

UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN PHYSIQUE

PAR

ANDRÉ ST-AMAND

APPROCHE DES MODÈLES DE CROISSANCE
PAR LES AUTOMATES CELLULAIRES

MAI 1997

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire ou de cette thèse a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire ou de sa thèse.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire ou cette thèse. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire ou de cette thèse requiert son autorisation.

Et comme la multitude des lois fournit souvent des excuses aux vices, en sorte qu'un État est bien mieux réglé, lorsque, n'en ayant que fort peu, elles y sont fort étroitement observées; ainsi, au lieu de ce grand nombre de préceptes dont la logique est composée, je crus que j'aurais assez des quatre suivants, pourvu que je prisse une ferme et constante résolution de ne manquer pas une seule fois à les observer. Le premier était de ne recevoir jamais aucune chose pour vraie, que je ne la connusse évidemment être telle : c'est à dire d'éviter soigneusement la précipitation et la prévention ; et de ne comprendre rien de plus en mes jugements que ce qui se présenterait si clairement et si distinctement à mon esprit, que ne n'eusse aucune occasion de le mettre en doute. Le second, de diviser chacune des difficultés que j'examinerais, en autant de parcelles qu'il se pourrait et qu'il serait requis pour les mieux résoudre. Le troisième, de conduire mes pensées, en commençant par les objets les plus simples et les plus aisés à connaître, pour monter peu à peu, comme par degrés, jusqu'à la connaissance des plus composés ; et supposant même de l'ordre entre ceux qui ne se précédent point naturellement les uns des autres. Et le dernier, de faire partout des dénombremens si entiers, et des revues si générales, que je fusse assuré de ne rien omettre.

*Discours de la méthode
René Descartes*

REMERCIEMENTS

Je tiens tout d'abord à remercier mon directeur de mémoire *Paul Lavallée* pour son dévouement et l'ouverture d'esprit dont il a su faire preuve tout au long de l'accomplissement de ce travail. Je tiens également à remercier mes deux meilleurs amis *Robert Papineau* et *Sinh LeQuoc* pour l'intégrité et le support moral dont ils ont su faire preuve à mon égard dans les moments difficiles. Merci à ma mère *Marie-Paule* pour m'avoir permis, par son support financier, de poursuivre mes études sans tracas.

RÉSUMÉ

Ce mémoire analyse la possibilité de résoudre, via l'utilisation d'un réseau cartésien, un système d'équations différentielles correspondant à un modèle diffusif de croissance biologique donné. Pour confirmer ou infirmer cette possibilité, la construction d'un automate cellulaire probabiliste est entreprise. L'implantation de cet automate cellulaire sur un réseau cartésien permet la simulation de plusieurs modèles de croissance qui ont été historiquement postulés. L'analyse des résultats des simulations indique qu'il est possible, pour certains modèles de croissance, de résoudre adéquatement le système d'équations différentielles qui lui correspond.

TABLE DES MATIÈRES

REMERCIEMENTS.....	i
RÉSUMÉ.....	ii
LISTE DES FIGURES.....	v
LISTE DES TABLEAUX.....	vii
LISTE DES SYMBOLES.....	viii
INTRODUCTION.....	1
CHAPITRE I	
RÉSEAU ET DYNAMIQUE DES POPULATIONS.....	6
1.1 Définitions des concepts associés à un réseau.....	7
1.2 Opérations effectuées sur un réseau.....	9
1.3 Représentation de l'état d'une cellule.....	10
1.3.1 Implantation d'une cellule.....	12
1.3.2 Définition de l'automate cellulaire.....	16
1.4 Présentation du réseau couplé.....	17
1.5 Distribution statistique sur le réseau couplé.....	19
1.6 Obtention de la variation de la densité moyenne d'une espèce.....	23
CHAPITRE II	
MODÈLES CLASSIQUES DE CROISSANCE ET LOI D'ACTION DE MASSE.....	25
2.1 Loi d'action de masse.....	26
2.2 Type de mécanismes réactionnels.....	27
2.3 Construction des modèles de croissance via les mécanismes réactionnels.....	29
2.3.1 Construction des modèles classiques à 1 espèce.....	29
2.3.2 Construction des modèles classiques à 2 espèces.....	30
2.4 Discussion.....	31
2.5 Forme vectorielle de la loi d'action de masse.....	33
2.6 Loi d'action de masse et théorème de Chapmann-Kolmogorov.....	34
2.7 Équation fondamentale des probabilités de transitions.....	35

CHAPITRE III

ANALYSE THÉORIQUE DES MODÈLES CLASSIQUES DE CROISSANCE.....37

3.1 Analyse de faisabilité des modèles classiques à 1 espèce.....	38
3.1.1 Équations générales des modèles à 1 espèce.....	38
3.1.2 Présentation des contraintes à apporter aux équations.....	41
3.1.3 Analyse du modèle de <i>Malthus</i>	42
3.1.4 Analyse du modèle <i>logistique</i>	43
3.1.5 Rôle de l'échelle de temps « <i>h</i> ».....	45
3.2 Analyse de faisabilité des modèles classiques à 2 espèces.....	48
3.2.1 Équations associées au modèle de <i>Lotka Volterra étendu</i>	51
3.2.2 Analyse du modèle de <i>Lotka Volterra classique</i>	56
3.2.3 Analyse du modèle de <i>Lotka Volterra étendu</i>	57

CHAPITRE IV

EXPÉRIMENTATION DES MODÈLES CLASSIQUES SUR RÉSEAU.....62

4.1 Présentation du programme informatique.....	63
4.1.1 Exemples de calculs préliminaires.....	64
4.2 Analyse de stabilité des modèles classiques.....	70
4.3 Simulation du modèle <i>logistique</i>	74
4.3.1 Description de l' <i>expérience 1</i>	75
4.3.2 Analyse des résultats de l' <i>expérience 1</i>	75
4.3.3 Description de l' <i>expérience 2</i>	78
4.3.4 Analyse des résultats de l' <i>expérience 2</i>	78
4.4 Simulation du modèle de <i>Lotka Volterra étendu</i>	79
4.4.1 Description de l' <i>expérience 1</i>	79
4.4.2 Analyse des résultats de l' <i>expérience 1</i>	79
4.4.3 Description de l' <i>expérience 2</i>	84
4.4.4 Analyse des résultats de l' <i>expérience 2</i>	85
4.5 Discussion.....	85

CONCLUSION.....88

ANNEXE A

DÉMONSTRATION DE LA PROPOSITION (2.32).....91

ANNEXE B

PROGRAMME INFORMATIQUE « ÉCOSYSTÈME ».....108

BIBLIOGRAPHIE.....174

LISTE DE FIGURES

Figures	Page
1.1 Réseau cartésien de dimension 4×4	7
1.2 Disposition de 2 espèces sur un réseau cartésien.....	8
1.3 Extraction d'une cellule du réseau.....	8
1.4 État non admissible d'une cellule du réseau.....	8
1.5 Réseau initial.....	9
1.6 Réseau après l'opération « collision ».....	9
1.7 Réseau après l'opération « rotation ».....	9
1.8 Réseau après l'opération « propagation ».....	9
1.9 Fonction informatique « digit ».....	14
1.10 Fonction informatique « état espèce ».....	14
1.11 Bijection entre l'état du réseau et la matrice E.....	15
1.12 État du réseau réservé à l'espèce I.....	18
1.13 État du réseau réservé à l'espèce II.....	18
1.14 Mise en étage des réseaux.....	19
1.15 État d'une cellule reliée à ses voisins proches.....	21
2.1 Correspondance entre le champ de force et le champ des mécanismes réactionnels.....	34
3.1 Région R de faisabilité correspondant au modèle <i>logistique</i>	44
3.2 Extension de la région R de faisabilité du modèle <i>logistique</i>	47
3.3 Espace des phases S^2 et ses frontières.....	60
4.1 Courbe théorique correspondant au modèle <i>logistique</i>	71

4.2	Trois trajectoires types du modèle de <i>Lotka Volterra étendu</i>	73
4.3	Courbe expérimentale correspondant au modèle <i>logistique</i>	76
4.4	Valeur moyenne de la densité normalisée en fonction du coefficient a_2	76
4.5	Valeur minimale, moyenne et maximale de la densité en fonction du coefficient a_2	77
4.6	Densité d'équilibre normalisée en fonction du facteur d'échelle h	78
4.7	Première trajectoire observée correspondant au modèle de <i>Lotka Volterra étendu</i>	80
4.8	Deuxième trajectoire observée correspondant au modèle de <i>Lotka Volterra étendu</i>	81
4.9	Troisième trajectoire observée correspondant au modèle de <i>Lotka Volterra étendu</i>	82
4.10	Zoom de la troisième observée trajectoire du modèle de <i>Lotka Volterra étendu</i>	83
4.11	Quatrième trajectoire observée correspondant au modèle de <i>Lotka Volterra étendu</i>	84
4.12	Effet du facteur d'échelle h sur les densités d'équilibre normalisées.....	86
4.13	Effet du facteur d'échelle h sur les densités d'équilibre normalisées.....	87
A.1	Région R de sommation.....	97
A.2	Transformation de la région R de sommation.....	99
A.3	Région R^* de sommation.....	99
A.4	Ajout de deux points à la région R^* de sommation.....	99
A.5	Effet du changement de variables sur la région R^* de sommation.....	100

LISTE DES TABLEAUX

Tableaux	Page
3.1 Liste des équations correspondant au modèle de <i>Lotka Volterra étendu</i>	52
3.2 Liste des équations correspondant au modèle de <i>Lotka Volterra classique</i>	55
4.1 Algorithme du programme informatique « écosystème ».....	64
4.2 Coefficients α_1 et α_2 utilisés lors des simulations du modèle <i>logistique</i>	71

LISTE DES SYMBOLES

\subset	Inclusion ensembliste
\forall	Quantificateur universel
\exists	Quantificateur existentiel
\in	Appartenance ensembliste
\emptyset	Ensemble vide
\mathbb{N}	Ensemble des entiers naturels
	Indicateur de fin de démonstration.
G	Graphe
X	Ensemble de sommets
U	Ensemble d'arêtes
S	Ensemble des étiquettes associées aux arêtes
S^n	Ensemble résultant des n produits cartésiens de l'ensemble S ou espace des phases
P	Opérateur propagation
R	Opérateur rotation
C	Opérateur collision
∇^2	Opérateur Laplacien
t	Temps
h	Facteur d'échelle
m	Ordre du réseau
r	Localisation d'une cellule ou localisation d'une position
L^n	Réseau n -couplé
L_i	i -ème réseau correspondant à l'espèce de type i

$\alpha_i^k(t)$	Variable aléatoire du nombre d'êtres de type i occupant l'arête d'étiquette k au temps t
$\alpha_i(t)$	Variable aléatoire du nombre d'êtres de type i présent dans une cellule au temps t
$\alpha(t)$	Variable aléatoire vectorielle contenant les $\alpha_i(t)$
$p_i^k(t)$	Probabilité qu'un être de l'espèce i occupe l'arête k en 1 essai
$\beta(1, p)$	Fonction de probabilité correspondant à une variable aléatoire de type Bernoulli
$\beta(n, p)$	Fonction de probabilité correspondant à une variable aléatoire de type binomial
$\rho_i(t)$	Densité d'êtres correspondant à l'espèce i au temps t
$\rho(t)$	Vecteur densité contenant les densités $\rho_i(t)$
$\bar{\rho}_i(t)$	Densité moyenne d'êtres correspondant à l'espèce i au temps t
$\binom{n}{x}$	Coefficient du binôme de Newton dont la valeur est : $\frac{n!}{x!(n-x)!}$
E	Espérance mathématique ou valeur moyenne d'une variable aléatoire
ΔE	Variation de l'espérance mathématique
α_i	Quantité de la substance X_i nécessaire pour que la réaction aie lieu
β_i	Quantité de la substance X_i résultant de la réaction
k_i	Constante de transformation lorsque $\alpha_i X_i$ se transforme en $\beta_i X_i$
n_i	Nombre total de particules de la substance X_i au temps t
Ω	Volume dans lequel la réaction prend lieu
\mathbf{F}	Champ vectoriel générant la dynamique entre les diverses espèces
$F_i(\rho)$	i-ème composante du champ vectoriel induit par la loi d'action de masse.
α	Vecteur correspondant à l'état initial d'une cellule avant sa transformation
β	Vecteur correspondant à l'état final d'une cellule après sa transformation
$P_{\alpha \rightarrow \beta}$	Probabilité correspondante à la transition des états α vers β
$P_{\alpha \rightarrow \beta}$	Matrice des probabilités de transitions
$a_l^{(i)}$	Coefficients du polynôme induit par la loi d'action de masse réservé à l'espèce i
$\kappa_l^{(i)}$	Coefficients reliés à $a_l^{(i)}$ par une transformation linéaire
$\mathbf{1}$	Vecteur de dimension n contenant le nombre 1 à chaque entrée

R	Région de sommation ou région de faisabilité
R^*	Région R de sommation transformée ou extension de la région R de faisabilité
A_i	Ensemble des coefficients admissibles correspondant à l'espèce i
p_{sup}	Borne supérieure associée au facteur d'échelle \hbar
$\rho_{initiale}$	Condition initiale de la densité lors d'une simulation sur le réseau cartésien
$\rho_{équilibre}$	Valeur de la densité d'équilibre théorique
$\rho_{observé}$	Valeur de la densité d'équilibre observée lors d'une simulation sur le réseau cartésien
$\rho_{normalisé}$	Rapport de la densité d'équilibre observée avec la densité d'équilibre théorique
$\rho_{minimale}$	Valeur minimale de la densité observée lors d'une simulation sur le réseau cartésien
$\rho_{maximale}$	Valeur maximale de la densité observée lors d'une simulation sur le réseau cartésien

INTRODUCTION

Les modèles de croissance font l'objet d'une étude rigoureuse depuis le XVIII siècle. Historiquement, l'apparition de ces modèles se rattache à une grande variété de disciplines; si bien, qu'il est difficile de distinguer l'origine exacte de cet intérêt. Contrairement à beaucoup de champs d'études modernes qui se sont graduellement vus confiné par la voie de la spécialisation, l'étude des phénomènes de croissances a su préserver un caractère d'intérêt général parmi une panoplie de chercheurs. Tant et si bien le cas que la plupart des disciplines scientifiques d'aujourd'hui donnent à leur façon un visage toujours renouvelé de ces phénomènes. Qu'il soit question en chimie, de *production* d'une substance par une réaction entre plusieurs autres, en physique, de *désintégration* d'un matériau radioactif, en médecine, du *rythme* avec lequel une maladie contagieuse se propage, en économie, de *capitalisation* par intérêt composé, en science sociale, de la *prolifération* d'un message, et en biologie de *naissance et de mortalité* d'une espèce donnée, tous ces exemples recoupent une même idée : celle de mieux comprendre les phénomènes de croissance et de décroissance. La mise au point des équations différentielles a permis historiquement de modéliser, à un niveau ou à un autre, la plupart des exemples cités précédemment. Mais puisque le thème principal de ce mémoire est relié aux phénomènes de croissance dans un cadre essentiellement biologique, nous allons présenter par ordre chronologique d'apparition, les différents modèles de croissance qui ont vu le jour dans ce cadre de la biologie. Ces modèles seront présentés sous la forme d'équations différentielles.

Le modèle de croissance le plus simple qui soit porte étrangement^{*} le nom de *modèle Malthusien*³² ou plus simplement *modèle de Malthus* (1766-1834). Ce modèle propose que le taux de reproduction d'une espèce est proportionnel au nombre d'individus présents au temps t . Si on appelle $N(t)$ ce nombre, alors le modèle de *Malthus* se présente sous la forme différentielle suivante :

$$\frac{dN}{dt} = kN \quad \text{où } k > 0 \quad (1)$$

* La doctrine *Malthusienne* est fondée sur la prescription de règles d'abstinences car laissés à eux-même les individus occasionneront une explosion démographique dont la conséquence sera l'épuisement des ressources vitales du milieu.

Ainsi vu, ce modèle stipule que la reproduction d'une espèce est illimitée, et ce, quelles que soient les ressources dont elle dispose. En 1836, le démographe Belge *Pierre-François Verhulst* (1804-1849) apporte des corrections au modèle de *Malthus* en ajoutant qu'une espèce doit être limitée en nombre par la capacité M du milieu. Ce modèle porte le nom de **modèle de Verhulst**^{5,32}. Sous forme différentielle ce modèle propose que :

$$\frac{dN}{dt} = k N (N - M) \quad \text{où } k > 0 \quad (2)$$

À travers la littérature scientifique actuelle il est plutôt rare de retrouver (2) sous l'étiquette « *modèle de Verhulst* ». En effet, il est beaucoup plus fréquent de rencontrer (2) sous le nom de : **modèle logistique**. Dans le cadre de ce mémoire, nous utiliserons ce dernier nom pour désigner (2) puisqu'il semble faire l'unanimité en littérature scientifique. Une première analyse de (2) nous conduit aux considérations qui suivent. Si $N > M$ il y a croissance de la population et si $N < M$ il y a décroissance de la population; preuve de quoi la constante M joue le rôle d'une barrière sur le phénomène de reproduction. On peut raffiner²³ autant que l'on souhaite (1) et (2). Toutefois, parmi toutes les modifications dont (1) et (2) peuvent faire l'objet, seule celle qui incorpore la présence d'une nouvelle espèce sera retenue. On doit au mathématicien italien *Vito Volterra*³²⁻³³ (1860-1940) l'introduction du premier modèle à deux espèces où les thèmes de croissance et de compétition entre une proie et un prédateur sont incorporés. Si $N_1(t)$ et $N_2(t)$ représentent respectivement le nombre de prédateurs et le nombre de proies au temps t , alors la forme originale que *Vito Volterra* proposa en 1926 pour décrire, sous forme différentielle, l'interaction de la proie et du prédateur est la suivante :

$$\frac{dN_1}{dt} = -a N_1 + b N_1 N_2 \quad (3)$$

$$\frac{dN_2}{dt} = c N_2 - d N_1 N_2 \quad (4)$$

Dans (3) et (4) les constantes a, b, c et d sont toutes positives. Les termes mixtes « $N_1 N_2$ » jouent le rôle de compétition entre la proie et le prédateur, tandis que les termes simples jouent le rôle de reproduction ou de décès. Plus précisément, l'équation (3) révèle qu'en l'absence de proie l'espèce prédateur subit une décroissance tandis que la présence de proie favorise la croissance des prédateurs. Chez l'espèce prédateur, la décroissance est due au terme « $-a N_1$ » tandis que leur croissance est due au terme « $b N_1 N_2$ ». Pour l'équation (4), l'absence de prédateur favorise la croissance de l'espèce proie puisque le terme de croissance « $c N_2$ » l'emportera alors sur le terme de décroissance « $-d N_1 N_2$ ». Contrairement à (1) et (2), on remarque à propos de (3) et (4)

qu'il est plus difficile de faire un bilan conséquent de cette compétition pour la survie. En 1925, le scientifique *A.J.Lotka*^{23,26} postule (3) pour décrire le comportement de certaines réactions chimiques où deux substances semblent entretenir entre elles ce même type de dynamique. Pour cette raison, bien qu'il ne semble pas accepté de tous^{31,32}, certains auteurs donnent à (3) le nom de *modèle de Lotka Volterra*. Dans le cadre de ce mémoire, nous utiliserons cette dernière appellation pour désigner le système formé des équations (3) et (4).

Du point de vue théorique des équations différentielles, une différence majeure distingue le système de *Lotka Volterra*. En effet, à la différence des modèles à une espèce (1) et (2), le modèle de *Lotka Volterra* ne peut être résolu théoriquement. Dans ce contexte, l'ajout de termes visant à complexifier davantage (3) et (4) n'apportera certes pas plus de facilité à sa résolution. Cette dernière remarque n'a certes pas empêché les scientifiques de postuler une multitude d'autres modèles qui traduisent encore mieux la richesse du monde biologique^{5,23}. Une première extension du modèle de *Lotka Volterra* peut être obtenue par l'ajout du terme quadratique de l'équation logistique (2). On obtient alors :

$$\frac{dN_1}{dt} = -\alpha N_1 + b N_1 N_2 - c N_1^2 \quad . \quad (5)$$

$$\frac{dN_2}{dt} = d N_2 + e N_1 N_2 - f N_2^2 \quad (6)$$

Le système d'équations (5) et (6) forme en quelque sorte une synthèse des systèmes (1), (2) et (3),(4). Les équations (5) et (6) ne semblent pas porter de nom spécifique dans la littérature. Or, puisque nous y référerons régulièrement tout au long de ce mémoire, nous allons lui conférer le nom de *modèle de Lotka Volterra étendu*. Malheureusement, il semble que certains auteurs^{1,2} confondent (5) et (6) avec le modèle de *Lotka Volterra* formulé historiquement par les équations (3) et (4). Ainsi, pour éviter toute ambiguïté lors de ce travail, nous allons donc clairement distinguer (3) et (4) par le nom *modèle de Lotka Volterra classique*.

Le travail que se propose d'entreprendre ce mémoire est d'essayer de reproduire, par l'utilisation conjointe d'un automate cellulaire probabiliste et d'un réseau cartésien, les quatre modèles de croissances que nous venons d'exposer. La présentation des concepts de « réseau cartésien » et « d'automate cellulaire probabiliste » fait l'objet du chapitre I. La méthodologie, basée sur l'utilisation des réseaux, connaît actuellement plusieurs succès dans des disciplines comme la mécanique des fluides et la chimie. En effet, il semble possible¹⁻¹⁴ d'utiliser un réseau cartésien pour étudier le comportement de certaines réactions chimiques. Or, comme les mécanismes de croissance qui caractérisent cette dernière discipline semblent aussi avoir leurs images dans un

cadre de nature biologique (à ce propos, il nous suffit pour l'instant de rappeler l'origine du nom de « modèle de *Lotka Volterra* »), il est légitime de se demander s'il est possible d'étendre l'utilisation de ce réseau à la simulation de phénomènes de croissances de nature biologique. Toutefois, pour parvenir à ce but, nous devrons nous inspirer de la méthodologie servant à décrire, sous forme différentielle, la transformation d'une ou plusieurs substances dans le cadre de la dynamique moléculaire. Il nous faudra donc identifier quelles sont les idées qui sont susceptibles d'être transportées du champ de la dynamique moléculaire au champ de la biologique. Cette recherche de similitude entre deux champs d'activité, qui semble en apparence assez éloignés, fait l'objet du chapitre II. Par la suite, pour réussir à simuler de façon adéquate ces systèmes d'équations différentielles, nous nous devrons d'établir les règles et l'outillage nécessaires à la réalisation de cette simulation. Cette recherche de cohérence entre un modèle exprimé sous forme d'équations différentielles et sa simulation sur un réseau fait l'objet du chapitre III et de l'annexe A. Une fois ce travail accompli, nous pourrons alors entreprendre la simulation des modèles de croissance dont nous avons fait la description. Cette partie à saveur expérimentale fait l'objet du chapitre IV.

Globalement, ce mémoire cherche donc à explorer divers modèles de croissance par l'utilisation d'un automate cellulaire. Nous allons voir qu'un automate cellulaire est une structure microscopique capable de contrôler le nombre d'individus pouvant naître ou disparaître à un temps t sur un territoire (nous verrons au chapitre I que le « réseau » joue le rôle du « territoire ») et d'ajuster ce contrôle à un modèle de croissance fixé *a priori*. On peut imaginer le rôle que doit tenir cet automate par la situation simple suivante. En forêt, un loup rencontre un lapin. Le loup mange-t-il le lapin systématiquement ? Si la réponse est négative, dans quelle proportion cet acte de prédation doit-il réussir ? Un lapin rencontre une lapine ou un loup rencontre une louve. Dans quelle proportion l'acte de reproduction doit-il s'accomplir pour chacune de ces espèces ? En apportant à ces questions des réponses arbitraires du type : un loup dévore un lapin une fois sur trois, un lapin se reproduit une fois sur deux tandis qu'un loup se reproduit une fois sur cinq, il devient possible de simuler, sur ordinateur, l'évolution du nombre de lapins et de loups. Bien que cette méthode puisse conduire à d'agréables surprises, bien qu'il soit effectivement possible de mesurer expérimentalement à quel rythme un lapin se reproduit ou se fait dévorer, il n'en demeure pas moins que cette méthodologie laisse la question suivante sans réponse : de quel modèle procède l'interaction entre les lapins et les loups ? Cette même question se pose lorsqu'elle est posée dans le contexte suivant : si l'espèce «loup» et l'espèce «lapin» interagissent suivant le modèle de *Lotka Volterra étendu*, alors dans quelle proportion l'acte de reproduction et de prédation doit-il s'accomplir pour chacune de ces espèces ? Nous verrons que c'est précisément l'automate cellulaire qui donne la possibilité aux espèces d'interagir suivant un modèle donné.

Sélectionner les états appropriés pour reproduire un modèle de croissance fixé a priori, tel sera le sens du concept d'*automate cellulaire*. En ce sens, on peut dire que la construction formelle de cette entité qu'est « l'automate cellulaire » constitue le thème principal de ce mémoire.

Mais avant toute chose, si l'unité fondamentale qui soutient les sciences dites exactes est le nombre, on peut dire alors que l'unité fondamentale qui fonde la biologie est « l'être vivant ». Par ailleurs, du point de vue de la biologie, la classification des différents êtres vivants sous l'appellation « d'espèce » est fondamentale car il s'agit à la base de bien signifier que chacune de ces « espèces », participant à un écosystème quelconque, a ses propres modalités d'existence. Dans une optique où on envisage de recréer par simulation sur ordinateur un système écologique ou plusieurs espèces évolueront, il nous faut donc aussi trouver une représentation numérique pouvant caractériser chacune de ces espèces ainsi que la façon dont elle survit au sein de son environnement géographique. Il y a donc là deux obstacles dont il faut s'affranchir avant même d'espérer faire interagir entre elles ces dites espèces. Ces deux obstacles s'énoncent sous la forme interrogative suivante :

- i) Comment caractériser « géographiquement » une espèce sur un réseau cartésien ?
- ii) Comment caractériser « quantitativement » une espèce sur un réseau cartésien ?

Nous allons donc maintenant aborder ces deux questions.

Chapitre I

RÉSEAU ET DYNAMIQUE DES POPULATIONS

Dans ce chapitre, nous allons caractériser l'évolution de plusieurs espèces sur un réseau cartésien. À ce titre, les opérateurs P , R et C sont respectivement décrits. De plus, le problème de la quantification de l'état d'une cellule est résolu. Suite à la présentation des fonctions de distributions statistiques associées à l'état d'une cellule, la variation de la densité moyenne est déduite par les processus de Markov et le théorème de Chapman-Kolmogorov.

1.1 DÉFINITIONS DES CONCEPTS ASSOCIÉS À UN RÉSEAU

Le concept de réseau cartésien est simple à saisir. Nous allons graduellement en faire la description complète. Pour situer des espèces vivantes dans l'espace, plusieurs représentations sont possibles lorsqu'on envisage de créer un territoire discret sur lequel évolueront plusieurs espèces. En effet, le nombre de façons de diviser un terrain en lots est infini. Parmi cette infinité de façons, il nous faut en choisir une qui soit simple et régulière. Le réseau cartésien en offre une. C'est probablement la plus simple qui soit. Elle consiste à subdiviser un territoire carré en un très grand nombre de petits carrés de même aire. Ce choix de réseau possède plusieurs avantages. Comme nous allons le voir, ce réseau coïncide avec un mode de représentation avec lequel nous sommes familiers. Formellement, un réseau cartésien est un *graphe* $G = G(X, U)$ où X est un ensemble fini de sommets et U est l'ensemble des arêtes qui relient les sommets. Plus précisément, si on définit un ensemble E par $E = \{0 \dots N\}$, $N \in \mathbb{N}$, alors $X = E \times E$. De plus, si on construit l'ensemble U par $U = \{\text{des arêtes adjointes entre un sommet et ses voisins proches}\}$ alors dans ce contexte on peut dire qu'un réseau cartésien est entièrement caractérisé par le graphe³⁰ G . Signalons que ce sont les arêtes qui permettront le déplacement des êtres vivants sur le réseau G . Nous appellerons *noeud* un sommet de coordonnées (i, j) . Une unité fondamentale sera utilisée tout au long de ce travail: c'est la *cellule*. Nous appellerons *cellule* un *noeud* muni des quatre arêtes qui le relient à ses voisins proches. Une illustration d'un réseau cartésien G est donnée à la figure 1.1. Du point de vue informatique, la structure de ce réseau permet l'utilisation de structure de données dont le type est une matrice. Il y a donc une correspondance directe à établir entre le territoire discret sur lequel évoluent plusieurs espèces et la configuration de la matrice à un temps donné; ces deux objets coïncideront. Le choix de ce réseau possède quelques inconvénients dont nous parlerons plus longuement plus loin. Bien que la littérature²⁸ propose l'utilisation d'un réseau hexagonal pour éliminer ces désavantages, le type de réseau qui sera emprunté lors de ce travail sera le *réseau cartésien*.

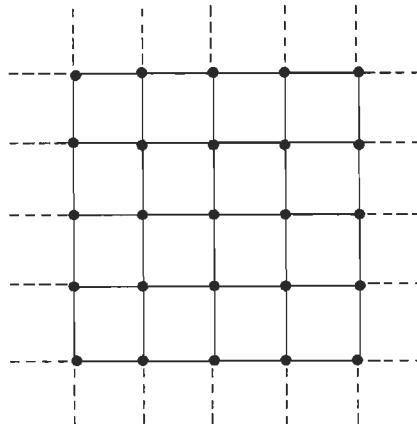


Fig 1.1 Réseau cartésien

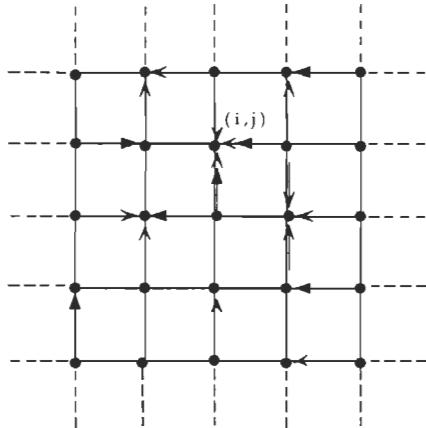


Fig 1.2 Disposition de 2 espèces sur un réseau cartésien.

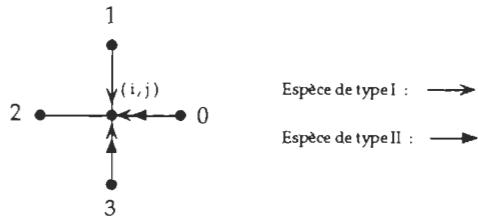


Fig 1.3 Extraction d'une cellule du réseau.

À présent, nous allons discuter de l'état dans lequel peut se retrouver une *cellule* de ce réseau. Nous définirons *l'état d'une cellule* par le nombre d'êtres total qui occupent ses arêtes et par l'ensemble des vecteurs indiquant les directions vers lesquelles se dirige chacun de ces êtres. Tel que l'illustre la figure 1.2, plusieurs espèces peuvent se retrouver au même instant à une *cellule* de ce réseau. Sans perte de généralité, l'état d'une *cellule* est illustré à la figure 1.3. Pour rendre plus claire la présente discussion, cette *cellule* a été détachée de la figure 1.2 et annotée à sa droite par la légende appropriée. Bien que ces figures rendent la représentation statique, on peut attribuer à chacune de ces flèches un caractère dynamique. Aux figures 1.2 et 1.3, deux types de flèches ont été utilisées dans le but de distinguer deux classes distinctes d'êtres. Ces flèches indiquent la direction vers laquelle les êtres vivant quitteront la coordonnée (i, j) du réseau et par extension on peut leur associer une mesure de vitesse. Nous allons graduellement voir comment caractériser de façon unique l'état de cette *cellule*. Pour l'instant, nous avons simplement présenté comment les différentes *cellules* se retrouvent affectées par les différentes espèces qui évoluent sur un réseau cartésien. Pour compléter cette description, il est important de signaler que nous devons apporter une contrainte sur les états possibles d'une *cellule*. En effet, on impose *qu'il ne puisse y avoir, à une cellule donnée de ce réseau, plus d'un individu par espèce qui possède la même direction et, par extension, la même vitesse sur une même arête*. Ce principe porte de nom de *principe d'exclusion*. Nous verrons pourquoi nous appliquons ce principe en chacune des *cellules*. En guise d'illustration simple, la figure 1.4 donne un exemple de configuration qui n'est pas admissible dans l'ensemble des états possibles pour une *cellule* (i, j).

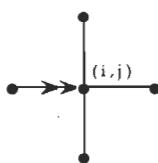


Fig 1.4 État non admissible d'une cellule du réseau cartésien

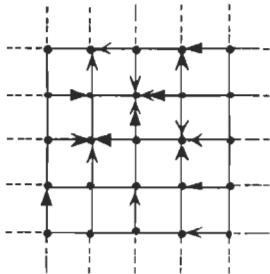


Fig 1.5 Réseau initial

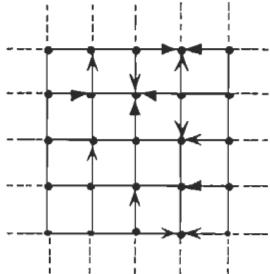


Fig 1.6 Réseau après l'opération « collisions »

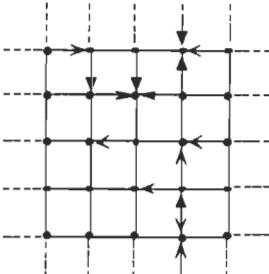


Fig 1.7 Réseau après l'opération « rotation »

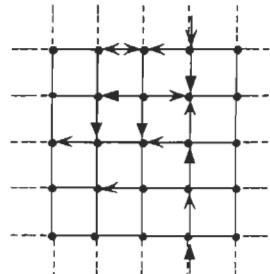


Fig 1.8 Réseau après l'opération « propagation »

1.2 OPÉRATIONS EFFECTUÉES SUR UN RÉSEAU

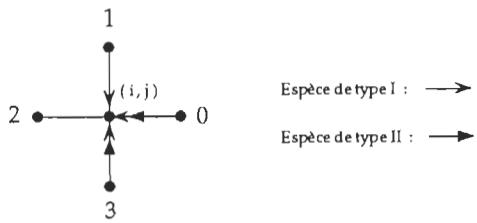
Nous avons donné une description simple mais complète sur la manière dont les différentes espèces sont représentées qualitativement sur un réseau cartésien. Les règles d'évolution ne sont pas pour autant caractérisées. À présent, nous allons préciser davantage ces règles qui seront instaurées à chacune des cellules du réseau. Par ailleurs, la description de ces règles nous aidera à identifier les obstacles dont il faut s'affranchir pour réaliser un programme informatique visant à les reproduire. En chacune des *cellules* nous allons définir trois règles^{1,4,7,9,11} ou opérations : la *propagation*, la *rotation*, et la *collision*. Elles seront respectivement notées par **P**, **R** et **C**. En principe se sont tous trois des opérateurs matriciels. Chacune des *cellules* du réseau se les verra appliquer successivement. Formellement, si on appelle **T** le résultat de cette composition alors $\mathbf{T} \equiv \mathbf{P} \circ \mathbf{R} \circ \mathbf{C}$. Au cours du présent travail nous aurons l'occasion de justifier pourquoi de telles opérations sont ainsi appliquées aux cellules du réseau. Pour l'instant, signalons que la présente discussion ne vise qu'à bien faire comprendre quel effet produit cette suite d'opérations sur une *cellule* (i, j) du réseau. Débutons tout d'abord par l'opérateur *collision* **C**. En termes simples nous allons supposer que la figure 1.5 constitue une photo du réseau, prise au temps t , avant que l'opération collision **C** n'ait été appliquée. Dans ce contexte la figure 1.6 illustre l'état du réseau après que l'opération collision ait été appliquée sur le réseau de la figure 1.5. Voyons l'effet produit. On remarque alors qu'à certaines *cellules* des êtres sont disparus, qu'à d'autres des êtres sont apparus, tandis qu'à d'autres *cellules* rien d'apparent ne semble s'être produit. En dynamique des populations, ces apparitions et disparitions seront l'analogie des naissances et des mortalités. Parmi les trois opérations **P**, **R** et **C**, la seule opération susceptible de modifier le nombre d'êtres total sur le réseau est la collision **C**. Indiquons simplement pour l'instant que l'opérateur **C** sera assujetti à des lois de probabilités. Lors de ce travail, nous allons voir que la construction de l'opérateur collision **C** est fort complexe. Très peu d'articles discutent de cette difficulté à mettre à jour cet opérateur collision **C** même s'il est le plus important des trois. Passons à la deuxième opération. Tel que convenu, l'opération collision est suivie de l'opération rotation **R**. Cette rotation est aléatoire et rigide. Elle consiste à modifier, de façon aléatoire et rigide à un

noeud (i, j), l'ensemble des vecteurs directions vers lesquelles chacun des êtres se dirige. Précisons à l'instant les termes *rigide* et *aléatoire*. Par rotation *rigide* \mathbf{R} nous entendons qu'à une cellule donnée, toutes les vecteurs directions seront simultanément pivotés du même angle. Pour effectuer cette rotation rigide, seules les valeurs d'angles de $\pi/2$ ou $-\pi/2$ seront admises^{7,9,11}. Cet angle de pivotement sera choisi au hasard. En ce sens, l'opération « *rotation* » est bel et bien *aléatoire*. Nous verrons pourquoi il en est ainsi au cours de ce travail. Le réseau de la figure 1.7 illustre un des scénarios possibles lorsqu'on applique la rotation rigide au réseau de la figure 1.6. Passons à la troisième opération : la propagation \mathbf{P} . Comme son nom l'indique, cette opération vise essentiellement à redistribuer chaque être vers la cellule avoisinante appropriée^{7,9,11}. Une image simple devrait élucider les mots « cellule avoisinante appropriée ». La figure 1.8 illustre ce qu'il advient du réseau de la figure 1.7 lorsque l'opération propagation \mathbf{P} lui est appliquée. La description de la règle de propagation n'est pas encore complétée car un petit détail a été omis. En effet, nous n'avons pas encore mentionné comment la propagation s'effectue lorsqu'un être se situe sur une frontière d'un réseau. Pour rendre régulière la règle de propagation, nous allons rendre périodiques⁷⁻¹¹ les frontières d'un réseau. La figure 1.8 illustre bien l'effet de cette périodicité aux frontières lorsque l'opération propagation est appliquée au réseau de la figure 1.7. Une fois la propagation terminée, le pas de temps t passe de t à $t + h$. À chaque pas de temps h la suite d'opérations $\mathbf{P} \circ \mathbf{R} \circ \mathbf{C}$ est appliquée au réseau entier. À ce temps $t + h$ le réseau de la figure 1.8 se voit appliquer la règle de collision \mathbf{C} (non illustré), suivie d'une rotation \mathbf{R} , suivi de la propagation \mathbf{P} , suite à quoi le temps passe de $t + h$ à $t + 2h$. Et le cycle recommence. La dynamique que fait naître sur un réseau G la transformation répétée $\mathbf{T} \equiv \mathbf{P} \circ \mathbf{R} \circ \mathbf{C}$ devrait être claire. Ajoutons en terminant que la description de ces trois règles indique ce qu'un programme informatique doit exécuter pour faire évoluer plusieurs espèces sur un réseau cartésien par la suite d'opérations $\mathbf{P} \circ \mathbf{R} \circ \mathbf{C}$. À présent, nous sommes en mesure de présenter la façon de quantifier l'état d'une *cellule*.

1.3 REPRÉSENTATION DE L'ÉTAT D'UNE CELLULE

Bien que la transformation $\mathbf{T} \equiv \mathbf{P} \circ \mathbf{R} \circ \mathbf{C}$ ait été présentée, deux questions demeurent en suspens. La première concerne la façon dont nous allons réussir à quantifier et à suivre dans le temps l'état d'un réseau. Le seconde, beaucoup plus complexe, concerne la façon dont nous allons réussir à identifier les règles de collisions sur les états microscopiques de sorte que l'évolution des espèces soit en accord avec une dynamique prescrite par des équations qui dictent les lois macroscopiques. Nous allons prendre ces questions une à une. Penchons-nous tout d'abord sur la première. Cette question concerne surtout le point de vue informatique puisque la façon dont sera quantifié l'état d'une cellule se traduit, dans le cadre de la préparation d'un document préalable à l'élaboration d'un programme, par : quelle structure de données allons nous utiliser pour quantifier

l'état d'une cellule ? Du point de vue informatique la façon de répondre à cette question est primordiale. Une réponse laborieuse et complexe donnera lieu à une implantation informatique qui sera assurément laborieuse et complexe. La problématique consistant à bien quantifier l'état d'une *cellule* doit donc être résolue avec le plus de simplicité et d'élégance possible. Voyons comment l'information pertinente sera codée. Pour caractériser l'état d'une *cellule* (i, j) en terme des espèces qui l'occupent, on utilisera la représentation binaire^{4,5,28}. Sur un réseau cartésien quatre arêtes sont adjacentes à chacun des noeuds. Pour quantifier l'état d'une cellule, nous allons tout d'abord étiqueter chacune des arêtes d'un noeud par les nombres entiers de 0 à 3. En général, si le réseau possédait m arêtes alors chacune des arêtes d'un noeud serait étiquetée par les nombres entiers de 0 à $m-1$. L'idée simple et astucieuse¹⁻¹¹ consiste à établir une correspondance biunivoque entre la disposition des espèces qui occupe cette *cellule* (i, j) et d'un mot binaire. Pour établir cette bijection à une *cellule* (i, j) nous allons réservé pour chaque espèce un mot de m bits. Notons que sur un réseau cartésien $m = 4$. En guise d'exemple, si deux espèces évoluent sur ce réseau alors deux mots de quatre bits seront nécessaires pour établir cette bijection. Si n espèces évoluent sur un réseau quelconque alors n mots de m bits sont nécessaire pour caractériser l'état d'une *cellule*. La règle de correspondance posée par la suite est simple : si un être occupe l'arête dont l'étiquette est k alors un 1 est placé dans la case $k+1$ du mot correspondant à cette espèce. Voyons un exemple. Supposons que deux espèces évoluent sur un réseau cartésien. Pour les fins de l'exemple distinguons les respectivement par espèces de type I et de type II. La figure 1.3 est reproduite ci-dessous. Nous supposerons que cette cellule occupe le noeud (i, j) de ce réseau. La configuration est donc la suivante :



Quelle est alors le mot binaire associé à l'état de ce noeud ? Ici, puisque nous sommes en présence de deux espèces et puisque $m = 4$ alors deux mots de 4 bits sont nécessaires. Le premier mot est réservé pour caractériser l'état de l'espèce de type I et le second, adjoint au mot binaire de type II est réservé pour caractériser l'état de l'espèce de type II. Globalement, la structure de donnée nécessaire à la représentation de ces deux mots est la suivante :

Numéro d'étiquette du type I				Numéro d'étiquette du type II			
3	2	1	0	3	2	1	0

En inspectant l'état d'une *cellule* (*i*, *j*) proposé ci-haut, et en vertu de la règle de correspondance déjà établie, l'état d'une *cellule* est caractérisé par le mot binaire contenu à la dernière ligne du tableau suivant :

Numéro d'étiquette du type I				Numéro d'étiquette du type II			
3	2	1	0	3	2	1	0
1	0	1	1	1	0	0	1

Ici, le mot binaire apparaît coupé en deux parties : une partie concerne « l'état du type I », et l'autre « l'état du type II ». En réalité, le sectionnement de ce mot binaire ne sera pas effectif. Pour unifier ces deux parties nous allons attacher la base 2 approprié à chaque entrée de ce mot. Ceci étant fait on obtient alors :

Numéro d'étiquette du type I				Numéro d'étiquette du type II			
3	2	1	0	3	2	1	0
1	0	1	1	1	0	0	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

En vertu de ce qui vient d'être dit, l'état d'une *cellule* (*i*, *j*) se retrouve donc quantifié par le nombre booléen $b = 1\ 0\ 1\ 1\ 1\ 0\ 0\ 1$. De plus, notons immédiatement qu'il y a une *bijection* entre l'état d'une *cellule* (*i*, *j*) et sa représentation booléenne. Toutefois, pour préserver cette bijection, il faut appliquer le *principe d'exclusion* à chaque espèce. Comme nous l'avons déjà mentionné auparavant, cela signifie que l'on s'interdit qu'une espèce donnée puisse occuper une même arête plus d'une fois. Finalement, puisque qu'il y a aussi une bijection entre un nombre en base deux et un nombre en base 10 il est alors clair que $b = 1\ 0\ 1\ 1\ 1\ 0\ 0\ 1$ en base 2 $\Leftrightarrow b = 185$ en base 10. La justification de ce changement de base apparaîtra pertinent au fil des lignes qui suivent.

1.3.1 IMPLANTATION D'UNE CELLULE.

Revenons au problème concernant la quantification de l'état d'une *cellule* (*i*, *j*). Jusqu'à présent nous avons su résoudre l'aspect statique de la quantification de l'état d'une *cellule* (*i*, *j*) en lui associant un nombre booléen « *b* ». Or, comme nous avons eu l'occasion de l'illustrer auparavant, la transformation $T \equiv P \circ R \circ C$ modifie régulièrement l'état d'une *cellule* (*i*, *j*); nous avons effectivement eu l'occasion de remarquer l'effet dynamique de cette transformation *T*. Nous voilà au coeur du problème de nature informatique : comment allons-nous gérer, de façon efficace, le nombre binaire « *b* » lorsque l'état de cette *cellule* se verra modifié par la suite d'opérations $P \circ R \circ C$? Il est clair que si l'état d'une *cellule* est modifié alors le nombre binaire qui lui correspond doit l'être aussi. Comment s'y prendre pour tenir ce nombre binaire à jour ? Cette question semble apparemment simple. Mais du point de vue informatique, mieux vaut nous

méfier de cette simplicité apparente. Puisque ce problème concerne avant tout le type de structure de données pertinentes à utiliser, certaines fonctions ont été détachées du programme « écosystème » pour y être commentées. Le programme entier est présenté à l'annexe B. Pour comprendre, il faut saisir le fonctionnement de ces fonctions car d'une part, elles effectuent les tâches liées au présent sujet de discussion, et d'autre part, le très grand nombre de fois où elles sont appelées en cours d'exécution signale qu'elles ont une certaine importance. Faisons le point en soulevant la question suivante : qu'est-ce qui est modifié par la transformation T ? Nous nous apercevons qu'à chaque cycle de calcul il faut mettre à jour :

- i) l'ensemble des vecteurs directions de toutes les espèces occupant la *cellule* (i, j).
- ii) le nombre d'êtres de chaque espèce occupant les arêtes du noeud (i, j).

Soit une *cellule* (i, j) à m arêtes sur lesquelles on superpose n espèces. Supposons qu'au temps t l'état d'une *cellule* (i, j) soit quantifié par b . Les opérations $R \circ C$ sont appliquées sur la *cellule* (i, j). Supposons que ce nouvel état soit quantifié par b' . Comment détecter combien d'êtres de type k ont disparu à cette *cellule* (i, j) suite à ces opérations ? Du point de vue informatique, il est important de répondre le plus simplement possible à cette question. Associer à la *cellule* (i, j) un tableau à $n \times m$ entrées et de le fouiller, pour vérifier si un 1 ou un 0 s'y trouve, est une opération ardue; trop d'opérations machines sont nécessaires et le jeu d'indices à mettre en branle pour soutirer une information concernant l'état d'une espèce de type k n'apporte aucune régularité. Une idée plus simple^{5,28} consiste à utiliser de l'algèbre de Boole en faisant appel aux fonctions élémentaires de cette algèbre. Ces fonctions élémentaires sont les suivantes : OR, OR NOT, AND, AND NOT, SHL et SHR. L'opération SHL est une abréviation anglaise pour « shift left » tandis que SHR est une abréviation anglaise pour « shift right ». Ces deux opérations ont respectivement pour effet de décaler un mot binaire vers la gauche et la droite. Les fonctions relevant de l'algèbre de Boole sont ni plus ni moins les outils de base pour effectuer de façon simple les calculs de comparaisons entre deux nombres b et b' en base 2. L'utilisateur du langage C ou Pascal peut recourir à ces fonctions qui lui seront parfaitement transparentes. Par transparence, il faut entendre que si b et b' sont en base 10 et que l'on effectue, par exemple, l'opération b OR b' alors le résultat est aussi retourné en base 10, quoique l'opération ait eu lieu en base 2. En guise d'exemple, si $b = 12$ et $b' = 5$ alors b AND $b' = 4$. À présent, nous sommes en mesure de répondre à la question soulevée précédemment : comment détecter combien d'êtres de type k ont disparu suite aux opérations $R \circ C$? La réponse à cette question est simple si on conjugue ces dernières opérations à un peu de code informatique. La fonction « *digit* » illustrée à la figure 1.9 compte le nombre de 1 présents dans le mot binaire associé à l'espèce de type k . Les variables de réception sont « *etat* » et « *espece* ». La variable « *etat* » est le nombre binaire b correspondant à l'état d'une *cellule* donnée, tandis que la deuxième variable « *espece* » est le type d'espèce à

dénombrer à une cellule donnée. Ici, l'espèce de type I est appelé « loup » et l'espèce de type II est appelé « lapin ».

```

function digit(etat:integer;espece:les_etres):integer;
var
  i,k,a,no_individu:integer;
begin
  case espece of
    loup: no_individu:=1 ;
    lapin: no_individu:=2 ;
  end;
  a:= etat shr ( 4 * ( nb_espece - no_individu ) );
  k:=0;
  for i:=1 to 4 do
    begin
      inc(k,a and 1);
      a:=a shr 1;
    end;
  digit:=k;
end;
```

Figure 1.9 Fonction informatique retournant le nombre d'êtres de type k présents dans une cellule.

Considérons deux exemples simples. L'évaluation de $digit(185, \text{loup})$ retourne la valeur 3 tandis que l'évaluation de $digit(185, \text{lapin})$ retourne la valeur 2. Ceci est bien conforme aux détails de calculs que nous avons fait précédemment. En guise d'exemple, l'opération $185 \text{ SHR } 4 = 11$. Dans ce cas, le mot binaire a été décalé de 4 espaces vers la droite. Comme on l'a vu précédemment, la représentation binaire du nombre 185 est la suivante :

1	0	1	1	1	0	0	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

Après avoir effectué l'opération $185 \text{ SHR } 4$ la représentation binaire du résultat est la suivante :

0	0	0	0	1	0	1	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

L'opération SHL produit essentiellement le même effet que SHR sauf que cette fois-ci les bits sont déplacés vers la gauche. Par exemple, après avoir effectué l'opération $185 \text{ SHL } 4$ la représentation binaire du résultat est la suivante :

1	0	0	1	0	0	0	0
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

S'il est intéressant de connaître le nombre d'êtres de type k qui sont présents à un noeud (i, j), il est aussi impératif de connaître le nombre binaire correspondant à l'état propre de l'espèce de type k . Par *état propre* il faut entendre l'état qui ne concerne qu'une espèce à une *cellule* (i, j). Dans le cas d'un réseau cartésien, cet état propre correspond forcément à un nombre binaire entre 0 et $2^4 - 1$. Sur un réseau quelconque à m arêtes, cet état propre correspond à un nombre binaire entre 0 et $2^m - 1$. La fonction « *etat_espece* », illustrée à la figure 1.10, retourne précisément ce nombre binaire correspondant à l'état propre de l'espèce de type k . Considérons deux exemples simples. L'évaluation de *etat_espece(185, loup)* retourne la valeur 11 tandis que l'évaluation de *etat_espece(185, lapin)* retourne la valeur 9. Ces valeurs sont bien celles qui correspondent respectivement aux états propres de l'espèce « loup » et « lapin ». Voilà deux fonctions qu'il apparaissait important de présenter ne serait-ce que pour illustrer ce qu'il faut faire pour gérer la dynamique de l'état d'un noeud (i, j). Les opérations OR, OR NOT, AND, AND NOT n'ont pas été commentées ici car elles relèvent de l'arithmétique bien connu de l'algèbre de Boole²⁸. Ces dernières ont été utilisées à profusion lors de la construction du programme informatique « écosystème ». L'utilisation de cette algèbre est de mise si l'on considère qu'un

```
function etat_espece(etat:integer,espece:les_etres):integer;
var
  no_individu,nombre,i,p:integer;
begin
  case espece of
    loup: no_individu:=1 ;
    lapin: no_individu:=2 ;
  end;
  nombre:=0;
  etat:=etat shr ( 4 * ( nb_espece - no_individu ) );
  for i:=1 to 4 do
    begin
      p:=exposant(2,i-1);
      nombre:=(nombre) or (etat and p);
    end;
  etat_espece:=nombre;
end;
```

Fig 1.10 Fonction informatique retournant le nombre binaire correspondant à l'état propre d'une espèce de type k

réseau minimal typique a des dimensions de 100×100 . Si telle est la dimension du réseau alors à chaque pas de temps « h » les fonctions « *digit* » et « *etat_espece* » seront utilisées au moins 100×100 fois. D'ailleurs, il est clair que la manière dont la *cellule* est construite est surtout motivée par la facilité avec laquelle l'algèbre booléenne peut être utilisée dans un cadre informatique.

La discussion qui vient d'avoir lieu s'est attardée à montrer comment l'état d'une *cellule* (i, j) est quantifiée. Plus globalement, l'ensemble des états des *cellules* du réseau sera contenu dans une matrice de même dimension que ce réseau. Pour rendre plus transparent l'état du réseau, chacune des entrées de la matrice **E** a été écrite en base 10 à la figure 1.11. Pour retrouver la configuration donnée d'une *cellule* (i, j) il faut prendre l'entrée (i, j) de la matrice **E** et traduire ce nombre en base 2. La figure 1.11 nous montre la correspondance entre l'état de ce réseau et la matrice **E**.

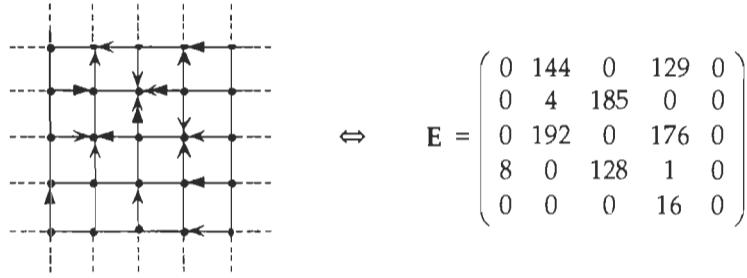


Fig 1.11 Bijection entre l'état d'un réseau et la matrice **E**

Il est clair que le contenu de **E** sera modifié lorsque les espèces vivantes évolueront sur ce territoire. Cette remarque devrait relever l'importance des fonctions « *digit* » et « *etat_espece* ». Le problème de la quantification et de la représentation des espèces qui évoluent sur un réseau étant achevé, considérons maintenant le problème concernant les règles décisionnelles que l'on va attacher aux *cellules* de ce réseau.

1.3.2 DÉFINITION DE L'AUTOMATE CELLULAIRE

Nous avons eu l'occasion de voir quel effet produit la suite d'opérations **P**, **R** et **C** sur une *cellule* (i, j). Seul l'opérateur **C** laisse planer une ambiguïté. En effet, bien que nous ayons dit qu'une *collision* consiste soit en l'annihilation d'un être (mortalité), soit en la création d'un être (naissance) soit en laissant inchangée chacune des espèces (survie), il n'en demeure pas moins que nous n'avons nullement dit quand et à quel rythme ces *collisions* auront lieu. Par là il faut entendre que si une *cellule* (i, j) est dans une configuration donnée la collision résultera-t-elle en la naissance, en la mortalité ou en la survie des individus en jeu ? Nous rejoignons là le second problème énoncé au début de la section 1.3. Ce problème s'énonçait ainsi : comment réussir à

identifier les règles de collisions pour que l'évolution des espèces soit en accord avec une dynamique prescrite par certaines lois ? Ce problème est de taille et sa réponse constitue l'objet même de ce mémoire. Il est à noter que l'on pourrait définir arbitrairement des règles de collisions, quitte à juger de l'intérêt du résultat une fois la simulation exécutée. Cette approche phénoménologique⁵ possède certes l'avantage de fournir rapidement une image du phénomène, mais elle enferme la pensée scientifique dans un amoncellement de faits expérimentaux qui, bien qu'intéressants, ne permettent plus de remonter à la cause du phénomène produit. Pour cette raison, cette approche phénoménologique ne sera donc pas utilisée dans le cadre de ce mémoire. Le point de vue essentiel qui occupe ce mémoire est de donner un modèle théorique pour que des règles de collisions puissent répondre adéquatement à un mécanisme prescrit *a priori*. Pour l'instant, indiquons simplement qu'une *cellule* munie des règles d'action associées à l'opérateur C sera plus généralement désignée par les mots « *automate cellulaire* ». On peut résumer ces derniers propos en disant que l'automate cellulaire sera entièrement caractérisé lorsque les règles qui régiront les interactions en chaque cellule(i, j) seront explicitées. Pour résoudre ce problème, nous allons tout d'abord essayer de déterminer la *fonction de probabilité associée au nombre d'êtres total pouvant se retrouver à une cellule (i, j)*.

