 Les réseaux d'automates booléens

Nous trouvons dans la littérature de nombreuses tentatives de modélisation de réseaux de régulation génétique par des réseaux booléens. Nous citerons notamment S. Kauffman (voir e.g. [39]) et R. Thomas (voir [75]). Nous proposons dans cette partie une définition d'un réseau d'automates booléens comme étant un cas particulier d'un réseau d'automates sur un graphe (cf. définition 3.7).

3.2.1 Structure d'un réseau booléen

Dans un réseau d'automates booléens, les automates sont généralement numérotés de 1 à n . L'ensemble des automates est donc l'ensemble fini $X = \{1, \dots, n\}$. Chacun d'entre eux peut prendre la valeur 0 (on dira que l'automate est *inactif* ou *éteint*) ou la valeur 1 (l'automate est alors *actif* ou *allumé*). L'ensemble S des états est donc le même pour tous les éléments $i \in X$ du réseau : $S = \{0, 1\}$. Soit i un élément du réseau, nous associons à l'automate $i \in X$ une *variable booléenne* x_i qui représente son état courant. Si t désigne un temps discret (i.e. $t \in \mathbb{N}$), on notera x_i^t ou $x_i(t)$ l'état de l'élément i au temps t .

La structure d'un réseau booléen est donnée par le graphe d'interaction $\mathcal{GI} = (X, \mathcal{A})$. \mathcal{A} est l'ensemble des arêtes ou connexions du réseau : $\mathcal{A} \subseteq X \times X$. Un élément de \mathcal{A} sera noté (i, j) ou encore $(i \rightarrow j)$, où i et j sont deux éléments de X . Si $(i \rightarrow j) \in \mathcal{A}$, on dira que l'élément i *influence* l'élément j . Il est souvent d'usage de préférer une notation matricielle pour définir les connexions du réseau :

Définition 3.10 (Matrice d'incidence) Soit un graphe d'interaction $\mathcal{GI} = (X, \mathcal{A})$ et soit $n = \text{card}(X) \in \mathbb{N}^*$. On définit la matrice d'incidence de \mathcal{GI} comme étant la matrice booléenne B de dimension $n \times n$ telle que :

$$\forall i, j \in \{1, \dots, n\}, \quad b_{ij} = \begin{cases} 1 & \text{si } j \text{ influence } i, \text{ i.e. si } (j \rightarrow i) \in \mathcal{A} \\ 0 & \text{sinon} \end{cases}$$

Il nous faut définir maintenant la notion de voisinage. Pour chaque automate $i \in X$, on note E_i l'ensemble des *entrées* de i , c'est-à-dire l'ensemble des automates du réseau qui l'influencent :

$$E_i = \{j \in X \mid (j \rightarrow i) \in \mathcal{A}\}$$

La cardinalité de l'ensemble E_i , que nous noterons k_i est appelée *connectivité* de l'élément i . Elle se calcule par la formule :

$$k_i = \sum_{j=1}^n b_{ij}$$

Nous reviendrons dans le chapitre suivant sur cette grandeur qui s'avèrera importante dans notre étude. Le vecteur des connectivités $k = (k_i)_{i=1\dots n}$ se retrouve aisément en additionnant les colonnes de B (addition dans \mathbb{N}). On notera K la connectivité maximale du réseau :

$$K = \max\{k_i \mid i \in X\}$$

Se donner un *voisinage* d'un élément $i \in X$ du réseau consiste à se donner un ordre sur E_i . Le voisinage de l'élément i , noté $\mathcal{N}(i)$, est donc une bijection de $\{1, \dots, k_i\}$ dans E_i . Nous choisissons ici l'ordre "naturel" des entiers :

$$\begin{aligned} \mathcal{N}(i)(1) &= \min E_i \\ \mathcal{N}(i)(2) &= \min(E_i \setminus \{\mathcal{N}(i)(1)\}) \\ \mathcal{N}(i)(3) &= \min(E_i \setminus \{\mathcal{N}(i)(1), \mathcal{N}(i)(2)\}) \\ &\vdots \\ \mathcal{N}(i)(k_i) &= \max E_i \end{aligned} \tag{3.8}$$

Nous supposons dans la suite que l'ordre des entrées utilise toujours cette convention.

La dernière étape de la construction du réseau consiste à se donner une fonction de transition globale. Pour cela, nous passons par des fonctions de transitions locales. Etant donné un élément $i \in X$ du réseau, on appelle fonction de transition locale de l'élément i toute fonction scalaire booléenne f_i dépendant de k_i variables :

$$f_i : \{0, 1\}^{k_i} \longrightarrow \{0, 1\}$$

La **mise à jour** de l'élément i peut donc s'exprimer par l'équation :

$$x_i^{t+1} = f_i(x_{\mathcal{N}(i)(1)}^t, \dots, x_{\mathcal{N}(i)(k_i)}^t)$$

pour tout $t \in \mathbb{N}$.

Pour simplifier cette expression, on pose $i_1 = \mathcal{N}(i)(1)$, \dots , $i_{k_i} = \mathcal{N}(i)(k_i)$. On considère ensuite \tilde{f}_i le prolongement de f_i sur $\{0, 1\}^n$: $\tilde{f}_i(x_1, \dots, x_n) = f_i(x_{i_1}, \dots, x_{i_{k_i}})$. S'il n'y a pas d'ambiguïté, on notera également f_i la fonction \tilde{f}_i . Nous pouvons exprimer alors la **fonction de transition globale** du réseau :

$$\begin{aligned} F : \quad \{0, 1\}^n &\longrightarrow \{0, 1\}^n \\ (x_1, \dots, x_n) &\longmapsto \begin{pmatrix} f_1(x_1, \dots, x_n) \\ \vdots \\ f_n(x_1, \dots, x_n) \end{pmatrix} \end{aligned}$$

Exemple 3.4

Soit R un réseau de dimension 4. Supposons que les entrées de l'automate 1 sont les automates 2 et 4 (i.e. $E_1 = \{2, 4\}$ et $k_1 = 2$), et que la fonction d'activation locale f_1 est :

$$\begin{aligned} f_1 : \quad \{0, 1\}^2 &\longrightarrow \{0, 1\} \\ (x_2, x_4) &\longmapsto x_2 \wedge x_4 \end{aligned}$$

On note \tilde{f}_1 le prolongement de f_1 sur $\{0, 1\}^4$:

$$\begin{aligned} f_1 : \quad \{0, 1\}^4 &\longrightarrow \{0, 1\} \\ (x_1, x_2, x_3, x_4) &\longmapsto x_2 \wedge x_4 \end{aligned}$$

On s'autorise alors l'abus de notation suivant : $\tilde{f}_1(x_1, x_2, x_3, x_4) = f_1(x_1, x_2, x_3, x_4) = f_1(x_2, x_4)$

Voici à présent une définition d'un réseau d'automates booléens. Cette définition se base sur la définition 3.7 ainsi que sur les remarques précédentes.

Définition 3.11 (Réseau d'automate booléen) *Un réseau d'automates booléens de dimension n est un triplet $\mathcal{R} = (X, B, F)$ où $X = \{1, \dots, n\}$, B est une matrice booléenne de dimension $n \times n$ et F une application de $\{0, 1\}^n$ dans lui-même.*

Remarque 3.6 *Faisons quelques remarques terminologiques relatives à cette définition :*

- La donnée du graphe d'interaction \mathcal{GI} est ici remplacée par la donnée de la matrice d'incidence B . Il s'agit simplement d'une question d'usage, on peut facilement vérifier que la donnée de ces deux objets est mathématiquement équivalente. Nous utiliserons donc par la suite indifféremment les notions de graphe d'interaction ou de matrice d'incidence.
- Les ensembles d'états $(S_i)_{i \in X}$ sont ici tous égaux à

$$S = \{0, 1\}$$

La fonction de voisinage \mathcal{N} est la fonction définie par le système (3.8).

- Pour chaque automate $i \in X$, l'ensemble $E_i = \{j \in X \mid b_{ij} = 1\}$ se nomme ensemble d'entrées de i . Il est égal à l'ensemble des prédécesseurs de i .
- La quantité $k_i = \sum_{j=1}^n b_{ij}$ est appelée connectivité apparente du nœud i . Nous reviendrons sur la notion fondamentale de connectivité dans l'exemple ci-dessous ainsi que dans le chapitre suivant.
- La fonction vectorielle $F : \{0, 1\}^n \rightarrow \{0, 1\}^n$ est la fonction de transition globale du réseau \mathcal{R} . Ses composantes f_i (i allant de 1 à n) sont les fonctions de transitions locales du réseau. Nous rappelons ici que chaque f_i est en fait définie sur $\{0, 1\}^{k_i}$ mais on notera également f_i le prolongement de cette fonction à $\{0, 1\}^n$ (cf. exemple 3.4).

Exemple 3.5

Considérons le réseau de dimension 4 défini par le graphe représenté par la figure suivante :

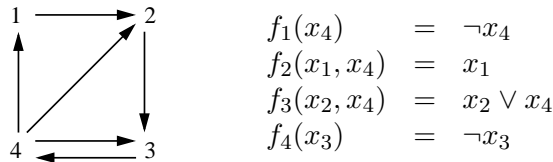


FIG. 3.2: Graphe d'interaction et fonctions de transitions locales associées à un réseau de dimension 4.

La donnée de ce graphe et de la fonction F définit entièrement le réseau. La matrice d'incidence peut être aisément retrouvée en examinant le graphe de la figure 3.2 :

$$B(F) = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

En examinant de plus près la fonction $f_2(x_1, x_4)$, nous voyons qu'en fait son expression ne dépend pas de la variable x_4 . La flèche reliant le gène 4 au gène 2 peut donc être enlevée du graphe, étant donné qu'elle n'a aucune influence réelle sur l'évolution du gène 2. La connectivité apparente du gène 2 est égale à 2, mais sa connectivité réelle semble en fait être égale à 1. Cet exemple nous montre que la définition 3.11 ne définit pas un réseau de façon unique. En effet, une fois la matrice d'incidence (ou de façon équivalente le graphe d'interaction) choisie, le choix des fonctions booléennes f_i peut faire diminuer la connectivité d'un ou plusieurs éléments du réseau. Une définition précise de la notion de connectivité réelle est donnée dans le chapitre suivant.

L'objet réseau d'automates booléens entre donc dans la définition 3.7 de réseaux d'automates sur un graphe. Dans ces définitions, le temps n'intervient pas : ce sont des définitions statiques. L'objet de la partie qui suit est de définir la dynamique, ou, comme nous allons le voir, les dynamiques possibles de tels réseaux. Nous avons déjà entraperçu, avec les automates cellulaires, que ces dynamiques s'expriment sous la forme d'itérations sur des ensembles dénombrables, dont nous avons rappelé les propriétés fondamentales au début de ce chapitre.

3.2.2 Dynamique d'un réseau booléen

Afin de donner une définition de la dynamique d'un réseau d'automates booléens, nous allons considérer un temps discret $t \in \mathbb{N}$. Il existe en fait plusieurs façons de faire évoluer un réseau au cours du temps. Nous pouvons décider de faire évoluer chaque cellule simultanément (c'est le cas des automates cellulaires), ou encore l'une après l'autre selon un ordre prédéfini. Nous pouvons considérer enfin un mode plus complexe, où l'ordre selon lequel évoluent les automates n'est pas fixe mais dépend par exemple du temps t ou de "l'histoire" du réseau, c'est-à-dire des précédentes mises à jour. Pour tout instant t , savoir quelle cellule (ou groupe de cellules) calculera son état au temps $t + 1$ nécessite une stratégie d'action. Une telle stratégie porte le nom de *mode opératoire* (cf. [62]). Une fois cette stratégie établie, la dynamique d'un réseau d'automates peut être vue comme l'itération d'une application sur un ensemble dénombrable.

Dans la suite, nous considérons un réseau booléen $\mathcal{R} = (X, B, F)$, où $X = \{1, \dots, n\}$, $B \in \mathfrak{M}_n(\{0, 1\})$ et $F : \{0, 1\}^n \rightarrow \{0, 1\}^n$. Nous établissons tout d'abord la définition suivante :

Définition 3.12 On appelle configuration du réseau \mathcal{R} toute application $x : \{1, \dots, n\} \rightarrow \{0, 1\}$ qui à chaque automate $i \in \{1, \dots, n\}$ associe son état $x(i) \in \{0, 1\}$ (noté également x_i). L'ensemble des configurations de \mathcal{R} est égal à $\{0, 1\}^n$ et sera noté $\mathcal{C}_{\mathcal{R}}$

Nous définissons alors la notion de mode opératoire :

Définition 3.13 On appelle mode opératoire toute fonction

$$\Psi : \mathbb{N} \longrightarrow \mathcal{P}(\{1, \dots, n\})$$

qui à tout instant $t \in \mathbb{N}$ associe une partie de l'ensemble des automates $\{1, \dots, n\}$.

Etant donné $t \in \mathbb{N}$, l'ensemble $\Psi(t)$ désigne l'ensemble des automates à mettre à jour à l'instant t . Afin de représenter le fonctionnement dynamique du réseau \mathcal{R} relativement à un mode opératoire Ψ , nous utilisons la démarche suivante :

Soit un instant discret $t \in \mathbb{N}$. Nous supposons que le réseau \mathcal{R} est dans la configuration $x(t) = x \in \mathcal{C}_{\mathcal{R}}$. Nous appelons x' la configuration au temps $t + 1$: $x' = x(t + 1)$. Considérons à présent un automate $i \in \{1, \dots, n\}$. Deux cas se présentent :

- L'automate i ne doit pas être mis à jour au temps t (c'est-à-dire que $i \notin \Psi(t)$), alors :

$$x'(i) = x(i)$$

- L'automate i doit être mis à jour au temps t (i.e. $i \in \Psi(t)$). L'état $x'(i)$ se calcule alors par la formule :

$$x'(i) = f_i(x(i_1), \dots, x(i_{k_i})) \quad (3.9)$$

où (i_1, \dots, i_{k_i}) désigne l'ensemble ordonné des entrées de i .

Quels que soient l'instant t et la configuration initiale x choisis, nous avons défini la configuration suivante x' . L'application qui associe à x la configuration x' dépend uniquement du réseau \mathcal{R} et du mode opératoire Ψ choisi : on la note $F_{\mathcal{R}, \Psi}$. Nous avons ainsi construit le système dynamique discret relatif au mode Ψ : $(\mathcal{C}_{\mathcal{R}}, F_{\mathcal{R}, \Psi})$. L'espace d'états est l'ensemble des configurations $\mathcal{C}_{\mathcal{R}}$ du réseau, il s'agit donc d'un ensemble fini : $\mathcal{C}_{\mathcal{R}} = \{0, 1\}^n$ (l'ensemble des mots binaires de longueur n). Afin d'exprimer l'équation d'évolution qui donne les trajectoires de ce système dynamique, nous distinguons deux cas, selon que le système est ou non autonome. Dans le cas d'un système autonome, c'est-à-dire si la fonction $F_{\mathcal{R}, \Psi}$ ne dépend pas explicitement du temps, alors l'équation d'évolution est donnée par :

$$(E_{\mathcal{R}, \Psi}) \begin{cases} x^0 \in \mathcal{C}_{\mathcal{R}} \\ x^{t+1} = F_{\mathcal{R}, \Psi}(x^t), \quad \text{pour } t \in \mathbb{N} \end{cases} \quad (3.10)$$

Dans le cas d'un système non autonome, c'est-à-dire si la fonction $F_{\mathcal{R}, \Psi}$ dépend explicitement du temps, alors l'équation d'évolution est donnée par :

$$(E_{\mathcal{R}, \Psi}) \begin{cases} x^0 \in \mathcal{C}_{\mathcal{R}} \\ x^{t+1} = F_{\mathcal{R}, \Psi}^t(x^t), \quad \text{pour } t \in \mathbb{N} \end{cases} \quad (3.11)$$

Etant donné un réseau booléen \mathcal{R} et un mode opératoire Ψ , nous avons donc à notre disposition deux graphes. Le premier graphe est le graphe d'interaction :

$$\mathcal{GI} = (\{1, \dots, n\}, \mathcal{A})$$

qui détermine la structure du réseau, c'est-à-dire les connexions entre les différents automates. Le second graphe est le graphe de transition (voir définition 3.3) :

$$\mathcal{GT} = (\mathcal{C}_{\mathcal{R}}, E_{\Psi})$$

qui détermine la dynamique de \mathcal{R} , c'est-à-dire comment le réseau passe d'une configuration à la suivante.

3.2.3 Les modes opératoires parallèle, série et mixte

Nous définissons ici les trois modes opératoires les plus simples, dans le sens où la fonction Ψ est soit constante, soit périodique. En d'autres termes, la stratégie de mise à jour des automates reste la même au cours du temps.

Nous considérons un réseau d'automate booléen $\mathcal{R} = (B, F)$ de dimension n . B est une matrice carrée booléenne de dimension n , F est un champ de vecteur booléen : $F : \{0, 1\}^n \rightarrow \{0, 1\}^n$.

Mode parallèle

Le premier mode que nous allons voir est le mode parallèle, dans lequel à chaque instant discret t , toutes les cellules évoluent simultanément. C'est le mode utilisé pour les automates cellulaires présentés dans la partie précédente. Formellement, le mode parallèle s'exprime ainsi :

$$\forall t \in \mathbb{N}, \quad \Psi_{\parallel}(t) = \{1, \dots, n\}$$

La fonction Ψ_{\parallel} étant constante, le système dynamique relatif au mode parallèle est donc autonome. Plus précisément, on a :

$$F_{\mathcal{R}, \Psi_{\parallel}} = F$$

où F est la fonction de transition globale du réseau \mathcal{R} . Le système dynamique discret associé au mode parallèle est donc le système $(\{0, 1\}^n, F)$. L'équation d'évolution (donnée par la formule (3.10)) est :

$$(E_{\parallel}) \begin{cases} x^0 \in \{0, 1\}^n \\ x^{t+1} = F(x^t), \quad \text{pour } t \in \mathbb{N} \end{cases}$$

Reprenons l'exemple 3.5 que nous avons développé dans la partie précédente.

Exemple 3.6

Nous considérons le réseau défini figure 3.2. L'étude de ce réseau en mode parallèle nous donne un graphe de transition parallèle dont les sommets sont les éléments de $\{0, 1\}^4$. En notant chaque mot binaire de $\{0, 1\}^4$ par son écriture décimale (poids fort en tête), on obtient le graphe suivant :

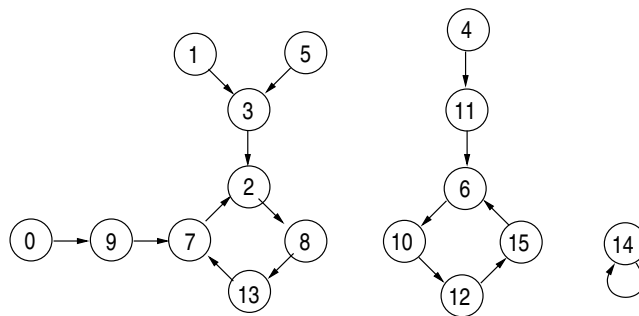


FIG. 3.3: Graphe d'itération parallèle du réseau défini figure 3.2.

Ce graphe contient trois bassins d'attraction. Deux d'entre eux contiennent un cycle de longueur 4 tandis que le troisième se réduit à un point fixe.

Mode série

Dans le mode série, à chaque instant t un seul automate est mis à jour. Chaque cellule évolue donc séparément, selon un ordre fixe prédéfini. Pour définir le mode série, il nous faut donc définir un ordre sur les automates, c'est-à-dire une permutation σ de $\{1, \dots, n\}$. Nous pouvons exprimer le mode série relatif à la permutation σ par une suite périodique de période n :

$$\begin{cases} \forall t \in \{0, \dots, n-1\}, & \Psi_\sigma(t) = \{\sigma(t+1)\} \\ \forall t \geq n, & \Psi_\sigma(t) = \Psi_\sigma(t-n) \end{cases}$$

Afin d'exprimer le système dynamique correspondant à l'exécution série de notre réseau, nous définissons les opérateurs $\phi_i : \{0, 1\}^n \rightarrow \{0, 1\}^n$ par :

$$\forall i \in \{1, \dots, n\}, \quad \phi_i(x) = \begin{pmatrix} x_1 \\ \vdots \\ f_i(x_1, \dots, x_n) \\ \vdots \\ x_n \end{pmatrix}$$

Ces opérateurs nous permettent d'établir la définition suivante (voir [62]) :

Définition 3.14 (Opérateur de Gauss-Seidel) *L'opérateur de Gauss-Seidel associé à la fonction F et à la permutation σ est l'opérateur*

$$G_\sigma = \phi_{\sigma(n)} \circ \dots \circ \phi_{\sigma(1)}$$

Dans le mode série, l'application $F_{\mathcal{R}, \Psi_\sigma}$ est égale à :

$$F_{\mathcal{R}, \Psi_\sigma}^t = \phi_{\sigma(\bar{t}+1)}$$

où \bar{t} désigne le reste dans la division euclidienne de t par n . On observe donc que le système dynamique discret associé au mode série est non autonome :

$$(E_\sigma) \begin{cases} x^0 \in \{0, 1\}^n \\ x^{t+1} = \phi_{\sigma(\bar{t}+1)}(x^t), \quad \text{pour } t \in \mathbb{N} \end{cases}$$

Afin d'éviter cela, nous allons considérer le changement de variable suivant : $t = n\tau$. Si l'on regarde l'évolution de la suite $y^\tau \stackrel{\text{def}}{=} x^t = x^{n\tau}$ (qui est une suite extraite de la suite x^t), on obtient alors :

$$\begin{aligned} y^1 &= x^n \\ &= \phi_{\sigma(n)}(x^{n-1}) \\ &= \phi_{\sigma(n)}(\phi_{\sigma(n-1)}(x^{n-2})) \\ &\vdots \\ &= \phi_{\sigma(n)} \circ \phi_{\sigma(n-1)} \circ \dots \circ \phi_{\sigma(1)}(x^0) \\ &= G_\sigma(x^0) \\ &= G_\sigma(y^0) \end{aligned}$$

Et de manière générale :

$$\forall \tau \in \mathbb{N}^*, \quad y^\tau = G_\sigma(y^{\tau-1})$$

On obtient ainsi une équation d'évolution autonome, dont le champ de vecteurs est l'opérateur de Gauss-Seidel associé à F et à la permutation σ :

$$(E'_\sigma) \begin{cases} y^0 \in \{0, 1\}^n \\ y^{\tau+1} = G_\sigma(y^\tau), \quad \text{pour } \tau \in \mathbb{N} \end{cases}$$

Exemple 3.7

Considérons à nouveau le réseau défini figure 3.2. L'étude de ce réseau en mode série avec $\sigma = (1\ 2\ 3\ 4)$ nous donne le graphe d'itération suivant :

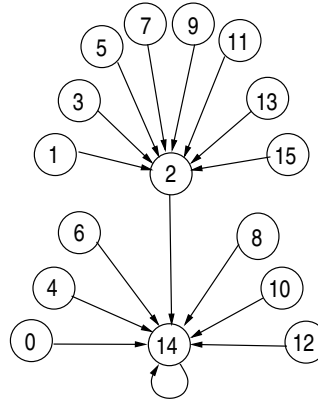


FIG. 3.4: Graphe d'itération série du réseau défini figure 3.2. L'ordre choisi est donné par la permutation $\sigma = (1\ 2\ 3\ 4)$.

Nous remarquons que le point fixe 14 est toujours présent. En revanche, les cycles ont disparu et le graphe ne contient plus qu'un unique bassin d'attraction.

Mode mixte

Le mode mixte (dit aussi série-parallèle) est intermédiaire entre le mode parallèle et le mode série. Dans le mode mixte, un groupe de cellules commence à évoluer simultanément, puis un second groupe et ainsi de suite jusqu'à ce que toutes les cellules du réseau aient évolué une et une seule fois. Pour le définir il faut donc se donner une partition de $\{1, \dots, n\}$, que nous notons P :

$$P = (P_k)_{k=1\dots m}$$

où les P_k sont des parties de $\{1, \dots, n\}$ deux à deux disjointes et $\bigcup_{k=1}^m P_k = \{1, \dots, n\}$.

Le mode mixte relatif à P s'exprime donc par la suite périodique de période m :

$$\begin{cases} \forall t \in \{0, \dots, m-1\}, & \Psi_P(t) = P_{t+1} \\ \forall t \geq m, & \Psi_P(t) = \Psi_P(t-m) \end{cases}$$

Définissons les opérateurs suivants :

Pour $k \in \{1, \dots, m\}$, soit l'opérateur $\phi_{P_k} : \{0, 1\}^n \rightarrow \{0, 1\}^n$ tel que pour $x \in \{0, 1\}^n$, $y = \phi_{P_k}(x)$ est défini par :

$$y_i = \begin{cases} f_i(x) & \text{si } i \in P_k \\ x_i & \text{sinon} \end{cases}$$

Puis, nous définissons l'opérateur G_P comme étant la composition de ces opérateurs :

$$G_P = \phi_{P_m} \circ \cdots \circ \phi_{P_1}$$

En faisant le changement de variable temporelle $t = m\tau$, la relation de récurrence qui nous donne l'équation des trajectoires de notre réseau est autonome (équation 3.10) :

$$(E_P) \begin{cases} x^0 \in X \\ x^{\tau+1} = G_P(x^\tau), \quad \text{pour } \tau \in \mathbb{N} \end{cases}$$

Nous pouvons donc exprimer le système dynamique discret correspondant à notre réseau R sous le mode mixte comme :

$$S_\sigma = (\{0, 1\}^n, G_P)$$

Exemple 3.8

Si nous appliquons au réseau défini figure 3.2 le mode mixte avec $P = (\{1, 3\} \{2\} \{4\})$, nous obtenons le graphe d'itération suivant :

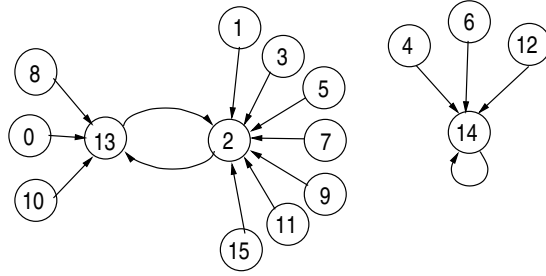


FIG. 3.5: Graphe d'itération mixte du réseau défini figure 3.2. La partition choisie est $P = (\{1, 3\} \{2\} \{4\})$.

Dans ce mode nous trouvons deux bassins d'attraction, dont un qui comporte encore le point fixe 14.

Comme pour le mode série, le seul élément de l'espace des phases qui semble invariant par le choix du mode est l'ensemble des points fixes.

Proposition 3.2 *Etant donné un réseau booléen $\mathcal{R} = (B, F)$ de dimension n , les points fixes de F sont des points d'équilibre pour le système dynamique $(\mathcal{C}_{\mathcal{R}}, F_{\mathcal{R}, \Psi})$, quel que soit le mode Ψ choisi.*

Démonstration

Soit ξ un point fixe de $F : F(\xi) = \xi$ et soit un mode Ψ . Montrons alors que $F_{\mathcal{R}, \Psi}(\xi) = \xi$. Soit $i \in \{1, \dots, n\}$ et $t \in \mathbb{N}$ tel que la configuration au temps t est égal à ξ . On note x' la configuration au temps $t + 1$. Deux cas se présentent : soit $i \notin \Psi(t)$, soit $i \in \Psi(t)$. Dans le premier cas, l'élément i n'est pas mis à jour, donc :

$$x'(i) = x(i) = \xi_i$$

Dans le second cas la mise à jour de l'élément i conduit à :

$$x'(i) = f_i(x(i_1), \dots, x(i_{k_i})) = f_i(\xi_{i_1}, \dots, \xi_{i_{k_i}})$$

Puisque ξ est un point fixe de F , on a :

$$x'(i) = \xi_i$$

D'où $F_{\mathcal{R},\Psi}(\xi) = \xi$. Ce résultat est indépendant du temps t . ■

La réciproque de cette proposition n'est pas vraie en toute généralité. En effet, un point d'équilibre pour un mode Ψ quelconque n'est pas obligatoirement un point fixe de F . Il suffit de considérer un mode dans lequel l'automate 1 (par exemple) n'est jamais mis à jour, i.e. :

$$\forall t \in \mathbb{N}, \quad 1 \notin \Psi(t)$$

Pour un tel mode, un point ξ vérifiant :

$$\begin{cases} f_1(\xi) \neq \xi_1 \\ \forall i \in \{2, \dots, n\}, \quad f_i(\xi) = \xi_i \end{cases}$$

est un point d'équilibre, et pourtant ce n'est pas un point fixe de F . En fait, la réciproque de la proposition précédente est vérifiée seulement si l'on suppose que le mode Ψ vérifie la propriété suivante (inspirée de [62]) :

Définition 3.15 *On dira qu'un mode $\Psi \in \{1, \dots, n\}^{\mathbb{N}}$ est pseudo-périodique si pour tout $i \in \{1, \dots, n\}$, l'ensemble $\{t \in \mathbb{N} \mid i \in \Psi(t)\}$ est infini. En d'autres termes, chaque automate est mis à jour un nombre infini de fois le long d'une trajectoire.*

Les modes série et mixte vérifient cette propriété. Nous pouvons alors énoncer le résultat suivant :

Proposition 3.3 *Soit un réseau booléen $\mathcal{R} = (B, F)$ de dimension n . On note \mathcal{E}_{\parallel} l'ensemble des points fixes de F , \mathcal{E}_{σ} l'ensemble des équilibres de l'itération sous le mode série relatif à la permutation σ et \mathcal{E}_P l'ensemble des équilibres sous le mode mixte relatif à la partition P . On a alors :*

$$\mathcal{E}_{\parallel} = \mathcal{E}_{\sigma} = \mathcal{E}_P$$

3.2.4 Application à la biologie

La modélisation de systèmes biologiques par des systèmes dynamiques discrets a été, et est encore largement présente dans la littérature. Nous citons ici les deux principales approches, menées conjointement depuis les années 70 par S. Kauffman [39–44] (voir également [87] qui propose un outil de visualisation de ces réseaux) et R. Thomas [75–78]. Ces deux approches sont toutes deux basées sur les trois observations suivantes :

- Tout d'abord, les mécanismes de transcription et de traduction qui sont à la base de l'expression d'un gène sont des mécanismes complexes, difficiles à modéliser car impliquant de nombreux processus dont le fonctionnement quantitatif est encore peu connu. Dans l'approche discrète, on ne les considère pas et on s'intéresse directement au niveau d'expression d'un gène.

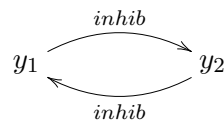
- La démarche de modélisation par système discret demande une démarche de simplification : on suppose donc que les niveaux d'expression d'un gène sont à valeurs dans un ensemble fini d'états. La simplification la plus couramment utilisée est l'idéalisation booléenne, dans laquelle on considère qu'un gène est soit transcrit (son niveau est alors 1), soit non transcrit (son niveau est alors 0).
- Les processus de régulation génétique que nous avons décrit au début de ce mémoire sont alors représentés par des interactions discrètes, plus précisément par un graphe d'interaction et par des fonctions booléennes. Comme nous l'avons vu dans le premier chapitre, les substances qui ont un rôle régulateur ne sont pas toutes des protéines : il peut s'agir de métabolites par exemple. Il nous faut donc considérer également que la concentration de ces substances puisse être approchée par une variable booléenne.

Le cadre de modélisation des réseaux génétiques discrets est donc celui des réseaux d'automates booléens que nous avons décrits dans ce chapitre. Soit un réseau $\mathcal{R} = (X, B, F)$. L'ensemble X représente l'ensemble des gènes ou des autres espèces entrant dans le réseau. On associe à chacun de ces éléments $i \in X$ une variable discrète x_i donnant son état (dans l'idéalisation booléenne, chacune de ces variables peut avoir deux niveaux : 0 ou 1). La matrice $B \in \mathfrak{M}_n(\{0, 1\})$ représente les connexions logiques entre les différentes espèces (i.e. quelle espèce agit sur quelle autre espèce). Enfin, l'application $F : \{0, 1\}^n \rightarrow \{0, 1\}^n$ détermine comment s'expriment ces connexions logiques.

Les deux modèles (de Kauffman et de Thomas) sont donc tous les deux basés sur le même objet mathématique. Leur différence se situe essentiellement dans l'étude de la dynamique de ces systèmes. Dans le modèle de Kauffman, on considère le mode parallèle qui consiste à faire évoluer toutes les variables simultanément à chaque top d'horloge. Cette hypothèse est très forte du point de vue de la modélisation, car elle néglige les différences de vitesse entre les processus en jeu. Cette hypothèse de synchronicité est relaxée dans la démarche de Thomas², conduisant à des résultats certainement plus proches des comportements réels, mais dont l'étude est plus délicate. Voyons sur un exemple simple comment fonctionnent ces deux modèles.

Exemple 3.9

On considère un système impliquant deux variables, notées y_1 et y_2 . On suppose que ces deux variables s'inhibent mutuellement, selon le schéma formel suivant :



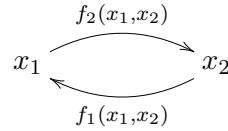
Supposons que ces deux variables sont continues. On peut alors faire l'hypothèse que les termes d'inhibition sont représentés par des fonctions de Hill (voir annexe A). On suppose également que ces deux variables suivent une dégradation linéaire. Nous sommes alors amenés au système différentiel suivant (voir [12]) :

$$\begin{cases} \dot{y}_1 &= \frac{k_2}{\theta_2 + y_2^n} - \nu_1 y_1 \\ \dot{y}_2 &= \frac{k_1}{\theta_1 + y_1^n} - \nu_2 y_2 \end{cases} \quad (3.12)$$

²Pour cette raison, les réseaux de Kauffman sont appelés *synchrones* et ceux de Thomas *asynchrones*.

Nous avons utilisé dans cette description des fonctions de Hill d'ordre n . Comme nous l'avons vu (cf. annexe A), ces fonctions, lorsque n augmente, s'approchent de la fonction de Heaviside (ou fonction step) $H_\theta(y)$.

Considérons à présent une version discrète de ce système. Nous considérons deux variables booléennes x_1 et x_2 connectées de la manière suivante :



avec :

$$\begin{cases} f_1(x_1, x_2) = \neg x_2 \\ f_2(x_1, x_2) = \neg x_1 \end{cases}$$

Ces deux fonctions sont en fait égales aux deux termes de production du système (3.12) dans lequel on a fait tendre n vers l'infini et où l'on a posé :

$$\begin{cases} x_1 = H_{\theta_1}(y_1) \\ x_2 = H_{\theta_2}(y_2) \end{cases}$$

Nous avons ainsi défini un réseau d'automates booléens de dimension 2 (B, F) avec :

$$B = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \text{et} \quad F(x_1, x_2) = \begin{pmatrix} \neg x_2 \\ \neg x_1 \end{pmatrix}$$

Intéressons-nous à présent à la dynamique parallèle de ce système. Le graphe de transition parallèle contient trois bassins d'attraction, deux d'entre eux se réduisent à un point d'équilibre tandis que le troisième se réduit à un cycle de longueur 2 :



FIG. 3.6: Graphe de transition parallèle du système discret \mathcal{R} .

Comparons à présent ce modèle discret et le modèle continu précédent. Selon la valeur des paramètres k_i et ν_i , le système (3.12) a un, deux ou trois points d'équilibre positifs (cf. [12] et figure 3.7 ci-dessous).

Nous allons nous intéresser plus particulièrement au cas où le système admet trois points d'équilibre (voir figure 3.7, graphe de gauche). Dans ce cas, on peut montrer (voir [12]) que le point au milieu est un équilibre instable tandis que les deux autres sont des équilibres stables. Ce système comporte donc deux attracteurs : dans le premier, la variable y_1 atteint une valeur faible $y_1^* < \theta_1$ et la variable y_2 une valeur élevée $y_2^* > \theta_2$ (et vice-et-versa dans le second attracteur). Dans le modèle discret, ces deux attracteurs correspondent aux points fixes $(x_1 = 0, x_2 = 1)$ et $(x_1 = 1, x_2 = 0)$ du graphe de transition parallèle. En revanche, le cycle $\{(0, 0), (1, 1)\}$ n'a aucune correspondance dans le système continu.

Voyons à présent comment se comporte le réseau d'automate \mathcal{R} dans le cas asynchrone. D'après la proposition 3.2, les deux points fixes 01 et 10 restent des points d'équilibre. Dans le cycle :



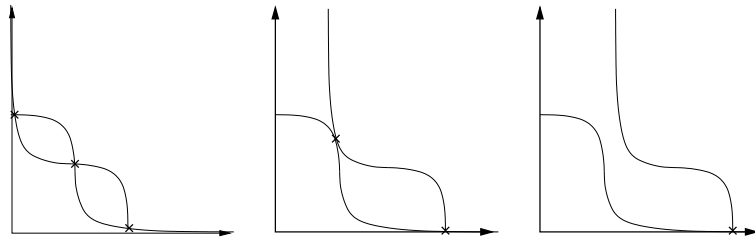
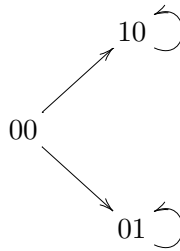
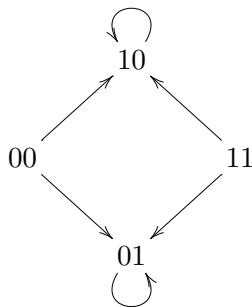


FIG. 3.7: Points d'équilibre du système dynamique continu (3.12). Ces points d'équilibre se trouvent à l'intersection des deux nullclines qui sont d'équations $y_1 = \frac{k_2}{\nu_1(\theta_2 + y_2^n)}$ et $y_2 = \frac{k_1}{\nu_2(\theta_1 + y_1^n)}$. Selon comment sont disposées ces deux courbes sigmoïdales, le système admet donc un (à droite), deux (au milieu) ou trois points d'équilibre (à gauche).

les deux transitions voient les deux automates changer d'état simultanément. Pour fixer les idées, considérons par exemple la transition $00 \rightarrow 11$. Si nous faisons l'hypothèse qu'un seul automate peut changer d'état à un instant t , alors cette transition disparaît au profit des deux transitions :



On obtient alors le graphe de transition asynchrone suivant :



Dans le cas asynchrone, nous n'avons donc plus que deux attracteurs, qui correspondent aux deux attracteurs du système continu (3.12).

Remarque 3.7 Dans le graphe asynchrone précédent, nous n'avons pas spécifié de mode opératoire. En d'autres termes, nous n'avons pas spécifié l'ordre dans lequel sont mis à jour les automates. Dans ce cas précis, nous n'en n'avons pas besoin car nous cherchons uniquement à déterminer les attracteurs du système. En toute généralité, il nous faudrait déterminer cet ordre pour savoir exactement vers quel attracteur on se dirige. L'hypothèse faite par Thomas est donc une hypothèse relativement large, puisqu'elle consiste simplement à imposer qu'un seul automate est mis à jour à chaque instant, sans spécifier à priori l'ordre de ces mises à jour.

3.3 Etude de la dynamique des réseaux booléens

Nous avons défini dans ce chapitre le modèle de réseaux d'automates booléens et nous avons vu qu'il fournit un cadre intéressant pour l'étude de systèmes biologiques³. Nous proposons à présent une synthèse des différents outils qui existent aujourd'hui pour permettre une analyse plus détaillée de la dynamique de ces réseaux. La littérature à ce sujet est conséquente et très variée, puisqu'elle se situe à la frontière des mathématiques et de l'informatique. Nous retenons ici trois voies qui nous semblent complémentaires et permettent d'obtenir des résultats intéressants.

La première de ces voies consiste à étudier plus avant les propriétés mathématiques des itérations discrètes sur un ensemble dénombrable. A partir d'outils métriques et différentiels, l'analyse d'un système dynamique particulier sur le n -cube $\{0, 1\}^n$ nous permet de trouver certaines propriétés du portrait de phase, telle l'attractivité des attracteurs, cycles ou points fixes. La seconde voie s'oriente non pas vers l'analyse détaillé d'un réseau particulier mais vers l'étude d'un ensemble de réseaux présentant certaines propriétés communes relatives à leurs structures. Il s'agit de la théorie des réseaux booléens aléatoires. Enfin, nous verrons une troisième voie, plus récente, dont l'objet est de trouver des liens entre le graphe d'interaction \mathcal{GI} et le graphe de transition \mathcal{GT} .

