UNIVERSITE DE NICE-SOPHIA ANTIPOLIS
FACULTE DES SCIENCES

Laboratoire de Mathématiques

THESE

Présentée pour obtenir le titre de
Docteur en SCIENCES
(discipline : **mathématiques**)

par

Hervé ELMOZNINO

INFLUENCE DU CYCLE DE VIE INDIVIDUEL SUR LA DYNAMIQUE SPATIALE D'UNE FORET MONOSPECIFIQUE. ANALYSE A TRAVERS UN AUTOMATE CELLULAIRE.

Soutenue le 10 mai 1999 devant le jury composé de :

MM.	Claude	LOBRY	Directeur
	François	HOULLIER	Rapporteurs
	René	LOZI	
	Bernard	RIERA	
	Jean-Luc	GOUZE	Examinateur
	Alain	PAVE	Président

A l'INRIA, Sophia Antipolis, 14h30

Résumé

La banalisation des moyens de calcul puissants dans les sciences de la vie a entraîné l'utilisation de modèles de plus en plus complets, mais aussi de plus en plus compliqués, dans le but d'étudier les phénomènes naturels, en l'occurrence la dynamique forestière. Ce travail, centré sur l'aspect spatial de la dynamique d'une forêt monospécifique, pose la question de la pertinence de tels modèles. Pour cela, nous avons choisi comme support l'objet le plus immédiat pour la modélisation mathématique de phénomènes distribués spatialement : l'automate cellulaire.

Après avoir posé les bases formelles de l'automate cellulaire, nous définissons la classe "Mosaic Cycle", adaptée à la "modélisation" des phénomènes considérés.

Puis nous présentons les résultats d'autres auteurs, à propos d'un automate appartenant à cette classe. Nous présentons ensuite une généralisation de cet automate. Ne parvenant pas à généraliser les résultats précédents, nous développons d'autres outils d'investigation, pour parvenir à une bonne compréhension de ce nouvel automate.

Nous nous interrogeons ensuite sur les conditions qui pourraient permettre de remplacer notre modèle spatialisé par un simple modèle en classe de type Leslie densité-dépendant.

Nous étudierons enfin un automate inspiré du modèle de Ch. Wissel, présenté dans "The Mosaic-Cycle Concept of Ecosystems" (H. Remmert).

Mots-clés : modélisation, automates cellulaires, dynamique forestière, simulation, modèle spatialisé.

Remerciements

Mille remerciements à Claude LOBRY pour son soutien constant, malgré toutes mes excentricités d'écolo-bio-homme-naif-des-bois-de-la-montagne-qui-a-même-pas-l'électricité.

Mille remerciements aux membres du jury, François HOULLIER, René LOZI, Bernard RIERA, Jean-Luc GOUZE et Alain PAVE pour l'attention qu'ils ont portée à ce travail.

Table des matières

Introduction

La représentation mathématique des phénomènes ayant une extension spatiale se fait essentiellement avec des équations aux dérivées partielles, comme par exemple

$$\frac{\partial u}{\partial t} = k^2 \Delta u \qquad u_0 = f \qquad u|_\Omega = 0 \tag{1}$$

pour représenter un phénomène de diffusion, ou

$$\frac{\partial^2 u}{\partial t^2} = k \frac{\partial^2 u}{\partial x^2} \qquad u_0 = \phi \qquad u'_0 = \psi \qquad u(0,t) = u(l,t) = 0 \tag{2}$$

pour représenter les vibrations d'une corde tendue. De très nombreux types d'équations aux dérivées partielles interviennent donc pour représenter toutes sortes de phénomènes physiques, et leur étude mathématique, qui a commencé il y a plus de deux siècles, constitue une partie considérable du travail de recherche actuel.

En écologie, l'importance de la théorie actuelle des E.D.P. pour représenter des phénomènes ayant une extension spatiale, comme par exemple la compétition entre deux espèces de végétaux, est bien moins évidente. Il existe bien entendu des situations écologiques, et plus généralement des situations du domaine des sciences de la vie, qui sont représentées par des E.D.P. (comme on peut en trouver de nombreuses dans le livres de Winfree [Win80]), mais elles ne sont pas fréquentes. Il y a une raison évidente à cela : l'absence de lois connues régissant ces questions. On ne met pas en équation une forêt comme on met en équations l'écoulement d'un fluide !

La banalisation des moyens de calcul puissants dans les sciences de la vie a créé il y a une vingtaine d'années une situation nouvelle. Il est possible maintenant de décrire dans des langages informatiques très conviviaux des systèmes d'interaction plus ou moins complexes, et de simuler leur évolution. Nous aborderons dans ce travail un cas particulier de ces nouvelles techniques : celui des automates cellulaires. Ces objets combinatoires (nous nous expliquerons plus loin sur ce mot) ont été rendus célèbres par le *Jeu de la Vie* de Conway ([Del96]), et les structurations spatiales de Wolfram ([Wol86]), et ont été utilisés dans plusieurs circonstances dans les sciences de la vie ([ACC+93], [BAC+93]). Nous allons nous y intéresser dans le cas de leur utilisation pour la représentation de la dynamique spatiale de forêts et de mosaïques végétales, comme le font par exemple Kazunori et Iwasa ([KI93]) et Wissel ([Wis91]). Notre étude ne portera pas sur la modélisation de tel ou tel système par des automates cellulaires particuliers, mais sur les moyens méthodologiques à mettre en oeuvre pour une utilisation rigoureuse de cet outil.

Un automate cellulaire se caractérise par un ensemble de "cellules" (pour nous, des carrés), qui réalisent un pavage d'une partie du plan. Chaque cellule adopte un état, et cet état évolue au cours du temps selon certaines règles de transitions, qui font intervenir l'état de la cellule considérée et ceux des cellules voisines. L'ensemble des états possibles d'un automate est un ensemble fini, généralement petit (deux pour le Jeu de la Vie), et qui surtout n'est pas destiné à tendre vers l'infini, pas plus que la maille de la grille à tendre vers 0. Ce choix délibéré entraîne que les outils de l'analyse infinitésimale sont à priori inutilisables. On peut penser, à priori, que cela n'est pas important puisque la déduction mathématique sera remplacée par une simple simulation. D'ailleurs, il semble que cette possibilité d'échapper au traitement mathématique soit en grande partie à la base du succès de ces simulations. Examinons la situation d'un peu plus près. Supposons que nous ayons à connaître $x(t)$ pour $t = 1$, alors que $x(0) = 0.1$, et que x est régi par l'équation différentielle :

$$x'(t) = x(t)(1 - x(t)) \tag{3}$$

La résolution effective ne pose pas de problème. On trouve que :

$$x(t) = \frac{e^t}{9 + e^t} \tag{4}$$

donc que :

$$x(1) = \frac{e}{9 + e} \tag{5}$$

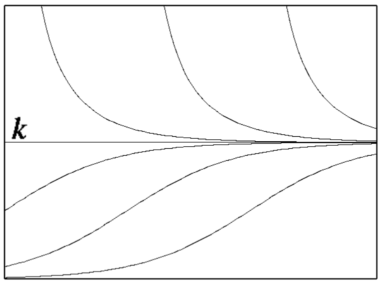
et une valeur approchée sera donnée par une calculette. Prenons maintenant l'automate cellulaire très simple du Jeu de la Vie. L'état d'une cellule à l'instant $t + 1$ vaut :

- 1, si son état à l'instant t est 0 et trois de ses voisines sont en 1,
- 1, si son état à l'instant t est 1, et que deux ou trois de ses voisines sont en 1,
- 0 sinon.

Pour connaître l'état après 10000 itérations, à partir d'une configuration donnée sur une grille de 100×100 cellules, il faudra quelques secondes de calcul à un PC moderne. Il semble que nous soyons dans la même situation que dans le cas de l'équation différentielle. En fait, il n'en est rien. Quand sommes nous en situation de vouloir prédire le futur à partir d'une condition initiale ? Seulement lorsque le modèle mathématique est une représentation fidèle de la réalité, et pour le savoir, un long travail est nécessaire, et ce travail repose sur une bonne **théorie du modèle** considéré. Qu'entendons nous par là ? Dans le cas de l'équation logistique :

$$x'(t) = rx(t)(1 - \frac{x(t)}{k}) \tag{6}$$

qui dépend des paramètres r et k, nous savons que toutes les solutions positives tendent vers un état d'équilibre $x_e = k$, que la croissance à partir d'une condition initiale petite est du type exponentiel, que les solutions pourront être données par une formule explicite. En gros, l'essentiel du comportement de (6) peut être résumé par le dessin et la formule suivante :



$$x(t) = \frac{ke^{rt}}{ck + e^{rt}}$$

Or une théorie du comportement du modèle est nécessaire pour comparer ce dernier à la réalité. Qu'est-ce qu'une théorie de phénomène dynamique ? Disons grossièrement qu'un modèle $M(\lambda)$ appartient à une classe de modèles $\mathcal{M}$ paramétrée par λ. En fonction de λ, on souhaite répondre à des questions du type :

- Y a-t-il un ou des comportements asymptotiques (états stationnaires, cycles, comportements chaotiques) ?
- Quelle est la durée des transitoires ?
- Comment sont structurées spatialement les solutions ?
- Quelle est la sensibilité par rapport aux paramètres ?
- Quelle est la robustesse par rapport à des bruits, par rapport à la structure du modèle ?

Prenons une classe $\mathcal{M}$ d'automates, sur une grille 100×100, prenant 2 états, dépendant d'un paramètre pouvant prendre 10 valeurs. L'étude directe de cet automate par des simulations systématiques est impossible. Il y a en effet 2^{10000} conditions initiales possibles. Il nous faut donc découvrir, si elles existent, les lois des automates de la classe $\mathcal{M}$. C'est ce travail que nous avons tenté, sur une classe d'automates que nous appelons *"AVN"*. Ce sont des automates dont la loi de transition dépend de deux paramètres entiers a et v tels que

$$1 \leq a < v \leq n - 1$$

où n est le nombre d'états possibles des cellules. Cette classe de modèles, outre le fait qu'elle a à priori un rapport avec la question des dynamiques végétales, a la particularité d'avoir été étudiée mathématiquement dans le cas particulier où $a = 1$. On trouvera dans [AR84] une théorie assez complète qui énonce en particulier que :

- Pour $v \geq 2$, toutes les solutions sont de période n.

L'automate considéré ici n'étant qu'une toute petite généralisation de celui étudié par Allouche et Reder, il semblait plausible que ses propriétés en seraient proches, et que les méthodes élaborées par Allouche et Reder pourraient être adaptées. Pour le premier point, les simulations ont montré que non; pour le second, nous n'avons pas su généraliser la méthode de Allouche et Reder, ni en trouver de nouvelles qui soient adaptées.

N'étant pas capable de proposer des résultats mathématiques significatifs, nous avons tenté une approche expérimentale. Notre analyse repose sur quelques dizaines de milliers de simulations mais, quoiqu'élevé, ce chiffre, nous l'avons remarqué, est dérisoire par rapport aux ordres de grandeur de 2^{10000} qui sont les nôtres. Il faut donc un minimum de méthode pour aborder le problème. Dans le cas des systèmes différentiels, les méthodes sont connues :

- recherche des équilibres,
- recherche des intégrales premières,
- linéarisation, etc ...

Dans le cas des automates, nous ne connaissons pas de méthode à priori. Nous en avons donc cherché une.

Nous avons défini une classe d'automates cellulaires, appelée *"Mosaic Cycle"*, basée sur le *"Mosaic Cycle Concept"* décrit dans [Rem91]. Les automates de cette classe représentent un couplage spatial entre des individus dont le cycle de vie passe par n stades successifs (par exemple, des classes de diamètres ou des types de végétation). Précisons un peu plus, en prenant un exemple :

Chaque cellule peut occuper un des états

$$0 \qquad 1 \qquad 2 \qquad \dots \qquad n-1$$

répartis en 4 groupes :

- $\{0\}$: l'état neutre,
- $\{1,2,\dots,a-1\}$: les états jeunes,
- $\{a,a+1,\dots,v-1\}$: les états adultes,
- $\{v,v+1,\dots,n-1\}$: les états vieux,

l'état 0 étant celui d'une place vide. On passe à chaque pas de temps de l'état e à l'état $e+1 \mod n$, sauf en ce qui concerne l'état 0, pour lequel l'état suivant est conditionné par l'état de ses voisins : de 0 on passe en 1 si un des voisins est adulte.

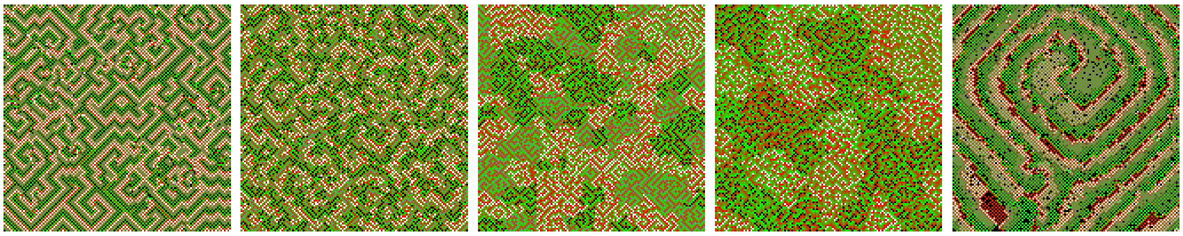
Des questions viennent immédiatement à l'esprit. La durée du cycle (le nombre n) influence-t-elle la dynamique ? Le comportement de l'automate est-il de période n ? On sait que c'est le cas dans le cas particulier d'Allouche et Reder [AR84], pour $v \geq 3$. Mais en ce qui nous concerne, la réponse est non : des simulations prouvent le contraire, mais elles sont rares. Nous n'avons pas été capables de caractériser les conditions sur a, v et n ainsi que les conditions initiales pour lesquelles la période est n, mais nous pouvons fournir des facteurs favorables à l'apparition de cette période (voir Chapitres 4 et 6). Pour effectuer une simulation, il nous faut une condition initiale. Faute d'une meilleure idée, nous avons pris ces dernières au hasard, en faisant toutefois varier la proportion de cellule dans l'état zéro. L'expérience montre que ce paramètre est significatif, de même que la taille du terrain, et là également, nous pouvons fournir des explications.

Le plan suivi est banal :

1. Pourquoi utiliser un modèle spatialisé ?
2. Formalisme : Le formalisme classique, plus quelques définitions qui nous seront utiles sont représentées. La classe d'automates étudiée est présentée.
3. Modèle de type *Milieu Excitable*. Nous exposons les résultats de Allouche et Reder.
4. Le modèle AVN. Nous exposons la méthode et les résultats relatifs à cet automate.

5. Vers un modèle non spatialisé. Nous nous interrogeons sur les conditions qui pourraient permettre de remplacer les modèles spatialisés par de simples modèles en classe de type Leslie densité-dépendants.
6. Modèle de type *Accélération*, inspiré par Wissel. Nous exposons la méthode et les résultats relatifs à cet automate.
7. Conclusion. Nous essayons de dégager quelques conclusions sur l'apport potentiel de la méthode des Automates Cellulaires.
8. En annexe, nous présentons quelques travaux, qui ne trouvent pas leur place dans le plan, mais qui ne sont pas sans rapport avec les sujets traités précédemment : Une étude statistique de la structure spatiale d'une espèce d'arbre (Eperua Falcata) dans son contexte naturel par la méthode de Ripley, et une tentative de briser l'aspect régulier des images obtenues par l'automate AVN.

Pour terminer cette introduction, voici une série de résultats de simulations que nous tenterons d'expliquer :



Chapitre 1

Modèle spatialisé, modèle non-spatialisé

Les modèles classiques de dynamique des population, de type matrice de Leslie ou système d'équations différentielles, font l'hypothèse d'un individu moyen, c'est-à-dire que tous les individus subissent le même environnement. Précisons sur l'exemple classique des cultures en continu en chémostat ([Ber95]).

$$\begin{cases} \dfrac{dS}{dt} = D(s_i - S) - V(S)X \\ \dfrac{dX}{dt} = YV(S)X - DX \end{cases} \tag{1.1}$$

- avec X, la concentration en phytoplancton (nombre de cellules par unité de volume),
- et S la concentration en substrat.

Le milieu est homogène, les micro-organismes et le substrat sont répartis uniformément dans l'espace, grâce au dispositif expérimental. Le terme $V(S)X$ est donc justifié. L'hypothèse qui sous-tend le modèle est acceptable pour la culture du phytoplancton en chémostat.

Cette hypothèse d'uniformité du milieu est acceptable, à des degrés divers, pour des organismes mobiles dont le comportement n'est pas trop grégaire (modèle de Lotka-Volterra pour des systèmes proie-prédateur). Mais elle devient inacceptable pour des végétaux; en effet, du fait que leur situation est fixe dans l'espace (ou presque !), les conditions d'ensoleillement par exemple, pour chaque arbre d'une forêt, peuvent être très différentes.

Pour illustrer notre propos, prenons l'exemple d'un automate cellulaire (cet objet sera défini plus précisément dans le chapitre suivant) dont les règles sont très simples :

- Les cellules sont dans les états 0 ou 1.
- Une cellule en 0 passe en 1 si une de ses voisines est en 1.

Cette règle entraîne simplement le remplissage de tout le domaine à partir des cellules en 1 dans la condition initiale. Mais si l'on regarde l'exemple illustré par la figure 1.1, on constate qu'on ne peut pas exprimer le nombre de cellules dans l'état 1 à l'instant $t+1$ en fonction de ce nombre à l'instant t, du fait que ce nombre dépend de la configuration spatiale des données. C'est ce que nous appellerons **"l'effet géométrique"**. Dans les deux

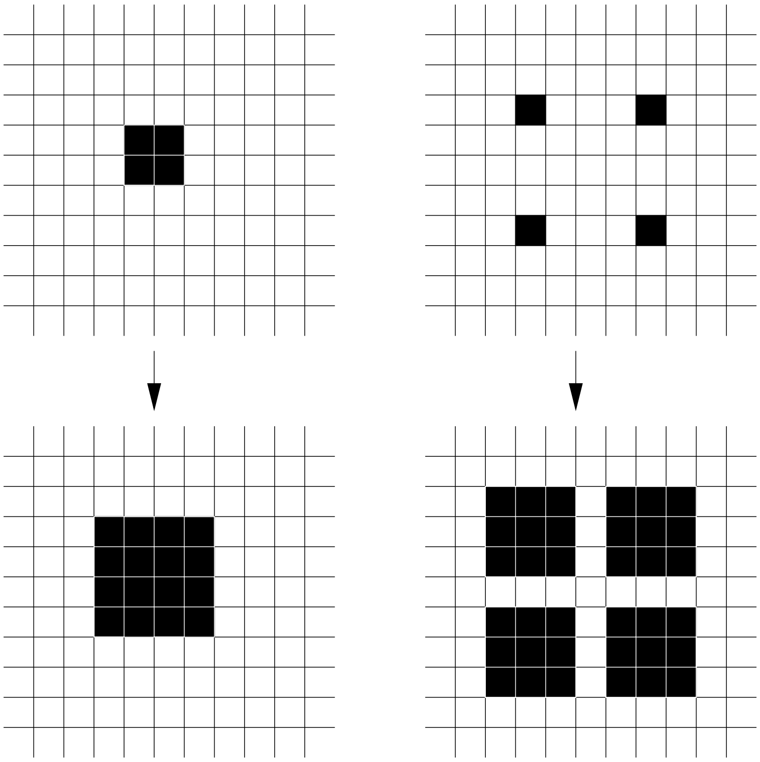


FIG. 1.1 – *Exemple simple recrutement spatialisé*

cas, le nombre initial de cellules dans l'état 1 est 4. A l'instant suivant, dans le premier cas, nous avons 16 cellules en 1; dans le cas suivant, on en trouve 36. Bien sûr, ce cas est un peu extrême dans la mesure où les cellules en 1 dans la condition initiale sont peu nombreuses, et on peut imaginer qu'avec un grand nombre de cellules en 1, disposées au hasard dans l'espace, on pourrait obtenir une *"fonction de recrutement"*. Nous aborderons d'ailleurs cette question dans le Chapitre 5. Ce petit exemple nous a montré que la modélisation d'un milieu non-homogène peut nécessiter une approche spatialisée. L'automate cellulaire étant le plus immédiat des systèmes mathématiques distribués spatialement, c'est sur lui que portera la suite de notre étude.

Chapitre 2

Automates Cellulaires: Formalisme et définitions

Dans la littérature ([Wol86]), les automates cellulaires sont en général présentés de manière informelle. Cela suffit à cerner l'objet en question, mais nous aurons besoin pour notre étude d'un environnement plus précis. Dans ce chapitre, nous allons donc nous donner les outils formels nous permettant de manipuler les automates cellulaires, ainsi que quelques propriétés générales qui en découlent, puis nous nous concentrerons sur le type d'automates qui nous intéresse plus particulièrement. Mais pour commencer, il nous faut définir ce qu'est un automate cellulaire.

2.1 L'Automate

2.1.1 Le Domaine

Un automate cellulaire $\mathcal{A}$ est tout d'abord composé de cellules, polygones disposés sur le plan. Ces cellules peuvent être de toutes formes et tailles, mais doivent constituer un pavage du plan, ou d'une partie du plan : en effet, un automate peut être constitué d'un nombre fini ou indéfini de cellules. En ce qui nous concerne, nous travaillerons avec des cellules carrées, de tailles identiques.

Définition 2.1.1. *Nous appellerons* ***domaine*** *et nous noterons D, l'ensemble des cellules de l'automate.*

On définit aussi un ensemble d'indices I, tel qu'il existe une bijection de D dans I, qui à chaque cellule fait correspondre son indice.

$$D = \{c_i\}_{i \in I} \tag{2.1}$$

Voici quelque exemples de domaines possibles :

– Domaine à une dimension, de taille indéfinie, indicé par ℕ.

...	c_{-3}	c_{-2}	c_{-1}	c_0	c_1	c_2	c_3	...

– Domaine à deux dimensions, de taille indéfinie, indicé par $\mathbb{N}^2$.

	$\vdots$	$\vdots$	$\vdots$	
$\dots$	$c_{-1,1}$	$c_{0,1}$	$c_{1,1}$	$\dots$
$\dots$	$c_{-1,0}$	$c_{0,0}$	$c_{1,0}$	$\dots$
$\dots$	$c_{-1,-1}$	$c_{0,-1}$	$c_{1,-1}$	$\dots$
	$\vdots$	$\vdots$	$\vdots$	

– Domaine à deux dimensions, de taille fini, indexé par $I = \{0..128\} \times \{0..128\}$.

$c_{0,128}$	$\dots$	$c_{128,128}$
$\vdots$		$\vdots$
$c_{0,0}$	$\dots$	$c_{128,0}$

2.1.2 Le Voisinage

Une fonction de voisinage est définie, liant chaque cellule à ses "voisines".

Définition 2.1.2. *Nous appellerons $\mathcal{V}$ la fonction de voisinage définie comme suit :*

$$\begin{array}{rrcl} \mathcal{V}: & D & \longrightarrow & P(D) \text{ (l'ensemble des parties de } D\text{)} \\ & c & \longmapsto & \mathcal{V}(c) \end{array}$$

où $\mathcal{V}(c)$ est l'ensemble des cellules voisines de la cellule c. Les cellules du voisinage sont numérotées $0,\ 1,\ 2\ \dots\ v$.

Cette définition est très générale, et permet des fonctions de voisinages très variées. Mais les fonctions de voisinage généralement utilisées sont les suivantes :

– Le voisinage de Von Neumann : $V_N(c_{(i,j)}) = \{c_{(i-1,j)}, c_{(i+1,j)}, c_{(i,j-1)}, c_{(i,j+1)}\}$
– Le voisinage de Moore : $V_M(c_{(i,j)}) = V_N(c_{(i,j)}) \cup \{c_{(i-1,j-1)}, c_{(i-1,j+1)}, c_{(i+1,j-1)}, c_{(i+1,j+1)}\}$

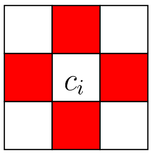


FIG. 2.1 – *Le voisinage de Von Neumann*

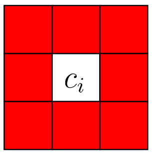


FIG. 2.2 – *Le voisinage de Moore*

Dans tout ce travail, nous considérerons des voisinages de Von Neumann.

2.1.3 L'ensemble des états

On se donne un ensemble (généralement fini) noté E. On associe à chaque cellule un état appartenant à E. Les automates cellulaires les plus étudiés, comme par exemple le jeu de la vie, de Conway, (voir paragraphe 2.1.5) ne disposent que de deux états : $E = \{0,1\}$. Dans la suite de notre étude, nous travaillerons plus généralement avec $E = \mathbb{Z}/n\mathbb{Z} = \{0 \ldots n-1\}$, c'est-à-dire n états. Nous pouvons maintenant définir la notion d'état appliquée à tout le domaine : c'est l' application $U(.)$, qui à chaque élément de D associe un élément de E :

$$\begin{array}{rccc} U(.): & D & \longrightarrow & E \\ & c & \longmapsto & U(c) \end{array}$$

2.1.4 La fonction de transition

Pour que notre automate soit complètement défini, il lui faut son aspect dynamique : la fonction de transition. Soit v, le nombre de voisins déterminé par la fonction de voisinage, la fonction de transition Tr est définie comme suit :

$$\begin{array}{rccc} Tr: & E \times E^v & \longrightarrow & E \\ & e_0,e_1,\ldots,e_v & \longmapsto & e'_0 \end{array} \tag{2.2}$$

La fonction de transition prend comme arguments l'état d'une cellule et celui de ses voisines, et retourne un élément de E. Par extension, on peut définir la fonction de transition sur tout le domaine de l'automate : Soit N le nombre total de cellules du domaine.

$$\begin{array}{rccl} Tr: & E^N & \longrightarrow & E^N \\ & U(.) & \longmapsto & U'(.) \text{ tel que} \\ & & & \forall c \in D, U'(c) = Tr(U(c),(U(d))_{d \in V(c)}) \end{array} \tag{2.3}$$

Reste le problème des conditions aux bords. En effet, si le domaine utilisé est borné, une cellule appartenant au bord du domaine ne disposera pas forcément de toutes ses voisines. Afin de palier ce problème, nous devons définir des conditions aux bords. Cela consistera à border le domaine de cellules ayant une dynamique propre et qui serviront de voisines aux cellules du bord du domaine. Ainsi, la notion de voisinage, et donc de fonction de transition, aura un sens pour toutes les cellules du domaine. Par exemple, et c'est généralement la solution que nous adopterons par la suite, nous pouvons border le domaine étudié de cellules dont l'état est fixé à zéro (figure 2.3).