1.4 PRÉSENTATION DU RÉSEAU COUPLÉ

Jusqu'à présent nous avons vu ce qu'est un réseau, comment une espèce est représentée par une *cellule* et de quelle manière est quantifié l'état des ces espèces sur le réseau entier. Le problème majeur que nous avons soulevé précédemment consiste à doter chaque *cellule* de règles de collisions bien précises. Pour nous affranchir de ce problème nous allons utiliser une approche basée sur des considérations probabilistes. Nous verrons graduellement en quoi cette approche nous permet d'apporter des éclaircissements quant à la nature des règles de collision à planter. Quoi qu'il en soit, le travail entrepris dans cette section a pour but de construire la distribution statistique du nombre d'êtres de type i , pouvant se trouver à un noeud fixe r du réseau. Tout d'abord, pour faciliter le traitement visant à obtenir cette distribution statistique nous allons supposer que chaque espèce évolue sur son propre réseau. Les figures 1.12 et 1.13 illustrent comment le réseau L^2 a été découplé^{1,4,7}. Cette approche n'altère en rien tout ce qui a été dit précédemment; elle n'est qu'une façon plus simple d'illustrer comment peut se construire la distribution conjointe du nombre d'êtres total présent à une *cellule* (i, j). Pour définir ce qu'il faut entendre par réseau couplé L^n , soit :

L_i : le i -ème réseau correspondant à l'espèce de type i .

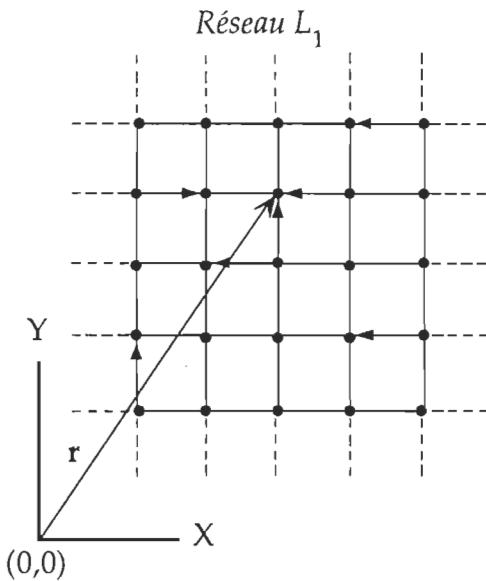


Fig 1.12 État du réseau réservé à l'espèce I

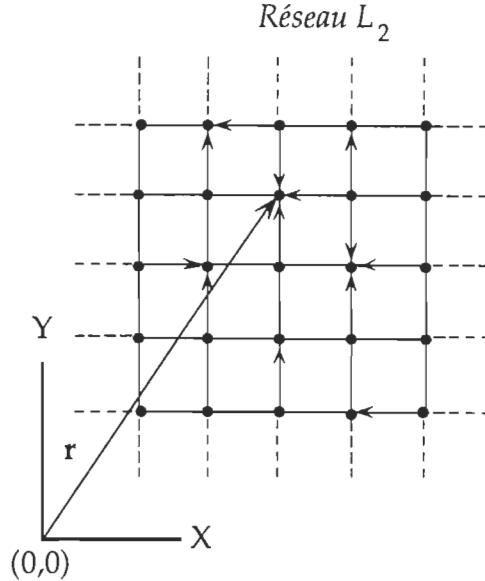


Fig 1.13 État du réseau réservé à l'espèce II

À présent, si on suppose que tous ces réseaux ont une origine $(0,0)$ commune, alors la *superposition* de ces n réseaux, origine à origine, conduit à reconstruire le réseau caractérisant la population entière. On dit alors que le réseau L^n est un *réseau n-couplé*^{1,4,7,10}. C'est ce modèle qui sera utilisé au long de ce travail. Il y a plusieurs avantages à décomposer ainsi le réseau complet en réseaux partiels. Illustrons ce que serait un réseau couplé dans le cas où l'on aurait deux espèces; une de type I, l'autre de type II. Dans ce cas, L^2 serait le résultat de la superposition en étage du réseau réservé à l'espèce de type I et celui réservé à l'espèce de type II. La figure 1.14 illustre comment le 2-couplage doit-être perçu. La superposition de ces deux réseaux doit être faite noeud à noeud, *et seulement noeud à noeud*. C'est ce que veut signifier le trait vertical de la figure 1.14. Si on fait *l'union* des états de ces *cellules* il est clair que l'on retrouve bel et bien la configuration initiale de la *cellule* avant cette mise en étage. Par ailleurs, il est aussi clair qu'en terme de quantification, la valeur correspondant à l'état de la *cellule* avant la mise en étage est le résultat de la somme des valeurs des états mis en étage. De plus, en utilisant ce modèle de réseaux couplés, il est clair qu'un noeud r du réseau L_i correspond au même noeud r du réseau avant le couplage. Ceci est dû à la similitude des origines pour chaque réseau L_i .

La figure 1.14 illustre aussi ce qu'il faut exactement entendre par l'appellation « conditions périodiques aux frontières ». Sur cette figure, une seule boucle a été illustrée. En réalité, les extrémités de chaque ligne pointillée doivent être reliées entre elles, de sorte qu'un être puisse franchir une frontière et se retrouver sur la frontière parallèle à celle qu'il traverse. Cette hypothèse de périodicité nous permettra de raisonner sur

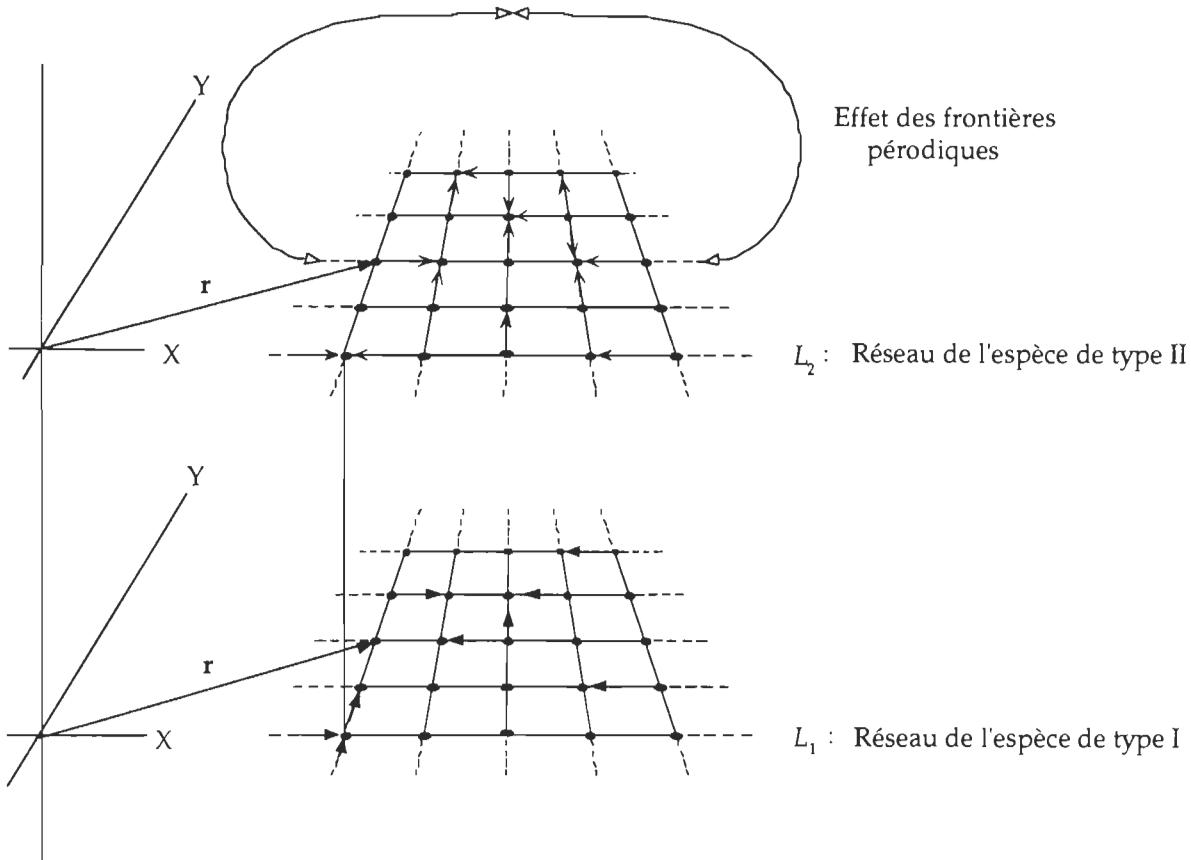


Fig 1.14 Mise en étages des réseaux 1 et 2 origine à origine

une cellule quelconque qu'elle soit sur la frontière ou non. Dans ce qui suit, cette hypothèse de périodicité aux frontières est essentielle puisque nous chercherons à construire la fonction de probabilité du nombre d'êtres présents dans une *cellule* que cette cellule soit choisie sur une frontière ou non.

1.5 DISTRIBUTION STATISTIQUE SUR LE RÉSEAU COUPLED

Fixons, sans perte de généralité, une cellule localisée en r sur un réseau L_i . À présent, définissons la variable aléatoire suivante :

$$\alpha_i^k(t) : \text{le nombre d'êtres de type } i \text{ occupant l'arête de direction } k \text{ à une cellule } r \text{ au temps } t \quad (1.1)$$

Le principe d'exclusion implique que la suite $\{\alpha_i^k(t)\}_{t \in \mathbb{N}}$ ne peut valoir que 0 ou 1. Plus précisément $\alpha_i^k(t) = \{0, 1\} \forall t \in \mathbb{N}, \forall k \in \{0, 1, 2, 3\}$. Pour bien percevoir qu'il s'agit d'un processus aléatoire voyons le temps comme le lieu de réalisation d'un essai. Considérons la figure 1.15 comme une photo prise au temps $t - 1$. Regardons la *cellule* localisée en r et posons-nous la question suivante : « quelle est la probabilité qu'il y apparaisse un être dans la

direction 3 au temps t » ? Il est bon de remarquer que la probabilité cherchée est directement reliée à la valeur de la densité de l'espèce i au temps t . Pour répondre à ce type de question nous allons émettre l'hypothèse que *chacune de ces cellules ignore l'état de toutes les cellules avoisinantes et n'en dépend pas*. L'interprétation physique de cette hypothèse est la suivante : un être de type i ne peut être influencé par la proximité d'un être de type j . Plus concrètement, cette hypothèse signifie, par exemple, qu'un prédateur ne peut être influencé par la proximité d'une proie, et par extension, la chasse de cette proie par une poursuite orientée ne lui est pas permise. Cette hypothèse d'indépendance entre les cellules se traduit aussi par l'énoncé suivant : chacun des êtres ne peut utiliser ses sens pour connaître l'état d'autres cellules du réseau. Quoi qu'il en soit, nous allons travailler tout de même en accord avec l'hypothèse selon laquelle chaque cellule est mutuellement indépendante des autres. Sous cette hypothèse on peut apporter une réponse à la question posée. Soit :

$$p_i^k(t) : \text{la probabilité qu'un être de l'espèce } i \text{ occupe l'arête } k \text{ en 1 essai.} \quad (1.2)$$

Un « essai » s'effectue durant le laps de temps $\Delta t = t - (t - \Delta t)$ suite à quoi une nouvelle propagation se réalise. Sans perte de généralité nous allons supposer Δt unitaire, et nous prendrons $\Delta t = 1$. Dans ce contexte il est clair que la variable aléatoire $\alpha_i^k(t)$ est distribuée selon une loi de Bernoulli c'est-à-dire : $\alpha_i^k(t) \sim \beta(1, p_i^k(t))$. À présent, nous sommes en mesure d'indiquer quel rôle joue l'opérateur rotation \mathbf{R} . Cet opérateur a pour objet de rendre le milieu isotrope. Dans un contexte où plusieurs espèces évoluent sur un réseau, la notion d'isotropie^{1,2} signifie en fait qu'aucune direction n'est privilégiée lorsque les espèces ont à se déplacer sur ce réseau. D'un point de vue probabiliste cette isotropie signifie qu'étant donné un certain nombre d'êtres dans une cellule \mathbf{r} , toutes les configurations possibles de ce nombre d'êtres sont équiprobables^{1,2,4,7}. De plus, lorsque l'hypothèse d'isotropie est respectée l'indice k de $p_i^k(t)$ peut donc être abandonné. Ce contexte ayant été précisé, on écrira donc plus simplement $\alpha_i^k(t) \sim \beta(1, p_i(t))$. À présent, nous allons estimer¹ la probabilité $p_i(t)$ par la densité moyenne $\bar{p}_i(t)/m$ du réseau L_i . Pour alléger la notation, la densité moyenne $\bar{p}_i(t)$ sera plus simplement notée par $p_i(t)$. Cette convention d'écriture, concernant la densité moyenne $\bar{p}_i(t)$, sera préservée tout au long de ce mémoire. Dès lors on peut écrire que la variable aléatoire $\alpha_i^k(t)$ se distribue de la sorte : $\alpha_i^k(t) \sim \beta(1, p_i(t)/m)$. Cette variable aléatoire ne contrôle que ce qui se passe à une et une seule arête d'une cellule \mathbf{r} . Or, pour connaître en probabilité le nombre total d'êtres d'une espèce i qui sont présents à une cellule \mathbf{r} nous devons alors définir la variable aléatoire suivante :

$$\alpha_i(t) : \text{le nombre d'êtres de type } i \text{ occupant une cellule } \mathbf{r} \text{ au temps } t \quad (1.3)$$

On remarque alors que la variable aléatoire définie en (1.3) peut s'écrire en termes de la variable aléatoire définie en (1.1) de la sorte :

$$\alpha_i(t) = \sum_{k=1}^m \alpha_i^k(t-1) \quad (1.4)$$

L'égalité proposée en (1.4) demande quelques explications. En effet, il peut paraître surprenant que la variable aléatoire du membre de droite correspond au temps $t-1$, tandis que la variable aléatoire du membre de gauche correspond au temps t . En voici l'explication. Pour gérer l'état de la cellule r au temps t , seuls les états des 4 cellules voisines importent. De plus, l'état de ces 4 cellules voisines à la cellule r , doit être observé au temps $t-1$. En observant ces cellules à ce temps on remarque que chacune d'elles ne contient qu'une et une seule direction qui soit susceptible de modifier l'état de la cellule r au temps t . Donc au total seulement 4 directions principales peuvent influencer l'état de la cellule r lors du passage du temps $t-1$ au temps t . C'est en ce sens que la valeur de la variable aléatoire $\alpha_i(t)$ est le résultat d'une somme sur l'indice k de la variable $\alpha_i^k(t)$; cet indice k représente les directions susceptibles de modifier l'état de la cellule r au temps t . L'ensemble des propos dont nous venons de discuter sont illustrés sur la figure 1.15. En dernier lieu, rappelons que seul l'opérateur propagation est responsable de la modification de l'état d'une cellule r entre les temps $t-1$ et t .

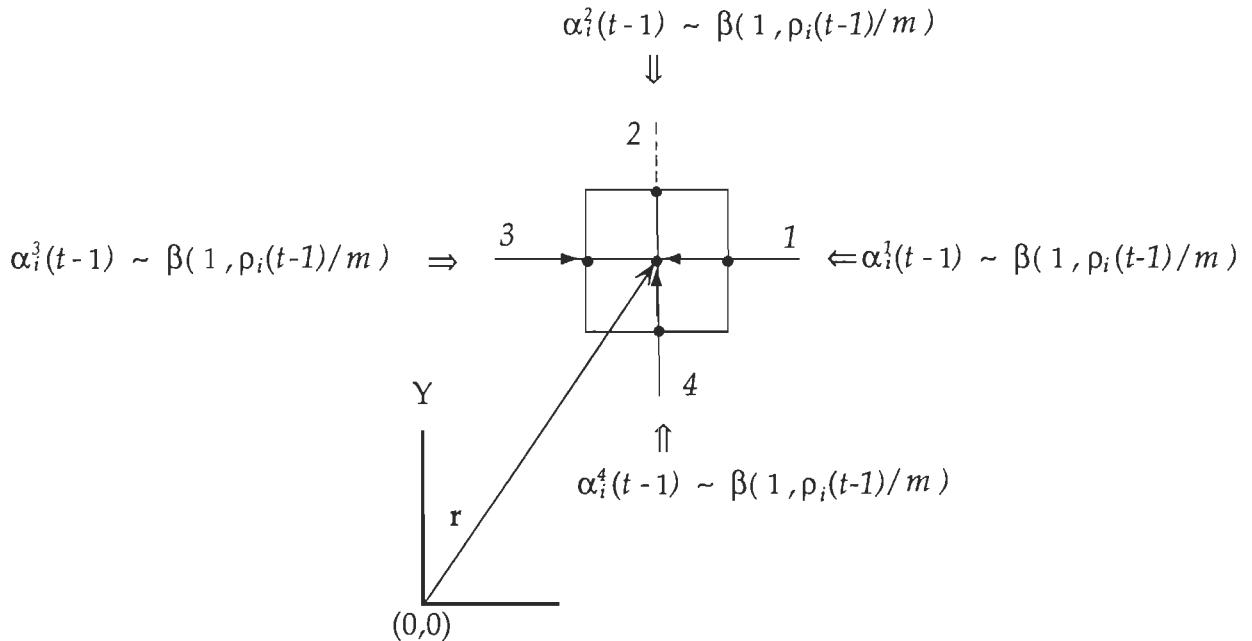


Fig 1.15 Cellule r reliée aux 4 directions principales des cellules voisines.

Seules les directions pointées jouent un rôle dans le changement d'état de la cellule r .

En ce qui nous concerne, il est clair que l'occupation de l'arête k ne dépend pas de l'occupation des autres arêtes puisque toutes les cellules en cause s'ignorent mutuellement. De ces considérations et du résultat (1.4) on obtient alors :

$$\alpha_i(t) \sim \beta(m, \rho_i(t)/m) \quad (1.5)$$

La distribution proposée en (1.5) est une binomiale. On a donc :

$$P(\alpha_i(t) = \alpha) = \binom{m}{\alpha} \left(\frac{\rho_i}{m}\right)^\alpha \left(1 - \frac{\rho_i}{m}\right)^{m-\alpha} \text{ avec } \alpha \in S_i = \{0, 1, 2, \dots, m\} \quad (1.6)$$

Dans l'équation (1.6), la variable aléatoire $\alpha_i(t)$ traduit, en termes probabilistes, le nombre total d'êtres se trouvant à la *cellule r* au temps t . Nous avons déduit la distribution du nombre d'êtres de l'espèce i pouvant se retrouver dans une *cellule r* au temps t . À présent, si on imagine que les n espèces n'interagissent pas mutuellement dans cette *cellule r* alors la conséquence en est la suivante : $\forall i \in \{1, 2, \dots, n\}$ les variables aléatoires $\alpha_i(t)$ sont donc aussi indépendantes et identiquement distribuées. Sous ces hypothèses d'indépendance mutuelle on obtient^{1,2,4,7,8} :

$$P(\alpha(t) = \alpha) = \prod_{i=1}^n \binom{m}{\alpha_i} \left(\frac{\rho_i}{m}\right)^{\alpha_i} \left(1 - \frac{\rho_i}{m}\right)^{m-\alpha_i} \text{ avec } \alpha \in S^n = S_1 \times S_2 \times \dots \times S_n \quad (1.7)$$

À ce stade, nous avons donc réussi à déduire à l'échelle microscopique du réseau cartésien, la distribution du nombre total d'êtres à un noeud r du réseau L^n sous l'hypothèse que les collisions entre ces n espèces sont purement élastiques. Ceci signifie qu'en principe la distribution (1.7) n'est exacte que sous les opérateurs matriciels P et R que constituent respectivement la *propagation* et la *rotation* à une *cellule r* du réseau L^n . Plus explicitement, si on appelle α l'état inconnu d'une *cellule r* sur le réseau L^n au temps t et qu'on fait agir les opérateurs P et R successivement sur chacun des L_i réseaux alors la distribution statistique de l'état α d'une *cellule r* est donnée par (1.7). L'introduction de l'opérateur collision C ne modifiera pas de façon appréciable (1.7) en autant que les événements « collision » seront rares^{1,2,4,7,9,11,17}.

Avant d'aborder la prochaine section, il faut noter que la suite d'opérations $R \circ P$ s'est effectuée dans un intervalle de temps $\Delta t = h = 1$. Ainsi, à travers notre traitement, nous avons choisi un pas de temps h unitaire. Il est clair qu'un choix quelconque de ce h n'affecterait en rien la façon dont nous avons décrit le changement d'état d'une cellule puisque ce pas de temps h est discret. Nous verrons au chapitre III que cette valeur de « h » porte le nom de *facteur d'échelle de temps* et que c'est ce facteur h qui nous permettra de rencontrer l'hypothèse des « événements rares ».

1.6 OBTENTION DE LA VARIATION DE LA DENSITÉ MOYENNE D'UNE ESPÈCE

Le but de cette section est d'établir une équation qui puisse donner une mesure du changement de la densité moyenne $\rho_i(t)$ entre le temps t et le temps $t+h$. Pour parvenir à obtenir ce résultat nous nous devons de faire ce court préambule.

Soit la variable aléatoire $X(t)$ où $t \in \mathbb{R}$. La suite $\{X(t)\}_{t \in \mathbb{R}}$ constitue un *processus stochastique*.

Une classe importante de processus stochastique est celle des processus Markoviens^{22,24}.

La suite $\{X(t)\}_{t \in \mathbb{R}}$ est dit Markovienne \Leftrightarrow

(1.8)

$$P(X(t) = x_t \mid X(t-1) = x_{t-1}, X(t-2) = x_{t-2}, \dots, X(0) = x_0) = P(X(t) = x_t \mid X(t-1) = x_{t-1})$$

En langage simple le processus est Markovien si la distribution future ne dépend que du présent et non du passé. On dit aussi qu'il s'agit de *phénomènes évolutifs sans mémoire*. Le mouvement Brownien est un exemple simple de processus Markovien. Pareillement on peut montrer²⁴ que si $X(t) \sim \beta(m, p_i(t))$ alors la suite des $X(t)$ est Markovienne. D'autre part, l'hypothèse d'un processus Markovien permet l'utilisation du théorème de Chapman-Kolmogorov qui s'énonce ainsi :

$$\Delta E(\rho(t)) = \sum_{\alpha \in S} \sum_{\beta \in S} (\beta - \alpha) P_{\alpha, \beta} P(\alpha(t) = \alpha_t) \quad (1.9)$$

Le résultat (1.9) donne la mesure ΔE de la variation de la valeur moyenne de la densité $\rho(t)$ si une seule espèce est en cause. Le résultat (1.9) se généralise au cas où n espèces sont en cause. Dans le cas à n espèces le théorème de Chapman-Kolmogorov prend la forme suivante¹ :

$$\Delta E(\rho_i(t)) = \sum_{\alpha \in S^n} \sum_{\beta \in S^n} (\beta_i - \alpha_i) P_{\alpha \rightarrow \beta} P(\alpha(t) = \alpha) \quad (1.10)$$

En adoptant la convention $\rho_i(t)$ pour décrire la densité moyenne de l'espèce i (1.10) devient^{1,4} :

$$\rho_i(t+h) - \rho_i(t) = \sum_{\alpha \in S^n} \sum_{\beta \in S^n} (\beta_i - \alpha_i) P_{\alpha \rightarrow \beta} P(\alpha(t) = \alpha) \quad (1.11)$$

En vertu des résultats (1.7) et (1.11) on obtient^{1,7-11} alors :

$$\rho_i(t+h) - \rho_i(t) = \sum_{\alpha \in S^n} \sum_{\beta \in S^n} (\beta_i - \alpha_i) P_{\alpha \rightarrow \beta} \prod_{i=1}^n \binom{m}{\alpha_i} \left(\frac{\rho_i}{m}\right)^{\alpha_i} \left(1 - \frac{\rho_i}{m}\right)^{m-\alpha_i} \quad (1.12)$$

L'expression (1.12) donne le résultat désiré. En effet, (1.12) permet de calculer la variation de la densité moyenne associée à l'espèce de type i en fonction d'un pas de temps h arbitraire. Physiquement le terme $\Delta E(\rho_i(t))$ fait possiblement appel au concept de collision puisque si $\Delta E(\rho_i(t)) \neq 0$ c'est possiblement que des êtres sont apparus ou disparus. C'est donc à travers le développement de ce terme que nous pourrons expliciter de quelle manière les collisions devront s'effectuer si on entend respecter une loi d'évolution donnée. Dans le traitement subséquent les mots « collision » et « réaction » seront pris dans un sens équivalent.

En résumé, nous avons décrit les règles d'évolution **R**, **P** et **C** qui seront utilisées pour créer une dynamique sur un réseau cartésien. Par la suite nous avons décrit comment sera quantifiée l'information concernant l'état de chacune des cellules de ce réseau. En dernier lieu, nous avons su donner, en termes probabilistes, une mesure du changement de la densité moyenne ρ_i d'une espèce i entre chaque pas de temps h .

Pour dégager le contenu du chapitre II, voici point par point les thèmes qui seront présentés lors de ce chapitre :

- i) Nous allons présenter la façon de construire les diagrammes servant à modéliser les interactions entre plusieurs espèces. À ce propos 3 types de d'interactions seront présentées : celles dites à action *réversible*, celles dites à action *consécutive* et finalement celles dites à action *parallèle*.
- ii) Nous verrons aussi comment la loi d'action de masse permet de poser la dynamique d'évolution de la densité moyenne d'êtres présents sur le réseau au temps t . Plus particulièrement nous verrons que cette loi de masse nous permet d'expliciter le terme $\Delta E(\rho_i(t))$ par un polynôme à plusieurs variables.

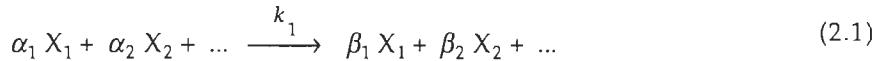
Chapitre II

MODÈLES CLASSIQUES DE CROISSANCE ET LOI D'ACTION DE MASSE

L'objet de ce chapitre est de montrer qu'il y a isomorphisme entre les diagrammes réactionnels utilisés dans le cadre de la chimie et les modèles classiques de croissance en dynamique des populations. Conséquemment, une correspondance est établie entre la loi d'action de masse et les probabilités de changement d'état d'une cellule du réseau cartésien. À ce titre, une proposition principale est dégagée qui permet la construction d'un automate cellulaire.

2.1 LOI D'ACTION DE MASSE

L'approche qui sera prise pour définir les mécanismes d'interaction de plusieurs espèces sur un réseau cartésien est très intimement liée aux diagrammes servant à modéliser une réaction chimique. Nous allons tout d'abord rappeler comment une réaction chimique se modélise. Le diagramme suivant sera assez général² pour poursuivre le travail entrepris.



Dans ce contexte rappelons que :

k_1 : est la constante de transformation lorsque $\alpha_i X_i$ se transforme en $\beta_i X_i$.

α_i : est la quantité de la substance X_i nécessaire pour que la réaction ait lieu.

β_i : est la quantité de la substance X_i résultant de la réaction.

Notons que ce diagramme indique que la réaction ne s'effectue que dans un seul sens. Pour caractériser le taux de variation associé à chaque substance il faut faire appel à la loi de Van Hoff²² :

$$\dot{E}(n_i) = \Omega k_1 (\beta_i - \alpha_i) \prod_{k=1}^j \left(E\left(\frac{n_k}{\Omega}\right) \right)^{\alpha_k} \quad (2.2)$$

où :

n_i : est le nombre total de particules de la substance X_i au temps t .

Ω : est le volume dans lequel la réaction a lieu.

Cette loi porte aussi le nom de loi d'action de masse. C'est précisément par elle que l'on pourra caractériser la dynamique d'un mécanisme de réaction. Pour une espèce chimique fixée, voyons à présent comment cette loi établit un lien entre un polynôme à plusieurs variables et la dérivée temporelle de la densité moyenne de cette espèce. Avant d'illustrer comment ceci se présente via quelques exemples pertinents à ce travail, nous allons auparavant montrer comment cette loi nous permet d'exprimer, en termes de densité, l'évolution d'une espèce chimique donnée. À propos de (2.2), notons que si le terme Ω est une constante et ne dépend pas de « t » (se sera le cas sur le réseau) alors on a :

$$\frac{\dot{E}(n_i)}{\Omega} = \frac{1}{\Omega} \frac{d}{dt} E(n_i) = \frac{d}{dt} \left(\frac{1}{\Omega} E(n_i) \right) = \frac{d}{dt} \left(E\left(\frac{n_i}{\Omega}\right) \right) = \frac{d}{dt} (E(p_i)) = \dot{E}(p_i) \quad (2.3)$$

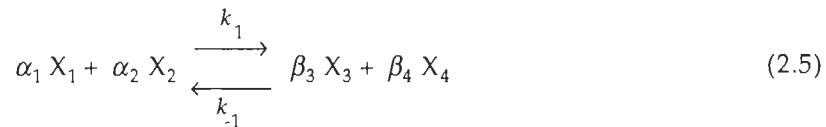
De plus, pour être conséquent avec la convention utilisé, nous remplaçons $\dot{E}(\rho_i)$ par $\dot{\rho}_i$ en sous entendant, bien entendu, que la valeur $\dot{\rho}_i$ signifie encore la dérivée de la valeur moyenne de la densité. Comme nous l'avons fait précédemment, on remplacera aussi $E(\rho_k)$ par ρ_k en sous entendant toujours que ρ_k signifie en fait la valeur moyenne de la densité. De ces remarques et du résultat (2.3), la loi d'action de masse (2.2) prend donc la forme suivante²² :

$$\dot{\rho}_i = k_1 (\beta_i - \alpha_i) \prod_{k=1}^j (\rho_k)^{\alpha_k} \quad (2.4)$$

Sous la forme de (2.4), la loi d'action de masse nous permet donc d'établir le bilan des variations moyennes des densités associées à l'espèce i . De façons différentes mais équivalentes, il est intéressant de noter que les équations (2.4) et (1.10) établissent le même bilan. Nous reviendrons sur ce lien qui existe entre les équations (2.4) et (1.10).

2.2 TYPES DE MÉCANISMES RÉACTIONNELS

Avant de présenter la forme vectorielle de cette loi d'action de masse voyons comment s'utilise l'équation (2.4). Soit la réaction :



Ce diagramme à double flèches caractérise une réaction dite *réversible*. Si l'on s'intéresse, par exemple, à l'évolution de la densité $\rho_1(t)$ de la substance X_1 alors le diagramme (2.5) conduit, selon (2.4), aux deux équations différentielles suivantes :

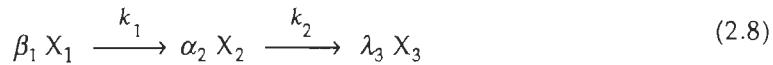
$$\frac{d\rho_1^-}{dt} = -k_1 \alpha_1 \rho_1^{\alpha_1} \rho_2^{\alpha_2} \quad \frac{d\rho_1^+}{dt} = k_{-1} \alpha_1 \rho_3^{\beta_3} \rho_4^{\beta_4} \quad (2.6)$$

Comme $\frac{d\rho_1}{dt} = \frac{d\rho_1^-}{dt} + \frac{d\rho_1^+}{dt}$ le taux de changement net de la densité $\rho_1(t)$ est donc:

$$\frac{d\rho_1}{dt} = -k_1 \alpha_1 \rho_1^{\alpha_1} \rho_2^{\alpha_2} + k_{-1} \alpha_1 \rho_3^{\beta_3} \rho_4^{\beta_4} \quad (2.7)$$

Cette façon de déduire les équations différentielles, donnant un regard macroscopique sur l'évolution d'une substance quelconque est connue depuis longtemps des chimistes¹⁻¹¹. Dans ce qui

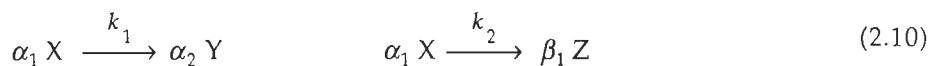
suit, nous allons montrer qu'il existe un isomorphisme entre les mécanismes de réaction chimique et les mécanismes de compétition entre deux ou plusieurs espèces vivantes. Avant d'établir ce lien voyons d'autres types de mécanismes auxquels nous pouvons être aussi confrontés dans le cadre d'une réaction chimique. Dans cette optique, on peut très bien concevoir que la création d'une certaine substance soit le résultat d'une *série de réactions consécutives*. Sous une version simplifiée, le diagramme type pour cette réaction est le suivant :



Ce diagramme conduit au système d'équations suivant :

$$\begin{aligned} \frac{d\rho_1}{dt} &= -k_1 \beta_1 \rho_1^{\beta_1} \\ \frac{d\rho_2}{dt} &= \frac{d\rho_2^+}{dt} + \frac{d\rho_2^-}{dt} = k_1 \beta_1 \rho_1^{\beta_1} - k_2 \alpha_2 \rho_2^{\alpha_2} \\ \frac{d\rho_3}{dt} &= -\frac{d\rho_2^-}{dt} = -(-k_2 \alpha_2 \rho_2^{\alpha_2}) = k_2 \alpha_2 \rho_2^{\alpha_2} \end{aligned} \quad (2.9)$$

Insistons sur le fait que le diagramme (2.8) n'a été écrit que pour permettre d'en déduire les équations correspondantes. Une telle réaction chimique n'est toutefois pas réaliste. En effet, il est difficile de croire qu'une substance X se transforme elle-même en une substance Y sans interagir avec une autre substance sinon qu'elle même. En tout dernier lieu voyons un dernier type d'interaction, soit celle où la création d'une substance est le résultat de plusieurs réactions qui s'effectuent *simultanément ou de façon parallèle*. Le diagramme type d'une telle situation est le suivant :



De ce diagramme (2.10) on peut construire la loi d'évolution pour chaque substance. En effet on a :

$$\begin{aligned} \frac{d\rho_X}{dt} &= -k_1 \alpha_1 \rho_X^{\alpha_1} - k_2 \alpha_1 \rho_X^{\alpha_1} \\ \frac{d\rho_Y}{dt} &= k_1 \alpha_1 \rho_X^{\alpha_1} \\ \frac{d\rho_Z}{dt} &= k_2 \alpha_1 \rho_X^{\alpha_1} \end{aligned} \quad (2.11)$$

Pour compléter cette discussion il faut souligner qu'une réaction chimique peut facilement faire intervenir en même temps les trois mécanismes mentionnés, soit ceux de *réversibilité*, de *réactions consécutives* et de *réactions parallèles*. Des mécanismes encore plus complexes sont possibles.

Pour cet ouvrage nous n'aurons toutefois pas à y référer. Voyons à présent comment utiliser ces mécanismes d'évolution dans le cadre de la dynamique des populations.

2.3 CONSTRUCTION DES MODÈLES VIA LA LOI D'ACTION DE MASSE

Il s'agit ici de montrer comment obtenir les modèles classiques, utilisés pour décrire l'évolution de plusieurs espèces en compétition, à partir des mécanismes de base présentés. Historiquement trois modèles de base ont été proposés^{31,32}. Se sont les suivants : le modèle de *Malthus*, le modèle *logistique* et le modèle de *Lotka Volterra*. Par ordre chronologique, voyons comment la loi d'action de masse s'utilise pour chacun de ces modèles.

2.3.1 CONSTRUCTION DES MODÈLES CLASSIQUES À 1 ESPÈCE

Dans le contexte de la dynamique à 1 espèce deux modèles simples ont été proposés. Se sont les suivants : le modèle de *Malthus* et le modèle *logistique*. Par ordre chronologique, voyons comment chacun de ces modèles est caractérisé en termes de diagramme réactionnel.

Considérons le mécanisme :



De (2.4) on obtient alors :

$$\frac{dp_1}{dt} = k_1 p_1 p_A \quad (2.13)$$

Si on suppose la ressource A comme inépuisable, alors sa variation dans le temps est à toutes fins pratiques nulle. Dans ce cas : $p_A \approx \text{constante}$. Sous cette hypothèse, et posant $a_1 = k_1 p_A$, on obtient de (2.13) :

$$\frac{dp_1}{dt} = a_1 p_1 \quad (2.14)$$

En dynamique des populations cette équation constitue le modèle de Malthus. La forme classique de cette loi fait intervenir le *nombre* d'êtres présents dans le système au temps t . Toutefois, comme la densité et le nombre d'êtres sont égaux à une constante près, nous nous exprimerons tout au long de ce travail, en termes de densité de population plutôt qu'en termes de nombre d'êtres.

Considérons le mécanisme :



De (2.4) on obtient alors :

$$\frac{d\rho_1}{dt} = k_1 \rho_1 \rho_A - k_{-1} \rho_1^2 \quad (2.16)$$

En considérant l'hypothèse $\rho_A \approx \text{constante}$ et posant $k_1 \rho_A \approx a_1 = k_1 \rho_A$, et $a_2 = k_{-1}$ on obtient de (2.16) :

$$\frac{d\rho_1}{dt} = a_1 \rho_1 - a_2 \rho_1^2 \quad (2.17)$$

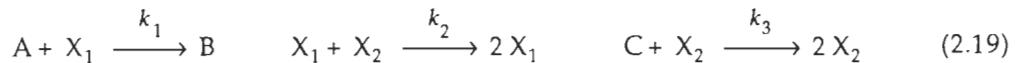
En dynamique des populations, cette équation porte le nom d'équation logistique. La forme standard de l'équation *logistique* se présente comme :

$$\frac{d\rho_1}{dt} = k \rho_1 (m - \rho_1) \quad (2.18)$$

Il est clair que (2.18) s'obtient aisément de (2.17). Toutefois, nous préférerons l'écriture proposée par (2.17) puisqu'elle fait intervenir un polynôme explicite en ρ_1 .

2.3.2 CONSTRUCTION DES MODÈLES CLASSIQUES À 2 ESPÈCES

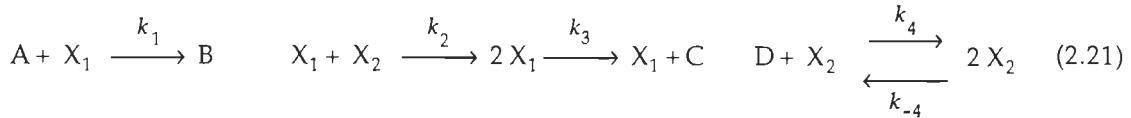
À présent, nous allons voir comment s'utilise (2.4) dans un contexte où une proie et un prédateur s'affrontent dans un combat pour la survie. Le modèle le plus simple est très certainement le modèle de *Lotka Volterra classique*. Dans ce modèle, seuls les concepts de reproduction et de nutrition sont incorporés. On peut étendre le modèle de *Lotka Volterra classique* à d'autres modèles plus sophistiqués qui font intervenir plus de concepts que les deux mentionnés précédemment. En ce qui nous concerne, nous allons ajouter à ce modèle un terme de compétition entre les êtres d'une même espèce. Par la suite, nous désignerons ce modèle par l'appellation : modèle de *Lotka Volterra étendu*. Notons que plusieurs autres variantes^{1-3,31,32} peuvent être construites à partir du modèle de *Lotka Volterra classique*. Toutefois, nous n'y référerons pas dans le cadre de ce travail. À présent, voyons comment ces deux modèles sont caractérisés en termes de diagramme réactionnel. Considérons les mécanismes de formation simultanés suivants :



En considérant ρ_A et ρ_C comme des constantes et en renommant les termes constants on obtient de (2.4) le système d'équations du modèle de *Lotka Volterra classique*^{31,32} :

$$\begin{aligned} \frac{d\rho_1}{dt} &= -a_{1,0} \rho_1 + a_{1,1} \rho_1 \rho_2 \\ \frac{d\rho_2}{dt} &= b_{0,1} \rho_2 - b_{1,1} \rho_1 \rho_2 \end{aligned} \quad (2.20)$$

Tel que nous l'avons déjà indiqué, on peut raffiner le modèle de *Lotka Volterra classique* en faisant intervenir un terme de compétition entre les êtres d'une même espèce. Pour obtenir ce terme on modifie le mécanisme (2.19) ainsi :



En considérant ρ_A , ρ_B , ρ_C et ρ_D comme des constantes, et en renommant les termes constants on obtient, par l'utilisation de (2.4) conjointement à (2.21), le système d'équations de *Lotka Volterra* avec l'ajout des termes de compétition intraspécifique. Il réfère au système d'équations différentielles suivant :

$$\begin{aligned} \frac{d\rho_1}{dt} &= -a_{1,0}\rho_1 + a_{1,1}\rho_1\rho_2 - a_{2,0}\rho_1^2 \\ \frac{d\rho_2}{dt} &= b_{0,1}\rho_2 - b_{1,1}\rho_1\rho_2 - b_{0,2}\rho_2^2 \end{aligned} \quad (2.22)$$

Nous appellerons ce système d'équations : modèle de *Lotka Volterra étendu*.

2.4 DISCUSSION

Le but du présent traitement était d'illustrer comment la loi de Van Hoff ou loi d'action de masse (2.4) permet de rattraper les mécanismes de croissance de population. Qu'il s'agisse de la croissance de NaCl ou de la croissance de bactéries nous constatons que la loi d'action de masse fait ressortir que « NaCl » et « bactéries » sont des étiquettes qui se rapportent aux mêmes équations différentielles¹⁵. À cet égard, nous avons démontré qu'il existe un isomorphisme entre « réaction chimique » et « dynamique de population ».

Puisque nous entendons faire évoluer diverses espèces sur un réseau cartésien, il est temps d'apporter quelques restrictions. D'une part, parce que le temps d'évolution sera caractérisé par un pas de temps discret h , et d'autre part parce que l'ordre m du réseau utilisé impose une limite quant au nombre d'êtres présents à une cellule. Voyons respectivement quelles sont les conséquences de ces deux remarques. La loi d'action de masse (2.4) fait intervenir la variable $\dot{\rho}_i$. Implicitement (2.4) considère donc la densité $\rho_i(t)$ comme une fonction continue. Or, puisque nous entendons faire évoluer les espèces à pas de temps h nous nous devons d'adapter (2.4) à cette circonstance. Pour un h suffisamment petit, une version discrète de (2.4) peut être obtenue de la sorte :

$$\dot{\rho}_i \approx [\rho_i(t+h) - \rho_i(t)] / h \quad (2.23)$$

L'utilisation de (2.23) permet alors d'adapter (2.4) à nos besoins. En effet, on peut alors écrire (2.4) comme :

$$[\rho_i(t+h) - \rho_i(t)]/h = k_1 (\beta_i - \alpha_i) \prod_{k=1}^j (\rho_k)^{\alpha_k} \quad (2.24)$$

Examinons l'expression (2.24) de plus près. À travers les exemples précédents nous avons pu remarquer que le membre de droite coïncide toujours avec un polynôme à plusieurs variables en ρ_k , $1 \leq k \leq j$ où j est le nombre d'espèces interagissant avec l'espèce i . En particulier, si une seule espèce évoluait dans le temps, la loi d'action de masse (2.4) conduirait à lier $\dot{\rho}$ au polynôme suivant :

$$\dot{\rho} = \sum_{l=0}^m a_l \rho^l \quad (2.25)$$

Si n espèces interagissent, via plusieurs mécanismes réactionnels, l'utilisation de la loi d'action de masse (2.4) conduit en général à un système de n équations différentielles dont la i -ème serait décrite par (annexe A) :

$$\dot{\rho}_i = \sum_{l=0}^m a_l^{(i)} \prod_{j=1}^n (\rho_j)^{l_j} \quad (2.26)$$

Ainsi, la version discrète de (2.26) s'obtient en

considérant (2.23). On obtient alors :

$$\rho_i(t+h) - \rho_i(t) = h \sum_{l=0}^m a_l^{(i)} \prod_{j=1}^n (\rho_j)^{l_j} \quad (2.27)$$

Le résultat (2.27) est équivalent à la loi d'action de masse présentée sous la forme (2.24). Toutefois, puisque (2.27) présente une grande régularité d'écriture, nous retiendrons la forme proposée en (2.27). À présent, indiquons quelle conséquence a l'ordre m du réseau sur (2.4) ou (2.27). Nous savons que l'ordre m du réseau fixe à m le maximum d'êtres pouvant se retrouver à une cellule. Dans cette optique il nous sera impossible d'envisager, par exemple, un mécanisme de ce type :



Ce dernier diagramme suppose que 5 unités de l'espèce X et une unité de l'espèce A devraient être présents dans une même cellule pour qu'une unité de l'espèce Y puisse apparaître à cette cellule. Or, sur un réseau cartésien, il est impossible de localiser 5 unités de l'espèce X dans une cellule sans du même coup violer le principe d'exclusion; c'est pourquoi ce type de mécanisme nous sera interdit.

Ceci étant dit l'utilisation de la loi d'action de masse via (2.4) ou (2.27) se fait sous la condition : $0 \leq \alpha_i, \beta_i \leq 4$ pour $0 \leq i \leq 4$.

2.5 FORME VECTORIELLE DE LA LOI D'ACTION DE MASSE

En général, pour réunir de façon compacte les n équations différentielles de (2.26) nous écrirons :

$$\dot{\rho} = F(\rho(t)) \quad (2.29)$$

C'est la forme vectorielle de la loi d'action de masse. Le résultat (2.29) est un cas particulier¹⁻¹¹ de la loi de *réaction-diffusion* que voici :

$$\frac{\partial}{\partial t} \rho(r, t) = F(\rho(r, t)) + D \cdot \nabla^2 \rho(r, t) \quad (2.30)$$

Selon (2.30) le vecteur densité $\rho(r, t) = \langle \rho_1(r, t), \rho_2(r, t), \dots, \rho_n(r, t) \rangle$ contient les fonctions de densités correspondant aux n espèces en jeu lors de la réaction. L'équation (2.30) fait aussi ressortir que les densités $\rho_i(r, t)$ soit une fonction spatiale qui dépend de la position où se localise l'espèce i . Dans cette optique r représente donc un point dans l'espace. D'autre part, la quantité F joue un rôle analogue à celui d'un champ vectoriel auquel sont astreintes les densités. Nous avons vu comment exhiber ce champ F lorsque l'utilisation de la loi d'action de masse (2.27) le permet. En ce sens, on doit ajouter que F n'est que la conséquence des diagrammes d'interactions qui caractérisent la compétition entre les diverses espèces. Le deuxième membre de droite de (2.30) correspond au terme de diffusion. Ce terme apparaît seulement lorsque qu'il y a dépendance spatiale en r . Dans ce cas D constitue la matrice (tenseur) des coefficients de diffusion. Il est à noter que l'évolution des différentes espèces sur un réseau force l'apparition de ce terme de diffusion. Il en est ainsi puisque la nature même d'une réaction sur un réseau dépend du lieu où se trouve les espèces formant une réaction. Pour imager cette dernière affirmation, imaginons une dynamique entre deux espèces où la première espèce occupe la moitié de gauche du réseau tandis que l'autre espèce occupe la moitié de droite du réseau. Ces deux espèces étant ainsi positionnées, il est clair que la nature de l'interaction dépendra dans ce cas de la coordonnée (x, y) du réseau. En résumé, on peut donc dire que le terme de diffusion $D \cdot \nabla^2 \rho(r, t)$ est inhérent à l'utilisation du réseau puisque ce dernier réfère à l'espace sur lequel les réactions auront lieu.

Lorsque chaque densité ne dépend pas de la position r , c'est-à-dire lorsque $\rho_i = \rho_i(t)$ alors le terme de diffusion s'annule. Cette dernière condition revient à éliminer le caractère spatial des réactions puisque seule la variable temporelle est alors responsable de la croissance des espèces. Cette condition pourrait être réalisée par l'implantation d'un et un seul automate cellulaire. Cet automate pourrait recréer à lui seul l'évolution temporelle de la réaction associée à la loi

d'action de masse, quitte à le doter de règle de collision approprié et à alimenter chacune de ces arêtes par des loi de Bernoulli (voir la figure 1.15). En fixant ainsi notre attention sur un et un seul automate ceci revient à dire que le terme de diffusion $\nabla^2 \rho(\mathbf{r}, t)$ disparaît de sorte que la dérivée partielle de (2.30) en t devient totale et cela nous conduit alors à l'équation (2.29). Ainsi, lorsque l'évolution d'une réaction possède un caractère strictement temporel alors l'équation (2.29) peut être utilisée. Dans ce contexte, l'équation (2.29) mérite quelque discussion. En géométrie différentielle cette équation porte le nom d'équation intégrale³³. Pour bien en saisir le sens il est intéressant de lui donner sa correspondance en mécanique. En effet, si l'on change $\rho(t)$ au profit de la fonction vectorielle $\mathbf{r}(t)$ donnant la position d'une particule au temps t alors la comparaison est possible. C'est cette comparaison qui est présentée à la figure 2.1. Dans l'ensemble, ces deux équations relèvent le fait que c'est la fonctionnelle F qui gère la dynamique du phénomène.

En effet, résoudre l'équation $v(t) = F(\mathbf{r}(t))$ revient à chercher quelle trajectoire suivra une particule placé dans ce champ. Dans le contexte de la compétition entre plusieurs espèces, résoudre l'équation $\dot{\rho}(t) = F(\rho(t))$ revient plutôt à se demander quel comportement aura chacune des espèces si on les dispose dans un milieu dont les mécanismes de contrôle sont caractérisés par F . Ajoutons que la théorie enseigne³¹⁻³³ que la connaissance du champ F ainsi que des conditions initiales du système confèrent à (2.29) une solution unique; preuve de quoi le comportement des espèces n'est pas assujetti aux concepts de liberté...

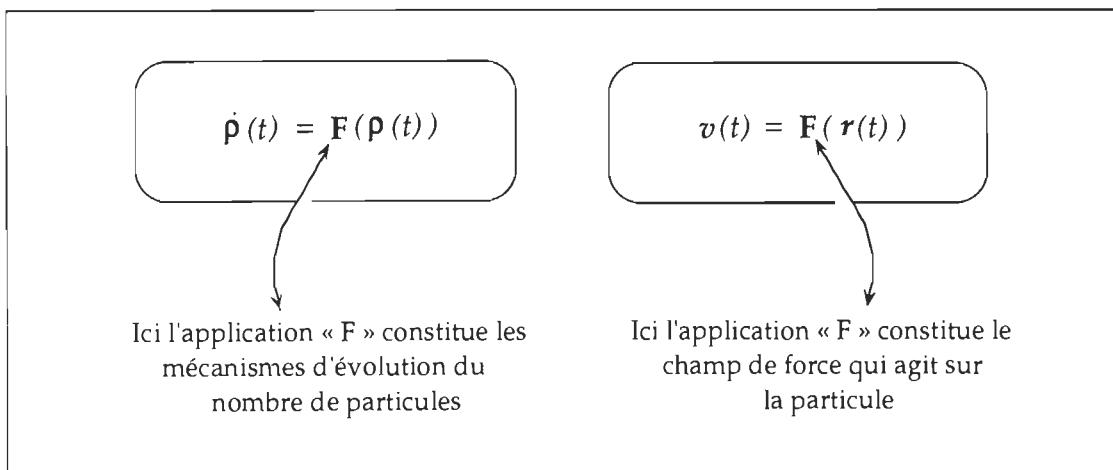


Fig 2.1 Correspondance entre le champ de force et les mécanismes d'évolutions

2.6 LOI D'ACTION DE MASSE ET THÉORÈME DE CHAPMANN-KOLMOGOROV

Tel que nous l'avons vu par les résultats (1.12) et (2.27) la mesure de $\rho_i(t+h) - \rho_i(t)$ peut s'écrire sous deux formes. D'une part (1.12) a été obtenu via le théorème de Chapman-Kolmogorov tandis que (2.27) signale qu'il est possible d'expliciter la mesure $\rho_i(t+h) - \rho_i(t)$ via l'utilisation de loi d'action de masse. Cette correspondance entre (1.12) et (2.27) permet d'écrire^{1,4,7,8} :

$$\sum_{\alpha \in S^n} \sum_{\beta \in S^n} (\beta_i - \alpha_i) P_{\alpha \rightarrow \beta} \prod_{i=1}^n \binom{m}{\alpha_i} \left(\frac{\rho_i}{m}\right)^{\alpha_i} \left(1 - \frac{\rho_i}{m}\right)^{m-\alpha_i} = h \sum_{l=0}^m a_l^{(i)} \prod_{j=1}^n (\rho_j)^{l_j} \quad (2.31)$$

La préoccupation principale que suggère (2.31) est la suivante : *il serait très intéressant de réussir à exprimer les probabilités de transitions $P_{\alpha \rightarrow \beta}$ en fonction des coefficients $a_l^{(i)}$* puisque se sont ces $a_l^{(i)}$ qui sont mesurables. Le but principal de ce mémoire est précisément de mettre en oeuvre une stratégie claire et précise visant à relier ces coefficients $a_l^{(i)}$ aux éléments de la matrice de transition $P_{\alpha \rightarrow \beta}$. Notons que se sont les probabilités de transition $P_{\alpha \rightarrow \beta}$ qui dicteront à quelle fréquence les collisions doivent s'effectuer si on entend respecter un mécanisme d'interaction fixé. Une fois ces éléments identifiés, nous pourrons alors dire que l'automate cellulaire est entièrement fondé puisque, pour une cellule donnée, la mesure de l'importance de la transition α vers β est précisément mesurée par $P_{\alpha \rightarrow \beta}$.

2.7 ÉQUATION FONDAMENTALE DES PROBABILITÉS DE TRANSITIONS

Nous venons de voir pourquoi il est essentiel de réussir à évaluer les probabilités $P_{\alpha \rightarrow \beta}$. Pour clarifier ce propos, rappelons comment les choses se passent du point de vue de la cellule. Nous raisonnons à partir d'un exemple où 2 espèces évoluent sur un réseau. Une cellule contient 2 êtres de type I et 3 êtres de type II. Selon cet exemple on aurait ici $\alpha = (2, 3)$. Dans ce contexte, si un mécanisme réactionnel est proposé, comment savoir si l'on doit modifier l'état de la cellule lorsqu'elle se trouve dans l'état $\alpha = (2, 3)$? Nous touchons là le problème consistant à définir adéquatement les règles décisionnelles associées aux collisions entre les espèces. Reprenons notre exemple. Une cellule se trouve au temps t dans l'état $\alpha = (2, 3)$. Quelle chance a-t-elle de se retrouver dans l'état $\beta = (3, 3)$? On voit donc que c'est par la valeur de $P_{\alpha \rightarrow \beta}$ que l'on pourra modifier chacune des cellules suivant l'état où elle se trouve. Du point de vue informatique, réaliser ces modifications d'états des cellules nécessite la connaissance de la valeur de probabilité $P_{\alpha \rightarrow \beta}$. La proposition (2.32) donne la réponse au problème soulevé. La démonstration détaillée de cette proposition fait l'objet de l'annexe A. L'obtention de ce résultat complète donc la présentation théorique de l'automate cellulaire comme outil pour explorer les mécanismes de croissance via la loi d'action de masse. À présent, nous pouvons apporter une réponse au problème que nous avions soulevé à la section 1.4 de ce mémoire.

Proposition

Soit la transition $\alpha \rightarrow \beta$ respectant les hypothèses qui mènent au résultat (2.31). Dans le cas où on restreint toutes les transitions à des sauts d'au maximum de 1 c'est-à-dire si $|\alpha - \beta| \leq 1$ alors les éléments de la matrice de transition $P_{\alpha \rightarrow \beta}$ sont obtenus par la résolution simultanée des n systèmes d'équations linéaires en $P_{\psi \rightarrow \xi}$ suivant :

(2.32)

$$\sum_{u=0}^2 (u_i - 1) P_{\alpha \rightarrow \alpha + u - 1} = h \sum_{v=0}^{\alpha} \prod_{j=1}^n \frac{\binom{\alpha_j}{v_j} m^{v_j}}{\binom{m}{v_j}} a_v^{(i)}$$

* Ces n systèmes d'équations sont obtenus en affectant à i toutes les valeurs distinctes pour $1 \leq i \leq n$ dans l'expression ci-dessus.

L'interrogation était la suivante : pour un mécanisme d'interaction fixé, comment reproduire sur un réseau la dynamique conforme à ce modèle ? L'interprétation que l'on doit faire à présent de cette question se traduit ainsi : comment simuler sur un réseau les modèles classiques de croissance ? Et, en tout premier lieu, sont-ils seulement reproductibles sur un réseau cartésien ? La résolution de ces problèmes constitue le second noyau de ce mémoire. Se sont ces nouveaux problèmes de « faisabilité » que nous allons à présent affronter.

Chapitre III

ANALYSE THÉORIQUE DES MODÈLES CLASSIQUES DE CROISSANCE ET LOI D'ACTION DE MASSE

Dans ce chapitre, nous vérifions s'il est possible de construire un automate cellulaire probabiliste visant à reproduire, sur un réseau, les modèles classiques de croissance. À travers cette étude de faisabilité on montre comment lever les contraintes reliées à la structure probabiliste de l'automate ainsi qu'aux contraintes reliées à l'ordre m du réseau utilisé. En particulier, nous montrons comment, d'une part, ajuster convenablement l'échelle de facteur h , et d'autre part, nous montrons comment poser des conditions de flux aux frontières de l'espace de phase S^n pour maintenir la trajectoire des densités à l'intérieur de cet espace de phase S^n .