3.3.1 Outil métrique et dérivée discrète

Soit un réseau d'automates booléens $\mathcal{R} = (X, B, F)$, avec $X = \{1, \dots, n\}$, $B \in \mathfrak{M}_n(\{0, 1\})$ et F une application du n -cube $\{0, 1\}^n$ dans lui-même. Nous rappelons que le fonctionnement d'un tel réseau dans un des modes parallèle, série ou mixte conduit à un système dynamique discret autonome, c'est-à-dire à l'itération d'une application $G : \{0, 1\}^n \rightarrow \{0, 1\}^n$. Dans le cas du mode parallèle, l'application G est égale à la fonction de transition globale F du réseau \mathcal{R} . Nous nous intéressons alors aux suites solutions de l'équation récurrente :

$$\begin{cases} x^0 \in \{0, 1\}^n \\ x^{t+1} = F(x^t), \text{ pour tout } t \in \mathbb{N} \end{cases} \quad (3.13)$$

Afin d'étudier de telles équations, nous avons besoin de plusieurs outils (nous nous référons pour toute cette étude à l'ouvrage [62]). Tout d'abord, nous définissons sur $\{0, 1\}^n$ une distance vectorielle :

Définition 3.16 *Pour tout $x, y \in \{0, 1\}^n$, nous définissons le vecteur booléen $d(x, y) \in \{0, 1\}^n$ par :*

$$d(x, y) = \begin{pmatrix} x_1 \oplus y_1 \\ \vdots \\ x_n \oplus y_n \end{pmatrix}$$

où \oplus désigne l'addition modulo 2 dans $\{0, 1\}$, c'est-à-dire :

$$\begin{cases} 0 \oplus 0 = 1 \oplus 1 = 0 \\ 0 \oplus 1 = 1 \oplus 0 = 1 \end{cases}$$

³Concernant l'application en biologie, il existe de nombreuses références dans la littérature. On pourra notamment consulter [65] qui montre sur un exemple biologique précis l'application du modèle asynchrone.

Grâce à cette distance discrète, nous pouvons énoncer l'inégalité suivante (correspondant à une inégalité de Lipschitz dans le cas continu) :

$$\forall x, y \in \{0, 1\}^n, \quad d(F(x), F(y)) \leq B d(x, y)$$

où B est la matrice d'incidence associée à la fonction F .

Remarque 3.8 *Dans cette partie, toutes les opérations sont booléennes. En particulier, le produit de la matrice B par le vecteur $d(x, y)$ dans l'inégalité précédente est un produit matrice-vecteur booléen. L'ordre \leq est l'ordre naturel des booléens ($0 < 1$).*

Cette inégalité est fondamentale puisqu'elle relie la dynamique du réseau à la matrice d'incidence B de ce réseau. Ainsi, si l'on considère un point fixe ξ de F , cette inégalité entraîne :

$$d(x^t, \xi) \leq B^t d(x^0, \xi)$$

qui nous fournit une condition nécessaire sur B pour que l'itération (3.13) atteigne le point fixe ξ au bout de t pas.

Un autre outil permettant d'étudier ces itérations est la notion de *jacobiennne discrète* qui est une version discrète de la jacobienne d'un système dynamique continu. Afin d'en donner une définition, nous posons d'abord la notation suivante :

Pour tout $x \in \{0, 1\}^n$ et $j \in \{1, \dots, n\}$, on note \tilde{x}^j le vecteur booléen défini par :

$$\tilde{x}^j = (x_1, x_2, \dots, \neg x_j, \dots, x_n)$$

L'ensemble des \tilde{x}^j , pour $j \in \{1, \dots, n\}$ est noté $V_1(x)$: il constitue l'ensemble des *premiers voisins* de x .

Définition 3.17 *On nomme jacobienne discrète de F en un point x du n -cube $\{0, 1\}^n$ la matrice booléenne $F'(x) = (f_{ij}(x))_{i,j \in \{1, \dots, n\}} \in \mathfrak{M}_n(\{0, 1\})$ telle que :*

$$\forall i, j \in \{1, \dots, n\}, \quad f_{ij}(x) = f_i(x) \oplus f_i(\tilde{x}^j)$$

On remarque de façon immédiate le résultat suivant :

$$\max_{x \in \{0, 1\}^n} F'(x) = B$$

(la fonction max est prise composante par composante) qui nous permet de relier la jacobienne de F à la matrice d'incidence B de F .

La notion de jacobienne discrète permet d'obtenir des résultats locaux de convergence de l'itération (3.13). En particulier, nous avons le résultat suivant :

Proposition 3.4 *Etant donné un point fixe ξ de F , pour que ξ soit attractif dans son voisinage premier, il faut et il suffit que les deux conditions suivantes soient réalisées :*

- $F'(\xi)$ a au plus un seul 1 par colonne.
- Il existe une matrice de permutation P telle que $P^T F'(\xi) P$ est triangulaire inférieure stricte.

La notion d'attractivité dans le voisinage premier d'un point fixe est à mettre en relation avec la stabilité d'un point d'équilibre dans un système continu. Etant donné un système dynamique discret, il est donc tout-à-fait envisageable de mener une analyse mathématique de son portrait de phase comme on le fait pour un système continu (recherche des attracteurs, analyse de leur stabilité). On consultera les très nombreux exemples de [62] pour avoir une meilleure idée du type d'applications entraînées par une telle étude.

3.3.2 Réseaux booléens aléatoires

Grâce à la propriété 3.1, nous savons que l'itération parallèle (3.13) admet un portrait de phase simple, composé d'un nombre fini de bassins d'attractions, chacun d'entre eux comportant un unique attracteur. Ce résultat théorique peut être complété par l'analyse plus détaillée de ce portrait de phase, en utilisant les outils décrits dans la partie précédente. Les systèmes dynamiques discrets dont l'équation d'évolution est donnée par (3.13) sont donc des systèmes déterministes dont l'étude théorique du portrait de phase conduit à la description exhaustive de leurs dynamiques.

Néanmoins, lorsque n prend des valeurs élevées (rappelons ici par exemple que le génome d'un être humain contient approximativement $n = 25\,000$ gènes), le cardinal de l'espace des phases atteint des valeurs gigantesques (pour $n = 25\,000$, $2^n \simeq 10^{7500}$). Il va de soi que même sur ordinateur, l'étude exhaustive du portrait de phase devient impossible. Nous rencontrons également un autre problème. Etant donné la dimension de l'espace d'états, nous pouvons voir apparaître dans le portrait de phase des cycles de longueur très grande. Ces grands cycles, bien qu'étant théoriquement des attracteurs du réseau, ne sont pas satisfaisant du point de vue de la modélisation car aucune simulation ne permet de voir réellement l'accomplissement du cycle. Bien que nous soyons dans un cas purement déterministe, le terme *chaotique* est parfois abusivement employé pour qualifier de tels comportements.

L'approche analytique d'un réseau booléen présente donc ses limites, surtout lorsque la dimension n du réseau est élevée. Une autre approche de ces réseaux consiste donc non plus à les analyser séparément, mais à les regrouper dans des classes de réseaux et à étudier certaines propriétés génériques de ces classes. Il s'agit de la théorie des réseaux booléens aléatoires (voir [19, 25, 26, 29]). L'une des façons les plus communes de définir ces réseaux est de fixer leur connectivité maximale K .

Définition 3.18 (Réseaux booléens aléatoires) *Soient deux entiers naturels non nuls N et K (on suppose $K \leq N$). L'ensemble $\mathcal{RBA}(N, K)$ est l'ensemble des réseaux booléens (X, B, F) tels que $X = \{1, \dots, N\}$ et tels que pour chaque automate $i \in X$, on ait :*

$$k_i = \sum_{j=1}^N b_{ij} \leq K$$

Choisir un réseau dans $\mathcal{RBA}(N, K)$ consiste donc à choisir aléatoirement une application F de $\{0, 1\}^n$ dans lui-même telle que la connectivité maximale des f_i soit inférieure ou égale à K . Une autre façon de définir les réseaux aléatoires est de ne pas considérer la connectivité mais d'imposer dans le choix des fonctions f_i la proportion $p \in [0.5, 1]$ de 0 (ou de 1, ce qui revient au même). Les articles [25] et [26] démontrent dans ce cadre certains résultats intéressants.

Dans le cadre de la définition précédente, le but de l'étude des réseaux aléatoires est de rechercher des propriétés génériques relatives à la notion de connectivité. Ces propriétés peuvent être obtenues de deux manières : soit de façon statistique en moyennant certains résultats sur un ensemble (plus ou moins grand) de simulations effectuées sur un ordinateur (voir e.g. [43]), soit en démontrant des résultats asymptotiques. A titre d'exemple, nous citons ici un résultat démontré pour la première fois (à notre connaissance) dans les années 60 par B. Harris [29] qui concerne le cas où la connectivité maximale K est égale à la dimension N . Dans ce cas, la variable aléatoire L déterminant la longueur des cycles d'un réseau admet pour moyenne :

$$E[L] \underset{N \rightarrow \infty}{\sim} \frac{\sqrt{2\pi} 2^N}{4} = O\left(2^{\frac{N}{2}}\right)$$

et pour variance :

$$\text{Var}(L) \underset{N \rightarrow \infty}{\sim} 2^N \left(\frac{2}{3} - \frac{2\pi}{16} \right) = O(2^N)$$

Ceci entraîne que dans le cas $K = N$ (c'est-à-dire en n'imposant aucune restriction à priori sur la connectivité), on trouve une grande proportion de réseaux présentant des “grands” cycles. Ce résultat est fondamental car il permet de relier une notion d'*ordre* dans le comportement des réseaux à la notion de connectivité. Si l'on considère l'application à la biologie, la notion d'*attracteur* des réseaux est souvent liée au phénomène de différenciation cellulaire (voir par exemple [43]). Ce que l'on peut déduire du résultat précédent est que la connectivité moyenne (c'est-à-dire le nombre moyen de gènes influençant chaque gène du réseau) dans un génome doit être relativement faible pour assurer un nombre moyen peu élevé d'attracteurs et surtout l'absence de grands cycles.

D'autres résultats concernant la classe $\mathcal{RBA}(N, K)$ (on pourra par exemple consulter [7] ainsi que sa bibliographie) sont disponibles. Ces résultats, qu'ils soient théoriques ou issus de simulations semblent confirmer qu'une connectivité faible tend à infléchir au réseau un comportement plus *ordonné*. Ceci est d'ailleurs conforté par les expériences récentes sur les réseaux de régulation génétique : on observe dans la nature une proportion relativement faible de gènes à forte connectivité (voir par exemple [74]).

3.3.3 Liens entre graphe de transition et graphe d'interaction

Le troisième et dernier type d'analyse de la dynamique des réseaux booléens que nous citons ici concerne un champ d'étude plus récent, qui consiste à rechercher, pour un réseau donné, des liens entre son graphe d'interaction \mathcal{GI} et son graphe de transition \mathcal{GT} . Dès les années 80, R. Thomas [76] propose des conjectures permettant d'explicitier certains de ces liens. Ces conjectures se basent sur les observations suivantes :

- Si le réseau (dans son comportement asynchrone) vérifie la propriété de *mutistationnarité*, c'est-à-dire si son graphe de transition comporte au moins deux points d'équilibre stables, alors son graphe d'interaction comporte une boucle de rétroaction positive (i.e. il existe un circuit fermé C dans \mathcal{GI} tel que chaque élément de C influence positivement l'élément qui le suit).
- A l'inverse, l'existence d'un cycle stable dans \mathcal{GT} semble impliquer dans \mathcal{GI} l'existence d'une boucle de rétroaction négative.

Ces conjectures ont été démontrées pour plusieurs classes de réseaux, non discrets. Nous citons ici différentes démonstrations, en laissant le soin au lecteur de constater que les différences essentielles se situent dans les hypothèses faites sur les classes de réseaux considérées : [13, 58, 71]. La démonstration la plus générale est à notre sens celle proposée par C. Soulé [73], qui montre que si un système dynamique continu :

$$\frac{dx}{dt} = F(x)$$

(où F est définie et différentiable sur un ouvert Ω) admet au moins deux points d'équilibre non dégénérés, alors il existe $a \in \Omega$ tel que le graphe d'interaction associé à la matrice jacobienne $J_F(a)$ de F en a (voir définitions dans [73]) admet un circuit positif.

L'intérêt de chercher des liens entre les graphes \mathcal{GI} et \mathcal{GT} est évident. Le nombre de sommets dans \mathcal{GI} étant de n , le graphe d'interaction est beaucoup plus simple à étudier que le graphe de transition (qui comporte un nombre exponentiel de sommets). Il existe cependant un second intérêt, tout aussi essentiel. En effet, lors de l'établissement de modèles discrets d'un réseau de régulation génétique, il faut une bonne connaissance du système biologique que nous voulons reproduire. Or, certaines parties de ce système peuvent être mal connues, voire tout simplement ignorées. Une analyse mathématique et informatique des différents modèles possibles peut donc conduire à certaines prédictions qui demanderont alors d'être vérifiées par l'expérience.

Nous avons défini dans ce chapitre la notion de réseau d'automates booléens, en nous basant sur la notion plus générale de réseau d'automates sur un graphe. Comme nous l'avons vu, les dynamiques de ces systèmes sont données par l'itération d'applications discrètes sur un ensemble dénombrable. Le champ d'étude de tels systèmes est vaste et varié. Nous avons notamment décrit trois domaines de recherche visant à comprendre comment s'organise la dynamique de ces réseaux. Nous allons voir dans le chapitre qui suit une application concrète en nous intéressant au problème de l'inférence.

Chapitre 4

Inférence d'un réseau booléen

Dans le chapitre précédent, nous avons défini la notion de réseau d'automates booléens (définition 3.11). Cette définition est statique ; afin d'y faire intervenir le temps, nous associons à un réseau une stratégie Ψ de mise à jour des automates, ou mode opératoire. Une fois cette stratégie choisie, nous pouvons associer à tout réseau R une application $F : \{0, 1\}^n \rightarrow \{0, 1\}^n$ et un graphe de transition \mathcal{GT} (cf. définition 3.3), c'est-à-dire un graphe dont l'ensemble des sommets est l'ensemble des configurations du réseau : $\mathcal{C} = \{0, 1\}^n$ (n est la dimension du réseau), et où les arêtes correspondent au passage d'une configuration donnée à sa configuration suivante (en d'autres termes : étant données deux configurations $c, c' \in \mathcal{C}$, il existe une arête allant de c à c' si et seulement si $c' = F(c)$). En utilisant la terminologie des systèmes dynamiques, l'ensemble des chemins de ce graphe de transition \mathcal{GT} constitue le *portrait de phase* du réseau R . Nous nous posons à présent le problème inverse : peut-on retrouver à partir d'une application F et de son graphe de transition le ou les réseaux qui, étant donnée une stratégie particulière, admettent comme portrait de phase le graphe \mathcal{GT} ? Nous allons voir dans ce chapitre qu'il est possible de répondre à cette question dans le cas d'une stratégie de mise à jour parallèle, c'est-à-dire où tous les automates évoluent simultanément à chaque top d'horloge (voir théorème 4.1). Ce problème porte le nom d'**inférence**, ou d'**identification** de réseau puisqu'il s'agit de retrouver, à partir des transitions entre configurations, quelle est la topologie du réseau (c'est-à-dire comment sont connectés les différents automates entre eux et quelles sont les fonctions d'activation des automates).

Dans l'application de la théorie des réseaux booléens à la biologie et l'étude de réseaux de régulation génétique, le problème de l'inférence est lié à la technologie déjà évoquée des biopuces à ADN. En effet, cette technique nous permet aujourd'hui d'obtenir, à l'échelle d'un organisme entier, le niveau d'expression des gènes à un instant donné. En répétant cette expérience à plusieurs instants successifs, nous obtenons ainsi un *patron d'expression* du génome de l'organisme sous certaines conditions fixées à l'avance. L'idée sous-jacente de l'inférence consiste donc à retrouver à partir de ces patrons les réseaux génétiques connus, et surtout de prédire certaines interactions encore inconnues.

Nous reprenons dans ce chapitre un algorithme présenté dans la littérature sous le nom de REVEAL (REVerse Engineering ALgorithm) (voir notamment [20, 46], voir aussi [2, 3]), dont nous proposons une preuve. Nous considérons également l'algorithme d'identification *partielle* permettant de retrouver l'ensemble des réseaux à partir d'un sous-graphe de \mathcal{GT} . Nous nous intéressons enfin au problème de consistance minimale pour lequel nous proposons notamment un résultat liant la taille des données et la probabilité d'inférer correctement le réseau correspondant (dans le cas d'un support minimal fixé).

4.1 Présentation du problème

4.1.1 L'inférence de réseaux booléens

Considérons un réseau d'automates booléens (X, B, F) . Nous rappelons ici que :

- X est l'ensemble (fini) des automates : $X = \{1, \dots, n\}$,
- B est une matrice booléenne carrée de taille n ,
- F est une application de $\{0, 1\}^n$ dans lui-même. Dans la suite, on notera f_i la i -ième coordonnée de F .

En choisissant comme stratégie de mise à jour le mode parallèle, nous pouvons exprimer la dynamique de ce réseau par le système dynamique discret autonome défini par l'itération suivante :

$$\begin{cases} x^0 \in \mathcal{C} \\ x^{t+1} = F(x^t), \text{ pour tout } t \in \mathbb{N} \end{cases}$$

où x^t est un vecteur booléen de dimension n quelconque et x^0 est un vecteur particulier (la condition initiale). L'ensemble $\mathcal{C} = \{0, 1\}^n$ est l'ensemble des configurations du réseau.

Comme nous l'avons évoqué dans le chapitre précédent, ce système peut être entièrement déterminé par son graphe de transition ou, de manière équivalente, par la table de vérité de la fonction F , c'est-à-dire par la table donnant, pour toute configuration $x \in \mathcal{C}$ la configuration suivante $F(x) \in \mathcal{C}$.

Afin de définir formellement la notion de table de vérité, nous définissons, pour tout vecteur booléen $x \in \{0, 1\}^n$ l'entier naturel $\sigma_n(x)$ compris entre 0 et $2^n - 1$ défini par :

$$\sigma_n(x) = \sum_{i=1}^n x_i 2^{n-i}$$

Pour tout $n \in \mathbb{N}^*$, l'application σ_n ainsi définie réalise une bijection de $\{0, 1\}^n$ dans l'ensemble $\{0, \dots, 2^n - 1\}$. Etant donné un entier $p \in \{0, \dots, 2^n - 1\}$, le vecteur $\sigma_n^{-1}(p)$ représente l'écriture binaire de p , poids fort en tête.

Nous définissons la table de vérité complète d'une fonction $F : \{0, 1\}^n \rightarrow \{0, 1\}^n$ par un couple (E, S) de matrices booléennes de taille $2^n \times n$, où :

- pour tout $i = 1 \dots 2^n$, la i -ième ligne de E est le vecteur booléen $L_i(E) = \sigma_n^{-1}(i - 1)$ représentant l'écriture en base 2 de l'entier $i - 1$ (poids fort en tête).
- pour tout $i = 1 \dots 2^n$, la i -ième ligne de S est donnée par : $L_i(S) = F(L_i(E))$

Exemple 4.1

Posons $n = 2$ et considérons la fonction

$$\begin{aligned} F : \{0, 1\}^2 &\longrightarrow \{0, 1\}^2 \\ (x_1, x_2) &\longmapsto (x_1 \vee x_2, x_1 \wedge x_2) \end{aligned}$$

La table de vérité de cette fonction est donnée par les matrices E et S suivantes :

$$E = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix} \quad S = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 1 \end{pmatrix}$$

On vérifie aisément que pour tout $1 \leq i \leq 4$, on a $L_i(S) = F(L_i(E))$.

Etant donnée une table de vérité $T = (E, S)$ complète, on dira qu'un réseau $R = (X, B, F)$ est **consistant** avec T si :

$$\forall i \in \{1, \dots, 2^n\}, \quad L_i(S) = F(L_i(E))$$

où F est la fonction de transition globale du réseau R .

Le problème de l'identification d'un réseau consiste donc à trouver, étant donnée la table de vérité T d'une fonction booléenne $F : \{0, 1\}^n \rightarrow \{0, 1\}^n$, s'il existe un ou plusieurs réseaux d'automates booléens consistants avec T .

4.1.2 Connectivité réelle d'une fonction booléenne

Dans tout ce qui suit, \mathcal{F}_n désigne l'ensemble des applications de $\{0, 1\}^n$ dans lui-même. L'espace des configurations $\mathcal{C} = \{0, 1\}^n$ étant fini, la donnée de la table de vérité T d'une application $F \in \mathcal{F}_n$ détermine entièrement F . Il est évident qu'il existe toujours au moins un réseau qui soit consistant avec T . Il suffit en effet de considérer le réseau complètement connecté R_c défini par :

- $X = \{1, \dots, n\}$,
- B est la matrice de dimension $n \times n$ ne contenant que des 1,
- F est l'application de \mathcal{F}_n dont la table de vérité est T .

Théoriquement, il existe donc toujours une solution au problème de l'identification de réseau. Néanmoins, le réseau R_c est d'un intérêt limité. En effet, la composante f_i de l'application F est définie sur $\{0, 1\}^n$, mais peut ne dépendre que de k variables, avec $k < n$. Or dans R_c , la *connectivité* de l'automate i , telle que nous l'avons définie précédemment vaut :

$$k_i = \sum_{j=1}^n b_{i,j} = n$$

Dans le cas où la fonction f_i dépend de $k < n$ variables, l'automate i admet k entrées effectives et non n . Le réseau complètement connecté R_c contient donc $n - k$ connexions inutiles. Afin de définir correctement le réseau, il nous faut déterminer quelles sont les connexions effectives entre les automates, c'est-à-dire de quelles variables les fonctions f_i dépendent effectivement. Ceci nous amène à définir la notion de connectivité *réelle* d'une fonction booléenne.

Définition 4.1 Soit une fonction booléenne $f : \{0, 1\}^n \rightarrow \{0, 1\}$. On dira que la variable x_i est muette dans l'expression de f si, pour tout $(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) \in \{0, 1\}^{n-1}$ on a :

$$f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n) = f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n)$$

Si une variable x_i de f n'est pas muette dans l'expression de f , elle est dite effective. Si au moins l'une des variables de f est muette, on dira que f est dégénérée.

On notera dans la suite $\mu(f)$ le nombre des variables muettes dans l'expression de f .

Notons $\alpha(n)$ le nombre de fonctions booléennes à n variables non dégénérées. Ce nombre est égal à (voir [30]) :

$$\alpha(n) = \sum_{k=0}^n C_n^k (-1)^{n-k} 2^k$$

Lorsque n tend vers $+\infty$, cette quantité est équivalente à 2^{2^n} (cf. [30]), ce qui signifie que la proportion de fonctions dégénérées décroît et tend vers 0 lorsque n augmente.

Cette définition nous permet d'exprimer la notion de connectivité réelle d'une fonction :

Définition 4.2 Soit une fonction booléenne $f : \{0,1\}^n \rightarrow \{0,1\}$. La connectivité réelle de la fonction f est l'entier $\kappa(f) = n - \mu(f)$. L'ensemble $\{i_j \mid j = 1, \dots, \kappa(f)\}$ des indices des variables effectives de f est appelé support de f et sera noté $\text{Supp}(f)$.

Par exemple, la fonction $f : \{0,1\}^n \rightarrow \{0,1\}$, pour $n > 3$ définie par :

$$\forall (x_1, \dots, x_n) \in \{0,1\}^n, \quad f(x_1, \dots, x_n) = x_4 \wedge (x_5 \vee x_2)$$

admet $n - 3$ variables muettes. Sa connectivité réelle vaut donc $\kappa(f) = 3$ et son support est : $\text{Supp}(f) = \{2, 4, 5\}$.

La notion de connectivité réelle nous permet d'exprimer un réseau booléen sous forme canonique, ce qui nous sera utile par la suite.

Définition 4.3 Un réseau $R = (X, B, F)$ est exprimé sous sa forme canonique si, pour chaque nœud $i \in \{1, \dots, n\}$, la connectivité réelle $\kappa(f_i)$ de la fonction f_i est égale à l'entier k_i défini par : $k_i = \sum_{j=1}^n b_{ij}$.

Remarque 4.1 Dans un réseau canonique, $b_{ij} = 1$ est donc équivalent au fait que la variable x_j est une variable effective de la fonction f_i . A l'inverse, $b_{ij} = 0$ si et seulement si x_j est muette dans l'expression de f_i . En d'autres termes,

$$\text{Supp}(f) = \{j \in \{1, \dots, n\} \mid b_{ij} = 1\}$$

Rappelons ici que la matrice B est la matrice d'incidence du réseau, et qu'elle représente les arêtes du graphe d'interaction $\mathcal{GI} = (X, \mathcal{A})$ (cf. définition 3.11). Dans le cas d'un réseau canonique, les connexions représentées par la matrice d'incidence B sont donc des connexions effectives, dans le sens que l'évolution de chaque nœud dépend effectivement des entrées données par la matrice B .

Ces définitions étant établies, nous proposons dans la partie suivante un théorème assurant l'existence d'un réseau canonique solution du problème d'identification énoncé plus haut. Dans le cas où la table de vérité donnée est complète, nous montrons de plus l'unicité de cette solution.

4.1.3 Théorème d'inférence de réseau dans le cas synchrone

Nous allons démontrer le théorème suivant :

Théorème 4.1 n est un entier naturel non nul. Etant donnée une table de vérité complète $T = (E, S)$ (E et S sont de taille $2^n \times n$), il existe un unique réseau d'automates booléens canonique $R = (X, B, F)$ tel que son graphe d'itération soit consistant avec la table T . De plus, il existe un algorithme qui construit ce réseau en temps fini.

Remarque 4.2 D'un point de vue dynamique, nous considérons ici uniquement la stratégie de mise à jour parallèle. Rappelons que dans ce cas, le graphe d'itération d'un réseau $R = (X, B, F)$ est défini à partir de l'application de transition globale F . Le problème de l'identification de réseaux qui est posé par ce théorème se ramène donc à retrouver l'ensemble X et la matrice B à partir de la donnée de la table de vérité de l'application F . Sous cette forme, nous voyons bien que le problème d'inférence de réseaux est donc un problème purement statique.

Pour démontrer l'existence d'une solution, nous proposerons dans la suite (partie 4.2) une méthode constructive en nous basant sur l'algorithme REVEAL [20, 46] dont nous prouverons la correction. Nous montrons ici l'unicité de cette solution.

Soit $T = (E, S)$ une table de vérité complète de dimension n . Cette table détermine une application F appartenant à \mathcal{F}_n . L'ensemble des automates d'un réseau consistant est bien sûr l'ensemble $X = \{1, \dots, n\}$. Supposons à présent qu'il existe deux réseaux canoniques $R = (X, B, F)$ et $R' = (X, B', F)$ consistants avec T . Nous allons montrer qu'alors $B = B'$, c'est-à-dire que les deux réseaux R et R' ont les mêmes connexions.

Lemme 4.2 *Si $R = (X, B, F)$ et $R' = (X, B', F)$ sont deux réseaux canoniques consistants avec une table de vérité complète T , alors nécessairement $B = B'$.*

Démonstration

R et R' étant des réseaux canoniques, les matrices B et B' vérifient :

$$\forall i \in \{1, \dots, n\}, \quad \sum_{j=1}^n b_{ij} = \sum_{j=1}^n b'_{ij} = \kappa(f_i)$$

Les i -ièmes lignes respectives de B et B' comportent donc le même nombre de 1. Il nous faut démontrer qu'elles sont égales. Supposons qu'il existe $i, j \in \{1, \dots, n\}$ tels que $b_{ij} \neq b'_{ij}$. Par exemple, supposons que $b_{ij} = 1$ et $b'_{ij} = 0$. Alors, R étant canonique, $b_{ij} = 1$ implique que x_j est une variable effective de f_i (cf. remarque 4.1). De même, puisque R' est canonique, $b'_{ij} = 0$ implique que x_j est muette dans l'expression de f_i . Cette contradiction entraîne donc l'égalité $b_{ij} = b'_{ij}$, pour tout i et j dans $\{1, \dots, n\}$. ■

Ainsi, s'il existe un réseau canonique consistant avec une table complète T , alors ce réseau est unique. L'objectif de la partie qui suit est de démontrer l'existence d'une telle solution et de donner une méthode algorithmique pour la construire.

4.2 Existence d'une solution au problème d'identification

4.2.1 Entropie de Shannon et fonctions booléennes

Afin de prouver l'existence d'un réseau consistant avec une table, nous allons utiliser la théorie de l'information (voir [70], l'un des articles fondateurs de cette théorie par Claude Shannon). Nous allons plus particulièrement utiliser la notion d'**entropie** d'une fonction booléenne.

Soit n un entier naturel non nul. Nous considérons une table de vérité complète de dimension n : $T = (E, S)$. T est la table d'une application $F \in \mathcal{F}_n$. Nous notons S_i la i -ième colonne de la matrice S et E_i la i -ième colonne de E (i allant de 1 à n). Le vecteur S_i appartient à $\{0, 1\}^{2^n}$ et représente la table de vérité de la fonction f_i . En effet, nous considérons l'application Φ qui, à tout vecteur $U \in \{0, 1\}^{2^n}$ associe la fonction $f : \{0, 1\}^n \rightarrow \{0, 1\}$ définie par :

$$\forall x \in \{0, 1\}^n, \quad f(x) \stackrel{\text{def}}{=} U_{\sigma_n(x)+1}$$

Cette application Φ réalise clairement une bijection de $\{0, 1\}^{2^n}$ vers $\{0, 1\}^{\{0, 1\}^n}$.

Nous avons alors, pour tout $i \in \{1, \dots, n\}$: $f_i = \Phi(S_i)$. En d'autres termes, chaque colonne S_i de la matrice S détermine de façon unique la fonction booléenne f_i .

Afin de définir l'entropie d'une fonction booléenne $f : \{0, 1\}^n \rightarrow \{0, 1\}$, nous allons définir l'entropie du vecteur $U = \Phi^{-1}(f) \in \{0, 1\}^{2^n}$ associé. Pour cela, nous considérons les deux ensembles :

$$I_U^0 = \{i \in \{0, 1\}^{2^n} \mid U_i = 0\}$$

$$I_U^1 = \{i \in \{0, 1\}^{2^n} \mid U_i = 1\}$$

Nous posons $P_U^0 = \frac{1}{2^n} \text{card}(I_U^0)$ et $P_U^1 = \frac{1}{2^n} \text{card}(I_U^1)$. Par définition, P_U^0 et P_U^1 sont positifs ou nuls, ils représentent respectivement les proportions de 0 et de 1 dans le vecteur U .

Enfin, nous définissons la fonction réelle :

$$\begin{aligned} \varphi :]0, +\infty[&\longrightarrow \mathbb{R} \\ p &\longmapsto -p \log(p) \end{aligned}$$

(où \log désigne la fonction logarithme en base 2) que nous prolongeons par continuité en zéro :

$$\varphi(0) \stackrel{\text{def}}{=} 0$$

Définition 4.4 *L'entropie d'un vecteur $U \in \{0, 1\}^{2^n}$ est le réel défini par :*

$$H(U) = \varphi(P_U^0) + \varphi(P_U^1)$$

Dans la suite, nous noterons $H(U) = -P_U^0 \log(P_U^0) - P_U^1 \log(P_U^1)$ en utilisant la convention suivante : si $P_U^0 = 0$, alors $P_U^0 \log(P_U^0) = 0$ (idem pour P_U^1).

On a par définition $P_U^1 = 1 - P_U^0$. Ceci nous permet d'exprimer l'entropie $H(U)$ comme une fonction de P_U^0 . Cette fonction permet de quantifier l'information contenue dans le vecteur U . P_U^0 étant compris entre 0 et 1, $H(U)$ est également comprise entre 0 et 1. Elle est maximale lorsque $P_U^0 = P_U^1 = 0,5$ (il y a autant de 0 que de 1 dans le vecteur U) et vaut 1. Au contraire, elle atteint sa valeur minimale 0 lorsque $P_U^0 = 0$ ou 1, c'est-à-dire lorsque le vecteur U n'est composé que de 0 ou que de 1 (voir figure 4.1 ci-après).

Afin de généraliser cette définition à un ensemble fini de vecteurs, il nous faut définir les ensembles suivants :

Soit $\mathcal{U} = (U_1, \dots, U_p)$ un p -uplet de vecteurs booléens de dimension 2^n . Pour $x \in \{0, 1\}^p$ fixé, nous définissons l'ensemble :

$$I_{\mathcal{U}}^x = \{i \in \{0, 1\}^{2^n} \mid \forall j \in \{1, \dots, p\}, (U_j)_i = x_j\}$$

Cet ensemble étant fini, on considère la quantité $P_{\mathcal{U}}^x = \frac{1}{2^n} \text{card}(I_{\mathcal{U}}^x)$. L'entropie de (U_1, \dots, U_p) se définit alors comme suit.

Définition 4.5 *Soit $\mathcal{U} = (U_1, \dots, U_p)$ un p -uplet de vecteurs booléens de dimension 2^n . L'entropie de \mathcal{U} est la quantité définie par :*

$$H(\mathcal{U}) = H(U_1, \dots, U_p) = - \sum_{x \in \{0, 1\}^p} P_{\mathcal{U}}^x \log(P_{\mathcal{U}}^x)$$

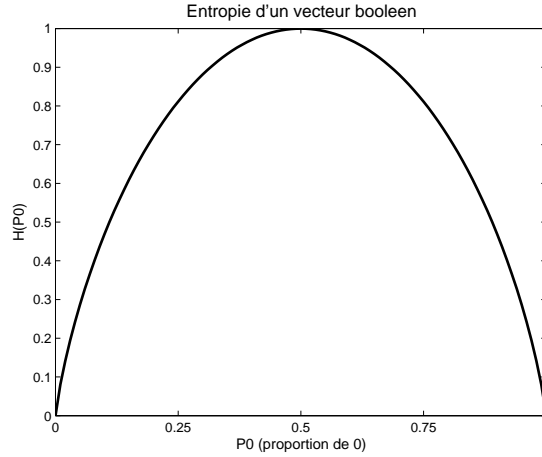


FIG. 4.1: Entropie d'un vecteur booléen U en fonction de la proportion P_U^0 de zéros dans U . L'entropie $H(U)$ est comprise entre 0 et 1, elle est maximale lorsque U contient autant de 0 que de 1, et minimale lorsque U ne contient que des 0 ou que des 1.

$H(U)$ représente la quantité d'information contenue par l'ensemble des vecteurs U_i . Considérons une table de vérité complète $T = (E, S)$. Comme nous l'avons déjà dit plus haut, les vecteurs colonnes S_i de S représentent les tables de vérité des fonctions booléennes $f_i : \{0, 1\}^n \rightarrow \{0, 1\}$. Afin de déterminer de quelles variables dépend effectivement ces fonctions, nous allons déterminer une condition nécessaire et suffisante basée sur le calcul des entropies de différents vecteurs colonnes de E et de S .

Proposition 4.3 n est un entier naturel non nul, $T = (E, S)$ est la table de vérité complète de $F \in \mathcal{F}_n$ (E et S sont de dimension $2^n \times n$).

Soit $i \in \{1, \dots, n\}$ et (i_1, \dots, i_k) un k -uplet d'indices appartenant à $\{1, \dots, n\}$. Les deux assertions suivantes sont équivalentes :

(i) Les variables x_j , pour $j \notin \{i_1, \dots, i_k\}$ sont muettes dans l'expression de la fonction f_i , en d'autres termes :

$$\text{Supp}(f_i) \subset \{i_1, \dots, i_k\}$$

(ii) L'égalité suivante est vérifiée : $H(S_i, E_{i_1}, \dots, E_{i_k}) = H(E_{i_1}, \dots, E_{i_k})$

Si de plus k est égal à la connectivité réelle de f_i , alors on a l'égalité :

$$\text{Supp}(f_i) = \{i_1, \dots, i_k\}$$

Démonstration

Posons $\mathbb{E} = (E_{i_1}, \dots, E_{i_k})$ et $\mathbb{E}' = (S_i, E_{i_1}, \dots, E_{i_k})$. Les entropies de \mathbb{E} et de \mathbb{E}' se calculent comme suit :

$$\begin{cases} H(\mathbb{E}) &= - \sum_{x \in \{0,1\}^k} P_{\mathbb{E}}^x \log(P_{\mathbb{E}}^x) \\ H(\mathbb{E}') &= - \sum_{x \in \{0,1\}^k} P_{\mathbb{E}'}^{0,x} \log(P_{\mathbb{E}'}^{0,x}) - \sum_{x \in \{0,1\}^k} P_{\mathbb{E}'}^{1,x} \log(P_{\mathbb{E}'}^{1,x}) \end{cases} \quad (4.1)$$

Dans la deuxième ligne, le terme $P_{\mathbb{E}'}^{0,x}$, pour $x \in \{0,1\}^k$ désigne en fait $P_{\mathbb{E}'}^{\tilde{x}}$ où $\tilde{x} = (0, x_1, \dots, x_k)$. De même, $P_{\mathbb{E}'}^{1,x}$ désigne $P_{\mathbb{E}'}^{\tilde{x}}$ où $\tilde{x} = (1, x_1, \dots, x_k)$. Pour x fixé dans $\{0,1\}^k$, nous avons par définition :

$$I_{\mathbb{E}}^x = \{j \in \{1, \dots, 2^n\} \mid (E_{i_1})_j = x_1, \dots, (E_{i_k})_j = x_k\}$$

L'ensemble $I_{\mathbb{E}}^x$ est donc l'union disjointe des deux ensembles suivants :

$$\begin{aligned} I_{\mathbb{E}'}^{0,x} &= \{j \in \{1, \dots, 2^n\} \mid (S_i)_j = 0, (E_{i_1})_j = x_1, \dots, (E_{i_k})_j = x_k\} \\ I_{\mathbb{E}'}^{1,x} &= \{j \in \{1, \dots, 2^n\} \mid (S_i)_j = 1, (E_{i_1})_j = x_1, \dots, (E_{i_k})_j = x_k\} \end{aligned}$$

On en déduit l'égalité $\text{card}(I_{\mathbb{E}}^x) = \text{card}(I_{\mathbb{E}'}^{0,x}) + \text{card}(I_{\mathbb{E}'}^{1,x})$ et donc :

$$P_{\mathbb{E}}^x = P_{\mathbb{E}'}^{0,x} + P_{\mathbb{E}'}^{1,x}$$

En utilisant ceci dans (4.1), nous obtenons :

$$H(\mathbb{E}) = - \sum_{x \in \{0,1\}^k} (P_{\mathbb{E}'}^{0,x} + P_{\mathbb{E}'}^{1,x}) \log(P_{\mathbb{E}'}^{0,x} + P_{\mathbb{E}'}^{1,x}) \quad (4.2)$$

Supposons à présent que l'on ait $H(\mathbb{E}) = H(\mathbb{E}')$ et montrons qu'alors f_i ne dépend pas des variables x_{i_1}, \dots, x_{i_k} . Grâce à (4.1) et (4.2), nous obtenons :

$$\sum_{x \in \{0,1\}^k} \left[P_{\mathbb{E}'}^{0,x} \left(\log(P_{\mathbb{E}'}^{0,x} + P_{\mathbb{E}'}^{1,x}) - \log(P_{\mathbb{E}'}^{0,x}) \right) + P_{\mathbb{E}'}^{1,x} \left(\log(P_{\mathbb{E}'}^{0,x} + P_{\mathbb{E}'}^{1,x}) - \log(P_{\mathbb{E}'}^{1,x}) \right) \right] = 0 \quad (4.3)$$

Par définition, nous savons que pour tout $x \in \{0,1\}^k$, les quantités $P_{\mathbb{E}'}^{0,x}$ et $P_{\mathbb{E}'}^{1,x}$ sont positives ou nulles. Dans l'équation (4.3), le membre de gauche est donc une somme de termes positifs ou nuls. Nous en déduisons que pour tout $x \in \{0,1\}^k$,

$$\begin{cases} P_{\mathbb{E}'}^{0,x} \left(\log(P_{\mathbb{E}'}^{0,x} + P_{\mathbb{E}'}^{1,x}) - \log(P_{\mathbb{E}'}^{0,x}) \right) = 0 & (1) \\ P_{\mathbb{E}'}^{1,x} \left(\log(P_{\mathbb{E}'}^{0,x} + P_{\mathbb{E}'}^{1,x}) - \log(P_{\mathbb{E}'}^{1,x}) \right) = 0 & (2) \end{cases}$$

Pour résoudre l'équation (1), deux cas se présentent :

- soit $P_{\mathbb{E}'}^{0,x} = 0$,
- soit $P_{\mathbb{E}'}^{0,x} \neq 0$ et alors $\log(P_{\mathbb{E}'}^{0,x} + P_{\mathbb{E}'}^{1,x}) - \log(P_{\mathbb{E}'}^{0,x}) = 0$

Ce deuxième cas entraîne :

$$\log \left(1 + \frac{P_{\mathbb{E}'}^{1,x}}{P_{\mathbb{E}'}^{0,x}} \right) = 0$$

ce qui nous conduit, étant donné que $P_{\mathbb{E}'}^{0,x} \neq 0$, à : $P_{\mathbb{E}'}^{1,x} = 0$.