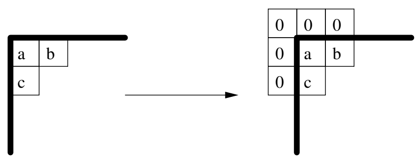


FIG. 2.3 – *Les conditions aux bords*

L'objet de notre étude étant maintenant complètement défini, nous pouvons constater que le champs d'investigation est très large. En effet, pour donner un ordre de grandeur, si l'on étudie par exemple un automate à 16 états, sur un domaine constitué de 100×100 cellules, on atteint un nombre d'états possibles de l'automate de 16^{10^4}, ce qui est énorme !

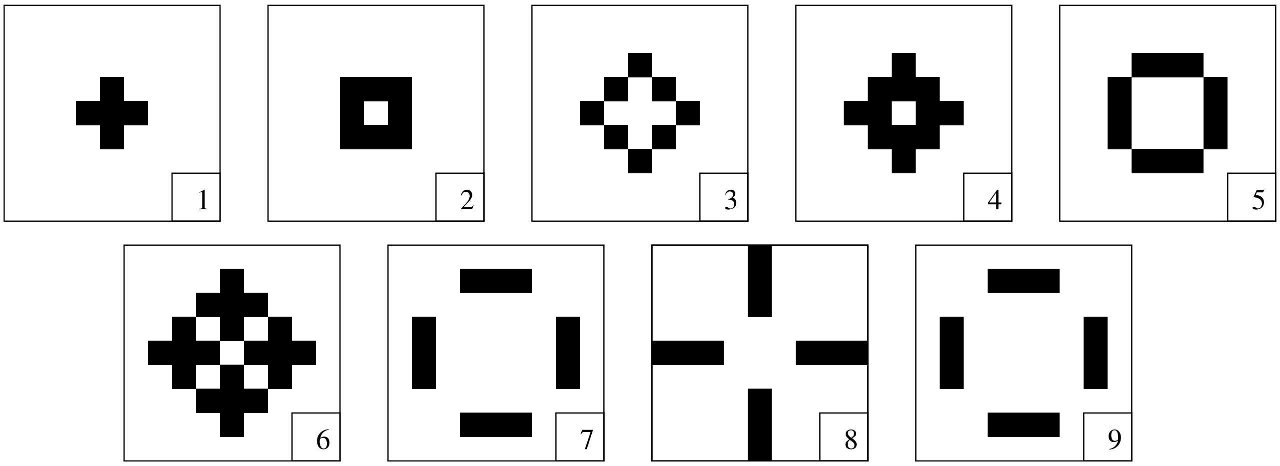


FIG. 2.4 – *Un exemple pour le Jeu de la Vie.*

Par comparaison, l'univers serait vieux de 16 milliards d'années, soit 10^{17} secondes ! Une étude exhaustive de tous les états du domaine est impensable, il faut donc développer des outils théoriques adaptés à l'étude des automates cellulaires.

2.1.5 Un exemple : Le jeu de la vie

Sous le nom de Jeu de la Vie (*Game of Life*), on désigne un jeu inventé par le mathématicien *John Horton Conway*, au début des années 70. C'est un automate cellulaire, et présente donc les caractéristiques exposées précédemment :

- Le domaine est à deux dimensions, en théorie infini; en effet, si l'on veut simuler une trajectoire sur un ordinateur, il faut bien se contenter d'un domaine fini, aussi grand soit-il.
- Le voisinage, c'est-à-dire l'ensemble des cellules qui vont intervenir pour décider de l'avenir d'une cellule, sera celui de Von Neumann (8 voisines).
- C'est un automate booléen. L'ensemble des états est donc $E = \{0,1\}$. Une cellule en l'état 1 est dite *vivante*. Une cellule en l'état 0 est dite *morte* ou *vide*.
- La règle de transition est la suivante :
  - Une cellule vivante ne survivra que si elle a 2 ou 3 voisines vivantes exactement : elle ne doit être ni isolée, ni étouffée.
  - Une cellule morte devient vivante si et seulement si elle dispose de trois voisines vivantes exactement (car il faut de la vie pour engendrer la vie).

En résumé : **2 ou 3 voisines pour survivre, 3 pour naître.**

La figure 2.4 représente un exemple de trajectoire issue de l'automate cellulaire du Jeu de la Vie : chaque carré noir représente une cellule vivante.

On peut constater que les étape 7 et 9 sont identiques : nous somme donc ici en présence d'une trajectoire ultimement périodique de période 2 (ces notions seront développées plus précisément dans la section 2.2).

Cet automate n'a pas pour but la modélisation d'un phénomène quelconque, mais il nous montre que des règles très simples peuvent donner lieu à une étonnante diversité de comportements. En effet, en utilisant l'un des nombreux logiciels existant pour la simulation du Jeu de la Vie, on s'aperçoit rapidement que son comportement est très riche, et difficilement prévisible. Depuis sa création, cet automate n'a cessé d'être étudié, simulé, et voit encore l'apparition de nouveaux résultats : existence de structures stables, périodiques, qui se déplacent, qui recouvrent le plan, etc... Ces résultats peuvent être de deux types différents. **Les résultats théoriques** montrent l'existence théorique d'un type de structure donné. **Les résultats pratiques**, c'est à dire les structures correspondantes, peuvent être beaucoup plus longs à trouver. Certaines structures sont encore inconnues, alors que leur existence est prouvée. En fait, le nombre de configurations possible est tellement grand, qu'il est impossible de chercher au hasard. Les résultats théoriques servent donc à orienter les recherches. Mais il reste encore de nombreuse questions ouvertes au sujet du *Jeu de la Vie* ([Del96]).

2.2 Les Trajectoires

Une fois qu'un automate cellulaire est complètement défini, tout son intérêt réside en l'observation et l'étude des suites d'états du domaine, calculées à partir de différentes conditions initiales. Nous appellerons ces suites des **trajectoires de l'automate cellulaire**.

2.2.1 Définitions

Une trajectoire d'un automate est une application qui à un temps t fait correspondre un état du domaine, deux états successifs étant liés par la fonction de transition.

Définition 2.2.1. *Soit $\mathcal{A}$ un automate cellulaire ; Soit $U_o(.)$ un état du domaine, nous appellerons $\mathcal{A}(.,t,U_0,t_0)$* ***la trajectoire de l'automate $\mathcal{A}$ issue de U_0 à partir de t_0.***

$$\begin{array}{rccl} \mathcal{A}(.,t,U_0,t_0): & \mathbb{N} & \longrightarrow & (D \to E) \\ & t & \longmapsto & \mathcal{A}(.,t,U_0,t_0) \textit{ tel que :} \\ & & & \left\{ \begin{array}{ll} \mathcal{A}(.,t+1,U_0,t_0) & = Tr(\mathcal{A}(.,t,U_0,t_0)) \\ \mathcal{A}(.,t_0,U_0,t_0) & = U_0(.) \end{array} \right. \end{array}$$

L'état $U_0(.)$ sera appelé ***état initial*** *de la trajectoire.*

Notation 2.2.2. *On pourra noter $\mathcal{A}(.,t,U_0)$ la trajectoire de l'automate $\mathcal{A}$ issue de U_0 à partir de 0.*

$$\mathcal{A}(.,t,U_0) = \mathcal{A}(.,t,U_0,0)$$

Notation 2.2.3. *De même, on notera $\mathcal{A}(.,t)$, une trajectoire de l'automate lorsqu'on ne tient pas à désigner de condition initiale.*

2.2.2 Périodicité

Définissons maintenant ce qui sera une caractéristique importante des trajectoires d'automates cellulaires : la notion de périodicité.

Définition 2.2.4. *Soit c une cellule. La trajectoire $\mathcal{A}(c,t)$ est dite* ***ultimement périodique de période T*** *si et seulement si :*

$$\exists t_0 \in \mathbb{N}, \exists T \in \mathbb{N}^* \text{ tel que } \forall t \geq t_0, \mathcal{A}(c,t+T) = \mathcal{A}(c,t) \tag{2.4}$$

Soit P l'ensemble des T non nuls vérifiant la condition (2.5). On appellera ***période de la trajectoire*** *l'entier T tel que :*

$$T = min(P)$$

Définition 2.2.5. *Une trajectoire $\mathcal{A}(.,t)$ est dite* ***ultimement périodique de période T*** *si et seulement si :*

$$\exists t_0 \in \mathbb{N}, \exists T \in \mathbb{N}^* \text{ tel que } \forall t \geq t_0, \mathcal{A}(.,t+T) = \mathcal{A}(,t) \tag{2.5}$$

Soit P l'ensemble des T non nuls vérifiant la condition (2.5). On appellera ***période de la trajectoire*** *l'entier T tel que :*

$$T = min(P)$$

Nous pouvons présenter ici une propriété évidente des automates cellulaires.

Remarque 2.2.6. *Soit un automate cellulaire $\mathcal{A}$ dont le domaine est constitué d'un nombre fini de cellules. Pour tout état de l'automate $U_o(.)$, il existe t_1 tel que la trajectoire $\mathcal{A}(.,t,U_0)$ soit périodique à partir de t_1.*

Preuve. Le nombre des états possibles pour chaque cellule, ainsi que le nombre de cellules étant fini, le nombre d'états du domaine entier est lui aussi fini. On retombe donc forcement sur un état déjà rencontré; c'est-à-dire que pour une trajectoire donnée, il existe un entier t_1 tel que tous les états de l'automate avant t_1 sont différents, et de plus, il existe un entier non nul T tel que :

$$\mathcal{A}(.,t_1+T,U_0) = \mathcal{A}(.,t_1,U_0)$$

La fonction de transition étant déterministe, on sait donc que :

$$\forall t \geq t_1, \mathcal{A}(.,t+T,U_0) = \mathcal{A}(.,t,U_0)$$

Notre trajectoire est donc périodique (et la longueur de son transitoire est t_1). □

Notons que ce théorème n'est pas très informatif : en effet, même si nous savons que la trajectoire d'un automate cellulaire sur un domaine borné finit par être périodique, le transitoire pourrait être tellement grand qu'un ordinateur pourrait calculer cette trajectoire pendant plusieurs années sans en trouver la période !

2.2.3 Restriction d'une trajectoire

Soit $\mathcal{A}$ un automate cellulaire défini sur un domaine D, $U_0(.)$ une condition initiale de cet automate et D' un sous-ensemble de D. La restriction de la trajectoire de $\mathcal{A}$ issue de $U_0(.)$ à D' est l'application suivante :

$$\begin{array}{rrcl} \mathcal{A}(.,t,U_0)_{|D'}: & \mathbb{N} & \longrightarrow & (D' \to E) \\ & t & \longmapsto & \mathcal{A}(.,t,U_0)_{|D'} \text{ tel que :} \\ & & & \forall c \in D', \mathcal{A}(c,t,U_0)_{|D'} = \mathcal{A}(c,t,U_0) \end{array}$$

En fait, une trajectoire restreinte est la même trajectoire, mais seulement vue sur les cellules de D' (figure 2.6).

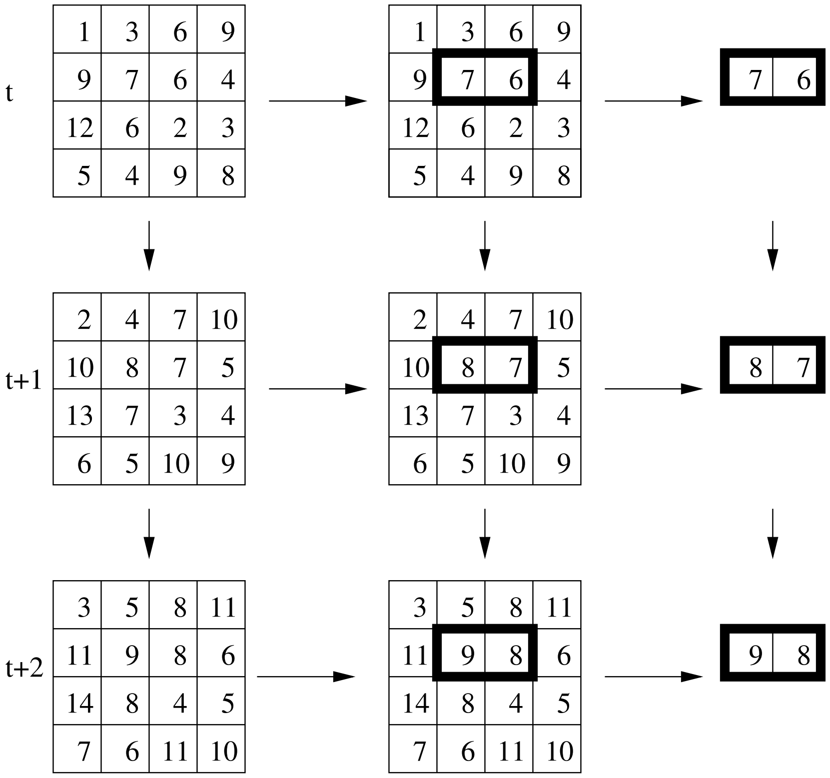
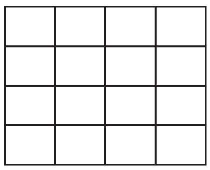
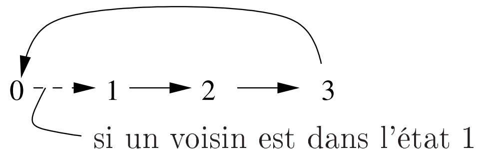


FIG. 2.6 – *Restriction d'une trajectoire*

Il importe de ne pas confondre **la restriction d'une trajectoire à un sous domaine**, et **la restriction de l'automate à un sous domaine**. Prenons un domaine

$D =$ 

et l'automate suivant :



La condition aux bords est 0. Prenons un sous-domaine D', et considérons le même automate sur D', avec la même condition aux bords, et partons d'une même condition initiale. Ici, l'état des cellules à l'intérieur de D' ne dépend pas de l'état des cellules extérieures à D' (figure 2.7). Afin d'exprimer la restriction de l'automate à un sous domaine

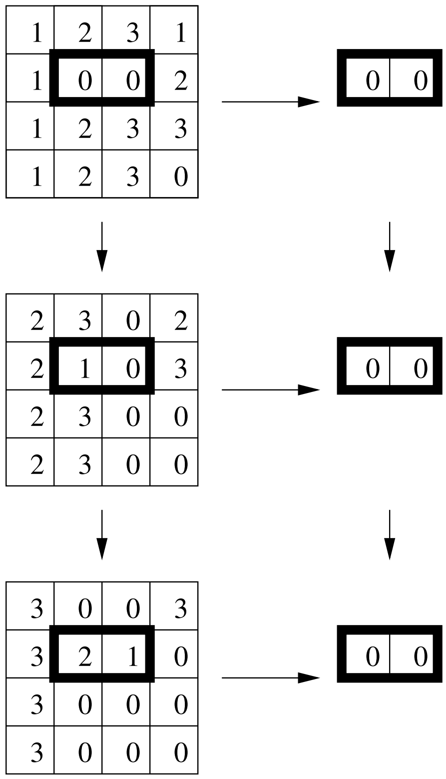


FIG. 2.7 – *Illustration de la notion de sous-domaine*

D', on utilise la notation suivante :

Notation 2.2.7. *On note $A_{|D'}$, l'automate restreint à D' avec la même condition aux bords.*

2.3 Les cycles

Nous allons ici définir une notion qui s'avèrera très importante dans la compréhension des différents automates que nous étudierons : c'est la notion de cycle. Définissons tout d'abord la notion de graphe de connexion (figure 2.8).

Définition 2.3.1. *Le graphe de connexion d'un ensemble de cellules est défini de la façon suivante:*

- *Ses sommets sont les centres des cellules de l'ensemble.*
- *Ses arêtes sont les segments de droite joignant le centre de chaque cellule au centre de* ***chaque cellule voisine*** *appartenant à l'ensemble.*

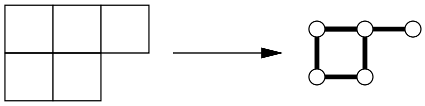


FIG. 2.8 – *graphe de connexion d'un ensemble de cellules*

A chaque sommet du graphe correspond donc une et une seule cellule, et à tout ensemble de cellules, un et un seul graphe de connexion. Nous pouvons maintenant définir ce que nous appellerons par la suite un cycle de cellules (figure 2.9) :

Définition 2.3.2. *Un* ***cycle*** *est un chemin* ***fermé sans point double*** *sur le graphe de connexion.*

Par abus de langage, nous dirons *cycle de cellules*, mais il faut bien noter que deux cycles distincts du graphe de connexion peuvent définir le même ensemble de cellules (voir figure 2.9).

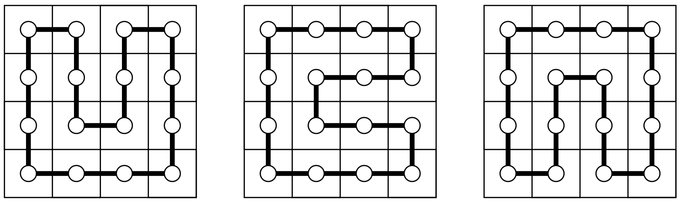


FIG. 2.9 – *trois cycles différents pour un même ensemble de cellules*

Maintenant que nous avons posé les principales notions qui nous serviront à l'étude des automates cellulaires, nous allons quelque peu réduire le champs d'investigation. En effet, une fonction de transition consiste à attribuer à chaque couple *"état d'une cellule"* et *"état du voisinage"*, un élément de l'ensemble E des états. Si l'on appelle n le nombre d'éléments de E, et v le nombre de cellules appartenant au voisinage (4 pour Von Neumann, 8 pour Moore, voir page 12), nous obtenons n^v états du voisinage possibles, soit n^{v+1} couples *(état d'une cellule, état du voisinage)*, avec pour chacun n choix pour le prochain état possibles. Cela nous donne $\mathbf{n^{n^{v+1}}}$ **fonctions de transitions possibles**. Si l'on étudie des automates booléens (n est égal à deux) et à une dimension, avec le voisinage le plus proche (soit $v = 2$), on obtient $2^{2^{2+1}} = 256$ fonctions de transitions. On peut alors en faire une étude exhaustive ([Wol86]). En ce qui nous concerne, les automates cellulaires étudiés sont bi-dimensionnels, avec quatre voisins, et un nombre d'états généralement de seize, ce qui nous donne $\mathbf{16^{16^5}}$ **règles possibles**! L'approche exhaustive n'est donc pas envisageable. Nous allons donc créer une classe d'automates cellulaires traduisant les bases du "mosaic cycle concept", qui contiendra les règles de transition que nous étudierons par la suite.

2.4 Automate Cellulaire de type "Mosaic-Cycle"

Nous allons donc ici définir la toile de fond qui servira par la suite de base aux différents automates que nous allons étudier. Cette classe d'automates cellulaires est inspirée du *"Mosaic Cycle Concept"* décrit dans [Rem91].

- Le domaine sera bi-dimensionnel, borné (par exemple un carré de 100×100 cellules) ou pas (tout le plan).
- L'ensemble des états sera l'ensemble $\{0 \ldots n-1\}$, constitué donc de n états.

L'idée de base de cette classe d'automates cellulaires consiste en une évolution cyclique dans l'ensemble des états : à chaque itération, on passe de l'état e à l'état $e+1 \mod n$.

$$\forall c \in D, \mathcal{A}(c,t+1) = \mathcal{A}(c,t) + 1 \mod n$$

Nous adopterons pour la suite deux types de représentation de ce cycle de vie (figures 2.10 et 2.11).

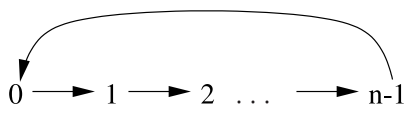


FIG. 2.10 – *Règle cyclique de base*

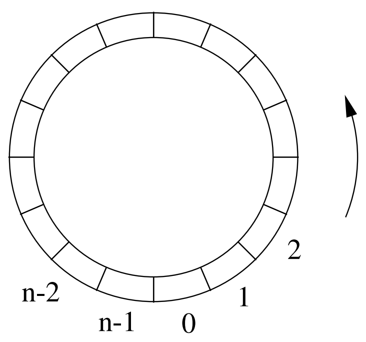


FIG. 2.11 – *Le cercle des états*

En ce qui concerne l'interprétation écologique de cette classe d'automates cellulaires, nous pouvons proposer par exemple :

- Chaque cellule représente un individu (par exemple un arbre), ou une parcelle de l'écosystème considéré (par exemple une forêt), et son état représente son état de maturation dans le cycle, par exemple l'âge ou le diamètre pour un arbre.
- L'état 0 aura un statut particulier puisqu'il sera interprété comme un état neutre. On pourra dire par exemple qu'une cellule en l'état 0 est "vide".

Jusqu'ici, la dynamique spatiale de notre modèle n'est pas passionnante, puisqu'elle consiste pour chaque cellule, en la succession cyclique des états. Il nous faut y ajouter l'aspect "interaction entre voisins", qui vient en quelque sorte "perturber" le déroulement régulier du cycle de vie d'une cellule, en conditionnant le passage d'un état au suivant par une situation particulière dans le voisinage. Nous pouvons imaginer toutes sortes d'interactions de voisinage, faisant intervenir

– La somme des états des cellules voisines,
– La moyenne,
– Le plus grand, le plus petit de ces états,
– Le nombre de cellules voisines dans un certains ensemble d'états,
– Ou même des interactions anisotropes, favorisant une direction.

Mais pour rester le plus simple possible, nous ne traiterons que des interactions de voisinage concernant un seul point du cycle, et en l'occurrence l'état neutre (état 0), du type

---

`si` présence dans le voisinage d'un état appartenant à un
 sous-ensemble prédéfini de E
`alors` ... ;
`sinon` ... ;

Les règles de transition seront représentées de la façon suivante :

– Une flèche blanche (—▷) indique la transition qui s'effectue en présence d'un des états encadrés dans le voisinage de la cellule considérée.
– L'autre flèche partant de l'état neutre (- -►) indique ce qu'il se passe dans le cas contraire.

Mais illustrons plutôt notre propos en présentant les deux types de fonctions de transition que nous utiliserons :

– Le type AVN.
– Le type Accélération.

2.4.1 Le type AVN

Un automate particulier de ce type est étudié par J.P. Allouche et C. Reder dans le contexte de la théorie des milieux excitables ([AR84] et Chapitre 3 du présent document). Mais nous verrons que leurs résultats ne sont pas immédiatement généralisables à tous les automates de ce type (Chapitre 4). Ici, une cellule en l'état neutre, ne passe en l'état un que si l'une de ces voisines est dans un état appartenant à l'ensemble des états matures ou adultes (i.e. susceptibles de produire des graines); on dira alors que cette cellule a été "fécondée". Il nous reste à déterminer quels sont les états adultes, et nous verrons que ce

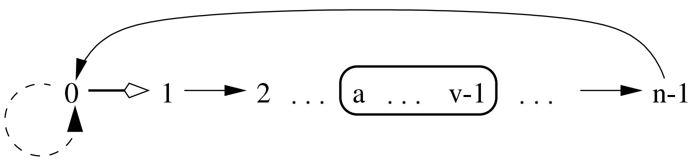


FIG. 2.12 – *Fonction de transition de type AVN*

n'est pas sans importance. Cette fonction de transition sera donc définie par le schéma de la figure 2.12, les états encadrés représentant l'ensemble des états excitateurs.

On peut maintenant se poser les questions suivantes, que nous serons appelés à développer dans les chapitres suivants :

- Cette règle de transition donnera-t-elle lieu à des trajectoires "pérennes" ? nous appellerons une trajectoire pérenne une trajectoire qui ne contient pas l'état identiquement nul, c'est-à-dire l'état de l'automate où toutes les cellules du domaine sont dans l'état neutre.
  La question est donc celle de la survie d'un écosystème ainsi modélisé.
- Quelles périodes seront atteintes par l'automate (n ou autres ?)
- Quelle sera la taille des transitoires ?
- A quel type de structurations spatiales donnera lieu l'automate cellulaire ainsi défini ?

2.4.2 Le type Accélération

Cet automate est inspiré du modèle que Ch Wissel étudie dans [Wis91]. Il représente en fait un seul des mécanismes invoqués dans ce modèle. Une cellule passe automatiquement de zéro en un, sauf si elle possède une voisine dans un état "entraînant", ce qui lui permettra de sauter un certain nombre d'états du cycle de vie (figure 2.13).

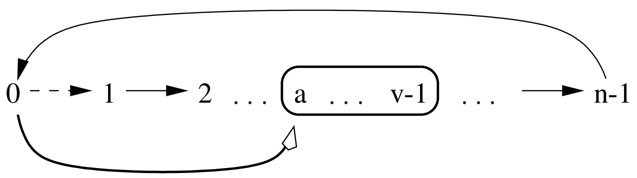


FIG. 2.13 – *Fonction de transition de type Accélération*

Ici, la question de la pérennité des trajectoires produites par un tel automates cellulaire ne se pose pas. En effet, une cellule ne reste jamais en l'état neutre, quoi-qu'il arrive. Mais les autres questions restent posées, soit :

- Les période atteintes par l'automate.
- La taille des transitoires.
- Les structurations spatiales produites.