3.1 ANALYSE DE FAISABILITÉ DES MODÈLES CLASSIQUE À 1 ESPÈCE

Dans cette section nous essayerons de voir s'il est possible de recréer les processus de croissance soutenus par le modèle de *Malthus* et par le modèle *logistique*. Cette tâche nous permettra de voir comment s'utilise la proposition (2.32) via des exemples. Nous abrégerons cette proposition par l'encadré suivant :

$$\boxed{\sum_{u=0}^2 (u_i - 1) P_{\alpha \rightarrow \alpha + u - 1} = h \sum_{v=0}^{\alpha} \prod_{j=1}^n \frac{\binom{\alpha_j}{v_j} m^{v_j}}{\binom{m}{v_j}} a_v^{(i)}} \quad (3.1)$$

3.1.1 ÉQUATIONS GÉNÉRALES DES MODÈLES À 1 ESPÈCE

Pour débuter, nous allons supposer qu'une seule espèce évolue dans un milieu. Nous avons vu, en particulier, que le modèle de *Malthus* et le modèle *logistique* entrent sous cette rubrique. Si une seule espèce évolue (3.1) se développe en posant $n = 1$, tandis que i peut seulement prendre la valeur 1. De plus, on remarque que les indices vectoriels de (3.1) deviennent simplement scalaires. En vertu de ces remarques le membre de gauche de (3.1) s'écrit comme :

$$\sum_{u=0}^2 (u_1 - 1) P_{\alpha \rightarrow \alpha + u - 1} = \sum_{u_1=0}^2 (u_1 - 1) P_{\alpha_1 \rightarrow \alpha_1 + u_1 - 1} \quad (3.2)$$

et d'autre part :

$$\sum_{v=0}^{\alpha} \prod_{j=1}^n \frac{\binom{\alpha_j}{v_j} m^{v_j}}{\binom{m}{v_j}} = \sum_{v_1=0}^{\alpha_1} \frac{\binom{\alpha_1}{v_1} m^{v_1}}{\binom{m}{v_1}} a_{v_1}^{(1)} \quad (3.3)$$

Dans ces derniers résultats nous délaisserons l'indice 1 des variables u, v et α puisqu'une seule espèce est ici considérée. Ces modifications étant apportées (3.1) devient alors :

$$\sum_{u=0}^2 (u - 1) P_{\alpha \rightarrow \alpha + u - 1} = h \sum_{v=0}^{\alpha} \frac{\binom{\alpha}{v} m^v}{\binom{m}{v}} a_v^{(1)} \quad (3.4)$$

En développant la somme de gauche de (3.4) on obtient :

$$P_{\alpha \rightarrow \alpha-1} + P_{\alpha \rightarrow \alpha+1} = h \sum_{v=0}^{\alpha} \frac{\binom{\alpha}{v} m^v}{\binom{m}{v}} a_v^{(1)} \quad (3.5)$$

Par l'égalité (3.5) on voit qu'à chaque valeur de α correspond une équation linéaire. Rappelons que α représente le nombre d'êtres contenus dans une cellule au temps t . On va donner respectivement à α les valeurs telles que : $0 \leq \alpha \leq m$ où m est l'ordre du réseau. Dans notre cas $m = 4$ puisque le réseau est cartésien. De plus, pour simplifier le développement de (3.5) nous allons poser l'échelle de temps $h = 1$. Nous reparlerons bientôt du rôle que jouera cette échelle de temps « h ». Ainsi, si α est affecté à chacune des valeurs mentionnées on obtient respectivement de (3.5) :

a)	$P_{0 \rightarrow 1} - P_{0 \rightarrow -1} = a_0^{(1)}$	(3.6)
b)	$P_{1 \rightarrow 2} - P_{1 \rightarrow 0} = a_0^{(1)} + a_1^{(1)}$	
c)	$P_{2 \rightarrow 3} - P_{2 \rightarrow 1} = a_0^{(1)} + 2 a_1^{(1)} + 8/3 a_2^{(1)}$	
d)	$P_{3 \rightarrow 4} - P_{3 \rightarrow 2} = a_0^{(1)} + 3 a_1^{(1)} + 8 a_2^{(1)} + 16 a_3^{(1)}$	
e)	$P_{4 \rightarrow 5} - P_{4 \rightarrow 3} = a_0^{(1)} + 4 a_1^{(1)} + 16 a_2^{(1)} + 64 a_3^{(1)} + 256 a_4^{(1)}$	

Il faut remarquer à propos de (3.4) que deux probabilités sont inappropriées. En effet, $P_{0 \rightarrow -1}$ et $P_{4 \rightarrow 5}$ n'ont pas de sens ici. Pour comprendre pourquoi elles apparaissent dans le développement de (3.5) on peut se référer à l'annexe A car une explication simple est impossible. Dans le cadre de cette annexe nous avons vu que : $P_{0 \rightarrow -1} = 0$ et $P_{4 \rightarrow 5} = 0$. Ceci étant dit le système d'équation (3.6) est en réalité constitué de 5 équations et 8 inconnues. On peut considérablement simplifier l'écriture de (3.6) si on utilise la variable intermédiaire κ_v suivante :

$$\kappa_v = C(v, m) a_v^{(1)} \quad \text{avec} \quad C(v, m) = m^v / \binom{m}{v} \quad (3.7)$$

Si on effectue ce changement de variables, l'équation (3.4) conduit à :

$$\sum_{u=0}^2 (u-1) P_{\alpha \rightarrow \alpha+u-1} = h \sum_{v=0}^{\alpha} \binom{\alpha}{v} \kappa_v \quad (3.8)$$

En posant $\hbar = 1$, le développement itéré de (3.8) conduit à son tour au système d'équations :

$$\begin{aligned}
 a) \quad P_{0 \rightarrow 1} - P_{0 \rightarrow -1} &= \kappa_0 \\
 b) \quad P_{1 \rightarrow 2} - P_{1 \rightarrow 0} &= \kappa_0 + \kappa_1 \\
 c) \quad P_{2 \rightarrow 3} - P_{2 \rightarrow 1} &= \kappa_0 + 2\kappa_1 + \kappa_2 \\
 d) \quad P_{3 \rightarrow 4} - P_{3 \rightarrow 2} &= \kappa_0 + 3\kappa_1 + 3\kappa_2 + \kappa_3 \\
 e) \quad P_{4 \rightarrow 5} - P_{4 \rightarrow 3} &= \kappa_0 + 4\kappa_1 + 6\kappa_2 + 4\kappa_3 + \kappa_4
 \end{aligned} \tag{3.9}$$

Il est rassurant de constater que les équations (3.9) sont les mêmes^{9,11} que celles utilisées pour simuler, sur réseau cartésien, une réaction chimique où une seule substance subit des modifications mesurables dans le temps. Dans cette optique, il faut noter que certains auteurs¹⁻¹¹ préfèrent s'exprimer en termes de la variable κ_v , quitte à revenir plus tard aux véritables coefficients d'intérêt $a_v^{(i)}$. En ce qui nous concerne, nous exprimerons les équations découlant de (3.1) en termes des coefficients $a_v^{(i)}$. Cette approche nous évitera d'avoir à effectuer deux calculs plutôt qu'un. En dernier lieu, il est remarquable que les coefficients de (3.9) soient ceux du binôme de Newton. Ce dernier changement de variable nous a donc permis de vérifier d'une part la pertinence de la proposition (2.13) puisqu'elle coïncide avec celle d'autres auteurs, et d'autre part, il aura mis en évidence la grande régularité que possèdent ces même équations puisque seul le coefficient du binôme semble nécessaire pour les générer.

3.1.2 PRÉSENTATION DES CONTRAINTES À APPORTER AUX ÉQUATIONS

Nous avons eu l'occasion de remarquer que le système d'équations (3.6) est constitué de 5 équations et de 8 inconnues. En principe, une infinité de solutions sont donc permises pour les $P_{\alpha \rightarrow \beta}$ du système d'équations (3.6). Il faut bien dire « en principe » car tel ne sera peut être pas le cas. En effet, la contrainte imposée par la structure probabiliste du système (3.6) nous force, d'une part, à respecter plusieurs conditions, et d'autre part, l'utilisation du réseau cartésien force la valeur de densité ρ_i associée à l'espèce de type i à être en tout temps compris à l'intérieur de l'intervalle $0 \leq \rho_i \leq 4$. Cette dernière inégalité relève donc le fait qu'en général l'ordre m du réseau constraint le système d'équations (3.6) via les coefficients $a_v^{(i)}$. Nous allons donc généraliser l'ensemble des conditions que l'on doit adjoindre à (3.1) lorsque l'on souhaite résoudre le système d'équations qui en découle. Un contexte général suppose que le réseau est d'ordre m et que n espèces y évoluent. Dans ce contexte, les lois de probabilités ainsi que les valeurs d'équilibre doivent satisfaire l'ensemble des inégalités de l'encadré suivant:

$$\boxed{\begin{aligned} & \forall \alpha, \beta \in S^n \\ i) \quad & P_{\alpha \rightarrow \alpha} + \sum_{\beta \in S^n} P_{\alpha \rightarrow \beta} = 1 \quad \text{avec } \alpha \neq \beta \\ ii) \quad & 0 \leq P_{\alpha \rightarrow \beta} \leq 1 \\ iii) \quad & 0 \leq \rho_i \leq m \quad \forall i, 1 \leq i \leq n \end{aligned}} \tag{3.10}$$

Ces contraintes sont incontournables et doivent être imposées en tout temps au système d'équations découlant de (3.1). L'effet des conditions que (3.10) impose à (3.1) se traduira irrémédiablement par une plus grande limitation dans nos choix des coefficients $a_v^{(i)}$. Ces restrictions sur les coefficients $a_v^{(i)}$ semblent si nombreuses qu'il est à présent permis de penser que loin d'admettre une infinité de solutions, un système d'équations déduit de (3.1) n'en admettra aucune. Et effectivement, nous allons voir que tel peut être le cas.

Le travail qui suit a pour objet d'essayer de déterminer, via des mécanismes précis, si le système d'équations (3.1) admet une solution consistante lorsqu'il est constraint par (3.10). Ce type de travail a été désigné sous le vocable « analyse de faisabilité ». C'est dans cette optique que les modèles classiques à 1 et 2 espèces ont été analysés dans ce chapitre. De ces analyses nous déduirons la démarche à utiliser pour résoudre en général le système d'équations (3.1) lorsqu'il est soumis aux contraintes de (3.10). Ces analyses a priori apparaissent donc indispensables si on espère effectuer une simulation informatique d'un modèle en particulier.

3.1.3 ANALYSE DU MODÈLE DE MALTHUS

Le résultat (2.14) confirme que le mécanisme $A + X \xrightarrow{k_1} 2X$ conduit à l'équation :

$$\frac{dp}{dt} = a_1 p \quad (3.11)$$

Si on identifie les coefficients $a_v^{(1)}$ du polynôme général avec les coefficients qui découlent de l'application de la loi d'action de masse (2.26) alors on a:

$$a_0^{(1)} = a_0 = 0, \quad a_1^{(1)} = a_1, \quad a_2^{(1)} = a_2 = 0, \quad a_3^{(1)} = a_3 = 0, \quad a_4^{(1)} = a_4 = 0$$

Le système d'équations (3.6) se particularise donc de la façon suivante :

a)	$P_{0 \rightarrow 1} - P_{0 \rightarrow -1} = 0$	(3.12)
b)	$P_{1 \rightarrow 2} - P_{1 \rightarrow 0} = a_1$	
c)	$P_{2 \rightarrow 3} - P_{2 \rightarrow 1} = 2a_1$	
d)	$P_{3 \rightarrow 4} - P_{3 \rightarrow 2} = 3a_1$	
e)	$P_{4 \rightarrow 3} = -4a_1$	

À ce stade, on est déjà en mesure de voir que l'équation e) de (3.12) est fautive. En effet, dans le modèle de croissance de *Malthus* le coefficient a_1 est supposé strictement positif. Puisque tel est le cas on a alors l'implication suivante : $a_1 > 0 \Rightarrow a_1 > P_{4 \rightarrow 3} < 0$ ce qui constitue une contradiction. Donc, le mécanisme qui nous conduit à l'équation (3.11) ne se prête pas à une simulation sur un réseau cartésien. Nous remarquons que les contraintes (3.10) apportées par la structure probabiliste de l'automate cellulaire peuvent effectivement rendre incompatible le système (3.1) pour un mécanisme d'interaction fixé. En ce qui concerne le modèle de *Malthus*, il aurait été surprenant qu'il puisse être simulé sur un réseau car la probabilité de transition $P_{4 \rightarrow 3}$ non nulle suppose l'existence d'une transition ayant pour effet de faire décroître la population, fait qui par ailleurs, est inadmissible dans ce modèle. Il n'y a toutefois pas là matière à déception si l'on constate que physiquement parlant ce modèle de croissance est irréaliste. En effet, comment imaginer qu'une population puisse indéfiniment croître sans que le milieu, dans lequel cette croissance s'effectue, le ramène à l'ordre ! Ce modèle simple nous a donc permis de voir à quel point il est important d'apporter une attention particulière aux contraintes que recèle (3.10). Passons à l'analyse du modèle *logistique*.

3.1.4 ANALYSE DU MODÈLE LOGISTIQUE

Le résultat (2.17) confirme que le mécanisme $A + X \xrightleftharpoons[k_{-1}]{k_1} 2X$ conduit à l'équation :

$$\frac{dp}{dt} = a_1 p - a_2 p^2 \quad (3.13)$$

Si on identifie les coefficients $a_v^{(1)}$ du polynôme général avec les coefficients qui découlent de l'application de la loi d'action de masse (2.26) alors on a:

$$a_0^{(1)} = a_0 = 0, \quad a_1^{(1)} = a_1, \quad a_2^{(1)} = -a_2, \quad a_3^{(1)} = a_3 = 0, \quad a_4^{(1)} = a_4 = 0$$

Le système d'équations (3.6) se particularise donc de la façon suivante :

a)	$P_{0 \rightarrow 1} = 0$
b)	$P_{1 \rightarrow 2} - P_{1 \rightarrow 0} = a_1$
c)	$P_{2 \rightarrow 3} - P_{2 \rightarrow 1} = 2a_1 - 8/3a_2$
d)	$P_{3 \rightarrow 4} - P_{3 \rightarrow 2} = 3a_1 - 8a_2$
e)	$P_{4 \rightarrow 3} = -4a_1 + 16a_2$

(3.14)

Pour ce modèle il semble y avoir une avenue. Du moins s'il y a contradiction elle est moins apparente que dans le modèle de *Malthus*. D'une part on remarque que les équations a) et e) de (3.14) sont d'ores et déjà déterminées; par conséquent elles ne laissent aucun choix. Seule les équations b) à d) de (3.14) offrent un choix. Nous avions déjà remarqué que le système d'équations (3.4) est susceptible d'être caractérisée par une infinité de solutions en autant que les $P_{\alpha \rightarrow \beta}$ satisfassent les contraintes de (3.10). Pour donner une solution unique à (3.14) il nous faut fixer 3 probabilités. Pour ce faire nous allons identifier les probabilités $P_{\alpha \rightarrow \alpha-1}$ à la somme des coefficients dont le signe est *négatif* et nous allons identifier les probabilités $P_{\alpha \rightarrow \alpha+1}$ à la somme des coefficients dont le signe est *positif*. Cette règle de correspondance fera l'objet d'une discussion plus approfondie à la prochaine section. Nous verrons pourquoi nous procéderons ainsi, en général, pour résoudre les équations constitutives de (3.1). En appliquant cette correspondance des signes aux équations b) à d) de (3.14) on réussit alors

aisément à obtenir une solution unique. Globalement, l'ensemble de ces considérations nous amènent à identifier les inconnues de la sorte :

$$\begin{aligned}
 a) \quad P_{0 \rightarrow 1} &= 0 & b) \quad P_{1 \rightarrow 2} &= a_1 & c) \quad P_{1 \rightarrow 0} &= 0 & d) \quad P_{2 \rightarrow 3} &= 2a_1 \\
 e) \quad P_{2 \rightarrow 1} &= 8/3 a_2 & f) \quad P_{3 \rightarrow 4} &= 3a_1 & g) \quad P_{3 \rightarrow 2} &= 8a_2 & h) \quad P_{4 \rightarrow 3} &= -4a_1 + 16a_2
 \end{aligned} \tag{3.15}$$

En utilisant (3.10) dans le cadre à 1 espèce, les équations *a)* à *h)* de (3.15) se retrouvent astreintes aux conditions suivantes :

$$\forall \alpha, \beta \in \{0, 1, 2, 3, 4\}$$

$$\begin{aligned}
 i) \quad P_{\alpha \rightarrow \alpha-1} + P_{\alpha \rightarrow \alpha} + P_{\alpha \rightarrow \alpha+1} &= 1 \\
 ii) \quad 0 \leq P_{\alpha \rightarrow \beta} &\leq 1 \\
 iii) \quad 0 \leq p \leq 4
 \end{aligned} \tag{3.16}$$

On peut montrer que l'application des conditions *i)* à *iii)* de (3.16) aux équations de (3.15) nous forcent à choisir les coefficients a_1 et a_2 , « en principe », dans la région ombragée de la figure 3.1.

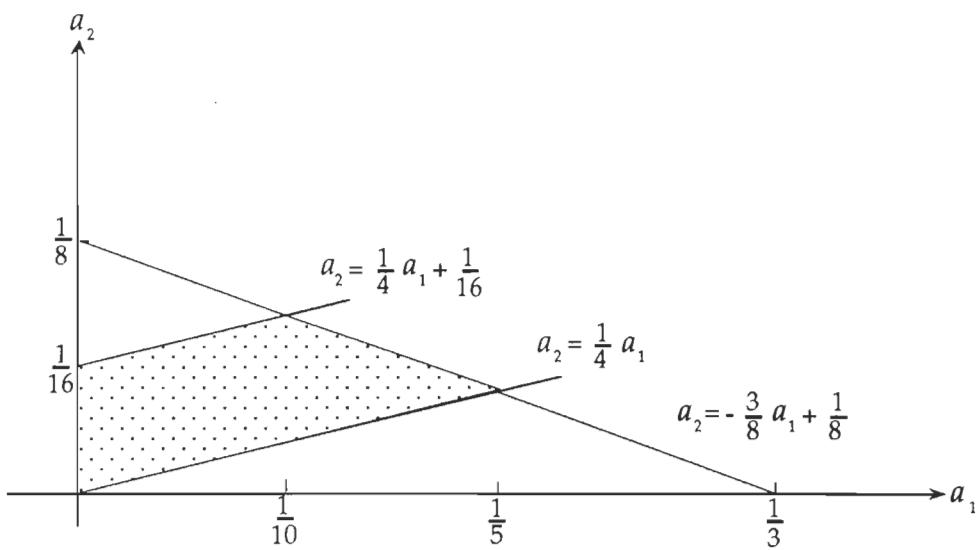


Fig 3.1 Région R de faisabilité correspondante au modèle logistique

3.1.5 RÔLE DE L'ÉCHELLE DE TEMPS « h ».

La figure 3.1 indique que si un couple de valeurs (a_1, a_2) est choisi ailleurs que dans la zone ombragée, alors il est certain qu'au moins une probabilité $P_{\alpha \rightarrow \beta}$ sera négative. Toutefois, nos façons de choisir un couple de valeurs (a_1, a_2) sont plus étendues que ne le laisse croire cette région R. Qu'en est-il exactement. Rappelons que dans le but de simplifier le développement de (3.1) nous avions posé l'échelle de temps $h = 1$. Or, il est fondamental de mentionner que ce facteur d'échelle de temps « h » demeure, en général, bel et bien présent lorsque l'on développe les équations de (3.1). Pour illustrer convenablement ce qui vient d'être dit, il nous suffit de remarquer que dans le cadre à 1 espèce les équations constitutives sont en réalité fournies par :

$$\begin{aligned}
 a) \quad P_{0 \rightarrow 1} - P_{0 \rightarrow -1} &= h(a_0^{(1)}) \\
 b) \quad P_{1 \rightarrow 2} - P_{1 \rightarrow 0} &= h(a_0^{(1)} + a_1^{(1)}) \\
 c) \quad P_{2 \rightarrow 3} - P_{2 \rightarrow 1} &= h(a_0^{(1)} + 2a_1^{(1)} + 8/3a_2^{(1)}) \\
 d) \quad P_{3 \rightarrow 4} - P_{3 \rightarrow 2} &= h(a_0^{(1)} + 3a_1^{(1)} + 8a_2^{(1)} + 16a_3^{(1)}) \\
 e) \quad P_{4 \rightarrow 5} - P_{4 \rightarrow 3} &= h(a_0^{(1)} + 4a_1^{(1)} + 16a_2^{(1)} + 64a_3^{(1)} + 256a_4^{(1)})
 \end{aligned} \tag{3.17}$$

En particulier les équations constitutives du modèle logistique sont en réalité fournies par :

$$\begin{aligned}
 a) \quad P_{0 \rightarrow 1} &= h(0) \\
 b) \quad P_{1 \rightarrow 2} - P_{1 \rightarrow 0} &= h(a_1) \\
 c) \quad P_{2 \rightarrow 3} - P_{2 \rightarrow 1} &= h(2a_1 - 8/3a_2) \\
 d) \quad P_{3 \rightarrow 4} - P_{3 \rightarrow 2} &= h(3a_1 - 8a_2) \\
 e) \quad P_{4 \rightarrow 3} &= h(-4a_1 + 16a_2)
 \end{aligned} \tag{3.18}$$

À présent, nous sommes en excellente position pour entreprendre une discussion du rôle que va jouer le facteur « h » que l'on nomme en anglais « *time scaling factor* »¹⁻¹⁴. Nous allons voir que la présence de ce facteur h permet d'étendre considérablement la région R de faisabilité qui a été construite sous l'hypothèse $h = 1$. Puisque telle est notre situation nous allons reprendre la démarche qui nous a conduit à l'obtention de la solution unique (3.15) sans, cette fois, attribuer de

valeur précise au facteur « h ». En appliquant la règle de correspondance des signes aux équations $b)$ à $d)$ de (3.18) on obtient :

$$\boxed{\begin{array}{lll} a) P_{1 \rightarrow 2} = h a_1 & c) P_{2 \rightarrow 3} = h 2 a_1 & e) P_{3 \rightarrow 4} = h 3 a_1 \\ b) P_{1 \rightarrow 0} = 0 & d) P_{2 \rightarrow 1} = h 8/3 a_2 & f) P_{3 \rightarrow 2} = h 8 a_2 \end{array}} \quad (3.19)$$

Un regard simple sur les équations de (3.19) nous indique que : $\forall \alpha, 1 \leq \alpha \leq 3$ alors il existe au moins un h tel que : $P_{\alpha \rightarrow \alpha \pm 1}$ soit positif et aussi près de zéro que l'on désire. Concrètement, cela signifie que : $\forall \alpha, 1 \leq \alpha \leq m-1$ toutes les contraintes du type : $0 \leq P_{\alpha \rightarrow \alpha \pm 1} \leq 1$ peuvent être délaissées en autant que l'on choisisse une valeur de « h » suffisamment petite. De plus $\forall \alpha, 1 \leq \alpha \leq 3$, la condition $P_{\alpha \rightarrow \alpha - 1} + P_{\alpha \rightarrow \alpha} + P_{\alpha \rightarrow \alpha + 1} = 1$ peut assurément être satisfaite car une valeur appropriée de « h » rend $P_{\alpha \rightarrow \alpha \pm 1}$ aussi près de zéro que l'on désire. Dans l'ensemble, le facteur h d'échelle de temps semble nous permettre de s'affranchir considérablement des contraintes $i)$ et $ii)$ de (3.16). La question à laquelle il nous faut à présent répondre est la suivante : comment trouver cette valeur de « h » qui satisfasse $i)$ et $ii)$ de (3.16). Plusieurs critères peuvent être utilisés pour choisir la valeur de h . Par exemple, on peut choisir h dans l'intervalle $0 < h \leq 1/p_{sup}$ où p_{sup} peut être calculé par :

$$p_{sup} = \underset{0 \leq \alpha \leq m}{\text{Max}} \left\{ \sum_{v=0}^{\alpha} \frac{\binom{\alpha}{v} m^v}{\binom{m}{v}} \left| a_v^{(1)} \right| \right\} = \sum_{v=0}^m m^v \left| a_v^{(1)} \right| \quad (3.20)$$

Dans un modèle à une espèce, une borne supérieure plus petite pour p_{sup} peut être obtenue par le calcul suivant :

$$p_{sup} = \underset{\substack{0 \leq \alpha \leq m \\ \alpha \neq \beta}}{\text{Max}} \left\{ \sum_{\beta=0}^m \frac{P_{\alpha \rightarrow \beta}}{h} \right\} \quad (3.21)$$

Les résultats (3.20) et (3.21) nous assurent que quelque soit le choix des valeurs du couple (a_1, a_2) on peut trouver une échelle de temps h satisfaisant : $P_{\alpha \rightarrow \beta} \leq 1, \forall \alpha, 0 \leq \alpha \leq m, \alpha - 1 \leq \beta \leq \alpha + 1$. Toutefois, il n'est pas encore garanti que si $\alpha = 0$ ou $\alpha = m$ la condition $P_{\alpha \rightarrow \beta} \geq 0$ soit remplie.

En effet, les équations $a)$ et $e)$ de (3.18) signalent que les valeurs de probabilités $P_{4 \rightarrow 3}$ et $P_{0 \rightarrow 1}$ peuvent demeurer négatives peu importe le choix de l'échelle de temps h . En principe, nous nous devons donc de traiter séparément les équations $a)$ et $e)$ en vérifiant pour quelles valeurs de a_1 et a_2 les deux contraintes $P_{4 \rightarrow 3} \geq 0$ et $P_{0 \rightarrow 1} \geq 0$ sont satisfaites. Il est dit « en principe » car nous verrons bientôt une façon plus élégante de vérifier si la contrainte $P_{\alpha \rightarrow \beta} \geq 0$ peut être respectée ou non.

À présent, reprenons les équations de (3.18) concernant le modèle logistique. Selon (3.18) on voit alors que $P_{0 \rightarrow 1} \geq 0$ est satisfaite tandis que $P_{4 \rightarrow 3} \geq 0 \Leftrightarrow a_2 \geq 0.25 a_1$. Cette contrainte $a_2 \geq 0.25 a_1$ devra donc continuer à être respectée lors du choix des valeurs de a_1 et a_2 . Globalement, avons-nous gagné plus de liberté à propos du choix des coefficients a_1 et a_2 ? La figure 3.2 résume l'ensemble des possibilités que nous a offert le facteur h . Sur la figure 3.2 cette ancienne région R a été démarquée par un noircissement plus prononcé. Cette figure veut surtout illustrer qu'un couple de points (a_1, a_2) situé au-dessus de la droite $a_2 = 0.25 a_1$ peut toujours être contractée dans la région R préalablement construite en prenant $h = 1$. L'extension qu'a subie cette ancienne région R est remarquable. Toutefois, dans un contexte où la croissance d'une espèce se modélise par l'équation logistique il est assez fréquent que les valeurs a_1 et a_2 soient quantitativement liées entre elles par un facteur 100. Expérimentalement du moins cela semble être le cas³². Or, l'hypothèse $a_2 \approx 0.01 a_1$ ne sera pas permise dans le cadre d'une simulation car, comme nous l'avons mentionné précédemment, la contrainte $a_2 \geq 0.25 a_1$ nous interdit d'envisager cette situation type où $a_2 \approx 0.01 a_1$. On peut objecter que cette impasse est une conséquence directe de la façon dont nous avons donné une solution unique à (3.14) et qu'un autre type d'affectations pourrait par la suite faire disparaître cette contrainte.

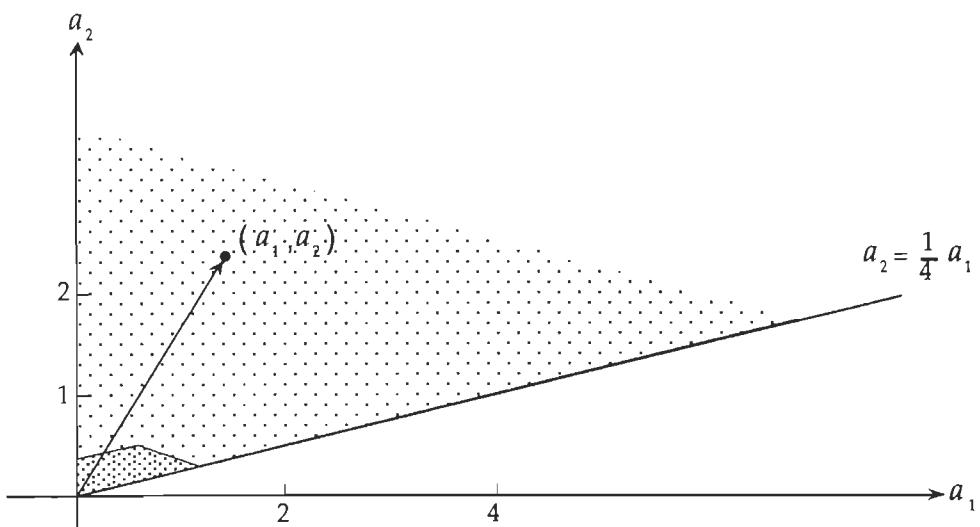


Fig 3.2 Extension de la région de faisabilité du modèle logistique via le facteur « h »

Bien quelle soit pertinente, cette objection ne tient pas. En effet, si l'on souhaite s'approcher de la condition expérimentale type : $a_2 \approx 0.01 a_1$, il faut alors noter que la contrainte la plus restrictive à cet égard est certainement $a_2 \geq 0.25 a_1$. Or, nous avons démontré que pour résoudre de façon unique le système d'équations (3.14), nous aurons toujours à choisir a_1 et a_2 tel que $a_2 \geq 0.25 a_1$, et ce, quelle que soit la façon dont nous affecterons les probabilités $P_{\alpha \rightarrow \beta}$. À la lueur de cette discussion, concernant le taux de croissance caractéristique du *modèle logistique*, il semble donc que le taux de croissance empirique type ne puisse être atteint. Il nous faudra donc apprendre à réfléchir avec ces contraintes restrictives liées aux lois de probabilités...

Nous voilà donc au terme de l'analyse de faisabilité des modèles classiques à 1 espèce. Bien que simples, il faut mentionner que ces modèles à 1 espèce nous ont permis de clarifier et d'illustrer plusieurs concepts et de préciser la nature et le rôle de tous les paramètres qui interviennent dans le résultat principal (2.13). En particulier, nous avons eu l'occasion d'illustrer quel rôle jouera le facteur d'échelle de temps h et comment le choisir de façon adéquate. De plus, nous avons su illustrer simplement comment s'utilise la proposition (2.13). Globalement, on peut donc dire que l'étude des modèles à 1 espèce s'est avérée utile. À présent, nous allons étendre cette démarche dans le cadre des modèles à 2 espèces. Nous aurons alors l'occasion d'énoncer un critère qui nous permettra, en général, de connaître aisément quelles sont les contraintes essentielles à préserver dans (3.10).

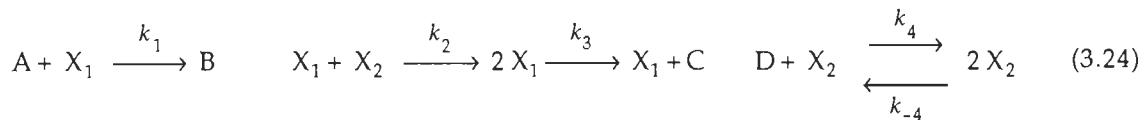
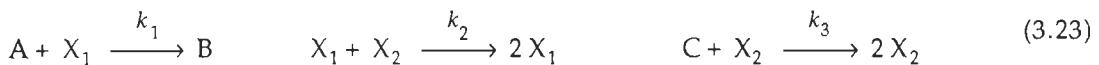
3.2 ANALYSE DE FAISABILITÉ DES MODÈLES CLASSIQUE À 2 ESPÈCES

Nous avons vu au chapitre II que les mécanismes (2.19) et (2.21) engendrent respectivement les systèmes d'équations de *Lotka Volterra classique et étendu*. Dans cette section nous essayerons de répondre à la question suivante : est-il possible de recréer, sur un réseau cartésien, ces deux mécanismes de compétition qui s'apparentent aux modèles de *Lotka Volterra classique et étendu*? Pour répondre à cette question, la démarche à suivre sera en tout point identique à celle qui a été empruntée dans le cadre des modèles à 1 espèce. Tout d'abord rappelons que la proposition (3.1) énonce :

$$\sum_{u=0}^2 (u_i - 1) P_{\alpha \rightarrow \alpha + u - 1} = h \sum_{v=0}^{\alpha} \prod_{j=1}^n \frac{\binom{\alpha_j}{v_j} m^{v_j}}{\binom{m}{v_j}} a_v^{(i)} \quad (3.22)$$

À deux espèces, les vecteurs α et β sont explicités de la sorte : $\alpha = (\alpha_1, \alpha_2)$ et $\beta = (\beta_1, \beta_2)$ où $\alpha_i, \beta_j \in S$ avec $S = \{0 \dots m\}$. Les indices vectoriels u et 1 devraient à leur tour être explicités de la sorte : $u = (u_1, u_2)$ tandis que $1 = (1, 1)$. Par ailleurs puisque deux espèces évoluent on a $n = 2$

tandis que i peut prendre la valeur 1 ou 2. Les sommes compactes qui sont contenues au résultat (3.1) devraient être explicitées pour que le calcul puisse être effectué. De plus, en inspectant cette égalité on remarque qu'à chaque état $\alpha = (\alpha_1, \alpha_2)$ fixé que peut prendre une cellule, il correspond une équation associée à l'espèce de type i . En principe, $m+1$ valeurs sont possibles pour chaque variable α_ξ , $1 \leq \xi \leq 2$. On déduit¹⁰ alors aisément qu'il y aura $(m+1)^2$ équations correspondant à l'espèce de type i . Si n espèces forment le modèle d'interaction alors on aura au total $n(m+1)^2$ équations à résoudre. Dans notre cas $m = 4$ puisque le réseau est cartésien et $n = 2$ car nous sommes en présence de 2 espèces. On aura donc à résoudre un système constitué de 50 équations dont 25 sont associées à la première espèce, tandis que les 25 autres sont associées à la seconde espèce. D'autre part, combien d'inconnues $P_{\alpha \rightarrow \beta}$ sont présentes dans ces équations ? Bien qu'elles soient un peu plus difficiles à dénombrer, un petit calcul indique qu'il y a $2m(2m+1)$ probabilités $P_{\alpha \rightarrow \beta}$ qui sont associées à chacune des espèces. Au total, on aura donc à résoudre un système de $2(m+1)^2$ équations à $4m(2m+1)$ inconnues. Dans notre cas, ceci produit donc un système constitué de 50 équations comportant 144 inconnues. Pour donner une solution unique à ce système, d'autres restrictions doivent donc être apportées puisque 94 probabilités $P_{\alpha \rightarrow \beta}$ demeurent libres. Notons que sur un réseau autre que cartésien $2(3m^2-1)$ inconnues seraient libres. Ces remarques devraient justifier pourquoi il est ici impossible de présenter explicitement le système d'équations qui caractérise, en général, une dynamique entre 2 espèces. À présent, voyons comment nous allons donner une solution unique à ce système. Le point de départ de cette discussion s'appuiera sur les diagrammes réactionnels des modèles de *Lotka Volterra classique* et étendu. Pour simplifier la discussion qui suivra, rappelons ces diagrammes :



L'inspection des mécanismes (3.23) et (3.24) révèle que les transitions se limitent bel et bien à des accroissements de 1, ou à des diminutions de 1. Cette contrainte, que nous avons apportée dans le cadre théorique de l'annexe A concernant les types de mécanismes envisagés, est donc ici respectée. Or, un regard plus attentif à (3.23) et (3.24) indique que non seulement ces accroissements ou diminutions sont de 1 mais qu'en plus, ces bonds de 1 s'effectuent uniquement *par une et une seule espèce à la fois*. Pour respecter cette caractéristique nous allons donc poser

égaux à zéro les $P_{\alpha \rightarrow \beta}$ correspondant aux transitions $(\alpha_1, \alpha_2) \rightarrow (\alpha_1 \pm 1, \alpha_2 \pm 1)$. Dans le cadre de notre problématique, ce diagramme fléché se traduit par l'énoncé suivant : *nous allons interdire que les 2 espèces effectuent en même temps, à une même cellule du réseau, une variation de population¹⁻¹¹*. Puisque se sont les mécanismes eux-mêmes qui suggèrent ce type de dynamique, l'application de cette dernière contrainte à travers les $P_{\alpha \rightarrow \beta}$ devrait rendre plus cohérente leur simulation sur le réseau cartésien¹¹. Toutefois, avant d'aller plus loin, notons que l'ajout de cette contrainte permet de réécrire le membre de gauche du résultat (3.1). En effet, s'il faut restreindre les transitions telles que nous l'avons mentionné on aura alors :

$$\sum_{u=0}^2 (u_i - 1) P_{\alpha \rightarrow \alpha + u - 1} = \begin{cases} P_{(\alpha_1, \alpha_2) \rightarrow (\alpha_1+1, \alpha_2)} - P_{(\alpha_1, \alpha_2) \rightarrow (\alpha_1-1, \alpha_2)} & \text{si } i = 1 \\ P_{(\alpha_1, \alpha_2) \rightarrow (\alpha_1, \alpha_2+1)} - P_{(\alpha_1, \alpha_2) \rightarrow (\alpha_1, \alpha_2-1)} & \text{si } i = 2 \end{cases} \quad (3.25)$$

Par (3.25) il est intéressant de constater que les probabilités liées aux processus de croissance se voient affectées d'un signe « + », tandis que les probabilités liées aux processus de décroissance se voient affectées d'un signe « - », et ce, pour chacune des espèces en jeu. D'autre part, nous savons que le signe des coefficients des équations différentielles, découlant des mécanismes réactionnels, est lui aussi associé à une croissance, s'il est positif, et à une décroissance, s'il est négatif. Les rudiments d'un cours d'équations différentielles nous enseignent aussi ce fait. Nous allons tirer profit de cette correspondance entre les signes des coefficients des équations différentielles et des signes qui affectent les $P_{\alpha \rightarrow \beta}$ de l'égalité (3.25), pour obtenir la solution unique désirée. Avant d'établir cette correspondance entre ces signes, notons que le membre de droite du résultat (3.1) s'écrit dans le présent contexte comme :

$$h \sum_{v_1=0}^{\alpha_1} \sum_{v_2=0}^{\alpha_2} \prod_{j=1}^2 \frac{\binom{\alpha_j}{v_j} m^{v_j}}{\binom{m}{v_j}} a_{(v_1, v_2)}^{(i)} = h \sum_{v_1=0}^{\alpha_1} \sum_{v_2=0}^{\alpha_2} \frac{\binom{\alpha_1}{v_1} \binom{\alpha_2}{v_2}}{\binom{m}{v_1} \binom{m}{v_2}} m^{v_1 + v_2} a_{(v_1, v_2)}^{(i)} \quad (3.26)$$

De (3.25) et (3.26) on obtient :

$$h \sum_{v_1=0}^{\alpha_1} \sum_{v_2=0}^{\alpha_2} \frac{\binom{\alpha_1}{v_1} \binom{\alpha_2}{v_2}}{\binom{m}{v_1} \binom{m}{v_2}} m^{v_1 + v_2} a_{(v_1, v_2)}^{(i)} = \begin{cases} P_{(\alpha_1, \alpha_2) \rightarrow (\alpha_1+1, \alpha_2)} - P_{(\alpha_1, \alpha_2) \rightarrow (\alpha_1-1, \alpha_2)} & \text{si } i = 1 \\ P_{(\alpha_1, \alpha_2) \rightarrow (\alpha_1, \alpha_2+1)} - P_{(\alpha_1, \alpha_2) \rightarrow (\alpha_1, \alpha_2-1)} & \text{si } i = 2 \end{cases} \quad (3.27)$$

Rappelons que les coefficients $a_{(v_1, v_2)}^{(i)}$ de (3.27) sont ceux qui apparaissent dans l'équation différentielle correspondant à l'espèce de type i . Cette nouvelle contrainte concernant le type de transitions qui seront permises ayant été posée, posons à présent la question suivante : combien de probabilités $P_{\alpha \rightarrow \beta}$ se trouvent alors à être affectées à zéro ? Un petit exercice de combinatoire montre qu'il y a $4m^2$ transitions du type : $(\alpha_1, \alpha_2) \rightarrow (\alpha_1 \pm 1, \alpha_2 \pm 1)$. Des calculs précédents, concernant le nombre d'inconnues des modèles à 2 espèces, on déduit donc qu'il reste malgré tout $4m(m+1)$ inconnues. Notre système est donc à présent constitué de 50 équations et de 80 inconnues. Il y a encore trop de degrés de liberté. Pour poursuivre la discussion concernant la méthode que nous allons utiliser pour donner une valeur unique aux $P_{\alpha \rightarrow \beta}$, nous allons nous donner un mécanisme précis sur lequel on pourra travailler.

3.2.1 ÉQUATIONS ASSOCIÉES AU MODÈLE DE LOTKA VOLTERRA ÉTENDU

Comme nous l'avons vu au chapitre II, le mécanisme (3.23) contient le modèle de *Lotka Volterra classique* en le généralisant un peu plus. Nous allons donc raisonner sur ce mécanisme (3.23), quitte à retrouver par la suite le modèle de *Lotka Volterra classique* comme un cas particulier de ce premier. Le résultat (2.22) confirme que le mécanisme (3.23) conduit aux équations différentielles suivantes:

$$\begin{aligned} \frac{d\rho_1}{dt} &= -a_{1,0}\rho_1 + a_{1,1}\rho_1\rho_2 - a_{2,0}\rho_1^2 \\ \frac{d\rho_2}{dt} &= b_{0,1}\rho_2 - b_{1,1}\rho_1\rho_2 - b_{0,2}\rho_2^2 \end{aligned} \tag{3.28}$$

Par le système (3.28) nous avons accès aux valeurs de $a_{(v_1, v_2)}^{(i)}$. Voici donc les valeurs de ces coefficients $a_{(v_1, v_2)}^{(i)}$ après les avoir identifiés au système (3.28) :

$$a_{(1,0)}^{(1)} = -a_{1,0} \quad a_{(1,1)}^{(1)} = +a_{1,1} \quad a_{(2,0)}^{(1)} = -a_{2,0}$$

$$a_{(0,1)}^{(2)} = b_{0,1} \quad a_{(1,1)}^{(2)} = -b_{1,1} \quad a_{(0,2)}^{(2)} = -b_{0,2}$$

Pour ce mécanisme il s'en suit donc que tous les autres $a_{(v_1, v_2)}^{(i)}$ sont nuls. Puisque l'occasion s'y prête, mentionnons que les équations différentielles associées au modèle de *Lotka Volterra classique* s'obtiennent en posant $a_{2,0} = b_{0,2} = 0$. Nous pouvons maintenant présenter les équations probabilistes qui découlent du système d'équations différentielles (3.28). Dans ce contexte, le développement de (3.27) nous donnera respectivement les équations qui sont associées à chacune

<u>Équations associées à l'espèce de type I (prédateur)</u>	<u>Équations associées à l'espèce de type II (proie)</u>
$ \begin{aligned} & + P_{(0,0) \rightarrow (1,0)} = 0 \\ & + P_{(0,1) \rightarrow (1,1)} = 0 \\ & + P_{(0,2) \rightarrow (1,2)} = 0 \\ & + P_{(0,3) \rightarrow (1,3)} = 0 \\ & + P_{(0,4) \rightarrow (1,4)} = 0 \\ & - P_{(1,0) \rightarrow (0,0)} + P_{(1,0) \rightarrow (2,0)} = -1 a_{1,0} \\ & - P_{(1,1) \rightarrow (0,1)} + P_{(1,1) \rightarrow (2,1)} = -1 a_{1,0} + 1 a_{1,1} \\ & - P_{(1,2) \rightarrow (0,2)} + P_{(1,2) \rightarrow (2,2)} = -1 a_{1,0} + 2 a_{1,1} \\ & - P_{(1,3) \rightarrow (0,3)} + P_{(1,3) \rightarrow (2,3)} = -1 a_{1,0} + 3 a_{1,1} \\ & - P_{(1,4) \rightarrow (0,4)} + P_{(1,4) \rightarrow (2,4)} = -1 a_{1,0} + 4 a_{1,1} \\ & - P_{(2,0) \rightarrow (1,0)} + P_{(2,0) \rightarrow (3,0)} = -2 a_{1,0} - 8/3 a_{2,0} \\ & - P_{(2,1) \rightarrow (1,1)} + P_{(2,1) \rightarrow (3,1)} = -2 a_{1,0} + 2 a_{1,1} - 8/3 a_{2,0} \\ & - P_{(2,2) \rightarrow (1,2)} + P_{(2,2) \rightarrow (3,2)} = -2 a_{1,0} + 4 a_{1,1} - 8/3 a_{2,0} \\ & - P_{(2,3) \rightarrow (1,3)} + P_{(2,3) \rightarrow (3,3)} = -2 a_{1,0} + 6 a_{1,1} - 8/3 a_{2,0} \\ & - P_{(2,4) \rightarrow (1,4)} + P_{(2,4) \rightarrow (3,4)} = -2 a_{1,0} + 8 a_{1,1} - 8/3 a_{2,0} \\ & - P_{(3,0) \rightarrow (2,0)} + P_{(3,0) \rightarrow (4,0)} = -3 a_{1,0} - 8 a_{1,1} \\ & - P_{(3,1) \rightarrow (2,1)} + P_{(3,1) \rightarrow (4,1)} = -3 a_{1,0} + 3 a_{1,1} - 8 a_{2,0} \\ & - P_{(3,2) \rightarrow (2,2)} + P_{(3,2) \rightarrow (4,2)} = -3 a_{1,0} + 6 a_{1,1} - 8 a_{2,0} \\ & - P_{(3,3) \rightarrow (2,3)} + P_{(3,3) \rightarrow (4,3)} = -3 a_{1,0} + 9 a_{1,1} - 8 a_{2,0} \\ & - P_{(3,4) \rightarrow (2,4)} + P_{(3,4) \rightarrow (4,4)} = -3 a_{1,0} + 12 a_{1,1} - 8 a_{2,0} \\ & - P_{(4,0) \rightarrow (3,0)} = -4 a_{1,0} - 16 a_{2,0} \\ & - P_{(4,1) \rightarrow (3,1)} = -4 a_{1,0} + 4 a_{1,1} - 16 a_{2,0} \\ & - P_{(4,2) \rightarrow (3,2)} = -4 a_{1,0} + 8 a_{1,1} - 16 a_{2,0} \\ & - P_{(4,3) \rightarrow (3,3)} = -4 a_{1,0} + 12 a_{1,1} - 16 a_{2,0} \\ & - P_{(4,4) \rightarrow (3,4)} = -4 a_{1,0} + 16 a_{1,1} - 16 a_{2,0} \end{aligned} $	$ \begin{aligned} & + P_{(0,0) \rightarrow (0,1)} = 0 \\ & + P_{(1,0) \rightarrow (1,1)} = 0 \\ & + P_{(2,0) \rightarrow (2,1)} = 0 \\ & + P_{(3,0) \rightarrow (3,1)} = 0 \\ & + P_{(4,0) \rightarrow (4,1)} = 0 \\ & - P_{(0,1) \rightarrow (0,0)} + P_{(0,1) \rightarrow (0,2)} = +1 b_{0,1} \\ & - P_{(1,1) \rightarrow (1,0)} + P_{(1,1) \rightarrow (1,2)} = +1 b_{0,1} - 1 b_{1,1} \\ & - P_{(2,1) \rightarrow (2,0)} + P_{(2,1) \rightarrow (2,2)} = +1 b_{0,1} - 2 b_{1,1} \\ & - P_{(3,1) \rightarrow (3,0)} + P_{(3,1) \rightarrow (3,2)} = +1 b_{0,1} - 3 b_{1,1} \\ & - P_{(4,1) \rightarrow (4,0)} + P_{(4,1) \rightarrow (4,2)} = +1 b_{0,1} - 4 b_{1,1} \\ & - P_{(0,2) \rightarrow (0,1)} + P_{(0,2) \rightarrow (0,3)} = +2 b_{0,1} - 8/3 b_{0,2} \\ & - P_{(1,2) \rightarrow (1,1)} + P_{(1,2) \rightarrow (1,3)} = +2 b_{0,1} - 8/3 b_{0,2} - 2 b_{1,1} \\ & - P_{(2,2) \rightarrow (2,1)} + P_{(2,2) \rightarrow (2,3)} = +2 b_{0,1} - 8/3 b_{0,2} - 4 b_{1,1} \\ & - P_{(3,2) \rightarrow (3,1)} + P_{(3,2) \rightarrow (3,3)} = +2 b_{0,1} - 8/3 b_{0,2} - 6 b_{1,1} \\ & - P_{(4,2) \rightarrow (4,1)} + P_{(4,2) \rightarrow (4,3)} = +2 b_{0,1} - 8/3 b_{0,2} - 8 b_{1,1} \\ & - P_{(0,3) \rightarrow (0,2)} + P_{(0,3) \rightarrow (0,4)} = +3 b_{0,1} - 8 b_{0,2} \\ & - P_{(1,3) \rightarrow (1,2)} + P_{(1,3) \rightarrow (1,4)} = +3 b_{0,1} - 8 b_{0,2} - 3 b_{1,1} \\ & - P_{(2,3) \rightarrow (2,2)} + P_{(2,3) \rightarrow (2,4)} = +3 b_{0,1} - 8 b_{0,2} - 6 b_{1,1} \\ & - P_{(3,3) \rightarrow (3,2)} + P_{(3,3) \rightarrow (3,4)} = +3 b_{0,1} - 8 b_{0,2} - 9 b_{1,1} \\ & - P_{(4,3) \rightarrow (4,2)} + P_{(4,3) \rightarrow (4,4)} = +3 b_{0,1} - 8 b_{0,2} - 12 b_{1,1} \\ & - P_{(0,4) \rightarrow (0,3)} = +4 b_{0,1} - 16 b_{0,2} \\ & - P_{(1,4) \rightarrow (1,3)} = +4 b_{0,1} - 16 b_{0,2} - 4 b_{1,1} \\ & - P_{(2,4) \rightarrow (2,3)} = +4 b_{0,1} - 16 b_{0,2} - 8 b_{1,1} \\ & - P_{(3,4) \rightarrow (3,3)} = +4 b_{0,1} - 16 b_{0,2} - 12 b_{1,1} \\ & - P_{(4,4) \rightarrow (4,3)} = +4 b_{0,1} - 16 b_{0,2} - 16 b_{1,1} \end{aligned} $

Tableau 3.1 Équations pour le modèle de Lotka Volterra étendu.

La valeur $h = 1$ a été posée lors du développement de (3.21)

des 2 espèces en jeu. Ces équations sont données explicitement au tableau 3.1. En inspectant ce tableau, un petit exercice d'arithmétique nous révèle qu'il y a effectivement 50 équations et 80 inconnues. À présent, nous sommes en mesure de poursuivre notre discussion concernant la façon de donner une solution unique à ce système. Comme nous l'avons déjà mentionné, pour obtenir une solution unique, nous allons utiliser une règle basée sur l'identification entre le signe qui affecte une probabilité $P_{\alpha \rightarrow \beta}$, et le signe des coefficients de l'égalité de droite. Voyons comment s'établit cette correspondance en regardant les équations associées à l'espèce de type I. Pour ce cas, la règle de correspondance s'énonce de la façon suivante :

$$P_{(\alpha_1, \alpha_2) \rightarrow (\alpha_1 + 1, \alpha_2)} = \sum_{v_1=0}^{\alpha_1} \sum_{v_2=0}^{\alpha_2} \frac{\binom{\alpha_1}{v_1} \binom{\alpha_2}{v_2}}{\binom{m}{v_1} \binom{m}{v_2}} m^{v_1 + v_2} a_{(v_1, v_2)}^{(1)} \quad \text{pour } a_{(v_1, v_2)}^{(1)} < 0 \quad (3.29)$$

$$P_{(\alpha_1, \alpha_2) \rightarrow (\alpha_1, \alpha_2 + 1)} = \sum_{v_1=0}^{\alpha_1} \sum_{v_2=0}^{\alpha_2} \frac{\binom{\alpha_1}{v_1} \binom{\alpha_2}{v_2}}{\binom{m}{v_1} \binom{m}{v_2}} m^{v_1 + v_2} a_{(v_1, v_2)}^{(1)} \quad \text{pour } a_{(v_1, v_2)}^{(1)} > 0 \quad (3.30)$$

En guise d'exemple, appliquons cette règle de correspondance à la 11^{ième} équation du tableau 3.1 associée à l'espèce de type I. La règle de correspondance appliquée à cette 11^{ième} équation permet de calculer les probabilités $P_{(2,0) \rightarrow (1,0)}$ et $P_{(2,0) \rightarrow (3,0)}$ de la sorte :

$$P_{(2,0) \rightarrow (1,0)} = 2 a_{1,0} + 8/3 a_{2,0} \quad (3.31)$$

$$P_{(2,0) \rightarrow (3,0)} = 0 \quad (3.32)$$

En appliquant cette règle de correspondance à la 12^{ième} équation on aurait :

$$P_{(2,1) \rightarrow (1,1)} = 2 a_{1,0} + 8/3 a_{2,0} \quad (3.33)$$

$$P_{(2,1) \rightarrow (3,1)} = 2 a_{1,1} \quad (3.34)$$

Ainsi de suite. Qu'advient-il de cette règle si une seule inconnue $P_{\alpha \rightarrow \beta}$ apparaît dans une équation ? Pour ce cas, nous n'avons pas à appliquer cette règle de correspondance car si $P_{\alpha \rightarrow \beta}$ apparaît seul dans une équation, alors c'est qu'elle est automatiquement déterminée

par les valeurs des coefficients connus situés à droite de l'égalité ! L'utilisation de cette règle de correspondance se valide lorsqu'on remarque la similitude entre les deux énoncés suivants :

- i) Un coefficient $a_{(v_1, v_2)}^{(i)} > 0$ d'une équation différentielle signale un processus de croissance, tandis qu'un coefficient $a_{(v_1, v_2)}^{(i)} < 0$ d'une équation différentielle signale un processus de décroissance.
- ii) Une probabilité $P_{\alpha \rightarrow \beta}$ affectée du signe « + » signale un accroissement de population, tandis qu'une probabilité $P_{\alpha \rightarrow \beta}$ affectée du signe « - » signale une diminution de population.

Cette façon^{1,2} de donner une solution unique aux équations n'est pas la seule qui puisse être envisagée. Nous abordons là un sujet délicat. En effet, les techniques décisionnelles visant à donner une solution unique à (3.1) sont à tout le moins imprécises¹¹. Pour rendre certaines solutions plus crédibles que d'autres, des motifs d'ordre physique peuvent être invoqués. Le but de ces considérations physiques vise ultimement à respecter le mieux possible la dynamique du mécanisme réactionnel que l'on souhaite reproduire par simulation. En ce qui nous concerne, nous avons solutionné^{1,2} les équations constitutives de (3.1) en apportant une règle de correspondance en signe. L'utilisation de cette règle donne à l'ensemble du traitement précédent une plus grande unité. En particulier, nous n'avons pas eu à justifier, par des motifs plus ou moins convaincants¹¹, la nature des affectations des probabilités $P_{\alpha \rightarrow \beta}$. Si les équations visant à fonder un automate cellulaire offrent effectivement un grand éventail de possibilités¹¹, alors la façon de résoudre que nous avons présentée doit certainement en faire partie; quitte à se justifier par la suite. Concernant cette latitude de choix qu'offrent les équations visant à fonder un automate cellulaire, une discussion supplémentaire sera apportée à la fin de ce travail. Nous aurons alors l'occasion d'identifier les avantages et les inconvénients de la méthode présentée dans le cadre du volet expérimental de ce travail. Ces brefs commentaires étant apportés, passons à l'analyse de faisabilité du modèle de *Lotka Volterra classique*.