Nous avons donc deux cas distincts : soit $P_{\mathbb{E}'}^{0,x} = 0$, soit $P_{\mathbb{E}'}^{1,x} = 0$. Par un raisonnement analogue, l'équation (2) nous conduit à la même conclusion.

Il nous faut à présent interpréter ce résultat. Posons $x \in \{0,1\}^k$. Nous notons $\mathcal{Y}(x)$ l'ensemble des vecteurs y de $\{0,1\}^n$ tels que :

$$\forall j \in \{1, \dots, k\}, \quad y_{i_j} = x_j$$

Soit A la matrice de dimension $2^n \times (n+1)$ composée des vecteurs colonnes (E_1, \dots, E_n, S_i) . Cette matrice représente la table de vérité de la fonction f_i sur $\{0, 1\}^n$. Nous notons $A(x)$ la sous-matrice de A composée uniquement des lignes d'indice $\sigma_n(y) + 1$, pour tout y appartenant à $\mathcal{Y}(x)$. La dernière colonne de $A(x)$ nous donne les valeurs prises par la fonction f_i aux points $y \in \mathcal{Y}(x)$.

Par définition, les quantités $P_{\mathbb{E}^i}^{0,x}$ et $P_{\mathbb{E}^i}^{1,x}$ représentent respectivement les proportions de 0 et de 1 dans cette dernière colonne. Le fait que $P_{\mathbb{E}^i}^{0,x} = 0$ (resp. $P_{\mathbb{E}^i}^{1,x} = 0$) entraîne donc que la dernière colonne est composée uniquement de 1 (resp. de 0). On en déduit que la fonction f_i est constante sur $\mathcal{Y}(x)$. Ceci équivaut au fait que les variables x_j , pour $j \notin \{i_1, \dots, i_k\}$, sont muettes dans l'expression de f_i . D'où :

$$\text{Supp}(f_i) \subset \{i_1, \dots, i_k\} \quad (4.4)$$

Réciproquement, si nous supposons que le support de f_i est inclus dans $\{i_1, \dots, i_k\}$, alors en remontant les calculs précédents, il est clair que l'on obtient l'égalité $H(\mathbb{E}) = H(\mathbb{E}')$.

Enfin, la relation (4.4) entraîne l'inégalité : $\kappa(f_i) \leq k$. Si nous supposons que $\kappa(f_i) = k$, alors nous pouvons en déduire que $\{i_1, \dots, i_k\}$ est égal au support de f_i . ■

Afin d'illustrer ce résultat, observons l'exemple suivant en dimension 3.

Exemple 4.2

Posons $n = 3$ et considérons la table de vérité donnée par le tableau suivant :

E_1	E_2	E_3	S_1	S_2	S_3
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	1	0	0
0	1	1	0	0	1
1	0	0	1	0	1
1	0	1	0	1	1
1	1	0	1	0	0
1	1	1	0	1	1

Considérons par exemple la fonction f_2 , dont la table est donnée par la 5-ème colonne du tableau (il s'agit de S_2 , la deuxième colonne de S).

Calculons tout d'abord l'entropie $H(E_1, E_3)$. Pour cela, nous calculons les quantités P_{E_1, E_3}^x , pour $x \in \{0, 1\}^2$:

$$P_{E_1, E_3}^{0,0} = P_{E_1, E_3}^{0,1} = P_{E_1, E_3}^{1,0} = P_{E_1, E_3}^{1,1} = \frac{1}{4}$$

d'où l'on déduit :

$$H(E_1, E_3) = -4 \times \frac{1}{4} \log\left(\frac{1}{4}\right) = 2$$

De la même manière, nous calculons $H(S_2, E_1, E_3)$ en calculant les quantités P_{S_2, E_1, E_3}^x , pour $x \in \{0, 1\}^3$:

$$\begin{cases} P_{E_1, E_3, S_2}^{0,0,0} = P_{E_1, E_3, S_2}^{0,1,0} = P_{E_1, E_3, S_2}^{1,0,0} = P_{E_1, E_3, S_2}^{1,1,1} = \frac{1}{4} \\ P_{E_1, E_3, S_2}^{0,0,1} = P_{E_1, E_3, S_2}^{0,1,1} = P_{E_1, E_3, S_2}^{1,0,1} = P_{E_1, E_3, S_2}^{1,1,0} = 0 \end{cases}$$

D'où l'on déduit que $H(E_1, E_3, S_2) = 2$.

D'après la proposition 4.3, nous pouvons donc conclure que x_2 est muette dans l'expression de f_2 . Afin de retrouver la table de f_2 , il nous faut l'extraire de la table initiale. Pour cela, nous allons considérer la matrice A qui est la matrice composée des colonnes (E_1, E_3, S_2) (voir figure 4.2, table du milieu).

Cette matrice est de taille $2^n \times (n+1) = 8 \times 3$ et contient donc des redondances. Plus précisément, puisque f_2 ne dépend effectivement que de deux variables, sa table de vérité doit être de dimension $2^2 \times 3 = 4 \times 3$. Afin d'éliminer les redondances et d'obtenir la table de $f_2(x_1, x_3)$ dans $\{0, 1\}^2$, nous extrayons de la matrice A la sous-matrice composée des lignes 1, 2, 5 et 6 (voir figure 4.2, table de droite).

On reconnaît alors la table de la fonction ET. La connectivité réelle de la fonction f_2 vaut donc $\kappa(f_2) = 2$ et elle s'exprime ainsi :

$$f_2(x_1, x_2, x_3) = f_2(x_1, x_3) = x_1 \wedge x_3$$

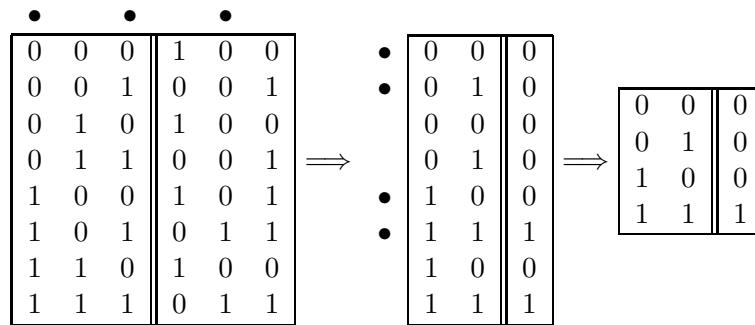


FIG. 4.2: Une fois connues les variables effectives de f_2 , nous extrayons de la table initiale (à gauche) la table de vérité de f_2 (au milieu). En enlevant les redondances, nous obtenons la table de droite correspondant à la fonction $f_2(x_1, x_3) = x_1 \wedge x_3$.

Grâce à cette proposition, nous pouvons à présent expliciter l'algorithme REVEAL et démontrer qu'il permet de construire un réseau d'automates booléens canonique et consistant avec une table de vérité complète.

4.2.2 Algorithme d'identification

Nous reprenons dans cette partie le principe de l'algorithme REVEAL (on se référera par exemple à [20, 46]). Cet algorithme permet, étant donnée une table de vérité T complète, de retrouver un réseau canonique, consistant avec T .

Le principe de cet algorithme est le suivant : Nous disposons d'une table de vérité complète $T = (E, S)$ de dimension n . Le but est de trouver, pour $i \in \{1, \dots, n\}$, l'ensemble V_i des variables effectives de la fonction f_i (c'est-à-dire le support de f_i). Pour cela, nous allons procéder par étapes, en regardant d'abord s'il existe des fonctions de connectivité réelle 1, puis, si certains V_i sont encore indéterminés, nous regardons s'il existe des fonctions dont la connectivité réelle est 2 et ainsi de suite. L'algorithme s'arrête dès que tous les V_i sont déterminés (voir algorithme 1 ci-dessous). Il existe une étape 0 non représentée dans l'algorithme 1, il s'agit du

Algorithme 1 - REVEAL - Identification de réseaux booléens.

Etant donnée une table de vérité complète de dimension n , cet algorithme trouve, pour tout $j \in \{1, \dots, n\}$ l'ensemble V_j qui est égal au support de la fonction f_j .

n : entier ≥ 1 ;

E, S : Matrices booléennes de dimension $2^n \times n$;

[Les matrices E et S représentent une table de vérité complète de dimension n . On suppose que les fonctions f_i définies par les colonnes S_i ne sont pas constantes sur $\{0, 1\}^n$.]

$I \leftarrow \{1, \dots, n\}$;

[I est l'ensemble des indices à traiter]

Pour k de 1 à n faire

[On cherche les fonctions de connectivité réelle k .]

Pour $i \in I$ faire

Pour (i_1, \dots, i_k) un k -uplet ordonné de $\{1, \dots, n\}$ faire

Si $(H(S_i, E_{i_1}, \dots, E_{i_k}) = H(E_{i_1}, \dots, E_{i_k}))$ **Alors**

[f_i est de connectivité k]

<Noeud 1>

$V_i \leftarrow (i_1, \dots, i_k)$;

[Les variables de la fonction f_i sont désormais trouvées : on l'enlève de I et on sort de cette boucle]

$I \leftarrow I \setminus \{i\}$;

Break ;

Fin Si

Fin Pour

Fin Pour

Retourner $V = (V_i)_{i=1..n}$;

Fin Pour

cas de la connectivité 0. Si f_i est de connectivité 0, cela signifie que c'est une fonction constante sur $\{0, 1\}^n$. Ce cas est facile à détecter et ne pose pas de problème algorithmique. Par un souci de concision, nous faisons donc l'hypothèse qu'aucune fonction f_i n'est constante sur $\{0, 1\}^n$.

Afin d'achever la preuve du théorème 4.1, il nous faut démontrer que l'algorithme 1 nous permet de construire un réseau canonique qui est consistant avec la table T . Pour cela, nous montrons le résultat suivant.

Proposition 4.4 *Etant donnée une table complète $T = (E, S)$ de dimension n , l'algorithme 1 renvoie en temps fini, pour chaque nœud $i \in \{1, \dots, n\}$ un sous-ensemble V_i de $\{1, \dots, n\}$. Nous pouvons alors construire le réseau $R = (X, B, F)$ dans lequel :*

- $X = \{1, \dots, n\}$,
- B est la matrice booléenne définie par :

$$\forall i, j \in \{1, \dots, n\}, \quad b_{ij} = \begin{cases} 1 & \text{si } j \in V_i \\ 0 & \text{sinon} \end{cases}$$

- $F = (f_i)_{i=1\dots n}$, où f_i est la fonction booléenne associée à la i -ième colonne de la matrice S . Le réseau R ainsi construit est canonique et consistant avec la table T .

Le réseau R est par construction consistant avec la table T . Il nous reste donc à démontrer qu'il est bien canonique, c'est-à-dire que la connectivité apparente des nœuds est bien égale à leur connectivité réelle. Nous allons démontrer ceci en procédant par induction sur l'entier k (indice de la première boucle, cf. alg. 1). Pour $1 \leq k \leq n$, nous appelons $I(k)$ l'ensemble des indices de $\{1, \dots, n\}$ pour lesquels l'algorithme passe par le <Nœud 1> (voir alg. 1). Si pour un automate i l'algorithme passe par le <Nœud 1>, alors i est enlevé de l'ensemble I des automates à traiter ; ceci nous assure que les ensembles $I(k) \subset \{1, \dots, n\}$ sont deux à deux disjoints. Pour tout $\xi \in I(k)$, nous notons V_ξ le k -uplet ordonné renvoyé par l'algorithme. Nous allons démontrer le lemme suivant :

Lemme 4.5 *Pour tout $k \in \{1, \dots, n\}$, pour tout $\xi \in I(k)$, la fonction f_ξ est de connectivité réelle k et V_ξ est égal au support de f_ξ . Réciproquement, si pour un automate $\xi \in \{1, \dots, n\}$ donné on a $\kappa(f_\xi) = k$, alors $\xi \in I(k)$.*

Démonstration

Montrons tout d'abord que ceci est vrai pour $k = 1$.

Soit $\xi \in I(1)$. L'ensemble V_ξ se réduit à un singleton $\{v\}$. D'après l'algorithme 1, nous savons que $H(S_\xi, E_v) = H(E_v)$, ce qui en utilisant la proposition 4.3 implique que la fonction f_ξ ne dépend pas des variables x_j , pour $j \neq v$. Comme nous avons de plus supposé que f_ξ n'est pas une fonction constante, nous en déduisons que $\kappa(f_\xi) = 1$ et donc que x_v est variable effective de f_ξ , c'est-à-dire :

$$\text{Supp}(f_\xi) = \{v\} = V_\xi$$

Réciproquement, considérons $\xi \in \{1, \dots, n\}$ tel que la fonction f_ξ est de connectivité réelle 1, alors le support de f_ξ se réduit à un singleton $\{v\}$, où $v \in \{1, \dots, n\}$. Dans ce cas, la proposition 4.3 nous assure que $H(E_v, S_\xi) = H(E_v)$, ce qui entraîne $\xi \in I(1)$.

Nous supposons à présent que la propriété est vérifiée pour $1 \leq k' \leq k - 1$, et nous montrons qu'elle est alors vérifiée au rang k .

Soit $\xi \in I(k)$, et $V_\xi = \{v_1, \dots, v_k\}$. Par hypothèse, nous savons que $\kappa(f_\xi) \geq k$, car sinon on aurait $\xi \in I(k')$ avec $k' < k$ (l'algorithme aurait déjà détecté l'automate ξ à une étape antérieure).

Puisque nous avons l'égalité $H(S_\xi, E_{v_1}, \dots, E_{v_m}) = H(E_{v_1}, \dots, E_{v_m})$, nous en déduisons grâce à la proposition 4.3 que f_ξ ne dépend pas des variables x_j , pour $j \notin V_\xi$. Ceci entraîne l'inégalité : $\kappa(f_\xi) \leq k$. On en déduit que $\kappa(f_\xi) = k$ et que $V_\xi = \text{Supp}(f_\xi)$. Réciproquement, si f_ξ est de connectivité réelle k , alors si nous notons v_1, \dots, v_k les éléments du support de f_ξ , nous avons l'égalité $H(S_\xi, E_{v_1}, \dots, E_{v_k}) = H(E_{v_1}, \dots, E_{v_k})$, qui implique que ξ appartient bien à $I(k)$. ■

Grâce à ce lemme, nous déduisons que l'algorithme 1 trouve, pour tout automate i , la connectivité réelle $\kappa(f_i)$ et l'ensemble $(x_{i_1}, \dots, x_{i_{\kappa(f_i)}})$ des variables effectives de f_i . De cette manière, nous sommes assurés que le réseau R défini dans la proposition 4.4 est canonique. En outre, comme les fonctions f_i sont des fonctions de $\{0, 1\}^n$ dans $\{0, 1\}$, leur connectivité réelle est au plus n , et donc l'algorithme trouve en temps fini l'ensemble des variables effectives de toutes les fonctions f_i .

Exemple 4.3

Reprenons la table de vérité T donnée dans l'exemple précédent :

E_1	E_2	E_3	S_1	S_2	S_3
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	1	0	0
0	1	1	0	0	1
1	0	0	1	0	1
1	0	1	0	1	1
1	1	0	1	0	0
1	1	1	0	1	1

Le déroulement de l'algorithme nous donne :

- Etape 1 : $I(1) = \{1\}$ et $V_1 = \{3\}$. La seule fonction de connectivité réelle 1 est donc la fonction f_1 . Sa variable effective est x_3 . En examinant la table de vérité, nous trouvons :

$$f_1(x_1, x_2, x_3) = f_1(x_3) = \neg x_3$$

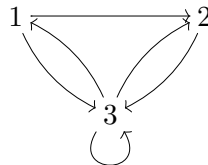
- Etape 2 : $I(2) = \{2\}$ et $V_2 = \{1, 3\}$. La seule fonction de connectivité réelle 2 est donc la fonction f_2 :

$$f_2(x_1, x_2, x_3) = f_2(x_1, x_3) = x_1 \wedge x_3$$

- Etape 3 : $I(3) = \{3\}$ et $V_3 = \{1, 2, 3\}$. La seule fonction de connectivité réelle 3 est donc la fonction f_3 :

$$f_3(x_1, x_2, x_3) = x_3 + x_1 \neg x_2$$

Nous en déduisons que l'unique réseau canonique consistant avec la table T est le réseau $R = (X, B, F)$ avec $X = \{1, \dots, n\}$ dont les connexions sont représentées par la graphe suivant :



qui correspond à la matrice d'incidence :

$$B = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

L'algorithme REVEAL nous fournit une preuve de l'existence, pour toute table de vérité complète T , d'un réseau canonique consistant avec T . Ceci achève donc la démonstration du théorème 4.1.

4.2.3 Discussion sur la complexité

Nous montrons dans cette partie que l'algorithme 1 présenté précédemment a, dans le pire des cas, une complexité exponentielle en n , la dimension du réseau.

Pour cela, calculons tout d'abord le nombre d'itérations pour le calcul d'une entropie. Soit q vecteurs booléens v^1, \dots, v^q de taille 2^n . On note $\alpha(n, q)$ le nombre d'opérations élémentaires nécessitées pour calculer la quantité $H(v^1, \dots, v^q)$. On appelle V la matrice ayant 2^n lignes et q colonnes dont les colonnes sont les vecteurs v^j , $j \in \{1, \dots, q\}$. Nous notons L_1, \dots, L_{2^n} les vecteurs formés par les lignes de V . Pour calculer la quantité $H(v^1, \dots, v^q)$, il faut avant tout calculer, pour chaque $i \in \{1, \dots, 2^n\}$, le nombre de lignes de V qui sont égales à L_i . Dans le pire cas, toutes les lignes de V sont distinctes, ce qui nécessite, $O(q2^n)$ opérations. Le calcul de l'entropie se fait ensuite en p opérations, où p est le nombre de lignes distinctes de V , d'où, dans le pire des cas :

$$\alpha(n, q) = O_{n \rightarrow \infty}(q2^n)$$

Considérons à présent une table de vérité complète de dimension n . Nommons $\beta(n)$ le nombre d'opérations nécessitant le déroulement de l'algorithme 1 dans le pire cas. A $k \in \{1, \dots, n\}$ fixé, l'algorithme doit, pour trouver les variables effectives des fonctions de connectivité réelle k , examiner tous les k -uplets ordonnés possibles de $\{1, \dots, n\}$. Il y en a C_n^k . Pour chaque k -uplet (i_1, \dots, i_k) et pour chaque $i \in \{1, \dots, n\}$, l'algorithme teste l'égalité $H(S_i, E_{i_1}, \dots, E_{i_k}) = H(E_{i_1}, \dots, E_{i_k})$, ce qui nécessite, lorsque n tend vers $+\infty$, $O(k2^n)$ opérations.

Nous en déduisons :

$$\beta(n) = O_{n \rightarrow \infty} \left(\sum_{k=1}^n C_n^k n k 2^n \right) = O_{n \rightarrow \infty} \left(n 2^n \sum_{k=1}^n k C_n^k \right) \quad (4.5)$$

Le terme 2^n provient du fait que la table de vérité donnée en entrée est complète et contient donc 2^n lignes. En ce qui concerne la somme $s(n) = \sum_{k=1}^n k C_n^k$, il est facile de vérifier que :

$$\forall n \in \mathbb{N}^*, \quad s(n) = n 2^{n-1}$$

La complexité de l'algorithme 1 est donc, dans le pire cas :

$$\beta(n) = O_{n \rightarrow \infty} \left(\frac{1}{2} n^2 2^{2n} \right)$$

Dans cette expression, le terme exponentiel 2^{2n} provient à la fois du fait que la table $T = (E, S)$ donnée en entrée est complète (i.e. les matrices E et S sont de dimension $2^n \times n$) et que l'algorithme recherche les fonctions de connectivité réelle k , pour tout $k \in \{1, \dots, n\}$. Afin de réduire

la complexité de l'algorithme, nous pouvons donc agir de deux manières :

- Soit nous décidons de ne donner en entrée qu'une partie de la table de vérité (il s'agit alors du problème d'inférence partielle, qui est traitée dans la suite de ce chapitre),
- Soit nous ne recherchons que les nœuds dont la connectivité réelle reste bornée par un entier $K \in \mathbb{N}^*$ donné à l'avance.

Nous inspirant du deuxième point, nous proposons une autre version de l'algorithme REVEAL dans laquelle nous cherchons uniquement les fonctions dont la connectivité est bornée (voir alg. 2 ci-après).

Cet algorithme nous renvoie un ensemble d'indices $I_2 \subset \{1, \dots, n\}$ qui contient l'ensemble des indices des fonctions dont la connectivité réelle est inférieure ou égale à K , où K est une constante fixée à l'avance. Pour chacun de ces indices $i \in I_2$, il renvoie ensuite le k -uplet ordonné des variables effectives de la fonction f_i . Dans le calcul de la complexité de cet algorithme, nous sommes amenés à remplacer dans (4.5) le terme $s(n) = \sum_{k=1}^n k C_n^k$ par la somme partielle $\sum_{k=1}^K k C_n^k$.

Proposition 4.6 *Pour tout $n \in \mathbb{N}^*$ et pour tout $1 \leq K \leq n$, on a :*

$$\sum_{k=1}^K k C_n^k \leq K^2 n^K$$

Démonstration

Soient $n \in \mathbb{N}^*$ et $1 \leq K \leq n$. Posons $s(n, K) = \sum_{k=1}^K k C_n^k$. En majorant k par K dans cette somme, on obtient l'inégalité :

$$s(n, K) \leq K \sum_{k=1}^K C_n^k \tag{4.6}$$

Par définition, $C_n^k = \frac{n!}{k!(n-k)!}$, donc, puisque $1 \leq k \leq K$, nous en déduisons :

$$C_n^k \leq \frac{n!}{(n-K)!} = n(n-1) \dots (n-K+1) \leq n^K \tag{4.7}$$

De (4.6) et de (4.7), nous déduisons la majoration voulue. ■

Ainsi, en fixant la connectivité maximale désirée K , le terme exponentiel $s(n)$ a été remplacé dans le calcul de la complexité $\beta(n)$ de REVEAL par le terme polynômial $s(n, K)$. La complexité de l'algorithme 2 est donc, dans le pire cas, en $O(K^2 n^{K+1} 2^n)$, lorsque K est fixé et n tend vers $+\infty$. Nous avons ainsi réduit l'ordre de la complexité du premier algorithme, toutefois celle-ci reste encore exponentielle en fonction de n , la dimension du réseau. Nous voyons dans ce qui suit comment pallier à cette difficulté en examinant le problème de l'inférence partielle.

Algorithme 2 - REVEAL2 - Identification de réseaux booléens de connectivité bornée.

Etant donnée une table de vérité complète de dimension n et un entier K , cet algorithme trouve les variables effectives des fonctions f_i dont la connectivité réelle $\kappa(f_i)$ est \leq à K .

n : entier ≥ 1 ;

K : entier $\leq n$;

E, S : Matrices booléennes de dimension $2^n \times n$;

[Les matrices E et S représentent une table de vérité complète de dimension n . On suppose que les fonctions f_i définies par les colonnes S_i ne sont pas constantes sur $\{0, 1\}^n$.]

$I_1 \leftarrow \{1, \dots, n\}$;

$I_2 \leftarrow \emptyset$;

[I_1 est l'ensemble des indices à traiter, I_2 est l'ensemble des nœuds identifiés.]

Pour k de 1 à K faire

[On cherche les fonctions de connectivité réelle k .]

Pour $i \in I_1$ faire

Pour (i_1, \dots, i_k) un k -uplet ordonné de $\{1, \dots, n\}$ faire

Si $(H(S_i, E_{i_1}, \dots, E_{i_k}) = H(E_{i_1}, \dots, E_{i_k}))$ **Alors**

[f_i est de connectivité k]

<Nœud 1>

$V_i \leftarrow i_1, \dots, i_k$;

[Les variables de la fonction f_i sont désormais trouvées : on l'enlève de I_1 et on l'ajoute à I_2 .]

$I_1 \leftarrow I_1 \setminus \{i\}$;

$I_2 \leftarrow I_2 \cup \{i\}$;

Break ;

Fin Si

Fin Pour

Fin Pour

Retourner $I_2, V = (V_i)_{i \in I_2}$;

Fin Pour

4.3 Le problème de l'identification partielle

Dans la partie précédente, nous avons démontré que la donnée d'une table de vérité complète permet d'identifier un réseau booléen canonique de manière unique. Comme nous l'avons vu, une table de vérité complète pour un réseau de dimension n est donnée par deux matrices booléennes de taille $2^n \times n$. Le problème de l'identification partielle consiste à se donner uniquement une partie de la table de vérité (en diminuant le nombre de lignes des matrices E et S) et de trouver s'il existe des réseaux consistants avec ces données.

4.3.1 Présentation du problème

Nous allons tout d'abord définir ce qu'est une table de vérité partielle. Pour cela, nous définissons l'ensemble suivant :

Définition 4.6 Soit $n \in \mathbb{N}^*$. Pour tout $p \in \{1, \dots, 2^n\}$, nous nommons $\mathcal{E}_{n,p}$ l'ensemble des matrices booléennes E à p lignes et n colonnes telles que les p lignes de E sont deux à deux distinctes.

Pour tout $p \in \mathbb{N}^*$, nous notons \mathcal{P}_p l'ensemble des matrices de permutation de taille p . Nous considérons alors sur $\mathcal{E}_{n,p}$ la relation binaire \equiv définie par :

$$\forall E_1, E_2 \in \mathcal{E}_{n,p}, \quad E_1 \equiv E_2 \Leftrightarrow (\exists P \in \mathcal{P}_p, E_1 = PE_2)$$

En d'autres termes, $E_1 \equiv E_2$ si et seulement si elles sont égales à une permutation des lignes près. La relation \equiv est clairement une relation d'équivalence. Etant donnée $E \in \mathcal{E}_{n,p}$, on notera \dot{E} la classe d'équivalence de E par la relation \equiv ainsi définie.

Définition 4.7 Soit $n \in \mathbb{N}^*$ et p un entier compris entre 1 et 2^n . Une table de vérité partielle est un couple $T = (E, S)$, où E appartient à $\mathcal{E}_{n,p}$ et S est une matrice booléenne quelconque à p lignes et n colonnes. n et p sont respectivement appelés dimension et taille de la table T .

Contrairement à une table complète, la donnée d'une table de vérité partielle ne détermine pas une unique application F de $\{0, 1\}^n$ dans lui-même. Il nous faut alors définir une notion de consistance.

Rappelons ici que nous notons \mathcal{F}_n l'ensemble $(\{0, 1\}^n)^{\{0, 1\}^n}$ des applications de $\{0, 1\}^n$ dans lui-même.

Définition 4.8 Considérons une application $F \in \mathcal{F}_n$ et une table de vérité partielle $T = (E, S)$ de dimension n et de taille p . On dira que F est consistante avec T si :

$$\forall i \in \{1, \dots, p\}, \quad L_i(S) = F(L_i(E))$$

où $L_i(E)$ et $L_i(S)$ désignent respectivement les i -ièmes lignes de la matrice E et de la matrice S .

Exemple 4.4

Posons $n = 3$ et $p = 6$. Nous définissons deux matrices E et S par :

$$E = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}, \quad S = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}$$

Nous vérifions aisément que la matrice E appartient à l'ensemble $\mathcal{E}_{3,6}$. (E, S) est donc une table de vérité partielle de dimension 3 et de taille 6. Nous la notons T .

Nous considérons à présent les deux applications $F, F' \in \mathcal{F}_3$ définies par :

$$\left\{ \begin{array}{l} f_1(x_1, x_2, x_3) = x_1 \vee x_3 \\ f_2(x_1, x_2, x_3) = x_1 \wedge x_3 \\ f_3(x_1, x_2, x_3) = x_1 \wedge \neg x_2 \end{array} \right. , \quad \left\{ \begin{array}{l} f'_1(x_1, x_2, x_3) = (x_1 \wedge x_2) \vee x_3 \\ f'_2(x_1, x_2, x_3) = x_1 \wedge x_2 \wedge x_3 \\ f'_3(x_1, x_2, x_3) = 0 \end{array} \right.$$

Leurs tables de vérités complètes sont données ci-dessous :

•	0	0	0	0	0	0
•	0	0	1	1	0	0
•	0	1	0	0	0	0
•	0	1	1	1	0	0
	1	0	0	1	0	1
	1	0	1	1	1	1
•	1	1	0	1	0	0
•	1	1	1	1	1	0
	entrées			sorties		

•	0	0	0	0	0	0
•	0	0	1	1	0	0
•	0	1	0	0	0	0
•	0	1	1	1	0	0
	1	0	0	0	0	0
	1	0	1	1	0	0
•	1	1	0	1	0	0
•	1	1	1	1	1	0
	entrées			sorties		

FIG. 4.3: Tables de vérité complètes de F (à gauche) et de F' (à droite). Les lignes marquées d'un • désignent les lignes de E (voir texte).

Dans ces tables, nous avons mis en évidence les lignes qui correspondent aux lignes de la matrice E . Nous vérifions d'une part que $F \neq F'$, et d'autre part que pour toutes les lignes i marquées d'une • on a :

$$F(L_i(E)) = F'(L_i(E)) = L_i(S)$$

nous en déduisons donc les applications F et F' sont toutes les deux consistantes avec la table T définie plus haut.

A partir de cette notion de consistance, nous pouvons définir la relation binaire suivante.

Définition 4.9 Soit $E \in \mathcal{E}_{n,p}$. On dira que $F_1, F_2 \in \mathcal{F}_n$ coïncident sur E (noté $F_1 \sim_E F_2$) s'il existe une matrice booléenne S de dimension $p \times n$ telle que F_1 et F_2 sont consistantes avec la table (E, S) .

Nous démontrons alors le résultat suivant.

Proposition 4.7 Etant donnée une matrice E appartenant à $\mathcal{E}_{n,p}$, nous notons \dot{E} la classe d'équivalence de E selon la relation \equiv définie plus haut. Nous avons alors les résultats suivants :

(i) Pour toute matrice $E' \in \dot{E}$,

$$\forall F_1, F_2 \in \mathcal{F}_n, \quad F_1 \sim_E F_2 \iff F_1 \sim_{E'} F_2$$

La relation \sim_E est donc égale à $\sim_{E'}$ pour tout $E' \in \dot{E}$. On pourra noter $\sim_{\dot{E}}$ au lieu de \sim_E .

(ii) la relation \sim_E est une relation d'équivalence.

Démonstration

Soit $E \in \mathcal{E}_{n,p}$ et $E' \in \dot{E}$, i.e. $E' = PE$, avec $P \in \mathcal{P}_p$. Montrons tout d'abord le point (i). Considérons deux éléments F_1 et F_2 de \mathcal{F}_n tels que $F_1 \sim_E F_2$. Par définition, il existe une matrice S telle que F_1 et F_2 sont consistants avec (E, S) . Posons $S' = PS$. Puisque F_1 et F_2 sont consistantes avec (E, S) , nous avons :

$$\forall i \in \{1, \dots, p\}, \quad L_i(S) = F_1(L_i(E)) \text{ et } L_i(S) = F_2(L_i(E))$$

La matrice P étant une matrice de permutation, les lignes de E (resp. S) sont les mêmes que les lignes de $E' = PE$ (resp. $S' = PS$) à une permutation près. Nous en déduisons donc :

$$\forall i \in \{1, \dots, p\}, \quad L_i(S') = F_1(L_i(E')) \text{ et } L_i(S') = F_2(L_i(E'))$$

ce qui implique que F_1 et F_2 sont consistantes avec (E', S') , et donc que $F_1 \sim_{E'} F_2$.

En supposant cette fois que $F_1 \sim_{E'} F_2$, nous montrons qu'alors $F_1 \sim_E F_2$ en remarquant simplement que $E = P^{-1}E'$, $S = P^{-1}S'$ et $P^{-1} \in \mathcal{P}_p$.

Montrons à présent le point (ii). D'après la définition 4.9, la relation \sim_E est clairement réflexive et symétrique. Afin de montrer la transitivité de \sim_E , considérons F_1, F_2, F_3 trois éléments de \mathcal{F}_n tels que $F_1 \sim_E F_2$ et $F_2 \sim_E F_3$. Il existe alors deux matrices booléennes S et S' de taille $p \times n$ telles que :

$$\begin{cases} \forall i \in \{1, \dots, p\}, & L_i(S) = F_1(L_i(E)) = F_2(L_i(E)) \\ \forall i \in \{1, \dots, p\}, & L_i(S') = F_2(L_i(E)) = F_3(L_i(E)) \end{cases}$$

Nous en déduisons que :

$$\forall i \in \{1, \dots, p\}, \quad L_i(S') = F_2(L_i(E)) = L_i(S)$$

et donc que $S = S'$. D'autre part, on a :

$$\forall i \in \{1, \dots, p\}, \quad L_i(S) = F_1(L_i(E)) = F_3(L_i(E))$$

ce qui entraîne $F_1 \sim_E F_3$. Ceci prouve la transitivité de \sim_E . ■

La relation $\sim_{\dot{E}}$ étant une relation d'équivalence, nous pouvons décomposer \mathcal{F}_n en classes d'équivalence. Etant donnée une application $F \in \mathcal{F}_n$, nous noterons \tilde{F}^E (ou $\tilde{F}^{\dot{E}}$) la classe d'équivalence de F . L'ensemble \tilde{F}^E est donc un élément de l'ensemble quotient $\mathcal{F}_n / \sim_{\dot{E}}$: il s'agit de l'ensemble des fonctions coïncidant avec F sur les entrées E . Pour tout $i \in \{1, \dots, n\}$, nous noterons $\tilde{f}_i^{\dot{E}}$ l'ensemble des i -ièmes composantes des fonctions appartenant à $\tilde{F}^{\dot{E}}$. Si nous reprenons l'exemple précédent, nous voyons que les applications F et F' de \mathcal{F}_3 coïncident sur E , c'est-à-dire que :

$$F \sim_E F'$$

Nous avons donc : $F' \in \tilde{F}^{\dot{E}}$.

Remarque 4.3 *Par soucis de concision, nous dirons par la suite qu'une application*

$$f_i : \{0, 1\}^n \rightarrow \{0, 1\}$$

est consistante avec une table T à la colonne i s'il existe une application $F \in \mathcal{F}_n$ consistante avec T admettant f_i comme i -ième coordonnée. Par abus de langage, s'il n'y a pas d'ambiguïtés, nous dirons tout simplement que f_i est consistante avec T .

Grâce à la notion de table de vérité partielle, nous pouvons à présent définir le problème d'identification partielle de réseau.

Définition 4.10 *Posons $n \in \mathbb{N}^*$ et $1 \leq p \leq 2^n$. Nous considérons une matrice $E \in \mathcal{E}_{n,p}$ et une table de vérité partielle $T = (E, S)$ de taille p . Soit un réseau booléen $R = (X, B, F)$ de dimension n . On dira que R est consistant avec T si l'application F est consistante avec T .*

Par définition, si le réseau R est consistant avec T , alors tous les réseaux $R' = (X, B', F')$ où F' appartient à \tilde{F}^E (la classe d'équivalence de F par \sim_E) sont également consistants avec T . Reprenons l'exemple précédent. Les réseaux $R = (X, B, F)$ et $R' = (X, B', F')$ définis par :

$$B = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \quad F(x_1, x_2, x_3) = \begin{pmatrix} x_1 \vee x_3 \\ x_1 \wedge x_3 \\ x_1 \wedge \neg x_2 \end{pmatrix}$$

$$B' = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix} \quad F'(x_1, x_2, x_3) = \begin{pmatrix} (x_1 \wedge x_2) \vee x_3 \\ x_1 \wedge x_2 \wedge x_3 \\ 0 \end{pmatrix}$$

sont tous les deux consistants avec la table T donnée. On peut vérifier facilement qu'ils sont tous les deux canoniques. Dans le cas de l'identification partielle, nous ne pouvons donc plus affirmer l'unicité d'un réseau canonique consistant avec la solution.

Nous allons par la suite utiliser une notion plus fine que la notion de canonicité.

Définition 4.11 *Soit $n \in \mathbb{N}^*$ et $F \in \mathcal{F}_n$. Le support de F est le n -uplet $\text{Supp}(F) = (I_1, \dots, I_n)$ appartenant à l'ensemble $\mathfrak{P}(\{1, \dots, n\})^n$, dans lequel, pour tout $i \in \{1, \dots, n\}$, $I_i = \text{Supp}(f_i)$. Nous appellerons dimension de ce support et nous noterons $D(F)$ le n -uplet (k_1, \dots, k_n) appartenant à $\{1, \dots, n\}^n$ où, pour chaque $i \in \{1, \dots, n\}$, $k_i = \kappa(f_i) = \text{card}(I_i)$.*

Cette définition nous permet de déterminer un critère de minimalité pour les réseaux d'automates booléens. Pour cela, nous allons utiliser l'ordre partiel suivant :

Etant donnés deux n -uplets $K = (k_1, \dots, k_n)$ et $K' = (k'_1, \dots, k'_n)$ appartenant à \mathbb{N}^n , on dira que K est inférieur ou égal à K' (que nous notons par $K \preceq K'$) si et seulement si :

$$\forall i \in \{1, \dots, n\}, \quad k_i \leq k'_i$$

On vérifie facilement que \preceq est réflexive, transitive et antisymétrique, c'est donc une relation d'ordre. On vérifie également qu'elle ne définit pas un ordre total, puisque par exemple pour $n = 3$, les triplets $(1, 2, 3)$ et $(2, 1, 1)$ ne sont pas comparables.

Définition 4.12 *Soient $n \in \mathbb{N}^*$ et $1 \leq p \leq 2^n$. Nous considérons une matrice $E \in \mathcal{E}_{n,p}$ et une table de vérité partielle $T = (E, S)$ de taille p . Un réseau $R = (X, B, F)$ est dit minimal par rapport à T s'il vérifie les deux conditions suivantes :*

- (i) R est canonique et consistant avec T ,
- (ii) Si F' est un élément quelconque de \tilde{F}^E , alors $D(F') \succeq D(F)$.

On notera $\mathcal{M}(T)$ l'ensemble des réseaux minimaux par rapport à T .

Remarque 4.4 La condition (ii) est équivalente à la condition suivante :

$$(ii') \quad \forall i \in \{1, \dots, n\}, \quad \kappa(f_i) = \min_{F' \in \tilde{F}^E} \{\kappa(f'_i)\}$$

C'est-à-dire que dans un réseau minimal R , les supports des fonctions f_i sont minimaux. En d'autres termes, en ôtant une connexion dans le graphe de connexions de R , nous perdons la consistance de R avec les données (c'est-à-dire avec la table partielle T).

Parmi les réseaux consistants avec une table T , les réseaux minimaux sont donc ceux dont le graphe de connexion possède le nombre minimum d'arêtes. Afin d'illustrer cette définition, considérons la table T donnée dans l'exemple précédent.

Exemple 4.4 (suite)

soit la table $T = (E, S)$ donnée précédemment :

$$E = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}, \quad S = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}$$

Nous avons donné deux réseaux $R = (X, B, F)$ et $R' = (X, B', F')$ canoniques et consistants avec T . Regardons à présent s'ils sont minimaux.

Considérons tout d'abord le premier nœud (première colonne de S). Les fonctions f_1 et f'_1 sont :

$$\begin{aligned} f_1(x_1, x_3) &= x_1 \vee x_3 \\ f'_1(x_1, x_2, x_3) &= (x_1 \wedge x_2) \vee x_3 \end{aligned}$$

Nous avons donc $\kappa(f_1) = 2 < \kappa(f'_1) = 3$. Il est facile de vérifier que la connectivité réelle minimale de ce nœud est 2. Pour cela, il nous suffit en effet de vérifier qu'il n'existe aucune fonction g_1 , pour tout $G \in \tilde{F}^E$, qui soit de connectivité réelle 0 ou 1.

Considérons à présent le second nœud :

$$\begin{aligned} f_2(x_1, x_3) &= x_1 \wedge x_3 \\ f'_2(x_1, x_2, x_3) &= x_1 \wedge x_2 \wedge x_3 \end{aligned}$$

D'où $\kappa(f_2) = 2 < \kappa(f'_2) = 3$. Comme dans le cas précédent, on vérifie aisément que 2 est la connectivité minimale du nœud 2.

Enfin, pour le troisième nœud :

$$\begin{aligned} f_3(x_1, x_3) &= x_1 \wedge \neg x_2 \\ f'_3(x_1, x_2, x_3) &= 0 \end{aligned}$$

D'où $\kappa(f_3) = 2 > \kappa(f'_3) = 0$. Ici, il est clair que 0 est la connectivité minimale du nœud 3.

Nous avons donc montré que :

$$\begin{cases} D(F) &= (2, 2, 2) \\ D(F') &= (3, 3, 0) \end{cases}$$

Or, un réseau minimal par rapport à T a une dimension égale à $(2, 2, 0)$. Aucun des deux réseaux R et R' n'est donc minimal par rapport à T . En revanche, nous pouvons construire le réseau minimal $R'' = (X, B'', F'')$ défini par :

$$\begin{aligned} f_1''(x_1, x_3) &= x_1 \vee x_3 \\ f_2''(x_1, x_3) &= x_1 \wedge x_3 \\ f_3'' &= 0 \end{aligned}$$

Nous pouvons vérifier dans ce cas que ce réseau est le seul réseau minimal par rapport à T , c'est-à-dire que l'ensemble $\mathcal{M}(T)$ se réduit au singleton $\{R''\}$.