Nous développerons ces aspects dans le chapitre 6. Nous avons donc créé une classe d'automates cellulaires adaptée à la modélisation du “mosaic cycle concept”. Nous allons maintenant étudier plus en détails les deux types d'automate présentés sommairement ici. Le chapitre suivant traite d'un cas particulier de l'automate de type AVN.

Chapitre 3

Modèle de type "milieu excitable"

3.1 Présentation

Le premier exemple que nous allons voir a été étudié entre autre J.M. Greenberg, C. Green et P. Hastings ([GGH80]) et J.P. Allouche et C. Reder qui écrivent à son sujet ([AR84]) :

> *L'automate cellulaire que nous étudions ici s'intègre dans la classe des mécanismes de "diffusion-réaction en milieu excitable" : c'est une classe très vaste qui regroupe tant des phénomènes concrets tels la propagation de l'influx nerveux ou certaines réactions chimiques (la plus célèbre étant celle de Belouzov-Zhabotiskii-Zaikin) [...]*

Bien que ces références ne soient pas d'ordre écologique, cet automate s'inscrit, de part sa définition, dans la classe des automates de type "Mosaic Cycle", et nous verrons dans le chapitre 4 qu'une petite généralisation donnera lieu à un type d'automate interprétable en termes de cycles écologiques. Nous allons tout d'abord le définir puis exposer les principales propriétés mises en évidence par J.P. Allouche et C. Reder. Les résultats qui vont suivre ne sont donc pas originaux, mais nous nous sommes permis quelques modifications de vocabulaire et de présentation afin de préserver la cohérence globale.

3.2 Règles de l'automate

- Le domaine sera à deux dimensions.
- Le voisinage sera celui de Von-Neumann (4 voisins).
- Avant de définir la fonction de transition, il nous faut distinguer trois sous-ensembles de l'ensemble des états E.
  - 0, l'état neutre.
  - $X = \{1..v-1\}$, l'ensemble des **états excitateurs**.
  - $R = \{v..n-1\}$, l'ensemble des **états réfractaires**.

– La fonction de transition sera définie comme suit :

`si` $\mathcal{A}(c,t) \neq 0$
`alors` $\mathcal{A}(c,t+1) = \mathcal{A}(c,t) + 1 \mod n$
`sinon`
  `si` $\exists\ d \in \mathcal{V}(c)$ telle que $\mathcal{A}(d,t) \in X$
  `alors` $\mathcal{A}(c,t+1) = 1$
  `sinon` $\mathcal{A}(c,t+1) = \mathcal{A}(c,t) = 0$

Chaque cellule passe à chaque itération à l'état suivant (modulo n), seul le passage de 0 en 1 étant conditionnel à la présence d'une cellule dans un état excitateur dans le voisinage.

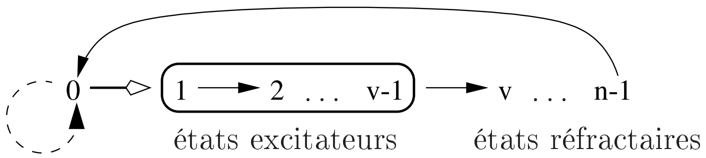


FIG. 3.1 – *L'automate cellulaire étudié par Allouche et Reder*

3.3 Définitions

Nous présentons ici les différents outils et notions de base permettant d'obtenir les résultats présentés par la suite.

Définition 3.3.1. *Une trajectoire de l'automate est dite pérenne si pour tout instant t il existe au moins une cellule du domaine telle que son état ne soit pas nul pour un certain instant ultérieur à t.*

En fait, la règle de transition de cet automate entraîne qu'une trajectoire est pérenne[1] si et seulement si elle ne contient jamais l'état identiquement nul.

Définition 3.3.2. *Étant donné un état de l'automate, on appelle support de cet état l'ensemble des cellules du domaine qui ne sont pas dans l'état neutre.*

Par la suite, ne seront étudiées que les trajectoire de l'automate issues de conditions initiales dont le support est borné. En effet, sans cette hypothèse, ou d'autres hypothèses aussi fortes sur la condition initiale, on ne peut espérer aucune régularité du comportement spatio-temporel des trajectoires de l'automate. Par exemple, et contrairement à ce que nous

1. "pérenne" se dit "vivante" dans [AR84].

verrons par la suite, on peut construire un état initial à support de taille infini tel qu'aucune restriction de la trajectoire ainsi obtenue ne soit périodique à partir d'un certain rang.

Exemple 1. *Voici un exemple sur un domaine à une dimension. On peut facilement le transposer dans un domaine à deux dimensions.*

$t=0:$	0	$1\ldots n-1$	0	0	$1\ldots n-1$	0	0	0	$1\ldots n-1$	0	0	0	0	*etc...*

$t=1:$	$1\ldots n-1$	0	0	$1\ldots n-1$	0	0	0	$1\ldots n-1$	0	0	0	0	1	*etc...*

Le support de la condition initiale n'est pas borné, et on peut aisément se rendre compte qu'aucune restriction d'une telle trajectoire n'est périodique.

Définissons maintenant la notion de distance entre deux états :

Définition 3.3.3. *Si a et b sont deux éléments de $E = 0\ldots n-1$, on définit la distance de a à b, que l'on note $d(a,b)$, par :*

$$d(a,b) = min(|b-a|, n-|b-a|)$$

C'est en fait la distance qui sépare deux éléments de E sur le cercle des états (figure 3.2).

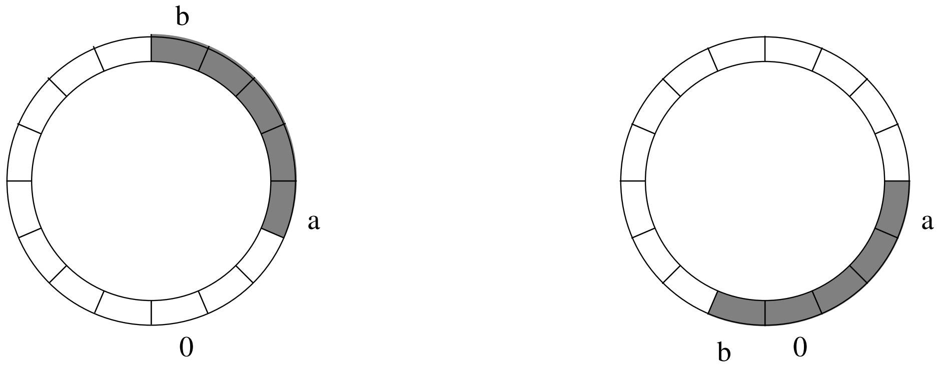


FIG. 3.2 – *Distance entre deux états sur le cercle des états.*

Nous pouvons maintenant définir la notion de squelette d'un état du domaine.

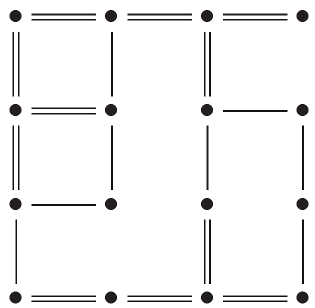
3.4 Squelette d'un état du domaine

Définition 3.4.1. *Soit $U(.)$ un état du domaine. Le squelette de U (figure 3.3) est le graphe partiel du graphe de connexion, bicolorié par les arêtes*[2]*, dont les sommets sont les sommets du graphe de connexion et dont les arêtes sont définies comme suit : si c et d sont deux cellules voisines,*

– *une arête double* $c =\!= d$ *les relie si* $d(U(c),U(d)) \leq 1$

0	0	0	0
5	4	1	3
0	2	5	2
4	5	0	0

État U d'un carré



Squelette de cet état

FIG. 3.3 – *Exemple de squelette d'un état : $n = 6$, $v = 3$ ou 5*

– *une arête simple* $\quad c$——d *les relie si* $\quad 2 \leq d(U(c),U(d)) \leq min(v-1,n-v+1)$

La proposition suivante, immédiate à vérifier, donne une première justification de l'introduction de la notion de squelette.

Proposition 3.4.2. *Soit $\mathcal{A}(.,t)$ une trajectoire de l'automate. Alors, les squelettes des états $\mathcal{A}(.,t)$ forment une suite croissante et plus précisément, si c et d sont deux cellules voisines,*

– *si elles sont reliées par une arête double dans le squelette de $\mathcal{A}(.,t)$, elles le sont aussi dans le squelette de $\mathcal{A}(.,t+1)$*
– *si elles sont reliées par une arête simple dans le squelette de $\mathcal{A}(.,t)$, elles sont reliées par une arête simple ou double dans le squelette de $\mathcal{A}(.,t+1)$ et dans ce cas, on a de plus l'inégalité suivante :*

$$d(\mathcal{A}(c,t+1),\mathcal{A}(d,t+1)) \leq d(\mathcal{A}(c,t),\mathcal{A}(d,t))$$

Le corollaire suivant est évident :

Corollaire 3.4.3. *Soit $\mathcal{A}(.,t)$ une trajectoire de l'automate à support initial borné. Alors, la suite des squelettes des états $\mathcal{A}(.,t)$ est constante à partir d'un certain rang. A une telle trajectoire de l'automate, on peut ainsi associer son squelette limite.*

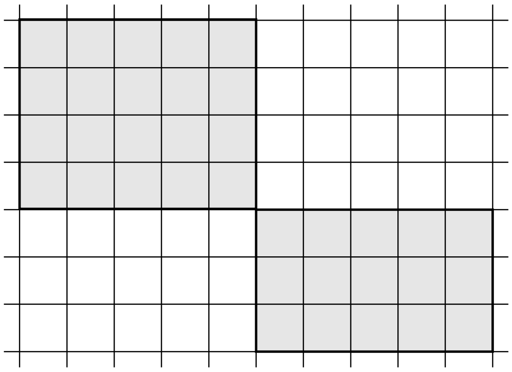
La proposition 3.4.2 va être utilisée pour réduire le domaine d'étude de l'automate à un sous-ensemble du domaine. Cette propriété est essentielle pour la modélisation de l'automate, la mémoire d'un ordinateur (ainsi que son écran) étant bornée. Mais elle est également indispensable dans l'étude de la structuration spatio-temporelle de l'automate.

3.5 Réduction du domaine à un borné

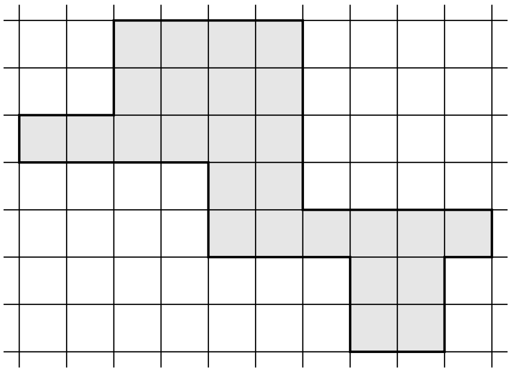
Quelles conditions doit-on imposer à un ensemble de cellules pour que son état à tout instant ne dépende pas de l'état des cellules qui lui sont extérieures.

Définition 3.5.1. *Nous dirons qu'un sous-ensemble C d'un domaine est connexe à coupes*

2. "bicolorié" signifie que les arêtes peuvent être de deux types : simples ou doubles.



C n'est pas connexe



C est un c.c.c.

FIG. 3.4 – *Exemple : C est l'ensemble des cellules hachurées*

convexes, et en abrégé "C est un c.c.c." (figure 3.4) si :

- *C est un ensemble connexe de cellules du domaine.*
- *l'intersection de C avec toute droite horizontale ou verticale est convexe.*

La proposition suivante précise ce que l'on entend par "réduction du domaine d'étude de l'automate".

Proposition 3.5.2. *Soit $\mathcal{A}(.,t)$ une trajectoire de l'automate. Si C est un c.c.c. contenant les cellules du support de l'état initial ainsi que leurs voisines au sens de Moore (8 voisines : 2.2), la trajectoire restreinte à C, $\mathcal{A}(.,t)|_C$ est égale à la trajectoire de l'automate restreint à C, $\mathcal{A}|_C(.,t)$. Autrement dit, si c est une cellule de C dans l'état neutre à l'instant t, son état à l'instant $t+1$ sera égal à 1 si et seulement si c à une voisine dans C dans un état excitateur à l'instant t.*

La démonstration de cette proposition utilise fortement l'existence du squelette limite (proposition 3.4.2 : deux cellules voisines extérieures au support de la condition initiale sont reliées par une arête double dans tous les squelettes). Si $\mathcal{A}(.,t)$ est une trajectoire de l'automate dont le support de la condition initiale est borné, et B un sous-ensemble fini de cellules du plan, alors, à partir d'un certain rang, la suite des états de B est périodique; en effet, il suffit de montrer ce résultat pour un c.c.c. C borné contenant B et vérifiant les hypothèses de la proposition 3.5.2; or le nombre d'états de C est fini, c'est immédiat grâce à la même proposition. On voit tout de suite qu'un problème se pose : existe-t-il une période limite commune aux restrictions d'une trajectoire à tous les bornés du domaine, et plus généralement, peut-on déduire de la connaissance de la restriction d'une trajectoire à un borné, l'état d'une cellule quelconque du domaine à tout instant ? La proposition suivante et son corollaire donnent une réponse à ces questions.

Proposition 3.5.3. *Soit $\mathcal{A}(.,t)$ une trajectoire issue d'une condition initiale dont le support est borné, et R un rectangle contenant les cellules du support de la condition initiale ainsi que leurs voisines. Soit c une cellule extérieure à R; notons c_0 sa projection sur R, c'est-à-dire la cellule de R dont la distance à c dans le graphe de connexion est minimale,*

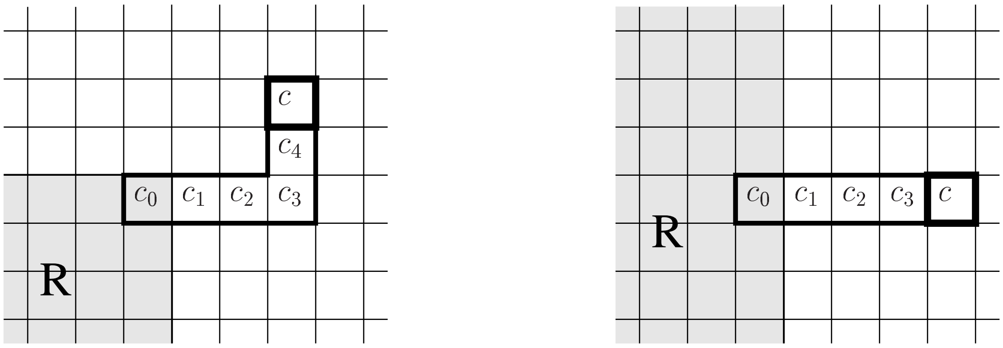


FIG. 3.5 – *Illustration de la proposition 3.5.3*

et soit λ cette distance. Alors,

- *si t est inférieur à λ, on a* : $\mathcal{A}(c,t) = 0$
- *si t est supérieur ou égal à λ, on a* : $\mathcal{A}(c,t) = \mathcal{A}(c_0,t-\lambda)$

Cette proposition se déduit des propositions 3.4.2 et 3.5.2 : soit $(c_0,c_1,\dots,c_\lambda = c)$ un chemin du graphe de connexion reliant c_0 à c et de longueur minimale. Remarquons que quel que soit i dans $\{1,2,\dots,\lambda\}$, l'ensemble C_i défini par :

$$C_i = R \cup \{c_1,\dots,c_i\}$$

est un c.c.c. vérifiant les hypothèses de la proposition 3.5.2, et qu'aucune des cellules $c_0,c_1,\dots,c_i$ n'est dans le support de la condition initiale. La cellule c_i ayant une seule voisine c_{i-1} dans C_i, il est facile de voir que pour tout état non nul de E, on a :

$$\mathcal{A}(c_i,t) = a \Longleftrightarrow \mathcal{A}(c_{i-1},t-1) = a$$

En itérant ce résultat, on obtient la proposition.

Corollaire 3.5.4. *Sous les hypothèses de la proposition 3.5.3, il existe un entier T positif et un instant t_0 ($t_0 \leq n^{cardR}$) tel que, pour toute cellule située à une distance λ de R, la trajectoire restreinte à cette cellule soit périodique de période T dès l'instant $t_0 + \lambda$.*

Il est clair que T peut être choisi comme étant la période minimale limite de la trajectoire de l'automate restreinte au rectangle R. On peut maintenant se demander quelle valeurs peut prendre cette période.

3.6 Périodes admissibles

Le lemme suivant est important pour la compréhension de la dynamique de cet automate, puisqu'il traite de la contamination de tout le domaine par la période n.

Lemme 3.6.1. *Supposons qu'il existe une cellule c du plan telle la trajectoire restreinte à cette cellule, $\mathcal{A}(.,t)_{|c}$, soit périodique de plus petite période n à partir d'un certain rang. Alors, toute cellule du plan à la même propriété.*

Preuve. Il suffit de prouver ce résultat pour une cellule d voisine de c. L'hypothèse faite sur la cellule c dans le lemme équivaut à l'existence d'un instant t_0 tel que, quel que soit l'entier k, on ait :

$$\mathcal{A}(c,t_0 + k.n) = 1$$

La cellule d étant voisine de c, on déduit immédiatement de la règle du jeu de l'automate que la suite $(\mathcal{A}(d,t_0 = k.n))_{k\in\mathbb{N}}$ est décroissante à valeurs dans $\{0,1,\ldots,n-1\}$, et donc constante à partir d'un certain rang k_0; nous avons donc deux possibilités :

- Soit $\forall t \in \mathbb{N}, \mathcal{A}(d,t_0 + k_0 n + t) = 0$: impossible car $\mathcal{A}(c,t_0 + kn) = 1, \forall k \in \mathbb{N}$.
- Soit $\forall t \in \mathbb{N}, \mathcal{A}(d,t_0 + k_0 n + t) = \mathcal{A}(d,t_0 + k_0 n) + t \mod n$.

D'où le résultat annoncé. □

La démonstration du théorème suivant, que nous ne reproduirons pas dans ce document (voir [AR84]) est assez technique, et utilise fortement l'existence du squelette limite (proposition 3.4.2).

Théorème 3.6.2. *Supposons le nombre d'états excitateurs e supérieur ou égal à 2. Soit $\mathcal{A}(.,t)$ une trajectoire pérenne de l'automate. Alors pour toute cellule c du domaine, la trajectoire restreinte à c, $\mathcal{A}(c,t)$ est périodique à partir d'un certain rang, de période égale au nombre d'états n.*

Nous savons maintenant que, ultimement, si le nombre d'états excitateurs n'est pas un, chaque cellule voit la succession cyclique de tous les états (période n). Mais qu'en est-il de l'organisation spatiale

3.7 Fronts d'onde

Définition 3.7.1. *Soit U un état du plan. Un front d'onde de U est une suite de cellules $(c_0,c_1,\ldots,c_n,\ldots)$, éventuellement finie, telle que :*

- *les états $U(c_i)$ sont égaux à 1 pour tout i dans $\{0,1,\ldots,n,\ldots\}$.*
- *les cellules c_i et c_{i+1} sont voisines au sens large pour tout i dans $\{0,1,\ldots,n-1,\ldots\}$.*
- *les cellules de cette suite sont toutes distinctes (sauf éventuellement c_0 et c_n si la suite est finie égale à $(c_0,\ldots,c_{n-1},c_n)$; on dira dans ce cas que le front d'onde est fermé).*

Définition 3.7.2. *Soit U un état du plan. On dira qu'un front d'onde fermé*

$$(c_0,\ldots,c_{n-1},c_n = c_0)$$

est "circulaire" si la région borné du plan délimitée par la ligne brisé fermée sans point double représentant ce front d'onde est telle que son intersection avec toute droite de pente 1 ou -1 est convexe; on dira qu'il entoure un borné B du plan si les centres des cellules de B sont contenus dans le compact délimité par ce front.

Donnons maintenant les premières propriétés de ces fronts d'onde.

Proposition 3.7.3. *Soient $\mathcal{A}(.,t)$ une trajectoire de l'automate dont la condition initiale est à support borné, et R un rectangle contenant les cellules de ce support, ainsi que leurs*

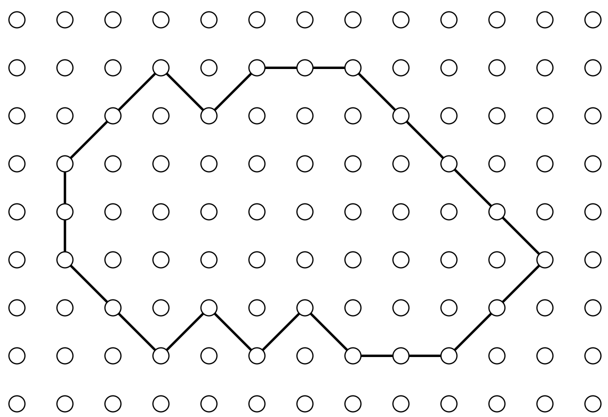


FIG. 3.6 – *Exemple de front d'onde circulaire*

voisines. Soit c une cellule dont l'état en un instant t est égal à 1. Supposons sa distance à R supérieure ou égale au périmètre de ce rectangle. Alors, cette cellule appartient à un unique front d'onde F de $\mathcal{A}(.,t)$; F est circulaire et entoure R (Si on suppose que c est extérieur à un front d'onde circulaire F_0 entourant R, on a le même résultat; de plus, F entoure F_0 et la distance de ces deux fronts est supérieure ou égale à n).

Cette proposition est elle aussi assez technique, et utilise fortement la notion de squelette limite. Nous allons maintenant donner une description globale d'une trajectoire de l'automate, à partir d'un instant "assez grand", en utilisant la notion de front d'onde.

Donnons nous $\mathcal{A}(.,t)$ une trajectoire de l'automate dont la condition initiale est à support borné et R un rectangle contenant les cellules de ce support ainsi que leurs voisines, et commençons par supposer que le nombre d'états excitateurs est supérieur ou égal à 2. Soit t_0 un instant à partir duquel la trajectoire restreinte à R est, soit identiquement nulle, soit périodique de période n (nous savons que dans les deux cas, t_0 peut être majoré par n^{cardR}).

Tirons tout d'abord les conclusions de la proposition 3.5.3 et du théorème 3.6.2; nous avons le résultat formel suivant :

Si c est une cellule extérieure à R, c_0 sa projection sur R et λ la distance de c à c_0, alors, pour tout instant t, on a :

$$\begin{cases} 0 \leq t < \lambda \Rightarrow \mathcal{A}(c,t) = 0 \\ \lambda \leq t < t_0 + \lambda \Rightarrow \mathcal{A}(c,t) = \mathcal{A}(c_0,t-\lambda) \\ t_0 + \lambda \leq t \Rightarrow \begin{cases} \mathcal{A}(c,t) = 0 & \text{si } \mathcal{A}(c,t) \text{ n'est pas pérenne} \\ \mathcal{A}(c,t) = \mathcal{A}(c_0,t_0) + t - t_0 - \lambda \mod n & \text{si } \mathcal{A}(c,t) \text{ est pérenne} \end{cases} \end{cases}$$

En utilisant ce résultat et la proposition 3.7.3 sur les fronts d'ondes circulaires, on montre sans peine le théorème suivant :

Théorème 3.7.4. *Supposons le nombre d'états excitateurs supérieur ou égal à 2. Soient $\mathcal{A}(.,t)$ une trajectoire de l'automate dont la condition initiale est à support borné, inclus* *dans un rectangle R^0, R le rectangle contenant les cellules de R^0 ainsi que leurs voisines et t_0 un instant à partir duquel la trajectoire restreinte à R est, soit nulle, soit périodique (de période n). Pour tout entier positif ou nul λ, notons $B(\lambda)$ l'ensemble des cellules situées à une distance de R inférieure ou égale à λ. Alors :*

- *Pour tout instant t supérieur à zéro, l'état du complémentaire de $B(t-1)$ est nul.*
- *Si la trajectoire $\mathcal{A}(.,t)$ n'est pas pérenne, pour tout instant supérieur ou égal à t_o, l'état de $B(t-t_0)$ est nul; et si $\mathcal{A}(.,t)$ n'est pas identiquement nul, on observe dans $B(t-1)\backslash B(t-t_0)$ un ou plusieurs fronts d'ondes, tous circulaires et entourant R.*
- *Si la trajectoire $\mathcal{A}(.,t)$ est pérenne et si en un instant supérieur ou égal à $_0$, il existe un front d'onde F circulaire entourant R et contenu dans $B(t-t_0)$, (cette propriété étant nécessairement vérifiée si t est supérieur ou égal à $t_0+n-1+\frac{3}{2}$(périmètre de R)), l'intersection des fronts d'ondes de $\mathcal{A}(.,t)$ avec $B(t-t_0)\backslash R^0$ est constitués de cellules de $B(t-t_0)\backslash R^0$ situées à une distance multiple de n du front F et tous les fronts extérieurs à F sont circulaires.*
- *Dans tous les cas, s'il existe un front d'onde circulaire entourant R en un instant t, il se dilate à la vitesse l : les cellules situées à la distance l de ce front et extérieures à celui-ci forment un front d'onde circulaire de $\mathcal{A}(.,t+1)$.*

Pour conclure ce chapitre, remarquons que nous n'avons décrit la structuration spatio-temporelle créée par une trajectoire de l'automate qu'à l'extérieur d'un rectangle R^0 contenant le support de la condition initiale de cette trajectoire. En ce qui concerne la restriction de la trajectoire à ce rectangle, nous n'avons donné de résultat que sur son comportement temporel.