<u>Équations associées à l'espèce de type I (prédateur)</u>	<u>Équations associées à l'espèce de type II (proie)</u>
$ \begin{aligned} + P_{(0,0) \rightarrow (1,0)} &= 0 \\ + P_{(0,1) \rightarrow (1,1)} &= 0 \\ + P_{(0,2) \rightarrow (1,2)} &= 0 \\ + P_{(0,3) \rightarrow (1,3)} &= 0 \\ + P_{(0,4) \rightarrow (1,4)} &= 0 \\ - P_{(1,0) \rightarrow (0,0)} + P_{(1,0) \rightarrow (2,0)} &= - 1 a_{1,0} \\ - P_{(1,1) \rightarrow (0,1)} + P_{(1,1) \rightarrow (2,1)} &= - 1 a_{1,0} + 1 a_{1,1} \\ - P_{(1,2) \rightarrow (0,2)} + P_{(1,2) \rightarrow (2,2)} &= - 1 a_{1,0} + 2 a_{1,1} \\ - P_{(1,3) \rightarrow (0,3)} + P_{(1,3) \rightarrow (2,3)} &= - 1 a_{1,0} + 3 a_{1,1} \\ - P_{(1,4) \rightarrow (0,4)} + P_{(1,4) \rightarrow (2,4)} &= - 1 a_{1,0} + 4 a_{1,1} \\ - P_{(2,0) \rightarrow (1,0)} + P_{(2,0) \rightarrow (3,0)} &= - 2 a_{1,0} \\ - P_{(2,1) \rightarrow (1,1)} + P_{(2,1) \rightarrow (3,1)} &= - 2 a_{1,0} + 2 a_{1,1} \\ - P_{(2,2) \rightarrow (1,2)} + P_{(2,2) \rightarrow (3,2)} &= - 2 a_{1,0} + 4 a_{1,1} \\ - P_{(2,3) \rightarrow (1,3)} + P_{(2,3) \rightarrow (3,3)} &= - 2 a_{1,0} + 6 a_{1,1} \\ - P_{(2,4) \rightarrow (1,4)} + P_{(2,4) \rightarrow (3,4)} &= - 2 a_{1,0} + 8 a_{1,1} \\ - P_{(3,0) \rightarrow (2,0)} + P_{(3,0) \rightarrow (4,0)} &= - 3 a_{1,0} \\ - P_{(3,1) \rightarrow (2,1)} + P_{(3,1) \rightarrow (4,1)} &= - 3 a_{1,0} + 3 a_{1,1} \\ - P_{(3,2) \rightarrow (2,2)} + P_{(3,2) \rightarrow (4,2)} &= - 3 a_{1,0} + 6 a_{1,1} \\ - P_{(3,3) \rightarrow (2,3)} + P_{(3,3) \rightarrow (4,3)} &= - 3 a_{1,0} + 9 a_{1,1} \\ - P_{(3,4) \rightarrow (2,4)} + P_{(3,4) \rightarrow (4,4)} &= - 3 a_{1,0} + 12 a_{1,1} \\ - P_{(4,0) \rightarrow (3,0)} &= - 4 a_{1,0} \\ - P_{(4,1) \rightarrow (3,1)} &= - 4 a_{1,0} + 4 a_{1,1} \\ - P_{(4,2) \rightarrow (3,2)} &= - 4 a_{1,0} + 8 a_{1,1} \\ - P_{(4,3) \rightarrow (3,3)} &= - 4 a_{1,0} + 12 a_{1,1} \\ - P_{(4,4) \rightarrow (3,4)} &= - 4 a_{1,0} + 16 a_{1,1} \end{aligned} $	$ \begin{aligned} + P_{(0,0) \rightarrow (0,1)} &= 0 \\ + P_{(1,0) \rightarrow (1,1)} &= 0 \\ + P_{(2,0) \rightarrow (2,1)} &= 0 \\ + P_{(3,0) \rightarrow (3,1)} &= 0 \\ + P_{(4,0) \rightarrow (4,1)} &= 0 \\ - P_{(0,1) \rightarrow (0,0)} + P_{(0,1) \rightarrow (0,2)} &= + 1 b_{0,1} \\ - P_{(1,1) \rightarrow (1,0)} + P_{(1,1) \rightarrow (1,2)} &= + 1 b_{0,1} - 1 b_{1,1} \\ - P_{(2,1) \rightarrow (2,0)} + P_{(2,1) \rightarrow (2,2)} &= + 1 b_{0,1} - 2 b_{1,1} \\ - P_{(3,1) \rightarrow (3,0)} + P_{(3,1) \rightarrow (3,2)} &= + 1 b_{0,1} - 3 b_{1,1} \\ - P_{(4,1) \rightarrow (4,0)} + P_{(4,1) \rightarrow (4,2)} &= + 1 b_{0,1} - 4 b_{1,1} \\ - P_{(0,2) \rightarrow (0,1)} + P_{(0,2) \rightarrow (0,3)} &= + 2 b_{0,1} \\ - P_{(1,2) \rightarrow (1,1)} + P_{(1,2) \rightarrow (1,3)} &= + 2 b_{0,1} - 2 b_{1,1} \\ - P_{(2,2) \rightarrow (2,1)} + P_{(2,2) \rightarrow (2,3)} &= + 2 b_{0,1} - 4 b_{1,1} \\ - P_{(3,2) \rightarrow (3,1)} + P_{(3,2) \rightarrow (3,3)} &= + 2 b_{0,1} - 6 b_{1,1} \\ - P_{(4,2) \rightarrow (4,1)} + P_{(4,2) \rightarrow (4,3)} &= + 2 b_{0,1} - 8 b_{1,1} \\ - P_{(0,3) \rightarrow (0,2)} + P_{(0,3) \rightarrow (0,4)} &= + 3 b_{0,1} \\ - P_{(1,3) \rightarrow (1,2)} + P_{(1,3) \rightarrow (1,4)} &= + 3 b_{0,1} - 3 b_{1,1} \\ - P_{(2,3) \rightarrow (2,2)} + P_{(2,3) \rightarrow (2,4)} &= + 3 b_{0,1} - 6 b_{1,1} \\ - P_{(3,3) \rightarrow (3,2)} + P_{(3,3) \rightarrow (3,4)} &= + 3 b_{0,1} - 9 b_{1,1} \\ - P_{(4,3) \rightarrow (4,2)} + P_{(4,3) \rightarrow (4,4)} &= + 3 b_{0,1} - 12 b_{1,1} \\ - P_{(0,4) \rightarrow (0,3)} &= + 4 b_{0,1} \\ - P_{(1,4) \rightarrow (1,3)} &= + 4 b_{0,1} - 4 b_{1,1} \\ - P_{(2,4) \rightarrow (2,3)} &= + 4 b_{0,1} - 8 b_{1,1} \\ - P_{(3,4) \rightarrow (3,3)} &= + 4 b_{0,1} - 12 b_{1,1} \\ - P_{(4,4) \rightarrow (4,3)} &= + 4 b_{0,1} - 16 b_{1,1} \end{aligned} $

Tableau 3.2 Équations pour le modèle de Lotka Volterra classique.

La valeur $h = 1$ a été posée lors du développement de (3.21)

3.2.2 ANALYSE DU MODÈLE DE LOTKA VOLTERRA CLASSIQUE

Comme nous l'avons précisé précédemment, ce modèle est un cas particulier du système d'équations différentielles que nous venons de présenter. Les équations constitutives de ce modèle s'obtiennent en posant $a_{2,0} = b_{0,2} = 0$ dans (3.28). En vertu de ce rappel, les équations qui découlent de ce modèle sont donc :

$$\boxed{\begin{aligned}\frac{d\rho_1}{dt} &= -a_{1,0} \rho_1 + a_{1,1} \rho_1 \rho_2 \\ \frac{d\rho_2}{dt} &= b_{0,1} \rho_2 - b_{1,1} \rho_1 \rho_2\end{aligned}} \quad (3.35)$$

Dans ce contexte, le développement de (3.1) via (3.27) conduit respectivement aux équations qui sont associées à chacune des 2 espèces. Ces équations sont présentées explicitement au tableau 3.2. À propos de ce tableau 3.2, il est intéressant de remarquer qu'il constitue un cas particulier du tableau 3.1. En effet, en posant $a_{2,0} = b_{0,2} = 0$ dans le tableau 3.1 on obtient bel et bien le tableau 3.2. Ceci indique donc que l'extension d'un modèle par un autre ne semble pas apporter d'irrégularité au niveau des équations probabilistes. À présent, revenons à l'analyse du modèle de *Lotka Volterra classique*. La 20^{*ème*} équation du tableau 3.2 associée à l'espèce de type II mentionne que :

$$-P_{(0,4) \rightarrow (0,3)} = +4b_{0,1} \quad (3.36)$$

Mais comme $b_{0,1} > 0 \Rightarrow P_{(0,4) \rightarrow (0,3)} < 0$ ceci signifie qu'il y a contradiction. Cette contradiction signale donc qu'il ne sera pas possible de simuler sur un réseau le modèle de *Lotka Volterra classique* à l'aide d'un automate cellulaire probabiliste. Cette contradiction n'est pas une conséquence de la règle de correspondance que nous avons utilisée pour donner une solution unique au système d'équations puisque nous ne l'avons pas appliquée pour obtenir cette 20^{*ème*} équation. Elle sera donc toujours prise en défaut. Contrairement au modèle de *Malthus*, il est subséquemment plus difficile de relever ce qui physiquement rend impropre la simulation du modèle classique de *Lotka Volterra* sur un réseau cartésien.

3.2.3 ANALYSE DU MODÈLE DE LOTKA VOLTERRA ÉTENDU

Les équations constitutives de ce modèle ont déjà été obtenues et présentées au tableau (3.1). À présent, il s'agit de faire l'analyse de faisabilité de ce modèle. Une inspection rapide de ces équations nous amène à constater qu'il ne semble pas y avoir de contradiction apparente. En particulier, l'équation (3.31) du modèle de *Lotka Volterra classique* devient, avec l'extension de ce modèle :

$$-P_{(0,4) \rightarrow (0,3)} = +4b_{0,1} - 16b_{0,2} \quad (3.37)$$

Ici, comme $b_{0,2} > 0$ la contradiction qui nous avait conduit à rejeter la faisabilité du modèle de *Lotka Volterra classique*, n'apparaît plus explicitement dans le modèle de *Lotka Volterra étendu*. Toutefois, bien qu'il n'y ait pas de contradiction apparente, le problème de faisabilité n'est pas pour autant résolu. En effet, posons la question suivante : quelles sont les valeurs des coefficients $a_{1,0}, a_{1,1}, a_{2,0}, b_{0,1}, b_{1,1}$ et de $b_{0,2}$ qui demeurent consistantes avec les restrictions qu'apportent (3.10) ? Cette question souleve un problème de taille. Dans le cadre théorique des modèles à 1 espèce nous avons esquissé une méthode partielle pour pouvoir graduellement s'affranchir des contraintes qu'imposent (3.10) à (3.1). À présent, nous allons énoncer trois critères pratiques qui permettent de savoir si (3.1) peut être validé ou non. Sous forme algorithmique ces critères s'énoncent de la sorte :

- a) Pour la i -ème espèce, $1 \leq i \leq n$, établir la correspondance en signe pour les équations qui le requièrent. Si l'on trouve un $P_{\alpha \rightarrow \beta} < 0$, alors l'automate cellulaire ne peut être construit.
- b) Pour la i -ème espèce, $1 \leq i \leq n$, trouver l'ensemble des coefficients $A_i = \{ a_v^{(i)} \}$ tels que le champ de vecteurs $\mathbf{F}(\mathbf{p}(t))$ pointe vers l'intérieur¹⁰ de l'hypercube S^n lorsque $\mathbf{F}(\mathbf{p}(t))$ est évalué sur chacune des frontières de S^n . Si un $A_i = \emptyset$ alors l'automate cellulaire ne peut être construit.
- c) Si $P_{\alpha \rightarrow \beta} \geq 0$ et $A_i \neq \emptyset$ alors $\forall \alpha, \beta \in S^n \exists h > 0$ tel que $0 \leq P_{\alpha \rightarrow \beta} \leq 1$. Le supremum de « h » est obtenu par $h = 1/p_{sup}$ où p_{sup} est obtenu par :

$$p_{sup} = \underset{\substack{0 \leq \alpha \leq m \\ \alpha \neq \beta}}{\text{Max}} \left\{ \sum_{\beta=0}^m \frac{P_{\alpha \rightarrow \beta}}{h} \right\} \quad (3.38)$$

Expliquons ce que signifie chacun de ces critères. Le critère *a*) signifie que pour obtenir un automate cellulaire cohérent, toutes les équations correspondant aux états $1 \leq \alpha_i \leq m-1$ doivent être positives ou nulles. Les cas où $\alpha_i = 0$ ou $\alpha_i = m$ sont traités via le critère *b*). Ce critère *b*) signifie que pour obtenir une simulation qui soit compatible avec l'ordre m du réseau, chacune des densités ρ_i doit demeurer à tout instant dans l'intervalle $0 \leq \rho_i \leq m$. Cette condition est respectée si le champ de vecteurs $F(\rho(t))$ pointe vers l'intérieur de l'hypercube S^n lorsqu'il est évalué aux frontières de S^n . Il est à noter que $F(\rho(t))$ réfère à l'ensemble des équations qui sont induites par la loi d'action de masse et que S^n forme l'espace des phases. Cet espace des phases est obtenu en associant à l'axe i la densité ρ_i . Ainsi, l'évaluation du flux sur chacune des frontières de S^n permet de savoir si chacune des densités ρ_i peut demeurer confinée à l'intérieur de l'espace maximal qui lui est réservé c'est-à-dire S^n . Si la dynamique d'une espèce i est telle que la densité ρ_i se retrouve à l'extérieur de S^n alors l'ordre m du réseau sera débordé et le principe d'exclusion violé. Il faut donc que les mécanismes réactionnels envisagés génèrent une dynamique qui puisse être contenue par S^n . Du point de vue de la simulation, la vérification aux frontières de S^n permet donc de savoir si les équations correspondant au cas $\alpha_i = 0$ ou $\alpha_i = m$ peuvent être validées ou non. S'il est possible de réaliser les critères *a*) et *b*) on est alors assuré que $P_{\alpha \rightarrow \beta} \geq 0$. Le critère *c*) donne la façon la plus pratique de trouver un bon candidat pour le facteur d'échelle h pour que $P_{\alpha \rightarrow \beta} \leq 1$. Ainsi, sont levées les contraintes provenant de (3.10). Cette méthode proposée est générale car elle s'applique au cas où n espèces évolueraient sur un réseau cartésien. Avant de poursuivre l'analyse de faisabilité pour le modèle de *Lotka Volterra étendu* nous allons passer en revue les modèles classiques déjà présentés en utilisant ces derniers critères. Ce petit exercice aura le grand avantage d'illustrer en quoi les critères énoncés ci-haut sont de portée générale. Tout d'abord, en ce qui a trait au modèle de *Malthus* le système d'équations (3.12) admet une correspondance en signe. Le critère *a*) est donc respecté. L'évaluation de (3.11) à la frontière $\rho = m$ de S^1 conduit à l'égalité :

$$\frac{d\rho}{dt} = \alpha_1 m \quad (3.39)$$

Puisque $\alpha_1 > 0$ le flux est donc positif. La densité ρ va donc sortir de l'espace de phase S^1 . Le critère *b*) n'est donc respecté. Ce modèle n'est donc pas viable.

En ce qui a trait au modèle *logistique* le système d'équations (3.14) admet une correspondance en signe. Le critère *a*) est donc respecté. L'évaluation de (3.14) à la frontière $\rho = m$ de S^1 conduit à l'égalité :

$$\frac{d\rho}{dt} = \alpha_1 m - \alpha_2 m^2 = m (\alpha_1 - \alpha_2 m) \quad (3.40)$$

Puisque $a_1 > 0$ et $a_2 > 0$ le flux sera négatif $\Leftrightarrow a_2 \geq a_1 / m$. Le critère b) est respecté car $A_1 = \{ (a_1, a_2) \mid a_2 \geq a_1 / m \} \neq \emptyset$ constitue un ensemble non vide de coefficients pour lequel le flux pointe à l'intérieur de l'espace de phase S^1 . Le critère c) peut être alors respecté car il existe un facteur d'échelle h approprié (voir la figure 3.2). Ce modèle est donc viable.

En ce qui a trait au modèle de *Lotka Volterra classique* l'équation (3.36) n'admet pas une correspondance en signe. Ce modèle n'est donc pas viable. Ces quelques exemples devraient avoir illustré comment vérifier la faisabilité d'un mécanisme réactionnel quelconque.

À présent, nous allonsachever l'étude théorique des modèles classiques de croissance en appliquant ces critères au modèle de *Lotka Volterra étendu*. En ce qui a trait à ce modèle, nous avons déjà vu qu'une correspondance en signe est possible. Le critère a) est donc respecté. Pour vérifier si le critère b) peut être respecté, il faut sonder point par point les frontières de l'espace de phase S^2 . Cet espace est reproduit à la figure 3.3. D'autre part, le mécanisme en cause produit un champ vectoriel $\mathbf{F}(\rho(t)) = \mathbf{F}(\rho_1, \rho_2)$ donné par :

$$\mathbf{F}(\rho_1, \rho_2) = (-a_{1,0} \rho_1 + a_{1,1} \rho_1 \rho_2 - a_{2,0} \rho_1^2) \mathbf{i} + (b_{0,1} \rho_2 - b_{1,1} \rho_1 \rho_2 - b_{0,2} \rho_2^2) \mathbf{j} \quad (3.41)$$

Pour s'assurer que la trajectoire ne franchisse pas ces frontières, le flux de $\mathbf{F}(\rho(t))$ doit être mesuré sur les 4 frontières de l'espace de phase S^2 . Dans ce contexte, les conditions de flux à imposer doivent être les suivantes :

$$\mathbf{F}(0, \rho_2) \cdot \mathbf{i} \geq 0 \quad (3.42)$$

$$\mathbf{F}(\rho_1, 0) \cdot \mathbf{j} \geq 0 \quad (3.43)$$

$$\mathbf{F}(\rho_1, m) \cdot \mathbf{j} \leq 0 \quad (3.44)$$

$$\mathbf{F}(m, \rho_2) \cdot \mathbf{i} \leq 0 \quad (3.45)$$

En développant les équations de (3.42) à (3.45) on obtient respectivement :

$$\mathbf{F}(0, \rho_2) \cdot \mathbf{i} = 0 \mathbf{i} \cdot \mathbf{i} = 0 \quad (3.46)$$

$$\mathbf{F}(\rho_1, 0) \cdot \mathbf{j} = 0 \mathbf{j} \cdot \mathbf{j} = 0 \quad (3.47)$$

$$\mathbf{F}(\rho_1, m) \cdot \mathbf{j} = (b_{0,1} m - b_{1,1} \rho_1 m - b_{0,2} m^2) \mathbf{j} \cdot \mathbf{j} = m (b_{0,1} - b_{1,1} \rho_1 - b_{0,2} m) \quad (3.48)$$

$$\mathbf{F}(m, \rho_2) \cdot \mathbf{i} = (-a_{1,0} m + a_{1,1} m \rho_2 - a_{2,0} m^2) \mathbf{i} \cdot \mathbf{i} = m (-a_{1,0} + a_{1,1} \rho_2 - a_{2,0} m) \quad (3.49)$$

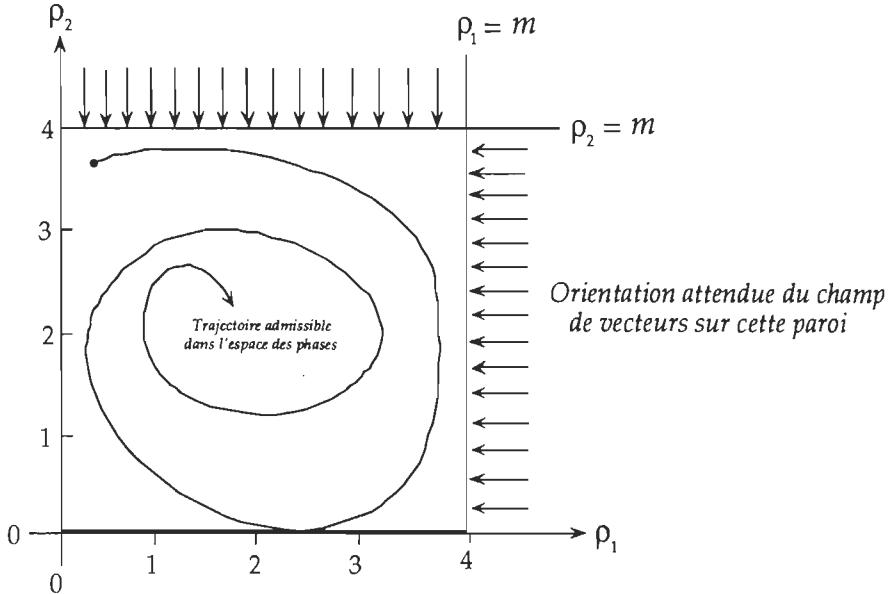


Fig 3.3 Espace des phases S^2 et ses frontières

Les équations (3.42) et (3.43) sont satisfaites en regard des résultats (3.46) et (3.47) et ce, quelle que soit la valeur des coefficients $a_{1,0}, a_{1,1}, a_{2,0}, b_{0,1}, b_{1,1}$ et $b_{0,2}$. Toutefois, les équations (3.44) et (3.45) demandent une étude un peu plus minutieuse. Tout d'abord, notons que (3.48) est la conséquence de (3.44). Il faut donc résoudre l'inégalité :

$$\begin{aligned}
 & m(b_{0,1} - b_{1,1}\rho_1 - b_{0,2}m) \leq 0 \\
 \Leftrightarrow & (b_{0,1} - b_{1,1}\rho_1 - b_{0,2}m) \leq 0 \\
 \Leftrightarrow & b_{0,1} \leq b_{1,1}\rho_1 + b_{0,2}m
 \end{aligned} \tag{3.50}$$

La densité ρ_1 apparaît encore dans (3.50). Il faut donc chercher quelle valeur de ρ_1 nous assure que (3.50) sera satisfaite peu importe la valeur de ρ_1 . La valeur de ρ_1 qui nous apporte cette garantie est $\rho_1 = 0$. En prenant en compte l'ordre du réseau cartésien, on obtient donc :

$$b_{0,2} \geq b_{0,1} / 4 \tag{3.51}$$

De la même manière, puisque (3.49) est la conséquence de (3.45), il faut donc résoudre l'inégalité :

$$\begin{aligned}
 & m(-a_{1,0} + a_{1,1}\rho_2 - a_{2,0}m) \leq 0 \\
 \Leftrightarrow & (-a_{1,0} + a_{1,1}\rho_2 - a_{2,0}m) \leq 0 \\
 \Leftrightarrow & a_{1,1}\rho_2 \leq a_{1,0} + a_{2,0}m
 \end{aligned} \tag{3.52}$$

La densité ρ_2 apparaît encore dans (3.52). Cette fois-ci, il faut chercher quelle valeur de ρ_2 nous assure que (3.52) sera satisfaite à tout instant. La valeur $\rho_2 = m$ apporte cette garantie. En prenant en compte l'ordre du réseau cartésien, on obtient donc :

$$a_{1,1} \leq a_{1,0} / 4 + a_{2,0} \quad (3.53)$$

En résumé, les ensembles A_1 et A_2 des coefficients admissibles sont les suivants :

$$A_1 = \{ (a_{1,0}, a_{1,1}, a_{2,0}) \mid a_{1,1} \leq a_{1,0} / 4 + a_{2,0}, \forall a_{1,0} > 0, \forall a_{2,0} > 0 \} \quad (3.54)$$

$$A_2 = \{ (b_{0,1}, b_{1,1}, b_{0,2}) \mid b_{0,2} \geq b_{0,1} / 4, \forall b_{0,1} > 0, \forall b_{1,1} > 0 \} \quad (3.55)$$

Puisque les critères *a*) et *b*) sont respectés alors le critère *c*) peut aussi l'être. Il existe donc un facteur d'échelle h qui contracte la trajectoire dans l'espace de phase S^2 . On peut donc construire un automate cellulaire probabiliste visant à simuler le modèle de *Lotka Volterra étendu*.

Chapitre IV

SIMULATIONS DES MODÈLES CLASSIQUES DE CROISSANCE SUR RÉSEAU

Dans ce chapitre, une présentation algorithmique du programme informatique est donnée. De plus, des exemples de calculs qui doivent être faits avant la mise en exécution de ce programme, sont effectués. Les solutions théoriques aux modèles de croissance classiques sont par la suite présentées. La fin de ce chapitre fournit les résultats qui sont reliés aux simulations de ces modèles. En dernier lieu, une comparaison entre les résultats théoriques et expérimentaux est entreprise.

4.1 PRÉSENTATION DU PROGRAMME INFORMATIQUE

Nous sommes sur le point de mettre en marche le programme informatique « écosystème » dans le cadre où 1 et 2 espèces évoluent sur un réseau cartésien. Toutefois, avant d'ouvrir le volet expérimental de ce mémoire, il serait souhaitable d'expliquer brièvement le fonctionnement du programme « écosystème » présenté à l'annexe B. Seule une description algorithmique des tâches qu'effectue ce programme sera ici présentée. La description algorithmique du programme informatique est présentée étape par étape au tableau 4.1. Cet algorithme indique la procédure suivie pour simuler un mécanisme réactionnel donné. D'une certaine manière cet algorithme montre comment le travail effectué au cours de ce mémoire lui est intimement relié. Voyons ces liens plus en détail. Les étapes 1) et 2) se rapportent aux chapitres II, III ainsi qu'à l'annexe A. Les étapes 3) à 7) se rapportent essentiellement au chapitre I. De fait, c'est à travers ce chapitre I que nous avons eu l'occasion de voir comment, en terme d'implantation, se réalise l'étape 4). Pour compléter la compréhension du fonctionnement du programme «écosystème», quelques calculs préliminaires seront faits avant de passer au volet expérimental.

4.1.1 EXEMPLES DE CALCULS PRÉLIMINAIRES

À la lecture de l'étape 1) du tableau 4.1, la façon d'obtenir les ensembles A_i et le facteur d'échelle de temps h laisse planer une ambiguïté. En effet, l'étape 1) de cet algorithme présuppose la connaissance des ensembles A_i , des coefficients admissibles et elle présuppose aussi la connaissance de l'intervalle du facteur d'échelle de temps h . L'ambiguïté est la suivante : comment effectuer le calcul des ensembles A_i et de cet intervalle correspondant au facteur d'échelle de temps h ? Pour situer l'utilisateur potentiel de ce programme, il faut indiquer que ce programme est partiellement en mesure d'établir ces calculs préliminaires. Malheureusement les plus difficiles relèvent entièrement de l'utilisateur¹¹. Nous allons voir quels types de calculs doivent être faits avant la mise en exécution du programme informatique. Dans le cadre du chapitre III, nous avons su obtenir, tous les ensembles A_i pertinents aux mécanismes étudiés. En ce sens, une partie de l'étape 1) est déjà complétée. Seul le calcul de l'intervalle de temps h n'a pas été effectué. Pour clarifier comment obtenir l'intervalle $0 < h \leq 1 / p_{sup}$ qui soit consistant avec l'énoncé (3.38), nous allons effectuer le calcul de cet intervalle correspondant à chacun des mécanismes étudiés. Par la suite nous allons montrer, à partir d'une valeur de h choisie dans cet intervalle, comment calculer une matrice de transition $P_{\alpha \rightarrow \beta}$ compatible avec le ou les ensembles A_i . Nous pourrons mieux voir alors l'effet de ce facteur h sur la construction de la matrice de transition $P_{\alpha \rightarrow \beta}$. Ces exemples de calculs devraient en l'occurrence clarifier les concepts rencontrés au cours du chapitre III. Suite à ces exemples de calculs, nous ferons la description des types d'expériences qui ont été réalisées sur le réseau cartésien pour simuler les mécanismes étudiés.

- Étape 1) Dans le bloc de déclaration des constantes, affecter des coefficients $a_v^{(i)}$ en accord avec les ensembles A_i .
Dans ce bloc de déclaration fixer un temps d'échelle h inclus dans l'intervalle $0 < h \leq 1 / p_{sup}$.
- Étape 2) Effectuer le calcul de la matrice de transition $P_{\alpha \rightarrow \beta}$.
- Étape 3) Sur le réseau cartésien, générer de façon *homogène* les densités initiales associées à chaque espèce.
- Étape 4) Sur ce réseau cartésien, inspecter une cellule du réseau pour en découvrir son état α .
- Étape 5) Pour cette cellule du réseau, appliquer les règles décisionnelles probabilistes associées à cet état α .
Faire passer cette cellule de l'état α à l'état β si cela s'impose.
- Étape 6) Les étapes 4) et 5) sont répétées pour chaque cellule du réseau de dimension : « dimension x dimension ».
- Étape 7) Effectuer une rotation rigide en chaque cellule du réseau.
- Étape 8) Effectuer la propagation des espèces dans la nouvelle direction vers laquelle elle se dirige.
- Étape 9) Répéter les étapes 4) à 8) un nombre de fois égal à « temps total ».
- Étape 10) Via un graphique, illustrer l'évolution qu'a subie chaque espèce dans le laps de temps discret « temps total ».
- Étape 11) Via ce graphique indiquer les valeurs de stabilisation que le système dynamique atteint.

Tableau 4.1 Algorithme du programme informatique « écosystème »

Nous avons vu au chapitre III que seul le modèle *logistique* et le modèle de *Lotka Volterra étendu* peuvent être simulés sur un réseau cartésien via un automate cellulaire probabiliste. Les exemples de calculs préliminaires seront donc établis conformément à ces deux modèles. Débutons tout d'abord ces calculs pour le modèle *logistique*. Ce modèle suppose que nous sommes en présence d'une seule espèce. La matrice de transition à une espèce $P_{\alpha \rightarrow \beta}$ prend la forme :

$$P_{\alpha \rightarrow \beta} = \begin{pmatrix} P_{0 \rightarrow 0} & P_{0 \rightarrow 1} & P_{0 \rightarrow 2} & P_{0 \rightarrow 3} & P_{0 \rightarrow 4} \\ P_{1 \rightarrow 0} & P_{1 \rightarrow 1} & P_{1 \rightarrow 2} & P_{1 \rightarrow 3} & P_{1 \rightarrow 4} \\ P_{2 \rightarrow 0} & P_{2 \rightarrow 1} & P_{2 \rightarrow 2} & P_{2 \rightarrow 3} & P_{2 \rightarrow 4} \\ P_{3 \rightarrow 0} & P_{3 \rightarrow 1} & P_{3 \rightarrow 2} & P_{3 \rightarrow 3} & P_{3 \rightarrow 4} \\ P_{4 \rightarrow 0} & P_{4 \rightarrow 1} & P_{4 \rightarrow 2} & P_{4 \rightarrow 3} & P_{4 \rightarrow 4} \end{pmatrix} \quad (4.1)$$

L'obtention de cette matrice, via un cas particulier, nous permettra de commenter comment l'automate cellulaire réussit à modifier ses états du point de vue informatique. Nous savons que le modèle *logistique* conduit à l'équation différentielle :

$$\frac{dp}{dt} = a_1 p - a_2 p^2 \quad (4.2)$$

L'analyse de (4.2) via (3.40) a montré que l'ensemble A_1 des coefficients admissibles pour le modèle *logistique* est :

$$A_1 = \{ (a_1, a_2) \mid a_2 \geq a_1/m, \forall a_1 > 0 \} \quad (4.3)$$

Pour construire un ensemble A_1 précis, choisissons $a_1 = 8$. Dans le contexte d'une simulation sur un réseau cartésien, ce choix de a_1 conduit (4.3) à l'inégalité :

$$a_2 \geq 2 \quad (4.4)$$

Pour établir un calcul particulier, choisissons à présent $a_2 = 3$. Nous avons vu à la section 3.1.5 que cette liberté apparente concernant le choix du couple (a_1, a_2) doit être contrebalancée par un choix judicieux du facteur d'échelle de temps h . En effet, nous avons établi que ce facteur h doit être tel qu'il appartienne à l'intervalle : $0 < h \leq 1/p_{sup}$. Dans le cas à 1 espèce le résultat (3.21) peut être utilisé. Suivant ce résultat on a :

$$p_{sup} = \max_{\substack{0 \leq \alpha \leq m \\ \alpha \neq \beta}} \left\{ \sum_{\beta=0}^m \frac{P_{\alpha \rightarrow \beta}}{h} \right\} \quad (4.5)$$

À ce stade, il est bon d'introduire les équations pour le modèle *logistique*. Les résultats (3.18) et (3.19) nous indiquent que ces équations sont:

$$\begin{aligned} a) P_{1 \rightarrow 2} &= h a_1 & c) P_{2 \rightarrow 3} &= h 2 a_1 & e) P_{3 \rightarrow 4} &= h 3 a_1 & g) P_{0 \rightarrow 1} &= 0 \\ b) P_{1 \rightarrow 0} &= 0 & d) P_{2 \rightarrow 1} &= h \frac{8}{3} a_2 & f) P_{3 \rightarrow 2} &= h 8 a_2 & h) P_{4 \rightarrow 3} &= h (-4 a_1 + 16 a_2) \end{aligned} \quad (4.6)$$

Le développement de (4.5) conjointement avec (4.6) conduit à :

$$p_{sup} = \text{Max} \{ 0, 0 + a_1, \frac{8}{3} a_2 + 2 a_1, 8 a_2 + 3 a_1, -4 a_1 + 16 a_2 \} \quad (4.7)$$

Puisque nous travaillons avec les valeurs $a_1 = 8$ et $a_2 = 3$ on obtient de (4.7) :

$$p_{sup} = \text{Max} \{ 0, 8, 8 + 16, 24 + 24, -32 + 48 \} \quad (4.8)$$

$$p_{sup} = \text{Max} \{ 0, 8, 24, 48, 16 \} \quad (4.9)$$

$$p_{sup} = 48 \quad (4.10)$$

Un intervalle approprié pour le facteur d'échelle h est donc :

$$0 < h \leq 1 / 48 \quad (4.11)$$

Cet exemple de calcul montre en fait que la valeur de p_{sup} appropriée doit être plus petite que la valeur maximale de la somme de chacune des lignes de la matrice de transition, lorsque l'on exclut de chacune des lignes l'élément de transition $P_{\alpha \rightarrow \alpha}$ qui lui correspond. Par ailleurs, il est à noter que la simplicité d'écriture que propose (4.8), et par extension (3.38), ne semble pas être utilisé dans le cadre de la littérature scientifique¹⁻¹⁵. Nous verrons ultérieurement que l'utilisation de (3.38), dans le cadre à plusieurs espèces permet en principe un calcul simple de la valeur de p_{sup} appropriée. Pour déterminer le système d'équations (4.6) choisissons

arbitrairement $h = 1/72$. Conjointement aux valeurs de $\alpha_1 = 8$ et $\alpha_2 = 3$, cette valeur de h confère à (4.6) la solution suivante :

$$\begin{array}{llll} a) P_{1 \rightarrow 2} = 1/9 & c) P_{2 \rightarrow 3} = 2/9 & e) P_{3 \rightarrow 4} = 1/3 & g) P_{0 \rightarrow 1} = 0 \\ b) P_{1 \rightarrow 0} = 0 & d) P_{2 \rightarrow 1} = 1/9 & f) P_{3 \rightarrow 2} = 1/3 & h) P_{4 \rightarrow 3} = 2/9 \end{array} \quad (4.12)$$

La matrice de transition $P_{\alpha \rightarrow \beta}$ correspondante est donc :

$$P_{\alpha \rightarrow \beta} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 8/9 & 1/9 & 0 & 0 \\ 0 & 1/9 & 6/9 & 2/9 & 0 \\ 0 & 0 & 1/3 & 1/3 & 1/3 \\ 0 & 0 & 0 & 2/9 & 7/9 \end{pmatrix} \quad (4.13)$$

Pour commenter comment l'automate cellulaire prend une décision, supposons qu'une cellule soit dans l'état $\alpha = 1$. La matrice de transition révèle les faits suivants : dans 0% des cas il faudra que la cellule passe de l'état $\alpha = 1$ à l'état $\alpha = 0$, dans $8/9 = 0.88 = 88.88\%$ des cas il faudra que la cellule passe de l'état $\alpha = 1$ à l'état $\alpha = 1$, tandis que dans $1/9 = 0.11 = 11.11\%$ des cas il faudra que la cellule passe de l'état $\alpha = 1$ à l'état $\alpha = 2$. Ainsi doivent être comprises les règles décisionnelles probabilistes attachées à chacune des cellules du réseau. Nous avions énoncé au chapitre I qu'une cellule munie de règles d'actions porte le nom d'*automate cellulaire*. À présent, nous avons éclairci le sens de cette définition dans un contexte probabiliste.

Nous venons de voir par un exemple simple, comment un choix approprié du facteur h rend la matrice de transition $P_{\alpha \rightarrow \beta}$ bien définie. Toutefois le rôle principal que doit tenir ce facteur est de rendre la plus exacte possible l'hypothèse des interactions faibles. En effet, le résultat (1.7), concernant la distribution statistique du nombre d'êtres présents à une cellule a été obtenue dans un cadre où les collisions sont purement élastiques. Ces collisions élastiques ne permettent que l'utilisation des opérateurs **R** et **P**. Par la suite, nous avons introduit la règle de collision **C** sous l'hypothèse que ces collisions allaient constituer des événements rares¹⁻¹¹. Pour illustrer comment satisfaire cette dernière hypothèse, reprenons l'exemple en cours. Tout d'abord notons que la matrice $P_{\alpha \rightarrow \beta}$ donnée en (4.13) contient des probabilités de transitions qui s'apparentent peu à la notion d'événements rares. En guise d'exemple, la transition $\alpha = 3$ vers l'état $\alpha = 2$ doit s'effectuer une fois sur trois. Ce rythme de changement d'état semble trop élevé pour qu'il puisse entrer sous la rubrique « événements rares ». La façon de respecter cette dernière hypothèse se fera en

choisisant une valeur de h très petite. Par exemple, le choix de $h = 1 / 7200$ conduit à la matrice de transition suivante :

$$P_{\alpha \rightarrow \beta} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 899/900 & 1/900 & 0 & 0 \\ 0 & 1/900 & 897/900 & 2/900 & 0 \\ 0 & 0 & 1/300 & 298/300 & 1/300 \\ 0 & 0 & 0 & 2/900 & 898/900 \end{pmatrix} \quad (4.14)$$

Ce dernier calcul illustre bien en fait que lorsque h tend vers zéro, alors tous les éléments de transitions $P_{\alpha \rightarrow \alpha}$ tendent vers un⁹. Cette dernière limite signale que pour approcher le plus possible l'hypothèse selon laquelle les collisions constituent des événements rares, les éléments de transitions $P_{\alpha \rightarrow \alpha}$ doivent être tels que : $P_{\alpha \rightarrow \alpha} \approx 1$. Et effectivement, plus ces éléments de transitions $P_{\alpha \rightarrow \alpha}$ seront près de un, plus les changements d'états, autrement dit les collisions, seront rares. De façon stricte, si les éléments de transitions $P_{\alpha \rightarrow \alpha} = 1$, ceci revient à dire que les seules règles d'évolution **R** et **P** sont considérées. Mais comme nous avons introduit la règle de collision **C** (naissances et mortalités) il faudra donc être conséquent en choisissant notre échelle de temps h suffisamment petite. Il reste toutefois à déterminer ce que signifient les termes « suffisamment petite ». À cet égard, nous aurons l'occasion de revenir sur le sens de ces mots à travers le volet expérimental. Ce bref exemple numérique devrait avoir clarifié le double rôle du temps d'échelle dans le cadre de la simulation d'une dynamique de population sur un réseau cartésien.

À présent, effectuons un exemple de calcul préliminaire de cette valeur de h pour le modèle de *Lotka Volterra étendu*. Nous avons vu que ce modèle conduit au système d'équations différentielles :

$$\begin{aligned} \frac{d\rho_1}{dt} &= -a_{1,0} \rho_1 + a_{1,1} \rho_1 \rho_2 - a_{2,0} \rho_1^2 \\ \frac{d\rho_2}{dt} &= b_{0,1} \rho_2 - b_{1,1} \rho_1 \rho_2 - b_{0,2} \rho_2^2 \end{aligned} \quad (4.15)$$

L'analyse de (4.15) via les équations (3.46) à (3.49) a démontré que les ensembles A_1 et A_2 des coefficients admissibles pour ce modèle sont respectivement :

$$A_1 = \{ (a_{1,0}, a_{1,1}, a_{2,0}) \mid a_{1,1} \leq a_{1,0} / 4 + a_{2,0}, \forall a_{1,0} > 0, \forall a_{2,0} > 0 \} \quad (4.16)$$

$$A_2 = \{ (b_{0,1}, b_{1,1}, b_{0,2}) \mid b_{0,2} \geq b_{0,1} / 4, \forall b_{0,1} > 0, \forall b_{1,1} > 0 \} \quad (4.17)$$

Dans le but de construire ces ensembles A_1 et A_2 choisissons $a_{1,0} = 0.8$, $a_{2,0} = 0.3$, $b_{0,1} = 1.6$, $b_{1,1} = 0.2$.

Dans le contexte d'une simulation sur un réseau cartésien, les résultats (4.16) et (4.17) conduisent aux deux inégalités :

$$a_{1,1} \leq 0.5 \quad (4.18)$$

$$b_{0,2} \geq 0.4 \quad (4.19)$$

Dans le but d'entreprendre un calcul particulier, choisissons arbitrairement $a_{1,1} = 0.2$ et $b_{0,2} = 0.5$.

À présent, nous devons établir un intervalle pour l'échelle de temps d'échelle h . Le résultat (3.38) appliquée dans le cadre à 2 espèces conduit à :

$$p_{sup} = \underset{\substack{(0,0) \leq (\alpha_1, \alpha_2) \leq (4,4) \\ (\alpha_1, \alpha_2) \neq (\beta_1, \beta_2)}}{\text{Max}} \left\{ \sum_{\beta_1=0}^4 \sum_{\beta_2=0}^4 \frac{P_{(\alpha_1, \alpha_2) \rightarrow (\beta_1, \beta_2)}}{h} \right\} \quad (4.20)$$

Pour le modèle de *Lotka Volterra étendu*, le développement de (4.20) est possible par l'utilisation des équations du tableau 3.1. Rappelons que le membre de droite de chacune de ces équations est en réalité multiplié par h . Selon (4.20), la recherche du maximum porte sur un ensemble de 25 éléments puisque le couple (α_1, α_2) est susceptible de prendre 25 valeurs distinctes. En principe, l'utilisation du tableau 3.1 permet le calcul de ce maximum. Toutefois, ce calcul est très long à réaliser à la main. L'utilisation d'une simple procédure informatique permet de déterminer aisément la valeur de ce maximum. Malgré tout nous allons donner quelques éléments types qui forment l'ensemble recherché. Ainsi, l'utilisation du tableau 3.1 conjointement avec (4.20) permet d'exhiber partiellement (4.20) de la sorte :

$$p_{sup} = \frac{1}{h} \text{Max} \{ P_{(0,0) \rightarrow (1,0)} + P_{(0,0) \rightarrow (0,1)}, \dots, P_{(2,1) \rightarrow (1,1)} + P_{(2,1) \rightarrow (3,1)} + P_{(2,1) \rightarrow (2,0)} + P_{(2,1) \rightarrow (2,2)}, \dots \} \quad (4.21)$$

Par la règle de correspondance en signe on peut écrire du tableau 3.1 et de (4.21) :

$$p_{sup} = \text{Max} \{ 0 + 0, \dots, 2a_{1,0} + 8/3 a_{2,0} + a_{1,1} + 2b_{1,1} + 1b_{0,1}, \dots \} \quad (4.22)$$

Puisque nous travaillons avec les triplets :

$$(a_{1,0}, a_{1,1}, a_{2,0}) = (0.8, 0.2, 0.3) \quad (4.23)$$

$$(b_{0,1}, b_{1,1}, b_{0,2}) = (1.6, 0.2, 0.5) \quad (4.24)$$

on obtient alors de (4.23) et (4.24) :

$$p_{sup} = \text{Max} \{ 0, \dots, 4.6, \dots \} \quad (4.25)$$

Ce calcul partiel n'indique pas le maximum; il ne sert qu'à illustrer la méthode. Toutefois, à l'aide d'une procédure informatique on montre en réalité que $p_{sup} = 17.2$. Un intervalle approprié pour le facteur d'échelle h est donc :

$$0 < h \leq 1 / 17.2 \quad (4.26)$$

Pour effectuer le calcul de la matrice de transition $P_{\alpha \rightarrow \beta}$, nous devrions, à ce stade, utiliser de nouveau les équations présentées au tableau 3.1. Comme la dimension de la matrice de transition $P_{\alpha \rightarrow \beta}$ est de 25x25, nous allons donc nous exempter de présenter cette matrice. Une simple indication sur la façon de compléter le calcul numérique devrait suffire. D'une part, on pourrait fixer arbitrairement la valeur de h à 1/20. D'autre part, nous avons vu que le calcul des inconnues $P_{\alpha \rightarrow \beta}$ fait appel à la règle de correspondance en signe. On trouvera au tableau 3.1 toutes les informations nécessaires concernant l'utilisation des égalités (3.29) et (3.30). Au stade de ces exemples de nature numérique il devrait être clair que l'utilisation d'un programme informatique s'avère nécessaire pour effectuer, en général, le calcul de la matrice de transition $P_{\alpha \rightarrow \beta}$.

4.2 ANALYSE DE STABILITÉ DES MODÈLES CLASSIQUES

Nous nous dirigeons graduellement vers la simulation, sur réseau cartésien, du modèle *logistique* et du modèle de *Lotka Volterra étendu*. Pour ce faire il est souhaitable de dégager les états stables que ces systèmes dynamiques sont susceptibles d'atteindre. Cette analyse est importante puisqu'elle nous servira de critère pour évaluer si l'état autour duquel les densités se stabilisent sur le réseau sont conformes à celle que la théorie prévoit. De plus ce critère simple sera utilisé pour vérifier si la simulation respecte bien la dynamique de ces modèles. Nous allons débuter cette analyse avec le modèle *logistique* et poursuivre avec le modèle de *Lotka Volterra étendu*.

Tout d'abord précisons que la classification des états d'un système dynamique s'effectue par la recherche des points d'équilibres de ce système. En général, pour un système d'équations linéaires d'ordre un, les points d'équilibre s'obtiennent par la résolution de :

$$\frac{d\rho}{dt} = 0 \quad (4.27)$$

La recherche des points d'équilibre correspondants au modèle *logistique* se traduit donc par :

$$a_1 \rho - a_2 \rho^2 = 0 \quad (4.28)$$

$$\Rightarrow \rho (a_1 - a_2 \rho) = 0 \quad (4.29)$$

$$\Rightarrow \rho = 0 \text{ ou } \rho = a_1 / a_2 \quad (4.30)$$

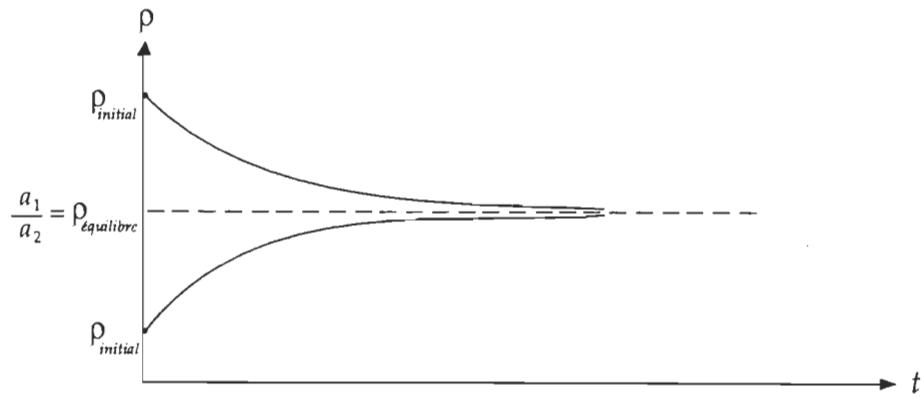


Fig 4.1 Dynamique caractéristique du modèle *logistique*

On montre facilement que le point d'équilibre $p = 0$ correspond à un état instable tandis que le point d'équilibre $p = a_1 / a_2$ correspond à un état stable. Cette situation conduit à penser qu'une simulation sur réseau cartésien du modèle logistique devrait systématiquement conduire à l'état d'équilibre stable $p = a_1 / a_2$. La figure 4.1 illustre la dynamique qui est caractéristique^{32,33} du modèle *logistique*. On y voit en particulier que quelle que soit la condition initiale (autre que $p = 0$) le système dynamique tend vers une densité d'équilibre $p = a_1 / a_2$. Toutefois, nous savons à présent que les valeurs de a_1 et a_2 sont restreintes, en vertu de (4.3), par :

$$a_2 \geq a_1 / 4 \quad (4.31)$$

Lors des simulations, nous allons donc être confronté à choisir un ensemble de coefficients a_1 et a_2 pertinents. Le tableau 4.2 illustre ces choix de a_1 et a_2 .

Valeur de a_1	Valeur de a_2	$p_{équilibre}$
3	3	1
3	2.8	1.0714
3	2.6	1.1538
3	2.4	1.25
3	2.2	1.3636
3	2.0	1.5
3	1.8	1.6666
3	1.6	1.875
3	1.4	2.1428
3	1.2	2.5
3	1	3

Tableau 4.2 Choix de coefficients compatibles avec (4.32)

Nous référerons plus explicitement au tableau 4.2 lors de la description des expériences qui ont été réalisées. Ceci termine donc l'analyse de stabilité du modèle *logistique*.

La recherche des points d'équilibres correspondant au modèle de *Lotka Volterra étendu* s'effectue, en vertu de (4.27), par la résolution du système :

$$\begin{aligned} -a_{1,0} \rho_1 + a_{1,1} \rho_1 \rho_2 - a_{2,0} \rho_1^2 &= 0 \\ b_{0,1} \rho_2 - b_{1,1} \rho_1 \rho_2 - b_{0,2} \rho_2^2 &= 0 \end{aligned} \quad (4.32)$$

Il est facile de montrer que le système (4.32) peut admettre les 4 points d'équilibres suivants :

$$(\rho_1, \rho_2) = (0, 0) \quad (4.33)$$

$$(\rho_1, \rho_2) = (-a_{1,0} / a_{2,0}, 0) \quad (4.34)$$

$$(\rho_1, \rho_2) = (0, b_{0,1} / b_{0,2}) \quad (4.35)$$

$$(\rho_1, \rho_2) = \left(\frac{b_{0,1} a_{1,1} - b_{0,2} a_{1,0}}{b_{1,1} a_{1,1} + b_{0,2} a_{2,0}}, \frac{b_{1,1} a_{1,0} + b_{0,1} a_{2,0}}{b_{1,1} a_{1,1} + b_{0,2} a_{2,0}} \right) \quad (4.36)$$

Parmi les points d'équilibres (4.33) à (4.36) on peut montrer que seul le point $(\rho_1, \rho_2) = (0, 0)$ correspond à un d'équilibre instable. Cet état n'est donc pas susceptible d'être atteint lors de la simulation sur réseau. Bien que le point donné par (4.34) corresponde à un équilibre stable, ce point se situe à l'extérieur de l'espace de phase S^2 . L'état stable qui lui correspond ne sera donc pas susceptible d'être atteint lui non plus. Seul les points d'équilibre stable donnés en (4.35) et (4.36) pourront être observés lors des simulations sur le réseau. Toutefois, nous attacherons une plus grande importance au point d'équilibre (4.36). L'état stable qui lui correspond est de loin le plus intéressant puisqu'il suppose qu'aucune des espèces n'est disparue. Les calculs théoriques des densités d'équilibre seront donc essentiellement établis à partir du résultat (4.36). S'il a été simple de trouver les coefficients pertinents pour simuler le modèle *logistique*, le problème du choix de « bons » coefficients est tout autre pour le modèle de *Lotka Volterra étendu*. En effet, le type de dynamique qu'est susceptible de faire naître ce modèle est plus étendu en possibilités que celle du modèle *logistique*. Nous avons d'ailleurs présenté à la figure 4.1 toutes les caractéristiques de cette dynamique. Une présentation aussi simple n'est toutefois pas possible pour le *Lotka Volterra étendu*. Malgré cela, la figure 4.2 présente trois trajectoires types qui sont susceptibles d'être observées dans l'espace de phase S^2 .

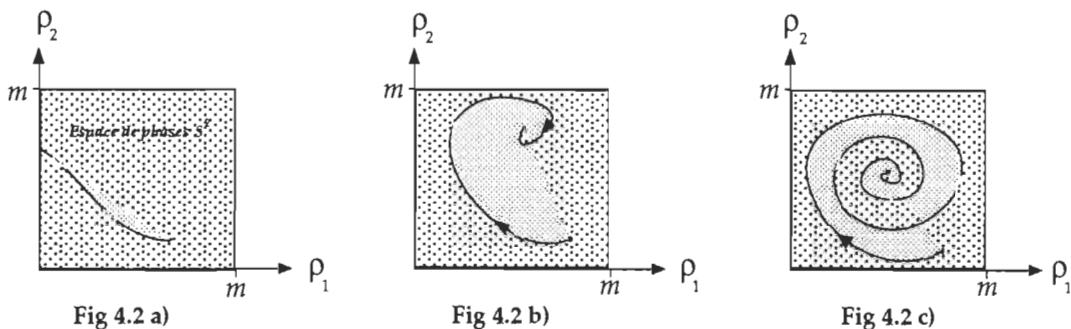


Fig 4.2 Trois trajectoires types du modèle de Lotka Voltera étendu dans l'espace de phase S^2

La trajectoire de la figure 4.2 a) correspond à une dynamique qui tend vers l'état stable dont la coordonnée du point d'équilibre est donnée par (4.35). Ce type d'évolution est caractérisé par la disparition de l'espèce « prédateur ». Bien que ce type de dynamique ne soit pas d'une grande richesse nous tenterons tout de même de vérifier si elle peut être reproduite sur réseau. La trajectoire de la figure 4.2 b) correspond, quant à elle, à une dynamique qui tend vers l'état stable dont la coordonnée du point d'équilibre est donnée par (4.36). Ce type de dynamique est déjà plus riche que la précédente puisque la notion de cycle commence à s'y installer. Toutefois, la figure 4.2 b) illustre que ces cycles sont rapidement amortis. La situation se redresse considérablement à la figure 4.2 c). En effet, cette figure met en évidence les nombreux cycles de croissance des 2 espèces en jeu. Comme pour le cas de la figure 4.2 b), le point d'équilibre stable de ce type de dynamique est donné par (4.36). À proprement parler, seul le nombre de cycles distingue les dynamiques des figures 4.2 b) et 4.2c). Des trois types de dynamique dont nous venons de discuter, celle indiquée à la figure 4.2c) semble donc celle qui offre le plus de richesse. En ce sens, il serait donc particulièrement intéressant d'y apporter une plus grande importance lors des simulations. Il est important de signaler que la modification d'un type de dynamique s'obtient en modifiant la ou les valeurs des coefficients du système d'équations différentielles (4.15). Or, puisque les coefficients de (4.15) doivent satisfaire les contraintes (4.16) et (4.17) il n'est donc pas encore assuré que nous puissions obtenir les trois types de trajectoires que nous venons de présenter, via une simulation sur réseau. L'obstacle qui peut donc se dresser devant nous est le suivant : est-il possible que les contraintes (4.16) et (4.17) bloquent l'accès à certains types de dynamique ? De fait, il semble malheureusement que les contraintes (4.16) et (4.17) nous interdisent d'obtenir le type de trajectoire illustrée à la figure 4.2c). Pour se convaincre que tel est le cas, il s'agit d'affecter les triplets ($a_{1,0}, a_{1,1}, a_{2,0}$) et ($b_{0,1}, b_{1,1}, b_{0,2}$) conformément à (4.16) et (4.17) puis de résoudre le système d'équation différentielle (4.15) par la méthode numérique de Runge Kutta^{32,33}. Ce travail a été effectué pour un très grand nombre de triplets,

mais ces affectations ne semblent pas en mesure de reproduire, à une même échelle de grandeur, les nombreux cycles de la figure 4.2 c). Cette dernière affirmation mérite explication. Il est possible de trouver des triplets coefficients $(a_{1,0}, a_{1,1}, a_{2,0})$ et $(b_{0,1}, b_{1,1}, b_{0,2})$ qui donnent lieu à des trajectoires de la même classe que celles de la figure 4.2 c), à cette distinction près que pour observer 3 ou 4 cycles, l'échelle des densités doit être ramenée à un dixième de l'unité. En réalité, ceci signifie qu'il est en principe possible d'observer autant de cycles que l'on veut, quitte à zoomer convenablement sur le point d'équilibre et son voisinage. Du point de vue des simulations le problème qui se pose alors est le suivant : est-ce qu'un réseau possède un pouvoir de résolution assez grand pour séparer des cycles lorsqu'ils se produisent à très petite échelle ? Nous allons tenter de voir ce qu'il en est dans le cadre expérimental. Toutefois notons qu'il est fort possible que nous puissions mettre en relief les nombreux cycles de croissance auxquels peut donner lieu le modèle de *Lotka Volterra étendu* ; cette possibilité dépend de la réponse que l'on peut donner à la question soulevée. Quoi qu'il en soit pour l'instant, il semble malgré tout que les conditions de flux que l'on doit imposer aux frontières de l'espace de phase S^2 , conditions dont découlent (4.15) et (4.16), soient la cause de l'amortissement rapide des cycles. Rappelons que cette condition de flux force la trajectoire à demeurer à tout instant à l'intérieur de l'espace de phase S^2 . Finalement, indiquons que le choix des triplets $(a_{1,0}, a_{1,1}, a_{2,0})$ et $(b_{0,1}, b_{1,1}, b_{0,2})$ permet facilement, en principe, de générer les types de dynamique illustrés aux figures 4.12 a) et 4.12b). Il était important de faire cette mise en situation avant d'aborder le côté proprement expérimental.