Si dans cet exemple nous avons trouvé un réseau minimal unique, ce n'est malheureusement pas toujours le cas. En effet, si nous considérons la matrice d'entrées E et le vecteur de sorties S suivants :

$$E = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}, \quad S = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

(dans cet exemple, nous n'avons donné qu'une seule colonne pour S , nous ne cherchons à identifier qu'un seul nœud). Il est clair qu'une fonction $f : \{0, 1\}^6 \rightarrow \{0, 1\}$ consistante avec cette table partielle ne peut être de connectivité réelle 0 ou 1. En revanche, nous trouvons deux fonctions f et g consistantes telles que $\kappa(f) = \kappa(g) = 2$:

$$\begin{cases} f(x_1, x_2) &= x_1 \text{ xor } x_2 \\ f(x_5, x_6) &= \neg(x_5 \text{ xor } x_6) \end{cases}$$

Il existe donc au moins deux fonctions consistantes avec ces données dont la dimension du support est minimale et vaut 2.

Nous sommes à présent en mesure d'énoncer plusieurs problèmes relatifs à l'inférence partielle de réseaux.

Nous considérons, pour $n \in \mathbb{N}^*$ et $1 \leq p \leq 2^n$, une table de vérité partielle $T = (E, S)$ de dimension n et de taille p .

Problème 1 (CONSISTANCE MINIMALE)

Construire un réseau d'automates booléens canonique minimal consistant avec la table T .

Problème 2 (ENUMÉRATION)

Enumérer les réseaux canoniques minimaux consistants avec T .

Problème 3 (CONSISTANCE MINIMALE À SUPPORT FIXÉ)

En fixant $\mathcal{S} \in \mathfrak{P}(\{1, \dots, n\})^n$, décider s'il existe un réseau d'automates booléens canonique minimal de support \mathcal{S} consistant avec la table T . Donner une méthode de construction d'un tel réseau et décider de son unicité.

Nous allons dans la suite proposer un algorithme permettant d'apporter une réponse à ces problèmes. Nous pouvons d'ores-et-déjà remarquer que si nous prouvons l'existence d'une solution pour le problème 3, nous avons également résolu le problème 1. En ce qui concerne le problème 2, nous avons montré par un contre-exemple qu'on ne peut, dans le cas général,

assurer l'unicité d'un réseau minimal. A notre connaissance, aucune hypothèse sur T assurant cette unicité n'a été trouvée. Nous nous contentons ici de proposer une méthode algorithmique permettant d'énumérer, pour T donnée, l'ensemble $\mathcal{M}(T)$ des réseaux minimaux consistants avec T . Enfin, nous nous intéressons dans la suite au problème 3 et en particulier à la question de l'unicité d'un réseau minimal à support fixé. Nous proposons notamment un résultat reliant l'unicité d'un tel réseau à p , la taille de T .

4.3.2 Algorithme d'identification partielle

Nous considérons deux entiers $n \in \mathbb{N}^*$ et $p \in \{1, \dots, 2^n\}$ et une table de vérité partielle $T = (E, S)$ de dimension n et de taille p . Nous recherchons les réseaux $R = (X, B, F)$ minimaux par rapport à T .

L'algorithme que nous proposons est une généralisation de l'algorithme 1. La résolution des problèmes 1, 2 et 3 se sépare en plusieurs étapes distinctes. Tout d'abord, il nous faut trouver la dimension minimale $d \in \mathbb{N}^n$ du support des fonctions F consistantes avec la table T . Une fois cette dimension trouvée, nous énumérons tous les supports \mathcal{S} de dimension d et, pour chacun d'entre eux, nous construisons une application $F \in \mathcal{F}_n$ de support \mathcal{S} consistante avec T . Puis nous déterminons si une telle application est déterminée de façon unique et, dans le cas contraire, nous comptons le nombre de ces applications.

Comme dans le cas de l'identification par rapport à une table complète (cf. partie 4.2), nous allons utiliser la notion d'entropie de Shannon. Avant de décrire l'algorithme, nous allons donner un résultat analogue à la proposition 4.3 dans le cas d'une table de vérité partielle. Avant cela, examinons l'exemple suivant.

Exemple 4.5

Considérons l'extrait de table de vérité partielle ci-dessous ($n = 8$, $p = 4$) :

$$E = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}, \quad S = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} \quad (4.8)$$

Nous cherchons les fonctions $f : \{0, 1\}^8 \rightarrow \{0, 1\}$ consistantes avec cette table. Parmi elles, nous vérifions facilement qu'il existe une fonction de support $\{3, 6\}$. Afin d'exprimer cette fonction, nous extrayons de (E, S) sa table de vérité. Nous trouvons :

x_3	x_6	f
0	0	1
0	1	1
1	0	0
1	1	1

La fonction f est donc entièrement déterminée, et elle vaut :

$$f(x_3, x_6) = x_6 \vee \neg x_3$$

En revanche, si nous ôtons à E et à S la troisième ligne, alors nous ne pouvons plus déterminer f de façon unique. E et S sont alors égales à :

$$E = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}, \quad S = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} \quad (4.9)$$

Après extraction, la table de vérité de f sur $\{0,1\}^2$ est incomplète :

x_3	x_6	f
0	0	1
0	1	×
1	0	0
1	1	1

En choisissant 0 ou 1 à la place du signe \times , nous obtenons alors deux fonctions f et g distinctes, de même support $\{3,6\}$, consistantes avec les données :

$$f(x_3, x_6) = x_6 \vee \neg x_3, \quad g(x_3, x_6) = (\neg x_6 \wedge \neg x_3) \vee (x_6 \wedge x_3)$$

Ceci vient du fait que parmi les lignes de E , aucune ne comporte en troisième et sixième position le patron 01.

Cet exemple montre un point déterminant dans la proposition qui va suivre. Avant de l'énoncer, nous allons définir la notation suivante :

Définition 4.13 Soient $n \in \mathbb{N}^*$, $p \in \{1, \dots, 2^n\}$, $E \in \mathcal{E}_{n,p}$ et $I = \{i_1, \dots, i_k\}$ une partie de $\{1, \dots, n\}$ de cardinal $k > 0$. Pour tout $x \in \{0,1\}^k$, nous notons $\Xi_I(x)$ l'ensemble des indices j des lignes L_j de E telles que :

$$\forall m \in \{1, \dots, k\} \quad (L_j)_{i_m} = x_m$$

Dans l'exemple précédent, nous avons, pour $I = \{3,6\}$ dans le premier cas (table 4.8) :

$$\Xi_I(0,0) = \{4\}, \quad \Xi_I(0,1) = \{3\}, \quad \Xi_I(1,0) = \{2\}, \quad \Xi_I(1,1) = \{1,5\}$$

alors que dans le deuxième cas (table 4.9) :

$$\Xi_I(0,0) = \{3\}, \quad \Xi_I(0,1) = \emptyset, \quad \Xi_I(1,0) = \{2\}, \quad \Xi_I(1,1) = \{1,4\}$$

Nous remarquons que les $\Xi_I(x)$ vérifient de manière évidente la propriété suivante, qui sera utile par la suite :

$$\forall x, x' \in \{0,1\}^k, \quad \Xi_I(x) \cap \Xi_I(x') = \emptyset \quad (4.10)$$

Grâce à la définition de ces ensembles $\Xi_I(x)$, nous pouvons énoncer la proposition suivante, analogue de la proposition 4.3 dans le cas d'une table de vérité partielle.

Proposition 4.8 Soient $n \in \mathbb{N}^*$ et $p \in \{1, \dots, 2^n\}$. $T = (E, S)$ est une table de vérité partielle de dimension n et de taille p . (les matrices E et S sont donc de dimension $p \times n$). Nous notons respectivement E_j et S_j , pour $j \in \{1, \dots, n\}$, la j -ième colonne de E et de S . Soient $i \in \{1, \dots, n\}$ et $I = \{i_1, \dots, i_k\}$ une partie de $\{1, \dots, n\}$ de cardinal k . Les deux assertions suivantes sont équivalentes :

(i) Il existe une fonction $f_i : \{0, 1\}^n \rightarrow \{0, 1\}$, consistante avec T à la colonne i dont le support est inclus dans I .

(ii) L'égalité suivante est vérifiée : $H(S_i, E_{i_1}, \dots, E_{i_k}) = H(E_{i_1}, \dots, E_{i_k})$

Nous notons N_n^k le cardinal de l'ensemble $\{x \in \{0, 1\}^k \mid \Xi_I(x) = \emptyset\}$. Alors, le nombre de fonctions f_i de support inclus dans I consistantes avec T est égal à $2^{N_n^k}$.

Démonstration

La démonstration de l'équivalence entre les assertions (i) et (ii) suit un raisonnement analogue à la démonstration de la proposition 4.3. Nous posons $\mathbb{E} = (E_{i_1}, \dots, E_{i_k})$ et $\mathbb{E}' = (S_i, E_{i_1}, \dots, E_{i_k})$. Nous rappelons que l'entropie $H(\mathbb{E}')$ se calcule par la formule :

$$H(\mathbb{E}') = - \sum_{x \in \{0, 1\}^k} P_{\mathbb{E}'}^{0,x} \log(P_{\mathbb{E}'}^{0,x}) - \sum_{x \in \{0, 1\}^k} P_{\mathbb{E}'}^{1,x} \log(P_{\mathbb{E}'}^{1,x})$$

où $P_{\mathbb{E}'}^{\epsilon,x} = \frac{1}{p} \text{card}(I_{\mathbb{E}'}^{\epsilon,x})$ (avec $\epsilon \in \{0, 1\}$), et $I_{\mathbb{E}'}^{\epsilon,x}$ est l'ensemble défini par :

$$I_{\mathbb{E}'}^{\epsilon,x} = \{j \in \{1, \dots, p\} \mid (S_i)_j = \epsilon, (E_{i_1})_j = x_1, \dots, (E_{i_k})_j = x_k\}$$

D'après la démonstration de la proposition 4.3, nous savons que (ii) est équivalente à :

(iii) Pour tout $x \in \{0, 1\}^k$, soit $P_{\mathbb{E}'}^{0,x} = 0$, soit $P_{\mathbb{E}'}^{1,x} = 0$.

Pour $x \in \{0, 1\}^k$ fixé, nous notons $\mathcal{Y}_I(x)$ l'ensemble des vecteurs $y \in \{0, 1\}^n$ tels que :

$$\forall j \in \{1, \dots, k\}, \quad y_{i_j} = x_j$$

Nous avons alors :

$$\{0, 1\}^n = \bigsqcup_{x \in \{0, 1\}^k} \mathcal{Y}(x)$$

(le signe \sqcup signifie qu'il s'agit d'une union disjointe). Par définition, si un vecteur $y \in \mathcal{Y}_I(x)$ est une ligne de E , alors son indice appartient à l'ensemble $\Xi_I(x)$.

L'assertion (iii) est équivalente au fait que, pour tout $x \in \{0, 1\}^k$, la fonction que nous cherchons à identifier est constante sur $\mathcal{Y}_I(x)$. En reprenant la démonstration de la proposition 4.3, il est clair que (i) implique (ii) et donc que (i) implique (iii). Montrons à présent que (iii) implique (i).

Pour cela, nous allons construire une fonction f_i de support inclus dans I qui est consistante avec T . Tout d'abord, étant donné que $p \geq 1$, il existe nécessairement un vecteur $x \in \{0, 1\}^k$ tel que $\Xi_I(x) \neq \emptyset$. L'assertion (iii) nous assure alors qu'il existe $\varepsilon(x) \in \{0, 1\}$ tel que :

$$\forall \xi \in \Xi_I(x), \quad (S_i)_\xi = \varepsilon(x)$$

Nous posons donc :

$$\forall y \in \mathcal{Y}_I(x), \quad f_i(y) = \varepsilon(x)$$

Nous procédons de la même manière pour tout $x \in \{0, 1\}^k$ tel que $\Xi_I(x) \neq \emptyset$. Nous notons \tilde{X} l'ensemble des vecteurs $x \in \{0, 1\}^k$ tels que $\Xi_I(x) \neq \emptyset$. Nous avons alors défini $f_i(y)$, pour

tout vecteur y appartenant à l'ensemble :

$$\mathcal{X} = \bigsqcup_{x \in \tilde{X}} \mathcal{Y}_I(x)$$

Premier cas : pour tout $x \in \{0, 1\}^k$, on a : $\Xi_I(x) \neq \emptyset$ (i.e. $\tilde{X} = \{0, 1\}^k$). Nous avons alors déterminé notre fonction f_i sur $\{0, 1\}^n$ tout entier. Elle est par construction consistante avec T et son support est inclus dans I . Dans ce cas, la fonction f_i est l'unique fonction consistante avec T dont le support est inclus dans I .

Deuxième cas : il existe des $x \in \{0, 1\}^k$ tels que $\Xi_I(x) = \emptyset$ (i.e. $\{0, 1\}^k \setminus \tilde{X} \neq \emptyset$). Il nous reste alors à déterminer les valeurs que prend f_i sur $\{0, 1\}^n \setminus \mathcal{X}$. Afin que le support de ces f_i reste inclus dans I , nous devons imposer f_i constante sur chaque $\mathcal{Y}_I(x)$, pour tout $x \in \{0, 1\}^k$. Pour chaque $x \in \{0, 1\}^k \setminus \tilde{X}$, nous avons donc le choix entre les deux valeurs 0 ou 1. N_n^k étant le cardinal de l'ensemble $\{0, 1\}^k \setminus \tilde{X}$, il y a en tout $2^{N_n^k}$ fonctions possibles. Par construction, les fonctions ainsi construites sont consistantes avec T et leurs supports sont inclus dans I . ■

En nous inspirant de l'algorithme 1, nous donnons ci-après (voir alg. 3) une méthode algorithmique permettant l'identification de réseaux minimaux à partir d'une table de vérité partielle.

Le principe de cet algorithme est le suivant : nous disposons d'une table de vérité partielle $T = (E, S)$ de dimension $n \in \mathbb{N}^*$ et de taille $p \in \{1, \dots, 2^n\}$. Le but est de trouver, pour tout automate $i \in \{1, \dots, n\}$, la dimension minimale $d(i)$ du support de cet automate. En d'autres termes, nous cherchons parmi les fonctions f_i consistantes avec T , celles dont le support est minimal. La dimension de ce support est donc un entier $d(i) \in \{0, \dots, n\}$. Par souci de commodité d'écriture, nous avons supposé dans l'algorithme 3 que S ne contient aucune colonne égale à $(0, \dots, 0)^t$ ou $(1, \dots, 1)^t$, ce qui impose que $d(i)$ est strictement positif. Il s'agit là d'une hypothèse de travail qui ne nuit en rien à la généralité de l'algorithme : il suffit de tester au départ si S contient une colonne S_i égale à $(\epsilon, \dots, \epsilon)^t$ (ϵ appartenant à $\{0, 1\}$); alors pour de telles i , $k(i)$ vaut zéro et i est ôté de l'ensemble J . Nous supposons donc dans ce qui suit que pour tout automate i , $d(i)$ est strictement positif.

Afin de trouver l'entier $d(i)$ pour $i \in \{1, \dots, n\}$, nous procédons par étapes. Nous cherchons tout d'abord s'il existe une fonction f_i consistante avec T dont le support est de dimension $k = 1$. Si tel est le cas, alors nous fixons $d(i) = 1$, et nous cherchons tous les supports $I = \{i_1\}$ de cardinal 1 tels qu'il existe f_i de support I consistante avec T . Nous rangeons ces supports dans la variable $\mathcal{S}(i)$. Tous les nœuds i dont la dimension minimale $d(i)$ est égal à 1 sont alors ôtés de l'ensemble J des indices à traiter. Nous entrons ensuite dans l'étape $k = 2$, c'est-à-dire que nous cherchons parmi J l'ensemble des automates dont la dimension minimale du support est de 2. Nous procédons ainsi jusqu'à ce que $d(i)$ et $\mathcal{S}(i)$ soient trouvés pour tout $i \in \{1, \dots, n\}$.

Tel qu'il est présenté, l'algorithme 3 ne fait pas mention de la construction de la fonction f_i . Nous avons vu grâce à la proposition 4.8 que s'il existe, pour un support I donné, un vecteur $x \in \{0, 1\}^k$ tel que $\Xi_I(x) = \emptyset$, alors la fonction f_i n'est pas unique. Nous supposons pour l'instant que nous sommes capables de compter le nombre des fonctions f_i consistantes avec T dont le support I est fixé. Nous reviendrons en détail sur ce point dans les parties 4.3.3 et 4.3.4. Nous montrons alors le résultat suivant, qui prouve que l'algorithme 3 permet de résoudre en temps fini les problèmes 1 et 2.

Algorithme 3 - Identification partielle de réseaux booléens.

Etant donnée une table de vérité partielle de dimension n et de taille p , cet algorithme trouve, pour tout $i \in \{1, \dots, n\}$ la dimension minimale $d(i)$ du nœud i , puis l'ensemble $\mathcal{S}(i)$ des supports de dimension $d(i)$ pour lesquels il existe une fonction f_i consistante avec les données.

n : entier ≥ 1 ;

p : entier compris entre 1 et 2^n ;

E, S : Matrices booléennes de dimension $p \times n$;

[Les matrices E et S représentent une table de vérité partielle de dimension n et de taille p . Nous supposons donc que les lignes de la matrice E sont toutes distinctes. Nous supposons de plus qu'aucune colonne de S n'est égale à $(0, \dots, 0)^t$ ou $(1, \dots, 1)^t$.]

Initialisation :

$J \leftarrow \{1, \dots, n\}$;

[J est l'ensemble des indices à traiter]

Pour i de 1 à n faire

$d(i) \leftarrow 0$; $\mathcal{S}(i) \leftarrow \emptyset$;

 [$d(i)$ va contenir la dimension minimale du support du nœud i . $\mathcal{S}(i)$ va contenir l'ensemble des supports minimaux du nœud i pour lesquels il existe une fonction f_i consistante avec T .]

Fin Pour

Recherche :

Pour k de 1 à n faire

 [On cherche les supports de dimension k .]

Pour $i \in J$ faire

Pour (i_1, \dots, i_k) un k -uplet ordonné de $\{1, \dots, n\}$ faire

Si $(H(S_i, E_{i_1}, \dots, E_{i_k}) = H(E_{i_1}, \dots, E_{i_k}))$ **Alors**

 [i_1, \dots, i_k est alors un support minimal du nœud i]

$d(i) \leftarrow k$

$\mathcal{S}(i) \leftarrow \mathcal{S}(i), (i_1, \dots, i_k)$

 [Nous avons trouvé la dimension minimale du nœud i ainsi qu'un support minimal.]

Fin Si

Fin Pour

Si $(\mathcal{S}(i) \neq \emptyset)$ **Alors**

$J \leftarrow J \setminus \{i\}$;

 [Le nœud i a été identifié.]

Fin Si

Fin Pour

Fin Pour

Retourner $d = (d(i))_{i=1..n}$, $\mathcal{S} = (\mathcal{S}(i))_{i=1..n}$;

Proposition 4.9 Soit une table de vérité partielle $T = (E, S)$ de dimension n et de taille p . L'algorithme 3 renvoie en temps fini, pour tout automate $i \in \{1, \dots, n\}$ un entier $d(i)$ compris entre 1 et n , et une liste $\mathcal{S}(i)$ non vide de parties de $\{1, \dots, n\}$ dont le cardinal est égal à $d(i)$. Alors, pour tout $i \in \{1, \dots, n\}$, pour tout $\Sigma \in \mathcal{S}(i)$, il existe une fonction booléenne f_i de support Σ consistante avec la i -ième colonne de S . Nous pouvons donc construire le réseau d'automates booléens $R = (X, B, F)$ avec :

- $X = \{1, \dots, n\}$
- B est la matrice booléenne définie par :

$$\forall i, j \in \{1, \dots, n\}, \quad b_{ij} = \begin{cases} 1 & \text{si } j \in \Sigma \\ 0 & \text{sinon} \end{cases}$$

- $F = (f_i)_{i=1 \dots n}$

Le réseau R appartient à $\mathcal{M}(T)$, i.e. il est canonique et minimal par rapport à T . De plus, en nommant m_i le nombre d'éléments dans $\mathcal{S}(i)$ et \mathcal{N}_i^j le nombre des fonctions f_i consistantes avec T de support $\mathcal{S}(i)(j)$, alors :

$$\text{card}(\mathcal{M}(T)) = \prod_{i=1}^n \left(\sum_{j=1}^{m_i} \mathcal{N}_i^j \right) \quad (4.11)$$

Le fait que R appartienne à $\mathcal{M}(T)$ nous permet de résoudre le problème de l'existence d'un réseau minimal par rapport à une table partielle (problème 1). La formule (4.11) nous permet de résoudre algorithmiquement¹ le problème 2.

Afin de prouver la proposition précédente, nous allons démontrer que pour tout automate $i \in \{1, \dots, n\}$, il existe bien une fonction f_i consistante avec T dont le support est minimal. Comme dans le cas de la proposition 4.4, nous allons procéder par induction sur k , l'indice de la boucle principale.

Pour tout $1 \leq k \leq n$, nous notons $I(k)$ l'ensemble des indices i pour lesquels l'algorithme a réussi à calculer $d(i)$ et $\mathcal{S}(i)$. Une fois ce calcul effectué, nous ôtons i de J , l'ensemble des indices à traiter. Nous en déduisons donc que les $I(k)$ sont disjoints et que :

$$\bigoplus_{k=1}^n I(k) \subset \{1, \dots, n\}$$

Nous démontrons le lemme suivant.

Lemme 4.10 Soit $T = (E, S)$ une table de vérité partielle de dimension n et de taille p . Pour $k \in \{1, \dots, n\}$ et pour $\xi \in I(k)$, nous notons $d = d(\xi)$ et $\Sigma = \mathcal{S}(\xi)$, $\Sigma = \{\Sigma_1, \dots, \Sigma_m\}$ ($d(\xi)$ et $\mathcal{S}(\xi)$ sont les quantités renvoyées par l'algorithme 3).

Alors les deux conditions suivantes sont vérifiées :

- Pour tout $j = 1 \dots m$, $\text{card}(\Sigma_j) = d = k$,
- Pour tout $j = 1 \dots m$, il existe au moins une fonction notée f_ξ de support Σ_j , consistante avec la table T à la colonne ξ . De plus, si g_ξ est une fonction quelconque consistante avec T à la colonne ξ , alors $\text{card}(\text{Supp}(g_\xi)) \geq k$.

¹voir annexe B pour le calcul détaillé d'un majorant sur un exemple

Réciproquement, si pour un automate $\xi \in \{1, \dots, n\}$, il existe une fonction f_ξ consistante avec T de support Σ' avec $\text{card}(\Sigma') = k$ et pour toute fonction g_ξ consistante avec T on a : $\text{card}(\text{Supp}(g_\xi)) \geq k$, alors $\xi \in I(k)$.

Démonstration

Montrons tout d'abord que ceci est vrai pour $k = 1$.

Soit $\xi \in I(1)$. Il est immédiat que $d(\xi) = 1$ et $\mathcal{S}(\xi) = \{\Sigma_1, \dots, \Sigma_m\}$, avec :

$$\forall j \in \{1, \dots, m\}, \quad \text{card}(\Sigma_j) = 1$$

Pour tout $j \in \{1, \dots, m\}$, nous notons $\Sigma_j = \{v_j\}$. D'après l'algorithme, nous avons, pour tout $j \in \{1, \dots, m\}$ l'égalité $H(S_\xi, E_{v_j}) = H(E_{v_j})$. En utilisant la proposition 4.8, nous démontrons qu'il existe une fonction f_ξ^j de support inclus dans Σ_j consistante avec T à la colonne ξ . Puisque nous avons supposé que la matrice S ne comportait aucune colonne constante (i.e. égale à $(0, \dots, 0)^t$ ou $(1, \dots, 1)^t$), nous pouvons en déduire que pour toute fonction g_ξ consistante avec T , on a :

$$\text{card}(\text{Supp}(g_\xi)) \geq 1$$

ce qui entraîne que quel que soit $j \in \{1, \dots, m\}$, $\text{Supp}(f_\xi^j) = \Sigma_j$.

Réciproquement, si pour un $\xi \in \{1, \dots, n\}$ donné il existe une fonction f_ξ consistante avec T de support $\{v\}$, alors $H(S_\xi, E_v) = H(E_v)$ et nous pouvons en déduire que $d(\xi) = 1$, $\{v\} \in \mathcal{S}(\xi)$ et $\xi \in I(1)$.

Nous supposons à présent que la propriété est vérifiée pour $1 \leq k' \leq k - 1$ et nous montrons qu'elle est alors vérifiée au rang k .

Soit $\xi \in I(k)$. On note $d = d(\xi)$ et $\Sigma = \mathcal{S}(\xi)$ ($\Sigma = \{\Sigma_1, \dots, \Sigma_m\}$). D'après l'algorithme, il est clair que $d = k$ et $\text{card}(\Sigma_j) = k$ pour tout $j \in \{1, \dots, m\}$. Soit $j \in \{1, \dots, m\}$, nous notons $\{v_1, \dots, v_k\}$ les éléments de Σ_j . De l'égalité $H(S_\xi, E_{v_1}, \dots, E_{v_k}) = H(E_{v_1}, \dots, E_{v_k})$, nous déduisons en utilisant la proposition 4.8 qu'il existe une fonction f_ξ^j consistante avec T à la colonne ξ dont le support est inclus dans Σ_j . D'après notre hypothèse de récurrence, le cardinal de $\text{Supp}(f_\xi^j)$ est supérieur ou égal à k (sinon l'algorithme aurait détecté le nœud ξ à une étape antérieure), ce qui implique que :

$$\text{Supp}(f_\xi^j) = \Sigma_j$$

Réciproquement, supposons que pour un automate $\xi \in \{1, \dots, n\}$ il existe une fonction f_ξ consistante avec T , de support Σ' de cardinal k . Alors en notant v_1, \dots, v_k les éléments de Σ' , on a l'égalité $H(S_\xi, E_{v_1}, \dots, E_{v_k}) = H(E_{v_1}, \dots, E_{v_k})$ qui entraîne que $d(\xi) = k$, $\Sigma' \in \mathcal{S}(\xi)$ et $\xi \in I(k)$. ■

Grâce à ce lemme, la première partie de la proposition 4.9 est démontrée. Pour montrer la seconde partie, il nous suffit de compter les réseaux construits par l'algorithme. Pour chaque automate $i \in \{1, \dots, n\}$, il faut choisir un support $\Sigma = \mathcal{S}(i)(j)$ dans $\mathcal{S}(i)$. Il y a pour ce choix m_i possibilités. Puis, pour chaque support Σ ainsi choisi, il faut choisir parmi \mathcal{N}_i^j fonctions booléennes, d'où :

$$\text{card}(\mathcal{M}(T)) = \prod_{i=1}^n \left(\sum_{j=1}^{m_i} \mathcal{N}_i^j \right)$$

Nous proposons dans l'annexe B le calcul d'un majorant de cette quantité sur un exemple particulier.

L'algorithme 3 permet l'identification de réseaux booléens minimaux par rapport à une table de vérité partielle T . Nous proposons dans ce qui suit un ensemble de tests qui illustrent l'efficacité de cet algorithme. Nous examinons également sa complexité. Par la suite, nous reviendrons plus en détail sur le problème 3, c'est-à-dire le problème de consistance à support fixé.

4.3.3 Résultats

Afin de tester l'efficacité de l'algorithme 3, nous avons produit un ensemble de fonctions utilitaires en Matlab. Nous trouverons dans l'annexe B une description de ces programmes et fonctions.

Le principe du test est le suivant : nous commençons par générer un réseau de taille N fixé et dont la connectivité maximale vaut K . Nous choisissons de générer N fonctions booléennes dont la connectivité réelle vaut k , avec $k \in \{K - 1, K\}$. Les supports de ces fonctions sont choisis aléatoirement dans $\mathfrak{P}(\{1, \dots, N\})$. En assurant que la fonction booléenne donnée soit bien de connectivité réelle k (i.e. qu'elle n'est pas dégénérée), nous imposons ainsi au réseau R d'être canonique. Une fois ce réseau R construit, nous pouvons générer, pour P compris entre 1 et 2^N une table de vérité partielle $T = (E, S)$ de taille P . Le programme de génération assure que toutes les lignes de E sont distinctes. Etant donnée T , nous lançons ensuite le programme d'identification et nous recherchons parmi les réseaux trouvés le réseau initial R .

En fixant $N = 9$, nous avons lancé ce test pour K variant de 1 à 9. Pour chacune de ces expériences, nous avons fixé plusieurs valeurs de P (voir figure 4.4). Pour chacune de ces valeurs, nous avons généré aléatoirement 100 tables de vérité partielles et nous avons compté pour chacune de ces tables le taux de nœuds correctement induits. Les courbes de la figure 4.4 présentent, pour P prenant 21 valeurs entre 10 et 256, la moyenne du taux de nœuds du réseau initial correctement induits en fonction de P .

Remarquons ici que lorsque l'algorithme identifie une fonction f_i de support fixé, il se peut que les données ne soient pas suffisantes pour déterminer f_i partout (cf. l'exemple 4.5 et la démonstration de la proposition 4.8). Dans la partie 4.3.4, nous étudierons le comportement de l'algorithme dans le cas où l'on demande en plus que chaque fonction f_i soit déterminée de façon unique.

L'étude des courbes 4.4 montre l'efficacité de cet algorithme. En effet, même pour le cas de la connectivité maximale $K = 9$, nous voyons que pour $P \simeq 256$, le taux de nœuds induits est déjà très proche de 1. Pour inférer le réseau initial, il suffit donc de générer aléatoirement la moitié de la table de vérité complète (comme $N = 9$, la taille de la table de vérité complète est $2^9 = 512$). Pour des connectivités plus faibles, le réseau est déjà quasiment complètement induit pour un P encore plus petit : $P \simeq 64 = \frac{1}{8}2^N$ pour $K = 6$.

Ces courbes semblent montrer que pour N fixé, il existe un lien entre la connectivité K d'un réseau et la taille P^* nécessaire pour induire ce réseau. Dans le cas où nous imposons à notre algorithme de retrouver exactement le réseau initial (c'est-à-dire de déterminer complètement et de façon unique, pour chaque automate i , la fonction booléenne correspondante), nous donnons dans la suite une formule explicite entre P^* et K .

Etudions à présent la complexité de cet algorithme. Dans le cas où T est complète, nous

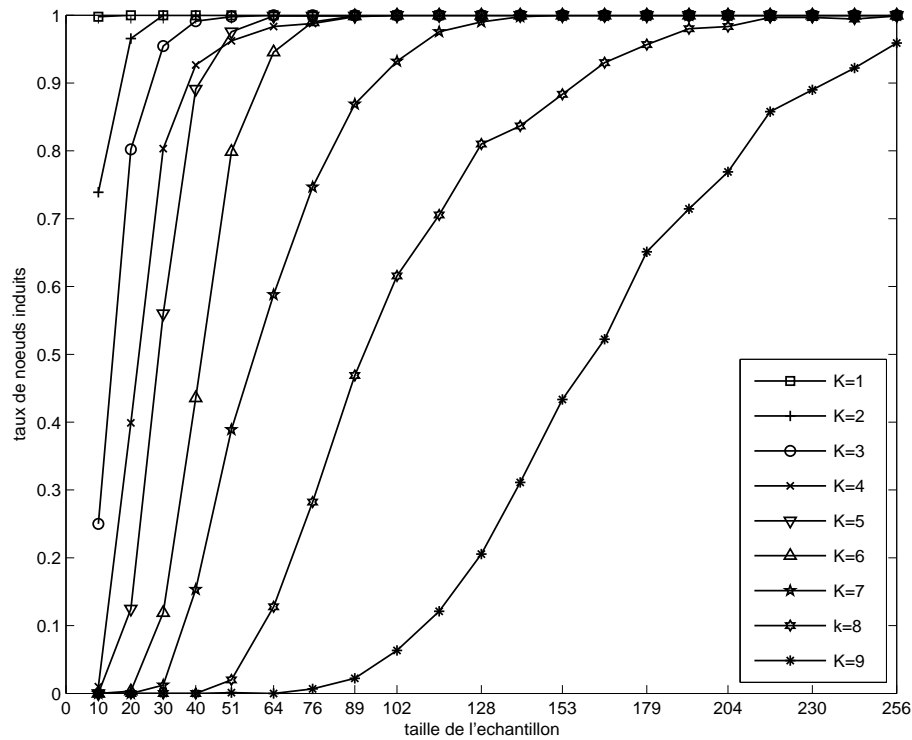


FIG. 4.4: Test de l'algorithme d'identification partielle. La dimension vaut $N = 9$. Pour chaque K variant de 1 à 9, nous générons un réseau de dimension N et de connectivité maximale K . Pour chacun de ces réseaux, nous générons 100 tables de vérités partielles de taille P (P prenant 21 valeurs distinctes entre 10 et 256) et nous traçons la moyenne des nœuds du réseau initial correctement induit. Nous autorisons les fonctions f_i à n'être que partiellement déterminées (voir texte).

avons vu plus haut que la complexité de l'algorithme 1 était, dans le pire cas :

$$\beta(n) = O\left(\frac{1}{2}n^2 2^{2n}\right)$$

(où n désigne la dimension du réseau à inférer).

Nous avons vu ensuite qu'en ne cherchant que les nœuds de connectivité inférieure à K fixé, nous obtenions l'algorithme 2 dont la complexité est, dans le pire cas :

$$\beta'(n, K) = O\left(K^2 n^{K+1} 2^n\right)$$

Dans le cas de l'identification à partir d'une table complète, la complexité contient un terme exponentiel 2^n qui provient du fait que les matrices E et S comportent 2^n lignes. Le calcul de l'entropie $H(E_{i_1}, \dots, E_{i_k})$ nécessite donc dans le cas d'une table complète : $O(k2^n)$ opérations ; alors que dans le cas d'une table partielle de taille p , ce calcul ne nécessite que $O(kp)$ opérations.

Nous notons $\gamma(n, p)$ la complexité en pire cas de l'algorithme 3. Comme dans la partie 4.2, nous considérons également une version *tronquée* de l'algorithme 3, dans le sens où nous ne cherchons à identifier que les nœuds de connectivité inférieure ou égale à K , K étant une constante fixée à l'avance. Nous noterons $\gamma'(n, p, K)$ la complexité en pire cas de cette version tronquée. En reprenant les calculs effectués dans la partie 4.2, nous montrons que :

$$\gamma(n, p) = O_{n \rightarrow \infty} \left(\frac{1}{2} p n^2 2^n \right)$$

Dans cette expression, le terme 2^n provient de la somme $s(n) = \sum_{k=1}^n k C_n^k$. Dans la version tronquée, cette somme est remplacée par la somme partielle $s(n, K) = \sum_{k=1}^K k C_n^k$, ce qui nous donne :

$$\gamma'(n, p, K) = O_{n \rightarrow \infty} (K^2 p n^{K+1})$$

L'algorithme 3, nous permet donc, pour $K \in \{1, \dots, n\}$ et pour $p \in \{1, \dots, 2^n\}$ fixés, d'inférer, à partir d'une table partielle de taille p , l'ensemble des nœuds de connectivité inférieure ou égale à K en un temps qui est asymptotiquement polynômial en n , la dimension du réseau.

Remarque 4.5 *Il est bien entendu que l'inférence d'un nœud peut échouer si p est trop petit. En effet, il est clair que plus la connectivité d'un nœud est élevée, plus la taille des données p doit être importante pour permettre d'identifier de façon unique la fonction correspondante. Pour un p faible, l'algorithme 3 se contente d'énoncer toutes les fonctions de support minimales possibles telles qu'elles soient consistantes avec les données. Si les données sont trop peu nombreuses (ou si elles sont mal réparties), l'algorithme ne permet pas d'identifier les fonctions f_i de manière unique.*

4.3.4 Résolution du problème de consistance minimale à support fixé

La remarque précédente nous amène à nous poser la question de l'efficacité de l'algorithme d'identification relativement à p , la taille des données. Il est clair que pour que l'algorithme soit efficace, il doit être capable d'inférer les fonctions f_i à partir d'une table de taille p où p est le plus petit possible. Certains résultats asymptotiques ont déjà été proposés (voir notamment [1] et [2]).

Ce que nous proposons ici est une formule reliant explicitement p^* , la taille nécessaire des données à K , la connectivité du réseau. Etant donnée une table de vérité partielle de taille p , un automate $i \in \{1, \dots, n\}$ et un support de dimension k (minimale) $I \subset \{1, \dots, n\}$, nous avons vu qu'il peut exister plusieurs fonctions f_i de support inclus dans I consistantes avec T à la colonne i . Grâce à la proposition 4.8, nous avons vu qu'il en existe $2^{N_n^k}$, où N_n^k est le cardinal de l'ensemble :

$$A(T, I) = \left\{ x \in \{0, 1\}^k \mid \Xi_I(x) = \emptyset \right\}$$

Plus précisément, nous nous intéressons à la probabilité que cet ensemble $A(T, I)$ soit vide, c'est-à-dire que la fonction f_i soit déterminée de manière unique. Afin de calculer cette probabilité, nous allons tout d'abord formaliser le problème.

I est une partie de $\{1, \dots, n\}$ de cardinal k . Nous considérons l'expérience aléatoire suivante :

(Ex) On choisit aléatoirement un n -uplet $x = (x_1, \dots, x_n)$ appartenant à $\{0, 1\}^n$

En répétant p fois cette expérience, nous générons une matrice booléenne E de dimension $p \times n$ dont les lignes sont les vecteurs x . Afin que E appartienne à $\mathcal{E}_{n,p}$, il faut en plus assurer que les lignes de E soient toutes distinctes. Cela revient à supposer que la répétition de l'expérience (Ex) se fait sans remise. Une répétition de p expériences (Ex) successives sera appelée un p -tirage. un p -tirage est donc un p -uplet d'éléments de $\{0, 1\}^n$.

Pour tout $y \in \{0, 1\}^k$, nous notons $\mathcal{C}(y)$ l'ensemble des éléments $x \in \{0, 1\}^n$ tels que pour tout $j = 1 \dots k$, $x_{i_j} = y_j$, où $\{i_1, \dots, i_k\} = I$. Ces ensembles $\mathcal{C}(y)$ vérifient de façon évidente les propriétés suivantes :

$$(i) \quad \forall y, y' \in \{0, 1\}^k, \quad (y \neq y' \implies \mathcal{C}(y) \cap \mathcal{C}(y') = \emptyset)$$

$$(ii) \quad \bigsqcup_{y \in \{0, 1\}^k} \mathcal{C}(y) = \{0, 1\}^n$$

En d'autres termes, les $\mathcal{C}(y)$ forment une partition de $\{0, 1\}^n$. Nous démontrons facilement qu'en outre, ces ensembles sont au nombre de 2^k et qu'ils ont tous le même cardinal : 2^{n-k} . La probabilité que nous cherchons est donc la probabilité que, après une répétition de p expériences (Ex) successives sans remise on ait, pour tout $y \in \{0, 1\}^k$ au moins un élément x qui appartienne à $\mathcal{C}(y)$.

Par la suite, étant donné $y \in \{0, 1\}^k$, on dira que la classe $\mathcal{C}(y)$ est *représentée* s'il existe au moins un x appartenant à $\mathcal{C}(y)$. Nous considérons la variable aléatoire discrète Q égale au nombre de classes représentées dans un p -tirage, pour $p \geq 1$. Q est à valeurs dans $\{1, \dots, 2^k\}$. Afin de calculer formellement la distribution de la variable Q , nous allons relaxer l'hypothèse *sans remise*. Nous discuterons ensuite du bien-fondé d'un tel choix.

Proposition 4.11 *Nous considérons des p -tirages indépendants (c'est-à-dire avec remise). Pour tout $q \in \{1, \dots, 2^k\}$, on a :*

$$P(Q = q) = \begin{cases} \frac{1}{2^{kp}} C_{2^k}^q \sum_{i=0}^q C_q^i (-1)^i (q-i)^p & \text{si } p \geq q \\ 0 & \text{si } p < q \end{cases} \quad (4.12)$$

Démonstration

Soit $q \in \{1, \dots, 2^k\}$. Nous allons ramener le problème à un problème équivalent. Nous posons $\Omega = \{0, 1\}^k$. Les ensembles $\mathcal{C}(y)$ formant une partition de $\{0, 1\}^n$, choisir un élément dans $\{0, 1\}^n$ puis le ranger dans la classe $\mathcal{C}(y)$ correspondante revient à choisir un élément dans Ω . Comme les $\mathcal{C}(y)$ ont tous le même cardinal et que nous avons supposé que nos p -tirages étaient indépendants, tous les éléments de Ω sont équiprobables. Un p -tirage correspond donc à un p -uplet de Ω . L'ensemble des possibles est Ω^p qui est de cardinal $(2^k)^p = 2^{kp}$.