Citons Allouche et Reder :

> *pour terminer, résumons de manière totalement informelle les propriétés que nous avons exposées dans cet article concernant les suites de l'automate étudié : Sous une hypothèse peu restrictive, le* ***support de la condition initiale*** *agit comme un "centre de pulsations" qui bat de façon désordonnée pendant un certain temps, puis, soit bat à une fréquence parfaitement déterminée, soit cesse définitivement de battre; chacune de ces pulsations déclenche une onde circulaire qui se propage dans le milieu, et toutes ces ondes se propagent à la même vitesse.*

Chapitre 4

Modèle de type AVN

4.1 Présentation

L'automate cellulaire étudié précédemment (Chapitre 3) est en fait un cas particulier de l'automate dont nous allons traiter dans ce chapitre. Soient deux entiers a et v tels que $a < n$, $v < n$ et $a < v$. On définit quatre sous-ensembles de l'ensemble des états E :

- 0 : l'état neutre.
- $J = \{1..a-1\}$: l'ensemble des états jeunes.
- $A = \{a..v-1\}$: l'ensemble des états adultes.
- $V = \{v..n-1\}$: l'ensemble des états vieux.

La fonction de transition est définie comme suit, et illustrée par la figure 4.1:

`si` $\mathcal{A}(c,t) \neq 0$
`alors` $\mathcal{A}(c,t+1) = \mathcal{A}(c,t) + 1 \mod n$
`sinon`
  `si` $\exists\ d \in \mathcal{V}(c)$ telle que $\mathcal{A}(d,t) \in A$
  `alors` $\mathcal{A}(c,t+1) = 1$
  `sinon` $\mathcal{A}(c,t+1) = \mathcal{A}(c,t) = 0$

Chaque cellule passe à chaque itération à l'état suivant (modulo n), seul le passage de 0 en 1 étant conditionnel à la présence d'une cellule dans un état adulte dans le voisinage. Par rapport à l'automate précédent, nous avons seulement inséré l'ensemble des états jeunes entre l'état neutre et l'ensemble des états excitateurs que nous appelons ici les états adultes.

Une interprétation "forestière" possible pour notre automate, représentée par la figure 4.2, est la suivante : chaque cellule représente potentiellement un arbre, et l'état de la cellule représente son état de maturité.

- L'état 0 est une parcelle vide.
- Les jeunes ne produisent pas de graines.

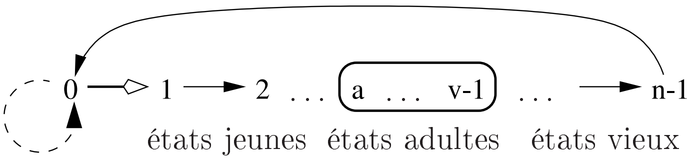


Fig. 4.1 – *L'automate cellulaire*

- Les adultes produisent des graines, et peuvent donc ensemencer leurs voisins vides.
- Les vieux ne produisent pas de graine, mais occupent la cellule.

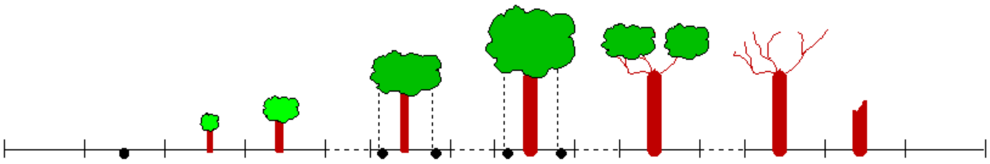


Fig. 4.2 – *Interprétation "forestière" de l'automate "AVN"*

4.2 Simulations préliminaires

Nous pouvons nous rappeler ici les principaux résultats obtenus dans le cas particulier de cet automate que représente l'automate cellulaire de type Milieu Excitable, en espérant qu'ils soient vrais dans le cas plus général de l'automate de type AVN :

- Proposition 3.5.2 : Réduction du domaine à un borné; l'état des cellules appartenant à un certain sous-ensemble borné du domaine ne dépend pas de l'état des cellules extérieurs à ce sous-ensemble.
- Section 3.7 : Organisation sous forme de fronts d'onde à l'extérieur du support de la condition initiale.
- Théorème 3.6.2 : Si le nombre d'états excitateurs est supérieur ou égal à deux, les trajectoire issues de l'automate sont soit non-pérennes, soit ultimement périodiques, de période n, le nombre total d'états.

Dans un premier temps, quelque simulations vont nous permettre de savoir à quoi nous en tenir en ce qui concerne ces résultats.

- Pour ce qui est de la réduction du domaine, nous ne pouvons simuler notre automate que sur un domaine borné. En conséquence, nous avons choisit de tester l'indépendance d'un rectangle de cellules contenant les cellules du support de la condition initiale ainsi que leur voisines au sens large, par rapport à ses voisines au sens large. Pour cela nous allons simuler notre automate avec la même condition initiale, sur deux domaines, le deuxième étant muni d'une bordure supplémentaire de cellules dans l'état neutre (figure 4.3). Nous avons testé tous les couples $\{a,v\}$ pour $n = 15$ (91 expériences), $n = 16$ (105 expériences), et $n = 17$ (120 expériences), pour dix conditions initiales différentes, soit 3160 expériences. Les trajectoire obtenues pour chaque cellule du sous-domaine commun ont été identiques, ce qui tend à nous convaincre de l'indépendance d'un rectangle contenant le support de la condition initiale ainsi que ses voisines au sens large.

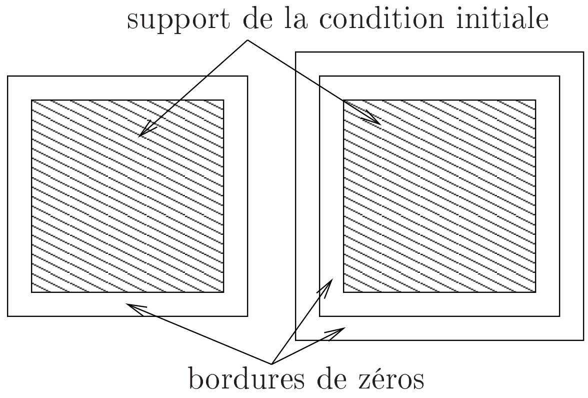
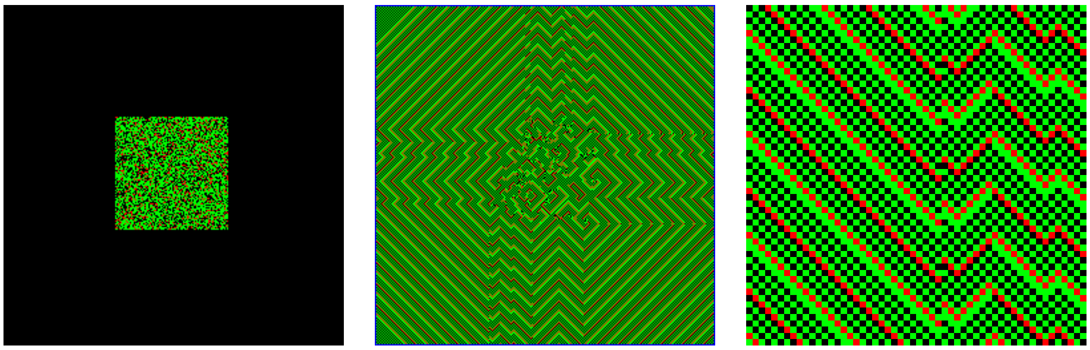


FIG. 4.3 – *Expérience avec une bordure neutre.*

– Pour ce qui est des fronts d'ondes, le comportement, à première vue, à l'air d'être comparable. Mais si l'on regarde de plus prés (figure 4.4), il ne se forme pas toujours de front d'onde fermé, au sens défini pour l'automate de type milieu excitable (voir section 3.7).



(a) état initial (b) un état de la période (c) détail

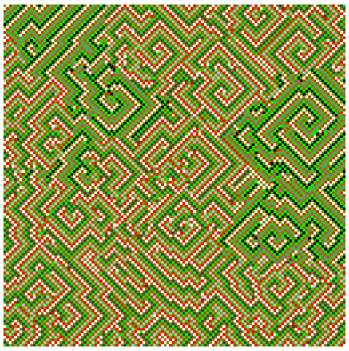
FIG. 4.4 – *Automate de type AVN et Fronts d'Ondes.*

– Enfin, les premières simulations effectuées ont eut l'air de confirmer le dernier point. Mais après un grand nombre de simulations, nous avons observé des trajectoires de périodes différentes de n, avec deux excitateurs, ce qui remet en question ce résultat, pour notre automate.

A la lumière de ces premières simulations, nous allons analyser l'automate, afin de mieux comprendre sa dynamique.

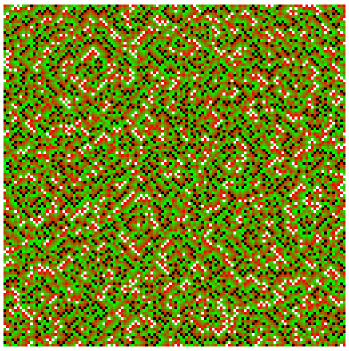
4.3 Remarques à caractère théorique

La figure 4.5 représente trois images obtenues par simulation de l'automate "AVN", en utilisant trois jeux de paramètres différents et le même état initial. Chaque image est constituée de 128×128 pixels, chaque pixel étant associé à une cellule. On a ici représenté l'état neutre en blanc, les états jeunes en vert, les états adultes en rouge et les états vieux en noir. On peut voir que les résultats peuvent être très différents, bien que les trois jeux de paramètres soient assez proches.



(a) a=10 v=11 n=16

(b) a=11 v=12 n=16



(c) a=13 v=15 n=16

FIG. 4.5 – *Trois exemples pour le même état initial.*

Pour comprendre le fonctionnement de notre automate, nous allons tout d'abord l'étudier sur un domaine très réduit, composé de deux cellules.

4.3.1 L'automate bicellulaire

Le domaine est constitué ici uniquement de deux cellules, évidemment voisines l'une de l'autre, que nous appellerons c_1 et c_2. Par la suite, nous noterons $\boxed{e_1 \mid e_2}$ l'état de l'automate qui à la cellule c_1 fait correspondre l'état e_1, et à c_2 l'état e_2. De plus $U \xrightarrow{p} V$ signifie que l'on passe de l'état U à l'état V par p itérations de l'automate. Lorsque p est égal à un, on note $U \rightarrow V$.

Remarque 4.3.1. *Nous pouvons, sans perte de généralité n'étudier que les trajectoires issues d'états initiaux du type* $\boxed{0 \mid e}$. *En effet, si aucune des deux cellules n'est en l'état*

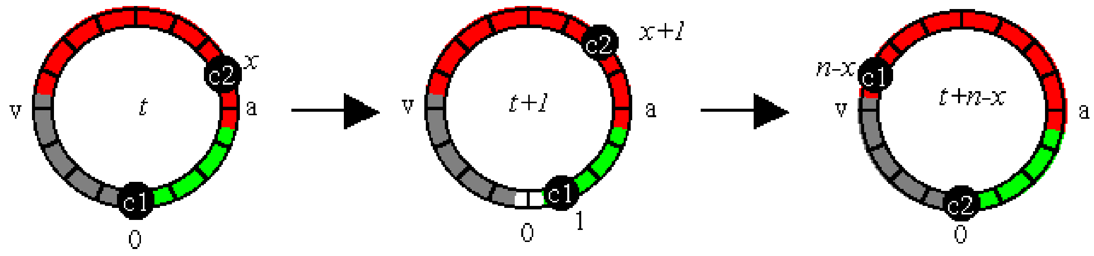


FIG. 4.6 – *Illustration de la proposition 4.3.2*

neutre, il suffit d'itérer l'automate jusqu'à qu'une des deux cellules soit en l'état neutre. Soient e_1 et e_2 deux entiers non nuls appartenant à E tels que $e_1 > e_2$:

$$\forall t \geq n - e_1, \mathcal{A}(.,t,0,\begin{array}{|c|c|}\hline e_1 & e_2 \\ \hline\end{array}) = \mathcal{A}(.,t,n-e_1,\begin{array}{|c|c|}\hline 0 & n-e_1+e_2 \\ \hline\end{array})$$

En effet, $\mathcal{A}(.,n-e_1,0,\begin{array}{|c|c|}\hline e_1 & e_2 \\ \hline\end{array}) = \begin{array}{|c|c|}\hline 0 & n-e_1+e_2 \\ \hline\end{array} = \mathcal{A}(.,n-e_1,n-e_1,\begin{array}{|c|c|}\hline 0 & n-e_1+e_2 \\ \hline\end{array})$.

Proposition 4.3.2. *Si c_1 est excitée par c_2 en l'état x, alors quand c_2 sera en zéro, c_1 sera en l'état $n-x$, état symétrique de x sur le cercle des états par rapport à l'axe vertical passant par zéro (figure 4.6).*

Preuve. Soit x un état excitateur :

$$\begin{array}{|c|c|}\hline 0 & x \\ \hline\end{array} \rightarrow \begin{array}{|c|c|}\hline 1 & x+1 \\ \hline\end{array} \xrightarrow{n-x-1} \begin{array}{|c|c|}\hline n-x & 0 \\ \hline\end{array}$$

□

Théorème 4.3.3. *Si v est inférieur à $n/2$, aucune trajectoire n'est pérenne.*

Preuve. Soit x un état excitateur.

$$\begin{array}{|c|c|}\hline 0 & x \\ \hline\end{array} \xrightarrow{n-x} \begin{array}{|c|c|}\hline n-x & 0 \\ \hline\end{array}$$

Mais $x < n/2 \Rightarrow n-x > n/2 \Rightarrow n-x > v$ d'où

$$\begin{array}{|c|c|}\hline n-x & 0 \\ \hline\end{array} \xrightarrow{x} \begin{array}{|c|c|}\hline 0 & 0 \\ \hline\end{array}$$

□

Théorème 4.3.4. *Soit $v > n/2$. Une trajectoire est pérenne si et seulement si la première excitation est causée par un état appartenant à l'intervalle $]n-v;v[$ (figure 4.7).*

Preuve. Soit x un état excitateur.

$$\begin{array}{|c|c|}\hline 0 & x \\ \hline\end{array} \xrightarrow{n-x} \begin{array}{|c|c|}\hline n-x & 0 \\ \hline\end{array}$$

– Soit $x < n-v \Rightarrow n-x > v$, d'où $\begin{array}{|c|c|}\hline n-x & 0 \\ \hline\end{array} \xrightarrow{x} \begin{array}{|c|c|}\hline 0 & 0 \\ \hline\end{array}$: trajectoire non pérenne.

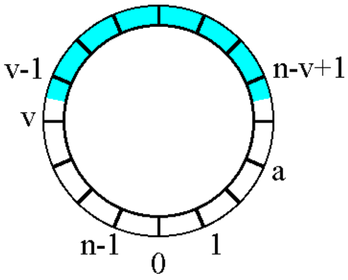


FIG. 4.7 – *Illustration de la proposition 4.3.4*

- Soit $x \geq n - v \Rightarrow n - x \leq v$.
  - Soit $n - x \geq a$ d'où $\boxed{n-x \mid 0} \xrightarrow{x} \boxed{0 \mid x}$: trajectoire pérenne.
  - Soit $n - x < a$ d'où $\boxed{n-x \mid 0} \xrightarrow{a-n+x} \boxed{a \mid 0} \xrightarrow{n-a} \boxed{0 \mid n-a}$
    - Soit $n - a \geq a$ d'où $\boxed{0 \mid n-a} \xrightarrow{a} \boxed{a \mid 0}$: trajectoire pérenne.
    - Soit $n - a < a$ d'où $\boxed{0 \mid n-a} \xrightarrow{2a-n} \boxed{0 \mid a}$: trajectoire pérenne.

□

Théorème 4.3.5. *La période d'une trajectoire vivante est* $max(2a,n)$.

Preuve. Il suffit de reprendre la démonstration précédente, et de compter le nombre d'itérations constituant une période pour chaque trajectoire vivante :

- $n - x \geq a$: $\boxed{0 \mid x} \xrightarrow{n-x} \boxed{n-x \mid 0} \xrightarrow{x} \boxed{0 \mid x}$: $n - x + x = n$ itérations. Mais
  $\left.\begin{array}{l} n - x \geq a \\ x \quad \geq a \end{array}\right\} \Rightarrow 2a \leq n$. La période est donc $n = max(2a,n)$.
- $n - x < a$:
  - $n - a \geq a \Leftrightarrow 2a \leq n$: $\boxed{a \mid 0} \xrightarrow{n-a} \boxed{0 \mid n-a} \xrightarrow{a} \boxed{a \mid 0}$: $n - a + a = n$ itérations. La période est donc $n = max(2a,n)$.
  - $n - a < a \Leftrightarrow 2a > n$: $\boxed{a \mid 0} \xrightarrow{n-a} \boxed{0 \mid n-a} \xrightarrow{2a-n} \boxed{0 \mid a} \xrightarrow{n-a} \boxed{n-a \mid 0} \xrightarrow{2a-n}$ $\boxed{a \mid 0}$: $n - a + 2a - n + n - a + 2a - n = 2a$ itérations. La période est donc $n = max(2a,n)$.

□

Maintenant que nous connaissons parfaitement le comportement de notre automate sur un terrain composé uniquement de deux cellules, nous allons tenter d'utiliser ces résultats pour comprendre sa dynamique sur un domaine à deux dimensions.

4.3.2 La période n

Nous étudierons ici l'automate sur un domaine à deux dimensions, et plus particulièrement les caractéristiques de la période n. Nous pourrons ainsi utiliser les résultats de l'automate bi-cellulaire pour la compréhension de sa dynamique sur un domaine bi-dimensionnel.

Théorème 4.3.6. *Soit D un domaine à deux dimensions, et D' un sous-domaine de celui-ci. Soit U_0 un état de D. Si $\mathcal{A}_{|D'}(.,t,0,U_{0|D'})$ est n-périodique, alors*

$$\mathcal{A}_{|D'}(.,t,0,U_{0|D'}) = \mathcal{A}(.,t,0,U_0)_{|D'}$$

Ce théorème signifie que l'état de tout groupe de cellules de trajectoire n-périodique ne dépend pas des cellules extérieures au groupe.

Preuve. Soit c une cellule de D' telle que

$$\begin{cases} \mathcal{A}_{|D'}(.,t,0,U_{0|D'}) = \mathcal{A}(.,t,0,U_0)_{|D'} \\ \mathcal{A}_{|D'}(c,t+1,0,U_{0|D'}) \neq \mathcal{A}(c,t+1,0,U_0)_{|D'} \end{cases}$$

Le passage de 0 à 1 étant le seul dépendant de l'état des voisins, nous pouvons conclure que :
$\mathcal{A}_{|D'}(c,t,0,U_{0|D'}) = \mathcal{A}(c,t,0,U_0) = 0$, d'où
$\mathcal{A}_{|D'}(c,t+1,0,U_{0|D'}) = 1$, car $\mathcal{A}(.,t,0,u_0)$ est n-périodique, d'où
$\mathcal{A}(c,t+1,0,U_0)_{|D'} = 0$.
Mais, $\exists d \in V(c) \cap D'$ telle que $\mathcal{A}(d,t,0,U_0) \in E$. D'où $\mathcal{A}(c,t+1,0,U_0)_{|D'} = 1$: contradiction. □

Théorème 4.3.7. *Soit c une cellule de D et U_0 un état de D tel que $\mathcal{A}(c,t,0,U_0)$ soit n-périodique. Alors, pour une cellule d de D, il existe un entier t tel que $\mathcal{A}(d,t,0,U_0)$ soit n-périodique à partir de t.*

Ce théorème, si le domaine D est borné, signifie que toutes les cellules du domaine finissent par adopter une trajectoire n-périodique si au moins l'une d'entre elles est dans ce cas.

Preuve. Il nous suffit de montrer le théorème pour d, une cellule voisine de c. Soit t_0 un entier positif tel que :

$$\forall k \in \mathbb{N}^*, \mathcal{A}(c,t_0 + kn) = a$$

Alors la suite $(A(d,t_0 + kn))_{k\in\mathbb{N}}$ est décroissante. En effet

- Soit $\mathcal{A}(d,t_0 + kn) = x$ avec $x \neq 0 \Rightarrow \mathcal{A}(d,t_0 + (k+1)n) \leq x$.
- Soit $\mathcal{A}(d,t_0 + kn) = 0 \Rightarrow \mathcal{A}(d,t_0 + kn + 1) = 1 \Rightarrow \mathcal{A}(d,t_0 + (k+1)n) = 0$.

D'où la suite $(A(d,t_0+kn))_{k\in\mathbb{N}}$ devient constante, c'est-à-dire que la trajectoire de la cellule d devient n-périodique. □

Nous savons maintenant que lorsqu'un groupe de cellules adopte la période n, non seulement il la conserve quoiqu'il arrive mais de plus, il entraîne toutes les autres cellules. Mais nous pouvons nous poser la question dans l'autre sens, à savoir que se passe-t-il lorsqu'une trajectoire de l'automate est n-périodique?

4.3.3 Les cycles auto-excitateurs n-périodiques.

Pour clarifier, mais aussi justifier ce titre, nous allons tout d'abord définir ce qu'est un cycle auto-excitateur.

Définition 4.3.8. *Un cycle auto-excitateur est un cycle de cellules* $(c_1,c_2,\ldots,c_q)$ *(définition 2.3.2) tel que :*

$$\left.\begin{array}{l}\mathcal{A}(c_i,t)=0\\ \mathcal{A}(c_i,t+1)=1\end{array}\right\}\Rightarrow \mathcal{A}(c_{i+1},t)\in A$$

En clair, un cycle auto-excitateur est un chemin fermé sans point double sur le graphe de connexion, pour lequel chaque cellule est excitée par la cellule précédente. Par exemple, l'automate bi-cellulaire étudié précédemment peut être un exemple de cycle auto-excitateur de période n, constitué de deux cellules.

Afin de comprendre le rôle des cycles auto-excitateurs dans la dynamique de notre automate, prenons un état de la période d'un automate n-périodique. Quand une de ses cellules est en zéro, une de ses voisines l'excite (période n). Au bout de n pas de temps, la même voisine sera excitatrice. Choisissons une cellule dans l'état neutre, et une de ses voisines excitatrice. Quand cette cellule se trouve à son tour dans l'état neutre, elle possède une voisine excitatrice, ainsi de suite. Le nombre de cellules étant fini, on retombe forcement sur une cellule déjà rencontrée : nous avons construit un cycle auto-excitateur n-périodique. Nous pouvons donc énoncer le théorème suivant :

Théorème 4.3.9. *Si une trajectoire est n-périodique, elle contient au moins un cycle auto-excitateur n-périodique.*

A propos des cycles auto-excitateur n-périodiques, nous pouvons ennoncer la propriété suivante :

Proposition 4.3.10. *Un cycle de cellules* $(c_1,\ldots,c_q)$ *est auto-excitateur de période n si et seulement si il vérifie pour tout t :*

$$\mathcal{A}(c_i,t)+a\leq\mathcal{A}(c_{i+1},t)\leq\mathcal{A}(c_i,t)+v-1 \tag{4.1}$$

Preuve. En effet, un cycle $(c_1,\ldots,c_q)$ auto-excitateur et n-périodique, d'après la définition 4.3.8 est tel que :

$$\mathcal{A}(c_i,t)=0\Rightarrow\mathcal{A}(c_{i+1},t)\in A$$

De plus, la trajectoire de toutes les cellules du cycle est n-périodique, d'où

$$\forall e\in E,\mathcal{A}(c_i,t)=e\Rightarrow e+a\leq\mathcal{A}(c_{i+1},t)\leq e+v-1$$

$$\Leftrightarrow 4.1$$

De plus, si l'équation 4.1 est vérifiée :

$$\mathcal{A}(c_i,t)=0\Rightarrow\mathcal{A}(c_{i+1},t)\in A\Rightarrow\mathcal{A}(c_i,t+1)=1$$

□

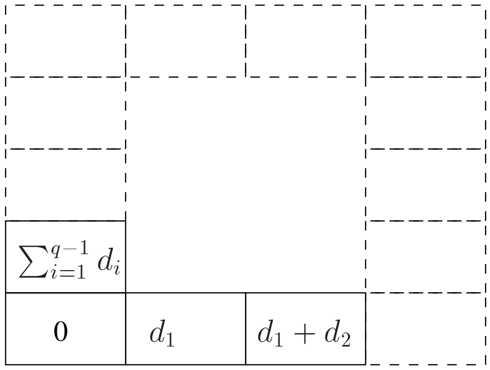


FIG. 4.8 – *État d'un cycle auto-excitateur n-périodique*

Nous avons besoin ici d'une autre définition :

Définition 4.3.11. *Soit $(c_1, \ldots, c_q)$ un cycle auto-excitateur n-périodique. Nous appellerons **déphasage** entre les cellules c_i et c_{i+1}, et nous noterons d_i, l'entier tel que :*

$$\mathcal{A}(c_{i+1}, t) = \mathcal{A}(c_i, t) + d_i$$

Ces considérations sur les cycles auto-excitateurs n-périodiques vont nous permettre d'établir une relation entre la longueur minimum d'un de ces cycles, et les paramètres de l'automate. Prenons un cycle de q cellules (q doit être pair). La relation à tout instant entre les états des cellules qui se suivent est donc (figure 4.8) :

$$\mathcal{A}(c_i) = \mathcal{A}(c_{i+1}) + d_i \mod n, \forall i \in \{1,2\ldots q\} \tag{4.2}$$

D'où:

$$\mathcal{A}(c_1) = \mathcal{A}(c_1) + \sum_{i=1}^{q} d_i \mod n \tag{4.3}$$

Nous pouvons donc énoncer la propriété suivante:

$$\sum_{i=1}^{q} d_i = k \times n \text{ avec } k \text{ entier} \tag{4.4}$$

Soit n_e le nombre d'états excitateurs ($n_e = v - a$). Le décalage d_i est compris entre a et $v - 1 = a + n_e - 1$, d'où

$$q \times a \leq \sum_{i=1}^{q} d_i \leq q \times (a + n_e - 1) \tag{4.5}$$

On obtient donc la relation suivante :

$$qa \leq kn \leq q(a + n_e - 1) \tag{4.6}$$

Cas particulier: Un seul excitateur : a
la relation (4.6) devient:

$$qa = kn \Leftrightarrow q = \frac{kn}{a} \tag{4.7}$$

Mais nous savons que q et k sont entiers, d'où:

$$q = k \times \frac{\frac{n}{\mathrm{PGCD}(n,a)}}{\frac{a}{\mathrm{PGCD}(n,a)}} = k' \times \frac{a}{\mathrm{PGCD}(n,a)} \times \frac{\frac{n}{\mathrm{PGCD}(n,a)}}{\frac{a}{\mathrm{PGCD}(n,a)}} \tag{4.8}$$

On peut ainsi déterminer q_{min} la longueur minimum possible d'un cycle périodique, dans le cas où l'on a un seul excitateur :

$$q_{min} = \begin{cases} \frac{n}{\mathrm{PGCD}(n,a)} & \text{si } \frac{n}{\mathrm{PGCD}(n,a)} \text{ est pair} \\ 2 \times \frac{n}{\mathrm{PGCD}(n,a)} & \text{sinon} \end{cases} \tag{4.9}$$

Nous pouvons illustrer cette relation par l'exemple suivant : il est constitué de quatre séries de dix expériences :

- Dix conditions initiales différentes, mais semblables : l'état initial de chaque cellule est tiré au sort uniformément parmi tous les états.
- Quatre jeux de paramètres, avec un seul excitateur : $a = 3{,}4{,}5{,}6$.