Aux sections 4.3 et 4.4 nous donnons la description détaillée du contenu des simulations qui ont été effectuées sur le réseau cartésien. Pour chacun des modèles dont nous venons de discuter, deux expériences ont été réalisées. Elles ont été respectivement désignées par « *expérience 1* » et « *expérience 2* ». On retrouvera à travers la description de ces expériences plusieurs informations pertinentes concernant la ou les valeurs des paramètres qui ont été sélectionnées pour les effectuer.

4.3 SIMULATIONS DU MODÈLE LOGISTIQUE

Les simulations basées sur le modèle *logistique* devraient nous permettent d'explorer en détail la dynamique simple qui en découle. Nous avons brièvement indiqué à la figure 4.1 quelle est la caractéristique fondamentale de ce type de mécanisme réactionnel. De plus, nous avons vu que l'évolution de la seule espèce en jeu doit se stabiliser à long terme, et ce, quelle que soit la valeur de densité $p_{initial}$. Nous allons donc tenter d'explorer ces différents thèmes à l'aide de deux expériences. Voici en quoi consistent ces expériences.

4.3.1 DESCRIPTION DE L'EXPÉRIENCE 1

Dans le cadre d'une première simulation, il serait intéressant de vérifier la fiabilité de l'automate cellulaire. D'une part nous chercherons à vérifier si la valeur d'équilibre stable qui est atteinte sur un réseau est conforme à celle qui est attendue théoriquement par l'équation logistique. D'autre part, nous vérifierons si la dynamique sur le réseau est du même type que celle qui a été présentée à la figure 4.1. Pour montrer ces deux aspects nous procéderons ainsi : le coefficient a_1 de l'équation logistique sera maintenu fixe, tandis que le coefficient a_2 fera l'objet de plusieurs variations régulières. Plus précisément, le paramètre fixé est : $a_1 = 3$, tandis que le paramètre a_2 sera initialisé à $a_2 = 3$ et subira des variations $\Delta a_2 = 0.2$. L'ensemble de ces conditions initiales sont résumées au tableau 4.1. Au total, 11 simulations seront effectuées de la sorte. Voici les résultats de cette expérience.

4.3.2 ANALYSE DES RÉSULTATS DE L'EXPÉRIENCE 1

Les résultats des deux premières simulations sont présentés à la figure 4.3. La première simulation correspond à la courbe de condition initiale $p_{initial} = 0.5$. Cette courbe indique bien que la densité $p(t)$ croît vers l'état d'équilibre stable $p_{equilibre} = 1$. La deuxième simulation correspond à la courbe de condition initiale $p_{initial} = 1.5$. Cette courbe indique bien cette fois-ci que la densité $p(t)$ décroît vers l'état d'équilibre stable $p_{equilibre} = 1$. Il semble donc que ces deux courbes s'apparentent de très près à celles qui ont été présentées à la figure 4.1. Globalement, ces deux simulations semblent donc concluantes du point de vue de l'évolution de la densité de la seule espèce en jeu.

La figure 4.4 illustre le comportement de la densité moyenne d'équilibre lorsque le coefficient a_2 subit des variations régulières. Pour permettre une comparaison plus efficace, ces densités moyennes ont été normalisées. Notons que la densité normalisée s'obtient par le calcul suivant :

$$p_{normalisé} = p_{observé} / p_{équilibre} \quad (4.37)$$

Le cas idéal suppose que $p_{normalisé} = 1$. Dans ce contexte il est intéressant de remarquer que plus le coefficient a_2 diminue plus la densité normalisée s'approche de un. Une explication de ce phénomène est possible au regard des équations (4.11). En effet, selon (4.11) on peut remarquer que la diminution du coefficient a_2 abaisse aussi les probabilités de transitions $P_{2 \rightarrow 3}$ et $P_{3 \rightarrow 2}$. En un certain sens cette diminution de a_2 semble donc contribuer à rendre certaines transitions plus rares. Dans cette optique il n'est peut être pas surprenant qu'une plus grande précision sur la valeur de la densité $p_{normalisé}$ puisse être observée. De fait la figure 4.4 indique que le maximum de précision est atteint lorsque le coefficient $a_2 = 1$. La figure 4.5 présente les mêmes résultats que ceux de la figure 4.4. Toutefois sur la figure 4.5 on indique les valeurs de densités normalisées

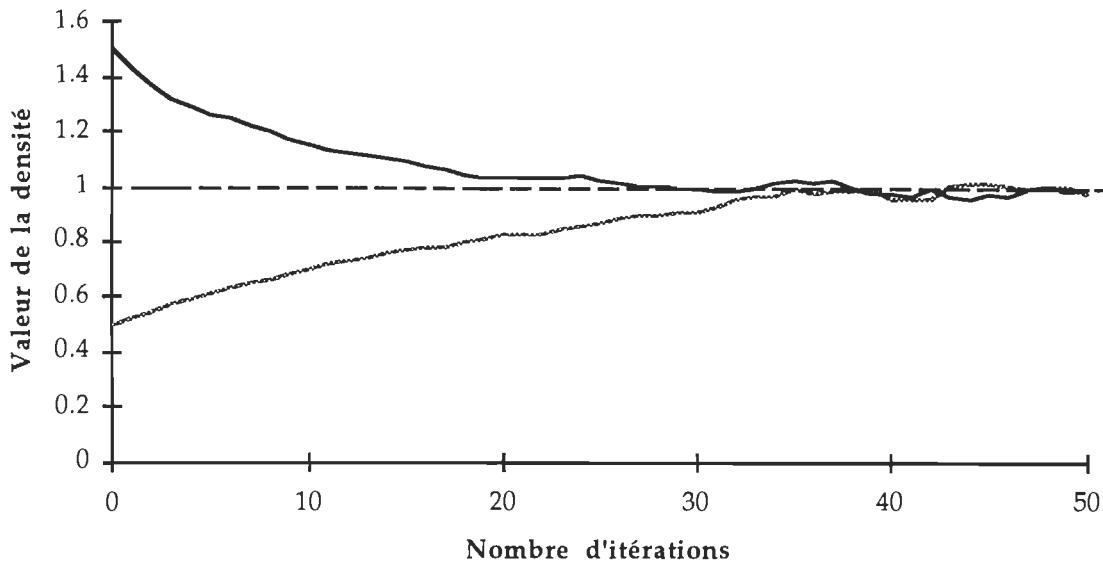


Fig 4.3 Deux courbes types résultant de la simulation du modèle *logistique* sur le réseau.
 (La dimension du réseau est 64×64 , le coefficient $a_1 = a_2 = 3$ et le temps d'échelle $h = 1/36$.
 La valeur de densité d'équilibre attendue est $\rho_{équilibre} = 1$. Les deux simulations portent sur 1000 itérations) .

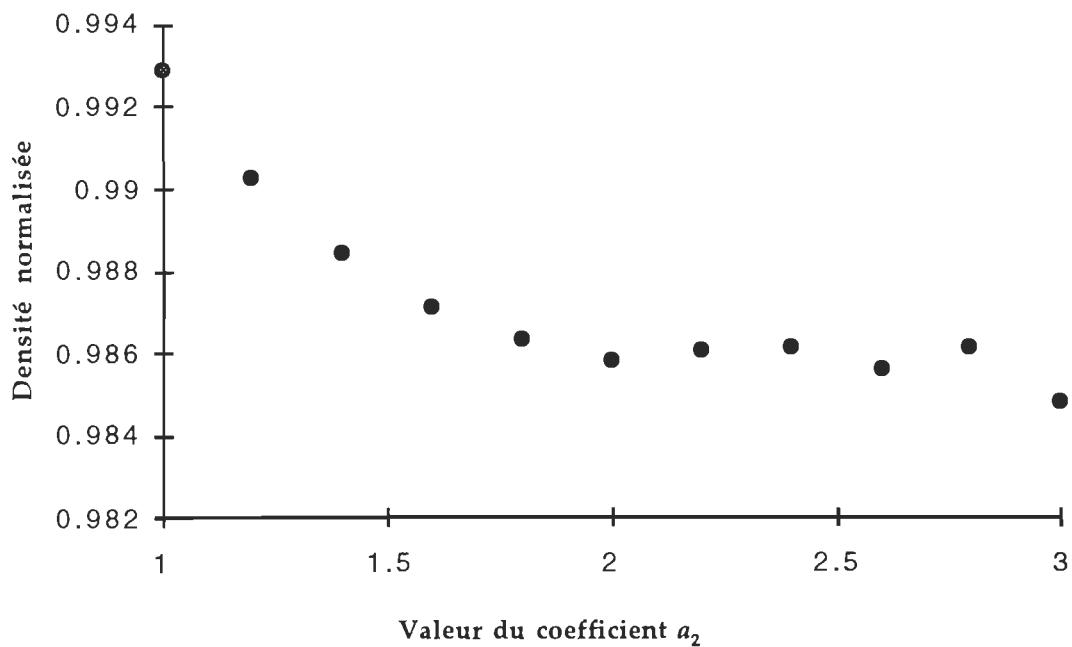


Fig 4.4 Rapport des densités d'équilibre observées avec le densités d'équilibres théoriques en fonction du coefficient a_2 (La dimension du réseau est 64×64 , le coefficient $a_1 = 3$, le temps de relaxation est 500 , la moyenne de chaque densité d'équilibre porte sur 10000 itérations) .

minimales et maximales qui ont été observées lors de chaque simulation. Selon cette figure il est intéressant de noter que l'intervalle [ρ_{minimum} , ρ_{maximum}] se rétrécit aussi lorsque le coefficient a_2 diminue. Ce phénomène semble consistant avec l'hypothèse selon laquelle la diminution du coefficient a_2 semble avoir partiellement le même effet que la diminution du facteur d'échelle h . Globalement, les résultats de cette *expérience 1* indique que le mécanisme réactionnel basé sur le modèle logistique peut bel et bien être reproduit par simulation sur un réseau cartésien.

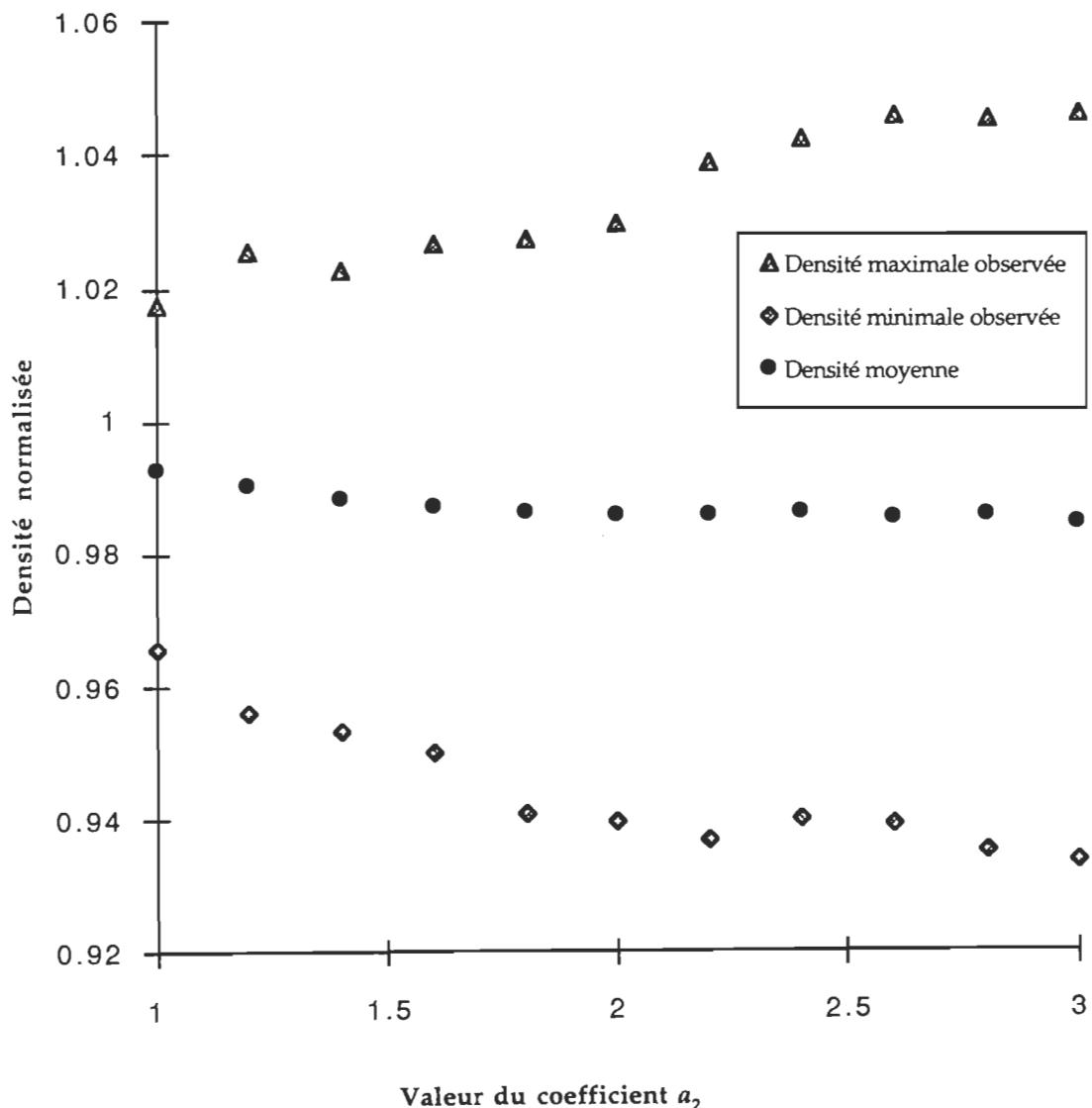


Fig 4.5 Rapport des densités d'équilibre observées avec le densités d'équilibres théoriques en fonction du coefficient a_2 (La dimension du réseau est 64×64 , le coefficient $a_1 = 3$, le temps de relaxation est 500 , la moyenne de chaque densité d'équilibre porte sur 10000 itérations) .

4.3.3 DESCRIPTION DE L'EXPÉRIENCE 2

Expérience 2

L'*expérience 2* se déroule dans des conditions similaires à l'*expérience 1*. Toutefois, cette fois-ci les coefficients a_1 et a_2 ont tous deux été maintenus fixes. Pour mettre en relief l'importance de l'hypothèse des « événements rares » 7 simulations ont été effectuées à différentes valeurs du facteur d'échelle h . Par cette expérience nous chercherons à vérifier l'effet du facteur d'échelle h sur la précision avec laquelle la valeur théorique de l'équilibre stable est atteinte sur réseau.

4.3.3 ANALYSE DES RÉSULTATS DE L'EXPÉRIENCE 2

Les résultats des simulations sont présentés à la figure 4.6. On y remarque en particulier que la diminution du facteur d'échelle h rend bel et bien la densité normalisée près de un. Cet effet semble donc bien confirmer que plus les collisions se font rares plus les hypothèses théoriques, concernant la distribution statistique, semblent respectées.

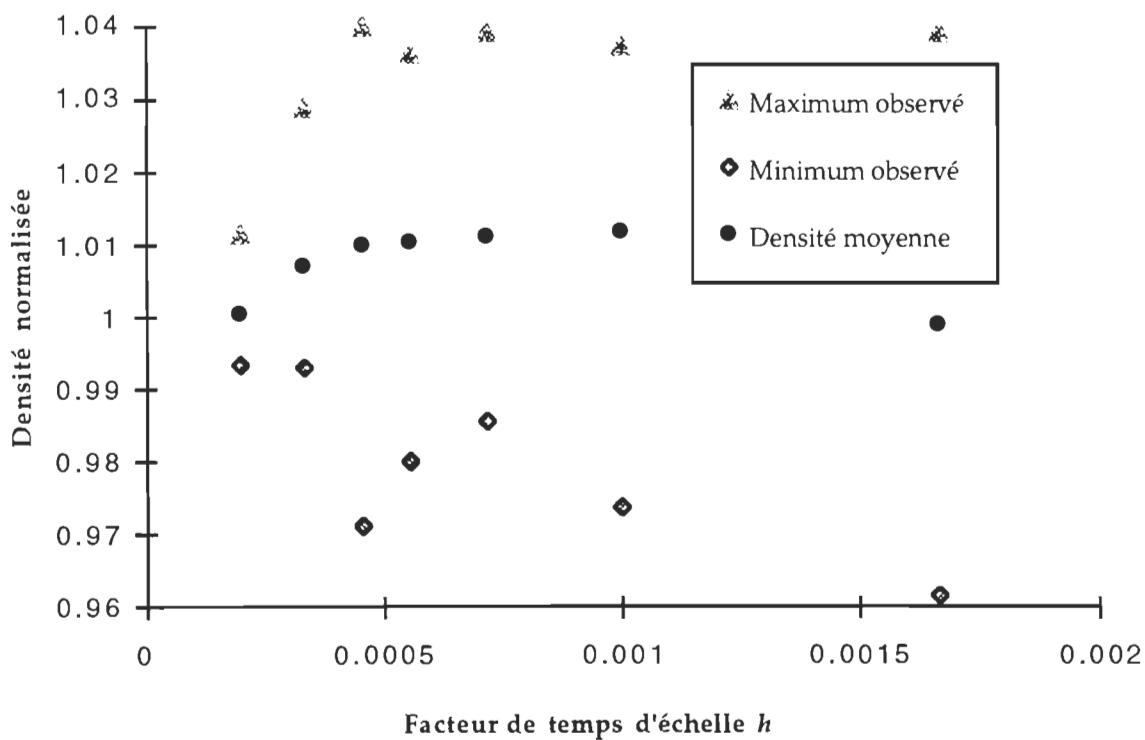


Fig 4.6 Rapport des densités d'équilibre observées avec le densités d'équilibres théoriques en fonction du facteur d'échelle de temps h (La dimension du réseau est 64×64 , les paramètres expérimentaux sont : $a_1 = 3$, $a_2 = 3$, le temps de relaxation est de 500 , la moyenne de chaque densité d'équilibre porte sur 10000 itérations).

4.4 SIMULATION DU MODÈLE DE LOTKA VOLTERRA ÉTENDU

Pour ce modèle plusieurs types de simulations seront entreprises. En un premier temps, nous allons essayer de reproduire la famille de trajectoires types qui ont été présentées à la figure 4.2. En un deuxième temps, nous allons analyser via une simulation appropriée, la capacité du réseau à séparer des trajectoires qui sont quantitativement comparables. Ceci nous amènera à discuter du pouvoir de résolution d'un réseau cartésien de dimension fixée. En dernier lieu, nous tenterons de vérifier l'effet du facteur h sur la densité d'équilibre atteinte sur le réseau. La description des deux expériences recouvrant ces thèmes sont présentées ci-dessous.

4.4.1 DESCRIPTION DE L'EXPÉRIENCE 1

Par cette expérience nous allons recréer les diverses dynamiques que peut engendrer le modèle de *Lotka Volterra étendu*. Pour exposer convenablement les trajectoires types, un choix approprié des coefficients a du être fait. Les triplets $(a_{1,0}, a_{1,1}, a_{2,0})$ et $(b_{0,1}, b_{1,1}, b_{0,2})$ ont été fixés de la sorte : $(a_{1,0}, a_{1,1}, a_{2,0}) = (-4, 2, -1)$ et $(b_{0,1}, b_{1,1}, b_{0,2}) = (1, b_{1,1}, 0.25)$. Selon (4.17) le coefficient $b_{1,1}$ est entièrement libre. Ainsi, les trois premières simulations ont été effectuées en donnant respectivement les valeurs suivantes au coefficient $b_{1,1}$:

$$b_{1,1} = 0.35 \quad b_{1,1} = 1 \quad b_{1,1} = 3 \quad (4.38)$$

Ces trois choix suffisent à reproduire le type de dynamique illustré aux figures 4.2 b) et 4.2 c). Le choix de $b_{1,1} = 0.35$ correspond à la dynamique de la figure 4.2 b), tandis que le choix de $b_{1,1} = 3$ correspond à la dynamique de la figure 4.2 c); la valeur $b_{1,1} = 1$ donne lieu à une dynamique intermédiaire. Une attention spéciale sera apportée au cas $b_{1,1} = 3$. Nous verrons que ce cas précis se prête bien à une discussion sur le pouvoir de résolution du réseau cartésien. Finalement, mentionnons que pour faciliter les comparaisons entre les diverses trajectoires obtenues, la condition initiale $\rho_{initial} = (4, 4)$ a été préservée tout au long des simulations. Cette condition initiale semble un peu excessive; nous allons voir pourquoi il en est ainsi lors de l'analyse des résultats des simulations que voici.

4.4.2 ANALYSE DES RÉSULTATS DE L'EXPÉRIENCE 1

La figure 4.7 illustre une trajectoire caractéristique des systèmes dynamiques dont le temps d'amortissement, pour atteindre l'équilibre stable, est très court. Ce type de dynamique se reconnaît surtout par la quasi-absence de cycle. Sur la figure 4.7 seul un maigre cycle est perceptible. Pour bien mettre en évidence ce genre de trajectoire dont l'amortissement est très rapide, la condition initiale $\rho_{initial} = (4, 4)$ a dû être posée. Dans un autre ordre d'idée on remarque

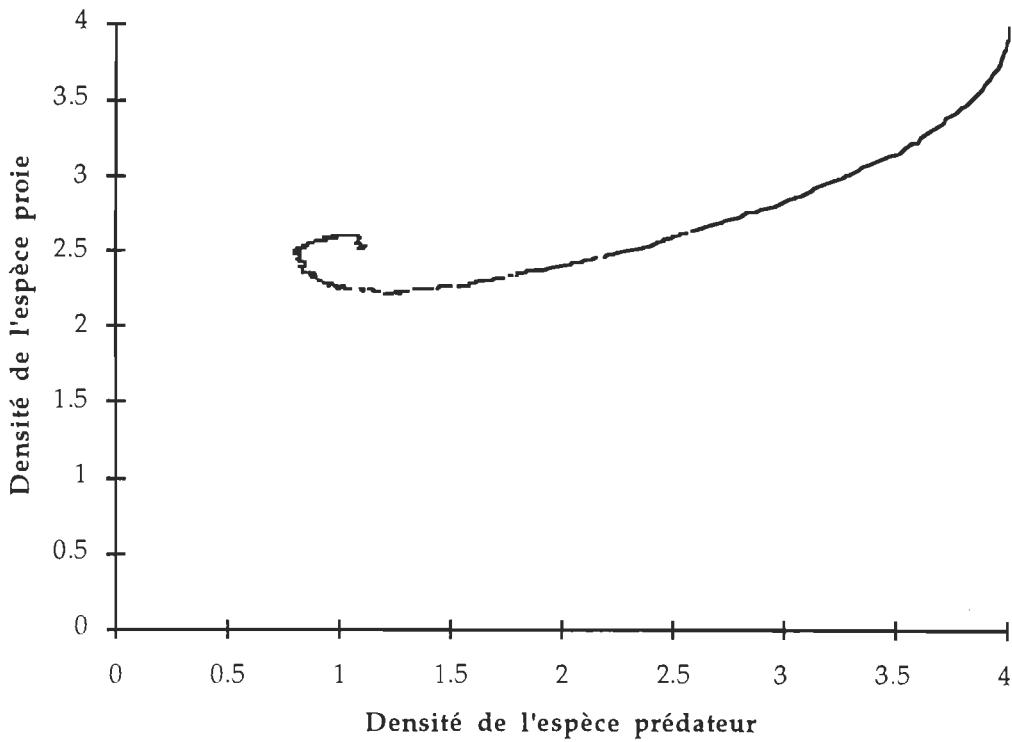


Fig 4.7 Trajectoire des densités dans l'espace de phase S^2 formé des axes (ρ_1 , ρ_2).
 (La dimension du réseau est 64×64 , le facteur d'échelle $h = 1/200$, les coefficients sont : $(a_{1,0}, a_{1,1}, a_{2,0}) = (-4, 2, -1)$, le triplet $(b_{0,1}, b_{1,1}, b_{0,2}) = (1, -0.35, -0.25)$. La simulation porte sur 640 itérations).

que l'aboutissement de la trajectoire coïncide « assez bien » avec la valeur d'équilibre théorique qui est : $\rho_{équilibre} = (1.05, 2.52)$. Les mots « assez bien » feront l'objet d'une analyse beaucoup plus exhaustive lors de l'*expérience 2*.

En comparaison avec la figure 4.7, la figure 4.8 illustre une trajectoire dont le comportement est caractéristique d'un temps d'amortissement plus grand. Ceci se traduit par la présence de deux cycles bien perceptibles. Toutefois pour réussir à mettre en évidence ces deux cycles, la condition initiale $\rho_{initial} = (4, 4)$ a dû être préservée. Cette même condition initiale permet par ailleurs une comparaison plus aisée entre les deux figures. On note que la densité minimale ρ_1 qui a été atteinte sur le réseau par l'espèce prédateur est d'environ $\rho_1 \approx 1/6$. Sur un réseau cartésien de dimension 64×64 cette densité minimale correspond à une présence de 682 êtres de type prédateur. Il est intéressant de noter que malgré la présence d'oscillations dues aux fluctuations aléatoires, un réseau de dimension 64×64 ne semble pas avoir de difficulté à identifier de petites

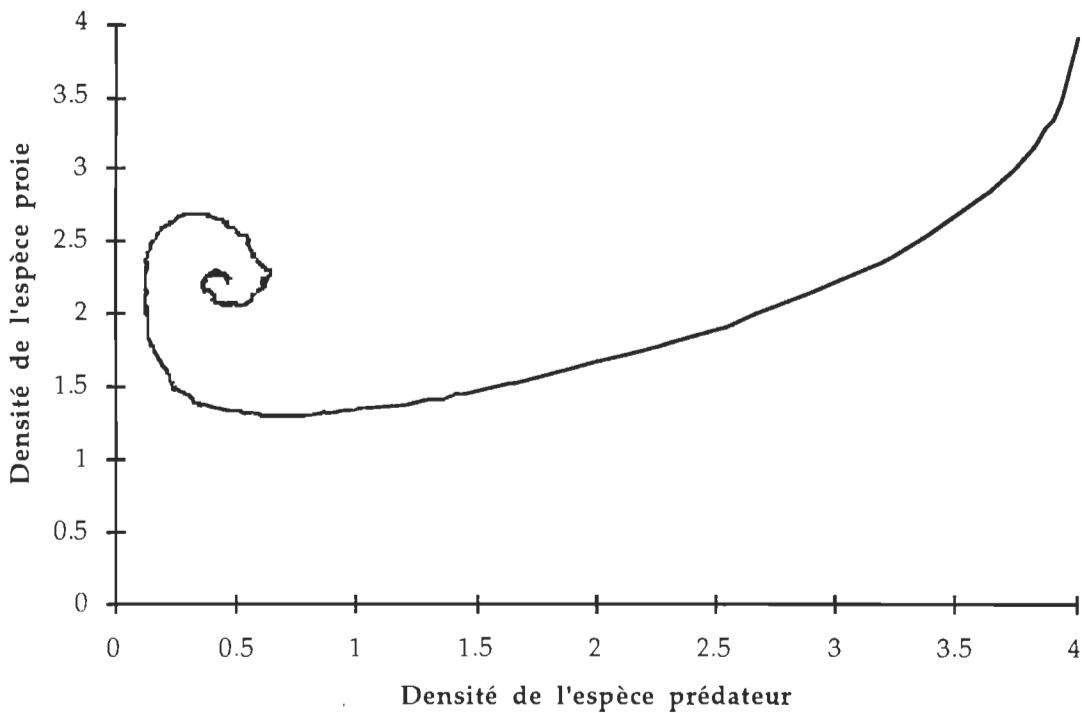


Fig 4.8 Trajectoire des densités dans l'espace de phase S^2 formé des axes (p_1 , p_2).
 (La dimension du réseau est 64×64 , le facteur d'échelle $h = 1/200$, les coefficients sont : $(a_{1,0}, a_{1,1}, a_{2,0}) = (-4, 2, -1)$, le triplet $(b_{0,1}, b_{1,1}, b_{0,2}) = (1, -1, -0.25)$. La simulation porte sur 2000 itérations).

densités d'environ $p_1 \approx 1/6$. Finalement, mentionnons que la densité d'équilibre attendue est la suivante : $p_{\text{équilibre}} = (0.44, 2.22)$. La trajectoire de la figure 4.8 se termine donc vraisemblablement au bon endroit.

À la figure 4.9 une dynamique similaire est présentée. Toutefois cette trajectoire semble présenter une anomalie puisque la densité p_1 semble atteindre la valeur $p_1 = 0$. En réalité, il n'en est rien. Voyons ce phénomène de plus près. Lorsque (4.15) est affecté des coefficients indiqués à la légende de la figure 4.9, on peut montrer que la valeur minimale de la densité p_1 est : $p_1 \approx 1/1000$. Sur un réseau de dimension 64×64 cela correspond à la présence de 4 êtres de l'espèce prédateur; façon de dire que cette espèce devra frôler l'extinction. Le problème qui se pose lors de la simulation est le suivant : il est fort possible que les fluctuations aléatoires fassent passer le nombre d'êtres prédateur de quatre à zéro. Or, selon les équations du tableau 3.2 l'état $p_1 = 0$ est un état absorbant puisque toutes les probabilités de transitions $P_{(0,\alpha) \rightarrow (1,\alpha)}$ sont nulles. Ceci signifie que si l'état $p_1 = 0$ est rencontré alors la densité p_1 ne reprendra jamais d'autres

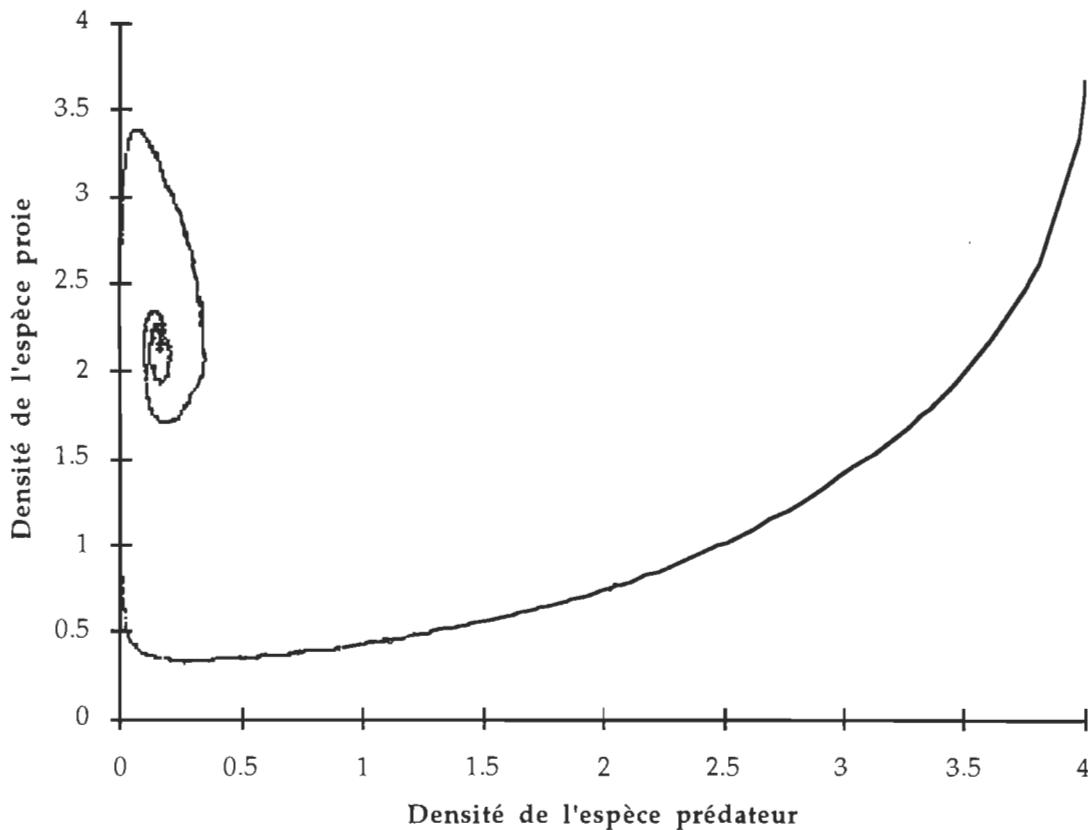


Fig 4.9 Trajectoire des densités dans l'espace de phase S^2 formé des axes (p_1 , p_2).
 (La dimension du réseau est 64×64 , le facteur d'échelle $h = 1/4000$, les coefficients sont : $(a_{1,0}, a_{1,1}, a_{2,0}) = (-4, 2, -1)$, le triplet $(b_{0,1}, b_{1,1}, b_{0,2}) = (1, -3, -0.25)$. La simulation porte sur 25000 itérations).

valeurs que $p_1 = 0$. Diverses simulations ont effectivement conduit à cet état absorbant $p_1 = 0$. Il a donc fallu apporter une plus grande minutie à la réalisation de cette simulation. La façon de contourner cette impasse apparente a été de rendre le facteur d'échelle h très petit puisque c'est précisément ce facteur qui contrôle l'amplitude des oscillations aléatoires.

La figure 4.10 illustre le comportement de la densité p_1 lorsqu'elle frôle la valeur $p_1 = 0$. Cette figure met donc en relief le fait qu'à aucun instant la trajectoire de la figure 4.9 ne touche l'axe p_2 correspondant à l'espèce proie. Il est bon d'ajouter que lors de la simulation le nombre minimal d'êtres prédateurs observé se chiffre à un. Du point de vu de la valeur de la densité minimale observée cela correspond à : $p_1 = 1/64^2 \approx 0.00025$. Or en théorie la densité minimale observée devrait être $p_1 = 4/64^2 \approx 0.000976$. Il semble donc qu'il ne soit pas possible de distinguer deux trajectoires qui sont séparées par une distance plus petite

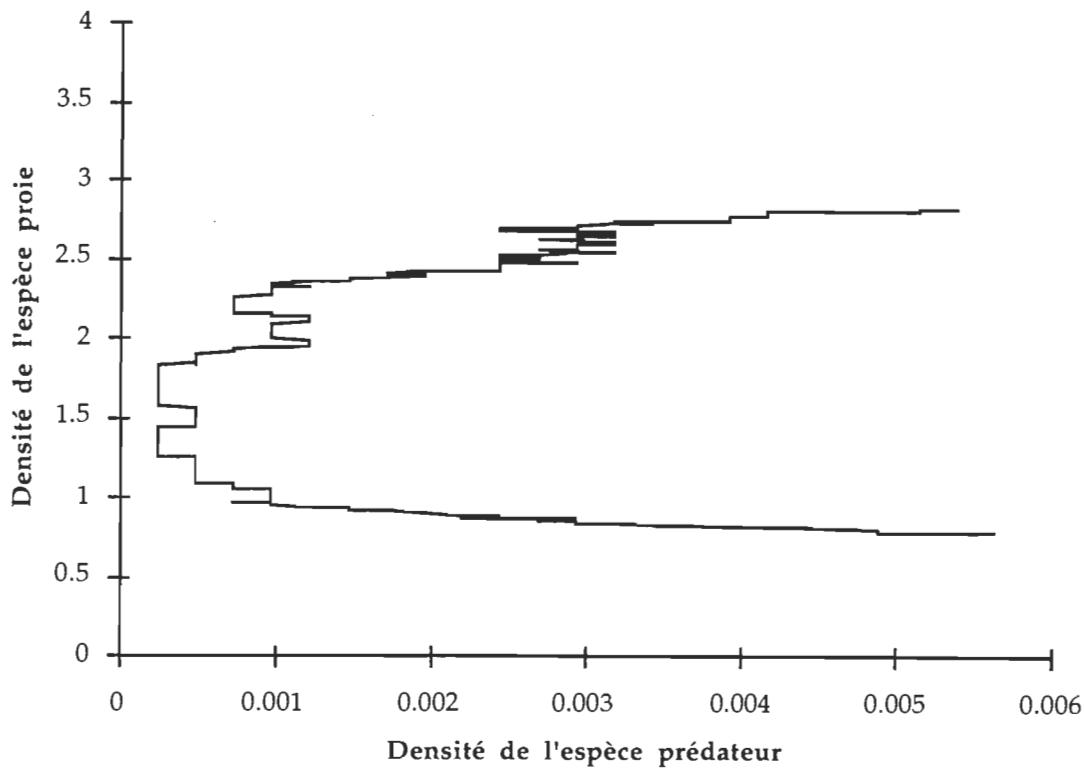


Fig 4.10 Zoom sur la trajectoire des densités dans l'espace de phase S^2 formé des axes (p_1, p_2). (La dimension du réseau est 64×64 , le facteur d'échelle $h = 1/4000$, les coefficients sont : $(a_{1,0}, a_{1,1}, a_{2,0}) = (-4, 2, -1)$, le triplet $(b_{0,1}, b_{1,1}, b_{0,2}) = (1, -3, -0.25)$. La simulation porte sur 2000 itérations).

que $(4 - 1)/64^2 \approx 0.000732$. Ceci nous amène donc à dire qu'un réseau de dimension 64×64 possède un pouvoir de séparation d'environ 0.000732. De plus, cette circonstance fait ressortir que la valeur $b_{1,1} = 3$ constitue presque une borne maximale pour la simulation du modèle de Lotka Volterra étendu sur un réseau 64×64 , tous les autres coefficients étant, bien entendu, laissés identiques.

Les résultats de la dernière simulation de *l'expérience 1* sont présentés à la figure 4.11. Cette figure illustre une trajectoire type où l'équilibre stable coïncide avec la disparition de l'espèce prédateur. Ce type de trajectoire réfère aussi à la figure 4.2a). Pour bien mettre en évidence ce type de comportement, la condition initiale $p_{initial} = (4, 0.1)$ a cette fois-ci été sélectionnée. De plus, pour la simulation en cause la valeur d'équilibre attendue est $p_{équilibre} = (0, 3)$. Selon la figure 4.11 cet état d'équilibre est bel et bien respecté. Il est à noter qu'il est impossible que l'espèce proie soit assujettie à l'extinction. Ce dernier fait est confirmé par l'équation (4.34).

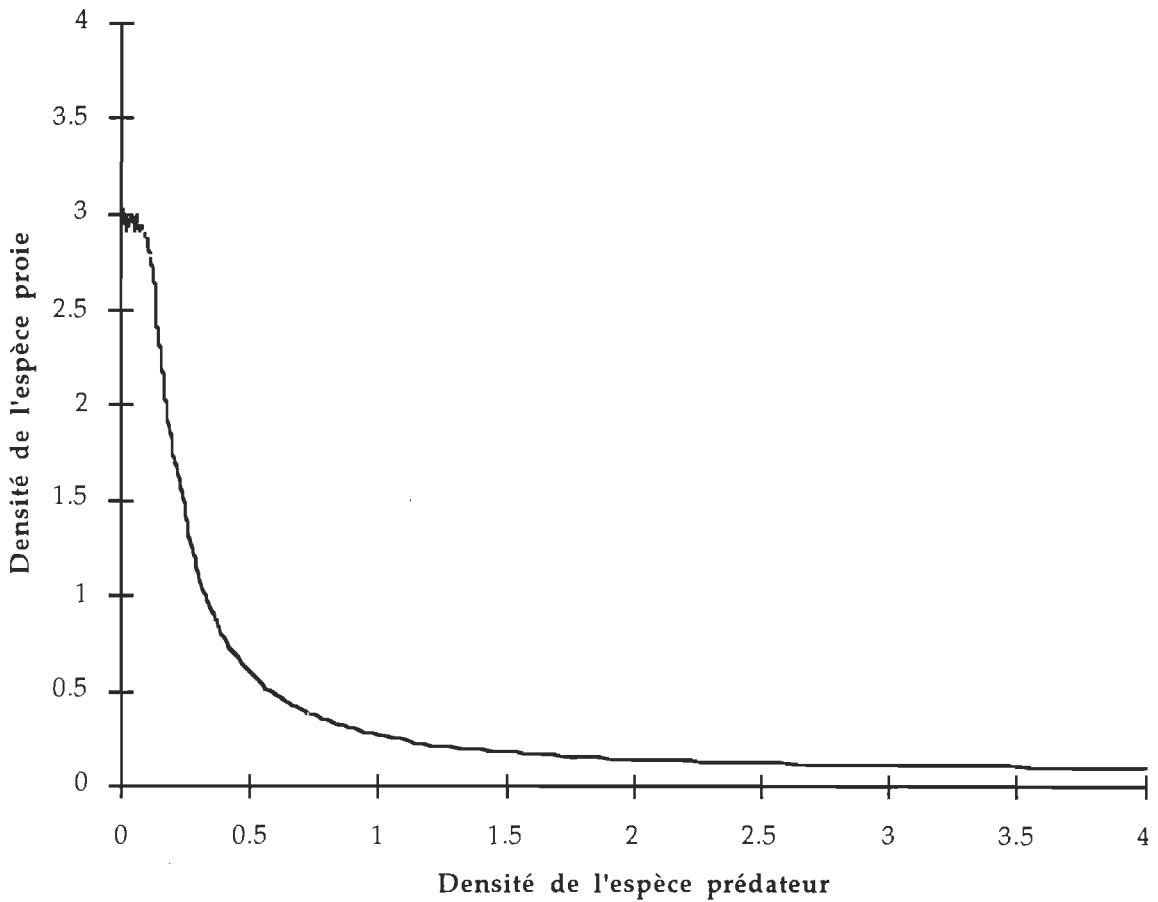


Fig 4.11 Trajectoire des densités dans l'espace de phase S^2 formé des axes (p_1 , p_2).
 (La dimension du réseau est 64×64 , le facteur d'échelle $h = 1/200$, les coefficients sont : $(a_{1,0}, a_{1,1}, a_{2,0}) = (-8, 2, -3)$, le triplet $(b_{0,1}, b_{1,1}, b_{0,2}) = (15, -2, -5)$. La simulation porte sur 1000 itérations).

4.4.3 DESCRIPTION DE L'EXPÉRIENCE 2

Expérience 2

Lors de cette *expérience 2* nous chercherons à vérifier l'effet du facteur d'échelle sur la précision avec laquelle la valeur théorique de l'équilibre stable peut être atteinte sur réseau. Ainsi pour mettre en relief l'importance de l'hypothèse des « événements rares » 10 simulations à différents facteurs d'échelle h ont été effectuées. Pour permettre une plus grande précision graphique ces 10 simulations ont été séparées en deux groupes de cinq.

4.4.4 ANALYSE DES RÉSULTATS DE L'EXPÉRIENCE 2

Les résultats des 5 premières simulations sont présentés à la figure 4.12. Il est important de mentionner que pour toutes ces simulations le facteur d'échelle h doit respecter la contrainte suivante : $h \leq 1/62.5$. De plus, rappelons que la diminution du facteur h devrait en principe faire tendre la densité d'équilibre normalisée vers $p_{\text{normalisé}} = (1, 1)$. Dans ce contexte le point situé à l'extrême gauche mérite une attention particulière puisque ce point correspond à la valeur $h = 1/65$. Il n'est donc pas surprenant que la précision de la densité normalisée soit de piètre qualité pour ce point. Par contre la diminution du facteur h semble bien accentuer cette dernière précision. En effet, la figure 4.11 montre que la suite de points allant de gauche à droite tend de façon assez prononcée vers la valeur de densité normalisée $p_{\text{normalisé}} = (1, 1)$.

La figure 4.12 indique les résultats des cinq autres simulations pour des facteurs d'échelle h plus petits. C'est dans cette optique que la suite de points de la figure 4.13 complète celle de la figure 4.12. Cette fois-ci il est intéressant de remarquer que la suite de points allant de gauche à droite semble tendre davantage vers la valeur de densité normalisée $p_{\text{normalisé}} = (1, 1)$. Toutefois, la figure 4.13 indique aussi que la précision avec laquelle la densité normalisée de l'espèce proie est atteinte est meilleure que celle de l'espèce prédateur pour un même facteur d'échelle h .

Globalement, les résultats des *expériences 1 et 2* révèlent que le mécanisme réactionnel correspondant au modèle de *Lotka Volterra étendu* est respecté par simulation sur réseau cartésien.

4.5 DISCUSSION

La simulation des modèles de croissance que nous avions visés sont à ce stade terminées. Dans l'ensemble, il est intéressant de remarquer que les courbes obtenues lors de ces simulations s'apparentent aux courbes prévisibles par la théorie^{32,33}. De plus, il ressort que la précision avec laquelle peuvent être obtenu les points d'équilibre stables correspondant à un modèle de croissance donné, peut être aussi bonne qu'on le souhaite quitte à amenuiser suffisamment le facteur d'échelle h .

Globalement les résultats de ces simulations confirment l'hypothèse selon laquelle l'implémentation d'un automate cellulaire probabiliste, via un réseau cartésien, puisse permettre la résolution d'un système d'équations différentielles de réaction-diffusions induit par la loi d'action de masse.

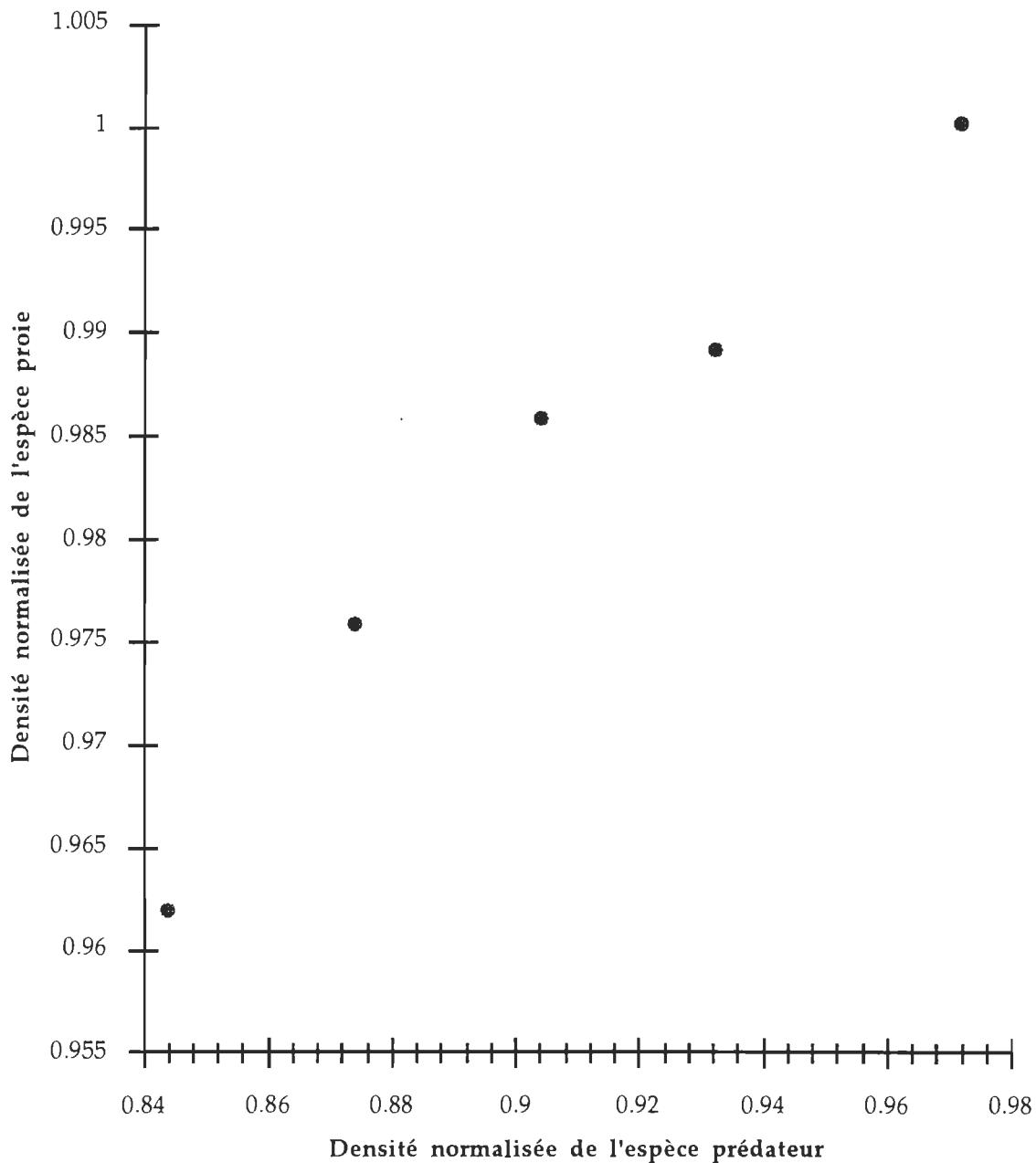


Fig 4.12 Densités moyenne normalisée dans l'espace de phase S^2 formé des axes (p_1 , p_2). Les facteurs d'échelle h correspondant aux points de gauches à droite sont respectivement : $h = 1/65$, $h = 1/70$, $h = 1/80$, $h = 1/90$ et $h = 1/100$. La dimension du réseau est 64×64 , les coefficients sont : $(a_{1,0}, a_{1,1}, a_{2,0}) = (-4, 2, -1)$ et $(b_{0,1}, b_{1,1}, b_{0,2}) = (4, -0.5, -1)$. Le calcul de chaque densité moyenne porte sur 10000 itérations tandis que la condition initiale de chaque simulation est : $p_{initial} = p_{équilibre} = (2, 3)$.

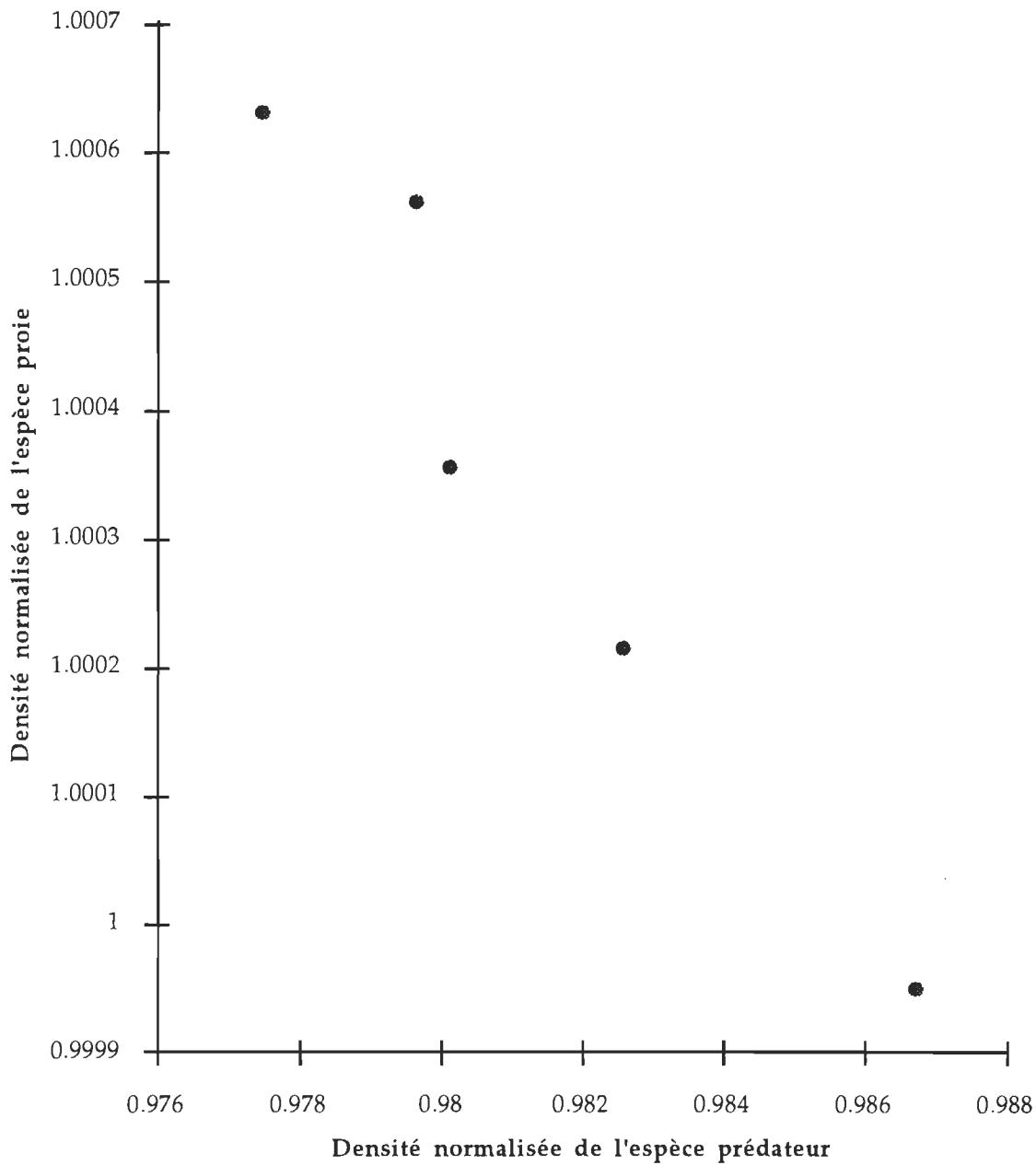


Fig 4.13 Densités moyenne normalisée dans l'espace de phase S^2 formé des axes (ρ_1 , ρ_2). Les facteurs d'échelle h correspondants aux points de gauches à droite sont respectivement : $h = 1/500$, $h = 1/1000$, $h = 1/1500$, $h = 1/2000$ et $h = 1/3000$. La dimension du réseau est 64×64 , les coefficients sont : $(a_{1,0}, a_{1,1}, a_{2,0}) = (-4, 2, -1)$ et $(b_{0,1}, b_{1,1}, b_{0,2}) = (4, -0.5, -1)$. Le calcul de chaque densité moyenne porte sur 10000 itérations tandis que la condition initiale de chaque simulation est : $\rho_{initial} = \rho_{équilibre} = (2, 3)$.

CONCLUSION

Rendu au terme de ce travail, plusieurs conclusions intéressantes doivent être dégagées. Mais avant nous allons retracer les jalons directeurs que ce mémoire a posés. Nous avons tout d'abord défini et construit l'automate cellulaire. Lors de la construction de l'automate cellulaire, des analyses minutieuses nous ont indiqué que seuls les mécanismes admettant un automate cellulaire cohérent peuvent être considérés. C'est ainsi que le terme « cohérent » a dû être précisé par les conditions *a), b) et c)* reliées à l'équation (3.38). Ces conditions ayant été posées nous avons pu par la suite entreprendre l'investigation des modèles classiques de croissance par l'utilisation d'un automate cellulaire. L'analyse de ces modèles classiques (*Malthus, logistique, Lotka Volterra et Lotka Volterra étendu*) nous a permis de distinguer plusieurs faits. D'une part, nous avons vu que les contraintes apportées par la structure probabiliste de cet automate (conditions *a), b) et c)* reliées à l'équation (3.38)) peuvent exclure la possibilité de recréer certains de ces mécanismes même si ceux-ci semblent coller à une réalité physique tangible. En particulier, nous avons vu que le modèle de *Lotka Volterra* n'admet pas la construction d'un automate cellulaire cohérent. Pourtant, il semble que A.J.Lotka ait montré que ce modèle puisse être utilisé pour décrire l'interaction entre certaines espèces^{31,32,36}.

Parmi les modèles classiques de croissance admettant un automate cellulaire cohérent, seuls les modèles *logistique* et de *Lotka Volterra étendu* ont pu être retenus et analysés plus en profondeur. En ce qui a trait à ces deux modèles, nous avons en particulier illustré de quelle manière s'utilise la proposition (2.32) et comment s'affranchir des conditions *a), b) et c)* reliées à l'équation (3.38). Une fois ce travail terminé, nous avons alors eu l'occasion de réaliser, à travers plusieurs expériences, que l'automate cellulaire est effectivement en mesure de recréer ces deux mécanismes de croissance fixés a priori. Dans l'ensemble, ce travail a donc exposé la méthodologie à utiliser pour répondre aux questions suivantes : un modèle de croissance donné peut-il être recréé par l'utilisation d'un automate cellulaire ? Et comment le construit-on ? En ce sens, on peut dire que ce mémoire donne les limites et les avenues qui sont susceptibles d'être empruntées pour simuler ces modèles à l'aide d'un automate cellulaire. Malgré le fait que ces modèles soient étudiés et classifiés depuis longtemps dans le cadre théorique des équations différentielles, ils n'en constituent pas moins une source d'intérêt importante pour ceux qui cherchent à explorer des

dynamiques plus complexes sur les réseaux^{1,2}. Par exemple, le simple modèle logistique contient en lui-même la famille quadratique³⁷, famille qui est propre à l'analyse des points de bifurcations et des points fixes périodiques des systèmes dynamiques non linéaires. Il pourrait être effectivement intéressant de vérifier si l'utilisation d'un réseau permet d'identifier des points de cette nature. Cette même interrogation pourrait être étendue graduellement aux modèles à 2 espèces et plus^{1,2}.