Calculer la probabilité $P(Q = q)$ revient donc à dénombrer, parmi Ω^p , le nombre de p -uplets dans lesquels il y a exactement q éléments distincts. Nous voyons immédiatement qu'alors, si $p < q$, la probabilité $P(Q = q)$ est égale à 0.

Nous supposons par la suite que $p \geq q$. Nous voulons dénombrer l'ensemble $B_{p,q} \subset \Omega^p$ des p -uplets de Ω qui contiennent exactement q éléments distincts. Pour cela, il nous faut tout d'abord choisir quels sont ces q éléments. Cela revient à choisir q éléments $\omega_1, \dots, \omega_q$ parmi 2^k .

Il y a $C_{2^k}^q$ choix possibles. Une fois ces éléments choisis, nous posons l'application :

$$\Phi_\omega : \Omega^p \longrightarrow \mathbb{N}^q$$

qui, à tout p -uplet X de Ω^p associe $(n_1, \dots, n_q) \in \mathbb{N}^q$ tel que, pour tout $i \in \{1, \dots, q\}$, n_i est le nombre de fois où l'élément ω_i apparaît dans X . Φ_ω est une surjection. Le problème que nous nous posons est de calculer $C_\omega(p, q)$ le nombre d'éléments $X \in \Omega^p$ tels que $\Phi_\omega(X) = (n_1, \dots, n_q)$ vérifie les deux propriétés suivantes :

$$\forall i \in \{1, \dots, q\}, \quad n_i \in \{1, \dots, p\} \quad (4.13)$$

$$\sum_{i=1}^q n_i = p \quad (4.14)$$

La propriété (4.13) équivaut au fait que tous les éléments ω_i sont présents au moins une fois dans X . La propriété (4.14), quant à elle, équivaut au fait que X ne contient aucun élément autre que les ω_i . Une fois la quantité $C_\omega(p, q)$ calculée, on a alors :

$$P(Q = q) = \frac{1}{2^{kp}} C_{2^k}^q C_\omega(p, q)$$

Nous notons $D_{p,q}$ l'ensemble des $N = (n_1, \dots, n_q) \in \mathbb{N}^q$ qui vérifient les propriétés (4.13) et (4.14). Nous avons alors :

$$C_\omega(p, q) = \sum_{N \in D_{p,q}} \text{card}(\Phi_\omega^{-1}(\{N\}))$$

Pour $N \in D_{p,q}$ fixé, le cardinal de $\Phi_\omega^{-1}(\{N\})$ est connu (on pourra par exemple se référer à [88], page 170). Il s'agit du coefficient multinomial :

$$\text{card}(\Phi_\omega^{-1}(\{N\})) = \frac{p!}{n_1! n_2! \dots n_q!}$$

Nous sommes donc ramenés au calcul de la quantité :

$$C_\omega(p, q) = \sum_{\substack{n_i > 0 \\ n_1 + \dots + n_q = p}} \frac{p!}{n_1! n_2! \dots n_q!}$$

Montrons à présent que, p étant fixé ($p \geq 1$), on a :

$$\forall 1 \leq q \leq p, \quad C_\omega(p, q) = \sum_{i=0}^q C_q^i (-1)^i (q-i)^p$$

On vérifie facilement que cette propriété est vérifiée pour $p = 1$. Nous supposons désormais qu'elle est vérifiée pour tout $1 \leq p' \leq p$ et nous démontrons qu'elle est alors vérifiée au rang

$p + 1$. Par définition, nous avons, pour $q \leq p + 1$:

$$\begin{aligned}
C_\omega(p + 1, q) &= \sum_{(n_1, \dots, n_q) \in D_{p+1, q}} \frac{(p + 1)!}{n_1! n_2! \dots n_q!} \\
&= \sum_{n_1=1}^{p+1} \frac{1}{n_1!} \sum_{(n_2, \dots, n_q) \in D_{p+1-n_1, q-1}} \frac{(p + 1)!}{n_2! \dots n_q!} \\
&= \sum_{n_1=1}^{p+1} \frac{(p + 1)!}{n_1! (p + 1 - n_1)!} \sum_{(n_2, \dots, n_q) \in D_{p+1-n_1, q-1}} \frac{(p + 1 - n_1)!}{n_2! \dots n_q!} \\
&= \sum_{n_1=1}^{p+1} C_{p+1}^{n_1} C_\omega(p + 1 - n_1, q - 1)
\end{aligned}$$

n_1 étant strictement positif, en utilisant l'hypothèse de récurrence, nous avons donc :

$$C_\omega(p + 1, q) = \sum_{n_1=1}^{p+1} C_{p+1}^{n_1} \left(\sum_{i=0}^{q-1} C_{q-1}^i (-1)^i (q - 1 - i)^{p+1-n_1} \right)$$

D'où, en intervertissant les deux sommes :

$$C_\omega(p + 1, q) = \sum_{i=0}^{q-1} C_{q-1}^i (-1)^i \left(\sum_{n_1=1}^{p+1} C_{p+1}^{n_1} (q - 1 - i)^{p+1-n_1} \right)$$

En utilisant la formule du binôme de Newton, nous obtenons :

$$\begin{aligned}
C_\omega(p + 1, q) &= \sum_{i=0}^{q-1} C_{q-1}^i (-1)^i [(q - i)^{p+1} - (q - 1 - i)^{p+1}] \\
&= \sum_{i=0}^{q-1} C_{q-1}^i (-1)^i (q - i)^{p+1} - \sum_{i=0}^{q-1} C_{q-1}^i (-1)^i (q - 1 - i)^{p+1}
\end{aligned}$$

Par un changement de variable dans la deuxième somme, la formule devient :

$$\begin{aligned}
C_\omega(p + 1, q) &= \sum_{i=0}^{q-1} C_{q-1}^i (-1)^i (q - i)^{p+1} - \sum_{i=1}^q C_{q-1}^{i-1} (-1)^{i-1} (q - i)^{p+1} \\
&= \sum_{i=0}^{q-1} C_{q-1}^i (-1)^i (q - i)^{p+1} + \sum_{i=1}^q C_{q-1}^{i-1} (-1)^i (q - i)^{p+1} \\
&= q^{p+1} + \sum_{i=1}^q (C_q^i + C_q^{i-1}) (-1)^i (q - i)^{p+1}
\end{aligned}$$

Grâce à la formule du triangle de Pascal :

$$\begin{aligned}
C_\omega(p + 1, q) &= q^{p+1} + \sum_{i=1}^q C_{q+1}^i (-1)^i (q - i)^{p+1} \\
&= \sum_{i=0}^q C_{q+1}^i (-1)^i (q - i)^{p+1}
\end{aligned}$$

Ce qui montre que la formule est vraie au rang $p + 1$. Nous en déduisons que pour tout $p \geq 1$, nous avons :

$$\forall 1 \leq q \leq p, \quad C_\omega(p, q) = \sum_{i=0}^q C_q^i (-1)^i (q - i)^p$$

Par suite, nous trouvons la formule voulue pour $P(Q = q)$. ■

En remplaçant q par 2^k dans (4.12), nous obtenons la probabilité pour que toutes les classes $\mathcal{C}(y)$ soient représentées dans un p -tirage. Cette probabilité est égale à la probabilité qu'une fonction de connectivité réelle k soit déterminée de façon unique à partir d'une table de taille p . Nous la notons P_k :

$$P_k = P(Q = 2^k) = \frac{1}{2^{kp}} \sum_{i=0}^{2^k} C_{2^k}^i (-1)^i (2^k - i)^p \quad (4.15)$$

Cette formule est valable pour $p \geq 2^k$ (sinon P_k vaut zéro). Nous remarquons que P_k est indépendante de n , la dimension du réseau. Ceci provient du fait que pour calculer l'expression (4.12), nous avons fait l'hypothèse que toutes les lignes de la matrice d'entrées E étaient choisies de manière indépendante. Ceci n'est pas le cas dans notre algorithme puisque nous avons décidé de générer uniquement des lignes distinctes. Néanmoins, lorsque n tend vers $+\infty$, il est clair que la probabilité de choisir deux fois la même ligne tend vers 0. Nous attendons donc de notre algorithme qu'il donne une probabilité $P(Q = 2^k)$ proche de l'expression (4.15), et même de s'en rapprocher lorsque n grandit (voir résultats expérimentaux figure 4.5).

Grâce à cette formule, nous pouvons trouver numériquement un lien entre la connectivité k d'une fonction et la taille p de la table qu'il faut donner en entrée pour pouvoir identifier cette fonction de manière unique. Par exemple, pour une fonction de connectivité $k = 4$, nous pouvons calculer la probabilité P_4 :

$$P_4 = 1 - \frac{16}{16^p} - 16 \left(\frac{15}{16}\right)^p + 120 \left(\frac{7}{8}\right)^p - 560 \left(\frac{13}{16}\right)^p + 1820 \left(\frac{3}{4}\right)^p - 4368 \left(\frac{11}{16}\right)^p + 8008 \left(\frac{5}{8}\right)^p - \\ 11440 \left(\frac{9}{16}\right)^p + 12870 \frac{1}{2^p} - 11440 \left(\frac{7}{16}\right)^p + 8008 \left(\frac{3}{8}\right)^p - 4368 \left(\frac{5}{16}\right)^p + 1820 \frac{1}{4^p} - 560 \left(\frac{3}{16}\right)^p + 120 \frac{1}{8^p}$$

Nous pouvons alors résoudre numériquement l'équation :

$$P_4 = 0.9$$

Nous trouvons $\bar{p} \simeq 78.11$, ce qui signifie que, dans le cas où n est suffisamment grand, si nous voulons avoir 90% de chances d'identifier une fonction de connectivité réelle 4, il faut une table de vérité de taille supérieure ou égale à $p^* = 79$ (voir fig. 4.5).

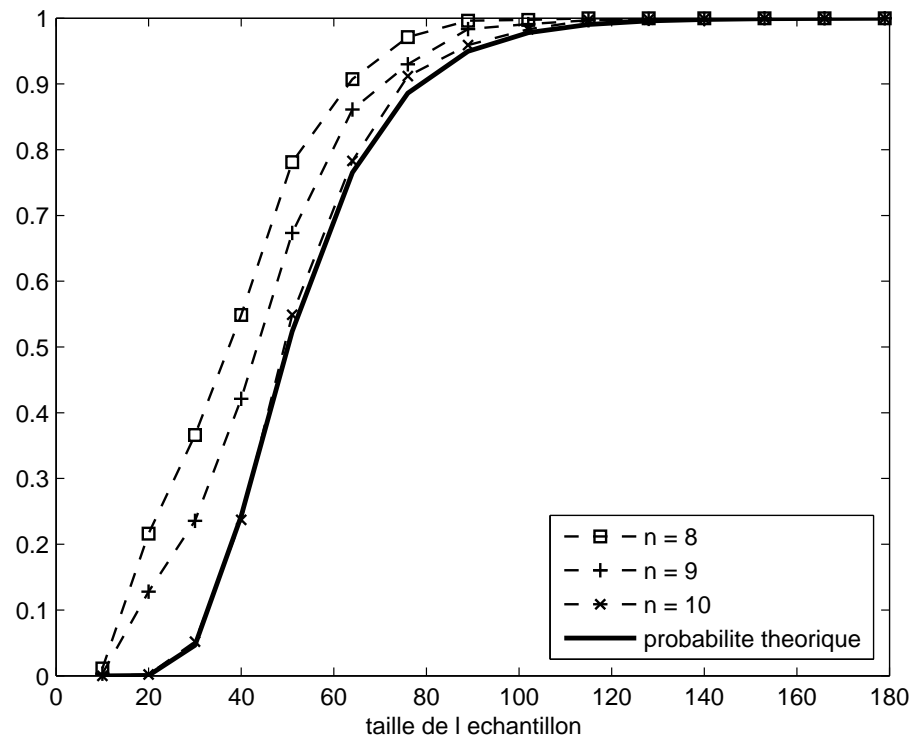


FIG. 4.5: Les courbes en pointillets sont issues d'une répétition de l'algorithme d'inférence partielle pour $N = 8, 9$ et 10 . Pour chacune de ces valeurs, nous avons généré un réseau de taille N et de connectivité K . Puis, pour différentes valeurs de P , nous avons généré 100 tables de taille P et lancé 100 fois l'algorithme 3 et nous avons calculé la moyenne du taux de nœuds correctement induits, c'est-à-dire tels que les fonctions f_i sont déterminées de manière unique. La courbe en gras représente quant à elle la probabilité théorique issue de la formule (4.15).

Troisième partie

Etude de systèmes dynamiques comportant des lois de puissances

Chapitre 5

Approximation de systèmes par des lois de puissance

Nous avons vu dans le premier chapitre de cette thèse qu'il existe différentes classes de systèmes dynamiques continus utilisées dans la modélisation de systèmes biologiques. Nous avons notamment cité la classe des S-systèmes, qui sont des systèmes définis par des équations différentielles comportant des lois de puissance (voir [67–69], [72, 80, 81], voir également [5, 6]). Dans cette partie, nous allons étudier plus précisément cette classe de systèmes. Nous verrons notamment qu'elle est liée à la classe plus générale des systèmes quasi-monomiaux [9, 10, 28].

Les systèmes quasi-monomiaux et les S-systèmes sont des classes de systèmes qui présentent selon nous deux intérêts majeurs. Tout d'abord, ils offrent un large cadre de modélisation particulièrement bien adapté aux problèmes biologiques. On peut montrer à cet effet (voir [10]) qu'ils peuvent se ramener aux systèmes de Lotka-Volterra qui sont une classe de systèmes très largement utilisés en modélisation.

D'autre part, et c'est un point qui nous semble important, leur formulation mathématique autorise un traitement symbolique (ou symbolique-numérique) qui permet de *prover* certaines de leurs propriétés. Leur utilisation ne se réduit donc pas à des simulations numériques, comme c'est parfois le cas pour d'autres types de modèles continus. A ce titre, ils constituent un outil de modélisation utile qui, tout comme les systèmes dynamiques discrets, apporte des résultats spécifiques permettant d'améliorer la compréhension des phénomènes représentés.

Nous proposons dans ce chapitre un algorithme symbolique-numérique¹ basé sur ce que nous nommons la *S-approximation* (voir [81]). Cet algorithme nous permet de converger, à partir d'un point initial donné, vers un point d'équilibre hyperbolique d'un système dynamique dont le second membre est défini sur $(\mathbb{R}_+^*)^n$. Simultanément, il calcule un S-système particulier, qui approxime le système initial autour de cet équilibre. Nous reviendrons plus en détail sur cette approximation, et nous verrons notamment que le S-système approximant et le système initial sont topologiquement conjugués autour de l'équilibre. Ce type d'approximation offre donc une alternative à la technique classique de la linéarisation, en proposant une forme canonique non linéaire, sous forme d'équations différentielles du type monomiales. Nous verrons notamment sur un exemple que la S-approximation semble s'avérer plus fine que la linéarisation.

¹Ce travail a fait l'objet d'un article et d'une présentation à la conférence de calcul formel *ISSAC'05* (voir [79])

5.1 La classe des S-systèmes

5.1.1 Définition

Nous donnons ici une définition de la classe des S-systèmes. Cette définition est tirée de l'ouvrage [80].

Définition 5.1 Soient $n \in \mathbb{N}^*$ et $m \in \mathbb{N}$. On considère $\alpha = (\alpha_1, \dots, \alpha_n)$ et $\beta = (\beta_1, \dots, \beta_n)$ deux vecteurs appartenant à $(\mathbb{R}_+^*)^n$, ainsi que G et H deux matrices réelles à n lignes et $n+m$ colonnes. Un S-système de paramètres α, β, G et H , noté $\mathcal{S}(\alpha, \beta, G, H)$ est un système dynamique de dimension n défini par les équations différentielles autonomes :

$$\forall i \in \{1, \dots, n\}, \quad \dot{x}_i = \alpha_i \prod_{j=1}^{n+m} x_j^{g_{ij}} - \beta_i \prod_{j=1}^{n+m} x_j^{h_{ij}}$$

où $x(t) = (x_1(t), \dots, x_n(t))$ est le vecteur des variables et $\mathcal{X} = (x_{n+1}, \dots, x_{n+m}) \in (\mathbb{R}_+^*)^n$ est un vecteur de paramètres, qui sont supposés constants en fonction du temps.

L'ouvrage [80] utilise une terminologie légèrement différente : les paramètres x_{n+1}, \dots, x_{n+m} sont appelés variables *indépendantes*, par opposition aux variables *dépendantes* qui sont les $x_i(t)$, pour $i \in \{1, \dots, n\}$. L'intérêt de faire apparaître explicitement ces paramètres dans les équations de \mathcal{S} réside essentiellement dans une démarche de modélisation d'un processus biologique réel. En effet, un système biologique comporte généralement de nombreux paramètres, et parmi ceux-ci on distingue les paramètres sur lesquels on veut pouvoir agir en pratique (par exemple, la concentration d'une espèce extérieure au système) de ceux qui ont une valeur fixée, connue ou inconnue, sur laquelle on ne peut agir directement (par exemple, les constantes chimiques d'une réaction, qui ne dépendent que de la température et de la pression). Les paramètres du premier type, que E. Voit nomme variables indépendantes sont donc des grandeurs physiques que l'on peut vouloir faire évoluer d'une expérience à une autre afin d'étudier leur influence sur la dynamique du système. Il peut donc être utile de les faire apparaître explicitement dans les équations.

En revanche, du point de vue mathématique, étant donné que x_{n+1}, \dots, x_{n+m} sont supposées être des constantes strictement positives, on peut tout-à-fait les faire rentrer dans les paramètres α_i et β_i , et ainsi se ramener à une forme plus compacte des équations :

$$\mathcal{S}(\alpha, \beta, G, H) : \quad \left(\forall i \in \{1, \dots, n\}, \quad \dot{x}_i = \alpha_i \prod_{j=1}^n x_j^{g_{ij}} - \beta_i \prod_{j=1}^n x_j^{h_{ij}} \right)$$

C'est cette forme que nous utiliserons dans la suite. Etant donnés α, β, G et H nous introduisons le champ de vecteur de dimension n :

$$F(x) = \begin{pmatrix} f_1(x_1, \dots, x_n) \\ \vdots \\ f_n(x_1, \dots, x_n) \end{pmatrix}$$

avec :

$$\forall i \in \{1, \dots, n\}, \quad f_i(x_1, \dots, x_n) = \alpha_i \prod_{j=1}^n x_j^{g_{ij}} - \beta_i \prod_{j=1}^n x_j^{h_{ij}}$$

Le domaine de définition de F dépend bien sûr des paramètres g_{ij} et h_{ij} . Cependant, sur l'ouvert $\Omega = (\mathbb{R}_+^*)^n$, F est définie quelles que soient les valeurs des paramètres. De plus, F est de classe \mathcal{C}^∞ sur Ω . Le théorème de Cauchy-Lipschitz (voir par exemple [83]) nous assure donc l'existence et l'unicité d'une solution maximale du système \mathcal{S} dans Ω , étant donnée une condition initiale $x(0) = x^0 \in \Omega$.

Dans la suite de ce chapitre, nous allons utiliser une notation² particulièrement utile. Étant donné un vecteur v appartenant à Ω et une matrice A carrée à coefficients réels de dimension n , nous définissons le vecteur noté v^A ("*v* puissance A ") appartenant à Ω tel que :

$$\forall i \in \{1, \dots, n\}, \quad (v^A)_i = \prod_{j=1}^n v_j^{a_{ij}}$$

Une justification de cette notation réside dans le résultat suivant :

Proposition 5.1 *Soient $v \in \Omega$ et $A \in \mathfrak{M}_n(\mathbb{R})$. Pour tout vecteur $u \in \Omega$, on note $\ln u$ le vecteur $(\ln u_1, \dots, \ln u_n)$. $\ln u$ est un vecteur de \mathbb{R}^n , appelé logarithme du vecteur u . Nous avons alors :*

$$\ln(v^A) = A \ln v$$

Démonstration

Notons $w = v^A$. Pour tout $i \in \{1, \dots, n\}$, nous avons :

$$\ln(w_i) = \ln \left(\prod_{j=1}^n v_j^{a_{ij}} \right) = \sum_{j=1}^n a_{ij} \ln v_j$$

D'où nous déduisons que $\ln w = A \ln v$. ■

Grâce à cette notation, nous pouvons donner une nouvelle définition des S-systèmes :

Définition 5.2 *Soient $n \in \mathbb{N}^*$, $\alpha, \beta \in \Omega$ et $G, H \in \mathfrak{M}_n(\mathbb{R})$. Le S-système $\mathcal{S}(\alpha, \beta, G, H)$ est le système dynamique défini sur l'ouvert $\Omega = (\mathbb{R}_+^*)^n$ par :*

$$\mathcal{S}(\alpha, \beta, G, H) \begin{cases} \dot{x}_i = \alpha_i(x^G)_i - \beta_i(x^H)_i, & i = 1 \dots n \\ x(0) = x^0 \in \Omega \end{cases} \quad (5.1)$$

Les coefficients α_i et β_i sont parfois appelés taux cinétiques et les coefficients g_{ij} et h_{ij} ordres cinétiques.

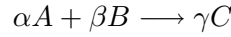
5.1.2 Discussion sur le modèle "S-systèmes"

Nous faisons ici deux remarques sur les S-systèmes qui nous semblent importantes. La première concerne le lien qui peut être fait avec les systèmes quasi-monomiaux. Les S-systèmes s'inscrivent donc dans un formalisme ancien centré sur l'étude des lois de puissance dans les équations algébriques et différentielles (on peut notamment citer le travail de A. Bruno [11]). La seconde remarque montre un lien évident entre les S-systèmes et les équations de la cinétique chimique, insistant ainsi sur la pertinence de l'utilisation des S-systèmes pour modéliser des réseaux d'interaction biochimiques.

²Il s'agit d'une notation classique, voir par exemple [11]

Intérêt des S-systèmes pour la biochimie

Afin de rendre compte du lien qui existe entre les S-systèmes et la cinétique chimique, nous allons considérer l'exemple d'une réaction chimique classique, dans laquelle deux espèces A et B se transforment en une espèce C :



En appliquant la loi d'action de masse (cf. le cours sur les réseaux de réactions chimiques de M. Feinberg disponible sur le web [24]), on obtient l'équation différentielle régissant l'évolution de la concentration de l'espèce C :

$$\dot{c}_c = k_1 c_a^\alpha c_b^\beta$$

où k_1 est une constante positive qui dépend de γ , de la réaction et des conditions extérieures. Si nous considérons à présent le réseau de réactions :



alors nous obtenons :

$$\dot{c}_c = k_1 c_a^\alpha c_b^\beta - k_2 c_c^\gamma \quad (5.3)$$

Comme nous le voyons sur cet exemple, les lois de puissance arrivent naturellement dans l'étude de la dynamique de réactions ou de réseaux de réactions chimiques. Dans cet exemple particulier, l'équation différentielle vérifiée par la variable c_c a exactement la forme d'un S-système. Pour des réseaux de réactions plus complexes, on sera éventuellement amenés à considérer une somme de plus de deux monômes. Dans ce cas, nous retombons sur le formalisme QM précédemment décrit. Cette exemple nous permet de justifier la terminologie que nous avons définie dans la définition 5.2 : les coefficients multiplicatifs sont appelés *taux* cinétiques et les exposants *ordres* cinétiques.

Dans cet exemple, les exposants α , β , γ , δ et ϵ sont des coefficients stœchiométriques et sont donc supposés entiers. Néanmoins, comme nous allons le voir par la suite, les supposer à valeurs dans \mathbb{R} permet de considérer une plus grande classe de systèmes, et donc une plus grande variété de réactions chimiques. En effet, si nous supposons que la réaction précédente est catalysé par un enzyme, alors les équations de la dynamique doivent tenir compte de ce fait en incluant un terme représentant cette catalyse : le plus souvent on utilise l'équation de Michaelis-Menten que nous avons rappelée dans le premier chapitre. Comme nous l'avons vu, le terme de Michaelis est un terme fractionnel qui ne rentre pas dans le formalisme des S-systèmes. Dans la suite de ce chapitre, nous proposons une méthode (appelée S-approximation) permettant d'approximer localement n'importe quel champ de vecteurs positifs par un champ monomial du type :

$$\alpha_i \prod_{j=1}^n x_j^{g_{ij}}$$

où $g_{ij} \in \mathbb{R}$. En appliquant cette méthodologie au terme de Michaelis, nous obtenons alors une approximation de l'équation de \dot{c}_c sous la forme (5.3). Nous reviendrons sur l'approximation du terme de Michaelis dans l'exemple 5.2

5.1.3 Etude des points d'équilibres d'un S-système

Afin d'étudier les points d'équilibre du système $\mathcal{S}(\alpha, \beta, G, H)$ (5.1), nous devons résoudre dans Ω le système d'équations :

$$\alpha_i \prod_{j=1}^n x_j^{g_{ij}} = \beta_i \prod_{j=1}^n x_j^{h_{ij}}, \quad i = 1 \dots n$$

qui se ramène à :

$$\prod_{j=1}^n x_j^{g_{ij} - h_{ij}} = \frac{\beta_i}{\alpha_i}, \quad i = 1 \dots n$$

D'où, en posant $b_i = \frac{\beta_i}{\alpha_i}$:

$$x^{G-H} = b$$

En prenant le logarithme népérien composante par composante, et en utilisant la proposition 5.1, nous obtenons :

$$(G - H) \ln x = \ln b$$

Trouver les points d'équilibre d'un S-système revient donc algorithmiquement à résoudre un système linéaire de matrice $G - H$. On a la proposition suivante :

Proposition 5.2 *Le S-système $\mathcal{S}(\alpha, \beta, G, H)$ admet un unique point d'équilibre positif (i.e. dans Ω) si et seulement si la matrice $G - H$ est inversible. Ce point d'équilibre, noté \tilde{x} est obtenu par la formule :*

$$\tilde{x} = b^{(G-H)^{-1}} \quad (5.4)$$

où b est le vecteur $(\beta_i/\alpha_i)_{i=1, \dots, n}$.

Nous supposons dans la suite que $G - H$ est inversible. Nous cherchons à présent à analyser la stabilité du point d'équilibre \tilde{x} . Pour cela, nous étudions tout d'abord le cas unidimensionnel ($n = 1$).

Un S-système unidimensionnel est constitué d'une équation différentielle scalaire :

$$\dot{x} = f(x) = \alpha x^g - \beta x^h$$

où $\alpha, \beta > 0$ et $g, h \in \mathbb{R}$. Le système $\mathcal{S}(\alpha, \beta, g, h)$ admet un unique point d'équilibre $\tilde{x} > 0$ à la condition nécessaire et suffisante que $g - h \neq 0$. La stabilité linéaire (voir [83]) de \tilde{x} dépend du signe de la dérivée de f en \tilde{x} :

$$\begin{aligned} \frac{\partial f}{\partial x}(\tilde{x}) &= \alpha g \tilde{x}^{g-1} - \beta h \tilde{x}^{h-1} \\ &= \alpha g \tilde{x}^{g-1} - \alpha h \tilde{x}^{g-1} \quad (\text{par définition de } \tilde{x}) \\ &= \alpha \tilde{x}^{g-1} (g - h) \end{aligned}$$

La stabilité de \tilde{x} dépend donc directement du signe de $g - h$. Si $g - h < 0$, alors \tilde{x} est linéairement stable donc asymptotiquement stable, si $g - h > 0$, alors \tilde{x} est instable.

Considérons à présent le cas où la dimension n est supérieure à 1. Dans ce cas, l'analyse de la stabilité linéaire de \tilde{x} est basé sur l'étude du spectre de la matrice jacobienne de F en \tilde{x} . Nous notons $J_F(\tilde{x})$ la jacobienne du champ de vecteurs F en \tilde{x} :

$$J_F(\tilde{x}) = \left(\frac{\partial f_i}{\partial x_j}(\tilde{x}) \right)_{i,j \in \{1, \dots, n\}}$$

En différentiant F , nous obtenons, pour tout $i, j \in \{1, \dots, n\}$:

$$\frac{\partial f_i}{\partial x_j}(\tilde{x}) = \frac{\alpha_i}{\tilde{x}_j} \left(\prod_{k=1}^n \tilde{x}_k^{g_{ik}} \right) (g_{ij} - h_{ij}) \quad (5.5)$$

Cette formule nous permet de lier la jacobienne $J_F(\tilde{x})$ à la matrice $G - H$. Nous rappelons ici la définition de la stabilité et de la semi-stabilité de matrices :

Définition 5.3 Une matrice carrée A de dimension n à coefficients réels est dite stable (respectivement semi-stable) si toutes ses valeurs propres λ_i , pour i allant de 1 à n ont une partie réelle strictement négative (respectivement négative ou nulle).

Remarque 5.1 Grâce à cette définition, la notion de stabilité linéaire d'un point d'équilibre peut s'exprimer comme suit. Soit un système dynamique autonome de dimension n :

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

où G est de classe \mathcal{C}^1 sur un ouvert U de \mathbb{R}^n , et soit $x_0 \in U$ un point d'équilibre isolé de ce système. L'équilibre x_0 est linéairement stable si et seulement si la matrice $J_G(x_0)$ est stable.

Revenons à présent au S-système $\mathcal{S}(\alpha, \beta, G, H)$. Grâce à la formule 5.5 reliant la jacobienne $J_F(\tilde{x})$ à la matrice $G - H$, nous cherchons à déterminer quelles propriétés de $G - H$ permettent d'assurer la stabilité de l'équilibre \tilde{x} . En observant le cas unidimensionnel, on pourrait espérer que la stabilité de $G - H$ suffise à assurer la stabilité de $J_F(\tilde{x})$ et donc celle de \tilde{x} . Ce n'est néanmoins pas le cas, comme le montre l'exemple suivant :

Exemple 5.1

Soit $n = 2$. Nous considérons le S-système (S_1) défini par :

$$(S_1) : \begin{cases} \dot{x} &= 3xy^2 - 2x^4 \\ \dot{y} &= 4x^3y^4 - x^5y^3 \end{cases}$$

Le système (S_1) est donc le S-système $\mathcal{S}(\alpha, \beta, G, H)$, avec :

$$\alpha = \begin{pmatrix} 3 \\ 4 \end{pmatrix}, \beta = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, G = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}, H = \begin{pmatrix} 4 & 0 \\ 5 & 3 \end{pmatrix}$$

La matrice $G - H$ est égale à :

$$G - H = \begin{pmatrix} -3 & 2 \\ -2 & 1 \end{pmatrix}$$

Le polynôme caractéristique de cette matrice peut se calculer simplement :

$$\chi_{G-H}(\lambda) = (\lambda + 1)^2$$

Ce polynôme comporte une racine double : $\lambda = -1$ strictement négative. La matrice $G - H$ est donc une matrice stable. Puisqu'elle est inversible, (S_1) admet un unique point d'équilibre \tilde{x} dans Ω . Ce point peut se calculer par la formule 5.4 :

$$b = \begin{pmatrix} \frac{2}{3} \\ \frac{1}{4} \end{pmatrix}, \quad (G - H)^{-1} = \begin{pmatrix} 1 & -2 \\ 2 & -3 \end{pmatrix}$$

D'où :

$$\begin{aligned} \tilde{x} &= \left(\left(\frac{2}{3} \right) \left(\frac{1}{4} \right)^{-2}, \left(\frac{2}{3} \right)^2 \left(\frac{1}{4} \right)^{-3} \right) \\ &= \left(\frac{32}{3}, \frac{256}{9} \right) \end{aligned}$$

En calculant la jacobienne du système (S_1) en \tilde{x} , nous trouvons qu'elle admet deux valeurs propres positives, entraînant l'instabilité de l'équilibre \tilde{x} .

La stabilité de $G - H$ étant insuffisante pour déterminer la stabilité de \tilde{x} , nous allons utiliser une notion plus forte qui est la notion de stabilité signée (voir [37] et [49]).

Définition 5.4 Soient A et B deux matrices réelles carrées de dimension n . On dira que A et B ont le même patron de signes si :

$$\forall i, j \in \{1, \dots, n\}, \quad \text{sgn}(a_{ij}) = \text{sgn}(b_{ij})$$

Où sgn désigne la fonction signe classique :

$$\forall x \in \mathbb{R}, \quad \text{sgn}(x) = \begin{cases} +1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$$

Définition 5.5 Soit $A \in \mathfrak{M}_n(\mathbb{R})$. A est dite signée stable si toute matrice $B \in \mathfrak{M}_n(\mathbb{R})$ ayant le même patron de signes est stable. De manière analogue, on dira que A est signée semi-stable si toute matrice B ayant le même patron de signes est semi-stable.

Nous trouvons dans [37] une caractérisation intéressante de la semi-stabilité signée :

Théorème 5.3 (Quirk-Ruppert-Maybee) Soit A une matrice carrée réelle de dimension n .

A est signée semi-stable si et seulement si elle vérifie les trois conditions suivantes :

- (i) $\forall i \in \{1, \dots, n\}, \quad a_{ii} \leq 0$
- (ii) $\forall i, j \in \{1, \dots, n\}, \quad (i \neq j \implies a_{ij}a_{ji} \leq 0)$
- (iii) Pour chaque séquence de $k \geq 3$ indices distincts de $\{1, \dots, n\} : (i_1, \dots, i_k)$, on a :

$$a_{i(1)i(2)} \cdots a_{i(k-1)i(k)} a_{i(k)i(1)} = 0$$

La troisième condition est équivalente au fait que le graphe associé à A n'admette aucun cycle de longueur supérieure ou égale à 3 (voir [37]).

Grâce à cette notion, nous pouvons donner un critère sur la matrice $G - H$ permettant d'assurer la stabilité du point d'équilibre \tilde{x} :

Proposition 5.4 Soit $n \in \mathbb{N}^*$, $\alpha, \beta \in \Omega$ et $G, H \in \mathfrak{M}_n(\mathbb{R})$. On suppose que $G - H$ est inversible et on note \tilde{x} l'unique point d'équilibre positif du S-système $\mathcal{S}(\alpha, \beta, G, H)$. On suppose également que \tilde{x} est un point d'équilibre hyperbolique (c'est-à-dire qu'aucune valeur propre de $J_F(\tilde{x})$ n'a de partie réelle nulle).

Si la matrice $G - H$ est signée semi-stable (i.e. si elle vérifie les conditions (i), (ii) et (iii) du théorème précédent), alors \tilde{x} est un point d'équilibre asymptotiquement stable.

Démonstration

Nous notons $P = G - H$. La formule (5.5) nous conduit à :

$$\forall i, j \in \{1, \dots, n\}, \quad \frac{\partial f_i}{\partial x_j}(\tilde{x}) = \frac{\gamma_i}{\tilde{x}_j} p_{ij} \quad (5.6)$$

avec $\gamma_i = \alpha_i \prod_{k=1}^n \tilde{x}_k^{g_{ik}}$. Etant donné que pour tout $i, j \in \{1, \dots, n\}$, on a $\gamma_i > 0$ et $\tilde{x}_j > 0$, on

déduit de la formule précédente que $J_F(\tilde{x})$ et P ont le même patron de signe. P étant supposée signée semi-stable, $J_F(\tilde{x})$ est semi-stable. \tilde{x} étant hyperbolique, $J_F(\tilde{x})$ est donc stable, ce qui entraîne que \tilde{x} est linéairement stable donc qu'il est asymptotiquement stable. ■

Remarque 5.2 En notation matricielle, la formule (5.6) s'exprime ainsi :

$$J_F(\tilde{x}) = \Gamma P D^{-1}$$

où :

$$\Gamma = \begin{pmatrix} \gamma_1 & & \\ & \ddots & \\ & & \gamma_n \end{pmatrix}, \quad D = \begin{pmatrix} \tilde{x}_1 & & \\ & \ddots & \\ & & \tilde{x}_n \end{pmatrix}$$

sont des matrices inversibles. P étant supposée inversible, ceci entraîne que la jacobienne $J_F(\tilde{x})$ n'a pas de valeurs propres nulles. Il nous a toutefois fallu supposer l'hyperbolicité de \tilde{x} afin d'assurer qu'aucune valeur propre de $J_F(\tilde{x})$ ne soit imaginaire pure. L'hypothèse de l'hyperbolicité de \tilde{x} est donc une hypothèse essentielle dans la proposition précédente.

5.2 S-approximation et recherche d'équilibres positifs

L'étude des S-systèmes est intéressante car ils constituent un bon outil d'approximation de systèmes dynamiques issues de la biologie. Ils fournissent notamment une expression canonique non linéaire de champs de vecteurs positifs (voir [81]). Nous allons voir que cette expression est utile car elle est à la base d'un algorithme itératif de recherche d'équilibres positifs d'un système dynamique autonome quelconque (voir [79]).

5.2.1 Approximation monomiale d'un champ de vecteurs positifs

Nous considérons un champ de vecteur positif V :

$$V : (\mathbb{R}_+^*)^n \longrightarrow (\mathbb{R}_+^*)^n$$

$$x \longmapsto \begin{pmatrix} v_1(x_1, \dots, x_n) \\ \vdots \\ v_n(x_1, \dots, x_n) \end{pmatrix}$$

Nous supposons que V est de classe \mathcal{C}^1 sur $\Omega = (\mathbb{R}_+^*)^n$. Nous considérons également le changement de variables suivant : pour tout vecteur $x \in \Omega$, nous posons $y = \ln(x)$, où \ln désigne le logarithme népérien pris composante par composante (voir proposition 5.1). On a alors : $x = e^y$, où e^y désigne le vecteur $(e^{y_1}, \dots, e^{y_n})$ appartenant à Ω . En appliquant \ln à $V(x)$, nous obtenons :

$$\forall x \in \Omega, \quad \ln(V(x)) = \ln(V(e^y)) \stackrel{\text{def}}{=} W(y)$$

La fonction $W : \mathbb{R}^n \rightarrow \mathbb{R}^n$ est de classe \mathcal{C}^1 sur \mathbb{R}^n . Pour tout $i \in \{1, \dots, n\}$, nous notons w_i la i -ième coordonnée de W . Nous considérons un point arbitraire $y^0 \in \mathbb{R}^n$; W étant de classe \mathcal{C}^1 sur \mathbb{R}^n , nous pouvons écrire son développement de Taylor à l'ordre 1 en y^0 :

$$\forall i = 1 \dots n, \quad w_i(y) = w_i(y^0) + \sum_{j=1}^n (y_j - y_j^0) \frac{\partial w_i}{\partial y_j}(y^0) + o_{y \rightarrow y^0}(\|y - y^0\|)$$

Pour i parcourant $\{1, \dots, n\}$, introduisons les fonctions :

$$\begin{aligned} \tilde{w}_i(y) &\stackrel{\text{def}}{=} w_i(y^0) + \sum_{j=1}^n (y_j - y_j^0) \frac{\partial w_i}{\partial y_j}(y^0) \\ \tilde{v}_i(x) &\stackrel{\text{def}}{=} \exp(\tilde{w}_i(y)) \end{aligned}$$

On a alors :

$$\begin{aligned} \tilde{v}_i(x) &= e^{\tilde{w}_i(y)} \\ &= e^{w_i(y^0)} \exp\left(\sum_{j=1}^n (y_j - y_j^0) \frac{\partial w_i}{\partial y_j}(y^0)\right) \\ &= e^{w_i(y^0)} \prod_{j=1}^n \exp\left((y_j - y_j^0) \frac{\partial w_i}{\partial y_j}(y^0)\right) \end{aligned}$$

Etant donné que $y = \ln(x)$ et que $w_i(y) = \ln(v_i(x))$, nous déduisons, en posant $x^0 = e^{y^0}$:

$$\tilde{v}_i(x) = v_i(x^0) \prod_{j=1}^n \left(\frac{x_j}{x_j^0}\right)^{\frac{\partial w_i}{\partial y_j}(y^0)}$$

Calculons à présent les dérivées partielles de w_i . On considère i et j deux entiers de $\{1, \dots, n\}$:

$$\begin{aligned} \frac{\partial w_i}{\partial y_j}(y) &= \frac{\partial}{\partial y_j} (\ln(v_i(e^y))) \\ &= \frac{1}{v_i(e^y)} \frac{\partial}{\partial y_j} (v_i(e^y)) \\ &= \frac{1}{v_i(x)} e^{y_j} \frac{\partial v_i}{\partial x_j}(e^y) \\ &= \frac{x_j}{v_i(x)} \frac{\partial v_i}{\partial x_j}(x) \end{aligned}$$

Nous avons ainsi défini un champ de vecteur positif \tilde{V} . Ce champ de vecteur est un champ monomial, que nous allons utiliser comme approximation du champ de départ V .