Nous avons, pour chaque série d'expériences, calculé q_{min}, relevé les périodes atteintes ainsi que la longueur des transitoires. Les résultats sont consignés dans le tableaux suivant :

a	v	n	$q = nk/a$	q_{min}	période	estimation des transitoires
3	4	12	$4k$	4	12	70
4	5	12	$3k$	6	12	500 à 1000
5	6	12	$12k/5$	12	40, 50 ou 80	1000 à 8000
6	7	12	$2k$	2	12	40

- Dans la première configuration ($a = 3$), des cycles de longueur 4 peuvent se former.
- Pour la suivante ($a = 4$), ce sont des cycles de longueur 6, plus rares: le transitoire est plus long; mais la période est la même: 12.
- Pour le troisième cas ($a = 5$), les cycles n-périodiques possibles contiendraient 12 cellules; dans ces exemples, il ne s'en forme pas, d'où des périodes de 40, 50 ou 80 itération, et non pas 12.
- Pour le dernier cas ($a = 6$) , des cycles n-périodiques de deux cellules peuvent se former; d'où une période 12 et un transitoire encore plus court que dans les deux premiers cas.

Plus nous disposons d'états excitateurs, plus il est facile de construire des cycles n-périodiques courts vérifiant l'égalité 4.4. Nous comprenons donc ici pourquoi le nombre de trajectoires de période différente de n diminue rapidement lorsque le nombre d'excitateurs augmente.

4.4 Reprise systématique des simulations et analyse

Nous allons ici présenter plusieurs séries de simulations de l'automate cellulaire de type AVN. Cet automate dispose de trois paramètres (a, v et n), et les conditions initiales possibles sont innombrables. Il nous faut donc réduire le champ expérimental, et pour cela, utiliser les résultats théoriques obtenus précédemment. Les différentes séries de simulations présentées possèdent donc des orientations différentes, et chacune sera analysée. Ces séries d'expériences n'ayant pas été effectuées au même moment, et dans le même état de compréhension de l'automate, les protocoles adoptés seront quelque peu différents.

4.4.1 Les périodes observées

Cette série d'expériences se concentre sur les périodes des trajectoires de l'automate de type AVN. Chaque jeu de paramètres tel que n, le nombre total d'états, soit inférieur ou égal à 16 va être visité. Pour un n donné, il existe $\frac{(n-1)(n-2)}{2}$ jeux de paramètres, ce qui nous donne en tout $\displaystyle\sum_{n=3}^{16} \frac{(n-1)(n-2)}{2} = 560$ jeux de paramètres. Pour chacun d'entre eux, vont être testés quatre groupes de dix conditions initiales de même type :

- L'état initial de chaque cellule tiré au hasard dans $[0..n-1]$.
- Un quart des cellules (choisies au hasard) en 0, les autres tirées au hasard dans $[0..n-1]$.
- La moitié en 0, les autres tirées au hasard dans $[0..n-1]$.
- Trois quart en 0, les autres tirées au hasard dans $[0..n-1]$.

Au total, cette campagne représente $560 \times 4 \times 10$ soit 22400 trajectoires simulées. Les tableaux 4.1 à 4.12 synthétisent l'information concernant les périodes observées lors des simulations. Chaque cellule concerne 10 expériences du même type, c'est-à-dire 10 conditions initiales du même type, mais de tirages différents. Elles contiennent n si n est la seule période obtenue, et $p_{min} : p_{max}$, les périodes minimum et maximum observées sinon.

Il y a trois groupes de tableaux, rassemblant les expériences effectuées avec:

- un seul excitateur: les tableaux 4.1, 4.2, 4.3 et 4.4.
- deux excitateurs: les tableaux 4.5, 4.6, 4.7 et 4.8.
- trois excitateurs: les tableaux 4.9, 4.10, 4.11 et 4.12.

Les autres expériences donnant toujours une période n, elles ne figurent pas ici!

n \a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	n														
3	n	n													
4	n	n	n												
5	n	5:16	n	n											
6	n	n	n	n	n										
7	n	7:44	24:30	n	7:150	36:36									
8	n	n	8:18	n	n	n	8:42								
9	n	9:40	n	20:72	n	n	28:28	48:96							
10	n	n	10:12	32:64	n	n	10:42	10:176	54:108						
11	11:12	12:24	12:12	24:80	40:50	n	56:56	80:80	36:450	80:80					
12	n	n	n	n	40:80	n	n	n	n	n	88:88				
13	13:14	28:40	30:42	40:40	30:70	48:60	13:14	80:160	54:108	40:40	66:66	96:96			
14	n	n	14:30	14:104	30:60	48:96	n	n	14:72	60:100	44:44	72:72	104:104		
15	16:16	16:16	n	16:16	n	48:48	56:70	16:16	n	n	110:110	48:48	78:78	112:140	
16	n	n	16:36	n	50:50	16:84	56:70	n	16:162	n	66:66	n	52:182	84:84	120:150

Tab. 4.1 – *États initiaux équirépartis, 1 excitateur*

n \a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	n														
3	n	n													
4	n	n	n												
5	n	5:16	n	5:56											
6	n	n	n	n	n										
7	n	7:60	24:30	n	50:50	36:36									
8	n	n	8:18	n	n	n	42:42								
9	n	28:28	n	32:40	n	n	28:28	48:96							
10	n	n	10:42	32:32	n	n	10:84	32:112	54:108						
11	n	12:12	12:12	24:48	40:80	n	56:56	80:80	36:162	80:120					
12	n	n	n	n	40:50	n	n	n	n	n	66:88				
13	n	28:40	30:54	40:80	30:70	48:60	13:14	80:160	54:108	40:40	66:66	72:168			
14	n	n	14:30	14:136	30:60	48:60	n	n	14:72	100:100	44:44	72:72	78:104		
15	n	16:32	n	16:16	n	15:48	56:70	15:16	n	n	110:154	48:48	78:78	84:112	
16	n	n	16:18	n	50:50	16:72	56:70	n	16:162	n	66:66	n	52:286	84:84	120:120

Tab. 4.2 – *Un quart des cellules initialement en zéro, 1 excitateur*

n \a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	n														
3	n	n													
4	n	n	n												
5	n	12:16	n	5:72											
6	n	n	n	n	n										
7	n	7:60	8:24	n	50:50	36:36									
8	n	n	8:18	n	n	n	42:42								
9	9:10	20:40	n	24:32	n	n	28:28	48:64							
10	n	n	10:42	24:48	n	n	10:42	32:32	54:72						
11	11:12	12:24	12:12	24:24	30:50	11:12	56:56	80:80	36:126	60:60					
12	n	n	n	n	40:40	n	n	n	n	12:220	66:88				
13	13:14	28:28	30:42	40:88	30:60	48:72	13:14	80:80	54:54	40:40	66:110	72:96			
14	n	14:16	30:30	14:16	30:60	36:72	n	14:128	72:72	60:100	44:44	72:72	78:104		
15	15:16	16:32	n	16:16	n	16:72	56:56	15:16	n	n	110:154	48:48	78:78	84:112	
16	n	n	18:18	n	50:110	36:36	42:56	n	16:162	n	66:66	n	52:130	84:84	90:120

Tab. 4.3 – *La moitié des cellules initialement en zéro, 1 excitateurs*

n \a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	n														
3	n	n													
4	n	n	n												
5	5:6	6:12	n	16:16											
6	n	n	n	n	6:20										
7	7:8	8:8	18:24	n	30:60	36:36									
8	n	n	18:18	n	n	n	28:98								
9	10:10	10:32	n	24:32	n	n	28:28	48:48							
10	n	10:12	10:12	24:24	n	n	42:84	32:32	54:72						
11	12:12	12:12	12:12	24:24	30:40	n	56:56	48:80	36:36	60:100					
12	1:12	n	n	12:56	30:40	n	n	n	n	40:40	66:66				
13	1:14	14:16	15:30	16:40	30:30	36:48	13:14	80:80	54:54	40:40	44:66	72:72			
14	1:14	16:16	15:48	16:16	30:30	16:48	n	n	72:72	60:100	44:44	48:72	78:78		
15	1:16	16:16	15:18	16:16	n	36:48	42:56	15:16	n	n	66:110	48:48	52:78	84:84	
16	1:16	n	18:18	n	20:50	16:36	42:56	n	16:180	n	66:66	n	52:52	56:140	90:90

Tab. 4.4 – *Trois quarts des cellules initialement en zéro, 1 excitateurs*

n \a	1	2	3	4	5	6	7	8	9	10	11	12	13	14
3	n													
4	n	n												
5	n	n	n											
6	n	n	n	n										
7	n	n	n	n	n									
8	n	n	n	n	n	n								
9	n	n	n	n	n	n	n							
10	n	n	n	n	n	n	n	n						
11	n	n	n	n	n	n	n	n	n					
12	n	n	n	n	n	n	n	n	n	n				
13	n	n	n	n	13:40	n	n	n	n	n	n			
14	n	n	n	n	n	n	n	n	n	n	n	n		
15	n	n	n	n	n	15:48	n	n	n	n	n	n	n	
16	n	n	n	n	n	n	n	n	n	n	n	n	n	n

TAB. 4.5 – *États initiaux équirépartis, 2 excitateurs*

n \a	1	2	3	4	5	6	7	8	9	10	11	12	13	14
3	n													
4	n	n												
5	n	n	n											
6	n	n	n	n										
7	n	n	n	n	n									
8	n	n	n	n	n	n								
9	n	n	n	n	n	n	n							
10	n	n	n	n	n	n	n	n						
11	n	n	n	n	n	n	n	n	n					
12	n	n	n	n	n	n	n	n	n	n				
13	n	n	n	n	13:40	n	n	n	n	13:340	13:132			
14	n	n	n	n	n	n	n	n	n	n	n	14:72		
15	n	n	n	15:16	n	15:48	n	n	n	n	n	n	15:78	
16	n	n	n	n	n	n	n	n	n	n	n	n	n	n

TAB. 4.6 – *Un quart des cellules initialement en zéro, 2 excitateurs*

n \a	1	2	3	4	5	6	7	8	9	10	11	12	13	14
3	n													
4	n	n												
5	n	n	n											
6	n	n	n	n										
7	n	n	n	n	n									
8	n	n	n	n	n	n								
9	n	n	n	n	n	n	n							
10	n	n	n	n	n	n	n	n						
11	n	n	n	11:24	n	n	n	n	n					
12	n	n	n	n	n	n	n	n	n	n				
13	n	n	n	n	40:60	n	n	n	n	13:40	13:66			
14	n	n	n	n	14:30	n	n	n	n	n	14:44	14:72		
15	n	n	n	15:16	n	48:48	n	n	n	n	n	n	15:78	
16	n	n	n	n	n	16:36	n	n	n	n	n	n	n	16:84

TAB. 4.7 – *La moitié des cellules initialement en zéro, 2 excitateurs*

n \a	1	2	3	4	5	6	7	8	9	10	11	12	13	14
3	n													
4	n	n												
5	n	n	n											
6	n	n	n	n										
7	n	n	n	n	n									
8	n	n	n	n	n	n								
9	n	n	n	n	n	n	n							
10	n	n	n	n	n	n	n	n						
11	n	n	11:12	11:24	n	n	n	n	11:90					
12	n	n	n	n	n	n	n	n	n	n				
13	n	n	n	n	30:40	n	n	n	n	40:40	66:66			
14	1:14	n	n	14:16	14:30	n	n	n	n	n	14:44	72:72		
15	1:15	n	n	15:16	n	36:48	n	n	n	n	n	15:48	78:78	
16	1:16	n	16:18	n	n	16:36	n	n	n	n	16:66	n	16:52	84:84

TAB. 4.8 – *Trois quarts des cellules initialement en zéro, 2 excitateurs*

n \a	1	2	3	4	5	6	7	8	9	10	11	12	13
4	n												
5	n	n											
6	n	n	n										
7	n	n	n	n									
8	n	n	n	n	n								
9	n	n	n	n	n	n							
10	n	n	n	n	n	n	n						
11	n	n	n	n	n	n	n	n					
12	n	n	n	n	n	n	n	n	n				
13	n	n	n	n	n	n	n	n	n	n			
14	n	n	n	n	n	n	n	n	n	n	n		
15	n	n	n	n	n	n	n	n	n	n	n	n	
16	n	n	n	n	n	n	n	n	n	n	n	n	n

TAB. 4.9 – *États initiaux équirépartis, 3 excitateurs*

n \a	1	2	3	4	5	6	7	8	9	10	11	12	13
4	n												
5	n	n											
6	n	n	n										
7	n	n	n	n									
8	n	n	n	n	n								
9	n	n	n	n	n	n							
10	n	n	n	n	n	n	n						
11	n	n	n	n	n	n	n	n					
12	n	n	n	n	n	n	n	n	n				
13	n	n	n	n	n	n	n	n	n	n			
14	n	n	n	n	n	n	n	n	n	n	n		
15	n	n	n	n	n	n	n	n	n	n	n	n	
16	n	n	n	n	n	n	n	n	n	n	n	n	n

TAB. 4.10 – *Un quart des cellules initialement en zéro, 3 excitateurs*

n \a	1	2	3	4	5	6	7	8	9	10	11	12	13
4	n												
5	n	n											
6	n	n	n										
7	n	n	n	n									
8	n	n	n	n	n								
9	n	n	n	n	n	n							
10	n	n	n	n	n	n	n						
11	n	n	n	n	n	n	n	n					
12	n	n	n	n	n	n	n	n	n				
13	n	n	n	n	n	n	n	n	n	n			
14	n	n	n	n	n	n	n	n	n	n	n		
15	n	n	n	n	n	n	n	n	n	n	n	n	
16	n	n	n	n	n	n	n	n	n	n	n	n	n

TAB. 4.11 – *La moitié des cellules initialement en zéro, 3 excitateurs*

n \a	1	2	3	4	5	6	7	8	9	10	11	12	13
4	n												
5	n	n											
6	n	n	n										
7	n	n	n	n									
8	n	n	n	n	n								
9	n	n	n	n	n	n							
10	n	n	n	n	n	n	n						
11	n	n	n	n	n	n	n	n					
12	n	n	n	n	n	n	n	n	n				
13	n	n	n	n	n	n	n	n	n	n			
14	n	n	n	14:16	n	n	n	n	n	n	n		
15	n	n	n	n	n	n	n	n	n	n	n	n	
16	n	n	n	n	n	n	n	n	n	n	n	n	n

TAB. 4.12 – *Trois quarts des cellules initialement en zéro, 3 excitateurs*

Nous pouvons faire plusieurs remarques sur ces tableaux :

- Le nombre de trajectoires de période n augmente rapidement avec le nombre d'excitateurs. En fait, la plupart des trajectoires de périodes différentes de n adviennent lorsqu'il n'y a qu'un seul excitateur. En effet, l'inégalité 4.6 ($qa \leq kn \leq q(a + n_e)$, avec q la longueur d'un cycle n-périodique, et k un entier) devient alors une égalité. Dés que l'on a au moins deux excitateurs, il est alors beaucoup plus facile de construire un cycle n-périodique court.
- Moins la condition initiale est dense, plus il y a de trajectoires de période différente de n.
- Pratiquement toutes les trajectoires simulées sont pérennes, sauf lorsque a est égal à 1, et que la condition initiale est peu dense. Il semble que l'existence des états jeunes favorise la pérennité des trajectoires.

Nous avons évoqué ici le rôle joué par les cycles auto-excitateurs n-périodiques. Nous allons maintenant mettre en évidence leur importance.

4.4.2 Influence des cycles auto-excitateurs n-périodiques

Nous allons reprendre ici les expériences précédentes, dans le but de mettre en évidence l'influence des cycles auto-excitateurs n-périodiques dans la dynamique de l'automate de type AVN. Lorsque l'on a un seul excitateur, on peut calculer la longueur minimum d'un cycle auto-excitateur n-périodique (voir section 4.3.3). Nous allons donc maintenant comparer ces valeurs, avec les périodes observées, dans le cas d'un seul excitateur.

Le tableaux 4.13 représente les valeurs calculées des q_{min}. Dans le tableaux 4.14, nous avons repris le tableaux présenté précédemment, rapportant les périodes observées sur des séries de dix expériences, avec un seul excitateur. Les cellules inscrites en blanc sur noir représentent les cas où q_{min} est supérieur ou égal à douze. Nous pouvons observer que cela représente une assez bonne prédiction. Pour être plus précis, nous pouvons calculer l'erreur de prédiction du critère, par rapport à la longueur de cycle choisie. Soit q la longueur du plus petit cycle possible choisie. Nous considérerons comme erreur tous les cas tels que :

- $q_{min} \geq q$, et toutes les trajectoires observées sont de période n.
- $q_{min} < q$, et au moins une des trajectoires observées est de période différente de n.

La figure 4.9 représente ces valeurs.

n \a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	2														
3	6	6													
4	4	2	4												
5	10	10	10	10											
6	6	6	2	6	6										
7	14	14	14	14	14	14									
8	8	4	8	2	8	4	8								
9	18	18	6	18	18	6	18	18							
10	10	10	10	10	2	10	10	10	10						
11	22	22	22	22	22	22	22	22	22	22					
12	12	6	4	6	12	2	12	6	4	6	12				
13	26	26	26	26	26	26	26	26	26	26	26	26			
14	14	14	14	14	14	14	2	14	14	14	14	14	14		
15	30	30	10	30	6	10	30	30	10	6	30	10	30	30	
16	16	8	16	4	16	8	16	2	16	8	16	4	16	8	16

TAB. 4.13 – *Présentation des q_{min}.*

n\a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	n														
3	n	n													
4	n	n	n												
5	n	5:16	n	n											
6	n	n	n	n	n										
7	n	7:44	24:30	n	7:150	36:36									
8	n	n	8:18	n	n	n	8:42								
9	n	9:40	n	20:72	n	n	28:28	48:96							
10	n	n	10:12	32:64	n	n	10:42	10:176	54:108						
11	11:12	12:24	12:12	24:80	40:50	n	56:56	80:80	36:450	80:80					
12	n	n	n	n	40:80	n	n	n	n	n	88:88				
13	13:14	28:40	30:42	40:40	30:70	48:60	13:14	80:160	54:108	40:40	66:66	96:96			
14	n	n	14:30	14:104	30:60	48:96	n	n	14:72	60:100	44:44	72:72	104:104		
15	16:16	16:16	n	16:16	n	48:48	56:70	16:16	n	n	110:110	48:48	78:78	112:140	
16	n	n	16:36	n	50:50	16:84	56:70	n	16:162	n	66:66	n	52:182	84:84	120:150

TAB. 4.14 – *États initiaux équirépartis, 1 excitateur*

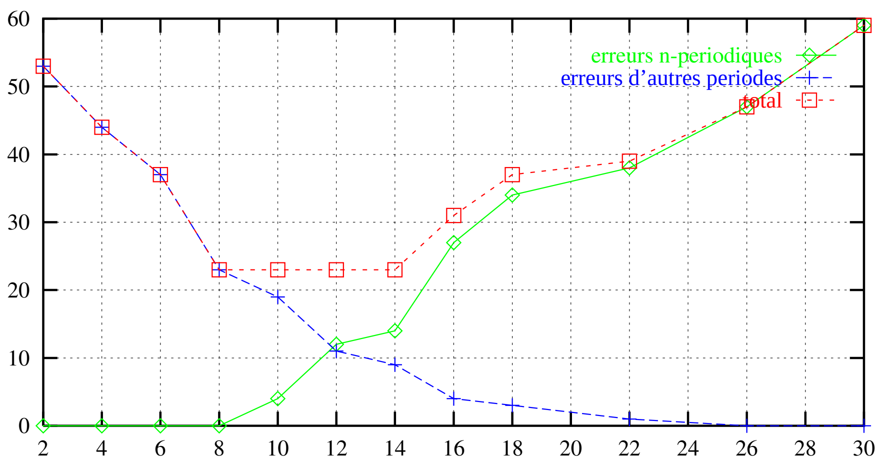


FIG. 4.9 – *Influence des cycles auto-excitateurs n-périodiques*

Nous pouvons constater que le nombre d'erreurs minimum, 23 erreurs, soit à peu prés 19%, est atteint pour q variant de 8 à 14.

4.4.3 Influences de la taille du terrain sur les cycles n-périodiques.

L'idée motivant cette série d'expériences est celle que lorsque le terrain s'agrandit, des cycles longs ont plus de chances de se former. Nous allons donc choisir des cas pour lesquels les cycles n-périodiques ont du mal à se former (c'est-à-dire avec $qmin$ grand, et un seul excitateur), et nous allons calculer pour chacun 50 trajectoires, issues de 50 conditions initiales différentes de même type (états initiaux équirépartis).

$a = 3$, $v = 4$, $n = 9$: $qmin = 6$		
taille	période n	autre périodes
16	62 %	12, 30, 54
32	98 %	30
64	100 %	
128	100 %	
256	100 %	

$a = 3$, $v = 4$, $n = 8$: $qmin = 8$		
taille	période n	autre périodes
16	2 %	9, 18, 30, 42, 66
32	16 %	18, 42, 60
64	36 %	18, 36, 42, 60
128	94 %	18
256	100 %	

$a = 4$, $v = 5$, $n = 7$: $qmin = 14$		
taille	période n	autre périodes
16	6 %	8
32	74 %	8
64	100 %	
128	100 %	
256	100 %	

$a = 9$, $v = 10$, $n = 16$: $qmin = 16$		
taille	période n	autre périodes
16	0 %	18
32	0 %	18, 81, 162
64	16 %	18, 81, 162, 180
128	60 %	162
256	100 %	

$a = 5$, $v = 6$, $n = 9$: $qmin = 18$		
taille	période n	autre périodes
16	0 %	10
32	14 %	10
64	82 %	10
128	100 %	
256	100 %	

$a = 4$, $v = 5$, $n = 9$: $qmin = 18$		
taille	période n	autre périodes
16	0 %	24, 32, 40, 48, 56
32	0 %	24, 32, 40, 56, 80
64	0 %	20, 32, 40, 64
128	0 %	20, 32, 40, 64
256	0 %	20, 40, 48, 80

Dans le dernier cas, $qmin$ est égal à 18, comme pour le cas précédent, mais pourtant, la période n n'est atteinte pour aucune des simulations.

Pour les autres cas, notre conjoncture est vérifiée. En effet, plus le domaine est grand, plus on observe de trajectoires n-périodiques. Et pour un domaine de 256^2 cellules, toutes les trajectoires observées ici sont n-périodiques, sauf pour le cas $a = 4$, $v = 5$ et $n = 9$ (voir figure 4.10).

Afin de généraliser la remarque précédente, on peut montrer le résultat suivant : Soit l'automate cellulaire de type AVN, sur un domaine non-borné, et une condition initiale du type suivant

– L'état de chaque cellule du support de la condition initiale est choisi au hasard.

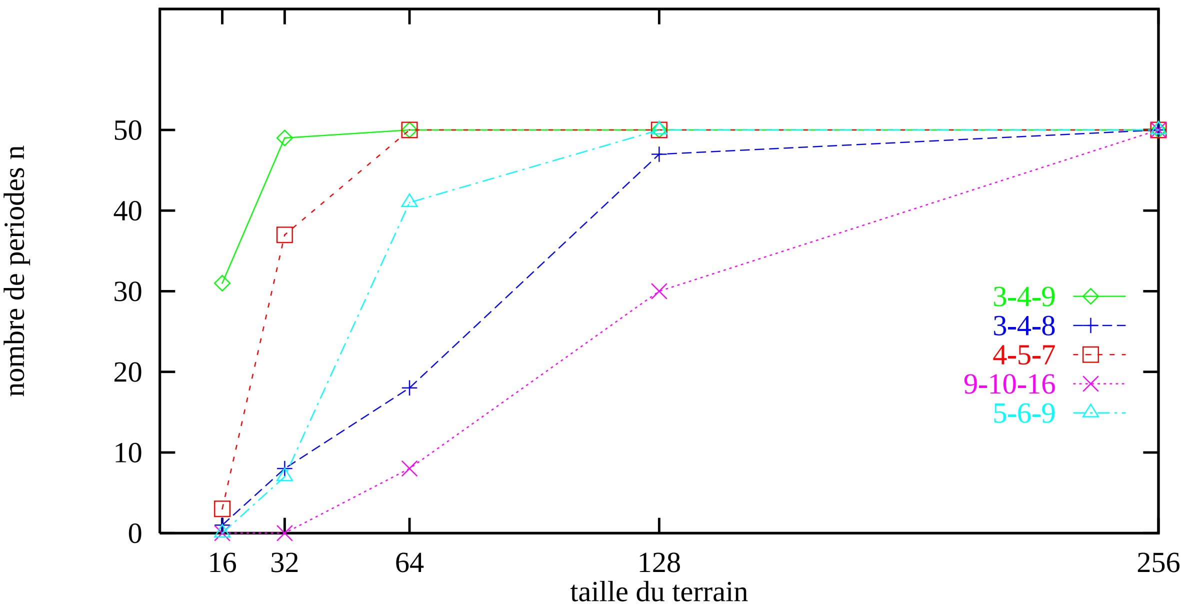


FIG. 4.10 – *Influences de la taille du terrain sur les périodes n*

– La probabilité qu'une cellule contiennent initialement un état donné est non nulle et la même pour chaque cellule.