Toutefois, bien que ce mémoire n'a considéré que quatre mécanismes classiques, soit ceux provenant des modèles de *Malthus*, *logistique*, de *Lotka Volterra* et de *Lotka Volterra étendu*, il n'en demeure pas moins que la méthodologie basée sur l'utilisation de l'automate cellulaire se généralise à une multitude d'autres modèles. À ce titre, il faut mentionner que la proposition (2.32) qui a été obtenue, permet d'envisager l'interaction de « n » espèces différentes. Par exemple, il est possible de penser recréer un « système écologique » composé de végétaux, d'insectes, d'oiseaux et de mammifères. À plusieurs points de vue cette perspective semble intéressante. Toutefois, ce mémoire a su aussi insister sur le fait que la construction de l'automate cellulaire est assujettie aux conditions *a), b) et c)* de l'équation (3.38). À propos de ces conditions, il est utile de mentionner que la condition de flux (condition *b*) aux frontières de l'espace de phase pour un nombre d'espèces $n > 3$, est plus difficile à appliquer. Malgré cette difficulté, il n'est pas interdit de penser que l'on peut faire interagir autant d'espèces que l'on souhaite sur un réseau cartésien.

Lors de la présentation du concept de réseau nous avons eu l'occasion de noter que ce réseau joue en fait le rôle du territoire sur lequel se produisent certaines interactions. Ceci revient à dire que le réseau préserve donc l'information concernant la distribution spatiale des différentes espèces lorsqu'elles se meuvent et interagissent. Lors des simulations il est donc possible de voir de quelle manière les espèces se répartissent géographiquement. Cette dimension territoriale, qu'apporte l'utilisation d'un réseau, est un ajout important aux modèles classiques puisque ces modèles n'incorporent que la dimension temporelle de l'évolution des différentes espèces. On pourrait donc concevoir des modèles encore plus sophistiqués en introduisant sur le réseau des obstacles (rivières, types de sol, forêt dense, etc.) ayant pour rôle de limiter le déplacement de certaines espèces. Ce sont autant d'avenus qu'il est permis d'explorer.

Par contre, l'utilisation de l'automate cellulaire que nous avons construit possède malgré tout un niveau de limitation plus grand qu'on ne le pense. Qu'en est-il exactement ? La construction de l'automate cellulaire repose sur plusieurs hypothèses qui, d'un point de vue biologique, doivent apparaître très simplificatrices. À ce sujet, mentionnons que l'hypothèse de cellules qui s'ignorent mutuellement sur un réseau (hypothèse d'équiprobabilité concernant l'occupation des arêtes d'une cellule), et l'hypothèse selon laquelle la suite de transformations que subit chaque cellule est sans

mémoire (hypothèse de Markov), constituent une limite sérieuse quant au type de phénomènes de croissance que l'on peut se permettre d'envisager dans un contexte biologique réel. On pourrait penser par exemple à raffiner la notion de reproduction en la rendant *cyclique*. En fait, cette modification apparaît importante car s'il y a une notion qui occupe une place prépondérante dans un contexte biologique réel, c'est bien celle de « *phénomènes cycliques* ». Or, l'introduction du concept de reproduction cyclique suffit à rendre les équations différentielles à temps *explicite* (l'utilisation de la loi d'action de masse nous conduit toujours à des équations différentielles à temps *implicite*), suite à quoi la proposition (2.32) n'est plus applicable. D'autre part, il est aussi fort possible que l'hypothèse de Markov, qui considère que la réalisation d'un événement n'est pas influencée par le temps qui s'est écoulé, doivent elles aussi être abandonnés dans une telle circonstance. Globalement, il faut donc mentionner que toutes les circonstances ayant pour effet de rendre l'automate cellulaire sensible aux conditions du passé rendent inefficace la proposition (2.32). Ainsi, si un processus biologique semble indiquer qu'il ne s'agit pas simplement de se nourrir, de combattre et de se reproduire au sens de la loi d'action de masse pour survivre, alors ce processus biologique ne sera pas traduisible par les éléments de ce travail. La question que l'on peut se poser alors est la suivante : cette notion de mémoire du temps passé est-elle essentielle à l'avènement d'un phénomène de nature biologique ou est-elle secondaire ?

Annexe A

DÉMONSTRATION DE LA PROPOSITION (2.32)

Précisons tout d'abord les notations, ainsi que le sens des variables, qui seront utilisés tout au long de cette démonstration. La notation utilisée pour les sommes imbriquées sera la suivante :

$$\sum_{\alpha \in S^n} \sum_{\beta \in S^n} = \sum_{\alpha_1=0}^m \sum_{\alpha_2=0}^m \dots \dots \sum_{\alpha_n=0}^m \sum_{\beta_1=0}^m \sum_{\beta_2=0}^m \dots \dots \sum_{\beta_n=0}^m \quad (\text{A.1})$$

L'ensemble S est défini par $S = \{ 0 \dots m \} m \in \mathbb{N}$ où \mathbb{N} est l'ensemble des entiers naturels. Physiquement S représente un *ensemble étiquette* de directions vers lesquelles une particule peut se diriger. Il va donc de soi que $S^n = S \times S \dots \times S$. Les états α et β , du système physique analysé, sont des n -uplets ordonnés. Lorsqu'il sera nécessaire, nous les expliciterons par $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$ et par $\beta = (\beta_1, \beta_2, \dots, \beta_n)$ où $\alpha_i, \beta_j \in S$. Dans un autre ordre d'idées, le développement d'un polynôme à n variables sera noté ainsi :

$$F_i(\rho) = F_i(\rho_1, \rho_2, \dots, \rho_n) = \sum_{l_1=0}^m \sum_{l_2=0}^m \dots \dots \sum_{l_n=0}^m a_{l_1, l_2, \dots, l_n}^{(i)} \prod_{j=1}^n (\rho_j)^{l_j} = \sum_{l=0}^m a_l^{(i)} \prod_{j=1}^n (\rho_j)^{l_j} \quad (\text{A.2})$$

Ici, l'indice (i) situé à l'exposant des coefficients $a_{l_1, l_2, \dots, l_n}^{(i)}$ de (A.2) signifie que la fonction de densité de la i -ème particule est un polynôme du même type que celui illustré ci-dessus. Ce développement en série de puissances finies est induit par la *loi d'action de masse* de Van Hoff. Dans le but de préciser davantage les correspondances physiques de ce travail, insistons sur les points suivants. L'état d'un système physique au temps t est désigné par α tandis que β désigne l'état de ce même système physique au temps $t+1$. Pour compléter cette description, mentionnons que le phénomène physique étudié nous donne à penser que β est le résultat d'une transformation de α . Cette transformation est probabiliste et s'effectue entre les unités de temps t et $t+1$. Bien que cela ne soit pas primordial, nous dirons que $t \in \mathbb{N}$. Pour noter ce passage d'état de α à β nous utiliserons le diagramme fléché $\alpha \rightarrow \beta \equiv (\alpha_1, \alpha_2, \dots, \alpha_n) \rightarrow (\beta_1, \beta_2, \dots, \beta_n)$. Physiquement, α_i peut être vu ici comme l'état de la i -ème espèce au temps t alors que β_i est l'état de cette même i -ème espèce au temps $t+1$ après transformation. La présente étude considère plus que la transformation $\alpha \rightarrow \beta$. Plus généralement, nous nous intéressons à la chaîne de transformations $\alpha \rightarrow \beta \rightarrow \gamma \rightarrow \xi \dots \rightarrow \psi$ lorsque $t \rightarrow \infty$. Cette chaîne revêt donc un caractère *dynamique* qui ne saurait cesser de se modifier dans le temps à moins qu'un *équilibre stable* du système physique, représenté par la chaîne, soit atteint. Sous l'hypothèse que cette chaîne d'états est markovienne, nous pouvons alors considérer que les deux seuls éléments d'intérêt de cette chaîne correspondent respectivement à l'état du système au temps t et à l'état du système au temps $t+1$. Ceci revient donc à dire que l'état ultérieur dans lequel pourra se retrouver le système est à un pas, et seulement à un pas, de mémoire. Ce pas caractérise le changement d'état du

système entre les temps t et $t+1$. Au cours du présent traitement, ces deux états principaux ont été dégagés, sans perte de généralité, par α et β ; tout autre état n'influencant pas l'état à venir. C'est dans cette optique que la quantité scalaire $P_{\alpha \rightarrow \beta}$ désigne la probabilité que le système effectue la transition de l'état α vers l'état β tandis que $P_{\alpha \rightarrow \beta}$ représente la matrice contenant les éléments de transitions $P_{\alpha \rightarrow \beta}$. Ce petit préliminaire est suffisant pour saisir le texte qui suit. Nous avons établi en (2.31) que les probabilités de transitions $P_{\alpha \rightarrow \beta}$ s'exprime en fonction des coefficients $a_l^{(i)}$ par l'égalité suivante :

$$\sum_{\alpha \in S^n} \sum_{\beta \in S^n} (\beta_i - \alpha_i) P_{\alpha \rightarrow \beta} \prod_{j=1}^n \binom{m}{\alpha_j} \left(\frac{\rho_j}{m}\right)^{\alpha_j} \left(1 - \frac{\rho_j}{m}\right)^{m-\alpha_j} = h \sum_{l=0}^m a_l^{(i)} \prod_{j=1}^n \left(\frac{\rho_j}{m}\right)^{l_j} \quad (\text{A.3})$$

Rappelons que le but principal de cette démonstration est de mettre en oeuvre une stratégie claire et précise visant à déterminer les éléments $P_{\alpha \rightarrow \beta}$ de la matrice de transitions $P_{\alpha \rightarrow \beta}$. Pour parvenir à réaliser ce programme l'outil de base utilisé sera le développement d'un polynôme à n variables. Nous sommes maintenant prêts à entreprendre la démonstration de la proposition (2.32) de ce mémoire. Débutons cette construction en utilisant le développement du binôme de Newton, soit :

$$\left(1 - \frac{\rho_j}{m}\right)^{m-\alpha_j} = \sum_{l=0}^{m-\alpha_j} \binom{m-\alpha_j}{l} (-1)^l \left(\frac{\rho_j}{m}\right)^l \quad (\text{A.4})$$

Substituer (A.4) dans (A.3) conduit à :

$$\sum_{\alpha \in S^n} \sum_{\beta \in S^n} (\beta_i - \alpha_i) P_{\alpha \rightarrow \beta} \prod_{j=1}^n \binom{m}{\alpha_j} \left(\frac{\rho_j}{m}\right)^{\alpha_j} \left[\sum_{l=0}^{m-\alpha_j} \binom{m-\alpha_j}{l} (-1)^l \left(\frac{\rho_j}{m}\right)^l \right] = h \sum_{l=0}^m a_l^{(i)} \prod_{j=1}^n \left(\frac{\rho_j}{m}\right)^{l_j} \quad (\text{A.5})$$

Le facteur d'échelle de temps h est un paramètre fixé. Pour alléger les développements algébriques subséquents, nous allons absorber cette constante de temps à travers les coefficients $a_l^{(i)}$. Nous allons dire que $a_l^{(i)} := h a_l^{(i)}$ quitte à inverser cette affectation un fois le travail terminé.

Cette remarque appliquer à (A.5) conduit à écrire:

$$\sum_{\alpha \in S^n} \sum_{\beta \in S^n} (\beta_i - \alpha_i) P_{\alpha \rightarrow \beta} \prod_{j=1}^n \binom{m}{\alpha_j} \left(\frac{\rho_j}{m}\right)^{\alpha_j} \left[\sum_{l=0}^{m-\alpha_j} \binom{m-\alpha_j}{l} (-1)^l \left(\frac{\rho_j}{m}\right)^l \right] = \sum_{l=0}^m a_l^{(i)} \prod_{j=1}^n \left(\frac{\rho_j}{m}\right)^{l_j} \quad (\text{A.6})$$

Un réaménagement élémentaire des termes de (A.6) conduit à :

$$\sum_{\alpha \in S^n} \sum_{\beta \in S^n} (\beta_i - \alpha_i) P_{\alpha \rightarrow \beta} \prod_{j=1}^n \left[\sum_{l=0}^{m-\alpha_j} \binom{m}{\alpha_j} \left(\frac{\rho_j}{m}\right)^{\alpha_j} \binom{m-\alpha_j}{l} (-1)^l \left(\frac{\rho_j}{m}\right)^l \right] = \sum_{l=0}^m a_l^{(i)} \prod_{j=1}^n \left(\frac{\rho_j}{m}\right)^{l_j} \quad (\text{A.7})$$

À un changement de variables près, le résultat (A.7) est identique à celui présenté par Wu et Kapral^{1,2}. La plupart des auteurs, traitant de la dynamique moléculaire entre deux ou trois substances, réfèrent régulièrement à ce dernier résultat. Ces auteurs ne précisent pas la méthodologie à emprunter pour bâtir la matrice $P_{\alpha \rightarrow \beta}$ nécessaire à la simulation d'une dynamique de croissance sur un réseau cartésien. Les quelques calculs présentés dans ces articles le sont pour des cas particuliers, et ils semblent, à en juger par les lourdes notations utilisées^{1,2} pour les présenter, fort laborieux à effectuer. Du point de vue de la littérature scientifique il semble donc que (A.7) n'ait pas été présenté sous une forme simple. Poursuivons le travail. De (A.2) et (A.7) on obtient :

$$F_i(\rho) = \sum_{\alpha \in S^n} \sum_{\beta \in S^n} (\beta_i - \alpha_i) P_{\alpha \rightarrow \beta} \prod_{j=1}^n \left[\sum_{l=0}^{m-\alpha_j} \binom{m}{\alpha_j} \left(\frac{\rho_j}{m}\right)^{\alpha_j} \binom{m-\alpha_j}{l} (-1)^l \left(\frac{\rho_j}{m}\right)^l \right] \quad (\text{A.8})$$

$$F_i(\rho) = \sum_{\alpha \in S^n} \sum_{\beta \in S^n} (\beta_i - \alpha_i) P_{\alpha \rightarrow \beta} \sum_{l_1=0}^{m-\alpha_1} \sum_{l_2=0}^{m-\alpha_2} \dots \sum_{l_n=0}^{m-\alpha_n} \left[\prod_{j=1}^n (-1)^{l_j} \binom{m}{\alpha_j} \binom{m-\alpha_j}{l_j} \left(\frac{\rho_j}{m}\right)^{l_j} \left(\frac{\rho_j}{m}\right)^{\alpha_j} \right] \quad (\text{A.9})$$

$$F_i(\rho) = \sum_{\alpha \in S^n} \sum_{\beta \in S^n} (\beta_i - \alpha_i) P_{\alpha \rightarrow \beta} \sum_{l_1=0}^{m-\alpha_1} \sum_{l_2=0}^{m-\alpha_2} \dots \sum_{l_n=0}^{m-\alpha_n} \left[\prod_{j=1}^n (-1)^{l_j} \binom{m}{\alpha_j} \binom{m-\alpha_j}{l_j} \left(\frac{\rho_j}{m}\right)^{l_j + \alpha_j} \right] \quad (\text{A.10})$$

Dans le but de rendre l'exposant de ρ_j plus simple, posons maintenant :

$$l_j = k_j - \alpha_j \Rightarrow k_j = l_j + \alpha_j \Rightarrow \begin{cases} \text{Si } l_j = 0 \Rightarrow k_j = \alpha_j \\ \text{Si } l_j = m - \alpha_j \Rightarrow k_j = m \end{cases} \quad (\text{A.11})$$

Ce changement de variables sur (A.10) implique l'égalité suivante :

$$F_i(\rho) = \sum_{\alpha \in S^n} \sum_{\beta \in S^n} (\beta_i - \alpha_i) P_{\alpha \rightarrow \beta} \sum_{k_1=\alpha_1}^m \sum_{k_2=\alpha_2}^m \dots \sum_{k_n=\alpha_n}^m \left[\prod_{j=1}^n (-1)^{k_j - \alpha_j} \binom{m}{\alpha_j} \binom{m-\alpha_j}{k_j - \alpha_j} \left(\frac{\rho_j}{m}\right)^{k_j} \right] \quad (\text{A.12})$$

En utilisant la notation convenue pour les n sommes imbriquées, (A.12) prend la forme :

$$F_i(\rho) = \sum_{\alpha \in S^n} \sum_{\beta \in S^n} (\beta_i - \alpha_i) P_{\alpha \rightarrow \beta} \sum_{k=\alpha}^m \left[\prod_{j=1}^n (-1)^{k_j - \alpha_j} \binom{m}{\alpha_j} \binom{m - \alpha_j}{k_j - \alpha_j} \left(\frac{\rho_j}{m} \right)^{k_j} \right] \quad (A.13)$$

En utilisant l'identité combinatoire

$$\binom{m}{\alpha_j} \binom{m - \alpha_j}{k_j - \alpha_j} = \binom{m}{k_j} \binom{k_j}{\alpha_j} \quad (A.14)$$

on peut écrire :

$$F_i(\rho) = \sum_{\alpha \in S^n} \sum_{\beta \in S^n} (\beta_i - \alpha_i) P_{\alpha \rightarrow \beta} \sum_{k=\alpha}^m \left[\prod_{j=1}^n (-1)^{k_j - \alpha_j} \binom{m}{k_j} \binom{k_j}{\alpha_j} \left(\frac{\rho_j}{m} \right)^{k_j} \right] \quad (A.15)$$

Selon (A.2) le résultat (A.15) s'écrit aussi comme :

$$\sum_{l=0}^m a_l^{(i)} \prod_{j=1}^n (\rho_j)^{l_j} = \sum_{\alpha \in S^n} \sum_{\beta \in S^n} \sum_{k=\alpha}^m (\beta_i - \alpha_i) P_{\alpha \rightarrow \beta} \prod_{j=1}^n (-1)^{k_j - \alpha_j} \binom{m}{k_j} \binom{k_j}{\alpha_j} \left(\frac{\rho_j}{m} \right)^{k_j} \quad (A.16)$$

Rappelons que le but visé est de lier les éléments de transitions $P_{\alpha \rightarrow \beta}$ aux coefficients $a_l^{(i)}$ du développement polynomial que propose le membre de gauche de (A.16). En prenant en compte le critère d'égalité entre deux polynômes finis, la route à suivre devrait être claire. Ainsi, de (A.16) on a:

$$\sum_{l=0}^m a_l^{(i)} \prod_{j=1}^n (\rho_j)^{l_j} = \sum_{\alpha \in S^n} \sum_{\beta \in S^n} \sum_{k=\alpha}^m (\beta_i - \alpha_i) P_{\alpha \rightarrow \beta} \prod_{j=1}^n \left(\frac{(-1)^{k_j - \alpha_j}}{m^{k_j}} \right) \binom{m}{k_j} \binom{k_j}{\alpha_j} \prod_{j=1}^n \rho_j^{k_j} \quad (A.17)$$

Pour simplifier l'écriture de (A.17), posons :

$$G(k, \alpha, m) = G(k_1, k_2, \dots, k_n, \alpha_1, \alpha_2, \dots, \alpha_n, m) = \prod_{j=1}^n \left(\frac{(-1)^{k_j - \alpha_j}}{m^{k_j}} \right) \binom{m}{k_j} \binom{k_j}{\alpha_j} \quad (A.18)$$

De cette dernière écriture :

$$\Rightarrow \sum_{l=0}^m a_l^{(i)} \prod_{j=1}^n (\rho_j)^{l_j} = \sum_{\alpha \in S^n} \sum_{\beta \in S^n} \sum_{k=\alpha}^m (\beta_i - \alpha_i) P_{\alpha \rightarrow \beta} G(k, \alpha, m) \prod_{j=1}^n \rho_j^{k_j} \quad (A.19)$$

Pour discuter clairement de l'égalité entre ces deux polynômes, il faudrait, d'une part, retrouver un regroupement de n sommes sommant de 0 à m , et, d'autre part, regrouper

ces sommes à la gauche immédiate du membre de droite de l'égalité (A.19). Dans ce contexte, l'indice de sommation $k = \alpha$ apparaît donc un peu embêtant. Voyons ce qu'il est possible de faire. En dépliant la « triple somme » de (A.19) l'égalité devient :

$$\sum_{\alpha_1=0}^m \sum_{\alpha_2=0}^m \cdots \sum_{\alpha_n=0}^m \sum_{\beta_1=0}^m \sum_{\beta_2=0}^m \cdots \sum_{\beta_n=0}^m \sum_{k_1=\alpha_1}^m \sum_{k_2=\alpha_2}^m \cdots \sum_{k_n=\alpha_n}^m (\beta_i - \alpha_i) P_{\alpha \rightarrow \beta} G(k, \alpha, m) \prod_{j=1}^n p_j^{k_j} \quad (\text{A.20})$$

En (A.20) les sommes sur β_j , $0 \leq j \leq n$, peuvent toutes être décalées vers la droite car, au total, le domaine de sommation s'effectue sur une grille rectangulaire R de points où $R = S^n$. De cette considération (A.20) peut aussi s'écrire de la sorte :

$$\sum_{\alpha_1=0}^m \sum_{\alpha_2=0}^m \cdots \sum_{\alpha_n=0}^m \sum_{k_1=\alpha_1}^m \sum_{k_2=\alpha_2}^m \cdots \sum_{k_n=\alpha_n}^m (\beta_i - \alpha_i) P_{\alpha \rightarrow \beta} G(k, \alpha, m) \prod_{j=1}^n p_j^{k_j} \quad (\text{A.21})$$

Un traitement similaire, quoique plus partiel, peut être appliqué aux sommes de (A.21) pour α_j , $0 \leq j \leq n$, lorsque l'on remarque que leur domaine de sommation s'effectue aussi sur une grille de points $R \subset S^n$. Toutefois, nous devons apporter une réserve. En effet, lors de son décalage vers la droite chaque α_j , $0 \leq j \leq n$, est bloqué par un borne de sommation inférieure $k_j = \alpha_j$, $0 \leq j \leq n$. Toutefois, pour $\xi \neq j$, $0 \leq \xi \leq n$, les sommes faisant intervenir la variable α_ξ peuvent librement passer au travers de la j -ème somme. Une fois ce traitement effectué sur tous les indices pertinents de (A.21), on trouve alors l'égalité :

$$\sum_{\alpha_1=0}^m \sum_{k_1=\alpha_1}^m \cdots \sum_{\alpha_2=0}^m \sum_{k_2=\alpha_2}^m \cdots \sum_{\alpha_l=0}^m \sum_{k_l=\alpha_l}^m \cdots \sum_{\alpha_n=0}^m \sum_{k_n=\alpha_n}^m \sum_{\beta_1=0}^m \sum_{\beta_2=0}^m \cdots \sum_{\beta_n=0}^m (\beta_i - \alpha_i) P_{\alpha \rightarrow \beta} G(k, \alpha, m) \prod_{j=1}^n p_j^{k_j} \quad (\text{A.22})$$

Notons qu'en terme de variables dépendantes, toutes les sommes de (A.22) apparaissent par groupes de deux ou moins, *et seulement par groupes de deux ou moins*. Fixons notre attention sur la paire de sommes dont les variables en jeu sont k_l et α_l , $0 \leq l \leq n$, soit la paire :

$$\sum_{\alpha_l=0}^m \sum_{k_l=\alpha_l}^m \quad (\text{A.23})$$

La grille de points $R \subset S^2$ sur laquelle s'étend cette paire de somme peut se représenter car la coupe résultante de $S^2 \cap S^n$ nous est géométriquement accessible. Cette coupe est illustrée

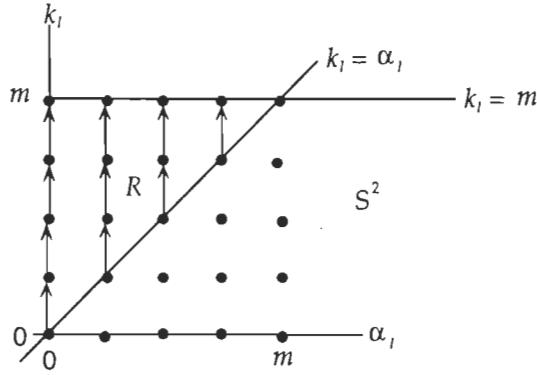


Fig A.1 Région R de sommation correspondante à (A.23)

à la figure A.1. De cette figure A.1 on déduit aisément une équivalence pour (A.23) c'est-à-dire :

$$\sum_{\alpha_l=0}^m \sum_{k_l=\alpha_l}^m \equiv \sum_{k_l=0}^m \sum_{\alpha_l=0}^{k_l} \quad (A.24)$$

En répétant n fois ce processus d'inversion des k_l et α_l , $0 \leq k_l \leq m$, $0 \leq \alpha_l \leq k_l$ l'expression (A.22) prend la forme suivante :

$$\sum_{\alpha_1=0}^m \sum_{k_1=\alpha_1}^m \cdots \sum_{\alpha_2=0}^m \sum_{k_2=\alpha_2}^m \cdots \sum_{\alpha_l=0}^m \sum_{k_l=\alpha_l}^m \cdots \sum_{\alpha_n=0}^m \sum_{k_n=\alpha_n}^m \sum_{\beta_1=0}^m \sum_{\beta_2=0}^m \cdots \sum_{\beta_n=0}^m (\beta_i - \alpha_i) P_{\alpha \rightarrow \beta} G(\mathbf{k}, \boldsymbol{\alpha}, m) \prod_{j=1}^n \rho_j^{k_j} \quad (A.25)$$

$$= \sum_{k_1=0}^m \sum_{\alpha_1=0}^{k_1} \cdots \sum_{k_2=0}^m \sum_{\alpha_2=0}^{k_2} \cdots \sum_{k_l=0}^m \sum_{\alpha_l=0}^{k_l} \cdots \sum_{k_n=0}^m \sum_{\alpha_n=0}^{k_n} \sum_{\beta_1=0}^m \sum_{\beta_2=0}^m \cdots \sum_{\beta_n=0}^m (\beta_i - \alpha_i) P_{\alpha \rightarrow \beta} G(\mathbf{k}, \boldsymbol{\alpha}, m) \prod_{j=1}^n \rho_j^{k_j} \quad (A.26)$$

En déplaçant vers la gauche toutes les sommes de (A.26) pour k_l , $0 \leq k_l \leq m$, on a :

$$= \sum_{k_1=0}^m \sum_{k_2=0}^m \cdots \sum_{k_l=0}^m \cdots \sum_{k_n=0}^m \sum_{\alpha_1=0}^{k_1} \sum_{\alpha_2=0}^{k_2} \cdots \sum_{\alpha_l=0}^{k_l} \cdots \sum_{\alpha_n=0}^{k_n} \sum_{\beta_1=0}^m \sum_{\beta_2=0}^m \cdots \sum_{\beta_n=0}^m (\beta_i - \alpha_i) P_{\alpha \rightarrow \beta} G(\mathbf{k}, \boldsymbol{\alpha}, m) \prod_{j=1}^n \rho_j^{k_j} \quad (A.27)$$

En utilisant la notation convenue pour les sommes imbriquées, l'égalité (A.19) prend finalement la forme :

$$\sum_{l=0}^m a_l^{(i)} \prod_{j=1}^n (\rho_j)^{l_j} = \sum_{k=0}^m \sum_{\alpha=0}^k \sum_{\beta=0}^m (\beta_i - \alpha_i) P_{\alpha \rightarrow \beta} G(\mathbf{k}, \boldsymbol{\alpha}, m) \prod_{j=1}^n \rho_j^{k_j} \quad (A.28)$$

Dans (A.28) l'indice de sommation k est muet. L'affectation $l := k$ est suffisante pour conclure de l'égalité entre ces deux polynômes. En effet, l'affectation $l := k$ dans (A.28) conduit à :

$$\sum_{l=0}^m a_l^{(i)} \prod_{j=1}^n (\rho_j)^{l_j} = \sum_{l=0}^m \sum_{\alpha=0}^l \sum_{\beta=0}^m (\beta_i - \alpha_i) P_{\alpha \rightarrow \beta} G(l, \alpha, m) \prod_{j=1}^n \rho_j^{l_j} \quad (\text{A.29})$$

De (A.29) on peut à présent déduire la valeur de $a_l^{(i)}$ c'est-à-dire :

$$a_l^{(i)} = \sum_{\alpha=0}^l \sum_{\beta=0}^m (\beta_i - \alpha_i) P_{\alpha \rightarrow \beta} G(l, \alpha, m) \quad (\text{A.30})$$

La quantité scalaire $P_{\alpha \rightarrow \beta}$ semble emprisonnée à l'intérieur des sommes de (A.30). Rappelons que le but de ce bref traitement est d'essayer d'établir le lien le plus simple qui soit entre $a_l^{(i)}$ et $P_{\alpha \rightarrow \beta}$. En principe, nous pourrions nous arrêter à ce stade, mais le développement itéré de (A.30) n'est pas très commode. Il serait davantage intéressant que ce soit les coefficients connus $a_l^{(i)}$ qui soient placés sous le signe de sommation. À présent, voyons comment on peut inverser la relation entre $a_l^{(i)}$ et $P_{\alpha \rightarrow \beta}$. Reprenons le travail en dépliant les sommes du membre de droite de (A.30).

On obtient :

$$a_l^{(i)} = \sum_{\alpha_1=0}^{l_1} \sum_{\alpha_2=0}^{l_2} \dots \sum_{\alpha_n=0}^{l_n} \sum_{\beta_1=0}^m \sum_{\beta_2=0}^m \dots \sum_{\beta_n=0}^m (\beta_i - \alpha_i) P_{\alpha \rightarrow \beta} G(l, \alpha, m) \quad (\text{A.31})$$

Rappelons que les transformations d'états du système physique se font de composante à composante dans le n -uplet qui représente cette transformation. Le diagramme fléché $\alpha \rightarrow \beta$ implique, en réalité, les n sous-diagrammes $\alpha_1 \rightarrow \beta_1, \alpha_2 \rightarrow \beta_2, \dots, \alpha_n \rightarrow \beta_n$. Ces derniers sous-diagrammes nous indiquent, en particulier, que lors de la transformation physique, la variable d'état $\alpha_k \in S$ n'est liée qu'à la variable d'état $\beta_k \in S$, $0 \leq k \leq n$. Le regroupement des sommes, sur les paires de variable α_k et β_k , $0 \leq k \leq n$, devient donc une façon symbolique fort intéressante de contenir cette dernière information. De (A.31) on a alors :

$$a_l^{(i)} = \sum_{\beta_1=0}^m \sum_{\alpha_1=0}^{l_1} \sum_{\beta_2=0}^m \sum_{\alpha_2=0}^{l_2} \dots \sum_{\beta_k=0}^m \sum_{\alpha_k=0}^{l_k} \dots \sum_{\beta_n=0}^m \sum_{\alpha_n=0}^{l_n} (\beta_i - \alpha_i) P_{\alpha \rightarrow \beta} G(l, \alpha, m) \quad (\text{A.32})$$

Fixons, encore une fois, notre attention sur la paire de sommes de (A.32) dont les variables en jeu sont α_k et β_k , $0 \leq k \leq n$, soit la paire :

$$\sum_{\beta_k=0}^m \sum_{\alpha_k=0}^{l_k} \quad (\text{A.33})$$

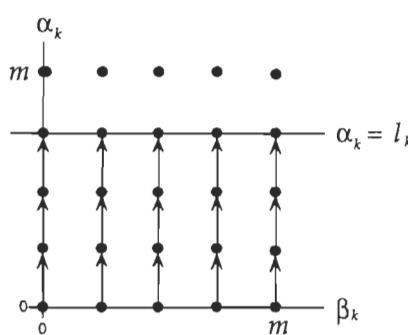


Fig A.2 Région R de sommation correspondante à (A.33)

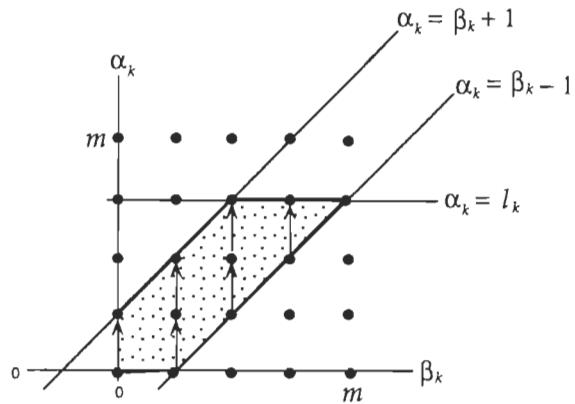


Fig A.3 Région R' de sommation correspondante à (A.33) sous la contrainte $|\alpha_k - \beta_k| \leq 1$

Il est clair à présent que la grille de points $R \subseteq S^2$ sur laquelle s'étend la paire de sommes de (A.33) nous est, elle aussi, géométriquement accessible. Cette coupe est illustrée à la figure A.2. Nous allons apporter maintenant une contrainte sur la transformation $\alpha_k \rightarrow \beta_k$. Les mécanismes de collisions que nous allons utiliser n'admettent pas de transformations dont les sauts soient plus grands que un. Dans le présent contexte, cette cela se traduit ainsi : $\forall k, 0 \leq k \leq n, |\alpha_k - \beta_k| \leq 1$. Cette dernière contrainte donne lieu à une nouvelle région $R' \subset R$ qui est illustrée à la figure A.3. Sur cette figure on voit que la région R' contient une irrégularité en $(0, 0)$ en ce sens que le couple de valeurs $\beta_k = -1$ et $\alpha_k = 0$ n'est pas inclus dans la bande ombragée, lorsque $\alpha_k = l_k \leq m$. Une autre irrégularité se produit lorsque $\alpha_k = l_k = m$ puisqu'à ce moment c'est le couple de valeurs $\beta_k = m + 1$ et $\alpha_k = m$ qui se retrouve exclu de la bande ombragée. Nous allons tout de même introduire ces deux points sur la région de sommation. L'ajout artificiel de ces points est immédiatement pardonné si l'on remarque que pour ces états $P_{\alpha \rightarrow \beta} = 0$. Ceci étant dit, l'égalité (A.32) ne sera donc pas entaché d'erreurs par l'introduction de ces points. La figure A.4 résume l'ensemble des propos que nous venons de tenir. Sur cette figure les deux points ajoutés ont été encerclés afin de signaler que leurs introductions suppriment les effets de bords auxquels nous étions confrontés avant de les introduire.

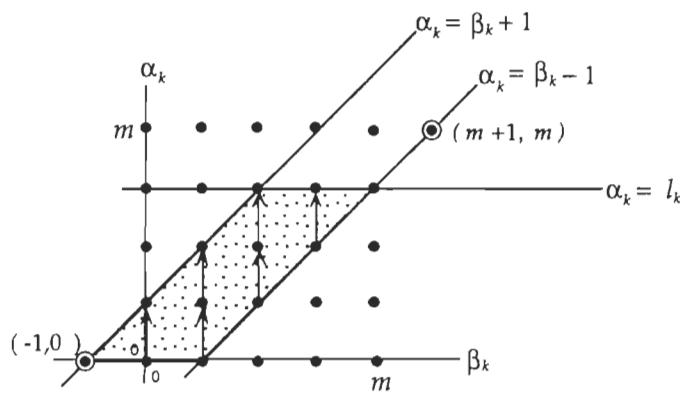


Fig A.4 Ajout de 2 points pour régulariser la région R de (A.33)

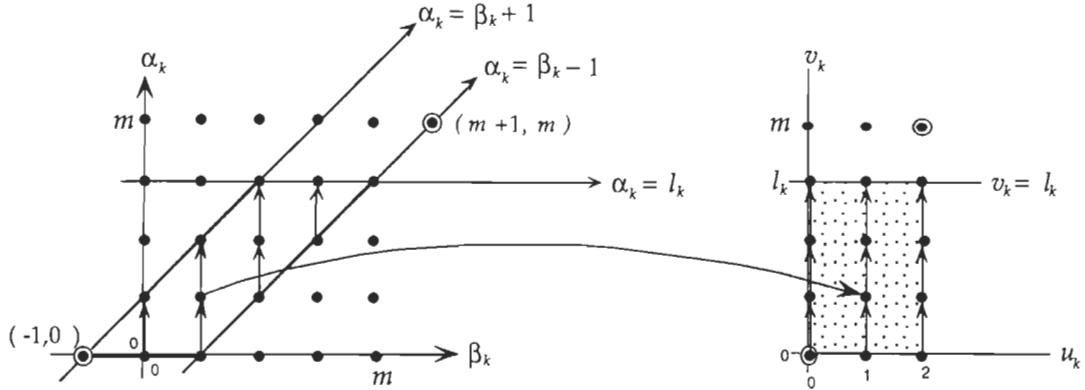


Fig A.5 Effet des changements de variables (A.34) à (A.37) sur la région R^*

Les changements de variables : $\beta_k = u_k + v_k - 1$ et $\alpha_k = v_k$ conduisent à :

$$\beta_k = \alpha_k - 1 \Rightarrow u_k + v_k - 1 = v_k - 1 \Rightarrow u_k = 0 \quad (\text{A.34})$$

$$\beta_k = \alpha_k + 1 \Rightarrow u_k + v_k - 1 = v_k + 1 \Rightarrow u_k = 2 \quad (\text{A.35})$$

$$\alpha_k = 0 \Rightarrow 0 = v_k \Rightarrow v_k = 0 \quad (\text{A.36})$$

$$\alpha_k = l_k \Rightarrow l_k = v_k \Rightarrow v_k = l_k \quad (\text{A.37})$$

Cette transformation produit l'effet illustrée à la figure A.5. Rappelons qu'avant d'introduire ces changements de variables on avait obtenu en (A.32) :

$$a_l^{(i)} = \sum_{\beta_1=0}^m \sum_{\alpha_1=0}^{l_1} \sum_{\beta_2=0}^m \sum_{\alpha_2=0}^{l_2} \dots \sum_{\beta_k=0}^m \sum_{\alpha_k=0}^{l_k} \dots \sum_{\beta_n=0}^m \sum_{\alpha_n=0}^{l_n} (\beta_i - \alpha_i) P_{\alpha \rightarrow \beta} G(l, \alpha, m) \quad (\text{A.38})$$

Après avoir introduit les variables u_k et v_k dans (A.38) pour $k, 0 \leq k \leq n$, l'égalité devient :

$$a_l^{(i)} = \sum_{u_1=0}^2 \sum_{v_1=0}^{l_1} \sum_{u_2=0}^2 \sum_{v_2=0}^{l_2} \dots \sum_{u_k=0}^2 \sum_{v_k=0}^{l_k} \dots \sum_{u_n=0}^2 \sum_{v_n=0}^{l_n} (v_i + u_i - 1 - v_i) P_{v \rightarrow v+u-1} G(l, v, m) \quad (\text{A.39})$$

$$\Rightarrow a_l^{(i)} = \sum_{v_1=0}^{l_1} \sum_{v_2=0}^{l_2} \dots \sum_{v_n=0}^{l_n} \sum_{u_1=0}^2 \sum_{u_2=0}^2 \dots \sum_{u_n=0}^2 (u_i - 1) P_{v \rightarrow v+u-1} G(l, v, m) \quad (\text{A.40})$$

$$\Rightarrow a_l^{(i)} = \sum_{v=0}^l \sum_{u=0}^2 (u_i - 1) P_{v \rightarrow v+u-1} G(l, v, m) \quad (\text{A.41})$$

$$\Rightarrow \quad a_l^{(i)} = \sum_{v=0}^l G(l, v, m) \sum_{u=0}^2 (u_i - 1) P_{v \rightarrow v+u-1} \quad (\text{A.42})$$

Dans le but de rendre l'expression (A.42) plus compacte, posons :

$$P(v) = \sum_{u=0}^2 (u_i - 1) P_{v \rightarrow v+u-1} \quad (\text{A.43})$$

$$a_l^{(i)} = \sum_{v=0}^l G(l, v, m) P(v) \quad (\text{A.44})$$

Indiquons toutefois que :

$$G(k, \alpha, m) = G(k_1, k_2, \dots, k_n, \alpha_1, \alpha_2, \dots, \alpha_n, m) = \prod_{j=1}^n \left(\frac{(-1)^{k_j - \alpha_j}}{m^{k_j}} \right) \binom{m}{k_j} \binom{k_j}{\alpha_j} \quad (\text{A.45})$$

En vertu de (A.45) le terme $G(l, v, m)$ de (A.42) vaut donc :

$$G(l, v, m) = \prod_{j=1}^n \left(\frac{(-1)^{l_j - v_j}}{m^{l_j}} \right) \binom{m}{l_j} \binom{l_j}{v_j} = \prod_{j=1}^n \left[\frac{1}{m^{l_j}} \binom{m}{l_j} \right] \prod_{j=1}^n \left[(-1)^{l_j - v_j} \binom{l_j}{v_j} \right] \quad (\text{A.46})$$

La substitution de (A.46) dans (A.42) conduit à :

$$a_l^{(i)} = \sum_{v=0}^l \left[\prod_{j=1}^n \left[\frac{1}{m^{l_j}} \binom{m}{l_j} \right] \prod_{j=1}^n \left[(-1)^{l_j - v_j} \binom{l_j}{v_j} \right] \right] P(v) \quad (\text{A.47})$$

$$\Rightarrow \quad a_l^{(i)} = \prod_{j=1}^n \left[\frac{1}{m^{l_j}} \binom{m}{l_j} \right] \left[\sum_{v=0}^l \prod_{j=1}^n (-1)^{l_j - v_j} \binom{l_j}{v_j} \right] P(v) \quad (\text{A.48})$$

$$\Rightarrow \quad \prod_{j=1}^n m^{l_j} / \binom{m}{l_j} a_l^{(i)} = \sum_{v=0}^l \prod_{j=1}^n (-1)^{l_j - v_j} \binom{l_j}{v_j} P(v) \quad (\text{A.49})$$

Posons :

$$A_l^{(i)} = \prod_{j=1}^n m^{l_j} / \binom{m}{l_j} a_l^{(i)} = a_l^{(i)} \prod_{j=1}^n m^{l_j} / \binom{m}{l_j} \quad (\text{A.50})$$

La substitution de (A.50) dans (A.49) conduit à l'encadré suivant :

$$A_l^{(i)} = \sum_{v=0}^l \prod_{j=1}^n (-1)^{l_j - v_j} \binom{l_j}{v_j} P(v) \quad (\text{A.51})$$

Pour parvenir à inverser la relation entre $A_l^{(i)}$ et $P(v)$ nous aurons recours au lemme suivant :

Lemme

Soit $\{f_v\}_{v \in S^n}$ et $\{g_v\}_{v \in S^n}$ deux suites de nombres réels telle que $f_l = \sum_{v=0}^l \prod_{j=1}^n (-1)^{l_j - v_j} \binom{l_j}{v_j} g_v$.
Alors :

$$f_l = \sum_{v=0}^l \prod_{j=1}^n (-1)^{l_j - v_j} \binom{l_j}{v_j} g_v \Leftrightarrow g_l = \sum_{v=0}^l \prod_{j=1}^n \binom{l_j}{v_j} f_v$$

Nous allons montrer le lemme (A.52) par induction sur n . Dans le cadre de la démonstration par induction nous aurons besoin d'utiliser le corollaire suivant :

Corollaire

Soit $\{f_k\}_{k=0}^n$ et $\{g_k\}_{k=0}^n$ deux suites de nombres réels telle que $f_n = \sum_{k=0}^n (-1)^{n-k} \binom{n}{k} g_k$.
Alors :

$$f_n = \sum_{k=0}^n (-1)^{n-k} \binom{n}{k} g_k \Leftrightarrow g_n = \sum_{k=0}^n \binom{n}{k} f_k$$

Base de l'induction.

Si $n = 1$ notons que $\mathbf{l} = (l_1)$ et $\mathbf{v} = (v_1) = v$. Comme ces n -uplets n'ont qu'une seule composante nous laisserons tomber, pour plus de simplicité, les indices 1 des variables l_1 et v_1 . En vertu de ces considérations il faut se demander si la double implication que propose (A.52) est vraie; soit :

$$f_l = \sum_{v=0}^l (-1)^{l-v} \binom{l}{v} g_v \Leftrightarrow g_l = \sum_{v=0}^l \binom{l}{v} f_v \quad (\text{A.54})$$

Cette double implication coïncide avec le corollaire (A.53). Ce dernier corollaire est établi dans le livre *Advanced Combinatorics*²⁹.

Hypothèse d'induction.

Nous supposerons vrai pour n la double implication :

$$f_l = \sum_{v=0}^l \prod_{j=1}^n (-1)^{l_j - v_j} \binom{l_j}{v_j} g_v \Leftrightarrow g_l = \sum_{v=0}^l \prod_{j=1}^n \binom{l_j}{v_j} f_v \quad (\text{A.55})$$

Pas d'induction.

Voyons si l'hypothèse d'induction (A.55) est aussi vraie pour $n + 1$. Notons que pour $n + 1$ les $n+1$ -uplets l et v deviennent $l = (l_1, l_2, \dots, l_n, l_{n+1})$ et $v = (v_1, v_2, \dots, v_n, v_{n+1})$.

Ceci étant dit, de (A.55) on peut alors écrire :

$$f_{l_1, l_2, \dots, l_n, l_{n+1}} = \sum_{v_1=0}^{l_1} \sum_{v_2=0}^{l_2} \dots \sum_{v_n=0}^{l_n} \sum_{v_{n+1}=0}^{l_{n+1}} \prod_{j=1}^{n+1} (-1)^{l_j - v_j} \binom{l_j}{v_j} g_{v_1, v_2, \dots, v_n, v_{n+1}} \quad (\text{A.56})$$

$$\Rightarrow f_{l_1, l_2, \dots, l_n, l_{n+1}} = \sum_{v_1=0}^{l_1} \sum_{v_2=0}^{l_2} \dots \sum_{v_n=0}^{l_n} \sum_{v_{n+1}=0}^{l_{n+1}} \prod_{j=1}^n (-1)^{l_j - v_j} \binom{l_j}{v_j} (-1)^{l_{n+1} - v_{n+1}} \binom{l_{n+1}}{v_{n+1}} g_{v_1, v_2, \dots, v_n, v_{n+1}} \quad (\text{A.57})$$

Comme $\forall k, 1 \leq k \leq n+1, l_k \in \{0, \dots, m\}$, $m \in \mathbb{N}$ alors (A.57) s'écrit comme :

$$f_{l_1, l_2, \dots, l_n, l_{n+1}} = \sum_{v_{n+1}=0}^{l_{n+1}} \sum_{v_1=0}^{l_1} \sum_{v_2=0}^{l_2} \dots \sum_{v_n=0}^{l_n} (-1)^{l_{n+1} - v_{n+1}} \binom{l_{n+1}}{v_{n+1}} \prod_{j=1}^n (-1)^{l_j - v_j} \binom{l_j}{v_j} g_{v_1, v_2, \dots, v_n, v_{n+1}} \quad (\text{A.58})$$

$$f_{l_1, l_2, \dots, l_n, l_{n+1}} = \sum_{v_{n+1}=0}^{l_{n+1}} (-1)^{l_{n+1} - v_{n+1}} \binom{l_{n+1}}{v_{n+1}} \sum_{v_1=0}^{l_1} \sum_{v_2=0}^{l_2} \dots \sum_{v_n=0}^{l_n} \prod_{j=1}^n (-1)^{l_j - v_j} \binom{l_j}{v_j} g_{v_1, v_2, \dots, v_n, v_{n+1}} \quad (\text{A.59})$$

Fixons notre attention sur les n dernières sommes du membre de droite de (A.59). Le résultat de ces n sommes constituerait, s'il fallait en avoir complété le calcul, une suite intermédiaire g^* qui serait à présent indiqué par :

$$g_{l_1, l_2, \dots, l_n, v_{n+1}}^* = \sum_{v_1=0}^{l_1} \sum_{v_2=0}^{l_2} \dots \sum_{v_n=0}^{l_n} \prod_{j=1}^n (-1)^{l_j - v_j} \binom{l_j}{v_j} g_{v_1, v_2, \dots, v_n, v_{n+1}} \quad (\text{A.60})$$

Cette dernière remarque nous permet d'écrire (A.59) de la sorte :

$$f_{l_1, l_2, \dots, l_n, l_{n+1}} = \sum_{v_{n+1}=0}^{l_{n+1}} (-1)^{l_{n+1} - v_{n+1}} \binom{l_{n+1}}{v_{n+1}} g_{l_1, l_2, \dots, l_n, v_{n+1}}^* \quad (\text{A.61})$$

À présent toutes les variables sont fixées sauf la variable v_{n+1} . De plus, notons que la somme en (A.61) ne s'effectue que sur v_{n+1} . Dans ce contexte le corollaire (A.53) s'utilise de sorte que :

$$g_{l_1, l_2, \dots, l_n, l_{n+1}}^* = \sum_{v_{n+1}=0}^{l_{n+1}} \binom{l_{n+1}}{v_{n+1}} f_{l_1, l_2, \dots, l_n, v_{n+1}} \quad (\text{A.62})$$

D'une part, nous avions en (A.60) :

$$g_{l_1, l_2, \dots, l_n, v_{n+1}}^* = \sum_{v_1=0}^{l_1} \sum_{v_2=0}^{l_2} \dots \sum_{v_n=0}^{l_n} \prod_{j=1}^n (-1)^{l_j - v_j} \binom{l_j}{v_j} g_{v_1, v_2, \dots, v_n, v_{n+1}} \quad (\text{A.63})$$

En appliquant la composition $v_{n+1} = l_{n+1}$ sur la suite $g_{l_1, l_2, \dots, l_n, v_{n+1}}^*$ on obtient de (A.63) :

$$g_{l_1, l_2, \dots, l_n, l_{n+1}}^* = \sum_{v_1=0}^{l_1} \sum_{v_2=0}^{l_2} \dots \sum_{v_n=0}^{l_n} \prod_{j=1}^n (-1)^{l_j - v_j} \binom{l_j}{v_j} g_{v_1, v_2, \dots, v_n, l_{n+1}} \quad (\text{A.64})$$

D'autre part, on a aussi établi en (A.62) :

$$g_{l_1, l_2, \dots, l_n, l_{n+1}}^* = \sum_{v_{n+1}=0}^{l_{n+1}} \binom{l_{n+1}}{v_{n+1}} f_{l_1, l_2, \dots, l_n, v_{n+1}} \quad (\text{A.65})$$

De (A.64) et (A.65) on a donc :

$$\sum_{v_{n+1}=0}^{l_{n+1}} \binom{l_{n+1}}{v_{n+1}} f_{l_1, l_2, \dots, l_n, v_{n+1}} = \sum_{v_1=0}^{l_1} \sum_{v_2=0}^{l_2} \dots \sum_{v_n=0}^{l_n} \prod_{j=1}^n (-1)^{l_j - v_j} \binom{l_j}{v_j} g_{v_1, v_2, \dots, v_n, l_{n+1}} \quad (\text{A.66})$$

L'indice $n+1$ de la variable l_{n+1} est rendu non-essentiel car il n'apparaît plus aux bornes supérieures des sommes du membre de droite de l'égalité. Pour bien signaler ce fait posons $a = l_{n+1}$.

$$\sum_{v_{n+1}=0}^a \binom{a}{v_{n+1}} f_{l_1, l_2, \dots, l_n, v_{n+1}} = \sum_{v_1=0}^{l_1} \sum_{v_2=0}^{l_2} \dots \sum_{v_n=0}^{l_n} \prod_{j=1}^n (-1)^{l_j - v_j} \binom{l_j}{v_j} g_{v_1, v_2, \dots, v_n, a} \quad (\text{A.67})$$

Dans le but de faire ressortir l'essentiel de la somme que propose le membre de gauche de (A.67), posons :

$$f_{l_1, l_2, \dots, l_n, a}^* = \sum_{v_{n+1}=0}^a \binom{a}{v_{n+1}} f_{l_1, l_2, \dots, l_n, v_{n+1}} \quad (\text{A.68})$$

La substitution de (A.68) dans (A.67) conduit au résultat suivant :

$$f_{l_1, l_2, \dots, l_n, a}^* = \sum_{v_1=0}^{l_1} \sum_{v_2=0}^{l_2} \dots \sum_{v_n=0}^{l_n} \prod_{j=1}^n (-1)^{l_j - v_j} \binom{l_j}{v_j} g_{v_1, v_2, \dots, v_n, a} \quad (\text{A.69})$$

Le paramètre a est fixe. Rappelons que $a \in \{0, \dots, m\}$ $m \in \mathbb{N}$. Remarquons alors que les véritables variables d'intérêt, soit celles qui apparaissent aux bornes supérieures des sommes de (A.69), correspondent aux éléments du n -uplet $l' = (l_1, l_2, \dots, l_n)$. C'est dans ce contexte que l'on va plutôt écrire : $F_{l_1, l_2, \dots, l_n} = f_{l_1, l_2, \dots, l_n, a}^*$ et $G_{v_1, v_2, \dots, v_n} = g_{v_1, v_2, \dots, v_n, a}$. De ces remarques :

$$F_{l_1, l_2, \dots, l_n} = \sum_{v_1=0}^{l_1} \sum_{v_2=0}^{l_2} \dots \sum_{v_n=0}^{l_n} \prod_{j=1}^n (-1)^{l_j - v_j} \binom{l_j}{v_j} G_{v_1, v_2, \dots, v_n} \quad (\text{A.70})$$

Par *hypothèse d'induction* on peut alors écrire :

$$G_{l_1, l_2, \dots, l_n} = \sum_{v_1=0}^{l_1} \sum_{v_2=0}^{l_2} \dots \sum_{v_n=0}^{l_n} \prod_{j=1}^n \binom{l_j}{v_j} F_{v_1, v_2, \dots, v_n} \quad (\text{A.71})$$

$$G_{v_1, v_2, \dots, v_n} = g_{v_1, v_2, \dots, v_n, a} \Rightarrow G_{l_1, l_2, \dots, l_n} = g_{l_1, l_2, \dots, l_n, a} \quad (\text{A.72})$$

$$F_{l_1, l_2, \dots, l_n} = f_{l_1, l_2, \dots, l_n, a}^* \Rightarrow F_{v_1, v_2, \dots, v_n} = f_{v_1, v_2, \dots, v_n, a}^* = \sum_{v_{n+1}=0}^a \binom{a}{v_{n+1}} f_{v_1, v_2, \dots, v_n, a} \quad (\text{A.73})$$

Or comme :

$$G_{l_1, l_2, \dots, l_n} = \sum_{v_1=0}^{l_1} \sum_{v_2=0}^{l_2} \dots \sum_{v_n=0}^{l_n} \prod_{j=1}^n \binom{l_j}{v_j} F_{v_1, v_2, \dots, v_n} \quad (\text{A.74})$$

$$g_{l_1, l_2, \dots, l_n, a} = \sum_{v_1=0}^{l_1} \sum_{v_2=0}^{l_2} \dots \sum_{v_n=0}^{l_n} \prod_{j=1}^n \binom{l_j}{v_j} f_{v_1, v_2, \dots, v_n, a}^* \quad (\text{A.75})$$

$$g_{l_1, l_2, \dots, l_n, a} = \sum_{v_1=0}^{l_1} \sum_{v_2=0}^{l_2} \dots \sum_{v_n=0}^{l_n} \prod_{j=1}^n \binom{l_j}{v_j} \sum_{v_{n+1}=0}^a \binom{a}{v_{n+1}} f_{v_1, v_2, \dots, v_n, a} \quad (\text{A.76})$$

En rappelant que nous avions posé $a = l_{n+1}$ on obtient de (A.76) :

$$g_{l_1, l_2, \dots, l_n, l_{n+1}} = \sum_{v_1=0}^{l_1} \sum_{v_2=0}^{l_2} \dots \sum_{v_n=0}^{l_n} \prod_{j=1}^n \binom{l_j}{v_j} \sum_{v_{n+1}=0}^{l_{n+1}} \binom{l_{n+1}}{v_{n+1}} f_{v_1, v_2, \dots, v_n, l_{n+1}} \quad (\text{A.77})$$

En utilisant les opérations sur les sommes et les produits on obtient finalement :

$$g_{l_1, l_2, \dots, l_n, l_{n+1}} = \sum_{v_1=0}^{l_1} \sum_{v_2=0}^{l_2} \dots \sum_{v_n=0}^{l_n} \sum_{v_{n+1}=0}^{l_{n+1}} \prod_{j=1}^n \binom{l_j}{v_j} \binom{l_{n+1}}{v_{n+1}} f_{v_1, v_2, \dots, v_n, l_{n+1}} \quad (\text{A.78})$$

$$\Rightarrow g_{l_1, l_2, \dots, l_n, l_{n+1}} = \sum_{v_1=0}^{l_1} \sum_{v_2=0}^{l_2} \dots \sum_{v_n=0}^{l_n} \sum_{v_{n+1}=0}^{l_{n+1}} \prod_{j=1}^{n+1} \binom{l_j}{v_j} f_{v_1, v_2, \dots, v_n, l_{n+1}} \quad (\text{A.79})$$

Selon la notation convenue :

$$g_l = \sum_{v=0}^l \prod_{j=1}^{n+1} \binom{l_j}{v_j} f_v \quad (\text{A.80})$$

Notons ici que $l = (l_1, l_2, \dots, l_n, l_{n+1})$ et $v = (v_1, v_2, \dots, v_n, v_{n+1})$. Cette dernière remarque est suffisante pour conclure la preuve par induction du lemme (A.52).

On voit très rapidement, en inspectant (A.52), pourquoi ce lemme nous est d'une précieuse utilité.