Définition 5.6 Soit un champ de vecteurs positifs $V : \Omega \rightarrow \Omega$ de classe \mathcal{C}^1 sur Ω , et un vecteur quelconque x^0 appartenant à Ω . La S-approximation de V en x^0 est le champ de vecteurs $\tilde{V}_{x^0} : \Omega \rightarrow \Omega$ défini par :

$$\forall i \in \{1, \dots, n\}, \quad (\tilde{V}_{x^0})_i(x) = \alpha_i(x^0) \prod_{j=1}^n x_j^{g_{ij}(x^0)} \quad (5.7)$$

$$\text{avec : } \begin{cases} \alpha_i(x^0) &= v_i(x^0) \prod_{j=1}^n (x_j^0)^{-g_{ij}(x^0)} \\ g_{ij}(x^0) &= \frac{x_j^0}{v_i(x^0)} \frac{\partial v_i}{\partial x_j}(x^0) \end{cases} \quad (5.8)$$

Une simple manipulation de ces formules permet d'énoncer le résultat suivant :

Proposition 5.5 Soit $x^0 \in \Omega$ et soit V un champ de vecteurs positifs. On note \tilde{V}_{x^0} la S-approximation de V en x^0 . Nous avons les deux égalités suivantes :

$$(i) \quad \tilde{V}_{x^0}(x^0) = V(x^0)$$

$$(ii) \quad \forall i, j \in \{1, \dots, n\}, \quad \frac{\partial \tilde{v}_i}{\partial x_j}(x^0) = \frac{\partial v_i}{\partial x_j}(x^0)$$

$$\text{ou, de manière équivalente : } J_V(x^0) = J_{\tilde{V}_{x^0}}(x^0).$$

Exemple 5.2

Posons $n = 1$. Nous considérons le champ de vecteur donné par :

$$\begin{aligned} V : \mathbb{R}_+ &\longrightarrow \mathbb{R}_+ \\ x &\longmapsto \frac{V_m x}{K_M + x} \end{aligned}$$

où V_m et K_M sont des constantes positives. Il s'agit du terme de Michaelis-Menten engendré par une réaction enzymatique simple (voir chapitre 1). Etant donné un réel positif x^0 , la S-approximation de V en x^0 se calcule aisément :

$$\tilde{V}_{x^0}(x) = \alpha x^g \quad (5.9)$$

avec :

$$\begin{cases} g &= \frac{K_M}{K_M + x^0} \\ \alpha &= \frac{V_m x^0}{K_M + x^0} (x^0)^{-g} \end{cases}$$

Par exemple, pour $x^0 = K_M$ et $x^0 = 10 K_M$ (voir figure 5.2), on obtient :

$$\begin{aligned} \tilde{V}_{K_M}(x) &= \frac{V}{2} \sqrt{\frac{x}{K_M}} \\ \tilde{V}_{10 K_M}(x) &= \frac{10V}{11} (10K_M)^{-\frac{10}{11}} x^{\frac{10}{11}} \end{aligned}$$

Selon où la concentration en substrat se situe par rapport à K_M (i.e. selon si l'enzyme est proche ou non de la saturation), nous pouvons donc exhiber différentes approximations du terme de Michaelis sous la forme d'un monôme (5.9).

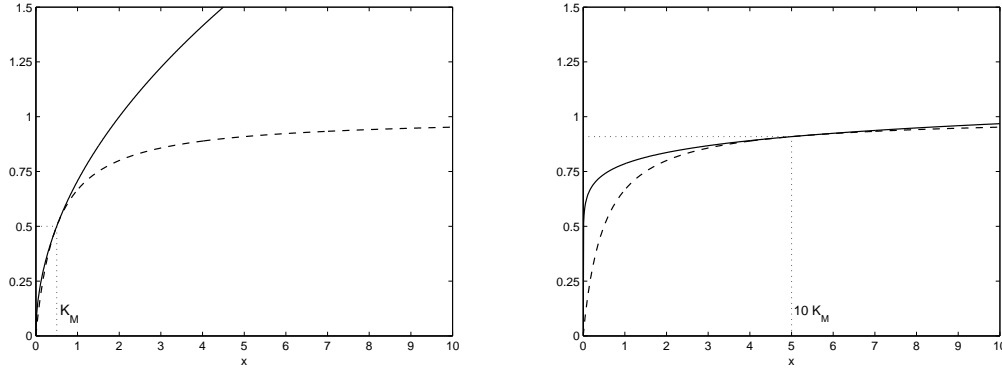


FIG. 5.1: *S*-approximations de la fonction de Michaelis (constantes : $V_m = 1$ et $K_M = 0.5$). La courbe en pointillet est la courbe de Michaelis, les courbes en trait plein sont les *S*-approximations respectives en $x^0 = K_M$ (à gauche) et en $x^0 = 10K_M$ (à droite).

5.2.2 Algorithme de recherche des points d'équilibres positifs

Nous considérons un système dynamique de dimension $n \in \mathbb{N}^*$ de la forme :

$$\dot{x} = V^+(x) - V^-(x) \quad (5.10)$$

où la variable $x(t)$ appartient à $\Omega = (\mathbb{R}_+^*)^n$ et où V^+ et V^- sont des champs de vecteurs positifs : $V^+, V^- : \Omega \rightarrow \Omega$. Ces champs de vecteurs sont supposés de classe \mathcal{C}^1 sur Ω . Pour chaque variable x_i , $i \in \{1, \dots, n\}$, le terme $v_i^+(x)$ correspond à un terme de production et le terme $v_i^-(x)$ à un terme de dégradation. Du point de vue de la modélisation de phénomène physique ou biologique, ces systèmes sont assez généraux et englobent plusieurs types de modèles (réseaux de réactions chimiques, systèmes proie-prédateurs de Lotka-Volterra, etc.)

Nous supposons que le système (5.10) admet au moins un point d'équilibre positif. Nous proposons dans cette partie un algorithme basé sur la théorie des *S*-systèmes dont le but est de trouver les points d'équilibre de (5.10) qui sont dans Ω . Pour cela, nous considérons, étant donné $x^0 \in \Omega$ les *S*-approximations $\tilde{V}_{x^0}^+$ et $\tilde{V}_{x^0}^-$ de V^+ et de V^- en x^0 . Nous notons alors (S_{x^0}) le *S*-système défini par :

$$(S_{x^0}) \quad \dot{x} = \tilde{V}_{x^0}^+(x) - \tilde{V}_{x^0}^-(x)$$

En utilisant (5.7) et (5.8), nous pouvons exprimer (S_{x^0}) sous la forme suivante :

$$(S_{x^0}) \quad \dot{x}_i = \alpha_i(x^0) \prod_{j=1}^n x_j^{g_{ij}(x^0)} - \beta_i(x^0) \prod_{j=1}^n x_j^{h_{ij}(x^0)}, \quad i = 1 \dots n$$

où les paramètres $\alpha_i(x^0)$, $g_{ij}(x^0)$, $\beta_i(x^0)$ et $h_{ij}(x^0)$ sont donnés par :

$$\begin{cases} \alpha_i(x^0) &= v_i^+(x^0) \prod_{j=1}^n (x_j^0)^{-g_{ij}(x^0)} \\ \beta_i(x^0) &= v_i^-(x^0) \prod_{j=1}^n (x_j^0)^{-h_{ij}(x^0)} \end{cases}$$

et :

$$\begin{cases} g_{ij}(x^0) &= \frac{x_j^0}{v_i^+(x^0)} \frac{\partial v_i^+}{\partial x_j}(x^0) \\ h_{ij}(x^0) &= \frac{x_j^0}{v_i^-(x^0)} \frac{\partial v_i^-}{\partial x_j}(x^0) \end{cases}$$

Le principe de l'algorithme que nous proposons est le suivant. Etant donné un point x^0 quelconque dans Ω , nous calculons les quantités $\alpha_i(x^0)$, $g_{ij}(x^0)$, $\beta_i(x^0)$ et $h_{ij}(x^0)$, nous permettant ainsi d'exprimer le S-système (S_{x^0}) . Dans le cas où ce S-système n'est pas dégénéré, c'est-à-dire dans le cas où la matrice $P(x^0) = G(x^0) - H(x^0)$ est inversible, il admet un unique point d'équilibre dans Ω . Nous notons ce point x^1 . Nous définissons la fonction $\Psi : \Omega \rightarrow \Omega$ qui, à tout $x^0 \in \Omega$ associe le point x^1 ainsi construit.

La méthode consiste ensuite à itérer l'application Ψ :

$$\begin{cases} x^0 \in \Omega \\ x^{m+1} = \Psi(x^m), \text{ pour tout } m \in \mathbb{N} \end{cases}$$

et à étudier la convergence des suites (x^m) ainsi trouvées. Nous montrons alors que les limites de ces suites (pour différentes valeurs du point initial x^0) sont les points d'équilibre positifs du système initial (5.10). L'algorithme 4 ci-après reprend ce mécanisme et renvoie, pour une précision ε donnée, un point x_m tel qu'il existe un point d'équilibre $\tilde{x} \in \Omega$ du système (5.10) tel que :

$$\|\tilde{x} - x_m\| \leq \varepsilon$$

Le point essentiel est que simultanément, l'algorithme calcule le S-système (S_{x^m}) qui est une approximation du système initial au voisinage du point \tilde{x} .

Preuve de la correction de l'algorithme

Afin de prouver la correction de cet algorithme, nous allons montrer d'une part qu'à moins de tomber sur un cas dégénéré, la suite (x^m) qu'il calcule est bien convergente, et d'autre part que la limite de cette suite est bien un équilibre positif du système (5.10) initial.

Nous notons F le champ de vecteur $V^+ - V^-$ et (S) le système différentiel :

$$(S) \quad \dot{x} = F(x) = V^+(x) - V^-(x)$$

Nous rappelons que, étant donné $y \in \Omega$, nous notons (S_y) le S-système de dimension n :

$$(S_y) \quad \dot{x} = \tilde{V}_y^+(x) - \tilde{V}_y^-(x)$$

où \tilde{V}_y^+ et \tilde{V}_y^- sont les S-approximations de V^+ et V^- en y . En utilisant (5.7) et (5.8), nous pouvons exprimer (S_y) sous la forme suivante :

$$(S_y) : \quad \dot{x}_i = \alpha_i(y) \prod_{j=1}^n x_j^{g_{ij}(y)} - \beta_i(y) \prod_{j=1}^n x_j^{h_{ij}(y)}, \quad i = 1 \dots n$$

avec :

$$\begin{cases} g_{ij}(y) &= \frac{y_j^0}{v_i^+(y)} \frac{\partial v_i^+}{\partial x_j}(y) \\ h_{ij}(y) &= \frac{y_j^0}{v_i^-(y)} \frac{\partial v_i^-}{\partial x_j}(y) \end{cases} \quad (5.11)$$

Algorithme 4 - Recherche de points d'équilibres positifs de systèmes dynamiques.

Etant donné un système dynamique (5.10) et un point initial, cet algorithme symbolique-numérique renvoie un point d'équilibre positif du système initial (5.10) ainsi qu'un S-système approximant le système initial autour de ce point d'équilibre.

n : Entier naturel ;
 x^0 : Vecteur appartenant à Ω ;
 $V^+, V^- : \Omega \rightarrow \Omega$, champs de vecteurs de classe \mathcal{C}^1 ;
 ε : Réel positif (précision) ;

$X \leftarrow x^0$;

Répéter

$Y \leftarrow X$;

Pour i de 1 à n faire

Pour j de 1 à n faire

$$g_{ij} \leftarrow \frac{X_j}{v_i^+(X)} \frac{\partial v_i^+}{\partial X_j} ;$$

$$h_{ij} \leftarrow \frac{X_j}{v_i^-(X)} \frac{\partial v_i^-}{\partial X_j} ;$$

Fin Pour

$$\alpha_i \leftarrow v_i^+(X) \prod_{j=1}^n (X_j)^{-g_{ij}} ;$$

$$\beta_i \leftarrow v_i^-(X) \prod_{j=1}^n (X_j)^{-h_{ij}} ;$$

$$b_i \leftarrow \frac{\beta_i}{\alpha_i}$$

Fin Pour

Si $(\det(G - H) \neq 0)$ **Alors**

$$| X \leftarrow b^{(G-H)^{-1}} ;$$

Sinon

 cas dégénéré : échec de l'algorithme \rightarrow relancer avec un nouveau x^0

Fin Si

jusqu'à ce que $(\|X - Y\| < \varepsilon)$

Retourner X

et :

$$\begin{cases} \alpha_i(y) &= v_i^+(y) \prod_{j=1}^n (y_j^0)^{-g_{ij}(y)} \\ \beta_i(y) &= v_i^-(y) \prod_{j=1}^n (y_j^0)^{-h_{ij}(y)} \end{cases} \quad (5.12)$$

Nous notons $G(y)$ et $H(y)$ les matrices :

$$G(y) = [g_{ij}(y)]_{i,j \in \{1, \dots, n\}} , \quad H(y) = [h_{ij}(y)]_{i,j \in \{1, \dots, n\}}$$

Puisque nous avons supposé les champs de vecteurs V^+ et V^- de classe \mathcal{C}^1 , nous en déduisons que $\alpha(y)$, $\beta(y)$, $G(y)$ et $H(y)$ sont continues.

En faisant l'hypothèse que $\det(G(y) - H(y)) \neq 0$, nous déduisons de la proposition 5.2 que le système (S_y) admet un unique point d'équilibre positif. Nous notons ce point $\Psi(y)$. D'après (5.4), nous avons :

$$\begin{aligned} \Psi : \Omega &\longrightarrow \Omega \\ y &\longmapsto b(y)^{(G(y)-H(y))^{-1}} \end{aligned} \quad (5.13)$$

où $b(y) = \left(\frac{\beta_i(y)}{\alpha_i(y)} \right)$. L'application Ψ est définie sur l'ensemble :

$$D_\Psi = \{y \in \Omega \mid \det(G(y) - H(y)) \neq 0\}$$

Grâce à la continuité de G , H ainsi que de la fonction \det , nous déduisons que D_Ψ est un ouvert de Ω .

L'algorithme 4 effectue l'itération de Ψ à partir d'un point $x^0 \in \Omega$ donné. En d'autres termes, il calcule la suite :

$$x^m = \Psi^m(x^0) , \quad m \in \mathbb{N}$$

Nous savons que la limite d'une telle suite (si elle converge) est un point fixe de l'application Ψ . La preuve de la correction repose donc sur le fait que les points fixes de Ψ sont les points d'équilibres de (S) , et qu'ils sont bien attractifs, c'est-à-dire que pour chaque point d'équilibre \tilde{x} de (S) , il existe un x^0 tel que la suite $\Psi^m(x^0)$ converge vers \tilde{x} . Avant de montrer ceci, nous montrons le lemme suivant.

Lemme 5.6 *Si \bar{x} est un point d'équilibre positif hyperbolique du système (S) , alors $\bar{x} \in D_\Psi$.*

Démonstration

Soit \bar{x} un point d'équilibre positif hyperbolique du système (S) . Nous notons $P(\bar{x})$ la matrice $G(\bar{x}) - H(\bar{x})$. En utilisant (5.11), on a :

$$\begin{aligned} p_{ij}(\bar{x}) &= g_{ij}(\bar{x}) - h_{ij}(\bar{x}) \\ &= \bar{x}_j \left(\frac{1}{v_i^+(\bar{x})} \frac{\partial v_i^+}{\partial x_j}(\bar{x}) - \frac{1}{v_i^-(\bar{x})} \frac{\partial v_i^-}{\partial x_j}(\bar{x}) \right) \end{aligned}$$

Comme \bar{x} est un point d'équilibre de (S) , par définition $V^+(\bar{x}) = V^-(\bar{x})$, donc :

$$\begin{aligned} p_{ij}(\bar{x}) &= \frac{\bar{x}_j}{v_i^+(\bar{x})} \left(\frac{\partial v_i^+}{\partial x_j}(\bar{x}) - \frac{\partial v_i^-}{\partial x_j}(\bar{x}) \right) \\ &= \frac{\bar{x}_j}{v_i^+(\bar{x})} \frac{\partial f_i}{\partial x_j}(\bar{x}) \end{aligned}$$

En notation matricielle, cette égalité donne :

$$P(\bar{x}) = \Delta^{-1} J_F(\bar{x}) D$$

où $J_F(\bar{x})$ est la jacobienne de F évaluée en \bar{x} et Δ et D les matrices diagonales :

$$\Delta = \begin{pmatrix} v_1^+(\bar{x}) & & \\ & \ddots & \\ & & v_n^+(\bar{x}) \end{pmatrix}, \quad D = \begin{pmatrix} \bar{x}_1 & & \\ & \ddots & \\ & & \bar{x}_n \end{pmatrix}$$

Puisque nous avons supposé \bar{x} hyperbolique, cela entraîne que $J_F(\bar{x})$ est inversible, et donc que $P(\bar{x})$ est inversible. On en déduit donc que \bar{x} appartient bien au domaine D_Ψ de Ψ . ■

La correction de l'algorithme découle des deux propositions suivantes :

Proposition 5.7 *Les équilibres positifs hyperboliques du système (S) sont les points fixes de l'application Ψ .*

Démonstration

Soit \bar{x} un point fixe de Ψ . On a, d'après (5.13) :

$$\Psi(\bar{x}) = b(\bar{x})^{(G(\bar{x})-H(\bar{x}))^{-1}}$$

où $b(\bar{x})$ est le vecteur $(\beta_i(\bar{x})/\alpha_i(\bar{x}))_{i \in \{1, \dots, n\}}$.

On a alors les équivalences suivantes :

$$\begin{aligned} \Psi(\bar{x}) = \bar{x} &\iff b(\bar{x})^{(G(\bar{x})-H(\bar{x}))^{-1}} = \bar{x} \\ &\iff b(\bar{x}) = \bar{x}^{(G(\bar{x})-H(\bar{x}))} \\ &\iff \forall i = 1 \dots n, \quad \frac{\beta_i(\bar{x})}{\alpha_i(\bar{x})} = \prod_{j=1}^n \bar{x}_j^{g_{ij}(\bar{x})-h_{ij}(\bar{x})} \\ &\iff \forall i = 1 \dots n, \quad \beta_i(\bar{x}) \prod_{j=1}^n \bar{x}_j^{h_{ij}(\bar{x})} = \alpha_i(\bar{x}) \prod_{j=1}^n \bar{x}_j^{g_{ij}(\bar{x})} \quad (\text{car } \bar{x} \in \Omega) \end{aligned}$$

Or, par définition :

$$\alpha_i(\bar{x}) \prod_{j=1}^n \bar{x}_j^{g_{ij}(\bar{x})} = \tilde{V}_x^+(\bar{x}) \quad \text{et} \quad \beta_i(\bar{x}) \prod_{j=1}^n \bar{x}_j^{h_{ij}(\bar{x})} = \tilde{V}_x^-(\bar{x})$$

Ceci entraîne $\tilde{V}_x^+(\bar{x}) = \tilde{V}_x^-(\bar{x})$. En utilisant la proposition 5.5, nous en déduisons que :

$$\Psi(\bar{x}) = \bar{x} \iff V^+(\bar{x}) = V^-(\bar{x})$$

■

Proposition 5.8 *Nous supposons que les champs V^+ et V^- sont de classe \mathcal{C}^2 sur Ω . Etant donné un point d'équilibre hyperbolique \bar{x} de (S), il existe un point $x^0 \in \Omega$ tel que la suite $(\Psi^m(x^0))_{m \in \mathbb{N}}$ converge vers \bar{x} . En d'autres termes, les points d'équilibres hyperboliques de (S) sont des points fixes attractifs de Ψ .*

Démonstration

Afin de prouver ce lemme, nous allons utiliser un critère de point fixe classique, connu sous le nom de théorème d'Ostrowski (voir par exemple [18]) :

Si la fonction Ψ est contractante sur un ouvert W et si $\bar{x} \in W$ est un point fixe de Ψ , alors \bar{x} est l'unique point fixe de Ψ dans W et il est attractif, c'est-à-dire que pour tout $x^0 \in W$, l'itération de Ψ à partir de x^0 converge vers \bar{x} .

Considérons un point fixe \bar{x} de Ψ . Pour démontrer que \bar{x} est attractif, il nous suffit de montrer qu'il existe un voisinage ouvert de \bar{x} dans lequel Ψ est contractante. Afin de montrer la contractance de Ψ , nous allons démontrer que les valeurs propres de la jacobienne de Ψ évaluée en \bar{x} sont toutes de modules strictement inférieur à 1. Puisque \bar{x} est un point d'équilibre hyperbolique de (S) , nous déduisons du lemme 5.6 que :

$$\det(G(\bar{x}) - H(\bar{x})) \neq 0$$

Par continuité du déterminant, il existe un voisinage ouvert W de \bar{x} tel que :

$$\forall x \in W, \quad \det(G(x) - H(x)) \neq 0$$

Puisque nous avons supposé V^+ et V^- de classe \mathcal{C}^2 , Ψ est différentiable sur W . En utilisant (5.12) et (5.11), et en posant :

$$\begin{cases} U^+ &= \log(V^+) \\ U^- &= \log(V^-) \\ U &= U^+ - U^- = \log\left(\frac{V^+}{V^-}\right) \\ P &= G - H \end{cases}$$

nous obtenons, pour tout $x \in W$:

$$\Psi_i(x) = x_i \prod_{j=1}^n \left(\frac{v_j^-(x)}{v_j^+(x)} \right)^{p_{ij}^{(-1)}(x)} \quad (5.14)$$

où les $p_{ij}^{(-1)}(x)$ désignent les coefficients de la matrice P^{-1} .

Calculons les coefficients p_{ij} :

$$\begin{aligned} p_{ij}(x) &= g_{ij}(x) - h_{ij}(x) \\ &= x_j \left(\frac{1}{v_i^+(x)} \frac{\partial v_i^+}{\partial x_j}(x) - \frac{1}{v_i^-(x)} \frac{\partial v_i^-}{\partial x_j}(x) \right) \\ &= x_j \left(\frac{\partial u_i^+}{\partial x_j}(x) - \frac{\partial u_i^-}{\partial x_j}(x) \right) \\ &= x_j \frac{\partial u_i}{\partial x_j}(x) \end{aligned}$$

En utilisant une notation matricielle, ceci est équivalent à : $P = J_U(x)\Delta$, où $J_U(x)$ est la jacobienne de U évaluée en x et Δ est la matrice diagonale :

$$\Delta = \begin{pmatrix} x_1 & & \\ & \ddots & \\ & & x_n \end{pmatrix}$$

Par conséquent, $P^{-1} = \Delta^{-1} J_U(x)^{-1} = \Delta^{-1} J_{U^{-1}}(x)$, où U^{-1} est la fonction réciproque de U . Nous en déduisons :

$$\forall i, j = 1 \dots n, \quad p_{ij}^{(-1)} = \frac{1}{x_i} \frac{\partial u_i^{-1}}{\partial x_j}(x)$$

En utilisant cette expression dans (5.14), nous avons, pour tout x appartenant à W et pour tout $i \in \{1, \dots, n\}$:

$$\begin{aligned} \Psi_i(x) &= x_i \prod_{j=1}^n \exp\left(-\frac{1}{x_i} u_j(x) \frac{\partial u_i^{-1}}{\partial x_j}(x)\right) \\ &= x_i \exp\left(-\sum_{j=1}^n \frac{u_j(x)}{x_i} \frac{\partial u_i^{-1}}{\partial x_j}(x)\right) \end{aligned} \quad (5.15)$$

Cette expression nous permet de calculer les dérivées partielles de Ψ (rappelons que Ψ est différentiable dans W). Par souci de clarté, nous omettons la dépendance en x :

$$\begin{aligned} \forall i \in \{1, \dots, n\}, \quad \forall k \neq i, \quad \frac{\partial \Psi_i}{\partial x_k} &= \left[\sum_{j=1}^n u_j \frac{\partial^2 u_i^{-1}}{\partial x_j \partial x_k} \right] \exp\left(-\sum_{j=1}^n \frac{u_j}{x_i} \frac{\partial u_i^{-1}}{\partial x_j}\right) \\ \forall i \in \{1, \dots, n\}, \quad \frac{\partial \Psi_i}{\partial x_i} &= \left[\sum_{j=1}^n u_j \frac{\partial^2 u_i^{-1}}{\partial x_j \partial x_k} - \frac{1}{x_i} \sum_{j=1}^n u_j \frac{\partial u_i^{-1}}{\partial x_j} \right] \exp\left(-\sum_{j=1}^n \frac{u_j}{x_i} \frac{\partial u_i^{-1}}{\partial x_j}\right) \end{aligned}$$

D'après la proposition 5.7, nous savons que les points fixes de Ψ sont des points d'équilibre de (S) . Nous en déduisons :

$$\forall k \in \{1, \dots, n\}, \quad u_k(\bar{x}) = 0$$

D'où :

$$\forall i, k \in \{1, \dots, n\}, \quad \frac{\partial \Psi_i}{\partial x_k}(\bar{x}) = 0$$

et donc, $J_\Psi(\bar{x}) = 0$. Cela entraîne que les valeurs propres de la jacobienne de Ψ évaluée en \bar{x} sont toutes nulles. La fonction Ψ est donc contractante au voisinage du point fixe \bar{x} . D'après le critère de point fixe énoncé plus haut, nous déduisons que le point fixe \bar{x} est bien un point fixe attractif de l'itération. Ceci achève la preuve de la proposition 5.8, ainsi que la correction de l'algorithme 4. ■

5.2.3 Analyse de stabilité de l'approximation par S-système

Nous reprenons les notations de la partie précédente : nous considérons un champ de vecteurs $F : \Omega \rightarrow \mathbb{R}^n$ tel que

$$\forall x \in \Omega, \quad F(x) = V^+(x) - V^-(x)$$

où $V^+, V^- : \Omega \rightarrow \Omega$ sont des champs de vecteurs positifs, supposés \mathcal{C}^2 . Nous rappelons que le système (S) est défini sur Ω par :

$$(S) \quad \dot{x} = F(x)$$

Etant donné $x^0 \in \Omega$ l'algorithme 4 calcule la suite $(x^m)_{m \in \mathbb{N}}$ telle que :

$$\forall m \in \mathbb{N}, \quad x^m = \Psi^m(x^0)$$

Nous avons montré que, à moins de tomber dans un cas dégénéré (voir alg. 4), la suite x^m converge vers un point fixe de Ψ , noté \tilde{x} , qui est un point d'équilibre du système initial : $F(\tilde{x}) = 0$.

A chaque étape $m \in \mathbb{N}$, l'algorithme calcule le point x^{m+1} à partir du point x^m . Lors de cette étape il est amené à calculer les vecteurs :

$$\begin{cases} \alpha^m &= \alpha(x^m) \\ \beta^m &= \beta(x^m) \end{cases}$$

ainsi que les matrices :

$$\begin{cases} G^m &= (g_{ij}^m)_{i,j} & \text{avec : } g_{ij}^m &= g_{ij}(x^m) \\ H^m &= (h_{ij}^m)_{i,j} & \text{avec : } h_{ij}^m &= h_{ij}(x^m) \end{cases}$$

A chaque étape, l'algorithme calcule donc le S-système $\mathcal{S}(\alpha^m, \beta^m, G^m, H^m)$. Grâce à (5.12) et (5.11), nous avons une expression explicite des paramètres :

$$\begin{cases} \forall i, j \in \{1, \dots, n\}, & g_{ij}^m &= \frac{x_j^{m-1}}{v_i^+(x^{m-1})} \frac{\partial v_i^+}{\partial x_j}(x^{m-1}) \\ \forall i, j \in \{1, \dots, n\}, & h_{ij}^m &= \frac{x_j^{m-1}}{v_i^-(x^{m-1})} \frac{\partial v_i^-}{\partial x_j}(x^{m-1}) \end{cases}$$

$$\begin{cases} \forall i \in \{1, \dots, n\}, & \alpha_i^m &= v_i^+(x^{m-1}) \prod_{j=1}^n (x_j^{m-1})^{-g_{ij}^{m-1}} \\ \forall i \in \{1, \dots, n\}, & \beta_i^m &= v_i^-(x^{m-1}) \prod_{j=1}^n (x_j^{m-1})^{-h_{ij}^{m-1}} \end{cases}$$

Puisque nous avons supposé que les champs V^+ et V^- étaient \mathcal{C}^2 et puisque x^m converge, nous en déduisons que les suites α^m, β^m, G^m et H^m convergent lorsque m tend vers $+\infty$:

$$\begin{cases} \alpha^m &\rightarrow \alpha(\tilde{x}) &\stackrel{\text{def}}{=} \tilde{\alpha} \\ \beta^m &\rightarrow \beta(\tilde{x}) &\stackrel{\text{def}}{=} \tilde{\beta} \\ G^m &\rightarrow G(\tilde{x}) &\stackrel{\text{def}}{=} \tilde{G} \\ H^m &\rightarrow H(\tilde{x}) &\stackrel{\text{def}}{=} \tilde{H} \end{cases} \quad (5.16)$$

Considérons le S-système $(\tilde{S}) = \mathcal{S}(\tilde{\alpha}, \tilde{\beta}, \tilde{G}, \tilde{H})$. Il est défini par les équations différentielles :

$$(\tilde{S}) : \quad \dot{x}_i = \tilde{\alpha}_i \prod_{j=1}^n x_j^{\tilde{g}_{ij}} - \tilde{\beta}_i \prod_{j=1}^n x_j^{\tilde{h}_{ij}}, \quad i = 1 \dots n$$

Nous voulons savoir en quel sens le système (\tilde{S}) approche le système (S) au voisinage de \tilde{x} . La proposition suivante nous permet d'éclaircir ce point :

Proposition 5.9 Soit $F = V^+ - V^-$ un champ de vecteurs défini sur Ω et supposé de classe \mathcal{C}^2 sur Ω . Soit (S) le système dynamique défini sur Ω par $\dot{x} = F(x)$. Soit \tilde{x} un point d'équilibre hyperbolique de (S) et soit (\tilde{S}) le S -système

$$\mathcal{S}(\tilde{\alpha}, \tilde{\beta}, \tilde{G}, \tilde{H}) \quad \dot{x} = \tilde{F}(x)$$

où $\tilde{\alpha}$, $\tilde{\beta}$, \tilde{G} et \tilde{H} sont définis par (5.16).

Nous avons alors les résultats suivants :

- (i) Le point d'équilibre \tilde{x} de (S) est un point d'équilibre de (\tilde{S}) .
- (ii) F et \tilde{F} sont topologiquement conjugués au voisinage de \tilde{x} , c'est-à-dire qu'il existe deux voisinages U et V de \tilde{x} et un homéomorphisme $h : U \rightarrow V$ tels que pour tout $x \in U$ tel que $\tilde{F}(x) \in U$,

$$h \circ \tilde{F}(x) = F \circ h(x)$$

Démonstration

(\tilde{S}) est le S -système $(S_{\tilde{x}})$. Le point (i) est alors évident avec la proposition 5.5.

Quant au point (ii), il est une directe conséquence du théorème de Hartman-Grobman (voir par exemple [83]). En effet, grâce à la proposition 5.5, nous avons :

$$J_F(\tilde{x}) = J_{\tilde{F}}(\tilde{x})$$

Or, le théorème de Hartman-Grobman induit que F et \tilde{F} sont topologiquement conjugués à leurs linéarisés. Par transitivité de la conjugaison topologique, nous déduisons que F et \tilde{F} sont topologiquement conjugués au voisinage de \tilde{x} . ■

Grâce à cette proposition, nous pouvons déduire de l'algorithme 4 deux résultats intéressants. Tout d'abord, il nous fournit une méthode itérative symbolique-numérique permettant de trouver les points d'équilibre positifs d'un système dynamique de dimension n donné par un système différentiel du type :

$$\dot{x} = \text{Production} - \text{Dégradation}$$

Et il nous fournit d'autre part un S -système (\tilde{S}) approximant ce système dynamique au voisinage de ce point d'équilibre. Puisque ce système est topologiquement conjugué au système initial, l'étude des trajectoires de (S) autour du point d'équilibre se ramène à celle des trajectoires de (\tilde{S}) . En particulier, la stabilité du point d'équilibre pour le système initial se déduit directement de celle du point d'équilibre pour (\tilde{S}) .

5.3 Résultats

5.3.1 Exemple en dimension 2 : bascule biologique

Nous considérons la système bidimensionnel d'équations :

$$\begin{cases} \dot{x}_1 = f(x_1, x_2) = \frac{k_1}{\theta_1 + x_2^n} - \nu_1 x_1 \\ \dot{x}_2 = g(x_1, x_2) = \frac{k_2}{\theta_2 + x_1^m} - \nu_2 x_2 \end{cases} \quad (5.17)$$

qui représente l'évolution de deux quantités positives x_1 et x_2 qui suivent une loi de dégradation linéaire et une loi de production de rétroaction : x_1 inhibe x_2 et x_2 inhibe x_1 . Ces inhibitions se font par l'intermédiaire de fonctions de Hill (voir annexe A). On consultera par exemple [12] pour une description détaillée de ces systèmes. Pour certaines valeurs des paramètres, on observe trois points d'équilibre positifs : deux stables et un instable.

Considérons le système (5.17) avec les paramètres suivants :

$$k_1 = 3, \quad k_2 = 6.75, \quad \nu_1 = 1, \quad \nu_2 = 1, \quad \theta_1 = 1, \quad \theta_2 = 3.375, \quad n = 2, \quad m = 3$$

Les équations deviennent :

$$\begin{cases} \dot{x}_1 = f(x_1, x_2) = \frac{3}{1 + x_2^2} - x_1 \\ \dot{x}_2 = g(x_1, x_2) = \frac{6.75}{3.375 + x_1^3} - x_2 \end{cases} \quad (5.18)$$

Ce système comporte trois équilibres positifs, comme le montre la figure ci-dessous.

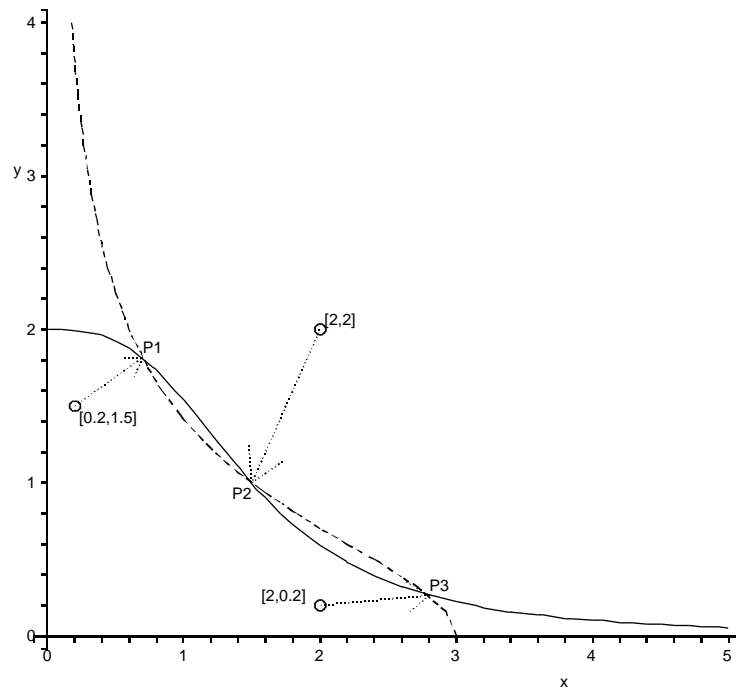


FIG. 5.2: Représentation du plan de phase du système (5.18). Les deux courbes sont les nullclines du système, c'est-à-dire les courbes d'équations $f(x_1, x_2) = 0$ et $g(x_1, x_2) = 0$. À l'intersection de ces courbes, nous trouvons les trois points d'équilibre P_1 , P_2 et P_3 du système.

Ces équilibres, notés P_1 , P_2 et P_3 valent :

$$P_1 \approx \begin{pmatrix} 0.697 \\ 1.818 \end{pmatrix} \quad P_2 = \begin{pmatrix} 1.5 \\ 1.0 \end{pmatrix} \quad P_3 \approx \begin{pmatrix} 2.802 \\ 0.266 \end{pmatrix}$$

On peut montrer que P_1 et P_3 sont asymptotiquement stables, alors que P_2 est instable (cf. [12]). En appliquant notre algorithme (programmé en Maple), nous trouvons facilement trois conditions initiales menant respectivement aux trois points d'équilibre :

- En partant de $(2, 2)$, l'algorithme converge vers P_2 en quatre itérations (avec une précision de 10^{-5}). Le S-système approximant est donné par :

$$\begin{cases} \dot{x}_1 &= 1.500 x_2^{-1} - x_1 \\ \dot{x}_2 &= 1.837 x_1^{-1.5} - x_2 \end{cases}$$

- En partant de $(0.2, 1.5)$, l'algorithme converge vers P_1 en quatre itérations (avec une précision de 10^{-5}). Le S-système approximant est donné par :

$$\begin{cases} \dot{x}_1 &= 1.745 x_2^{-1.535} - x_1 \\ \dot{x}_2 &= 1.647 x_1^{-0.274} - x_2 \end{cases}$$

- En partant de $(2, 0.2)$, l'algorithme converge vers P_3 en quatre itérations (avec une précision de 10^{-5}). Le S-système approximant est donné par :

$$\begin{cases} \dot{x}_1 &= 2.352 x_2^{-0.132} - x_1 \\ \dot{x}_2 &= 3.879 x_1^{-2.6} - x_2 \end{cases}$$

La figure 5.2 reprend ces trois expériences.

Analysons plus précisément le S-système approximant notre système autour du point P_2 :

$$\begin{cases} \dot{x}_1 &= 1.500 x_2^{-1} - x_1 \\ \dot{x}_2 &= 1.837 x_1^{-1.5} - x_2 \end{cases} \quad (5.19)$$

En traçant les trajectoires de ce système avec des points initiaux dans le voisinage de P_2 , on observe qu'elles sont très proches des trajectoires du système initial, tendant à confirmer la qualité de l'approximation par S-système (voir figure 5.3).

Sur cette figure, on voit bien le caractère local de la S-approximation. Si les trajectoires du système (5.19) approchent relativement bien celles de (5.18) au voisinage du point d'équilibre P_2 , elles s'en éloignent rapidement lorsqu'elles s'échappent de ce voisinage. En particulier, les trajectoires du système initial se dirigent vers les points stables P_1 et P_3 tandis que celles du système approché explosent (il ne peut en être autrement, puisque le S-système (5.19) a un unique point d'équilibre, P_2 et qu'il est instable).

5.3.2 Un exemple en dimension 3 : comparaison avec la linéarisation

L'un des avantages de l'approximation par S-système est qu'elle n'est pas trop limitée par la dimension n du système (en effet, dans l'algorithme 4, la majeure partie du temps de calcul d'une itération est occupée par la résolution d'un système linéaire de dimension n). Cette méthodologie peut donc s'appliquer à des systèmes de dimension supérieure à 2. Prenons par exemple le système dynamique de dimension 3 suivant :

$$\begin{cases} \dot{x}_1 &= \frac{x_1}{2+x_2} - \frac{x_1^2 x_2^4}{(3+x_1)(4+x_2^3)} \\ \dot{x}_2 &= \frac{5x_1}{3+x_3} - \frac{2x_2^3 x_3}{(1+x_1)(2+x_2)} \\ \dot{x}_3 &= \frac{x_3}{2+x_2} - \frac{4x_2^2 x_3^3}{(2+x_3)(1+x_2^4)} \end{cases} \quad (5.20)$$

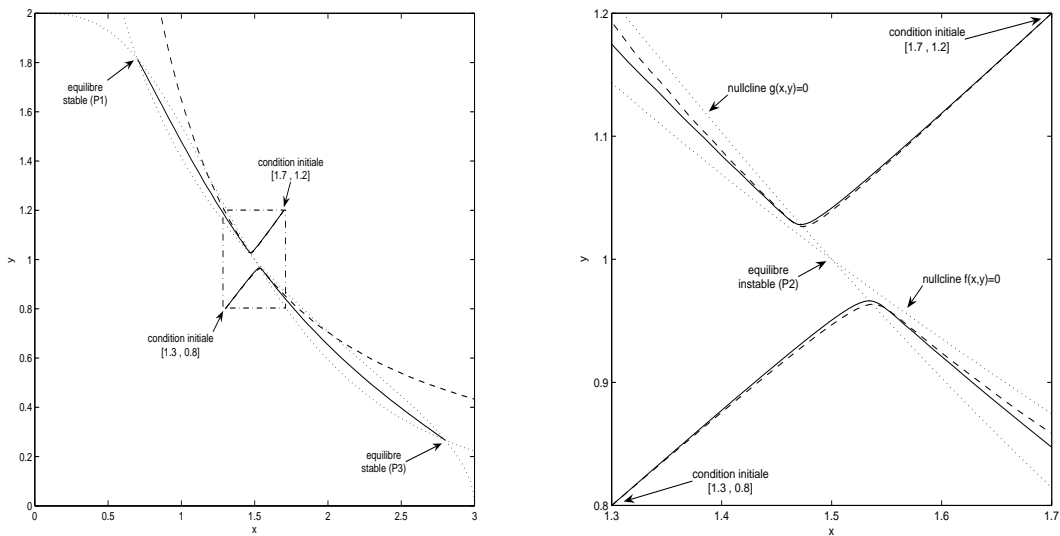


FIG. 5.3: Exemples de trajectoires autour du point d'équilibre instable P_2 . Les trajectoires en trait plein sont celles du système initial (5.18), celles en tirets sont celles du S -système approximant (5.19). Ces trajectoires correspondent à deux conditions initiales : $x^0 = (1.3, 0.8)$ et $x^0 = (1.7, 1.2)$. Sur le graphique de gauche, on voit que pour une même condition initiale les trajectoires de (5.18) et de (5.19) semblent proches aux alentours du point P_2 puis diffèrent en s'en éloignant. Sur le graphe de droite, un zoom sur la partie centrale permet de mieux voir le rapprochement des trajectoires au voisinage de P_2 . Remarque : les nullclines $f(x_1, x_2) = 0$ et $g(x_1, x_2) = 0$ sont représentés en pointillés.