Lorsque la taille du support de la condition initiale est très grande, la trajectoire obtenue est presque toujours de période n.

Preuve. Soient a, v et n les paramètres de l'automate. On peut calculer q_{min}, qui est la longueur minimale d'un cycle auto-excitateur n-périodique, pour les paramètres a, v et n. On divise le domaine en cycles de cellules de longueur q_{min}, disposées en spirale (figure 4.11). Il existe au moins un cycle auto-excitateur n-périodique de longueur q_{min}. Soit C l'un d'entre eux. Appelons ϵ la probabilité que C soit choisi. Si N est le nombre de cycles tirés au hasard, la probabilité P qu'aucun de ces cycles soit C est $(1-\epsilon)^N$. Lorsque N tend vers l'infini, P tend vers 0. $\square$

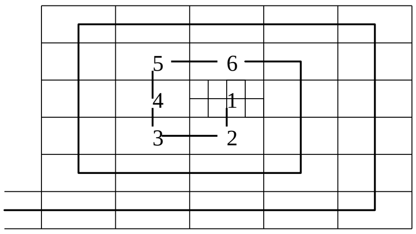


FIG. 4.11 – *On divise le domaine en cycles de cellules de longueur q_{min}, disposées en spirale.*

4.4.4 Les images

Ici, les expériences portent sur l'aspect spatial de l'automate de type AVN. En effet, pour chaque trajectoire simulée, on capture un état du domaine appartenant à son régime périodique. Afin de pouvoir visualiser cet état, nous affectons une couleur à chaque état. Nous adopterons ici les conventions suivantes :

– état neutre : blanc.
– états jeune : dégradé du vert au rouge.
– états adultes : dégradé du rouge au noir.
– états vieux : dégradé du noir au blanc.

De plus, sont associées à cette images les valeurs représentant la période et la taille du transitoire de la trajectoire simulée. Pour des raisons de limite de la mémoire des ordinateurs, nous avons effectué moins d'expériences que précédemment. En effet, chaque image obtenue prend $18Ko$ dans la mémoire de l'ordinateur. Une campagne exhaustive ($n < 16$) comme celle effectuée précédemment aurait donc produit $18.10^3 \times 22400 = 400Mo$ d'images, ce qui est considérable. De plus, cette collection d'images, pour être interprétable, ne doit pas être trop volumineuse. Nous avons donc visité tous les jeux de paramètres avec $n = 15$, $n = 16$ et $n = 17$. Les conditions initiales adoptées seront du type suivant : une proportion z de cellules sera initialement dans l'état neutre, et l'état des cellules restante sera tiré au hasard dans $[0..n-1]$. Pour chaque cas, nous adopterons trois types de conditions initiales, $z = 0$, $z = 2/3$ et $z = 9/10$. Pour chaque configuration, nous simulerons quatre trajectoires, sauf pour $n = 16$ et $z = 0$, où nous simulerons 20 trajectoires. Cela nous donne en tout 7656 images, ce qui représente environ $137Mo$ d'images. Nous ne pouvons pas ici représenter chaque image ! Nous allons donc, à travers quelques exemples, analyser ces images. Les analyser formellement demanderait de développer des outils d'analyse de l'image tels que la Transformée de Fourier, ou le Corrélogramme ([Rip81], [UF85]). Nous nous contenterons d'une analyse informelle des images, nous permettant de mieux comprendre la dynamique de l'automate de type AVN. Pour commencer, voici une collection complète pour $n = 16$ avec $z = 0$ (figures 4.12, 4.13 et 4.14), $z = 2/3$ (figures 4.15, 4.16 et 4.17) et $z = 9/10$ (figures 4.18, 4.19 et 4.20).

A la vue de cet exemple, nous pouvons nous demander quel est le rôle du hasard de la condition initiale dans l'organisation spatiale obtenue. En fait, si les périodes des trajectoires simulées sont les mêmes, les images obtenues seront semblables. Les trajectoires étant presque toujours n-périodiques, nous observerons des différences significatives dans les rares cas (1, 2, ou 3 excitateurs) des trajectoires de périodes différentes. Voici sept exemples, avec pour chacun, cinq hasards différents.