En appliquant (A.52) à (A.51) nous obtenons finalement le résultat convoité soit :

$$P(l) = \sum_{v=0}^l \prod_{j=1}^n \binom{l_j}{v_j} A_v^{(i)} \quad (\text{A.81})$$

En rappelant que l'on a posé :

$$P(v) = \sum_{u=0}^2 (u_i - 1) P_{v \rightarrow v+u-1} \Rightarrow P(l) = \sum_{u=0}^2 (u_i - 1) P_{l \rightarrow l+u-1} \quad (\text{A.82})$$

$$A_l^{(i)} = a_l^{(i)} \prod_{j=1}^n m^{l_j} / \binom{m}{l_j} \Rightarrow A_v^{(i)} = a_v^{(i)} \prod_{j=1}^n m^{v_j} / \binom{m}{v_j} \quad (\text{A.83})$$

En vertu des implications (A.81) à (A.83) on obtient :

$$P(l) = \sum_{u=0}^2 (u_i - 1) P_{l \rightarrow l+u-1} = \sum_{v=0}^l \prod_{j=1}^n \binom{l_j}{v_j} a_v^{(i)} \prod_{j=1}^n m^{v_j} \binom{m}{v_j} = \sum_{v=0}^l \prod_{j=1}^n \frac{\binom{l_j}{v_j} m^{v_j}}{\binom{m}{v_j}} a_v^{(i)} \quad (\text{A.84})$$

En (A.84), l'indice l est muet. Dans le but de rendre l'interprétation de ce dernier résultat plus près du problème physique traité, posons respectivement $\alpha = l$. De ces remarques l'égalité (A.84) s'écrit aussi comme :

$$\sum_{u=0}^2 (u_i - 1) P_{\alpha \rightarrow \alpha + u - 1} = \sum_{v=0}^{\alpha} \prod_{j=1}^n \frac{\binom{\alpha_j}{v_j} m^{v_j}}{\binom{m}{v_j}} a_v^{(i)} \quad (\text{A.85})$$

Finalement, en rappelant que l'affectation $a_v^{(i)} := h a_v^{(i)}$ a été effectué en (A.6), il nous faut à présent effectuer l'affectation inverse $h a_v^{(i)} := a_v^{(i)}$ dans (A.85). On obtient donc l'encadré suivant :

$$\sum_{u=0}^2 (u_i - 1) P_{\alpha \rightarrow \alpha + u - 1} = h \sum_{v=0}^{\alpha} \prod_{j=1}^n \frac{\binom{\alpha_j}{v_j} m^{v_j}}{\binom{m}{v_j}} a_v^{(i)} \quad \blacksquare$$

(A.86)

Le travail fait au cours de l'annexe A se résume par la proposition (2.32).

Annexe B

PROGRAMME INFORMATIQUE « ÉCOSYSTÈME »

```

{ ****
* Ce programme simule l'evolution d'une seule espece *
**** }

program ecosysteme;

uses graph,dos,crt;

const
  a_0 = 0.0;
  a_1 = 1.0;
  a_2 = -1.0;
  a_3 = 0.0;
  a_4 = 0.0;
  trouve_temps_ecelle=false;      temps_total=640;
  delta_temps_ecelle=0;          dimension=64;
  temps_ecelle:extended = 20;    nb_essai=1;
  temps_relaxation:longint=1;    taille_echantillon=640;
  densite_lapin_genere= 0.1;    nb_individu=2;
  densite_loup_genere=0.0;       nb_direction=4;

type

lesbords = (haut,bas,gauche,droite);
les_etres = (loup,lapin);
estimateur = (min,moy,s_xbar,max);
etat_noeud = ( naissance,meurt,extra_specifique,intra_specifique );
contour = array[ lesbords , 0..dimension+1 ] of integer;
vectboolean = array[ les_etres ] of boolean;
vecteur = array[ les_etres ] of longint;
racine = array[ les_etres , 1..nb_direction ] of extended;
echantillon = array[ 1..nb_essai,les_etres,estimateur ] of extended;
automate = array[ les_etres , 0..4 , 0..4 ] of extended;
prob = array[ les_etres , 0..nb_direction ] of extended;
graphe = array[ 1..nb_essai ,les_etres, 1..taille_echantillon ] of longint;
vecteur211=array[0..dimension+1] of integer;
vecteur1=array[0..dimension+1] of ^vecteur211;
vecteur21=array[0..3] of ^vecteur211;
vecteur2=vecteur21;
phrase=array[estimateur] of string;

var

bords:contour;
manger_par_le:vectboolean;
foret,ecran:vecteur1;
stat_etre:echantillon;
courbe:graphe;
grmode:integer;
no_essai,j,temp,nb_etre_max:longint;
nb_etre,tampon,delta_etre,etat_etre:vecteur;
kappa:prob;
transition:automate;
stabilite:racine;
remarque:phrase;
fichier,fichier_stat:text;
graphique_ok,debut,transi_ok:boolean;

```

```

{*****
 * Cette procedure test l'accessibilite *
 * de la carte graphique de l'ordinateur. *
*****}

procedure verifigraphe(var grapheok:boolean);
var grdriver,errcode:integer;
begin
grdriver:=detect;
initgraph(grdriver,grmode,'c:\tp\bg1');
errcode:=graphresult;
if errcode = grOk then
begin
grapheok:=true;
end
else
begin
writeln('Erreur graphique:', grapherrmsg(errcode));
grapheok:=false;
readln;
end;
end;

{*****
 * Initialisation des structures de donnees *
*****}

procedure init_etiquette(var etiquette:vecteur2;premier:byte);
var i: word;
begin
if premier=0 then
begin
new(etiquette[0]);
new(etiquette[1]);
new(etiquette[2]);
new(etiquette[3]);
end;
for i:=0 to dimension+1 do etiquette[0]^:=0;
etiquette[1]^:=etiquette[0]^;
etiquette[2]^:=etiquette[0]^;
etiquette[3]^:=etiquette[0]^;
end;

procedure init_foret(var foret:vecteur1;premier:byte);
var i:integer;
begin
if premier=0 then
begin
for i:=0 to dimension+1 do new(foret[i]);
end;
for i:=0 to dimension+1 do foret[0]^:=0;
for i:=0 to dimension+1 do foret[i]^:=foret[0]^;
end;

```

```

procedure init_bords(var bords:contour);
var i:integer;
  cote:lesbords;
begin
  for cote:=haut to droite do
  begin
    for i:=0 to dimension+1 do
    begin
      bords[cote,i]:=0;
    end;
  end;
end;

procedure init_transition(var transition:automate);
var j,k:integer;
  i:les_etres;
begin
  for i:=loup to lapin do
  for j:=0 to nb_direction do
    for k:=0 to nb_direction do transition[i,j,k]:=0;
  end;

procedure init_stable(var stabilite:racine);
var j:integer;
  i:les_etres;
begin
  for i:=loup to lapin do
    for j:=1 to nb_direction do stabilite[i,j]:=0;
  end;

procedure init_kappa(var kappa:prob);
begin
  kappa[lapin,0]:=a_0;
  kappa[lapin,1]:=a_1;
  kappa[lapin,2]:=a_2;
  kappa[lapin,3]:=a_3;
  kappa[lapin,4]:=a_4;
end;

procedure init_statistique(var stat_etre:echantillon);
var i:les_etres;
  j:estimateur;
  k:integer;
begin
  for k:=1 to nb_essai do
  for i:=loup to lapin do
    for j:=min to max do stat_etre[k,i,j]:=0;
  end;

```

```

{*****
 * Cette procedure prepare les donnees de depart du programme. *
 * En particulier le fichier de sortie est assigne. De plus   *
 * l'increment de la proportion est calcule.      *
*****}

procedure init_fichier(var fichier,fichier_stat:text; var debut:boolean);
begin
if debut then
begin
  assign(fichier,'c:\andre\sys_dyn.dat');
  assign(fichier_stat,'c:\andre\out_excel.dat');
  rewrite(fichier);
  rewrite(fichier_stat);
  debut:=false;
end
else
begin
  rewrite(fichier);
end;
end;

procedure init(var foret:vecteur1; var bords:contour; var etiquette:vecteur2;
              var transition:automate; var stat_etre:echantillon;
              var kappa:prob; var nb_etre_max:longint);
begin
  init_foret(foret,0);
  init_bords(bords);
  init_etiquette(etiquette,0);
  init_transition(transition);
  init_statistique(stat_etre);
  init_kappa(kappa);
  nb_etre_max:=0;
end;

{*****
 * Ces procedures dessinent un rectangle pour   *
 * indiquer que l'ordinateur est en execution. *
*****}

procedure peri_rect(droite,bas:word);
var l,h:word;
begin
l:=getmaxx;
h:=getmaxy;
setcolor(blue);
rectangle(l-(droite+1),101,l-(bas),9);
end;

```

```

procedure rempli_rect(couleur,indice,sup_indice,droite,bas:word);
  var l,h:word;
begin
  l:=getmaxx;
  h:=getmaxy;
  setcolor(couleur);
  moveto(l-droite,100-round(89*indice/sup_indice));
  lineto(l-bas,100-round(89*indice/sup_indice));
end;

{*****
 * Cette fonction transforme un nombre en texte *
*****}

function nentext(n:longint):string;
var s:string;
begin
  str(n:7,s); nentext:=s;
end;

function reentext(n:real):string;
var s:string;
begin
  str(n:10:3,s); reentext:=s;
end;

{*****
 * Cette fonction retourne l'exposant en base m d'un nombre entier *
*****}

function exposant(m,n:integer):integer;
var i,e :integer;
begin
  e:=1;
  if n=0 then e:=1 else for i:=1 to n do e:=m*e;
  exposant:=e;
end;

{*****
 * Cette fonction retourne l'exposant en base m d'un nombre reel *
*****}

function expose_reel(m:extended;n:integer):extended;
var
  i:integer;
  e:extended;
begin
  e:=1;
  if n=0 then e:=1 else for i:=1 to n do e:=m*e;
  expose_reel:=e;
end;

```

```

{*****
 * Cette fonction retourne le signe d'un entier *
*****}

function signe(n:integer):integer;
var s:integer;
begin
if n=0 then s:=0 else s:=round(n/abs(n));
signe:=s;
end;

{*****
 * Cette fonction retourne le signe d'un reel *
*****}

function signe_reel(n:extended):integer;
var s:integer;
begin
if n = 0 then
begin
s:=0;
end
else
begin
if n < 0 then s:=-1 else s:= 1
end;
signe_reel:=s;
end;

{*****
 * Cette fonction retourne le coefficient du binome *
*****}

function comb(l,n:integer):longint;
var i,fact_n_l,fact_l:longint;
begin
fact_l:=1;
fact_n_l:=1;
if l > 0 then for i:=1 to l do fact_l:=fact_l * i;
if n >= l then for i:=n-l+1 to n do fact_n_l:=fact_n_l * i;
comb:=round(fact_n_l/fact_l);
end;

{*****
 * Cette fonction trouve le minimum entre deux nombres. *
*****}

function minimum(nombre1,nombre2:longint):longint;
begin
If nombre1 < nombre2 then minimum:=nombre1
else minimum:=nombre2
end;

```

```

procedure ecrit_param_simul(var fichier:text);
begin
  writeln(fichier,' ****');
  writeln(fichier,' * Les donnees ci-dessous indiquent dans quelles *');
  writeln(fichier,' *   conditions la simulation a ete effectue.    *');
  writeln(fichier,' ****');
  writeln(fichier);
  writeln(fichier,' kappa_0 = ',a_0:6:4);
  writeln(fichier,' kappa_1 = ',a_1:6:4);
  writeln(fichier,' kappa_2 = ',a_2:6:4);
  writeln(fichier,' kappa_3 = ',a_3:6:4);
  writeln(fichier,' kappa_4 = ',a_4:6:4);
  writeln(fichier,' temps_ecelle = ',temps_ecelle:6);
  writeln(fichier,' temps_relaxation = ',temps_relaxation:6);
  writeln(fichier,' densite_lapin_genere = ',densite_lapin_genere:5:4);
  writeln(fichier,' densite_loup_genere = ',densite_loup_genere:5:4);
  writeln(fichier,' temps_total = ',temps_total:6);
  writeln(fichier,' dimension = ',dimension:3);
  writeln(fichier,' nb_essai = ',nb_essai:3);
  writeln(fichier,' taille_echantillon = ',taille_echantillon:3);
  writeln(fichier,' nb_individu = ',nb_individu:2);
  writeln(fichier,' nb_direction = ',nb_direction:2);
  writeln(fichier);
end;

{ ****
  * Cette procedure ecrit les resultats portant sur le systeme proie et *
  * predateur.Il y a " 2 x temps_final " donnees d'ecrite dans ce fichier. *
  * Les resultats sont ecrits par ligne. La premiere valeur est le nombre *
  * de loups restant sur le reseau de taille " dimension x dimension " *
  * La deuxieme valeur est le nombre de lapins restant sur le reseau de *
  * taille " dimension x dimension ". Le t-ieme coupe de donnees *
  * correspond respectivement au temps t. *
}

```

```

procedure ecrit_fichier(var nb_etre:vecteur; i:longint; var fichier:text);
begin
  writeln(fichier,nb_etre[loup]:20,nb_etre[lapin]:20);
end;

```

```

{ ****
  * Cette fonction trouve le maximum entre deux nombres. *
}

```

```

function maximum(nombre1,nombre2:longint):longint;
begin
  If nombre1 < nombre2 then maximum:=nombre2
  else maximum:=nombre1
end;

```

```

function maximum_reel(nombre1,nombre2:extended):real;
begin
  If nombre1 < nombre2 then maximum_reel:=nombre2
  else maximum_reel:=nombre1
end;

```

```

procedure commentaire(var fichier:text; var remarque:phrase; var kappa:prob);
begin
  writeln(fichier,'*****');
  writeln(fichier,' * Toutes les donnees qui suivent doivent etre lu en ligne      *');
  writeln(fichier,' * et par groupe de ',nb_individu,'. Elles indiquent respectivement, par      *');
  writeln(fichier,' * hierarchie decroissante de predation, le nombre d"etres      *');
  writeln(fichier,' * present a un temps t. Ces donnees, relatives a kappa_0 =      *');
  writeln(fichier,' ',kappa[lapin,0]:10:6,'      *');
  writeln(fichier,' * sont le resultat d'un echantillonnage uniforme portant      *');
  writeln(fichier,' * sur ',temps_total,' donnees.      *');
  writeln(fichier,'*****');
  writeln(fichier);
  remarque[min]:= 'nombre minimal d"etres = ';
  remarque[moy]:= ' nombre moyen d"etres = ';
  remarque[s_xbar]:= ' ecart s_xbas d"etres = ';
  remarque[max]:= 'nombre maximal d"etres = ';
  { remarque[stable]:= 'densite d"equilibre possible = '}
end;

procedure ecrit_stat(var fichier:text; var courbe:graphe; no_essai:longint;
                     var stat_etre:echantillon; var kappa:prob);
var
  i,t:longint;
  estime:estimateur;
  etre:les_etres;
  rem:phrase;

begin
  commentaire(fichier,rem,kappa);
  { for i:=1 to taille_echantillon do
  begin
    for etre:=loup to lapin do
    begin
      write(fichier,courbe[no_essai,etre,i],chr(9));
    end;
    if (i mod 10 = 0) then writeln(fichier);
  end;}
  writeln(fichier);
  for estime:=min to max do
  begin
    { write(fichier,rem(estime));}
    for etre:=loup to lapin do
    begin
      write(fichier,stat_etre[no_essai,etre,estime]:10:3,chr(9));
    end;
    writeln(fichier);
  end;
  writeln(fichier_stat);
  writeln(fichier_stat);
end;

```

```

procedure ecrit_loi_action(var fichier:text; var kappa:prob);
var
  i:integer;
  coeff:extended;
  etre:les_etres;
begin
  writeln(fichier,'*****');
  writeln(fichier,' * La loi d"action de masse est la suivante : *');
  writeln(fichier,'*****');
  writeln(fichier);
  writeln(fichier,'y=');
  for i:=nb_direction downto 0 do
    begin
      coeff:=kappa[lapin,i];
      if i > 0 then write(fichier,coeff:8:7,'x_',i,'+')
      else write(fichier,coeff:8:7,'x_',i);
    end;
  writeln(fichier);
  writeln(fichier);
end;

procedure ecrit_stable(var fichier:text; var stabilite:racine);
var
  i:integer;
  etre:les_etres;
begin
  writeln(fichier,'*****');
  writeln(fichier,' * Les densites d"equilibre possible sont les suivantes : *');
  writeln(fichier,'*****');
  writeln(fichier);
  for etre:=loup to lapin do
    begin
      for i:=1 to nb_direction do
        begin
          write(fichier,stabilite[etre,i]:10:7,chr(9));
        end;
      writeln(fichier);
    end;
  writeln(fichier);
  writeln(fichier);
end;

procedure resultat_actif(var fichier:text;nb_etre:vecteur;stabilite:racine);
begin
  cleardevice;
  peri_rect(51,41);
  impri_etre(nb_etre,340,temp,stabilit);
  ecrit_fichier(nb_etre,temp,fichier);
end;

```

```

procedure choisit_point(var fichier:text; var courbe:graphique;no_essai,temp:longint);

var
  t,l,h,compte,posi,nb_loup,nb_lapin:longint;
  proportion_trop,alea:extended;
  supprime:boolean;

begin
  l:=getmaxx;
  h:=getmaxy;
  proportion_trop:=l/temp;
  compte:=1;
  posi:=10;
  supprime:=false;
  reset(fichier);
  if proportion_trop < 1 then supprime:=true;
  for t:=1 to temp do
  begin
    if supprime then
    begin
      alea:=random;
      if alea > proportion_trop then
      begin
        read(fichier,nb_loup,nb_lapin);
      end
      else
      begin
        if compte <= taille_echantillon then
        begin
          read(fichier,nb_loup,nb_lapin);
          courbe[no_essai,lapin,compte]:=nb_lapin;
          courbe[no_essai,loup,compte]:=nb_loup;
          compte:=compte+1;
        end;
      end;
    end;
  end
  else
  begin
    read(fichier,nb_loup,nb_lapin);
    courbe[no_essai,lapin,t]:=nb_lapin;
    courbe[no_essai,loup,t]:=nb_loup;
  end;
end;
end;

```

```

procedure point_ecran(var fichier:text;var nb_loup,nb_lapin,i,passe:longint;
                      proportion_trop,dilate_y:extended;supprime:boolean);

var l,h:longint;
    alea:extended;
begin
l:=getmaxx;
h:=getmaxy;
if supprime then
begin
alea:=random;
if alea > proportion_trop then
begin
read(fichier,nb_etre[loup],nb_etre[lapin]);
passe:=passe+1;
end
else
begin
read(fichier,nb_etre[loup],nb_etre[lapin]);
moveto(i-1-passe,h-round(nb_loup*dilate_y));
setcolor(blue);
lineto(i-passe,h-round(nb_etre[loup]*dilate_y));
moveto(i-1-passe,h-round(nb_lapin*dilate_y));
setcolor(white);
lineto(i-passe,h-round(nb_etre[lapin]*dilate_y));
nb_loup:=nb_etre[loup];
nb_lapin:=nb_etre[lapin];
end;
end
else
begin
read(fichier,nb_etre[loup],nb_etre[lapin]);
moveto(i-1,h-round(nb_loup*dilate_y));
setcolor(blue);
lineto(i,h-round(nb_etre[loup]*dilate_y));
moveto(i-1,h-round(nb_lapin*dilate_y));
setcolor(white);
lineto(i,h-round(nb_etre[lapin]*dilate_y));
nb_loup:=nb_etre[loup];
nb_lapin:=nb_etre[lapin];
end;
end;

```

```

procedure trace_courbe(var fichier:text; nb_etre:vecteur;var stabilite:racine;
                      nb_etre_max,temp:longint);

var
  x,y,i,l,h,nb_loup,nb_lapin,passe,position_relax,equilibre:longint;
  proportion_trop,alea,dilate_y,dilate_x:extended;
  supprime:boolean;

begin
  l:=getmaxx; h:=getmaxy;
  dilate_x:=1; dilate_y:=(h/nb_etre_max);
  proportion_trop:=l/temp;passe:=0;
  for i:=1 to nb_direction do
    begin
      equilibre:=round(dimension * dimension * stabilite[lapin,i]);
      nb_etre_max:=maximum(nb_etre_max,equilibre);
    end;
  supprime:=false;
  if proportion_trop < 1 then
    begin
      supprime:=true;
      position_relax:=round(temp_relaxation * proportion_trop);
    end
  else
    begin
      position_relax:= temp_relaxation;
    end;
  reset(fichier);
  read(fichier,nb_etre[loup],nb_etre[lapin]);
  nb_loup:=nb_etre[loup];
  nb_lapin:=nb_etre[lapin];
  moveto(round(1),h-round(nb_etre[loup]*dilate_y));
  cleardevice;
  for i:=2 to temp do
    begin
      point_ecran(fichier,nb_loup,nb_lapin,i,passe,proportion_trop,dilate_y,supprime);
      setcolor(lightgray);
      for j:=1 to nb_direction do
        begin
          equilibre:=round(dimension * dimension * stabilite[lapin,j]);
          y:= h-round(equilibre * dilate_y);
          moveto(0,y);
          lineto(639,y);
        end;
      setcolor(yellow);
      x:= position_relax; moveto(x,y);
      setcolor(yellow); lineto(x,100);
      outtextxy(x,y+105,'temp de relaxation');
      moveto(x,y); lineto(x-10,y+10);
      moveto(x,y);
      lineto(x+10,y+10);
    end;
  end;
end;

```

```

*****
* Cette procedure code, pour l'ecran, l'état de la foret. *
* Chaque couleur représente un état particulier de   *
* l'automate cellulaire à la case (i,j) du réseau.   *
* Ce réseau de taille dimension x dimension est ici   *
* associé à l'écran. Les codes de couleurs utilisés,   *
* pour reconnaître l'état de l'automate cellulaire,   *
* sont les suivant :          *
*                         *
* bleu : nb de loups (seul) > 1.           *
* blanc : nb de lapins (seul) > 1.           *
* rouge : il y a : lapins et loups.          *
* brun : il n'y a pas de vie à ce noeud.      *
*                         *
*****
}

```

```

procedure code_couleur(i,j:integer;var couleur:integer;
                      nb_loup,nb_lapin:integer);
begin
  case nb_loup of
    0: begin
      case nb_lapin of
        0: couleur:=red;      {brun sombre}
        1: couleur:=white;    {gris pale}
        2: couleur:=green;    {vert}
        3: couleur:=green;    {bleu}
        4: couleur:=green;    {blanc}
      end;
    end;
    1..4: begin
      case nb_lapin of
        0: begin
          couleur:=blue; {bleu}
        end;
        1..4: begin
          couleur:=lightgray; {gris pale}
        end;
      end;
    end;
  end;
end;

```

```

{*****
 * Les 5 procedures qui suivent servent uniquement a des *
 * fin d'impression a l'ecran pour facilite l'analyse *
 * de l'exactitude du fonctionnement du programme. *
*****}

procedure impri_foret(var foret:vecteur1; y:integer);
var i,j:longint;
begin
for i:=0 to dimension+1 do
begin
  moveto(120,y+10*i);
  for j:=0 to dimension+1 do
  begin
    outtext(ngettext(foret[i]^ [j]));outtext(' ');
  end;
end;
end;

procedure impri_etiq(var etiq:vecteur2; x,y:integer);
var i,j:longint;
begin
  outtextxy(x,y-10,'etiquette =');
  for i:=0 to dimension+1 do
  begin
    for j:=0 to dimension+1 do
    begin
      outtextxy(30*j+x,y+10*i,ngettext(etiq[i]^ [j]));
    end;
  end;
end;
end;

procedure impri_etre(var nb_etre:vecteur; y:integer;temps:longint;
                     stabilite:racine);
var
i,equilibre:longint;
begin
  setcolor(white);
  outtextxy(485,y,'nb_loups =' + ngettext(nb_etre[loup]));
  outtextxy(485,y+10,'nb_lapins =' + ngettext(nb_etre[lapin]));
  outtextxy(485,y+20,'temps =' + ngettext(temps));
  for i:=1 to nb_direction do
  begin
    begin
      equilibre:=round(dimension*dimension*stabilite[lapin,i]);
      outtextxy(485,y+20+10*i,'equilibre =' + ngettext((equilibre)));
    end;
  end;
end;

```

```

procedure impri_bord(var bords:contour);
  var i:integer;
    cote:lesbords;
begin
  for cote:=haut to droite do
  begin
    for i:=0 to dimension+1 do
    begin
      writeln(bords[cote,i]);
    end;
    writeln;
  end;
end;

{*****
 * Cette fonction ecrit le nombre binaire *
 * correspondant a un nombre en base 10. *
*****}

procedure nb_binaire(foret:vecteur1; y:integer);
  var a,i,j,k,l,m:integer;
begin
  k:=0;m:=0;
  for i:=1 to dimension do
  begin
    for j:=1 to dimension do
    begin
      a:=foret[i]^ [j];
      for l:=1 to 8 do
      begin
        inc(k, a and 1);
        outtextxy(80-10*l,y+10*m,nentext(a and 1));
        a:=a shr 1;
      end;
      m:=m+1;
    end;
  end;
end;

{*****
 * Cette fonction evalue le polynome representant *
 * la loi d'action de masse *
*****}

function polynome(x:extended; var kappa:prob):extended;
var
  poly:extended;
  i:integer;
begin
  poly:=0;
  for i:=0 to nb_direction do poly:=poly + kappa[lapin,i] * expose_reel(x,i);
  polynome:=poly;
end;

```

```

{*****
 * Cette fonction retourne le nombre d'etres vivants *
 * d'une espece present a un noeud du reseau      *
*****}

function digit(entier:integer;espece:les_etres):integer;
var i,k,a:integer;
  no_individu:integer;
begin
  case espece of
    loup: no_individu:=1 ;
    lapin: no_individu:=2 ;
  end;
  a:= entier shr (4*(nb_individu-no_individu));
  k:=0;
  for i:=1 to 4 do
  begin
    inc(k,a and 1);
    a:=a shr 1;
  end;
  digit:=k;
end;

{*****
 * Cette procedure calcule l'estimation dela moyenne, la variance ainsi que l'etendue.   *
 * *
*****}

procedure statistique_etre(k,temp:longint; var nb_etre:vecteur; var stat_etre:echantillon);
var
  n:longint;    i:les_etres;    moy_carre:extended;
begin
  if temp=temp_relaxation+1 then
  begin
    stat_etre[k,loup,min]:=nb_etre[loup];      stat_etre[k,lapin,min]:=nb_etre[lapin];
  end;
  for i:=loup to lapin do
  begin
    stat_etre[k,i,min]:=minimum(round(stat_etre[k,i,min]),nb_etre[i]);
    stat_etre[k,i,moy]:=stat_etre[k,i,moy] + nb_etre[i];
    stat_etre[k,i,max]:=maximum(round(stat_etre[k,i,max]),nb_etre[i]);
    stat_etre[k,i,s_xbar]:=stat_etre[k,i,s_xbar]+nb_etre[i]*nb_etre[i];
  end;
  if temp=temp_total then
  begin
    n:=temp_total-temp_relaxation;
    for i:=loup to lapin do
    begin
      stat_etre[k,i,moy]:=stat_etre[k,i,moy]/n;
      moy_carre:=stat_etre[k,i,moy] * stat_etre[k,i,moy];
      stat_etre[k,i,s_xbar]:=stat_etre[k,i,s_xbar]/n - moy_carre;
      stat_etre[k,i,s_xbar]:=sqrt(stat_etre[k,i,s_xbar]);
    end;
  end;
end;

```

```

*****
* Cette procedure calcul les points de stabilites. *
*****}

procedure pt_stable(var stabilite:racine; var kappa:prob);
  var
    indice,signe_gauche,signe_droit,test_signe,signe_milieu:integer;
    nb_stable:longint;
    densite_initial,densite_final,epsilon,poly:extended;
    densite_milieu,delta_densite,densite_sup:extended;
begin
  init_stable(stabilite);
  indice:=1;  densite_initial:=0;
  delta_densite:=0.01;  epsilon:=0.0000000000000001;
  while densite_initial <= 4 do
  begin
    densite_final:=densite_initial + delta_densite;
    densite_sup:=densite_final;
    signe_gauche:=signe_reel(polynome(densite_initial,kappa));
    signe_droit:=signe_reel(polynome(densite_final,kappa));
    test_signe:=signe_gauche * signe_droit;
    if test_signe <= 0 then
    begin
      densite_milieu:=(densite_initial + densite_final)/2;
      poly:=polynome(densite_milieu,kappa);
      signe_milieu:=signe_reel(polynome(densite_milieu,kappa));
      while ( signe_milieu * poly > epsilon ) do
      begin
        densite_milieu:=(densite_initial + densite_final)/2;
        poly:=polynome(densite_milieu,kappa);
        signe_milieu:=signe_reel(poly);
        case signe_gauche of
          -1 : begin
            case signe_milieu of
              -1 : densite_initial:=densite_milieu;
              1 : densite_final:=densite_milieu;
            end;
            end;
          0 : densite_final:=densite_initial;
        1 : begin
          case signe_milieu of
            -1 : densite_final:=densite_milieu;
            1 : densite_initial:=densite_milieu;
          end;
          end;
        end;
        densite_milieu:=(densite_initial + densite_final)/2;
        poly:=polynome(densite_milieu,kappa);
        signe_milieu:=signe_reel(polynome(densite_milieu,kappa));
      end;
      stabilite[lapin,indice]:=densite_milieu;  indice:=indice+1;
    end;
    densite_initial:=densite_sup;
  end;
end;

```

```

procedure creer_loup(var foret:vecteur1);
var
  i,j,nb_loup_genere,compte_loup:longint;
  intersection,nb_bit_loup,etat_loup:integer;
begin
  nb_loup_genere:=round(dimension * dimension * densite_loup_genere);
{ writeln('nb_loup_genere = ',nb_loup_genere);
readln; }
  if densite_loup_genere <= 2.5 then
  begin
    nb_bit_loup:=1;
    compte_loup:=0;
    while compte_loup < nb_loup_genere do
    begin
      i:= 1 + round( int( dimension * random ) );
      j:= 1 + round( int( dimension * random ) );
      etat_loup:=exposant(2,round(4+int(4*random)));
      intersection:=digit( etat_loup and foret[i]^j ,lapin);
      compte_loup:=compte_loup + nb_bit_loup - intersection;
      foret[i]^j:= foret[i]^j or etat_loup ;
    end;
  end
  else
  begin
    for i:=1 to dimension do
      for j:=1 to dimension do foret[i]^j:=foret[i]^j or 228;
    nb_bit_loup:=1;
    compte_loup:= 4 * dimension * dimension;
    while compte_loup > nb_loup_genere do
    begin
      i:= 1 + round( int( dimension * random ) );
      j:= 1 + round( int( dimension * random ) );
      etat_loup:=exposant(2,round(4+int(4*random)));
      intersection:=digit( etat_loup and foret[i]^j ,lapin);
      compte_loup:=compte_loup - intersection;
      foret[i]^j:=foret[i]^j and not etat_loup;
    end;
  end;
end;

```

```

procedure creer_lapin(var foret:vecteur1);
var
  i,j,nb_lapin_genere,compte_lapin:longint;
  intersection,nb_bit_lapin,etat_lapin:integer;
begin
  nb_lapin_genere:=round(dimension * dimension * densite_lapin_genere);
{ writeln('nb_lapin_genere = ',nb_lapin_genere);
readln;}
  if densite_lapin_genere <= 2.5 then
  begin
    nb_bit_lapin:=1;
    compte_lapin:=0;
    while compte_lapin < nb_lapin_genere do
    begin
      i:= 1 + round( int( dimension * random ) );
      j:= 1 + round( int( dimension * random ) );
      etat_lapin:=exposant(2,round(int(4*random)));
      intersection:=digit( etat_lapin and foret[i]^ [j] ,lapin);
      compte_lapin:=compte_lapin + nb_bit_lapin - intersection;
      foret[i]^ [j]:=foret[i]^ [j] or etat_lapin;
    end;
  end
  else
  begin
    for i:=1 to dimension do
      for j:=1 to dimension do foret[i]^ [j]:=foret[i]^ [j] or 15;
    compte_lapin:= 4 * dimension * dimension;
    nb_bit_lapin:=1;
    while compte_lapin > nb_lapin_genere do
    begin
      i:= 1 + round( int( dimension * random ) );
      j:= 1 + round( int( dimension * random ) );
      etat_lapin:=exposant(2,round(int(4*random)));
      intersection:=digit( etat_lapin and foret[i]^ [j] ,lapin);
      compte_lapin:=compte_lapin - intersection;
      foret[i]^ [j]:=foret[i]^ [j] and not etat_lapin;
    end;
  end;
end;

{*****
 * Cette procedure genere deux especes vivantes. *
 * Pour l'instant aucune distribution n'est privilegier *
*****}

```

```

procedure genere_vie(var foret:vecteur1);
var
  genere_lapin,genere_loup,premier:boolean;
begin
  if densite_lapin_genere >= 0 then genere_lapin:=true;
  if densite_loup_genere >= 0 then genere_loup:=true;
  if genere_lapin then creer_lapin(foret);
  if genere_loup then creer_loup(foret);
end;

```

```

{*****
 * Cette fonction retourne le nombre entre 0 et 15 *
 * qui correspond a la disposition des etres a un *
 * noeud quelconque du reseau de la foret. *
*****}

function etat_espece(espece:les_etres;etat:integer):integer;
var no_individu,nombre,i,p:integer;
begin
case espece of
loup: no_individu:=1 ;
lapin: no_individu:=2 ;
end;
nombre:=0;
etat:=etat shr (4*(nb_individu-no_individu));
for i:=1 to 4 do
begin
p:=exposant(2,i-1);
nombre:=(nombre) or (etat and p);
end;
etat_espece:=nombre;
end;

{*****
 * Cette procedure compte le nombre d'etre de chaque *
 * especes vivantes qu'il y a dans la foret initialement. *
*****}

procedure compte_etre(var foret:vecteur1; var nb_etre:vecteur);
var
i,j,no_individu,entier:integer;
k:les_etres;
begin
for k:=loup to lapin do nb_etre[k]:=0;
for i:=1 to dimension do
begin
for j:=1 to dimension do
begin
entier:=foret[i]^j;
for k:=lapin downto loup do
begin
nb_etre[k]:=nb_etre[k] + digit(entier,k);
end;
end;
end;
end;
{ writeln('nb_etre[lapin]= ',nb_etre[lapin]);
readln;}
end;

```

```

procedure calcul_transition(var transition:automate;var kappa:prob;temps_ecelle:real);
var a,b:extended;
  i,j:integer;
begin
  for i:=1 to nb_direction-1 do
  begin
    for j:=0 to i do
    begin
      case signe_reel(kappa[lapin,j]) of
        1:begin
          a:=kappa[lapin,j];
          a:=a*comb(j,i)*exposant(nb_direction,j)/comb(j,nb_direction);
          transition[lapin,i,i+1]:=transition[lapin,i,i+1] + a;
        end;
        -1:begin
          a:=-kappa[lapin,j];
          a:=a*comb(j,i)*exposant(nb_direction,j)/comb(j,nb_direction);
          transition[lapin,i,i-1]:=transition[lapin,i,i-1] + a;
        end;
      end;
    end;
  end;
  transition[lapin,0,1]:=kappa[lapin,0];
  a:=0;
  for j:=0 to nb_direction do
  begin
    a:= a + exposant(nb_direction,j) * kappa[lapin,j];
  end;
  transition[lapin,nb_direction,nb_direction-1]:= -a;
  if trouve_temps_ecelle then
  begin
    a:=0;
    for i:=0 to nb_direction do
    begin
      b:=0;
      for j:=0 to nb_direction do
      begin
        b:=b+transition[lapin,i,j];
      end;
      a:=maximum_reel(a,b);
    end;
  end;
  restorecrtmode;
  writeln('Le temps d"echelle h doit etre plus petit que : 1/', a );
  writeln;
  writeln('Arreter l"execution et affecter le temps d"echelle approprié');
  readkey;
end
else
begin
  for i:=0 to nb_direction do
  begin
    a:=0;
    for j:=0 to nb_direction do
    begin

```

```

transition[lapin,i,j]:=transition[lapin,i,j]/temps_echelle;
a:=a + transition[lapin,i,j];
end;
transition[lapin,i,i]:=1-a;
end;
end;
for i:=0 to nb_direction do
begin
for j:=0 to nb_direction do
begin
if transition[lapin,i,j] < 0 then
begin
restorecrtmode;
writeln('transition[lapin,',i,',',j,']= ',transition[lapin,i,j]);
writeln('Erreur sur les hypotheses des coefficients a_i');
readkey;
end;
end;
end;
end;
end;

{*****
 * Cette procedure verifie si les probabilites *
 * de transition sont positives. *
*****}

procedure test_transition(var transition:automate; var transi_ok:boolean);
var j,k:integer;
i:les_etres;
begin
transi_ok:=true;
for i:=loup to lapin do
for j:=0 to nb_direction do
for k:=0 to nb_direction do
begin
if transition[i,j,k] < 0 then
begin
transi_ok:=false;
exit;
end;
end;
end;
end;

procedure message_erreur(var fichier:text;var kappa:prob;kappa_sup:extended);
begin
writeln(fichier,'Erreur; probabilites de transitions negatives.');
writeln(fichier,'La valeur de kappa_0 = ',kappa[lapin,0]:10:5,' est donc inadmissible *');
writeln(fichier);
end;

```

```

*****
* Cette procedure ecrit les probabilites de *
* transitions dans le fichier de sortie " stat " *
*****
}

procedure cumule_transition(var transition:automate);
var
somme_prob:extended;
i,j:integer;
begin
somme_prob:=0;
for i:=0 to nb_direction do
begin
for j:=0 to nb_direction do
begin
somme_prob:=somme_prob + transition[lapin,i,j];
transition[loup,i,j]:=somme_prob;
end;
somme_prob:=0;
end;
end;

procedure ecrit_transition(var fichier:text; var transition:automate);
var j,k:integer;
i:les_etres;
begin
for i:=loup to lapin do
begin
writeln(fichier,'*****');
writeln(fichier,' * Voici les matrices de transitions pour les etres *');
writeln(fichier,'*****');
writeln(fichier);
for j:=0 to nb_direction do
begin
for k:=0 to nb_direction do
begin
write(fichier,transition[i,j,k]:10:5,' ');
end;
writeln(fichier);
end;
writeln(fichier);
end;
end;
end;

*****
* Cette procedure definie comment les interactions, a un noeud fixe, *
* se produisent. A ce titre cette procedure est le noyau essentiel *
* du programme. Du moins c'est elle qui fera en sorte que le modele *
* utilise sera plausible ou non *
*****
}

```

```

procedure interaction(var manger_par_le:vectboolean; var transition:automate;
                     var nb_loup,nb_lapin:longint);
var
  alea,b:extended;
  etat_final_lapin,etat_initial_lapin,delta_lapin:integer;
begin
  manger_par_le[lapin]:=false;
  manger_par_le[loup]:=false;
  alea:=random;
  etat_initial_lapin:=nb_lapin;
  etat_final_lapin:=0;
  b:=transition[loup,etat_initial_lapin,etat_final_lapin];
  while b < alea do
  begin
    etat_final_lapin:=etat_final_lapin+1;
    if etat_final_lapin > nb_direction then
    begin
      restorecrtmode;
      Writeln('probleme avec la matrice de transition cumulative');
      readln;
    end;
    b:=transition[loup,etat_initial_lapin,etat_final_lapin];
  end;
  delta_lapin:= etat_final_lapin - etat_initial_lapin ;
  nb_lapin := nb_lapin + delta_lapin;
end;

{*****
 * Cette fonction effectue une rotation de pi/2 ou de 3pi/2 *
 * de la configuration des etres a un noeud donne. *
*****}

function rotation(etat_etre:integer):integer;
  var alea,bit_de_gauche,bit_de_droite:integer;
begin
  alea:=round(int(2*random));
  case alea of
    { rotation de pi/2 }
  0:begin
    bit_de_gauche:=etat_etre shr 3;
    etat_etre:=etat_etre - bit_de_gauche * exposant(2,3);
    etat_etre:=etat_etre shl 1;
    etat_etre:=etat_etre or bit_de_gauche;
    rotation:=etat_etre;
  end;
    { rotation de 3pi/2 }
  1:begin
    bit_de_droite:=etat_etre and 1;
    etat_etre:=etat_etre shr 1;
    etat_etre:=etat_etre + bit_de_droite * exposant(2,3);
    rotation:=etat_etre;
  end;
  end;
end;

```

```

*****+
* Cette fonction ajoute ou enleve le nombre *
* d'etres approprie au noeud (i,j) *
*****+}

function ajuste_noeud(etat_etre,delta:integer):integer;

var alea,bit_1,copie_etat:integer;
begin
  copie_etat:=etat_etre;
  case signe(delta) of
    { au moins un etre sera supprimer }
    -1:begin
      if abs(delta)=digit(etat_etre,lapin) then
        begin
          ajuste_noeud:=0;
        end
      else
        begin
          while digit(etat_etre,lapin) > (digit(copie_etat,lapin) + delta) do
            begin
              alea:=round(int(4*random));
              bit_1:=etat_etre and exposant(2,alea);
              etat_entre:=etat_entre - bit_1;
            end;
          ajuste_noeud:=etat_entre;
        end;
    end;
    { aucun etre sera supprimer ou ajouter }
    0:begin
      ajuste_noeud:=etat_entre;
    end;
    { au moins un etre sera ajouter }
    1:begin
      if abs(delta)=4-digit(etat_entre,lapin) then
        begin
          ajuste_noeud:=15;
        end
      else
        begin
          while digit(etat_entre,lapin) < (digit(copie_etat,lapin) + delta) do
            begin
              alea:=round(int(4*random));
              bit_1:=(etat_entre xor 15) and exposant(2,alea);
              etat_entre:=etat_entre + bit_1;
            end;
          ajuste_noeud:=etat_entre;
        end;
    end;
  end;
end;

```

```

procedure rempli_bord(i:integer;var foret:vecteur1;var etiquette:vecteur2);
var
m:integer;
begin
for m:=0 to dimension+1 do
begin
foret[0]^*[m]:=etiquette[0]^*[m];
foret[i-1]^*[m]:=etiquette[1]^*[m];
foret[i]^*[m]:=etiquette[2]^*[m];
foret[dimension+1]^*[m]:=etiquette[3]^*[m];
end;
{ frontiere(foret,bords);}
for m:=0 to dimension+1 do
begin
foret[1]^*[m]:=foret[1]^*[m] or foret[dimension+1]^*[m];
foret[dimension+1]^*[m]:=0;
foret[dimension]^*[m]:=foret[dimension]^*[m] or foret[0]^*[m];
foret[0]^*[m]:=0;
foret[m]^*[1]:=foret[m]^*[1] or foret[m]^*[dimension+1];
foret[m]^*[dimension+1]:=0;
foret[m]^*[dimension]:=foret[m]^*[dimension] or foret[m]^*[0];
foret[m]^*[0]:=0;
end;
end;

procedure rempli_interieur(i:integer;var foret:vecteur1;var etiquette:vecteur2);
var
m:integer;
begin
for m:=0 to dimension+1 do
begin
foret[i-1]^*[m]:=etiquette[1]^*[m];
etiquette[1]^*[m]:=etiquette[2]^*[m];
etiquette[2]^*[m]:=etiquette[3]^*[m];
etiquette[3]^*[m]:=0;
end;
end;

procedure calcul_et_ecrit(var fichier_stat:text;kappa:prob;
var transi_ok:boolean;var stabilite:racine;
var transition:automate);
begin
ecrit_loi_action(fichier_stat,kappa);
pt_stable(stabilite,kappa);
ecrit_stable(fichier_stat,stabilite);
calcul_transition(transition,kappa,temp_echelle);
cumule_transition(transition);
test_transition(transition,transi_ok);
ecrit_transition(fichier_stat,transition);
end;

```

```

{*****
 * Cette procedure gere le deplacement des etres dans la foret *
 * pendant que les collisions s'effectuent se soit fait. *
*****}

procedure distribue(i,j:integer; var foret:vecteur1; var bords:contour;
                    var tampon:vecteur;var etiquette:vecteur2);
var
  no_direction,no_individu,l,m,p:integer;
  h:les_etres;
begin
  if (i > 1) then l:= 2 else l:=1;
  for h:=loup to lapin do
  begin
    case h of
      loup: no_individu:=1 ;
      lapin: no_individu:=2 ;
    end;
    for no_direction:=0 to nb_direction-1 do
    begin
      p:=exposant(2,4-4*(no_individu-1) + no_direction );
      if ((tampon[h] and p) <> 0 ) then
      begin
        case no_direction of
          0:begin
            etiquette[l]^ [j-1]:=((etiquette[l]^ [j-1]) or (tampon[h] and p));
          end;
          1:begin
            etiquette[l+1]^ [j]:=((etiquette[l+1]^ [j]) or (tampon[h] and p));
          end;
          2:begin
            etiquette[l]^ [j+1]:=((etiquette[l]^ [j+1]) or (tampon[h] and p));
          end;
          3:begin
            etiquette[l-1]^ [j]:=((etiquette[l-1]^ [j]) or (tampon[h] and p));
          end;
        end;
      end;
      if ((i=dimension) and (j=dimension)) then
      begin
        rempli_bord(i,foret,etiquette);
      end;
      if ((i > 1) and (i <> dimension) and (j=dimension)) then
      begin
        rempli_interieur(i,foret,etiquette);
      end;
    end;
  end;
end;

```

```

*****
* Cette procedure s'occupe d'affecter correctement les noeuds *
* de la foret apres que les collision entre les etres se soit fait. *
*****
procedure collision(var foret:vecteur1; var bords:contour;
                    var nb_etre,tampon:vecteur;
                    var temps,nb_etre_max:longint);
var
  i,j,no_individu,etat,alea,couleur:integer;
  loup_avant,lapin_avant,loup_apres,lapin_apres:longint;
  delta_lapin,delta_loup:integer;
  max,stabilite_population:longint;

begin
  for i:=1 to dimension do
  begin
    for j:=1 to dimension do
    begin
      etat:=foret[i]^*[j];
      loup_avant:=digit(etat,loup);
      lapin_avant:=digit(etat,lapin);
      loup_apres:=loup_avant;
      lapin_apres:=lapin_avant;
      nb_etre_max:=maximum(maximum(nb_etre[loup],nb_etre_max),nb_etre[lapin]);
      code_couleur(i,j,couleur,loup_avant,lapin_avant);
      putpixel(i,j,couleur);
      interaction(manger_par_le,transition,loup_apres,lapin_apres);
      delta_lapin:=lapin_apres - lapin_avant;
      delta_loup:=loup_apres - loup_avant;
      nb_etre[loup]:=nb_etre[loup] + delta_loup;
      nb_etre[lapin]:=nb_etre[lapin] + delta_lapin;
      if loup_apres <> 0 then
      begin
        tampon[loup]:=rotation(ajuste_noeud(etat_espece(loup,etat),delta_loup));
        tampon[loup]:=tampon[loup] shl 4;
      end
      else tampon[loup]:=0;
      if lapin_apres <> 0 then
      begin
        tampon[lapin]:=rotation(ajuste_noeud(etat_espece(lapin,etat),delta_lapin));
        tampon[lapin]:=tampon[lapin] shl 0;
      end
      else tampon[lapin]:=0;
      delta_etre[loup]:=loup_avant-loup_apres;
      delta_etre[lapin]:=lapin_avant-lapin_apres;
      etat_etre[loup]:=etat_espece(loup,etat);
      etat_etre[lapin]:=etat_espece(lapin,etat);
      distribue(i,j,foret,bords,tampon,etiquette);
    end;
    rempli_rect((temps mod 14)+1,i,dimension,50,42);
  end;
end;

```

```

{*****
 * Debut du programme principal. *
*****}

begin
{randomize}
clrscr;
debut:=true;
init(foret,bords,etiquette,transition,stat_etre,kappa,nb_etre_max);
init_fichier(fichier,fichier_stat,debut);
ecrit_param_simul(fichier_stat);
verifigraphe(graphique_ok);
setgraphemode(grmode);
for no_essai:=1 to nb_essai do
begin
  init(foret,bords,etiquette,transition,stat_etre,kappa,nb_etre_max);
  genere_vie(foret);
  compte_etre(foret,nb_etre);
  temps_ecelle:=temps_ecelle + delta_temps_ecelle;
  calcul_et_ecrit(fichier_stat,kappa,transi_ok,stabilite,transition);
  if transi_ok then
begin
  for temps:=1 to temps_total do
  begin
    resultat_actif(fichier,nb_etre,stabilite);
    if temps > temps_relaxation then
begin
      statistique_etre(no_essai,temps,nb_etre,stat_etre);
    end;
    collision(foret,bords,nb_etre,tampon,temps,nb_etre_max);
    init_etiquette(etiquette,1);
  end;
  close(fichier);
  trace_courbe(fichier,nb_etre,stabilite,nb_etre_max,temps_total);
  readln;
  choisit_point(fichier,courbe,no_essai,temps_total);
  écrit_stat(fichier_stat,courbe,no_essai,stat_etre,kappa);
  écrit_stable(fichier_stat,stabilite);
  init_fichier(fichier,fichier_stat,debut);
end
else
begin
 { message_erreur(fichier_stat,kappa,kappa_sup);}
end;
end;
close(fichier);
close(fichier_stat);
closegraph;
end.