En lançant notre algorithme avec $x^0 = (2, 2, 2)$, nous obtenons, avec une précision fixée à 10^{-5} le point d'équilibre positif :

$$\tilde{x} \approx (0.9709, 1.8054, 0.8116) \quad (5.21)$$

Ce point est obtenu au bout de seulement 3 itérations. On trouve également le S-système approximant :

$$\begin{cases} \dot{x}_1 &= \tilde{\alpha}_1 \tilde{x}_1^{\tilde{g}_{11}} \tilde{x}_2^{\tilde{g}_{12}} \tilde{x}_3^{\tilde{g}_{13}} - \tilde{\beta}_1 \tilde{x}_1^{\tilde{h}_{11}} \tilde{x}_2^{\tilde{h}_{12}} \tilde{x}_3^{\tilde{h}_{13}} \\ \dot{x}_2 &= \tilde{\alpha}_2 \tilde{x}_1^{\tilde{g}_{21}} \tilde{x}_2^{\tilde{g}_{22}} \tilde{x}_3^{\tilde{g}_{23}} - \tilde{\beta}_2 \tilde{x}_1^{\tilde{h}_{21}} \tilde{x}_2^{\tilde{h}_{22}} \tilde{x}_3^{\tilde{h}_{23}} \\ \dot{x}_3 &= \tilde{\alpha}_3 \tilde{x}_1^{\tilde{g}_{31}} \tilde{x}_2^{\tilde{g}_{32}} \tilde{x}_3^{\tilde{g}_{33}} - \tilde{\beta}_3 \tilde{x}_1^{\tilde{h}_{31}} \tilde{x}_2^{\tilde{h}_{32}} \tilde{x}_3^{\tilde{h}_{33}} \end{cases} \quad (5.22)$$

avec :

$$\tilde{\alpha} \approx \begin{pmatrix} 0.3478 \\ 1.2548 \\ 0.3478 \end{pmatrix}, \quad \tilde{\beta} \approx \begin{pmatrix} 0.0727 \\ 0.3478 \\ 0.9990 \end{pmatrix} \quad \text{et} \quad \tilde{G} - \tilde{H} \approx \begin{pmatrix} -0.7555 & -2.6885 & 0 \\ 1.4926 & -2.5256 & -1.2129 \\ 0 & 1.1814 & -1.7113 \end{pmatrix}$$

Montrons que la matrice $\tilde{P} = \tilde{G} - \tilde{H}$ est signée semi-stable.

Pour cela, il nous suffit de montrer qu'elle vérifie les trois conditions du théorème 5.3 :

(i) On a : $p_{11} < 0$, $p_{22} < 0$ et $p_{33} < 0$.

(ii) On vérifie facilement que pour tout $i, j \in \{1, 2, 3\}$, on a : $p_{ij} p_{ji} \leq 0$.

(iii) Enfin, étant donné que $p_{13} = p_{31} = 0$, on vérifie également que la matrice \tilde{P} ne contient pas de cycle de longueur 3.

La matrice $\tilde{G} - \tilde{H}$ est donc bien signée semi-stable. Comme l'indique la proposition 5.4, ceci entraîne que le point \tilde{x} est un équilibre asymptotiquement stable du système (5.22). Grâce à la proposition 5.9, nous en déduisons que \tilde{x} est un point d'équilibre asymptotiquement stable du système (5.20).

Comme pour l'exemple précédent, nous montrons, pour une condition initiale donnée, l'allure d'une trajectoire du système (5.20) et du système (5.22) autour de ce point d'équilibre. La figure 5.4 présente ces trajectoires dans \mathbb{R}^3 et la figure 5.5 donne leurs projections dans les plans (x_1, x_2) , (x_1, x_3) et (x_2, x_3) .

Nous voulons à présent comparer la qualité de l'approximation par S-système avec la linéarisation classique. Pour cela, nous calculons le système linéarisé du système (5.20) en \tilde{x} . Nous rappelons que le linéarisé est donné par :

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = J_F(\tilde{x}) \begin{pmatrix} x_1 - \tilde{x}_1 \\ x_2 - \tilde{x}_2 \\ x_3 - \tilde{x}_3 \end{pmatrix} \quad (5.23)$$

Afin de visualiser l'éloignement respectif des trajectoires du S-système et du linéarisé par rapport au système initial, on consultera la figure 5.5. On remarque sur cet exemple que l'approximation par S-système est plus proche du système initial que la linéarisation.

5.3.3 Discussion sur la vitesse de convergence de l'algorithme 4

Dans les exemples que nous avons décrits, le nombre d'itérations nécessaires pour parvenir à s'approcher suffisamment près du point d'équilibre est faible (de l'ordre de 4 ou 5). Cette vitesse

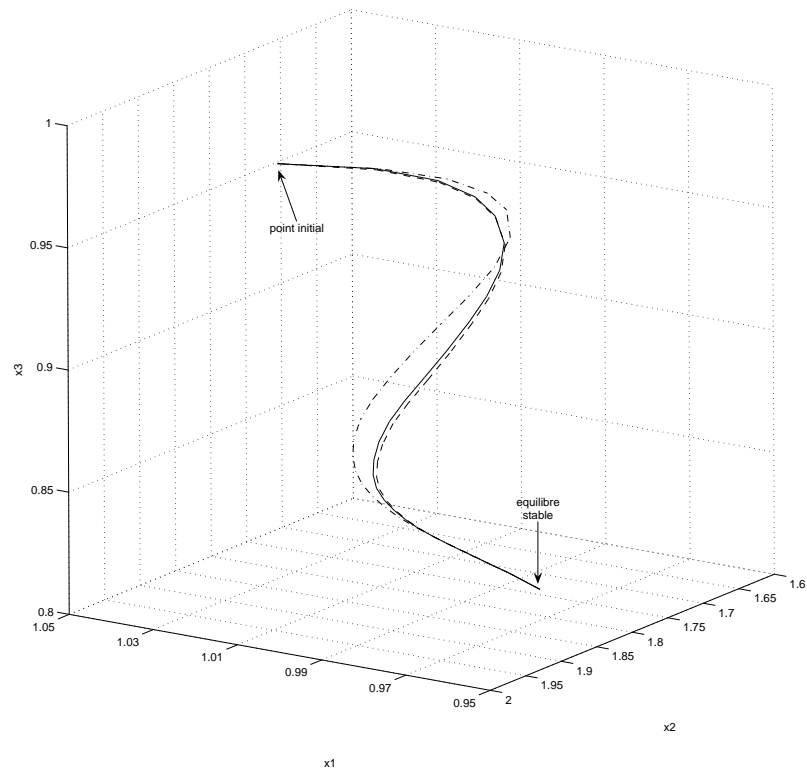


FIG. 5.4: Trajectoires du système (5.20) (trait plein), du S -système approximant (5.22) (tirets) et du système linéarisé (5.23) (tirets-points) avec pour condition initiale le point $(1, 2, 1)$. Ces trois trajectoires convergent vers le point d'équilibre stable \tilde{x} . On remarque que la trajectoire du S -système est plus proche de la trajectoire réelle que celle du linéarisé (ceci est confirmé par les projections sur la figure 5.5).

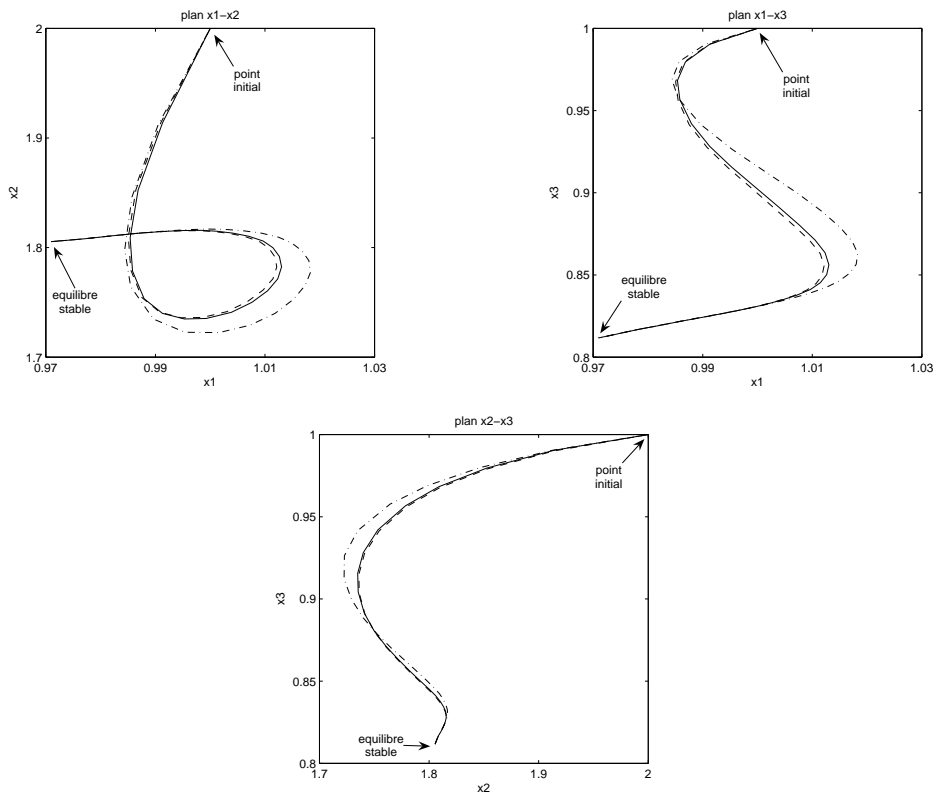


FIG. 5.5: Comparaison de l'approximation par S -système par rapport à la linéarisation. Les trois trajectoires représentent les projections sur les plans (x_1, x_2) , (x_1, x_3) et (x_2, x_3) des trajectoires issues du même point initial des systèmes (5.20) (système initial, trait plein), (5.22) (S -système, tirets) et (5.23) (système linéarisé, tirets-points).

de convergence est en fait une propriété générale de l'algorithme. En effet, nous rappelons que l'algorithme consiste à itérer une application $\Psi : \Omega \rightarrow \Omega$ à partir d'un point initial quelconque $x^0 \in \Omega$. Il calcule donc la suite :

$$\forall m \in \mathbb{N}, \quad x^m = \Psi^m(x^0)$$

Or, la vitesse de convergence d'une telle suite est reliée à la constante de Lipschitz de Ψ . Supposons que Ψ soit K -contractante, avec $0 \leq K < 1$. Alors :

$$\forall x, y \in \Omega, \quad \|\Psi(x) - \Psi(y)\| \leq K\|x - y\|$$

Nous pouvons alors écrire :

$$\begin{aligned} \forall m \in \mathbb{N}, \quad \|\tilde{x} - x^m\| &= \|\Psi(\tilde{x}) - \Psi(x^{m-1})\| \\ &\leq K\|\tilde{x} - x^{m-1}\| \\ &\leq K^2\|\tilde{x} - x^{m-2}\| \\ &\vdots \\ &\leq K^m\|\tilde{x} - x^0\| \end{aligned}$$

La suite (x^m) converge donc vers le point fixe \tilde{x} aussi rapidement que la suite géométrique de raison K converge vers 0. La vitesse de convergence en nombre d'itérations est donc d'autant plus rapide que le réel K est proche de zéro.

Dans notre cas, nous avons démontré que la jacobienne de Ψ s'annulait en \tilde{x} . Ceci implique que pour tout $K \in]0, 1[$, il existe un voisinage W_K de \tilde{x} tel que Ψ soit K -contractante sur W_K . En conséquence, si le point initial de l'itération ne se situe pas trop loin de \tilde{x} alors peu d'itérations suffiront pour converger vers \tilde{x} avec une bonne précision.

Pour des systèmes de dimension pas trop élevées (de l'ordre de la dizaine), le temps de calcul de l'algorithme est donc faible. Nous rappelons que l'essentiel du temps de calcul d'une itération se situe dans l'inversion d'un système linéaire de dimension n .

Chapitre 6

Etude d'un réseau métabolique chez *Arabidopsis thaliana*

Nous proposons dans ce chapitre un travail qui est actuellement encore en cours, en collaboration avec G. Curien (CEA, Grenoble) [14, 15, 56] qui concerne la synthèse d'acides aminés chez la plante *Arabidopsis thaliana*. Ce travail nous a incité à étudier de façon plus précise les équations différentielles régissant l'évolution de réseaux métaboliques. Devant la complexité de ces systèmes, il nous est apparu qu'il était capital d'avoir à disposition des classes de systèmes dynamiques continus qui allient une certaine généralité et une maniabilité des équations. Ce travail est donc à l'origine de notre étude des S-systèmes.

Les voies de synthèse des acides aminés essentiels chez les plantes et les bactéries ont été étudiées en détails depuis plus d'une cinquantaine d'années. Cette étude s'est effectuée à la fois pour des raisons pratiques (certains de ces acides aminés¹ ne sont pas synthétisés par les mammifères et doivent être ajoutés à leur alimentation ; et d'autre part certains enzymes sont la cible d'herbicides très puissants) et également pour des raisons plus fondamentales liées à la nature *branchée* de ces systèmes. L'organisation de ces voies de synthèse pose des questions intéressantes, liées à l'existence de nombreux points de branchement et à la complexité des contrôles allostériques. Nous faisons le choix de ne pas considérer les contrôles génétiques dans un premier temps, afin de nous focaliser sur l'étude du réseau métabolique.

Ces voies de synthèse chez les plantes et les bactéries font certainement partie des voies métaboliques (hors métabolisme énergétique) les mieux connues, au niveau de leurs composants (enzymes) et de leur localisation sub-cellulaire (le chloroplaste de la plante dans notre cas).

Après une description plus précise du système différentiel, nous verrons que la dynamique de ce système est régie par quatorze équations différentielles comportant des termes rationnels ainsi que neuf paramètres formels. L'étude directe de ce système est extrêmement difficile. Nous proposons alors d'appliquer l'algorithme présenté dans le chapitre précédent afin de chercher le point d'équilibre positif. Autour de ce point d'équilibre (qui correspond à un régime stationnaire de la plante), nous obtenons également une approximation du système différentiel sous la forme d'un S-système de dimension 14.

¹notamment la thréonine ainsi que les acides aminés à chaîne ramifiée comme la leucine, l'isoleucine et la valine

6.1 Présentation du modèle

6.1.1 La plante *Arabidopsis thaliana*

Le système que nous nous proposons d'étudier appartient à la plante *Arabidopsis thaliana* (voir figure 6.1). Plusieurs propriétés en font une plante "modèle" en biologie végétale. Elle a récemment fait l'objet de nombreuses études, notamment sur son métabolisme. La faible taille de son génome (environ 120 millions de paires de bases) en fait également un outil d'étude intéressant au point de vue génétique. Nous disposons notamment aujourd'hui d'une carte détaillée de ses cinq chromosomes.

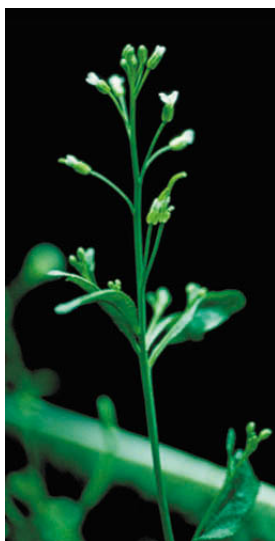


FIG. 6.1: La plante *Arabidopsis thaliana* (ou *Arabette des Dames*).

6.1.2 Réseau métabolique responsable de la production d'acides aminés

Le système qui nous a été proposé fait partie du réseau métabolique responsable de la synthèse de plusieurs acides aminés chez *Arabidopsis*. Bien que l'on dispose aujourd'hui de beaucoup de données qualitatives et quantitatives sur ce réseau, la somme des connaissances n'est encore que partielle. Néanmoins, ces données commencent à être suffisantes pour avoir une idée de l'enchaînement des différents processus qui conduisent à la synthèse de ces acides aminés. Comme le montre la figure 6.2, cet enchaînement conduit à l'étude d'un réseau d'interactions complexe. Ces interactions sont des interactions chimiques directes comme la transformation d'une substance par le biais d'un ou plusieurs enzymes, ou indirectes, par le biais des interactions allostériques à cause desquelles la concentration d'une substance intermédiaire ou finale peut influencer sur l'efficacité d'enzymes dont l'action se situe en amont. Le travail dans lequel nous nous inscrivons ici consiste à étudier le système tel qu'il est connu à l'heure actuelle et à proposer certains éléments de compréhension qui permettraient de mieux rendre compte de l'organisation globale de ces processus à l'échelle d'une cellule ou d'un tissu.

Cette tâche dépasse bien entendu le cadre de cette thèse. Le traitement mathématique d'un réseau tel que celui présenté figure 6.2 est très complexe. Comme nous allons le voir par la suite, les différents modèles d'équations différentielles utilisés pour représenter les interactions entre les différents composants génèrent des systèmes dynamiques de grandes dimensions, comportant des termes fractionnels complexes ainsi que de nombreux paramètres formels empêchant une

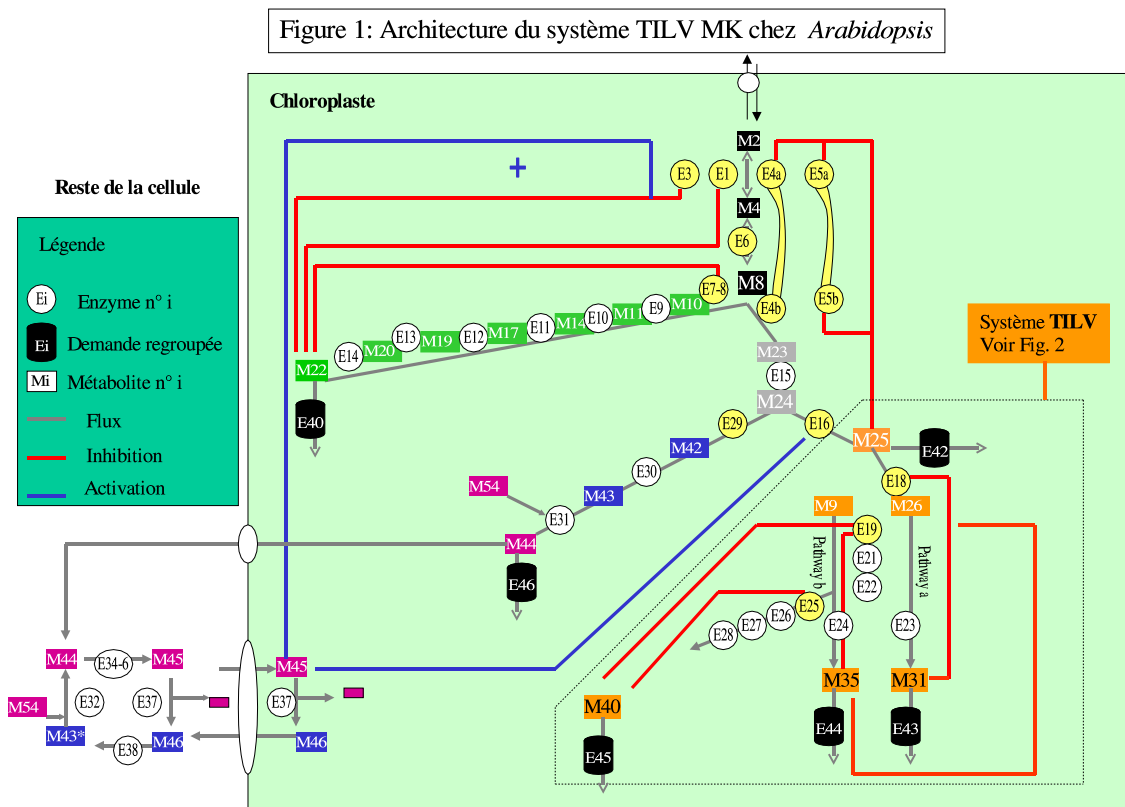


FIG. 6.2: Ce schéma présente l'organisation des différentes réactions biochimiques, telle qu'elle est connue aujourd'hui, responsable de la production d'acides aminés dans un chloroplaste de la plante *Arabidopsis thaliana*.

résolution analytique directe. La méthode la plus utilisée pour rendre compte de la dynamique de ces systèmes reste, à notre connaissance, la simulation numérique.

Comme nous l'avons déjà évoqué, une seconde approche devant la complexité de ces systèmes consiste à proposer des modèles plus ou moins éloignés des équations de la cinétique enzymatique. Dans cette optique, les S-systèmes nous semblent intéressants pour toutes les raisons évoquées dans le chapitre précédent. La modélisation de systèmes comportant des lois de puissance a d'ailleurs déjà été utilisée dans le cas de réseaux biochimiques, on consultera à cet effet l'ouvrage [80].

6.1.3 Le système TILV

La première étape vers l'étude du réseau précédent a consisté en l'isolement d'un "petit" sous-réseau comportant 14 variables. Ce réseau a été dénommé TILV par les initiales des quatre acides aminés dont il contrôle la production : la thréonine, l'isoleucine, la leucine et la valine. Le schéma de ce réseau est présenté sur la figure 6.3 ci-dessous.

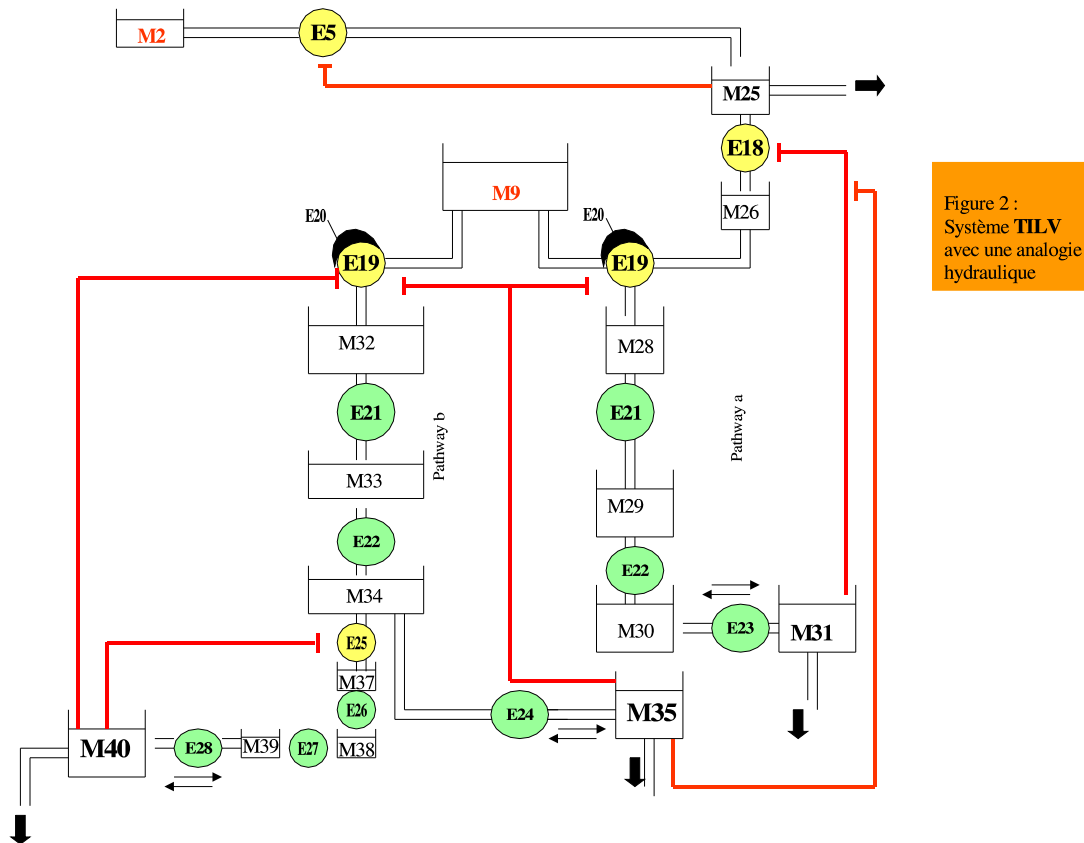


FIG. 6.3: Schéma du système TILV, responsable de la synthèse de thréonine, d'isoleucine, de leucine et de valine. Les liens rouges représentent les interactions allostériques (il s'agit d'inhibitions dans ce cas) connues entre les différents constituants. Les variables E_i désignent les enzymes et les variables M_i représentent les métabolites. Les métabolites M_{25} , M_{31} , M_{35} et M_{40} représentent respectivement la thréonine, l'isoleucine, la valine et la leucine (produits finaux).

6.1.4 Les équations de la dynamique

Afin d'expliciter les équations différentielles associées au système TILV, nous reprenons ici les principales hypothèses biologiques qui ont été faites par G. Curien à partir d'expériences réalisées *in vitro* ainsi que de références bibliographiques.

- Nous ne considérons que les phénomènes de contrôle opérant sur une échelle de temps courte (de l'ordre de la dizaine de minutes). Les concentrations en protéines sont donc considérées comme des paramètres constants. Cette hypothèse néglige les effets de dilution liés aux divisions cellulaires, ce qui est acceptable dans ce système.
- Nous considérons un milieu homogène (non compartimenté) dans lequel la diffusion se fait librement.
- Nous faisons l'hypothèse que les enzymes *intermédiaires* opèrent hors de la zone de saturation et que leur dynamique est donc *linéaire*.
- En ce qui concerne les enzymes ayant deux substrats, l'un étant spécifique à la voie et l'autre étant un métabolite général (comme l'ATP par exemple), on considèrera que le second substrat est maintenu à une valeur constante.
- Les différents termes de demande en les produits terminaux (M_{25} , M_{31} , M_{35} et M_{40}) ont été regroupés en un seul terme de type Michaelis-Menten.
- Les concentrations en enzymes, supposées constantes (on ne considère pas ici le phénomène de régulation génétique) sont estimées à une valeur de l'ordre de $1 \mu M$.

A partir de ces hypothèses générales ainsi que certaines hypothèses plus spécifiques, le système d'équations qui nous a été fourni a la forme suivante :

$$\dot{X} = F(X, \Theta) \quad (6.1)$$

où X est le vecteur des variables et Θ le vecteur des paramètres :

$$X(t) = \begin{pmatrix} M_{25}(t) \\ M_{26}(t) \\ M_{28}(t) \\ M_{29}(t) \\ M_{30}(t) \\ M_{31}(t) \\ M_{32}(t) \\ M_{33}(t) \\ M_{34}(t) \\ M_{35}(t) \\ M_{37}(t) \\ M_{38}(t) \\ M_{39}(t) \\ M_{40}(t) \end{pmatrix} \in (\mathbb{R}_+^*)^{14}, \quad \Theta = \begin{pmatrix} M_2 \\ a \\ b \\ c \\ d \\ K_1 \\ K_2 \\ K_3 \\ K_4 \end{pmatrix} \in (\mathbb{R}_+^*)^9$$

et :

$$F(X, \Theta) = \begin{pmatrix} v_3(M_{25}; \mathbf{M}_2) - v_{18}(M_{25}, M_{31}, M_{35}) - \mathbf{S}(M_{25}; \mathbf{d}, \mathbf{K}_4) \\ v_{18}(M_{25}, M_{31}, M_{35}) - v_{19a}(M_{26}, M_{35}, M_{40}) \\ v_{19a}(M_{26}, M_{35}, M_{40}) - v_{21a}(M_{28}) \\ v_{21a}(M_{28}) - v_{22a}(M_{29}) \\ v_{22a}(M_{29}) - v_{23}(M_{30}, M_{31}) \\ v_{23}(M_{30}, M_{31}) - \mathbf{S}(M_{31}; \mathbf{a}, \mathbf{K}_1) \\ v_{19b}(M_{26}, M_{35}, M_{40}) - v_{21b}(M_{32}) \\ v_{21b}(M_{32}) - v_{22b}(M_{33}) \\ v_{22b}(M_{33}) - v_{24}(M_{34}, M_{35}) - v_{25}(M_{34}, M_{40}) \\ v_{24}(M_{34}, M_{35}) - \mathbf{S}(M_{35}; \mathbf{b}, \mathbf{K}_2) \\ v_{25}(M_{34}, M_{40}) - v_{26}(M_{37}) \\ v_{26}(M_{37}) - v_{27}(M_{38}) \\ v_{27}(M_{38}) - v_{28}(M_{39}, M_{40}) \\ v_{28}(M_{39}, M_{40}) - \mathbf{S}(M_{40}; \mathbf{c}, \mathbf{K}_3) \end{pmatrix}$$

Les différents termes v_i sont des vitesses de réaction, définies comme suit :

$$\begin{aligned} v_3(M_{25}; M_2) &= \frac{1}{100} \frac{M_2}{1 + \left(\frac{M_{25}}{50}\right)^3} \\ v_{21a}(M_{28}) &= \frac{17}{20} M_{28} \\ v_{22a}(M_{29}) &= \frac{17}{20} M_{29} \\ v_{21b}(M_{32}) &= \frac{7}{50} M_{32} \\ v_{22b}(M_{33}) &= \frac{7}{50} M_{33} \\ v_{23}(M_{30}, M_{31}) &= \frac{1}{10} M_{30} - \frac{1}{100} M_{31} \\ v_{24}(M_{34}, M_{35}) &= \frac{1}{100} M_{34} - \frac{1}{1000} M_{35} \\ v_{25}(M_{34}, M_{40}) &= \frac{37}{100} \frac{M_2}{1 + \left(\frac{M_{40}}{10}\right)^3} \\ v_{26}(M_{37}) &= M_{37} \\ v_{27}(M_{38}) &= \frac{3}{10} M_{38} \\ v_{28}(M_{39}, M_{40}) &= \frac{1}{10} M_{39} - \frac{1}{100} M_{40} \end{aligned}$$

Les vitesses v_{18} et v_{19} correspondent à des *points de branchement*. Leurs expressions sont données par :

$$v_{18}(M_{25}, M_{31}, M_{35}) = \frac{9}{625} \frac{M_{25}}{1 + \left(\frac{M_{31}}{28 + \frac{73M_{35}}{462 + M_{35}}} \right)}$$

$$v_{19a}(M_{26}, M_{35}, M_{40}) = \frac{96M_{26}}{25(35 + M_{26})} \frac{1}{1 + \frac{M_{40}}{2 + \frac{329}{1 + \frac{5}{3}M_{35}}}}$$

$$v_{19a}(M_{26}, M_{35}, M_{40}) = \frac{96}{25(1 + \frac{M_{26}}{35})} \frac{1}{1 + \frac{M_{40}}{2 + \frac{329}{1 + \frac{5}{3}M_{35}}}}$$

Enfin, les fonctions de sortie sont données par :

$$S(M_i; \alpha_i, K_i) = \frac{\alpha_i M_i}{K_i + M_i} \quad i \in \{31, 35, 40, 25\}$$

6.2 Etude du système dynamique TILV

6.2.1 Approche formelle

Afin de calculer les points d'équilibre du système (6.1), nous avons tout d'abord tenté une approche directe, c'est-à-dire la résolution du système algébrique :

$$F(X, \Theta) = 0$$

Cette résolution est très complexe, étant donné le degré des polynômes et le nombre de paramètres formels. Nous avons ainsi pu isoler un intervalle relativement restreint dans lequel se trouvent les points d'équilibre possibles.

Toutefois, une telle étude présente peu d'intérêt. Certes, cette méthode permet de manipuler les paramètres de manière symbolique, elle est néanmoins inextricablement liée à la forme des équations (ainsi qu'aux valeurs numériques des différents coefficients) et est donc peu évolutive. Étant donné que l'établissement des équations du modèle appelle de nombreuses hypothèses simplificatrices, la forme des équations va être inévitablement amenée à changer, ce qui entraînera l'obsolescence des calculs.

L'approche formelle pour un tel type de système est donc discutable. Nous avons décidé par la suite de l'abandonner au profit de l'approche symbolique numérique présentée dans le chapitre précédent.

6.2.2 Approche par S-systèmes

L'avantage de l'algorithme présenté dans le chapitre précédent est qu'il fournit, en plus du fait de converger rapidement vers un point d'équilibre positif du système, un système approximant le système initial autour du point d'équilibre. Son inconvénient majeur réside dans le fait qu'il nécessite une évaluation numérique des paramètres.

Voici un exemple d'application de l'algorithme dans lequel nous avons fait les hypothèses suivantes :

$$M_2 = 1000, \quad a = b = c = d = 0.1, \quad K_1 = K_2 = K_3 = K_4 = 100$$

Sous cette hypothèse, l'algorithme trouve au bout de 7 itérations le point d'équilibre :

$$\tilde{X} \simeq \begin{pmatrix} 197.4986157 \\ 32.72232658 \\ 0.1097471946 \\ 0.1097471946 \\ 139.8556196 \\ 1389.227683 \\ 0.7127023426 \\ 0.7127023426 \\ 22.59708388 \\ 163.8685031 \\ 0.03767599239 \\ 0.1255866413 \\ 6.421941181 \\ 60.45181249 \end{pmatrix}$$

et le S-système approximant :

$$(\tilde{S}) : \dot{X}_i = \tilde{\alpha}_i \prod_{j=1}^n X_j^{\tilde{g}_{ij}} - \tilde{\beta}_i \prod_{j=1}^n X_j^{\tilde{h}_{ij}}$$

avec pour matrice $\tilde{G} - \tilde{H}$:

$$\begin{pmatrix} -3.6761 & 0 & 0 & 0 & 0 & 0.5651 & 0 & 0 & 0 & -0.1692 & 0 & 0 & 0 & 0 \\ 1 & -0.5168 & 0 & 0 & 0 & -0.9672 & 0 & 0 & 0 & 0.6445 & 0 & 0 & 0 & 0.9497 \\ 0 & 0.5168 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -0.3549 & 0 & 0 & 0 & -0.0949 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0067 & -1 & 0.9933 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -0.9938 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.4832 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -0.3549 & 0 & 0 & 0 & -0.9497 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.3785 & -1 & 0.6215 & 0 & 0 & 0 & 0.4268 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -0.8293 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & -2.9865 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0587 & -1 & 0.9413 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -0.9779 \end{pmatrix}$$

Nous avons d'autre part calculé la jacobienne du S-système (\tilde{S}) . Elle admet comme valeurs propres :

$$\begin{aligned}
\lambda_1 &\simeq -0.8501138204 \\
\lambda_2 &\simeq -0.8498861311 \\
\lambda_3 &\simeq -0.9999101592 \\
\lambda_4 &\simeq -0.3013904751 \\
\lambda_5 &\simeq -0.1437649189 \\
\lambda_6 &\simeq -0.1355295674 \\
\lambda_7 &\simeq -0.1077751391 \\
\lambda_8 &\simeq -0.1100010324 \\
\lambda_9 &\simeq -0.0124150866 \\
\lambda_{10} &\simeq -0.0029428511 \\
\lambda_{11} &\simeq -0.0000469401 \\
\lambda_{12} &\simeq -0.0010131177 + 0.0005596237 i \\
\lambda_{13} &\simeq -0.0010131177 - 0.0005596237 i \\
\lambda_{14} &\simeq -0.0018468428
\end{aligned}$$

Ces valeurs propres sont toutes de parties réelles négatives, ce qui signifie que le point d'équilibre est stable pour (\tilde{S}) et donc que c'est un point d'équilibre stable du système (6.1). Nous avons donc bien isolé le régime stationnaire observé dans les simulations numériques qui correspond au point de fonctionnement "normal" de la plante.

Notons ici que la méthode de la S-approximation est purement symbolique, et donc qu'elle permet de donner une approximation locale en n'importe quel point de l'espace des phases. Il serait donc tout-à-fait envisageable de proposer une approximation sous forme de S-système incluant les paramètres de manière formelle. Pour cela, il faut avoir une estimation du point autour duquel nous voulons faire cette approximation. Ceci nécessite alors une bonne connaissance biologique du système lui-même ainsi que de son environnement. Une telle étude est en cours.

Conclusion et Perspectives

L'objectif de ce travail était de proposer une étude du phénomène de régulation génétique, tel qu'il est connu à l'heure actuelle. Cette étude étant principalement mathématique, nous nous sommes efforcés de rendre compte de nos différentes lectures concernant les réseaux de régulation génétique ainsi que de leur modélisation. Nous reprenons ici les principaux points abordés dans ce mémoire.

Tout d'abord, nous avons proposé une description des réseaux génétiques, en nous basant notamment sur l'exemple historique de l'opéron lactose. Cette étude s'est conclue par la construction d'un modèle de l'opéron lactose utilisant la théorie des systèmes hybrides. Une étude mathématique de ce modèle ainsi que son implémentation et sa simulation sur ordinateur nous ont permis de vérifier que les systèmes hybrides constituent un outil mathématique et algorithmique puissant, particulièrement bien adapté à la modélisation des réseaux génétiques.

Nous avons par la suite étudié deux voies distinctes de modélisations fondées sur une approche "systèmes dynamiques". Tout d'abord, nous avons proposé un cadre discret, le modèle des graphes d'automates, unifiant les deux principaux modèles discrets de régulation génétique qui sont les modèles de Kauffman (synchrones) et de Thomas (asynchrone). L'étude de ces systèmes dynamiques discrets nous a permis de nous confronter au problème de l'identification de réseaux booléens. Nous avons à ce titre proposé une preuve de l'algorithme d'identification ainsi qu'une série de résultats (théoriques et pratiques) permettant de quantifier l'efficacité de cet algorithme, en reliant notamment le taux de réussite de l'identification avec la taille des données en entrée.

Nous nous sommes ensuite intéressés à une classe de systèmes dynamiques continus : les S-systèmes. Ceci nous a notamment permis de proposer un algorithme itératif de recherche de points d'équilibre positifs d'un système différentiel. Cette algorithme donne également une approximation canonique non linéaire du système initial autour de ce point d'équilibre, offrant ainsi une alternative à la technique classique de la linéarisation. Cette étude est suivie d'une application à un système biologique réel qui est le système TILV chez *Arabidopsis thaliana*.

Etant donnée la richesse du domaine, les perspectives de ce travail sont évidemment nombreuses.

Tout d'abord, les deux applications biologiques (c'est-à-dire les chapitres 2 et 6) sont toujours en cours d'étude. Comme nous l'avons évoqué dans l'introduction de ce mémoire, la modélisation en biologie est une discipline délicate, qui demande des connaissances dans des domaines scientifiques variés. La construction du modèle d'un système réel n'est bien entendu jamais idéale. Elle nécessite d'être critiquée, améliorée, parfois simplifiée ou au contraire complexifiée par de nouvelles connaissances biologiques. En ce qui concerne l'opéron lactose, il semble que l'approche par système hybride soit pertinente. Un travail reste à faire sur la si-

mulation et l'obtention de trajectoires continues. L'étape prochaine consisterait à automatiser une telle approche, de manière à la généraliser à d'autres réseaux de régulation génétiques.

L'étude du système TILV, quant à elle, bien que prometteuse, est encore loin d'être achevée. Il s'agit maintenant de confronter la S-approximation à une étude qualitative, afin d'en tirer certaines propriétés utiles pour la biologie.

En ce qui concerne la seconde partie, nous proposons deux travaux qui suivent logiquement l'étude présentée dans ce mémoire. En ce qui concerne l'inférence, il nous semble que l'application de l'algorithme à des données réelles pourrait être intéressante. Il faut néanmoins être prudent. En effet, même si la technologie des biopuces à ADN est aujourd'hui une avancée formidable, nous sommes encore limités dans la taille des données temporelles (c'est-à-dire du nombre d'expériences successives). Etant donné la large dimension des génomes, il n'est pas envisageable d'appliquer directement l'algorithme à de tels jeux de données. Ces données nécessitent un traitement statistique préalable afin de classifier les différents gènes, et d'isoler de petits groupes de gènes semblant participer à une même tâche. Un autre problème réside dans le fait que les données biopuces sont constituées de séries temporelles, c'est-à-dire de portions de trajectoires dans le graphe de transition. Nous avons supposé quant à nous que les données étaient des transitions aléatoires de ce graphe.