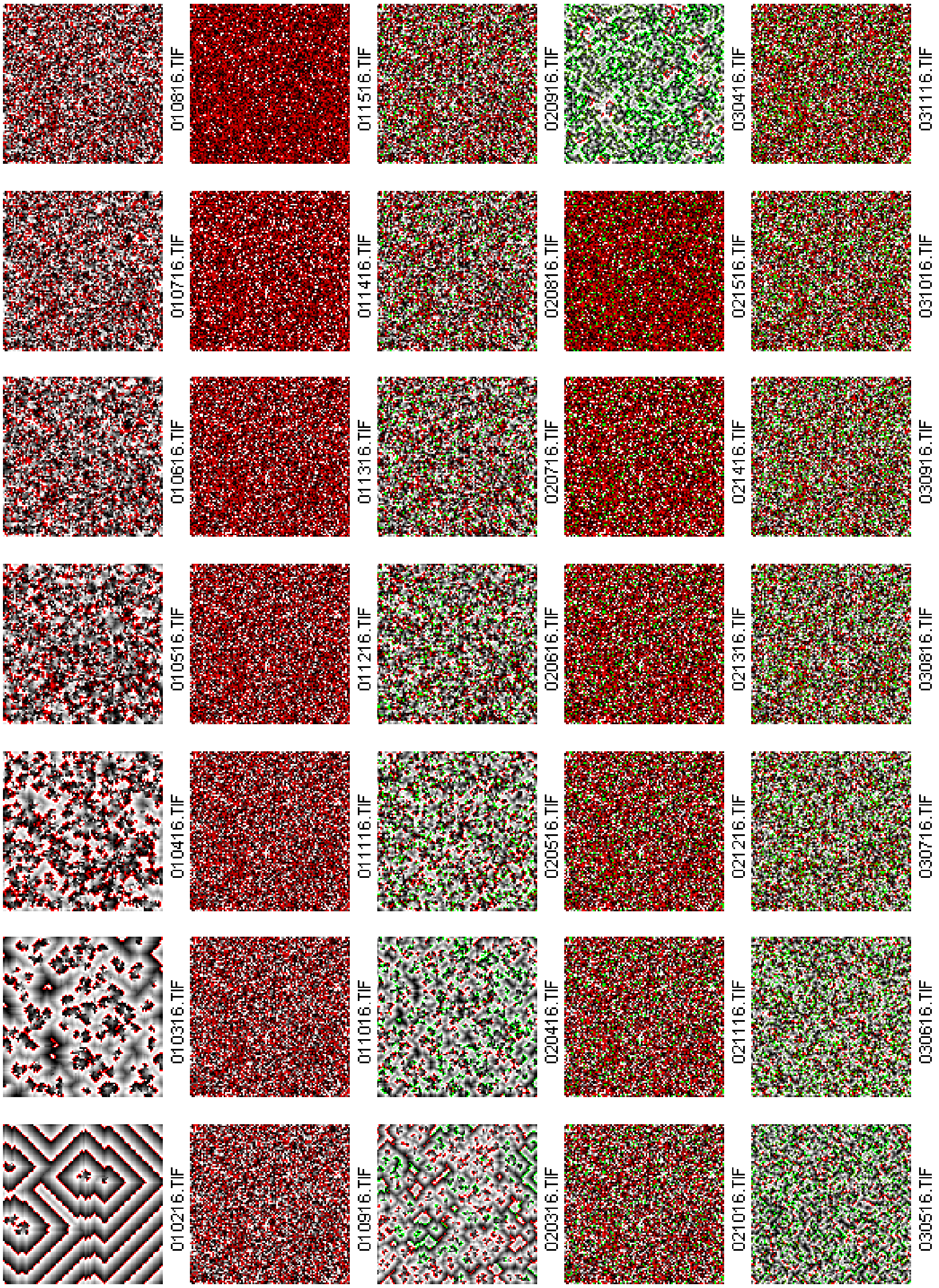


FIG. 4.12 – $n = 16$ *et* $z = 0$ *: première partie*

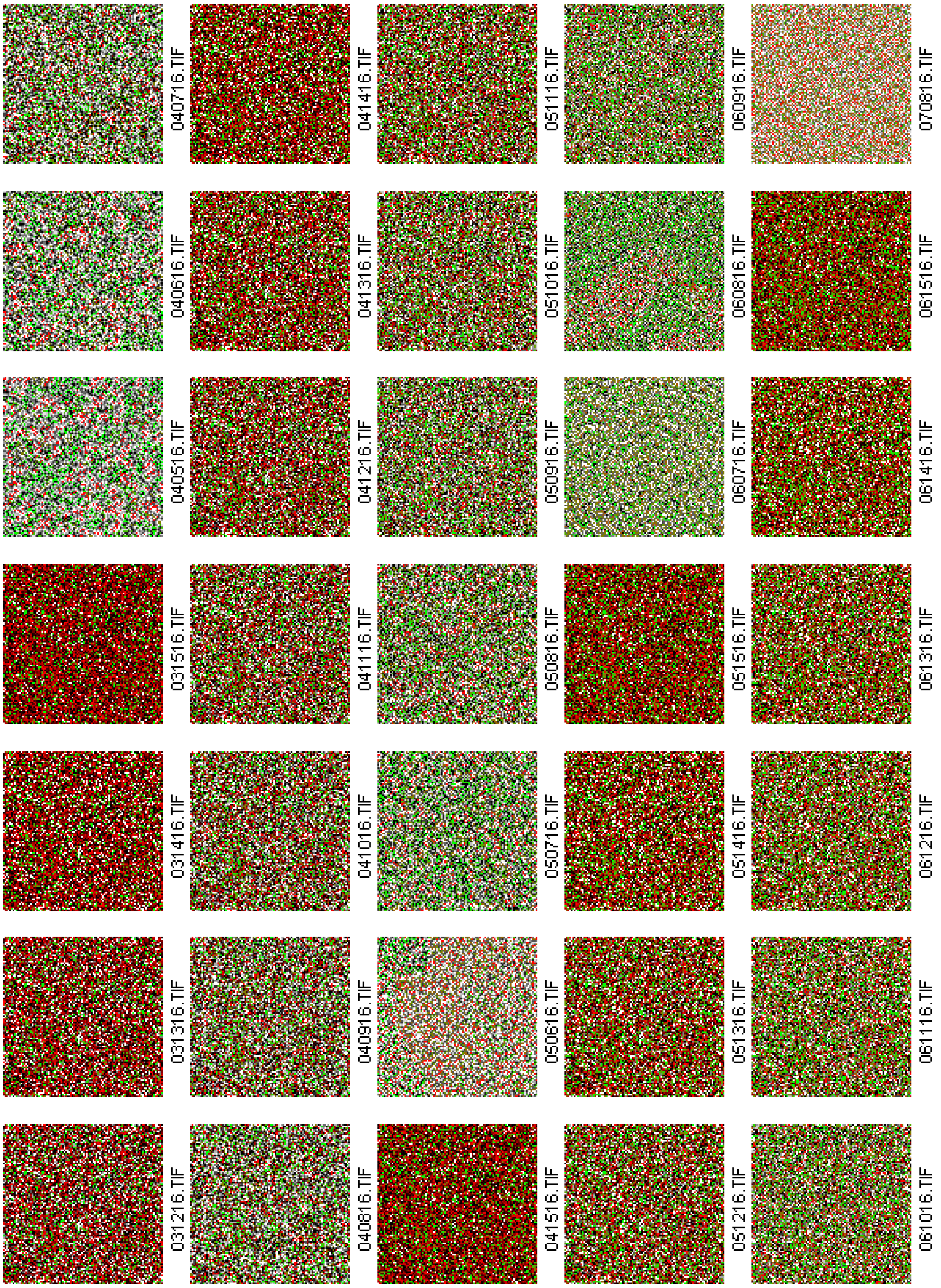


FIG. 4.13 – $n = 16$ *et* $z = 0$ *: deuxième partie*

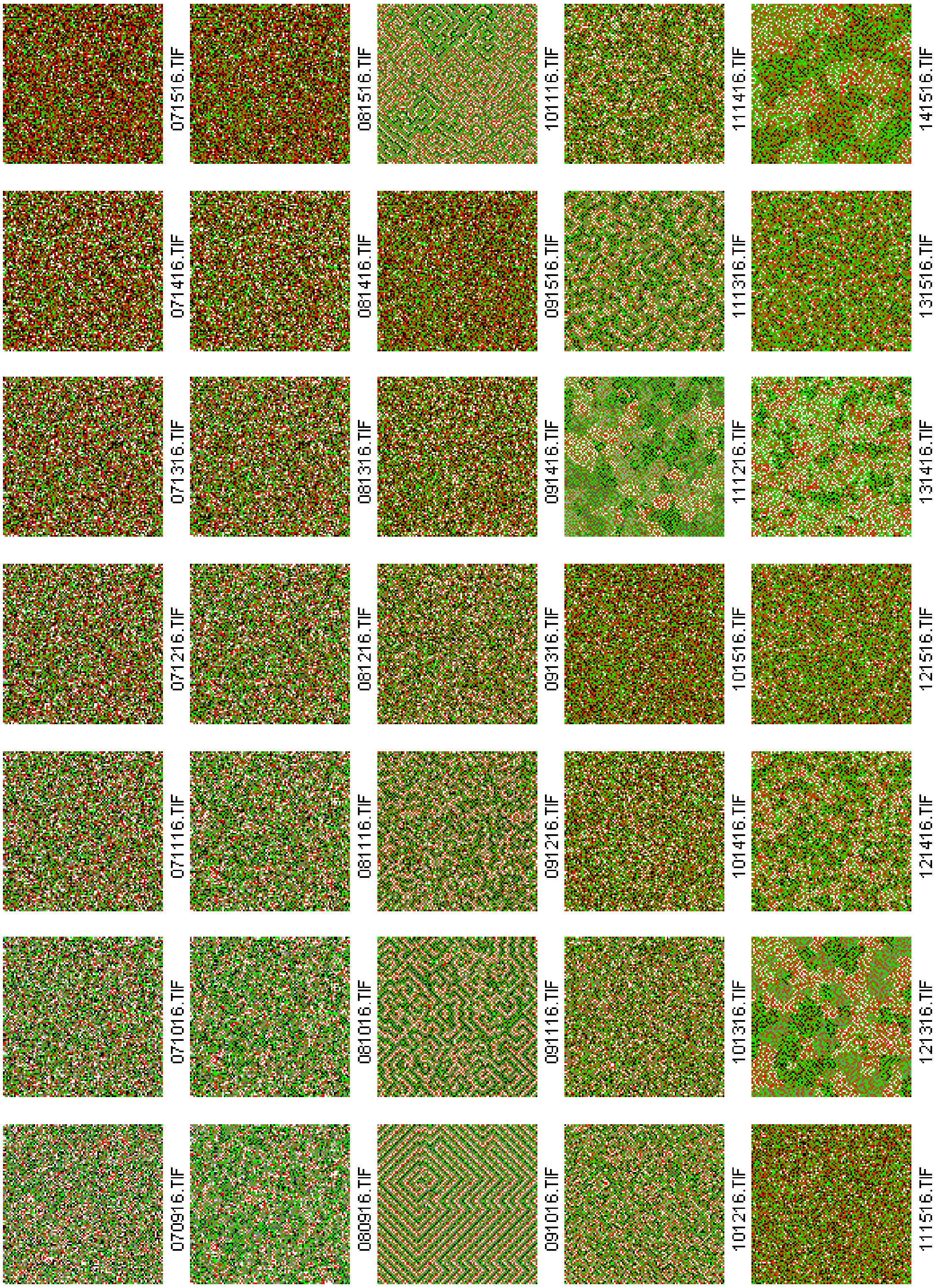


FIG. 4.14 – $n = 16$ *et* $z = 0$ *: troisième partie*

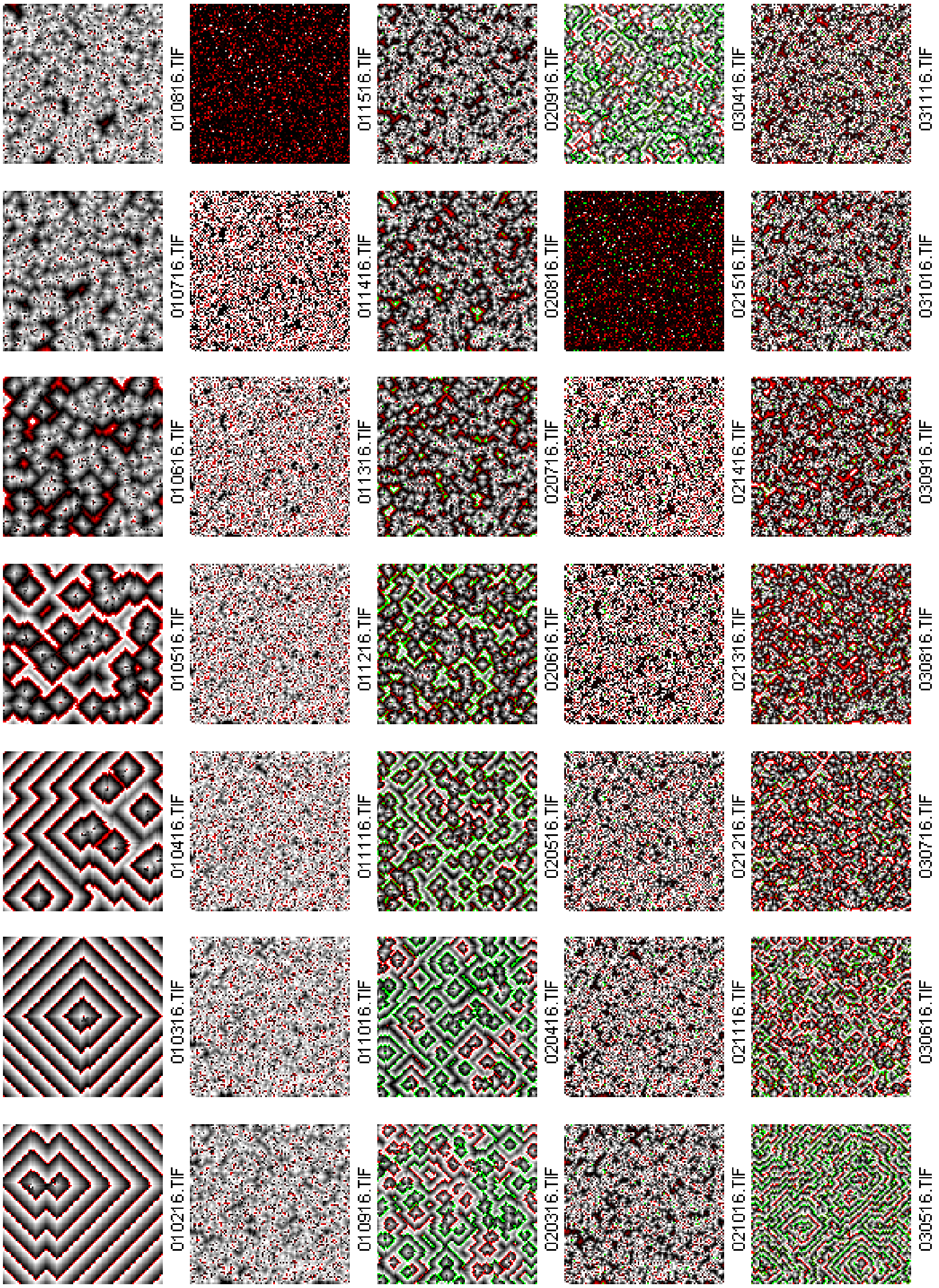


FIG. 4.15 – $n = 16$ *et* $z = 2/3$: *première partie*

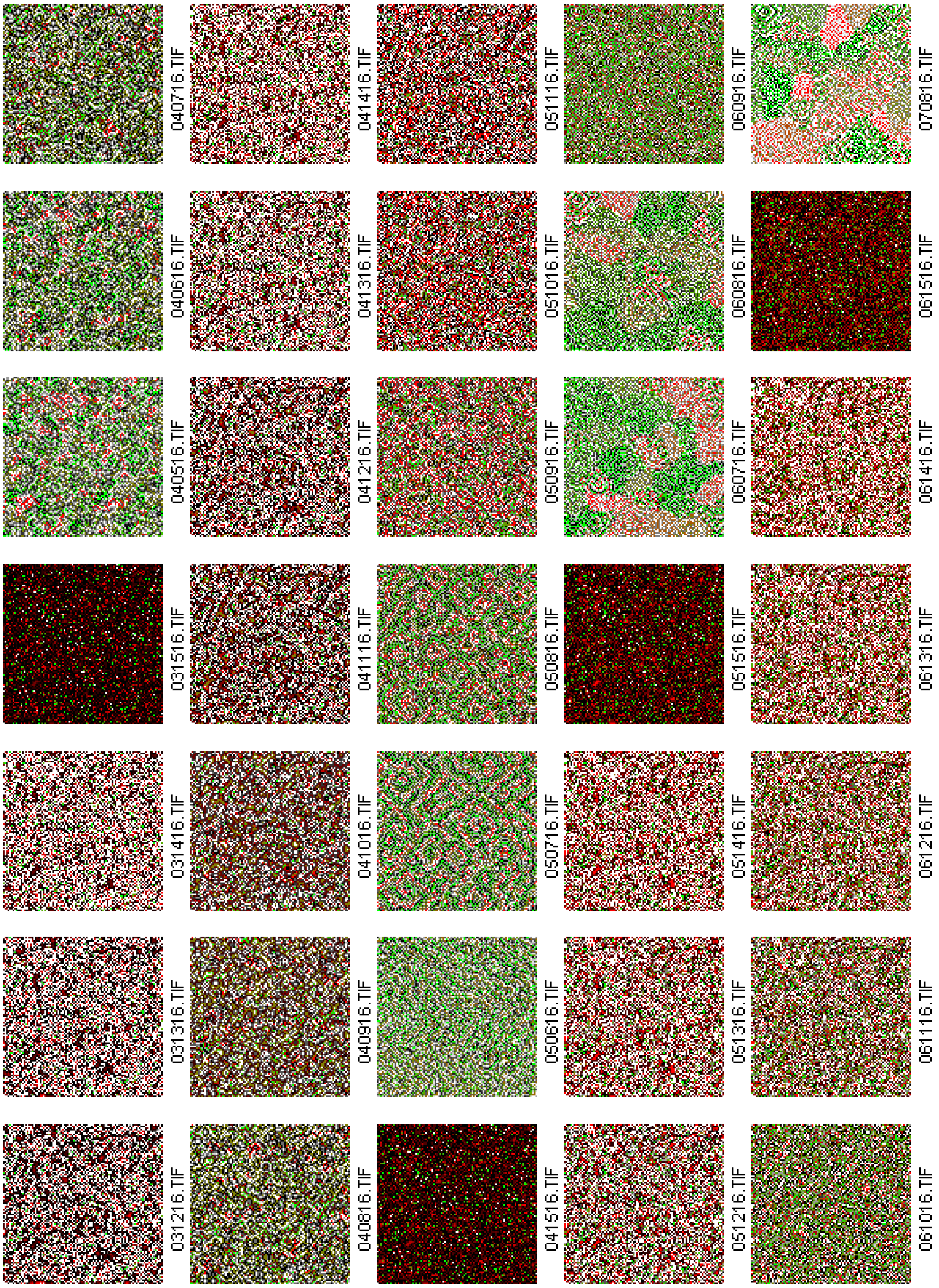


FIG. 4.16 – $n = 16$ *et* $z = 2/3$ *: deuxième partie*

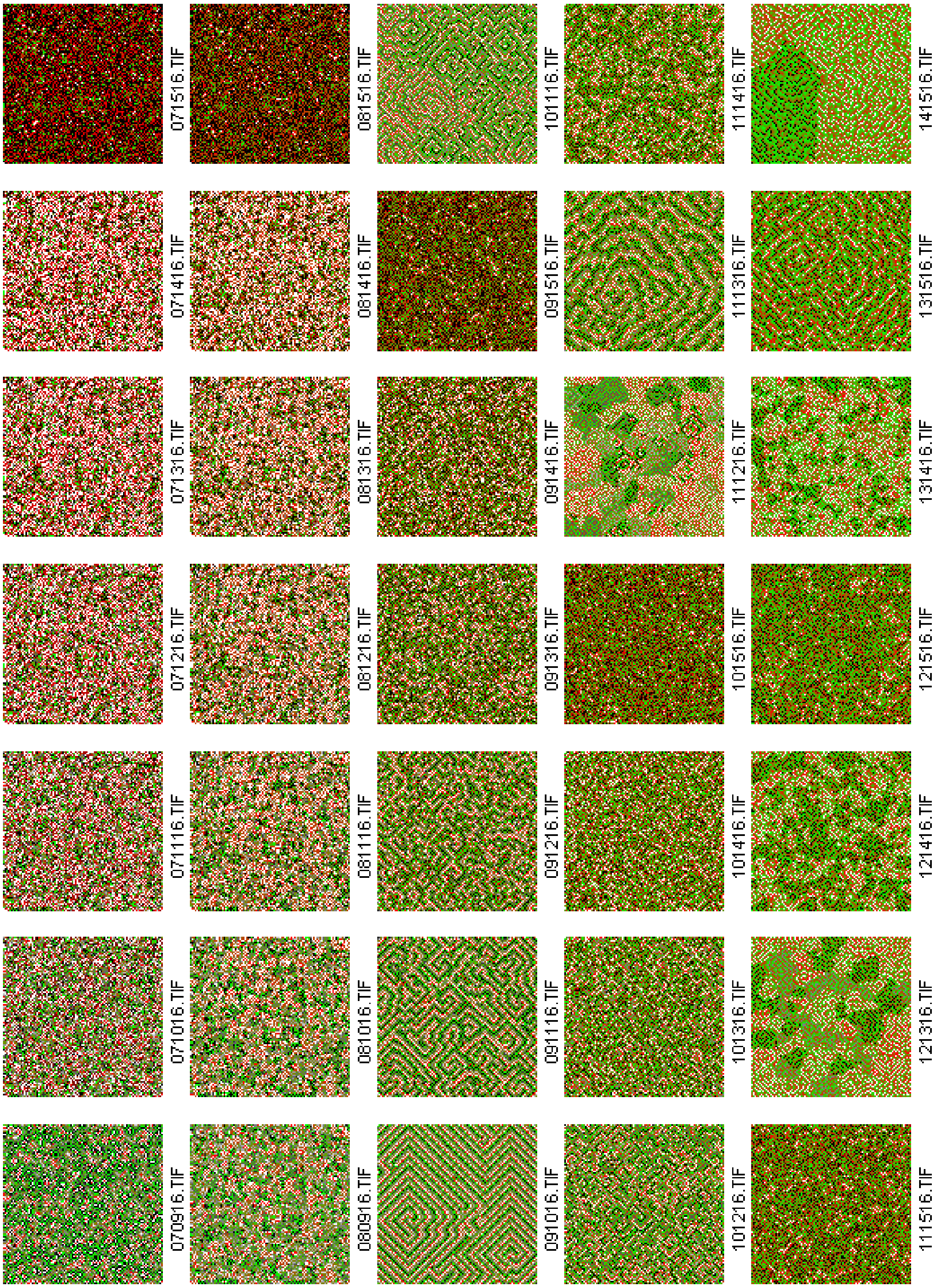


FIG. 4.17 – $n = 16$ *et* $z = 2/3$ *: troisième partie*

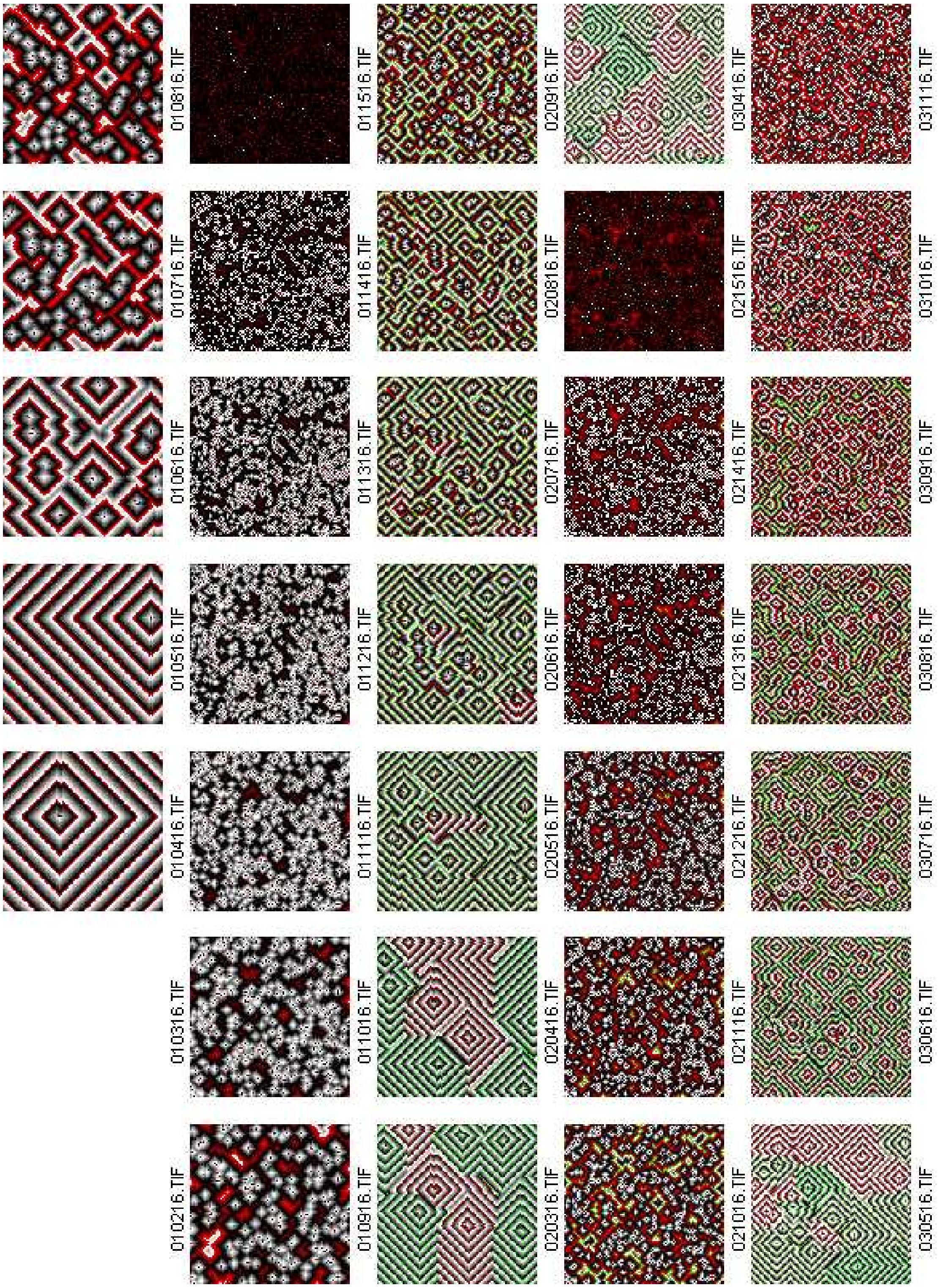


FIG. 4.18 – $n = 16$ *et* $z = 9/10$ *: première partie*

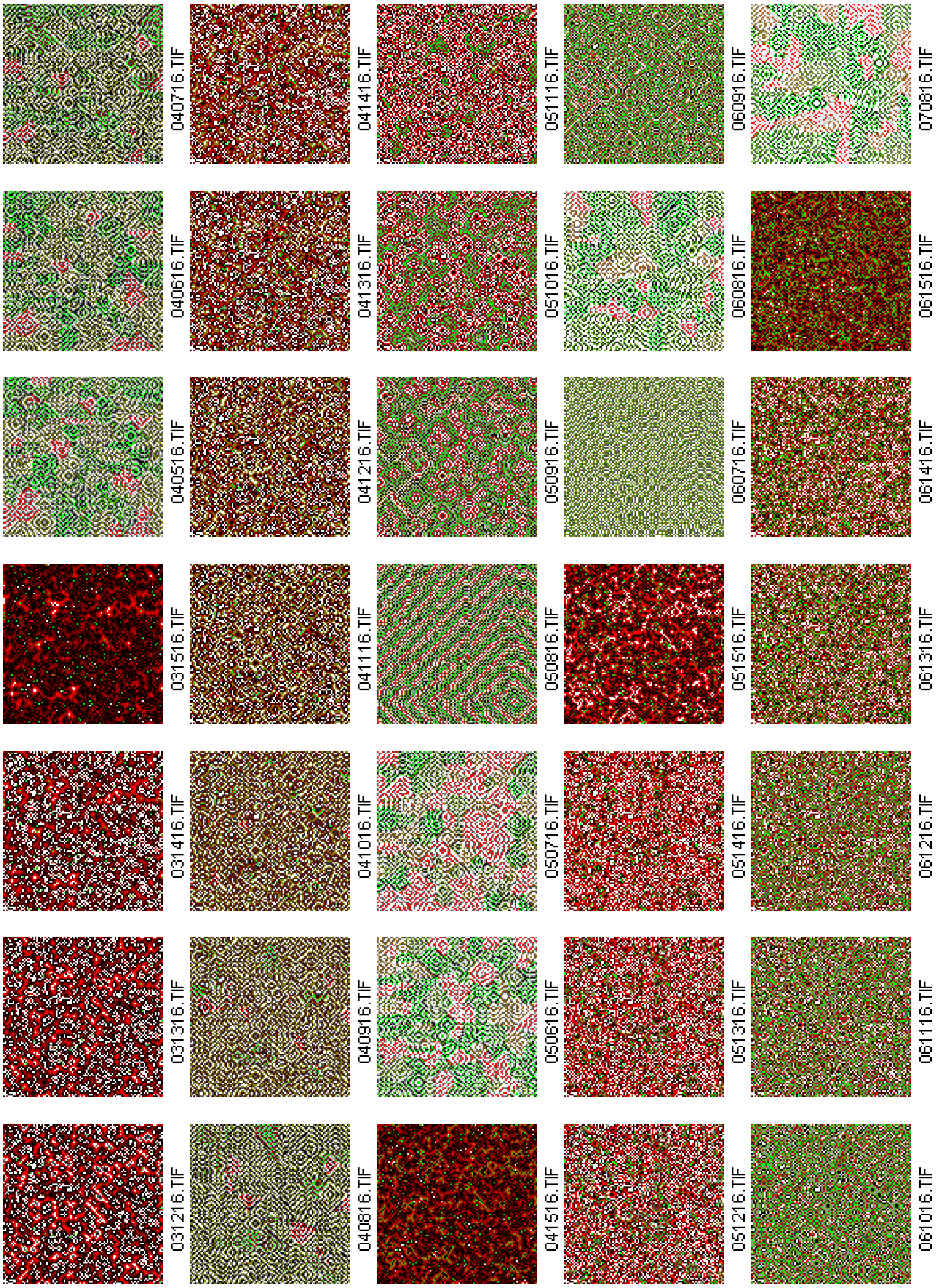


FIG. 4.19 – $n = 16$ *et* $z = 9/10$: *deuxième partie*

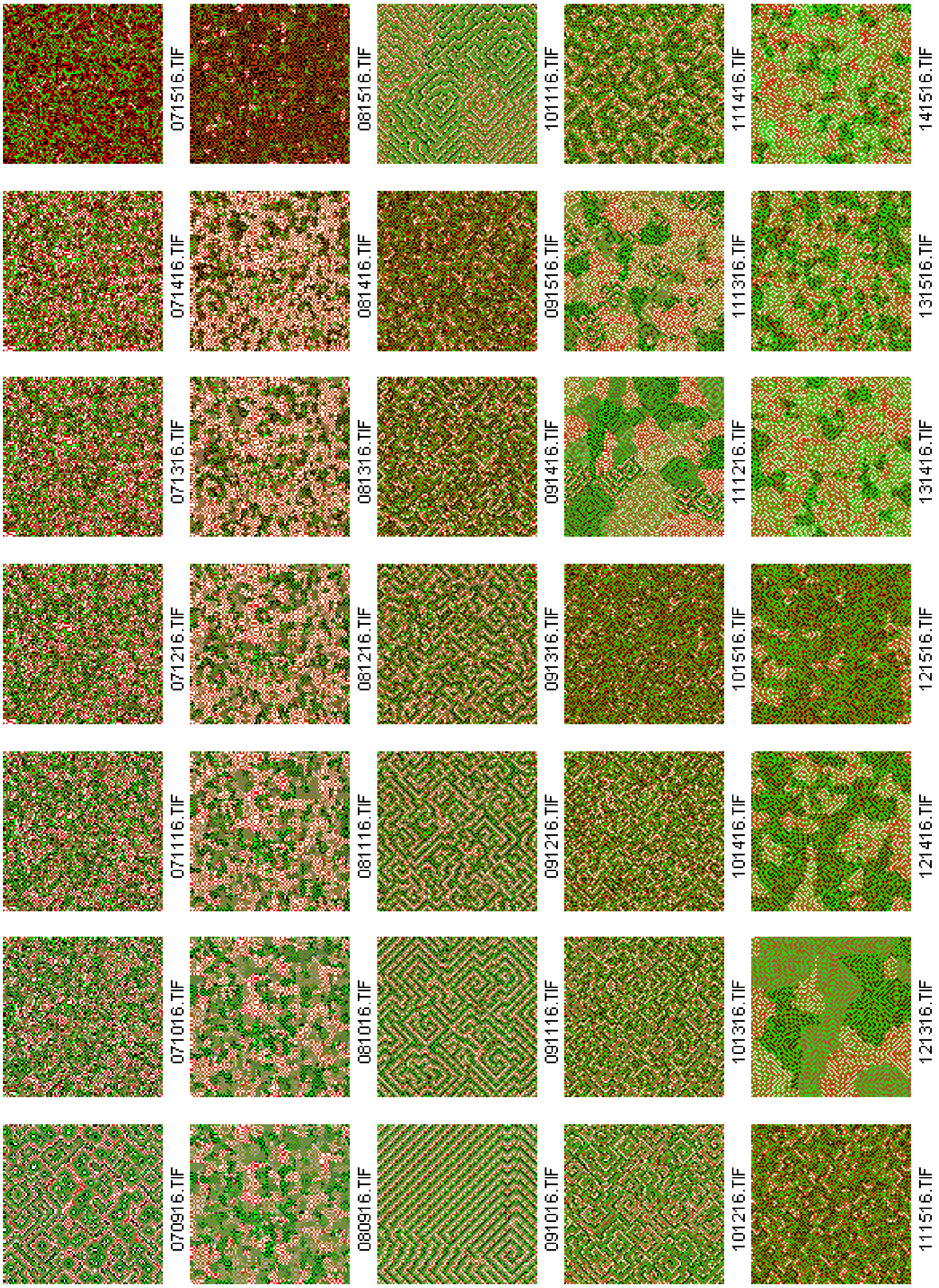


FIG. 4.20 – $n = 16$ *et* $z = 9/10$ *: troisième partie*

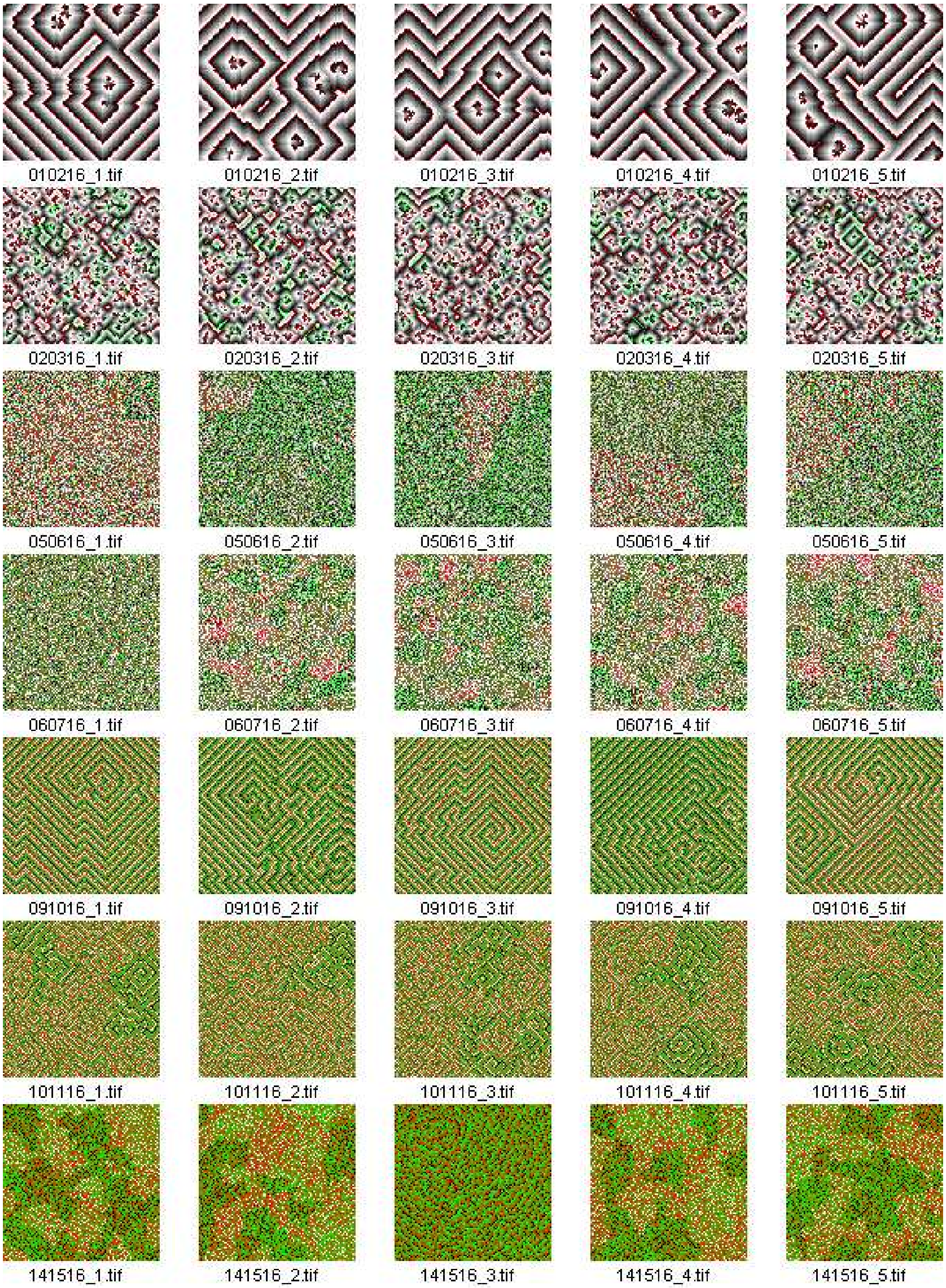


FIG. 4.21 – *Cinq hasard différents pour sept jeux de paramètres différents.*

On peut remarquer que pour chaque série, presque toutes les images sont semblables. Voici un tableaux représentant les périodes des trajectoires d'où sont extraites ces images :

	1	2	3	4	5
$a = 1$	16	16	16	16	16
$a = 2$	16	16	16	16	16
$a = 5$	50	50	50	50	50
$a = 6$	84	36	36	36	36
$a = 9$	16	162	16	162	162
$a = 10$	16	16	16	16	16
$a = 14$	84	84	16	84	84

On constate que les structurations spatiales différentes correspondent à des périodes différentes. Les structurations spatiales issues des trajectoires de mêmes paramètres et de mêmes périodes sont donc comparables. Mais essayons maintenant de mieux comprendre le lien entre la structuration spatiale observée, et la période d'une trajectoire.

Au cours du temps, la structuration spatiale du domaine va être dominée par l'apparition de cycles périodiques. S'ils sont très nombreux (beaucoup de couples auto-excitateurs n-périodiques, par exemple), l'organisation spatiale du domaine ressemblera au hasard initial, et le transitoire sera très court. Si au contraire, il apparaît très peu de ces cycles (q_{min} grand), le domaine aura alors un aspect très organisé, autour de ces cycles, et le transitoire sera alors très grand. C'est ce que nous montre la figure 4.22, avec deux séries d'images : a prend les valeurs 1 et 9, et v augmente. Ces images sont comparables, car elles proviennent toutes de trajectoires n-périodiques.

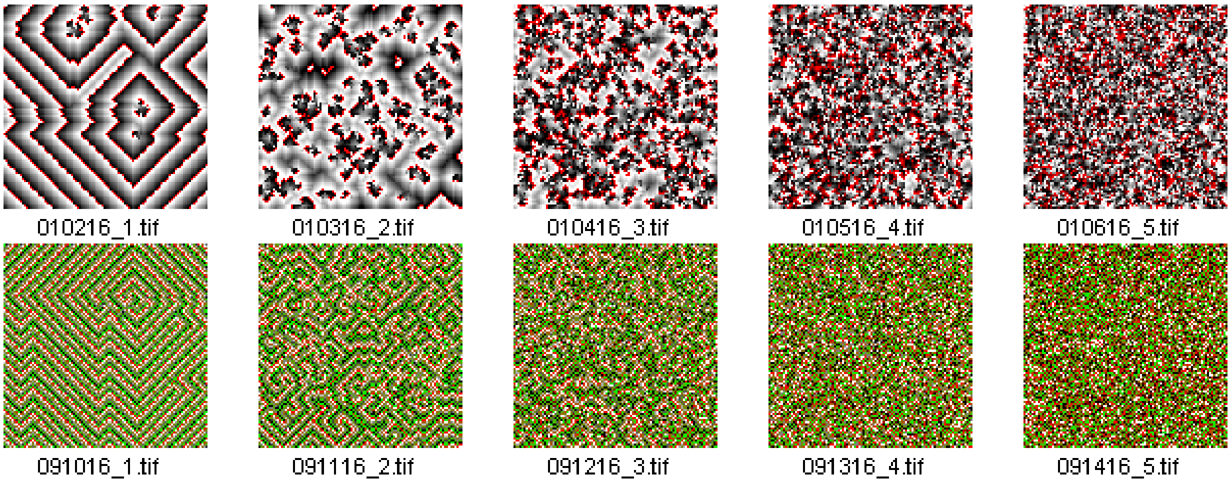


FIG. 4.22 – *Deux séries avec v qui augmente : $a = 1$ et $a = 9$. Quand le nombre d'excitateurs grandi, la structure spatiale se "resserre"*

Le tableau suivant exprime les transitoires correspondant :

nombre d'excitateurs	1	2	3	4	5
$a = 1$	141	56	52	42	35
$a = 9$	3782	705	392	293	164

Finalement, une image du régime périodique d'une trajectoire de notre automate consiste en une partition du domaine en zones d'influence (il peut n'y en avoir qu'une) autour des cycles auto excitateurs persistants. Ces zones sont caractérisées par la présence d'un sous-ensemble de l'ensemble des états, qui se traduit, à l'image, par une "couleur résultante". Dans le cas particulier d'une trajectoire n-périodique, avec un seul excitateur, nous pouvons prévoir que le nombre de "couleurs résultantes" différentes sera $PGCD(n,a)$. En effet, puisqu'il n'y a qu'un seul excitateur, chaque zone voit la succession des états suivants : 0, a, $2a \mod n$, etc ... en tout $\frac{n}{PGCD(n,a)}$ états, ce qui définit $PGCD(n,a)$ sous-ensembles différents de E, l'ensemble des états. En fait, le hasard de la condition initiale produit des zones d'influence dont le comportement est décalé dans le temps, ce qui donne des "couleurs résultantes" différentes. La figure 4.23 montre différentes interprétations d'un état du domaine, pour les paramètres suivants : $a = 10$, $v = 11$ et $n = 16$.

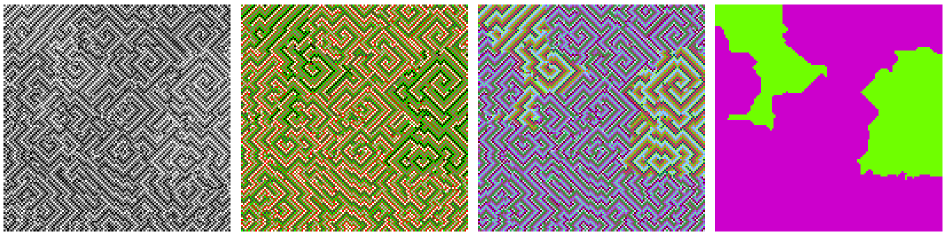


FIG. 4.23 – $a = 10$, $v = 11$ *et* $n = 16$ *: plusieurs palettes de couleurs*

Différentes palettes de couleurs sont utilisées pour représenter le même état du domaine :
- La première est une palette uniforme, du noir vers le blanc : on ne distingue pratiquement pas de zone.
- La seconde est la palette standard : on commence à distinguer deux zones.
- La troisième est une palette tirée au hasard : deux zones se dessinent clairement.
- La dernière palette est bicolore : les états pairs en mauve, les impairs en vert. Cette expérience met en évidence la propriété énoncée plus haut : $PGCD(10,16) = 2$; nous avons donc deux "couleurs résultantes" différentes.

Cet exemple met en évidence le fait que les couleurs que nous affectons à chaque état afin de visualiser un état du domaine jouent un grand rôle. Il nous faut donc nous méfier de l'interprétation que nous choisissons.

4.4.5 Robustesse

Nous allons dans cette section tester rapidement la robustesse des structurations spatiales observées, en ajoutant aux règles de l'automate cellulaire AVN, un aspect stochastique. Nous pourrions faire intervenir le hasard de plusieurs manières; nous choisirons ici de la faire agir sur le recrutement, de la manière suivante : la nouvelle règle, que nous appellerons *AVN-zéro*, sera basée sur la règle AVN, et la passage de 0 en 1 si l'une des voisines est adulte, se produira avec la probabilité p. Précisons en pseudo langage de programmation ce que nous entendons par là. On note W la variable aléatoire de Bernoulli de probabilité p ($P(W = 1) = p; P(W = 0) = 1 - p$) :

`si` $C_{i,j}(t) = 0$
`alors`
  `si` {"Un voisin au moins ce $C_{i,j}$ est adulte" `et` $W = 1$}
  `alors` $C_{i,j}(t+1) = 1$
  `sinon` $C_{i,j}(t+1) = 0$

Il peut apparaitre deux types de structuration spatiale, l'une due aux paramètres et l'autre induite par une condition initiale peu dense. Nous allons donc, pour effectuer nos tests, utiliser :

- deux conditions initiales : La proportion p de cellules neutres égale à 0, et 0,99,
- et deux jeux de paramètres, $\{a = 8,\ v = 9,\ n = 16\}$ et $\{a = 9,\ v = 10,\ n = 16\}$.