```

```

{*****
 * Fin du programme principal. *
*****}

```

```

*****
* Ce programme simule l'interaction entre une proie et *
* un predateur sur une grille carre de dimension n x n. *
*****
program ecosysteme;
uses graph,dos,crt;

const
  A00 = 0.0;  B00 = 0.0;
  A10 = -4.0; B01 = 4.0;
  A11 = 2.0;  B11 = -0.5;
  A20 = -1.0; B02 = -1.0;
  trouve_temps_echelle=false;
  temps_echelle:extended=200.0;
  delta_temps_echelle=0;
  temps_relaxation:longint=0;
  densite_lapin_genere=4.0;
  densite_loup_genere=4.0;
  temps_total=640;
  dimension=64;
  taille_echantillon=640;
  nb_essai=1;
  nb_individu=2;
  nb_direction=4;

type
  lesbords = (haut,bas,gauche,droite);
  les_etres = (total,loup,lapin);
  croissance = (moins,plus);
  estimateur = (min,moy,s_xbar,max);
  etat_noeud = ( naissance,meurt,extra_specifique,intra_specifique );
  contour = array[ lesbords , 0..dimension+1 ] of integer;
  vectboolean = array[ les_etres ] of boolean;
  vecteur = array[ les_etres ] of longint;
  racine = array[ les_etres , 1..nb_direction ] of extended;
  echantillon = array[ 1..nb_essai,les_etres,estimateur ] of extended;
  automate = array[ 0..4, 0..4, -1..5, -1..5 ] of extended;
{ prob = array[ les_etres , 0..25 ,croissance] of extended; }
  coefficient = array[les_etres,0..4,0..4] of extended;
  graphe = array[ 1..nb_essai ,les_etres, 1..taille_echantillon ] of longint;
  vecteur211=array[0..dimension+1] of integer;
  vecteur1=array[0..dimension+1] of ^vecteur211;
  vecteur21=array[0..3] of ^vecteur211;
  vecteur2=vecteur21;
  phrase=array[estimateur] of string;
  ensemble = set of 0..nb_direction;

```

```

var
bords:contour;
manger_par_le:vectboolean;
foret,ecran:vecteur1;
stat_etre:echantillon;
nb_etre,tampon,delta_etre,etat_etre:vecteur;
stabilite:racine;
kappa:coefficient;
transition:automate;
etiquette:vecteur2;
graphique_ok,debut,transi_ok:boolean;
courbe:graphe;
remarque:phrase;
resultat_simulation,mat_transition,stat_echantillon,equation:text;
gremode:integer;
no_essai,j,temp,nb_etre_max:longint;

```

```

{*****
 * Cette fonction retourne le coefficient du binome *
*****}

```

```

function comb(l,n:integer):longint;
var i,fact_n_l,fact_l:longint;
begin
fact_l:=1;
fact_n_l:=1;
if l > 0 then for i:=1 to l do fact_l:=fact_l * i;
if n >= l then for i:=n-l+1 to n do fact_n_l:=fact_n_l * i;
comb:=round(fact_n_l/fact_l);
end;

```

```

{*****
 * Cette fonction retourne l'exposant en base m d'un nombre entier *
*****}

```

```

function exposant(m,n:integer):longint;
var i,e :longint;
begin
e:=1;
if n=0 then e:=1 else for i:=1 to n do e:=m*e;
exposant:=e;
end;

```

```

{*****
 * Cette fonction retourne l'exposant en base m d'un nombre reel *
*****}

function exposé_reel(m:extended;n:integer):extended;
var
  i:integer;
  e:extended;
begin
  e:=1;
  if n=0 then e:=1 else for i:=1 to n do e:=m*e;
  exposé_reel:=e;
end;

{*****
 * Cette fonction trouve le maximum entre deux nombres. *
*****}

function maximum(nombre1,nombre2:longint):longint;
begin
  If nombre1 < nombre2 then maximum:=nombre2
  else maximum:=nombre1
end;

function maximum_reel(nombre1,nombre2:extended):real;
begin
  If nombre1 < nombre2 then maximum_reel:=nombre2
  else maximum_reel:=nombre1
end;
{*****
 * Cette fonction trouve le minimum entre deux nombres. *
*****}

function minimum(nombre1,nombre2:longint):longint;
begin
  If nombre1 < nombre2 then minimum:=nombre1
  else minimum:=nombre2
end;

{*****
 * Cette fonction retourne le signe d'un entier *
*****}

function signe(n:integer):integer;
var s:integer;
begin
  if n=0 then s:=0 else s:=round(n/abs(n));
  signe:=s;
end;

```

```

{*****
 * Cette fonction retourne le signe d'un reel *
*****}

function signe_reel(n:extended):integer;
var s:integer;
begin
if n = 0 then
begin
s:=0;
end
else
begin
if n < 0 then s:=-1 else s:= 1
end;
signe_reel:=s;
end;

function absoule(n:extended):extended;
begin
if signe_reel(n) < 0 then absoule:=-1 * n
else absoule:= n;
end;

{*****
 * Cette fonction écrit le nombre binaire *
 * correspondant à un nombre en base 10. *
*****}

procedure nb_binaire(foret:vecteur1; y:integer);
var a,i,j,k,l,m:integer;
begin
k:=0;m:=0;
for i:=1 to dimension do
begin
for j:=1 to dimension do
begin
a:=foret[i]^j;
for l:=1 to 8 do
begin
inc(k, a and 1);
outtextxy(80-10*l,y+10*m,nentext(a and 1));
a:=a shr 1;
end;
m:=m+1;
end;
end;
end;

```

```

{*****
 * Cette fonction retourne le nombre d'etres vivants *
 * d'une espece present a un noeud du reseau   *
*****}

function digit(entier:integer;espece:les_etres):integer;
var i,k,a:integer;
  no_individu:integer;
begin
  case espece of
    loup: no_individu:=1 ;
    lapin: no_individu:=2 ;
  end;
  a:= entier shr (4*(nb_individu-no_individu));
  k:=0;
  for i:=1 to 4 do
  begin
    inc(k,a and 1);
    a:=a shr 1;
  end;
  digit:=k;
end;

{*****
 * Cette fonction evalue le polynome representant *
 * la loi d'action de masse   *
*****}

function polynome(etre:les_etres;x,y:extended; var kappa:coefficient):extended;
var
  poly,a,b:extended;
  i,j:integer;
begin
  poly:=0;
  for i:=0 to nb_direction do
  begin
    for j:=0 to nb_direction do
    begin
      for j:=0 to nb_direction do
      begin
        a:=expose_reel(x,i) * expose_reel(y,j);
        b:=comb(i,4)*comb(j,4);
        poly:=poly + kappa[etre,i,j] * b * a / exposant(4,i+j);
      end;
    end;
    polynome:=poly;
  end;

```

```

*****+
* Cette fonction effectue une rotation de pi/2 ou de 3pi/2 *
* de la configuration des etres a un noeud donne. *
*****+}

function rotation(etat_etre:integer):integer;
  var alea,bit_de_gauche,bit_de_droite:integer;
begin
  alea:=round(int(2*random));
  case alea of
    { rotation de pi/2 }
  0:begin
    bit_de_gauche:=etat_etre shr 3;
    etat_etre:=etat_etre - bit_de_gauche * exposant(2,3);
    etat_etre:=etat_etre shl 1;
    etat_etre:=etat_etre or bit_de_gauche;
    rotation:=etat_etre;
  end;
    { rotation de 3pi/2 }
  1:begin
    bit_de_droite:=etat_etre and 1;
    etat_etre:=etat_etre shr 1;
    etat_etre:=etat_etre + bit_de_droite * exposant(2,3);
    rotation:=etat_etre;
  end;
  end;
end;

*****+
* Cette fonction retourne le nombre entre 0 et 15 *
* qui correspond a la disposition des etres a un *
* noeud quelconque du reseau de la foret. *
*****+}

function etat_espece(espece:les_etres;etat:integer):integer;
  var no_individu,nombre,i,p:integer;
begin
  case espece of
    loup: no_individu:=1 ;
    lapin: no_individu:=2 ;
  end;
  nombre:=0;
  etat:=etat shr (4*(nb_individu-no_individu));
  for i:=1 to 4 do
  begin
    p:=exposant(2,i-1);
    nombre:=(nombre) or (etat and p);
  end;
  etat_espece:=nombre;
end;

```

```

*****+
* Cette fonction ajoute ou enleve le nombre *
* d'etres approprie au noeud (i,j) *
*****}

function ajuste_noeud(etat_etre,delta:integer):integer;

var alea,bit_1,copie_etat:integer;
begin
  copie_etat:=etat_etre;
  case signe(delta) of
    { au moins un etre sera supprimer }
    -1:begin
      if abs(delta)=digit(etat_etre,lapin) then
      begin
        ajuste_noeud:=0;
      end
      else
      begin
        while digit(etat_etre,lapin) > (digit(copie_etat,lapin) + delta) do
        begin
          alea:=round(int(4*random));
          bit_1:=etat_etre and exposant(2,alea);
          etat_etre:=etat_etre - bit_1;
        end;
        ajuste_noeud:=etat_etre;
      end;
    end;
    { aucun etre sera supprimer ou ajouter }
    0:begin
      ajuste_noeud:=etat_etre;
    end;

    { au moins un etre sera ajouter }
    1:begin
      if abs(delta)=4-digit(etat_entre,lapin) then
      begin
        ajuste_noeud:=15;
      end
      else
      begin
        while digit(etat_entre,lapin) < (digit(copie_etat,lapin) + delta) do
        begin
          alea:=round(int(4*random));
          bit_1:=(etat_entre xor 15) and exposant(2,alea);
          etat_entre:=etat_entre + bit_1;
        end;
        ajuste_noeud:=etat_entre;
      end;
    end;
  end;
end;

```

```

{*****
 * Cette procedure test l'accessibilite *
 * de la carte graphique de l'ordinateur. *
*****}

procedure verifigraphe(var grapheok:boolean);
var grdriver,errcode:integer;
begin
grdriver:=detect;
initgraph(grdriver,grmode,'c:\tp\bgi');
errcode:=graphresult;
if errcode = grOk then
begin
  grapheok:=true;
end
else
begin
  writeln('Erreur graphique:', grapherrmsg(errcode));
  grapheok:=false;
  readln;
end;
end;

{*****
 * Initialisation des structures de donnees *
*****}

procedure init_etiquette(var etiquette:vecteur2;premier:byte);
var i: word;
begin
if premier=0 then
begin
  new(etiquette[0]);
  new(etiquette[1]);
  new(etiquette[2]);
  new(etiquette[3]);
end;
for i:=0 to dimension+1 do etiquette[0]^:[i]:=0;
etiquette[1]^:=etiquette[0]^;
etiquette[2]^:=etiquette[0]^;
etiquette[3]^:=etiquette[0]^;
end;

{*****
 * Cette fonction transforme un nombre en texte *
*****}

function nentext(n:longint):string;
var s:string;
begin
  str(n:2,s); nentext:=s;
end;

```

```

function reentext(n:real):string;
var s:string;
begin
  str(n:2:0,s); reentext:=s;
end;

{*****
 * Ces procedures dessinent un rectangle pour *
 * indiquer que l'ordinateur est en execution. *
*****}

procedure peri_rect(droite,bas:word);
var l,h:word;
begin
  l:=getmaxx;
  h:=getmaxy;
  setcolor(blue);
  rectangle(l-(droite+1),101,l-(bas),9);
end;

procedure rempli_rect(couleur,indice,sup_indice,droite,bas:word);
var l,h:word;
begin
  l:=getmaxx;
  h:=getmaxy;
  setcolor(couleur);
  moveto(l-droite,100-round(89*indice/sup_indice));
  lineto(l-bas,100-round(89*indice/sup_indice));
end;

procedure init_stable(var stabilite:racine);
var j:integer;
  i:les_etres;
begin
  for i:=loup to lapin do
    for j:=1 to nb_direction do stabilite[i,j]:=0;
end;

procedure impri_stable(var stabilite:racine);
var j:integer;
  i:les_etres;
begin
  for j:=1 to nb_direction do
  begin
    cleardevice;
    outtextxy(10,j*10,'equilibre_loup = '+ reentext(stabilite[loup,j]));
    outtextxy(10,(j+1)*10,'equilibre_lapin = '+ reentext(stabilite[lapin,j]));
  end;
end;

```

```

procedure init_foret(var foret:vecteur1;premier:byte);
  var i:integer;
begin
  if premier=0 then
    begin
      for i:=0 to dimension+1 do new(foret[i]);
    end;
    for i:=0 to dimension+1 do foret[0]^i:=0;
    for i:=0 to dimension+1 do foret[i]^:=foret[0]^;
  end;

procedure init_bords(var bords:contour);
  var i:integer;
    cote:lesbords;
begin
  for cote:=haut to droite do
  begin
    for i:=0 to dimension+1 do
    begin
      bords[cote,i]:=0;
    end;
  end;
end;

procedure init_kappa(var kappa:coefficient);
var
  i,j:integer;
  etre:les_etres;
begin
  for etre:=loup to lapin do
  begin
    for i:=0 to nb_direction do
    begin
      for j:=0 to nb_direction do
      begin
        kappa[etre,i,j]:=0;
      end;
    end;
  end;
  kappa[loup,0,0]:=A00;
  kappa[loup,1,0]:=A10;
  kappa[loup,1,1]:=A11;
  kappa[loup,2,0]:=A20;
  kappa[lapin,0,0]:=B00;
  kappa[lapin,0,1]:=B01;
  kappa[lapin,1,1]:=B11;
  kappa[lapin,0,2]:=B02;
end;

```

```

procedure init_transition(var transition:automate);
var
  i,j,k,l:integer;
begin
  for i:=0 to nb_direction do
  begin
    for j:=0 to nb_direction do
    begin
      for k:=0 to nb_direction do
      begin
        for l:=0 to nb_direction do
        begin
          transition[i,j,k,l]:=0;
        end;
      end;
    end;
  end;
end;

procedure init_statistique(var stat_etre:echantillon);
var i:les_tres;
  j:estimateur;
  k:integer;
begin
  for k:=1 to nb_essai do
  for i:=loup to lapin do
    for j:=min to max do stat_etre[k,i,j]:=0;
  end;

  {*****
   * Cette procedure prepare les donnees de depart du programme. *
   * En particulier le fichier de sortie est assigne. De plus   *
   *           l'increment de la proportion est calcule.           *
  *****}
procedure init_fichier(var stat_echantillon,resultat_simulation,mat_transition,equation:text;
  var debut:boolean);
begin
  if debut then
  begin
    assign(mat_transition,'c:\andre\transion.dat');
    assign(resultat_simulation,'c:\andre\resultat_simulation.dat');
    assign(stat_echantillon,'c:\andre\stat_echantillon.dat');
    assign(equation,'c:\andre\equation.dat');
    rewrite(mat_transition);
    rewrite(resultat_simulation);
    rewrite(equation);
    rewrite(stat_echantillon);
    debut:=false;
  end
  else
  begin
    close(stat_echantillon);  rewrite(stat_echantillon);
  end;
end;

```

```

procedure ferme_fichier(var mat_transition,resultat_simulation,stat_echantillon,
                        equation:text);
begin
  close(mat_transition);
  close(resultat_simulation);
  close(stat_echantillon);
  close(equation);
  closegraph;
end;

procedure init(var foret:vecteur1; var bords:contour; var etiquette:vecteur2;
              var transition:automate; var stat_etre:echantillon;
              var kappa:coefficient; var nb_etre_max:longint);
begin
  init_foret(foret,0);
  init_bords(bords);
  init_etaquette(etiquette,0);
  init_transition(transition);
  init_statistique(stat_etre);
  init_kappa(kappa);
  init_transition(transition);
  nb_etre_max:=0;
end;

procedure ecrit_param_simul(var resultat_simulation:text);
begin
  writeln(resultat_simulation,' ****');
  writeln(resultat_simulation,' * Les donnees ci-dessous indiquent dans quelles *');
  writeln(resultat_simulation,' *   conditions la simulation a ete effectue. *');
  writeln(resultat_simulation,' ****');
  writeln(resultat_simulation);
  writeln(resultat_simulation,' A10 = ',A10:6:4);
  writeln(resultat_simulation,' A11 = ',A11:6:4);
  writeln(resultat_simulation,' A20 = ',A20:6:4);
  writeln(resultat_simulation,' B01 = ',B01:6:4);
  writeln(resultat_simulation,' B11 = ',B11:6:4);
  writeln(resultat_simulation,' B02 = ',B02:6:4);
  writeln(resultat_simulation,' temps_ecelle = ',temps_ecelle:6);
  writeln(resultat_simulation,' temps_relaxation = ',temps_relaxation:6);
  writeln(resultat_simulation,' densite_lapin_genere = ',densite_lapin_genere:5:4);
  writeln(resultat_simulation,' densite_loup_genere = ',densite_loup_genere:5:4);
  writeln(resultat_simulation,' temps_total = ',temps_total:6);
  writeln(resultat_simulation,' dimension = ',dimension:3);
  writeln(resultat_simulation,' nb_essai = ',nb_essai:3);
  writeln(resultat_simulation,' taille_echantillon = ',taille_echantillon:3);
  writeln(resultat_simulation,' nb_individu = ',nb_individu:2);
  writeln(resultat_simulation,' nb_direction = ',nb_direction:2);
  writeln(resultat_simulation);
end;

```

```

procedure commentaire(var fichier:text; var remarque:phrase; var kappa:coefficient);
begin
  writeln(fichier,'*****');
  writeln(fichier,' * Toutes les donnees qui suivent doivent etre lu en ligne      *');
  writeln(fichier,' * et par groupe de ',nb_individu,'. Elles indiquent respectivement, par
*');
  writeln(fichier,' * hierarchie decroissante de predation, le nombre d"etres      *');
  writeln(fichier,' * present a un temps t. Ces donnees, relatives sont le resultat      *');
  writeln(fichier,' * d'un echantillonnage uniforme portant sur ',temps_total,' donnees.      *');
  writeln(fichier,'*****');
  writeln(fichier);
  remarque[min]:='nombre minimal d"etres = ';
  remarque[moy]:=' nombre moyen d"etres = ';
  remarque[s_xbar]:= 'ecart s_xbas d"etres = ';
  remarque[max]:= 'nombre maximal d"etres = ';
  { remarque[stable]:= 'densite d"equilibre possible = ' }
end;

procedure ecrit_loi_action(var resultat_simulation:text; var kappa:coefficient);
var
  i,X:integer;
  coeff,epsilon:extended;
  etre:les_etres;
begin
  X:=88;
  epsilon:=0.000000000000001;
  writeln(resultat_simulation,'*****');
  writeln(resultat_simulation,' * La loi d"action de masse est la suivante : *');
  writeln(resultat_simulation,'*****');
  writeln(resultat_simulation);
  for etre:=loup to lapin do
  begin
    write(resultat_simulation,chr(X),' = ');
    for i:=0 to nb_direction do
    begin
      for j:=0 to nb_direction do
      begin
        coeff:=kappa[etre,i,j] * comb(i,4) * comb(j,4)/exposant(4,i+j);
        if abs(coeff) > epsilon then
        begin
          if i < nb_direction then write(resultat_simulation,coeff:8:7,'x_',i,' y_',j,'+');
          else write(resultat_simulation,coeff:8:7,'x_',i,' y_',j);
        end;
      end;
    end;
  end;
  writeln(resultat_simulation);
  X:=X+1;
end;
writeln(resultat_simulation);
writeln(resultat_simulation);
end;

```

```

{*****
 * Cette procedure calcul les points de stabilities. *
*****}

procedure pt_stable(var stabilite:racine; var kappa:coefficient);
var
  a,b,c,d,e,f,rho_loup,rho_lapin:extended;
begin
  init_stable(stabilite);
  a:=kappa[lapin,0,1]; b:=kappa[lapin,1,1]; c:=kappa[lapin,0,2];
  d:=kappa[loup,1,0]; e:=kappa[loup,1,1]; f:=kappa[loup,2,0];
  a:=absolute(a); b:=absolute(b); c:=absolute(c);
  d:=absolute(d); e:=absolute(e); f:=absolute(f);
  stabilite[loup,1]:=(a*e-d*c)/(e*b+c*f);
  stabilite[lapin,1]:=(b*d+a*f)/(e*b+c*f);
  stabilite[loup,2]:=0;
  stabilite[lapin,2]:=a/c;

  { rho_loup := ( absolue(A11 * B01) - absolue(A10 * B02) ) / (absolue(A11 * B11) + absolue(A20 * B02) );
    rho_lapin := ( absolue(A10 * B11) + absolue(A20 * B01) ) / (absolue(A11 * B11) + absolue(A20 * B02));
  }
  writeln(' stabilite[loup,1] = ',stabilite[loup,1]);
  writeln(' stabilite[lapin,1] = ',stabilite[lapin,1]);
  writeln(' rho_loup = ',rho_loup);
  writeln(' rho_lapin = ',rho_lapin);
  readkey;
end;

procedure ecrit_stable(var fichier:text; var stabilite:racine);
var
  i:integer;
  etre:les_etres;
begin
  writeln(fichier,'*****');
  writeln(fichier,' * Les densites d"equilibre possible sont les suivantes : *');
  writeln(fichier,'*****');
  writeln(fichier);
  for etre:=loup to lapin do
  begin
    for i:=1 to nb_direction do
    begin
      write(fichier,stabilite[etre,i]:10:7,chr(9));
    end;
    writeln(fichier);
  end;
  writeln(fichier);
  writeln(fichier);
end;

```

```

procedure ecrit_stat(var stat_echantillon:text; var courbe:graphe; no_essai:longint;
                     var stat_etre:echantillon; var kappa:coefficient);
var
  delta_kappa:extended;
  i,t:longint;
  estime:estimateur;
  etre:les_tres;
  rem:phrase;
begin
  commentaire(stat_echantillon,rem,kappa);
{ for i:=1 to taille_echantillon do
begin
  for etre:=loup to lapin do
  begin
    write(stat_echantillon,courbe[no_essai,etre,i],chr(9));
    { write(stat_echantillon,courbe[no_essai,etre,i],chr(9));}
  end;
  writeln(stat_echantillon);
  { if (i mod 10 = 0) then writeln(stat_echantillon);
end;}
  writeln(stat_echantillon);
for estime:=min to max do
begin
  { write(fichier,rem(estime));}
  for etre:=loup to lapin do
  begin
    write(stat_echantillon,stat_etre[no_essai,etre,estime]:10:3,chr(9));
  end;
  writeln(stat_echantillon);
  writeln(stat_echantillon);
end;

procedure impri_etiq(var etiq:vecteur2; x,y:integer);
var i,j:longint;
begin
  outtextxy(x,y-10,'etiquette =');
  for i:=0 to dimension+1 do
  begin
    for j:=0 to dimension+1 do
    begin
      outtextxy(30*j+x,y+10*i,nentext(etiq[i]^ [j]));
    end;
  end;
end;

```

•

```

{*****
 * Cette procedure ecrit les probabilites de *
 * transitions dans le fichier de sortie " stat " *
*****}

```

```

procedure ecrit_transition(var mat_transition:text; var transition:automate);
var
  i,j,k,l,n:integer;
  etre:les_etes;
begin
{ for etre:=total to lapin do
begin}
  writeln(mat_transition,'*****');
  writeln(mat_transition,' * Voici la matrice de transition pour les etres *');
  writeln(mat_transition,'*****');
  writeln(mat_transition);
n:=1;
for i:=0 to nb_direction do
begin
  for j:=0 to nb_direction do
  begin
    for k:=0 to nb_direction do
    begin
      for l:=0 to nb_direction do
      begin
        write(mat_transition,transition[i,j,k,l]:6:4,' ');
      end;
    end;
    writeln(mat_transition,' = ',delta_prob[etre,n,plus]:4:2);
    writeln(mat_transition,' = ',delta_prob[etre,n,moins]:4:2);
    writeln(mat_transition);
    n:=n+1;
  end;
end;
writeln(mat_transition);
{ end; }
end;

{procedure calcul_parametre(var resultat_simulation:text;kappa:coefficient;
  var stabilite:racine);
begin
  ecrit_loi_action(resultat_simulation,kappa);
  pt_stable(stabilite,kappa);
  ecrit_stable(resultat_simulation,stabilite);
end;}

procedure impri_foret(var foret:vecteur1; y:integer);
var i,j:longint;
begin
  for i:=0 to dimension+1 do
  begin
    moveto(120,y+10*i);
    for j:=0 to dimension+1 do
    begin
      begin
        outtext(nentext(foret[i]^[[j]]));outtext(' ');
      end;
    end;
  end;
end;

```

```

procedure impri_etre(var nb_etre:vecteur;x,y:integer;temps:longint;
                      stabilite:racine);
var
  equilibre_loup,equilibre_lapin:longint;
begin
  setcolor(white);
  outtextxy(x,y,'nb_loups =' + nentext(nb_etre[loup]));
  outtextxy(x,y+10,'nb_lapins =' + nentext(nb_etre[lapin]));
  outtextxy(x,y+20,'temps =' + nentext(temps));
  equilibre_loup:=round(dimension*dimension*stabilite[loup,1]);
  equilibre_lapin:=round(dimension*dimension*stabilite[lapin,1]);
  outtextxy(x,y+30,'equilibre_loup =' + nentext(equilibre_loup));
  outtextxy(x,y+40,'equilibre_lapin =' + nentext(equilibre_lapin));
end;

procedure impri_bord(var bords:contour);
var i:integer;
  cote:lesbords;
begin
  for cote:=haut to droite do
  begin
    for i:=0 to dimension+1 do
    begin
      writeln(bords[cote,i]);
    end;
    writeln;
  end;
end;

procedure message_erreur(var resultat_simulation:text);
begin
  writeln(resultat_simulation,'Erreur; probabilites de transitions negatives.');
  writeln(resultat_simulation,'La valeur des coefficients Aij_kl et Bij_kl sont inadmissibles *');
  writeln(resultat_simulation);
end;

function u(t:integer):integer;
begin
  if t < 0 then u:=0
  else u:=1;
end;

function etat_k(a1,a2:integer):integer;
begin
  etat_k := ( a1 * ( nb_direction + 1 ) ) + a2 + 1;
end;

procedure inverse_etat_k(var k,b1,b2:integer);
begin
  b1:=(k-1)div(nb_direction+1);
  b2:=(k-1)mod(nb_direction+1);
end;

```

```

procedure trouve_etre(a1,a2,b1,b2:integer;var etre:les_etres);
begin
  if b2-a2 = 0 then etre:=loup
  else etre:=lapin;
end;

procedure test_transition(var transition:automate;var transi_ok:boolean);
var
  a1,a2,b1,b2:integer;
begin
  for a1:=0 to nb_direction do
  begin
    for a2:=0 to nb_direction do
    begin
      for b1:=0 to nb_direction do
      begin
        for b2:=0 to nb_direction do
        begin
          if (transition[a1,a2,b1,b2]<0)or(transition[a1,a2,b1,b2]>1) then
          begin
            transi_ok:=false;
            writeln('transition[',a1,',',a2,',',b1,',',b2,']= ',transition[a1,a2,b1,b2]);
            writeln('Erreur sur les hypotheses des coefficients a_ij et b_ij');
            readkey;
          end;
        end;
      end;
    end;
  end;
end;

procedure transi_sans_signe(var transition:automate;a1,a2,b1,b2:integer;etre:les_etres);
var
  v1,v2:integer;
  a,b,c:extended;
begin
  transition[a1,a2,b1,b2]:=0;
  for v1:=0 to a1 do
  begin
    for v2:=0 to a2 do
    begin
      a:=kappa[etre,v1,v2];
      b:=comb(v1,a1) * exposant(nb_direction,v1) / comb(v1,nb_direction);
      c:=comb(v2,a2) * exposant(nb_direction,v2) / comb(v2,nb_direction);
      a:=a*b*c;
      transition[a1,a2,b1,b2]:=-a + transition[a1,a2,b1,b2];
    end;
  end;
end;

```

```

procedure trouve_lien_signe(var transition:automate;var kappa:coefficient);
var
  i,a1,a2,b1,b2,v1,v2:integer;
  etat_k_init,etat_k_fin,mod_etat_k_fin,delta:integer;
  a,b,c,somme_prob:extended;
  direction:lesbords;{ = (haut,bas,gauche,droite);}
  etre:les_etros;
begin
  for direction:=haut to droite do
  begin
    for a1:=0 to nb_direction do
    begin
      for a2:=0 to nb_direction do
      begin
        etat_k_init:=etat_k(a1,a2);      i:=ord(direction)+1;
        delta:=-5*u(i-1)+10*u(i-2)-6*u(i-3)+2*u(i-4);  etat_k_fin:=etat_k_init + delta;
        mod_etat_k_fin:=etat_k_fin mod 5;
        case direction of
          haut:begin
            if etat_k_fin < 1 then
            begin
              etat_k_fin:=etat_k_init - delta;
              inverse_etaet_k(etat_k_fin,b1,b2);
              trouve_etre(a1,a2,b1,b2,etre);
              transi_sans_signe(transition,a1,a2,b1,b2,etre);
            end;
            end;
          bas:begin
            if ((etat_k_fin < 31) and (etat_k_fin > 25)) then
            begin
              etat_k_fin:=etat_k_init - delta;
              inverse_etaet_k(etat_k_fin,b1,b2);
              trouve_etre(a1,a2,b1,b2,etre);
              transi_sans_signe(transition,a1,a2,b1,b2,etre);
            end;
            end;
          gauche:begin
            if mod_etat_k_fin = 0 then
            begin
              etat_k_fin:=etat_k_init - delta;
              inverse_etaet_k(etat_k_fin,b1,b2);
              trouve_etre(a1,a2,b1,b2,etre);
              transi_sans_signe(transition,a1,a2,b1,b2,etre);
            end;
            end;
          droite:begin
            if mod_etat_k_fin = 1 then
            begin
              etat_k_fin:=etat_k_init - delta;
              inverse_etaet_k(etat_k_fin,b1,b2);
              trouve_etre(a1,a2,b1,b2,etre);
              transi_sans_signe(transition,a1,a2,b1,b2,etre);
            end;
            end;
          end;
        end;
      end;
    end;
  end;
end;

```

```

function loup_saute_de_1(a1,a2,b1,b2:integer):boolean;
begin
  if ((abs(b1-a1)=1) and (b2-a2=0)) then
  begin
    loup_saute_de_1:=true;
  end
  else
  begin
    loup_saute_de_1:=false;
  end;
end;

function lapin_saute_de_1(a1,a2,b1,b2:integer):boolean;
begin
  if ((b1-a1=0) and (abs(b2-a2)=1)) then
  begin
    lapin_saute_de_1:=true;
  end
  else
  begin
    lapin_saute_de_1:=false;
  end;
end;

procedure ajuste_transition(var transition:automate);
var
  a1,a2,b1,b2:integer;
  somme_prob:extended;
begin
  somme_prob:=0;
  for a1:=0 to nb_direction do
  begin
    for a2:=0 to nb_direction do
    begin
      for b1:=0 to nb_direction do
      begin
        for b2:=0 to nb_direction do
        begin
          transition[a1,a2,b1,b2]:=transition[a1,a2,b1,b2]/temps_ecelle;
          somme_prob:=somme_prob + transition[a1,a2,b1,b2];
        end;
      end;
      transition[a1,a2,a1,a2]:=1-somme_prob;
      somme_prob:=0;
    end;
  end;
end;

```

```

procedure transi_avec_signe(var transition:automate;var
kappa:coefficient;a1,a2:integer;etre:les_etres);
  var
    v1,v2:integer;
    a,b,c:extended;
  begin
    for v1:=0 to a1 do
    begin
      for v2:=0 to a2 do
      begin
        case etre of
        loup:begin
          if signe_reel(kappa[loup,v1,v2]) < 0 then
          begin
            a:=-kappa[loup,v1,v2];
            b:=comb(v1,a1)*exposant(nb_direction,v1)/comb(v1,nb_direction);
            c:=comb(v2,a2)*exposant(nb_direction,v2)/comb(v2,nb_direction);
            a:=a*b*c;
            transition[a1,a2,a1-1,a2]:= a + transition[a1,a2,a1-1,a2];
          end;
          if signe_reel(kappa[loup,v1,v2]) > 0 then
          begin
            a:=kappa[loup,v1,v2];
            b:=comb(v1,a1)*exposant(nb_direction,v1)/comb(v1,nb_direction);
            c:=comb(v2,a2)*exposant(nb_direction,v2)/comb(v2,nb_direction);
            a:=a*b*c;
            transition[a1,a2,a1+1,a2]:= a + transition[a1,a2,a1+1,a2];
          end;
        end;
        end;
        lapin:begin
          if signe_reel(kappa[lapin,v1,v2]) < 0 then
          begin
            a:=-kappa[lapin,v1,v2];
            b:=comb(v1,a1)*exposant(nb_direction,v1)/comb(v1,nb_direction);
            c:=comb(v2,a2)*exposant(nb_direction,v2)/comb(v2,nb_direction);
            a:=a*b*c;
            transition[a1,a2,a1,a2-1]:= a + transition[a1,a2,a1,a2-1];
          end;
          if signe_reel(kappa[lapin,v1,v2]) > 0 then
          begin
            a:=kappa[lapin,v1,v2];
            b:=comb(v1,a1)*exposant(nb_direction,v1)/comb(v1,nb_direction);
            c:=comb(v2,a2)*exposant(nb_direction,v2)/comb(v2,nb_direction);
            a:=a*b*c;
            transition[a1,a2,a1,a2+1]:= a + transition[a1,a2,a1,a2+1];
          end;
        end;
        end;
      end;
    end;
  end;
end;

```

```

procedure cherche_bond_de_1(var transition:automate;var kappa:coefficient);
var
  a1,a2,b1,b2:integer;
  saut_du_loup,saut_du_lapin:boolean;
begin
  saut_du_loup:=false;
  saut_du_lapin:=false;
  for a1:=0 to nb_direction do
  begin
    for a2:=0 to nb_direction do
    begin
      for b1:=0 to nb_direction do
      begin
        for b2:=0 to nb_direction do
        begin
          saut_du_loup:=loup_saute_de_1(a1,a2,b1,b2);
          saut_du_lapin:=lapin_saute_de_1(a1,a2,b1,b2);
          if saut_du_loup then
          begin
            transition[a1,a2,a1+1,a2]:=0;
            transition[a1,a2,a1-1,a2]:=0;
            transi_avec_signe(transition,kappa,a1,a2,loup);
            saut_du_loup:=false;
          end;
          if saut_du_lapin then
          begin
            transition[a1,a2,a1,a2+1]:=0;
            transition[a1,a2,a1,a2-1]:=0;
            transi_avec_signe(transition,kappa,a1,a2,lapin);
            saut_du_lapin:=false;
          end;
        end;
      end;
    end;
  end;
end;

```

```

procedure calcul_transition(var transition:automate;var kappa:coefficient);
begin
  cherche_bond_de_1(transition,kappa);
  trouve_lien_signe(transition,kappa);
  if trouve_temps_ecelle then
  begin
    calcul_facteur_h(transition);
  end
  else
  begin
    ajuste_transition(transition);
  end;
end;

```

```

procedure cumule_transition(var transition:automate);
var
  i,j,k,l:integer;
  etre:les_etres;
  somme_prob:extended;
begin
  somme_prob:=0;
  for i:=0 to nb_direction do
  begin
    for j:=0 to nb_direction do
    begin
      for k:=0 to nb_direction do
      begin
        for l:=0 to nb_direction do
        begin
          somme_prob:=somme_prob + transition[i,j,k,l];
          transition[i,j,k,l]:=somme_prob;
        end;
        end;
        somme_prob:=0;
      end;
      end;
      end;
    procedure calcul_facteur_h(var transition:automate);
    var
      a1,a2,b1,b2:integer;
      h,somme_prob:extended;
    begin
      h:=0;
      for a1:=0 to nb_direction do
      begin
        for a2:=0 to nb_direction do
        begin
          somme_prob:=0;
          for b1:=0 to nb_direction do
          begin
            for b2:=0 to nb_direction do
            begin
              somme_prob:=somme_prob + transition[a1,a2,b1,b2];
            end;
            end;
            h:=maximum_reel(h,somme_prob);
          end;
          end;
        end;
        writeln('Le temps d"echelle h doit etre plus grand que : ',h);
        writeln('Arreter l"execution et affecter un temps d"echelle approprie');
        readkey;
      end;
    end;
  
```

```

procedure creer_loup(var foret:vecteur1);
  var
    i,j,nb_loup_genere,compte_loup:longint;
    intersection,nb_bit_loup,etat_loup:integer;
  begin
    nb_loup_genere:=round(dimension * dimension * densite_loup_genere);
    outtextxy(10,10,'nb_loup_genere = '+ nentext(nb_loup_genere));
    if densite_loup_genere <= 2.5 then
      begin
        nb_bit_loup:=1;
        compte_loup:=0;
        while compte_loup < nb_loup_genere do
          begin
            i:= 1 + round( int( dimension * random ) );
            j:= 1 + round( int( dimension * random ) );
            etat_loup:=exposant(2,round(4+int(4*random)));
            intersection:=digit( etat_loup and foret[i]^j ,loup);
            compte_loup:=compte_loup + nb_bit_loup - intersection;
            foret[i]^j:= foret[i]^j or etat_loup ;
          end;
        end
      else
        begin
          for i:=1 to dimension do
            for j:=1 to dimension do foret[i]^j:=foret[i]^j or 240;
          nb_bit_loup:=1;
          compte_loup:= 4 * dimension * dimension;
          while compte_loup > nb_loup_genere do
            begin
              i:= 1 + round( int( dimension * random ) );
              j:= 1 + round( int( dimension * random ) );
              etat_loup:=exposant(2,round(4+int(4*random)));
              intersection:=digit( etat_loup and foret[i]^j ,loup);
              compte_loup:=compte_loup - intersection;
              foret[i]^j:=foret[i]^j and not etat_loup;
            end;
          end;
        end;
      end;
    
```

```

procedure creer_lapin(var foret:vecteur1);
var
  i,j,nb_lapin_genere,compte_lapin:longint;
  intersection,nb_bit_lapin,etat_lapin:integer;
begin
  nb_lapin_genere:=round(dimension * dimension * densite_lapin_genere);
  outtextxy(10,20,'nb_lapin_genere = ' + nentext(nb_lapin_genere));
  if densite_lapin_genere <= 2.5 then
  begin
    nb_bit_lapin:=1;
    compte_lapin:=0;
    while compte_lapin < nb_lapin_genere do
    begin
      i:= 1 + round( int( dimension * random ) );
      j:= 1 + round( int( dimension * random ) );
      etat_lapin:=exposant(2,round(int(4*random)));
      intersection:=digit( etat_lapin and foret[i]^j ,lapin);
      compte_lapin:=compte_lapin + nb_bit_lapin - intersection;
      foret[i]^j:=foret[i]^j or etat_lapin;
    end;
  end
  else
  begin
    for i:=1 to dimension do
      for j:=1 to dimension do foret[i]^j:=foret[i]^j or 15;
    compte_lapin:= 4 * dimension * dimension;
    nb_bit_lapin:=1;
    while compte_lapin > nb_lapin_genere do
    begin
      i:= 1 + round( int( dimension * random ) );
      j:= 1 + round( int( dimension * random ) );
      etat_lapin:=exposant(2,round(int(4*random)));
      intersection:=digit( etat_lapin and foret[i]^j ,lapin);
      compte_lapin:=compte_lapin - intersection;
      foret[i]^j:=foret[i]^j and not etat_lapin;
    end;
  end;
end;

{*****
 * Cette procedure genere deux especes vivantes. *
 * Pour l'instant aucune distribution n'est privilegier *
*****}

```

```

procedure genere_vie(var foret:vecteur1);
var
  genere_lapin,genere_loup,premier:boolean;
begin
  if densite_loup_genere >= 0 then genere_loup:=true;
  if densite_lapin_genere >= 0 then genere_lapin:=true;
  if genere_lapin then creer_lapin(foret);
  if genere_loup then creer_loup(foret);
end;

```

```

*****
* Cette procedure compte le nombre d'etre de chaque *
* especes vivantes qu'il y a dans la foret initialement. *
*****}

procedure compte_etre(var foret:vecteur1; var nb_etre:vecteur;var stabilite:racine);
var
  i,j,no_individu,entier:integer;
  k:les_etres;
begin
  for k:=loup to lapin do nb_etre[k]:=0;
  for i:=1 to dimension do
  begin
    for j:=1 to dimension do
    begin
      entier:=foret[i]^j;
      for k:=lapin downto loup do
      begin
        nb_etre[k]:=nb_etre[k] + digit(entier,k);
      end;
      end;
      end;
    { impri_etre(nb_etre,10,30,temp,stabilit);
      readkey;}
  end;

```

```

procedure rempli_bord(i:integer;var foret:vecteur1;var etiquette:vecteur2);
var
  m:integer;
begin
  for m:=0 to dimension+1 do
  begin
    foret[0]^m:=etiquette[0]^m;
    foret[i-1]^m:=etiquette[1]^m;
    foret[i]^m:=etiquette[2]^m;
    foret[dimension+1]^m:=etiquette[3]^m;
  end;
  { frontiere(foret,bords);}
  for m:=0 to dimension+1 do
  begin
    foret[1]^m:=foret[1]^m or foret[dimension+1]^m;
    foret[dimension+1]^m:=0;
    foret[dimension]^m:=foret[dimension]^m or foret[0]^m;
    foret[0]^m:=0;
    foret[m]^1:=foret[m]^1 or foret[m]^dimension+1;
    foret[m]^dimension+1:=0;
    foret[m]^dimension:=foret[m]^dimension or foret[m]^0;
    foret[m]^0:=0;
  end;
end;

```

```

{*****
 * Cette procedure calcule l'estimation ponctuel *
 * pour la moyenne, la variance ainsi que l'etendue. *
*****}

procedure statistique_etre(k,temp:longint; var nb_etre:vecteur;
                           var stat_etre:echantillon);
var
n:longint;
i:les_etres;
moy_carre:extended;
begin
if temp=temp_relaxation+1 then
begin
  stat_etre[k,loup,min]:=nb_etre[loup];
  stat_etre[k,lapin,min]:=nb_etre[lapin];
end;
for i:=loup to lapin do
begin
  stat_etre[k,i,min]:=minimum(round(stat_etre[k,i,min]),nb_etre[i]);
  stat_etre[k,i,moy]:=stat_etre[k,i,moy] + nb_etre[i];
  stat_etre[k,i,max]:=maximum(round(stat_etre[k,i,max]),nb_etre[i]);
  stat_etre[k,i,s_xbar]:=stat_etre[k,i,s_xbar]+nb_etre[i]*nb_etre[i];
end;
if temp=temp_total then
begin
  n:=temp_total-temp_relaxation;
  for i:=loup to lapin do
  begin
    stat_etre[k,i,moy]:=stat_etre[k,i,moy]/n;
    moy_carre:=stat_etre[k,i,moy] * stat_etre[k,i,moy];
    stat_etre[k,i,s_xbar]:=stat_etre[k,i,s_xbar]/n - moy_carre;
    stat_etre[k,i,s_xbar]:=sqrt(stat_etre[k,i,s_xbar]);
  end;
end;
end;

procedure rempli_interieur(i:integer;var foret:vecteur1;var etiquette:vecteur2);
var
m:integer;
begin
for m:=0 to dimension+1 do
begin
  foret[i-1]^ [m]:=etiquette[1]^ [m];
  etiquette[1]^ [m]:=etiquette[2]^ [m];
  etiquette[2]^ [m]:=etiquette[3]^ [m];
  etiquette[3]^ [m]:=0;
end;
end;

```

```

{*****
 * Cette procedure gere le deplacement des etres dans la foret *
 * pendant que les collisions s'effectuent.se soit fait. *
*****}

procedure distribue(i,j:integer; var foret:vecteur1; var bords:contour;
                    var tampon:vecteur;var etiquette:vecteur2);
var
  no_direction,no_individu,l,m,p:integer;
  h:les_etres;
begin
  if (i > 1) then l:= 2 else l:=1;
  for h:=loup to lapin do
  begin
    case h of
      loup: no_individu:=1 ;
      lapin: no_individu:=2 ;
    end;
    for no_direction:=0 to nb_direction-1 do
    begin
      p:=exposant(2,4-4*(no_individu-1) + no_direction );
      if ((tampon[h] and p) <> 0 ) then
      begin
        case no_direction of
          0:begin
            etiquette[l]^ [j-1]:=((etiquette[l]^ [j-1]) or (tampon[h] and p));
          end;
          1:begin
            etiquette[l+1]^ [j]:=((etiquette[l+1]^ [j]) or (tampon[h] and p));
          end;
          2:begin
            etiquette[l]^ [j+1]:=((etiquette[l]^ [j+1]) or (tampon[h] and p));
          end;
          3:begin
            etiquette[l-1]^ [j]:=((etiquette[l-1]^ [j]) or (tampon[h] and p));
          end;
        end;
      end;
    end;
  end;
  if ((i=dimension) and (j=dimension)) then
  begin
    rempli_bord(i,foret,etiquette);
  end;
  if ((i > 1) and (i <> dimension) and (j=dimension)) then
  begin
    rempli_interieur(i,foret,etiquette);
  end;
end;

```

```

{*****
 * Cette procedure code, pour l'ecran, l'état de la foret. *
 * Chaque couleur représente un état particulier de   *
 * l'automate cellulaire à la case (i,j) du réseau.    *
 * Ce réseau de taille dimension x dimension est ici   *
 * associé à l'écran. Les codes de couleurs utilisés,   *
 * pour reconnaître l'état de l'automate cellulaire,   *
 * sont les suivants :                                *
 *          *                                         *
 * bleu : nb de loups (seul) > 1.           *
 * blanc : nb de lapins (seul) > 1.           *
 * rouge : il y a : lapins et loups.          *
 * brun : il n'y a pas de vie à ce noeud.      *
 *****)}

```

```

procedure code_couleur(i,j:integer;var couleur:integer;
                      nb_loup,nb_lapin:integer);
begin
  case nb_loup of
    0: begin
      case nb_lapin of
        0: couleur:=red;      {brun sombre}
        1: couleur:=white;   {gris pale}
        2: couleur:=green;   {vert}
        3: couleur:=green;   {bleu}
        4: couleur:=green;   {blanc}
      end;
    end;
    1..4: begin
      case nb_lapin of
        0: begin
          couleur:=blue; {bleu}
        end;
        1..4: begin
          couleur:=lightgray; {gris pale}
        end;
      end;
    end;
  end;
end;

```

```

{*****
 * Cette procedure definie comment les interactions, a un noeud fixe, *
 * se produisent. A ce titre cette procedure est le noyau essentiel   *
 * du programme. Du moins c'est elle qui fera en sorte que le modele   *
 * utilise sera plausible ou non
*****}

procedure interaction(var manger_par_le:vectboolean; var transition:automate;
                      var nb_loup,nb_lapin:longint);
var
alea,b:extended;
init_loup,init_lapin,final_lapin,final_loup,
delta_loup,delta_lapin:integer;
begin
  manger_par_le[lapin]:=false;
  manger_par_le[loup]:=false;
  alea:=random;
  init_loup:=nb_loup;
  init_lapin:=nb_lapin;
  final_loup:=0;
  final_lapin:=0;
  b:=transition[init_loup,init_lapin,final_loup,final_lapin];
  while b < alea do
begin
  final_lapin:=final_lapin+1;
  if final_lapin > nb_direction then
begin
  final_lapin:=0;
  final_loup:=final_loup+1;
end;
  b:=transition[init_loup,init_lapin,final_loup,final_lapin];
  if final_loup > nb_direction then
begin
  restorecrtmode;
  writeln('il y a un probleme avec la procedure interaction');
  readln;
end;
end;
  delta_loup:= final_loup - init_loup ;
  nb_loup := nb_loup + delta_loup;
  delta_lapin:= final_lapin - init_lapin ;
  nb_lapin := nb_lapin + delta_lapin;
  if (abs(delta_loup) > 1) or (abs(delta_lapin) > 1) then
begin
  outtextxy(100,100,'Erreur;transition de plus de 1 changement.');
  readkey;
end;
end;

```

```

*****
* Cette procedure s'occupe d'affecter correctement les noeuds *
* de la foret apres que les collision entre les etres se soit fait. *
*****
procedure collision(var foret:vecteur1; var bords:contour;var nb_etre,tampon:vecteur;
                    var temps,nb_etre_max:longint);
var
  i,j,no_individu,etat,alea,couleur:integer;
  loup_avant,lapin_avant,loup_apres,lapin_apres:longint;
  delta_lapin,delta_loup:integer;
  max,stabilite_population:longint;
begin
for i:=1 to dimension do
begin
  for j:=1 to dimension do
  begin
    etat:=foret[i]^j];
    loup_avant:=digit(etat,loup);
    lapin_avant:=digit(etat,lapin);
    loup_apres:=loup_avant;
    lapin_apres:=lapin_avant;
    nb_etre_max:=maximum(maximum(nb_etre[loup],nb_etre_max),nb_etre[lapin]);
    code_couleur(i,j,couleur,loup_avant,lapin_avant);
    putpixel(i,j,couleur);
    interaction(manger_par_le,transition,loup_apres,lapin_apres);
    delta_lapin:=lapin_apres - lapin_avant;
    delta_loup:=loup_apres - loup_avant;
    nb_etre[loup]:=nb_etre[loup] + delta_loup;
    nb_etre[lapin]:=nb_etre[lapin] + delta_lapin;
    if loup_apres <> 0 then
    begin
      tampon[loup]:=rotation(ajuste_noeud(etat_espece(loup,etat),delta_loup));
      tampon[loup]:=tampon[loup] shl 4;
    end
    else
    begin
      tampon[loup]:=0;
    end;
    if lapin_apres <> 0 then
    begin
      tampon[lapin]:=rotation(ajuste_noeud(etat_espece(lapin,etat),delta_lapin));
      tampon[lapin]:=tampon[lapin] shl 0;
    end
    else
    begin
      tampon[lapin]:=0;
    end;
    delta_etre[loup]:=loup_avant-loup_apres;
    delta_etre[lapin]:=lapin_avant-lapin_apres;
    etat_etre[loup]:=etat_espece(loup,etat);
    etat_etre[lapin]:=etat_espece(lapin,etat);
    distribue(i,j,foret,bords,tampon,etiquette);
  end;
  rempl_rect((temps mod 14)+1,i,dimension,50,42); end; end;

```

```

procedure point_ecran(var fichier:text;var nb_loup,nb_lapin,i,passe:longint;
                      proportion_trop,dilate_y:extended;supprime:boolean);

var l,h:longint;
    alea:extended;
begin
l:=getmaxx;
h:=getmaxy;
if supprime then
begin
alea:=random;
if alea > proportion_trop then
begin
  read(fichier,nb_etre[loup],nb_etre[lapin]);
  passe:=passe+1;
end
else
begin
  read(fichier,nb_etre[loup],nb_etre[lapin]);
  moveto(i-1-passe,h-round(nb_loup*dilate_y));
  setcolor(lightblue);
  lineto(i-passe,h-round(nb_etre[loup]*dilate_y));
  moveto(i-1-passe,h-round(nb_lapin*dilate_y));
  setcolor(lightgray);
  lineto(i-passe,h-round(nb_etre[lapin]*dilate_y));
  nb_loup:=nb_etre[loup];
  nb_lapin:=nb_etre[lapin];
end;
end
else
begin
  read(fichier,nb_etre[loup],nb_etre[lapin]);
  moveto(i-1,h-round(nb_loup*dilate_y));
  setcolor(lightblue);
  lineto(i,h-round(nb_etre[loup]*dilate_y));
  moveto(i-1,h-round(nb_lapin*dilate_y));
  setcolor(lightgray);
  lineto(i,h-round(nb_etre[lapin]*dilate_y));
  nb_loup:=nb_etre[loup];
  nb_lapin:=nb_etre[lapin];
end;
end;

```

```

procedure trace_courbe(var fichier:text; nb_etre:vecteur;var stabilite:racine;
                      nb_etre_max,temp:longint);

var
  x,y,i,l,h,nb_loup,nb_lapin,passe,position_relax,
  equilibre_loup,equilibre_lapin:longint;
  proportion_trop,alea,dilate_y,dilate_x:extended;
  supprime:boolean;

begin
  cleardevice;
  l:=getmaxx;
  h:=getmaxy;
  setcolor(white);
  moveto(0,h);
  lineto(l,h);
  moveto(0,h);
  lineto(0,0);
  equilibre_lapin:=round(dimension * dimension * stabilite[lapin,1]);
  equilibre_loup:=round(dimension * dimension * stabilite[loup,1]);
  nb_etre_max:=maximum(nb_etre_max,equilibre_loup);
  nb_etre_max:=maximum(nb_entre_max,equilibre_lapin);
  dilate_x:=1;
  dilate_y:=(h/nb_entre_max);
  proportion_trop:=l/temp;
  passe:=0;
  supprime:=false;
  if proportion_trop < 1 then
    begin
      supprime:=true;
      position_relax:=round(temp_relaxation * proportion_trop);
    end
  else
    begin
      position_relax:= temp_relaxation;
    end;
  reset(fichier);
  read(fichier,nb_etre[loup],nb_etre[lapin]);
  nb_loup:=nb_etre[loup];
  nb_lapin:=nb_etre[lapin];
  moveto(round(1),h-round(nb_etre[loup]*dilate_y));
  for i:=2 to temp do
    begin
      point_ecran(fichier,nb_loup,nb_lapin,i,passe,proportion_trop,dilate_y,supprime);
      setcolor(lightgray);
      y:= h-round(equilibre_lapin * dilate_y);
      moveto(0,y);
      lineto(639,y);
      setcolor(lightblue);
      y:= h-round(equilibre_loup * dilate_y);
      moveto(0,y);
      lineto(639,y);
      setcolor(yellow);
    end;
end;

```

```

x:= position_relax;
moveto(x,y);
setcolor(yellow);
lineto(x,100);
outtextxy(x,y+105,'temps de relaxation');
moveto(x,y);
lineto(x-10,y+10);
moveto(x,y);
lineto(x+10,y+10);
end;
end;

procedure choisit_point(var fichier:text; var courbe:graphe;no_essai,temp:longint);

var
  t,l,h,compte,posi,nb_loup,nb_lapin:longint;
  proportion_trop,alea:extended;
  supprime:boolean;

begin
l:=getmaxx;
h:=getmaxy;
proportion_trop:=l/temp;
compte:=1;
posi:=10;
supprime:=false;
reset(fichier);
if proportion_trop < 1 then supprime:=true;
for t:=1 to temp do
begin
  if supprime then
    begin
      alea:=random;
      if alea > proportion_trop then
        begin
          read(fichier,nb_loup,nb_lapin);
        end
      else
        begin
          if compte <= taille_echantillon then
            begin
              read(fichier,nb_loup,nb_lapin);
              courbe[no_essai,lapin,compte]:=nb_lapin;
              courbe[no_essai,loup,compte]:=nb_loup;
              compte:=compte+1;
            end;
          end;
        end
      else
        begin
          read(fichier,nb_loup,nb_lapin);
          courbe[no_essai,lapin,t]:=nb_lapin;
          courbe[no_essai,loup,t]:=nb_loup;
        end;
    end;
  end;
end;

```

```

procedure calcul_et_ecrit(var fichier_stat:text;kappa:coefficient; var transi_ok:boolean;var
stabilite:racine;var transition:automate);
begin
  ecrit_loi_action(fichier_stat,kappa);
  pt_stable(stabilite,kappa);
  ecrit_stable(fichier_stat,stabilite);
  calcul_transition(transition,kappa);
  test_transition(transition,transi_ok);
{  ecrit_transition(mat_transition,transition); }
  cumule_transition(transition);
  test_transition(transition,transi_ok);
{  ecrit_transition(mat_transition,transition);}
end;

{*****
 * Cette procedure ecrit les resultats portant sur le systeme proie et   *
 * predateur.Il y a " 2 x temps_final " donnees d'ecrite dans ce fichier. *
 * Les resultats sont ecris par ligne. La premiere valeur est le nombre   *
 * de loups restant sur le reseau de taille " dimension x dimension "   *
 * La deuxieme valeur est le nombre de lapins restant sur le reseau de   *
 * taille " dimension x dimension ". Le t-ieme coupe de donnees          *
 * correspond respectivement au temps t.                                     *
*****}
procedure ecrit_fichier(var nb_etre:vecteur; i:longint; var fichier:text);
begin
  writeln(fichier,nb_etre[loup]:20,nb_etre[lapin]:20);
end;

procedure resultat_actif(var fichier:text;nb_etre:vecteur;stabilite:racine);
begin
  cleardevice;
  peri_rect(51,41);
  impri_etre(nb_etre,400,340,temps,stabilite);
  ecrit_fichier(nb_etre,temps,fichier);
end;

```

```

{*****
 * Debut du programme principal. *
*****}

begin
{randomize;}
debut:=true;
transi_ok:=true;
verifigraphe(graphique_ok);
setgraphmode(gremode);
cleardevice;
init_fichier(stat_echantillon,resultat_simulation,mat_transition,equation,debut);
ecrit_param_simul(resultat_simulation);
for no_essai:=1 to nb_essai do
begin
  init(foret,bords,etiquette,transition,stat_etre,kappa,nb_etre_max);
  genere_vie(foret);
  compte_etre(foret,nb_etre,stabilite);
  temps_ecelle:=temps_ecelle + delta_temps_ecelle;
  calcul_et_ecrit(resultat_simulation,kappa,transi_ok,stabilite,transition);
  if transi_ok then
  begin
    for temps:=1 to temps_total do
    begin
      resultat_actif(stat_echantillon,nb_etre,stabilite);
      if temps > temps_relaxation then
      begin
        statistique_etre(no_essai,temps,nb_etre,stat_etre);
      end;
      collision(foret,bords,nb_etre,tampon,temps,nb_etre_max);
      init_etaquette(etiquette,1);
    end;
    close(stat_echantillon);
    trace_courbe(stat_echantillon,nb_etre,stabilite,nb_etre_max,temps_total);
  readln;
  choisit_point(stat_echantillon,courbe,no_essai,temps_total);
  ecrit_stat(resultat_simulation,courbe,no_essai,stat_etre,kappa);
  ecrit_stable(resultat_simulation,stabilite);
  { init_fichier(stat_echantillon,resultat_simulation,mat_transition,equation,debut);}
end
else
begin
  { message_erreur(fichier_stat,kappa,kappa_sup);}
end;
end;
ferme_fichier(mat_transition,resultat_simulation,stat_echantillon,equation);
end.

{*****
 * Fin du programme principal. *
*****}

```

BIBLIOGRAPHIE

1. X.G. WU & R.KAPRAL, Effect of Fluctuations on Chemical Oscillations and Chaos, J. Chem. Phys., Vol 100, No 8, (1994), pp. 5936-5948.
2. X.G. WU & R.KAPRAL, Chemical Chaos and Internal Noise, Ann. N.Y. Acad. Sci. , Vol 706, No 13, (1993), pp. 186-195.
3. X.G. WU & R.KAPRAL, Internal Fluctuation and Deterministic Chemical Chaos , Am.Phys.Soc., Vol 70, No 13, (1993), pp. 1940-1943.
4. R.KAPRAL, ANNA LANWNICZAK & B.HASSLACHER, Molecular Turing Structures in the Biochemistry of the Cell, Chaos, Vol 3, No 1, (1993), pp. 7-13.
5. E.MCCAULEY, W.G.WILSSON & A.M. DE ROOS, Dynamics of Age Structured and Spatially Structured Predator Prey Interactions : Individual Based Models and Population Level Formulations, Am. Nat, (1993), Vol. 142, No 3, pp. 412-442.
6. R.KAPRAL, ANNA LANWNICZAK & S.P. DAWSON, Interaction of Turing and Flow Induced Chemical Instability, J. Chem. Phys., Vol 100, No 7, (1994), pp. 5211-5218.
7. R.KAPRAL, ANNA LANWNICZAK & P.MASCIAR, Reactive Dynamics in a Multispecies Lattice Gas Automaton, J. Chem. Phys., Vol 96, No 4, (1992), pp. 2762-2776.
8. R.KAPRAL, ANNA LANWNICZAK & P.MASCIAR, Oscillations and Waves in a Reactive Lattice Gas Automaton, Am. Phys. Soc., Vol 66, No 19, (1991), pp. 2539-2543.
9. R.KAPRAL, D.DAB, ANNA LANWNICZAK & J.P. BOON, Cellular Automaton Model for Reactive Systems, Am. Phys. Soc., Vol 64, No 20, (1990), pp. 2462-2465.

10. D.DAB, J.P. BOON & Y.X LI, Lattice-Gas Automata for Coupled-Reaction Diffusion Equations, Am. Phys. Soc., Vol 66, No 19, (1991), pp. 2535-2538.
11. ANNA LANWNICZAK, D.DAB, R.KAPRAL & J.P. BOON, Reactive Lattice Automata, Physica D 47 (1991) 132-158.
12. T.KARAPIPERIS, Comparaison of Cellular Automata Approach and Differential Equations Approaches to Reaction Transport Processes, ed. Perdang Lajeune, World Scientifics, (19??), pp. 187-199.
13. I.G. DARVEY & J.P. STAFF, Stochastic Approach Model for First Order Chemical Reaction Kinetics, J. Chem. Phys., Vol 44, No 3, (1966), pp. 990-997.
14. I.G. DARVEY, B.W. NINHAM & J.P. STAFF, Stochastic Model for Second Order Chemical Reaction Kinetics. The Equilibrium State, J. Chem. Phys., Vol 45, No 6, (1966), pp. 2145-2154.
15. G.NICHOLIS, K.BAESEN, Phénomènes non linéaire, ed. ???, Maison ???, (19??),
16. B.CHOPARD, M.DROZ, S.CORNELL & L.FRANCHEBOURG, Cellular Automata Approach to reaction diffusion systems : Theory and Applications, in Cellular Automata Prospect in Astrophysical Applications, ed. Perdang Lajeune, World Scientifics, (19??), pp. 157-185.
17. A.DeMASI, R.ESPOSITO, J.L.LEBOWITZ & E.PRESUTTI, Rigorous Result on Some Stochastic Cellular Automata, ed. ???, Maison ???, (19??), pp. 92-101.
18. A.F BARTHOLOMAY, On the Linear Birth and Death Processes of Biology as Markoff Chains, Bull. Math. Biophys, Vol 20, (1958), pp. 97-118.
19. A.F BARTHOLOMAY, Molecular set theory : a Mathematical representation for Chemical Reaction Mecanisms, Bull. Math. Biophys, Vol 22, (1960), pp. 285-307
20. B.WINDOM, Collision theory of chemical rates, Adv. Chem. Phys., Vol 5, (1962), pp.253-386

21. GARY D. DOOLEN, Lattice Gas Methods for PDE's, North Holland, (1991), 339 pages.
22. N.G. VAN KAMPEN, Stochastic Processes in Physics and Chemistry, North Holland , (1983) , 419 pages.
23. J.D.MURRAY, Mathematical Biology, Springer Verlag , (1993) , 745 pages.
24. S.KARLIN & H.M. TALOR, A Second Course in Stochastic Processus, Academic Press , (1981) , 539 pages.
25. J.HOFBAUER & K. SIGMUND, The Theory of Evolution and Dynamical System , Cambridge University Press, (1988) , 337 pages.
26. J.MAYNARD SMITH, Mathématique Ideas in Biology, Cambridge University Press , (1968) , 149 pages.
27. P.C. FIFE, Mathematical Aspect of Reacting and Diffusing System, Springer Verlag , (19??) , ??? pages.
28. A.NOULLEZ, Automate de Gaz sur Réseau, Thèse de Doctorat, Université Libre de Bruxelles (1990) , 241 pages
29. L.COMPTET, Advanced Combinatoric, D.Reidel Publusing Compagny, Dordrecht-Holland/Boston, 343 pages.
30. J.LABELLE, Théorie des Graphes, Modulo Éditeur, (1981), 183 pages.
31. G.F.SIMMONS, Différential Equations with Applications and Historical Notes, Mc Graw Hill, (1972), 465 pages.
32. C.H.EDWARDS, D.E.PENNY Elementary Différential Equations with Applications, Prentice-Hall, Third edition, (1994), 634 pages.
33. J.A.THORPE, Elementary Topics in Différential Geometry, Sprigner-Verlag, Second edition, (1985), 253pages.

34. D.A. McQUARRIES, Stochastic Approach to Chemical Kinetics, J. Appl. Prob., Vol 4, (1967), pp. 413-478.
35. V.VOLTERRA, Leçons sur la Théorie Mathématique de la Lutte pour la Survie, Gauthier-Villard, Paris, (1931).
36. A.J.LOTKA, Elements of Mathematical Biology, Dover, New-York, pp.88-94, (1956).
37. E.R.LEIGH, Some Mathematical Problems in Biology, American Mathematical Society, Providence, R.I. , (1968).
37. D. GULICK, Encounters With Chaos McGraw-Hill, (1992), 308pages.