Le second travail que nous voulons citer ici concerne l'étude des dynamiques de réseaux booléens. Plus particulièrement, une classe particulière de fonctions booléennes est fréquemment abordée dans la littérature, il s'agit des fonctions forçantes (ou canalisantes). On a pu vérifier sur des exemples (voir par exemple [42]) que ces fonctions ont des propriétés intéressantes sur la dynamique du réseau dans le cas synchrone. Ce champ de recherche reste à notre connaissance relativement ouvert. Ce n'est que récemment qu'a été publié un article proposant le calcul du nombre de fonctions forçantes ainsi que de leur probabilité d'apparition (voir [38]).

Nous terminons par les perspectives du travail sur les S-systèmes. Comme nous l'avons vu, la méthode de S-approximation est purement symbolique, elle peut donc être calculée en tout point de l'espace des phases, y compris en un point dépendant de paramètres formels. Malheureusement, l'algorithme de recherche d'équilibres positifs que nous proposons dans le chapitre 5 est, quant à lui, numérique (pour assurer la convergence, nous devons donner une valeur numérique au point initial x^0). Dans l'application au système TILV, cela nous empêche de traiter les paramètres de manière formelle et nous oblige donc à leur donner différentes valeurs pour chaque expérience. Etant donné que la S-approximation est une méthode formelle, une idée intéressante serait de proposer une S-approximation par morceaux du système (voir [67]), incluant ainsi les paramètres de manière symbolique. Cette idée est confortée par les exemples que nous avons montrés dans le chapitre 5, car ils indiquent que la S-approximation semble tout-à-fait correcte, même meilleure que la linéarisation, au moins dans un certain nombre de cas. Dans le cas de l'application à TILV, la connaissance biologique du système ainsi que de ses paramètres nous paraît essentielle pour mener à bien une telle étude.

Annexe A

Quelques propriétés mathématiques des fonctions de Hill

L'expression analytique de la fonction de Hill est donnée par :

$$\forall x \in \mathbb{R}_+, \quad H_{n,\theta}(x) = \frac{x^n}{x^n + \theta^n}$$

Cette fonction dépend de deux paramètres qui permettent d'adapter la forme de la courbe au modèle représenté. Le paramètre n , qui généralement est un entier naturel mais qui peut être étendu à \mathbb{R}_+^* est appelé *ordre* de la fonction. Le paramètre θ est un réel strictement positif généralement nommé *seuil*. Nous décrivons ici certaines propriétés mathématiques des fonctions de Hill (la figure A.1 donne notamment la forme de ces fonctions pour différentes valeurs de n). Les dérivées premières et secondes de $H_{n,\theta}$ se calculent facilement :

$$\begin{aligned} H'_{n,\theta}(x) &= n\theta^n \frac{x^{n-1}}{(\theta^n + x^n)^2} \\ H''_{n,\theta}(x) &= n\theta^n x^{n-2} \frac{(n-1)\theta^n - (n+1)x^n}{(\theta^n + x^n)^3} \end{aligned}$$

Remarque A.1 On fera attention au fait que $H_{n,\theta}$ n'est dérivable en zéro que si $n \geq 1$. Dans le cas où $0 < n < 1$, la courbe admet une tangente verticale en l'origine. Lorsque $n = 1$, on retrouve l'expression de la fonction de Michaelis (cf. figure 1.5 au chapitre 1).

Nous déduisons de ces expressions plusieurs propriétés intéressantes du point de vue de la modélisation. Tout d'abord, $H'_{n,\theta}$ est strictement positive sur \mathbb{R}_+ , ce qui implique que la fonction $H_{n,\theta}$ est strictement croissante. D'autre part :

$$H_{n,\theta}(0) = 0 \quad \text{et} \quad \lim_{x \rightarrow +\infty} H_{n,\theta}(x) = 1$$

Enfin, pour tout $n > 0$ et pour tout $\theta > 0$, on a :

$$H_{n,\theta}(\theta) = \frac{\theta^n}{2\theta^n} = \frac{1}{2}$$

Cette propriété explique pourquoi θ est généralement appelé *seuil* :

$$H_{n,\theta}(x) > \frac{1}{2} \iff x > \theta$$

La courbe de la fonction $H_{n,\theta}$ est donc strictement croissante sur \mathbb{R}_+ , elle vaut 0 en 0 et 1 en $+\infty$. Pour que cette courbe soit une *sigmoïde*, il nous faut en plus vérifier qu'elle admet un unique point d'inflexion sur \mathbb{R}_+ .

Remarque A.2 Nous avons défini dans le chapitre 1 une *sigmoïde* comme étant une courbe croissante, valant 0 en 0, ayant une limite finie en $+\infty$ et ayant un unique point d'inflexion sur \mathbb{R}_+ . On trouvera dans [59] une définition plus complète, mais nous ne rentrerons pas plus dans les détails ici.

Pour trouver les points d'inflexion de la courbe de $H_{n,\theta}$, il nous faut résoudre l'équation $H''_{n,\theta}(x) = 0$ dans \mathbb{R}_+ . Cette équation entraîne :

$$x^n = \frac{n-1}{n+1} \theta^n$$

Cette courbe admet donc un unique point d'inflexion ρ strictement positif si et seulement si $n > 1$. Ce point d'inflexion est donné par :

$$\rho = \theta \sqrt[n]{\frac{n-1}{n+1}}$$

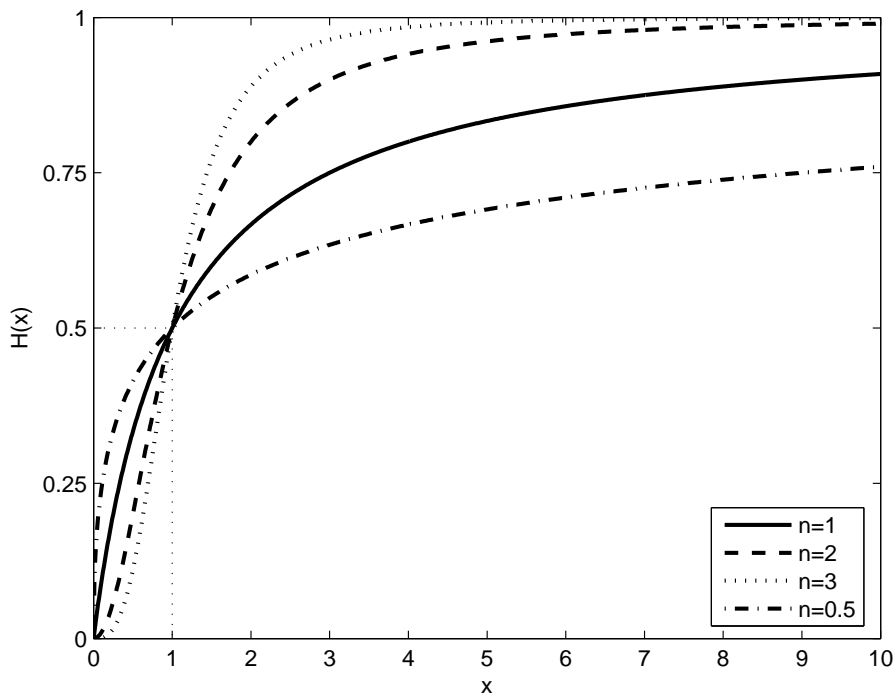


FIG. A.1: Formes de la courbe $H_{n,\theta}$, pour $\theta = 1$ et $n \in \{\frac{1}{2}, 1, 2, 3\}$. Nous retrouvons pour $n = 1$ la courbe de Michaelis. Pour $n = 2$ ou 3, $H_{n,\theta}$ a une forme sigmoïdale, tandis que pour $n = \frac{1}{2}$, la courbe présente une tangente verticale en l'origine.

Nous résumons à présent les différentes propriétés des fonctions de Hill (voir courbes sur la figure A.1 plus haut) :

- Pour tout $n > 0$ et tout $\theta > 0$, la fonction $H_{n,\theta}$ est positive et strictement croissante sur \mathbb{R}_+ . Elle vérifie :

$$H_{n,\theta}(0) = 0 \quad \text{et} \quad \lim_{x \rightarrow +\infty} H_{n,\theta}(x) = 1$$

- Dans le cas où $n > 1$, la courbe de $H_{n,\theta}$ admet un unique point d'inflexion strictement positif. Cette courbe est une sigmoïde.
- Dans le cas où $n = 1$, la courbe $H_{1,\theta}$ est du type courbe de Michaelis (cf. figure 1.5, chapitre 1).
- Dans le cas où $n < 1$, la courbe de $H_{n,\theta}$ n'admet pas de point d'inflexion positif. Elle admet au contraire une tangente verticale en 0.

Nous avons donné une interprétation de θ comme étant le seuil de la fonction $H_{n,\theta}$, c'est-à-dire la valeur qui sépare $[0, +\infty[$ en deux intervalles $I_1 = [0, K[$ et $I_2 =]K, +\infty[$ tels que :

$$\begin{cases} \forall x \in I_1, & H_{n,\theta}(x) < \frac{1}{2} \\ \forall x \in I_2, & H_{n,\theta}(x) > \frac{1}{2} \end{cases}$$

En ce qui concerne le paramètre n , que nous avons appelé *ordre* de $H_{n,\theta}$, il contrôle la forme de la courbe. Nous remarquons que :

$$H'_{n,\theta}(\theta) = \frac{n\theta^{2n-1}}{(2\theta^n)^2} = \frac{n}{4\theta}$$

Donc plus n augmente, plus la courbe est pentue au voisinage de θ . Lorsque n tend vers $+\infty$, la courbe de $H_{n,\theta}$ tend vers la courbe dite de Heaviside (ou créneau) :

$$H_\theta(x) = \begin{cases} 0, & \text{si } x < \theta \\ 1, & \text{si } x > \theta \end{cases}$$

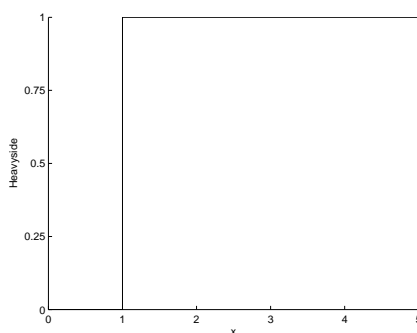


FIG. A.2: Fonction de Heaviside (ou créneau) H_θ , pour $\theta = 1$.

Annexe B

Implémentation de l'identification de réseaux

Nous proposons dans cette annexe une brève description de notre implémentation de l'algorithme d'inférence de réseaux d'automates booléens. Nous avons codé en `matlab` un certain nombre de fonctions permettant de créer et d'inférer de tels réseaux. Les figures contenues dans le chapitre 4 ont été obtenues grâce à ces fonctions. Nous montrons ensuite sur un exemple l'énumération de réseaux minimaux consistants avec une table partielle (cf. problème 2).

Représentation et création d'un réseau booléen

Nous rappelons qu'un réseau d'automate booléen est mathématiquement défini par un triplet $R = (X, B, F)$, dans lequel :

- X représente l'ensemble des automates. C'est un ensemble fini de cardinal n (la dimension du réseau). Généralement, les automates sont numérotés de 1 à n : $X = \{1, \dots, n\}$
- B est une matrice booléenne carrée de taille n
- F est une application de $\{0, 1\}^n$ dans lui-même.

Nous rappelons également qu'un réseau est sous sa forme *canonique* si, pour tout automate $i \in \{1, \dots, n\}$, l'ensemble $\{j \in \{1, \dots, n\} \mid b_{ij} = 1\}$ est égal au support de f_i .

Comme nous l'avons montré dans le chapitre 4, une fonction booléenne :

$$f : \{0, 1\}^n \longrightarrow \{0, 1\}$$

peut être identifiée de manière unique par un vecteur booléen U_f appartenant à $\{0, 1\}^{2^n}$ par :

$$\forall i \in \{1, \dots, 2^n\}, \quad (U_f)_i \stackrel{\text{def}}{=} f(\sigma_n^{-1}(i-1))$$

où $\sigma_n(x)$ est l'entier appartenant à $\{0, \dots, 2^n - 1\}$ dont le vecteur $x \in \{0, 1\}^n$ est la représentation en base 2 (i.e. $\sigma_n(x) = \sum_{i=0}^{n-1} x_{i+1} 2^i$). C'est cette représentation que nous avons choisie.

Grâce à cette représentation, nous pouvons considérer une fonction booléenne soit comme un vecteur booléen de dimension 2^n , soit comme un entier appartenant à $\{0, \dots, 2^{2^n} - 1\}$.

Afin de représenter un réseau, nous avons choisi d'utiliser deux tableaux à 2 dimensions (cell array) de tailles $n \times n$, notés *res* et *reg*. Les éléments d'un cell array en Matlab sont des matrices.

Le premier de ces deux tableaux contient les supports des fonctions f_i . Chaque colonne de *res* correspond à un automate du réseau. Si pour $i \in \{1, \dots, n\}$ la fonction f_i a un support de cardinal k , alors nous rangeons ce support à l'intersection de la i -ième colonne et de la k -ième ligne de *res* (rappelons que k est appelé connectivité réelle de f_i et est noté $\kappa(f_i)$). Le support de f_i est une partie $\Sigma = \{i_1, \dots, i_k\}$ de $\{1, \dots, n\}$ de cardinal k , nous choisissons de le représenter par le vecteur ligne (i_1, \dots, i_k) , dans lequel les i_j sont rangés dans l'ordre croissant. Le second tableau *reg* contient les fonctions booléennes f_i (parfois appelées règles booléennes), sous la forme de vecteurs lignes de dimension $2^{\kappa(f_i)}$. Ainsi, le réseau de dimension 3 défini par :

$$B = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}, \quad F(x_1, x_2, x_3) = \begin{pmatrix} x_2 \vee x_3 \\ \neg x_1 \\ x_1 \wedge x_2 \wedge x_3 \end{pmatrix}$$

sera représenté par les tableaux :

$$res = \begin{bmatrix} () & (1) & () \\ (2, 3) & () & () \\ () & () & (1, 2, 3) \end{bmatrix}, \quad reg = \begin{bmatrix} () & (1, 0) & () \\ (0, 1, 1, 1) & () & () \\ () & () & (0, 0, 0, 0, 0, 0, 0, 1) \end{bmatrix}$$

A partir de cette représentation, nous proposons un programme permettant de générer aléatoirement un réseau de dimension quelconque. Il faut pour cela générer, pour chaque automate $i \in \{1, \dots, n\}$ un support, c'est-à-dire une partie de $\{1, \dots, n\}$ de cardinal k (les éléments seront rangés dans l'ordre croissant), et une fonction booléenne $f_i : \{0, 1\}^k \rightarrow \{0, 1\}$, c'est-à-dire un vecteur booléen de dimension 2^k . En rentrant deux entiers N et K , l'utilisateur peut choisir de générer un réseau de dimension N dont tous les nœuds ont une connectivité réelle égale à K . Nous pouvons également générer un réseau dont tous les nœuds ont une connectivité réelle inférieure ou égale à K , où dont la moyenne des connectivité est égale à K , etc...

Afin d'assurer que les réseaux ainsi construits sont sous forme canonique, il nous reste à vérifier que les fonctions f_i produites ne sont pas dégénérées, c'est-à-dire qu'elles ne contiennent aucune variable muette. Par exemple, si pour $k = 3$ nous générons aléatoirement le vecteur de taille $2^3 = 8 : (1, 0, 1, 0, 1, 0, 1, 0)$, alors la génération a échoué puisqu'un tel vecteur correspond à la fonction :

$$f(x, y, z) = \neg z$$

qui est de connectivité réelle 1 et non 3. Nous devons donc générer un nouveau vecteur dans $\{0, 1\}^{2^3}$. Une fois cette vérification effectuée, nous avons généré un réseau canonique de dimension N .

Représentation d'un ensemble de réseaux

Le principal avantage d'une telle représentation est qu'elle nous permet de représenter, avec seulement deux cell arrays un ensemble de réseaux canoniques de dimension n .

Exemple B.1

Considérons les deux tableaux *res* et *reg* suivants :

$$res = \begin{bmatrix} (2) & (1) & () \\ (2,3) & () & (1,3) \\ () & (1,2,3) & (1,2,3) \end{bmatrix}, reg = \begin{bmatrix} (0,1) & (1,0) & () \\ (0,1,1,1) & () & (0,0,0,1) \\ () & (0,1,1,1,0,0,1,1) & (0,0,0,0,0,0,1) \end{bmatrix}$$

Ces deux tableaux représentent plusieurs réseaux, selon comment l'on choisit le support et la fonction correspondant à chaque nœud.

Par exemple, pour le nœud 1 (première colonne), nous avons le choix entre deux fonctions :

$$\begin{cases} f_1(x_2) & = x_2 \\ f_1(x_2, x_3) & = x_2 \vee x_3 \end{cases}$$

Pour le nœud 2 (deuxième colonne) :

$$\begin{cases} f_2(x_1) & = \neg x_1 \\ f_2(x_1, x_2, x_3) & = x_2 \vee (\neg x_1 \wedge x_3) \end{cases}$$

Et enfin pour le nœud 3 (troisième colonne) :

$$\begin{cases} f_3(x_1, x_3) & = x_1 \wedge x_3 \\ f_3(x_1, x_2, x_3) & = x_1 \wedge x_2 \wedge x_3 \end{cases}$$

Dans cet exemple, les tableaux *res* et *reg* représentent donc $2 \times 2 \times 2 = 8$ réseaux d'automates booléens canoniques différents.

Dans l'exemple que nous venons de proposer, pour chaque nœud $i = 1, 2$ ou 3 , nous avons le choix pour la fonction f_i entre plusieurs supports de dimensions différentes. Nous pouvons également ranger dans le tableau *res* plusieurs supports de même dimension. Supposons par exemple que, pour le nœud 3, nous ayons deux supports possibles de dimension 2 :

$$\begin{cases} f_3(x_1, x_3) & = x_1 \wedge x_3 \\ f_3(x_2, x_3) & = x_2 \vee x_3 \end{cases}$$

Alors l'élément $res[2, 3]$ sera une matrice de dimension 2×2 dont les lignes seront les deux supports :

$$res[2, 3] = \begin{pmatrix} 1 & 3 \\ 2 & 3 \end{pmatrix}$$

De même, l'élément $reg[2, 3]$ sera une matrice de dimension 2×4 dont les lignes seront les deux règles :

$$reg[2, 3] = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

L'utilité d'une telle représentation est de faciliter l'identification de réseaux. Lorsque nous lançons l'algorithme 3 d'identification partielle, nous cherchons les fonctions f_i consistantes avec une table, pour chaque nœud $i \in \{1, \dots, n\}$. Au fur et à mesure que nous trouvons une telle fonction $f_i(x_{i_1}, \dots, x_{i_{k_i}})$, nous rangeons son support $\Sigma = \{i_1, \dots, i_{k_i}\}$ dans $res[k_i, i]$ et la règle correspondante dans $reg[k_i, i]$. Une fois la recherche terminée, les deux tableaux *res* et *reg* contiennent un ensemble de réseaux consistants avec la table donnée en entrée. Nous pouvons alors les compter ou les énumérer.

Avant de passer à la partie suivante, qui montre en pratique comment répondre au problème 2 d'énumération des réseaux minimaux consistants par rapport à une table, nous revenons ici sur un point important évoqué dans le chapitre 4. Lorsque nous identifions des fonctions booléennes à partir d'une table de vérité partielle, nous ne sommes pas toujours en mesure d'identifier ces fonctions booléennes sur leur ensemble de définition tout entier (on se référera notamment à l'exemple 4.5). Ainsi, si pour un nœud de support $\{1, 2\}$, nous n'avons qu'une partie de la table de f :

x_1	x_2	$f(x_1, x_2)$
0	0	0
0	1	×
1	0	1
1	1	1

Alors notre programme renverra le vecteur : $(0, -1, 1, 1)$. Le -1 en seconde position signifie qu'il y a indétermination et donc que les deux vecteurs $v = (0, 1, 1, 1)$ et $v' = (0, 0, 1, 1)$ conviennent. Ces deux vecteurs correspondent aux fonctions :

$$\begin{cases} f(x_1, x_2) &= x_1 \vee x_2 \\ f'(x_1, x_2) &= x_2 \end{cases}$$

Sur cette exemple, nous voyons que $\kappa(f) = 2$ alors que $\kappa(f') = 1$. Donc même si l'algorithme a trouvé deux fonctions différentes consistantes avec les données, il n'y en a qu'une seule dont la connectivité réelle vaut 2. Puisque nous cherchons les réseaux minimaux, la fonction f' est à rejeter, car sinon l'algorithme l'aurait détecté à une étape antérieure et aurait trouvé que la dimension minimale du support de ce nœud était 1 et non 2.

De manière générale, si nous identifions une fonction f de support $\Sigma = \{i_1, \dots, i_k\}$, alors en posant \mathcal{J} le nombre d'indéterminées dans f (i.e. \mathcal{J} est égal au nombre de -1 dans le vecteur représentant f), alors le nombre de fonctions f consistantes avec les données, de support Σ est majoré par $2^{\mathcal{J}}$.

Identification partielle et énumération des réseaux minimaux

Avant de pouvoir lancer l'identification, nous devons disposer d'une table de vérité partielle $T = (E, S)$. Ces matrices E et S sont des matrices booléennes de dimension $p \times n$ (où p , la taille des entrées, est donnée par l'utilisateur). Nous rappelons que les lignes de E sont supposées deux à deux distinctes.

Une fois la table générée, nous pouvons lancer l'algorithme d'inférence partielle (alg. 3). Comme nous l'avons vu, cet algorithme établit d'abord, pour chaque nœud i la dimension minimale de son support : $d(i)$. Ensuite, il identifie tous les supports de dimension $d(i)$ pour lesquels il trouve une fonction f_i consistante avec T à la colonne i . Afin de stocker toutes ces informations, nous utilisons deux tableaux *res* et *reg* (cf. partie précédente). Dans *res*, nous entrons tous les supports minimaux trouvés. *res* $[d(i), i]$ contient donc une matrice $\mathcal{A}(d(i), i)$ à m_i lignes et $d(i)$ colonnes dont les lignes sont les supports de cardinal $d(i)$ du nœuds i . L'entier m_i nous donne le nombre de supports minimaux trouvés pour i .

D'après ce que nous avons expliqué plus haut, le tableau *reg* est construit de la même façon : *reg* $[d(i), i]$ va donc contenir une matrice $\mathcal{F}(d(i), i)$ à m_i lignes et 2^k colonnes dont les éléments appartiennent à $\{-1, 0, 1\}$. La j -ième ligne de $\mathcal{F}(k, i)$ contient la fonction f_i (sous la forme d'un

vecteur appartenant à $\{-1, 0, 1\}^{2^{d(i)}}$ associée au nœud i dont le support est la j -ième ligne de la matrice $\mathcal{A}(k, i)$.

Nous montrons dans ce qui suit sur un exemple particulier (en dimension 10) la résolution du problème 2, c'est-à-dire de l'énumération des réseaux minimaux par rapport à une table partielle.

Exemple B.2

Nous considérons la table partielle $T = (E, S)$ suivante ($n = 10, p = 20$) :

$$E = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix} \quad S = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

En lançant l'algorithme d'identification, nous trouvons le tableau *res* suivant :

$$res = \begin{bmatrix} () & () & () & () & () & () & () & () & () & () \\ () & () & () & () & () & \mathcal{A}(2,6) & \mathcal{A}(2,7) & () & () & () \\ () & () & \mathcal{A}(3,3) & \mathcal{A}(3,4) & () & () & () & \mathcal{A}(3,8) & \mathcal{A}(3,9) & () \\ \mathcal{A}(4,1) & \mathcal{A}(4,2) & () & () & () & () & () & () & () & \mathcal{A}(4,10) \\ () & () & () & () & \mathcal{A}(5,5) & () & () & () & () & () \\ () & () & () & () & () & () & () & () & () & () \\ () & () & () & () & () & () & () & () & () & () \\ () & () & () & () & () & () & () & () & () & () \\ () & () & () & () & () & () & () & () & () & () \end{bmatrix}$$

Avec, pour i allant de 1 à 10 :

$$m_1 = 1, m_2 = 6, m_3 = 1, m_4 = 2, m_5 = 7, m_6 = 1, m_7 = 1, m_8 = 1, m_9 = 11, m_{10} = 1$$

(Rappelons que m_i est égal au nombre de lignes des matrices $\mathcal{A}(d(i), i)$).

La lecture de ce tableau nous permet de trouver la dimension minimale du support de F : $D_{min}(F) = (4, 4, 3, 3, 5, 2, 2, 3, 3, 4)$. A partir des m_i , nous pouvons dénombrer l'ensemble des

supports minimaux consistants avec T . Appelons ce nombre n_s :

$$n_s = \prod_{i=1}^{10} m_i = 6 \times 2 \times 7 \times 11 = 924$$

Nous voulons à présent compter les réseaux minimaux par rapport à T . Pour cela, pour chaque automate $i \in \{1, \dots, n\}$ et pour chaque $j \in \{1, \dots, m_i\}$, il nous faut compter le nombre de -1 dans la j -ième ligne de la matrice $\mathcal{F}(i, d(i))$. Nous notons ce nombre $\mathfrak{J}_{i,j}$. Ainsi, pour le nœud 1, nous trouvons un seul support de dimension 4 :

$$\mathcal{A}(4, 1) = (3, 4, 8, 9)$$

et la fonction f_1 associée à ce support est donnée par le vecteur $\mathcal{F}(4, 1) \in \{-1, 0, 1\}^{2^4}$ suivant :

$$\mathcal{A}(4, 1) = (1, -1, 0, 0, 0, -1, 1, -1, 0, 0, 0, 0, 1, -1, 1, 0)$$

Ceci signifie que la fonction f_1 correspondante est indéterminée pour 4 valeurs, c'est-à-dire : $\mathfrak{J}_{1,1} = 4$.

Nous trouvons ensuite un majorant du nombre de réseaux minimaux par rapport à T :

$$\text{card}(\mathcal{M}(T)) \leq \prod_{i=1}^n \left(\sum_{j=1}^{m_i} 2^{\mathfrak{J}_{i,j}} \right)$$

Dans notre exemple, ceci donne :

$$\text{card}(\mathcal{M}(T)) \leq 8.57 \cdot 10^{11}$$

Sur cet exemple, nous voyons que ce majorant peut prendre des valeurs très élevées. Ceci provient du fait que la taille des données en entrée est faible $p = 20$ relativement à la taille du réseau ($n = 10$) et surtout à sa connectivité maximale. Ici, le nœud 5 est de connectivité maximale 5. Or, pour pouvoir identifier de façon unique une fonction à 5 variables il faut lui donner une valeur pour $2^5 = 32$ entrées distinctes.

Bibliographie

- [1] T. Akutsu, S. Kuhara, O. Maruyama, and S. Miyano. Identification of gene regulatory networks by strategic gene disruptions and gene overexpressions. *Proc. 9th ACM-SIAM Symposium on Discrete Algorithms*, pages 695–, 1998.
- [2] T. Akutsu, S. Miyano, and S. Kuhara. Identification of genetic networks from a small number of gene expression patterns under the boolean network model. *Proc. Pacific Symposium on Biocomputing*, pages 17–28, 1999.
- [3] T. Akutsu, S. Miyano, and S. Kuhara. Algorithms for inferring qualitative models of biological networks. *Pacific Symposium on Biocomputing*, 5 :290–301, 2000.
- [4] R. Alur, C. Belta, F. Ivančić, V. Kumar, M. Mintz, G.J. Pappas, H. Rubin, and J. Schug. Hybrid modeling and simulation of biomolecular networks. *HSCC 2001, LNCS 2034*, pages 29–32, 2001.
- [5] M. Antoniotti, B. Mishra, C. Piazza, A. Policriti, and M. Simeoni. Modeling cellular behavior with hybrid automata : bisimulation and collapsing. *Int. Workshop on Computational Methods in Systems Biology (CMSB03). LNCS. Springer.*, 2003.
- [6] M. Antoniotti, A. Policriti, N. Ugel, and B. Mishra. Xs-systems : extended s-systems and algebraic differential automata for modelling cellular behavior. *Proc. International Conference on High Performance Computing, HiPC*, pages 431–442, 2002.
- [7] R.J. Bagley and L. Glass. Counting and classifying attractors in high dimensional dynamical systems. *Journal of Theor. Biol.*, 183 :269–284, 1996.
- [8] C. Belta, J. Schug, T. Dang, V. Kumar, G. Pappas, H. Rubin, and P. Dunlap. Stability and reachability analysis of a hybrid model of luminescence in the marine bacterium vibrio fisheri. *CDC'01 - Conference on Decision and Control, Florida, USA.*, 2001.
- [9] L. Brenig and A. Goriely. Universal canonical forms for time-continuous dynamical systems. *Physical Review A*, 40(7) :4119–4122, 1989.
- [10] L. Brenig and A. Goriely. *Painlevé analysis and normal forms*, pages 211–238. Lecture Note Series 193. London Mathematical Society, 1994.
- [11] A.D. Bruno. *Power geometry in algebraic and differential equations*. North-Holland Mathematical Library, 2000.
- [12] J.L. Cherry and F.R. Adler. How to make a biological switch. *J. Theor. Biol.*, 203 :117–133, 2000.
- [13] O. Cinquin and J. Demongeot. Positive and negative feedback : striking a balance between necessary antagonists. *Journal of Theor. Biol.*, 216 :229–241, 2002.
- [14] G. Curien, D. Job, R. Douce, and R. Dumas. Allosteric activation of arabidopsis threonine synthase by s-adenosylmethionine. *Biochemistry*, 37 :13212–13221, 1998.

- [15] G. Curien, S. Ravanel, and R. Dumas. A kinetic model of the branch-point between the methionine and threonine biosynthesis pathways in *arabidopsis thaliana*. *European Journal of Biochemistry*, 270 :4615–4627, 2003.
- [16] H. de Jong. Modeling and simulation of genetic regulatory systems : a litterature review. *Rapport de recherche INRIA n° 4032*, 2000.
- [17] H. de Jong, J.L. Gouzé, C. Hernandez, M. Page, T. Sari, and J. Geiselmann. Qualitative simulation of genetic regulatory networks using piece-wise linear models. *Bull. Math. Biol.*, 66(2) :301–340, 2004.
- [18] J-P. Demailly. *Analyse numérique et équations différentielles*. Presses Universitaire de Grenoble, 1996.
- [19] B. Derrida and H. Flyvbjerg. The random map model : a disordered model with deterministic dynamics. *Journal of Phys.*, 48 :971–978, 1987.
- [20] P. D’haeseleer, S. Liang, and R. Somogyi. Gene expression data analysis and modeling. *Pacific Symposium on Biocomputing*, 1999.
- [21] J. Della Dora, A. Maignan, M. Mirica-Ruse, and S. Yovine. Hybrid computation. *Proc. International Symposium of Symbolic and Algebraic Computation*, pages 691–704, 2001.
- [22] R. Edwards and L. Glass. Combinatorial explosion in model gene networks. *Chaos*, 10(3) :691–704, 2000.
- [23] E. Farcot. *Etude d’une classe d’équations différentielles affines par morceaux modélisant des réseaux de régulation biologique*. Thèse de Doctorat, INP Grenoble, 2005.
- [24] M. Feinberg. *Lectures on chemical reaction networks*. Support de cours disponible à l’adresse : <http://www.che.eng.ohio-state.edu/~FEINBERG/LecturesOnReactionNetworks/>, 1980.
- [25] H. Flyvbjerg. An order parameter for networks of automata. *J. Phys. A : Math. Gen.*, 21 :955–960, 1988.
- [26] J.J. Fox and C.C. Hill. From topology to dynamics in biochemical networks. *Chaos*, 11(4) :809–815, 2001.
- [27] A. Girard. *Analyse algorithmique des systèmes hybrides*. Thèse de Doctorat, INP Grenoble, 2004.
- [28] A. Goriely. *Integrability and nonintegrability of dynamical systems : a singularity analysis approach*. Thèse de Doctorat de Sciences Physiques, Université Libre de Bruxelles, 1994.
- [29] B. Harris. Probability distributions related to random mappings. *Ann. Math. Stat.*, 31(4) :1045–1062, 1960.
- [30] M.A. Harrison. *Introduction to switching and automata theory*. McGraw-Hill series in systems science, New York - London, 1965.
- [31] M.W. Hirsch. The dynamical systems approach to differential equations. *Bulletin of American Mathematical Society*, 11(1) :1–64, 1984.
- [32] M.H. Holmes. *Introduction to perturbation methods*. Texts in Applied Mathematics, 20, Springer - Verlag, 1995.
- [33] J.J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences, USA*, 79 :2554–2558, 1982.

- [34] J.H. Hubbard and B.H. West. *Differential equations : a dynamical systems approach*. Texts in Applied Mathematics, Springer-Verlag, 2000.
- [35] F. Jacob and J. Monod. Genetic regulatory mechanisms in the synthesis of proteins. *J. Mol. Biol.*, 3 :318–356, 1961.
- [36] F. Jacob and J. Monod. On the regulation of gene activity. *Cold Spring Harbor Symp. Quant. Biol.*, 26 :193–211, 1961.
- [37] C. Jeffries, V. Klee, and P. Van den Driessche. When a matrix is sign stable?. *Can. J. Math.*, 29(2) :315–326, 1976.
- [38] W. Just, I. Shmulevich, and J. Konvalina. The number and probability of canalysing functions. *Physica D*, 197 :211–221, 2004.
- [39] S.A. Kauffman. Cellular homeostasis, epigenesis, and replication in randomly aggregated macro-molecular systems. *Journal of Cybernetics*, 1 :71–96, 1971.
- [40] S.A. Kauffman. Gene regulation networks : a theory for their global structure and behaviors. *Current Topics in Dev. Biol.*, 6 :145–182, 1971.
- [41] S.A. Kauffman. The large scale structure and dynamics of gene control : an ensemble approach. *Journal of Theor. Biol.*, 44 :167–190, 1974.
- [42] S.A. Kauffman. Emergent properties in random complex automata. *Physica D*, 10 :145–156, 1984.
- [43] S.A. Kauffman. *The origins of order, self-organization and selection in evolution*. Oxford University Press, 1993.
- [44] S.A. Kauffman. A proposal for using the ensemble approach to understand genetic regulatory networks. *Journal of Theor. Biol.*, 230 :581–590, 2004.
- [45] B. Lewin. *Genes VII*. Oxford University Press, 1999.
- [46] S. Liang, S. Fuhrman, and R. Somogyi. Reveal, a general reverse engineering algorithm for inference of genetic network architectures. *Pacific Symposium on Biocomputing*, 3 :18–29, 1998.
- [47] W.S. MacCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical BioPhysics*, 5 :115–133, 1943.
- [48] B. Martin. A geometrical hierarchy on graphs via cellular automata. *Rapport de recherche n° RR1999-35*, 1999.
- [49] J. Maybee and J. Quirk. Qualitative problems in matrix theory. *SIAM Review*, 11(1) :30–51, 1969.
- [50] H.H. McAdams and L. Shapiro. Circuit simulation of genetic networks. *Science*, 269 :650–656, 1995.
- [51] T. Mestl, E. Plahte, and S. Omholt. A mathematical framework for describing and analysing gene regulatory networks. *J. Theor. Biol.*, 176 :291–300, 1995.
- [52] M. Mignotte. *Mathematiques pour le calcul formel*. P.U.F. Paris, 1989.
- [53] J. Monod. *Le hasard et la nécessité*. Editions du Seuil, 1970.
- [54] J.D. Murray. *Mathematical biology*. Springer, Berlin (Biomathematics Vol. 19), 2002.
- [55] J. Von Neumann. *Theory of self-reproducing automata*. A.W. Burks Ed., University of Illinois Press, 1966.

- [56] S. Paris, C. Viemon, G. Curien, and R. Dumas. Mechanism of control of arabidopsis thaliana aspartate kinase-homoserine dehydrogenase by threonine. *Journal of Biological Chemistry*, 278(7) :5361–5366, 2003.
- [57] E. Plahte, T. Mestl, and S.W. Omholt. Global analysis of steady points for systems of differential equations with sigmoid interactions. *Dyn. Stab. Syst.*, 9 :275–291, 1994.
- [58] E. Plahte, T. Mestl, and S.W. Omholt. Feedback loops, stability and multistationarity in dynamical systems. *Journal of Biol. Syst.*, 3(2) :409–413, 1995.
- [59] E. Plahte, T. Mestl, and S.W. Omholt. A methodological basis for description and analysis of systems with complex switch-like interactions. *J. Math. Biol.*, 36 :321–348, 1998.
- [60] L.S. Pontryagin. *Ordinary differential equations*. Addison-Wesley Publishing Company, 1962.
- [61] M. Ptashne. *A genetic switch*. 2nd ed. Cell Press and Blackwell Scientific Publications, 1992.
- [62] F. Robert. *Les systèmes dynamiques discrets.*, volume 19. Springer, Mathématiques et Applications, 1995.
- [63] Z. Roka. Simulations between cellular automata on cayley graphs. *Theoretical Computer Science*, 225 :81–111, 1999.
- [64] A. Rondepierre. Hybrid optimal control. *International Journal of Control (soumis)*.
- [65] L. Sanchez and D. Thieffry. A logical analysis of the drosophila gap-gene system. *Journal of Theor. Biol.*, 211 :115–141, 2001.
- [66] M. Santillan and M.C. Mackey. Dynamic behavior in mathematical models of the tryptophane operon. *Chaos*, 11 :261, 2001.
- [67] M.A. Savageau. Alternative designs for a genetic switch : analysis of switching times using piecewise power-law representation. *Math. Biosci.*, 180 :237–253, 2002.
- [68] M.A. Savageau and E.O. Voit. Recasting nonlinear differential equations as s-systems : a canonical form. *Mathematical Biosciences*, 87 :83–115, 1987.
- [69] M.A. Savageau and E.O. Voit. Recasting nonlinear differential equations as s-systems : a canonical nonlinear form. *Math. Biosci.*, 87 :83–115, 1987.
- [70] C.E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27 :379–423, 623–656, 1948.
- [71] E.H. Snoussi. Necessary conditions for multistationarity and stable periodicity. *Journal of Biol. Syst.*, 6(1) :3–9, 1998.
- [72] A. Sorribas and M. Cascante. Structure identifiability in metabolic pathways : parameter estimation in models based on the power-law formalism. *Biochem. Journal*, 298 :303–311, 1994.
- [73] C. Soulé. Graphic requirements for multistationarity. *Complexus*, 1 :123–133, 2003.
- [74] D. Thieffry, A.M. Huerta, E. Perez-Rueda, and J. Collado-Vides. From specific gene regulation to genomic networks : a global analysis of transcriptional regulation in escherichia-coli. *Bioessays*, 20 :433–440, 1998.
- [75] R. Thomas. Laws for the dynamics of regulatory networks. *Int. J. Dev. Biol.*, 42 :479–485, 1998.
- [76] R. Thomas and R. D’Ari. *Biological feedback*. CRC Press, 1990.

- [77] R. Thomas and M. Kaufman. Multistationarity, the basis of cell differentiation and memory. i. structural conditions of multistationarity and other nontrivial behavior. *Chaos*, 11(1) :170–179, 2001.
- [78] R. Thomas and M. Kaufman. Multistationarity, the basis of cell differentiation and memory. ii. logical analysis of regulatory networks in terms of feedback circuits. *Chaos*, 11(1) :180–195, 2001.
- [79] L. Tournier. Approximation of dynamical systems using s-systems theory : application to biological systems. *Proc. of International Symposium on Symbolic and Algebraic Computation, Beijing, China*, pages 317–324, 2005.
- [80] E.O. Voit. *Computational analysis of biochemical systems*. Cambridge University Press, 2000.
- [81] E.O. Voit and M.A. Savageau. Accuracy of alternative representations for integrated biochemical systems. *Biochemistry*, 26 :6869–6880, 1987.
- [82] G. Weisbuch. *Dynamique des systèmes complexes, une introduction aux réseaux d'automates*. Editions du CNRS, 1989.
- [83] S. Wiggins. *Introduction to applied nonlinear dynamical systems and chaos*. Springer-Verlag, 1990.
- [84] S. Wolfram. Statistical mechanics of cellular automata. *Rev. Mod. Phys.*, 55 :601–644, 1983.
- [85] S. Wolfram. *Theory and Applications of Cellular Automata*. World Scientific Publishing, 1986.
- [86] P. Wong, S. Gladney, and J.D. Keasling. Mathematical model of the lac-operon : inducer exclusion, catabolite repression and diauxic growth on glucose and lactose. *Biotechnol. Prog.*, 13 :132–143, 1997.
- [87] A. Wuensche. Genomic regulation modeled as a network with basins of attraction. *Pacific Symposium on Biocomputing, World Scientific, Singapore*, pages 89–102, 1998.
- [88] N.H. Xuong. *Mathématiques discrètes et informatique*. Masson, 1992.
- [89] G.M. Ziegler. *Lectures on polytopes*. Graduate Texts in Mathematics, Springer-Verlag, New-York, 1995.