La figure 4.24 représente les deux conditions initiales, et des états des phases périodiques pour les deux jeux de paramètres.

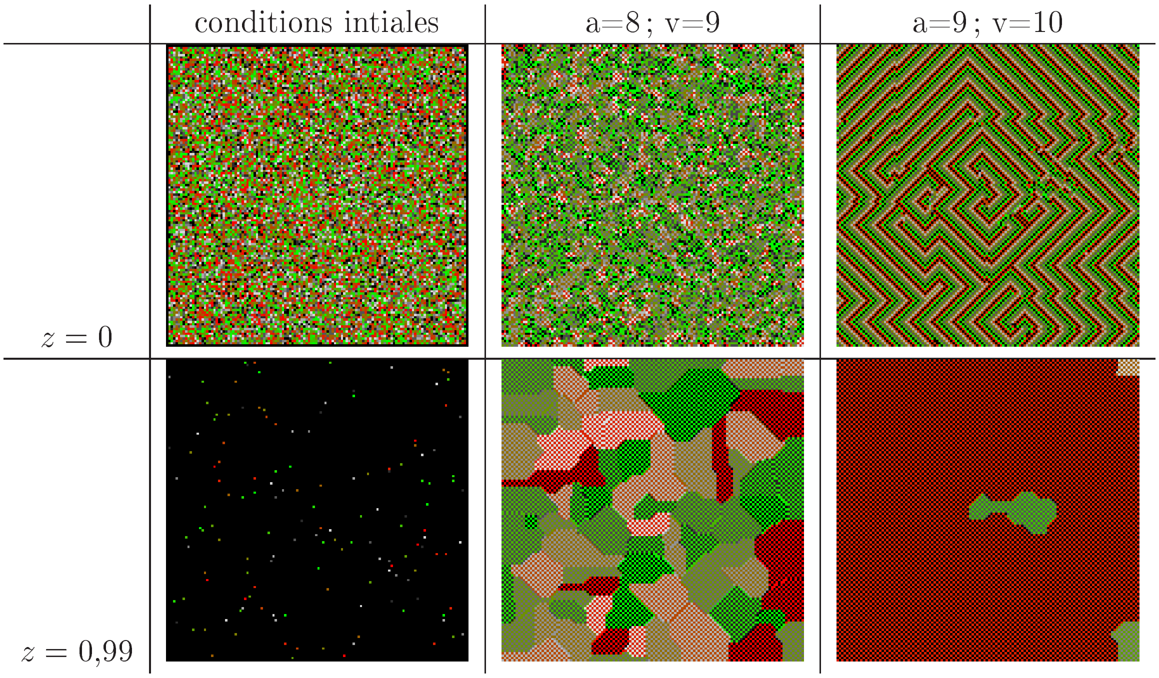


FIG. 4.24 – *Deux conditions intiales appliquées à deux jeux de paramètres.*

Nous pouvons maintenant comparer ces figures avec celles issues des expériences *"brui-*

tées". Mais en ce qui concerne les automates cellulaires stochastiques, on ne peut observer au plus qu'une *"pseudo période"*, et on ne peut, sans travail mathématique plus poussé, prévoire l'avenir des trajectoires à long terme. Nous conviendrons donc d'observer les images correspondant à l'itération 5000 des trajectoires simulées, ce qui semble correspondre à une certaine stabilité. Trois séries d'expériences seront menées, la probabilité de passage de 0 en 1 prenant successivement les valeurs 0.99 (figure 4.25), 0.95 (figure 4.26) et 0.8 (figure 4.27). Au delà de cette valeur, le hasard prend le pas sur la dynamique propre de l'automate, et on ne peut alors plus parler de robustesse.

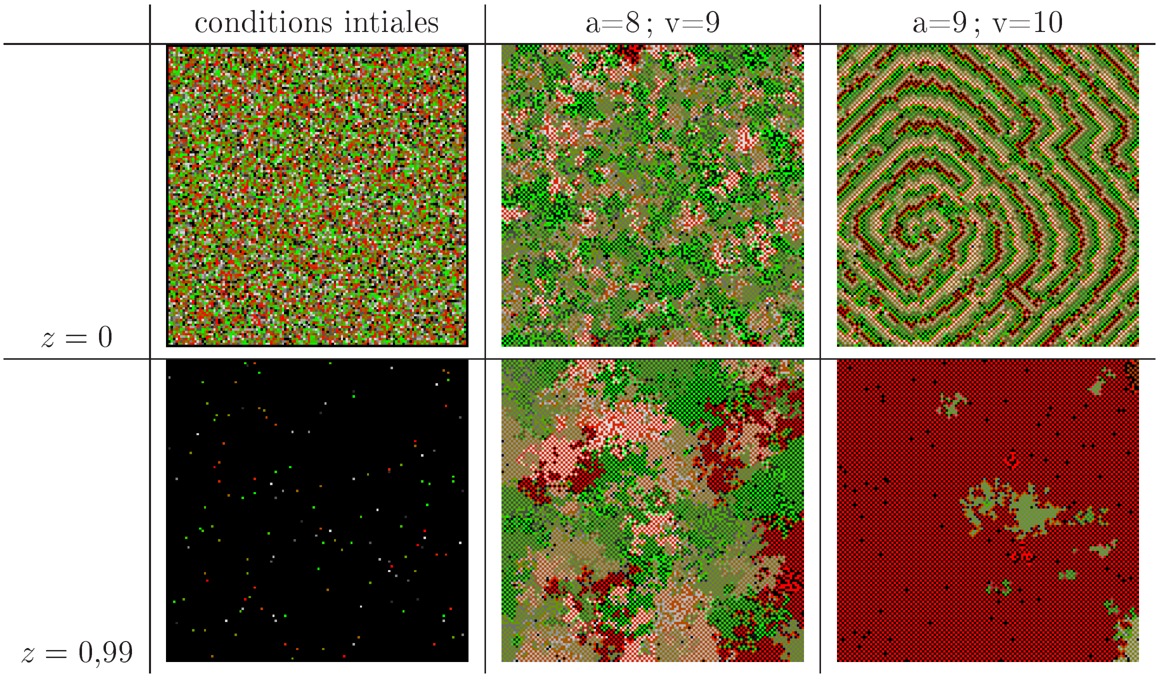


FIG. 4.25 – *La proportion de passage de* 0 *en* 1 *est* 0.99.

Nos observations sont les suivantes :

- $\{a = 8,\ v = 9\}$, condition initiale dense : Il semble que l'introduction du hasard ait ici un effet légèrement structurant; en effet, la taille des "zones" observées semble grandir avec p.
- $\{a = 8,\ v = 9\}$, condition initiale claire : La structuration spatiale en "tâches" se "brouille" de plus en plus. Mais il subsiste une structuration de l'espace.
- $\{a = 9,\ v = 10\}$, condition initiale dense : On observe une structure en spirale, qui est conservée pour $p = 0.99$, $p = 0.95$, mais qui se perd ensuite.
- $\{a = 9,\ v = 10\}$, condition initiale claire : Le même aspect assez uniforme est conservé, sauf pour $p = 0.8$.

En règle générale, l'introduction du hasard "arrondi les angles"; en effet, les "frontières" sont moins nettes, et les structures sont plus "rondes". Quand p augmente, les effets dûs à la différence de condition initiale ont l'air de s'estomper, et les images tendent vers un

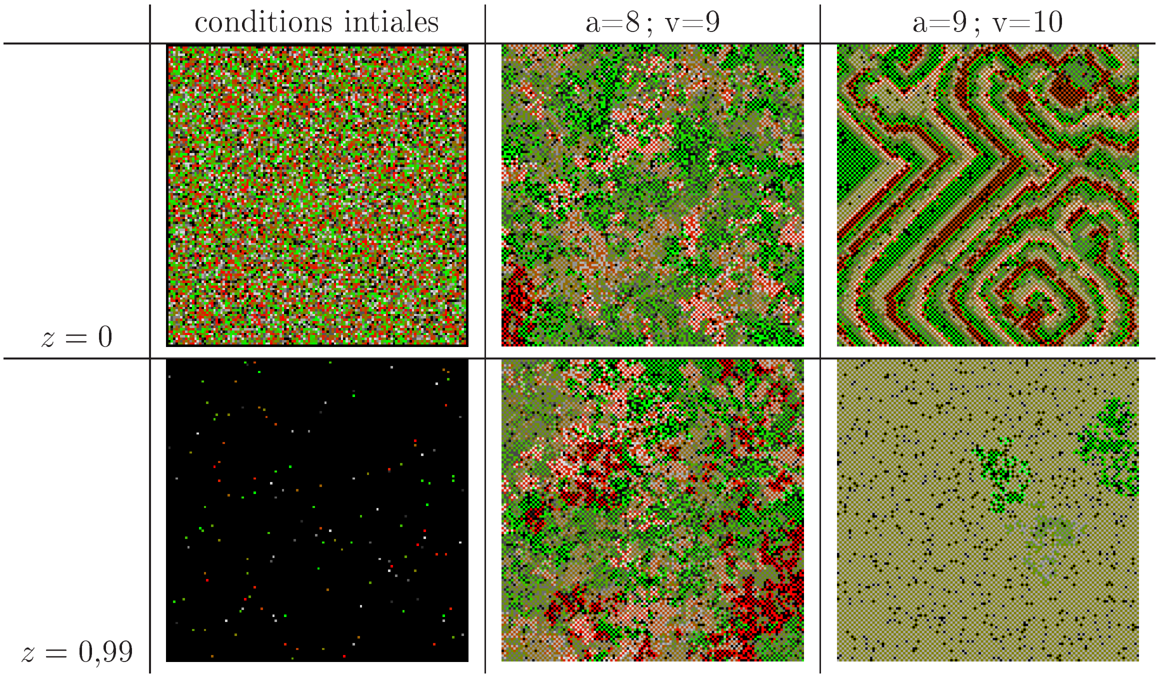


FIG. 4.26 – *La proportion de passage de* 0 *en* 1 *est* 0.95.

même aspect. Par contre, les structurations observées restent différentes pour les deux jeux de paramètres.

L'introduction d'un "bruit" dans les règles d'un automate cellulaire semble un travail interressant, notamment en ce qui concerne le "gommage" de l'"effet géométrique", ainsi que l'étude de la résistance au bruit d'une structure observée. Ce travail demanderait cependant beaucoup de temps pour être systématisé. Pour cela, il nous semblerait utile de définir un indicateur systématique et objectif de la struturation spatiale des images observées. Mais ce travail reste à faire.

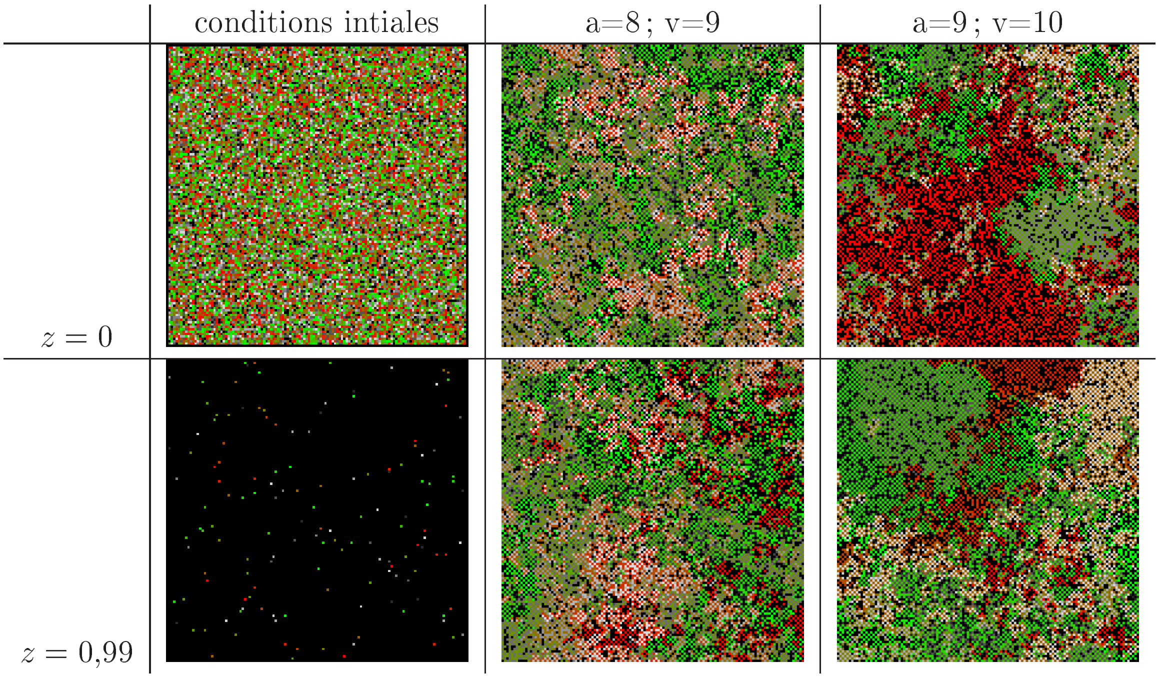


FIG. 4.27 – *La proportion de passage de 0 en 1 est 0.80.*

Chapitre 5

Vers un modèle non-spatialisé

Dans le chapitre , nous nous sommes convaincus de la nécéssité d'une modélisation à caractère spatial pour un phénomène non-homogène. Nous nous interrogeons ici sur les conditions qui pourraient tout de même mettre en cause la pertinence d'un tel choix.

Dans le chapitre 4, nous avons étudié l'automate cellulaire de type AVN pour une grande variété de valeurs possibles de a, v et n, et nous avons montré que ses propriétés dépendent de façon cruciale de certaines propriétés arithmétiques des entiers a, v et n. Dans ce chapitre, nous poursuivons cette étude de la façon suivante :

- Nous fixons $n = 16$, et des valeurs aux paramètres a et v telles que, selon notre étude précédente, les solutions soient de période n.
- Au lieu d'observer les figures produites sur la grille par l'automate, nous ne nous intéressons qu'à l'évolution au cours du temps du nombre de zéros, c'est-à-dire de la place libre dans l'image de la forêt. En particulier, si la proportion de cellules en l'état zéro devient 100%, cela signifie la disparition de la forêt.

Nous conduisons des expériences de simulations systématiques dont nous testons la robustesse à l'introduction d'un bruit dans la dynamique. Pour terminer, nous nous posons la question de l'existence d'un modèle non spatialisé de type Usher ([Fav95]), donc plus simple, qui serait capable de reproduire l'évolution du nombre de zéros du modèle spatialisé.

5.1 Conventions

Les expériences que nous mèneront ici auront toutes quelques points en commun :
- la taille du terrain, soit 128×128 cellules.
- le type de conditions initiales. Elles seront composées d'une bordure d'une cellule de large dans l'état neutre (afin d'éviter les problèmes de bord), d'une proportion p_z de cellules dans l'état neutre, uniformément réparties dans le terrain, l'état de chaque cellule restante choisi uniformément et indépendemment au hasard entre 1 et $n - 1$.

5.2 Transition Déterministe

Nous considérerons pour commencer, l'effet du paramètre p_z, "proportion de 0" dans la condition initiale, sur les trajectoires de l'automate avec transition déterministe. Afin de simplifier l'étude dans un premier temps, nous choisirons un jeu de paramètres tel que toutes les trajectoires obtenues aboutissent, si possible, à une solution de même période. En imposant aux paramètres a, v et n les conditions suivantes

$$\begin{cases} a \leq n/2 & (5.1) \\ a = n - v + 1 & (5.2) \end{cases}$$

nous avons montré que :

- toute cellule vide excitée forme avec chacune de ses cellules excitatrices un couple n-périodique (voir les théorèmes 4.3.4 et 4.3.6)
- et que, de plus, une trajectoire qui contient à un instant un tel couple aboutit à une solution n-périodique (voir le théorème 4.3.7).

En nous basant sur un nombre d'état égal à 16, nous choisirons donc le plus simple des jeux de paramètres vérifiant les conditions 5.1 et 5.2 : $\mathbf{a = 8}$, $\mathbf{v = 9}$, $\mathbf{n = 16}$.

De plus, nous sommes sûrs que dès qu'une cellule en l'état neutre se trouve dans le voisinage d'une cellule dans un état adulte ou jeune, elle sera excitée :

- Soit par sa voisine, si elle est dans un état adulte.
- Soit par la même voisine, lorsqu'elle aura atteint le premier état adulte.
- Soit par une autre cellule adulte de son voisinage.

Il nous reste donc à nous assurer que chaque trajectoire étudiée contienne à un moment donné une cellule dans un état adulte ou jeune dans le voisinage d'une cellule dans l'état neutre.

Si l'on partage le terrain en couples de cellules, la probabilité pour qu'aucun de ces couples ne soit constitué, dans la condition initiale, d'un zéro et d'un adulte ou un jeune est supérieure à la probabilité qu'il n'existe, sur le terrain, aucune cellule en zéro voisine d'une cellule adulte ou jeune. Un calcul de probabilité élémentaire nous montre donc que dès que $0{,}0015 < p_z < 0{,}9985$, la probabilité pour que la condition initiale ne contiennent pas une cellule vide voisine d'un jeune ou d'un adulte est inférieure à 10^{-6}, ce que nous considérons comme une (quasi)certitude.

Donc, dès que la proportion de 0 dans la condition initiale est un peu supérieure à 0 ou un peu inférieure à 1, nous avons la (quasi)certitude que la trajectoire ne sera jamais identiquement nulle, ou de période différente de 16.

Comme toutes les trajectoires finissent par être de période 16, il est clair que la moyenne sur 16 itérations du nombre de cellules vides finit par être un seizième du nombre total de cellules. En effet, chaque cellule en particulier affiche 0 une fois sur seize. C'est ce que nous observons sur la figure 5.2, qui représente une série de dix expériences, effectuées avec des densités initiales de cellules neutres variant de 0 à 90% ($p_z \in \{0{,}0.1{,}0.2{,}0.3{,}0.4{,}0.5{,}0.6{,}0.7{,}0.8{,}0.9\}$). Pour cette figure nous avons effectué une moyenne mobile de longueur 16 sur le nombre de cellules en l'état neutre observé afin de mettre en évidence le fait que toutes les trajectoires atteignent en moyenne sur 16 itérations, le même nombre de cellules en zéro.

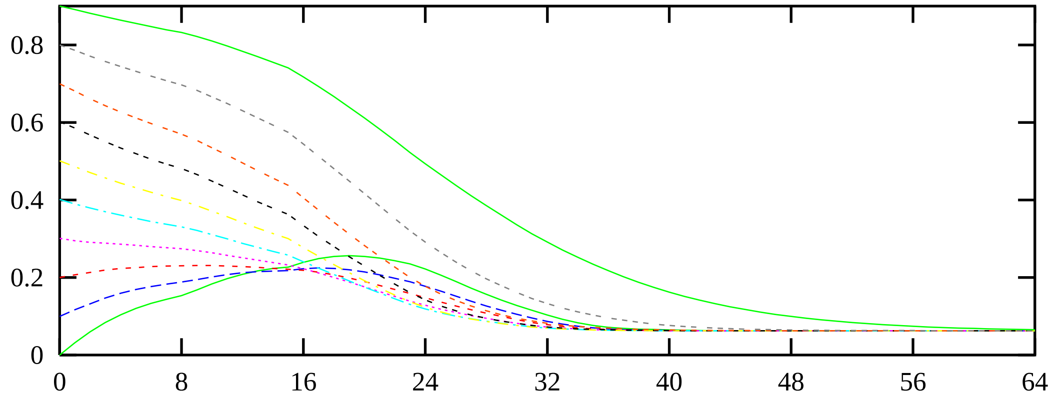


FIG. 5.2 – p_z *varie de* 0 *à* 90%. *Moyenne mobile sur* 16 *itérations.*

S'il est vrai que la proportion de zéros sur seize itérations finit par être égale à 1/16, il n'y a pas de raison qu'il en soit de même à chaque itération. Nous observons l'évolution de ce nombre sur la figure 5.3, qui représente la même expérience que la précédente, sans moyenne mobile.

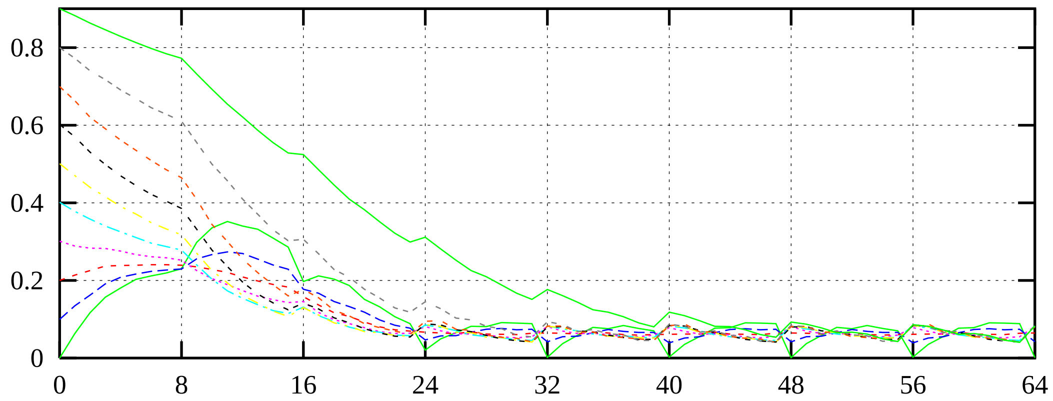


FIG. 5.3 – p_z *varie de* 0 *à* 90%.

Nous remarquons ici que les courbes ont l'air d'être "synchronisées"; en effet, elles présentent toutes leurs pics aux mêmes instants. Afin de tester la dépendance des courbes obtenues par rapport au hasard de la condition initiale, nous effectuons trois séries d'expériences, avec pour chacune dix hasards différents, et une proportion initiale de zéros $p_z = 0$ (figure 5.4), $p_z = 0.5$ (figure 5.5) et $p_z = 0.9$ (figure 5.6).

Les courbes se superposent quasi-parfaitement, montrant l'indépendance, au cours du temps, du nombre de cellules dans l'état neutre par rapport au hasard de la condition

FIG. 5.4 – *Dix hasards différents.*$p_z = 0.0$

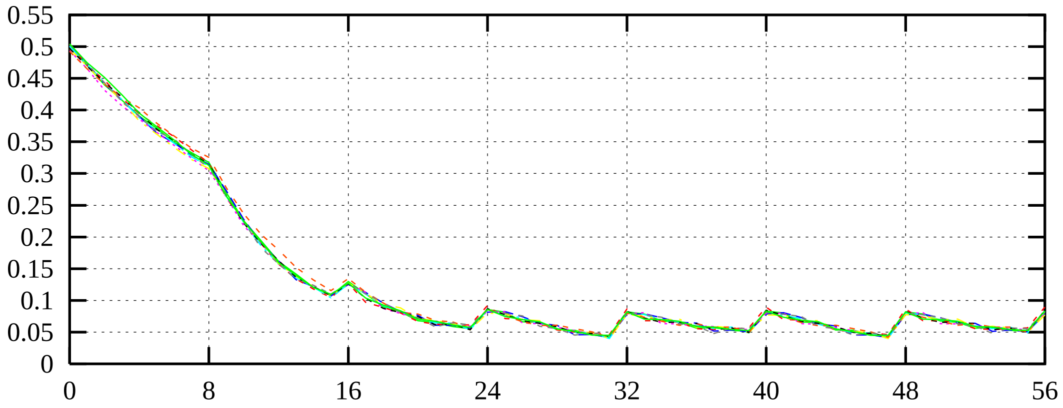


FIG. 5.5 – *Dix hasards différents.*$p_z = 0.5$

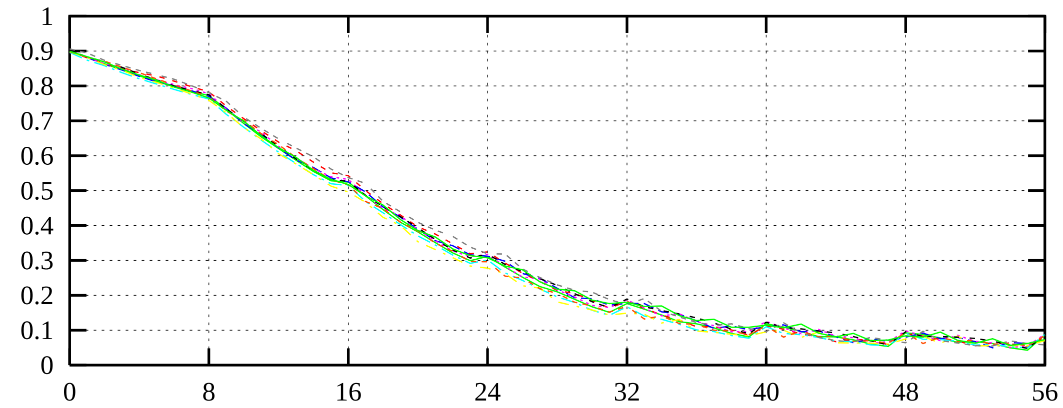


FIG. 5.6 – *Dix hasards différents.*$p_z = 0.9$

initiale. En fait, ce nombre de cellules vides est dépendant du type de condition initiale utilisée. Essayons maintenant de comprendre ce qui détermine l'allure de ces courbes. Les figures 5.7 et 5.8 nous montrent l'évolution du nombre de cellules vides issues de la même condition initiale, mais avec quatre jeux de paramètres différent, tous vérifiant les conditions exposées précédemment (5.1 et 5.2), et basés sur le nombre de seize état au total, soient **les couples {a,v} suivants : {8,9}, {6,11}, {4,13} et {2,15}**. On peut remarquer que toutes ces trajectoires possèdent un pic périodique aux itérations multiples de 16, mais qu'elles possèdent d'autres pics, distant des précédents de a itérations.

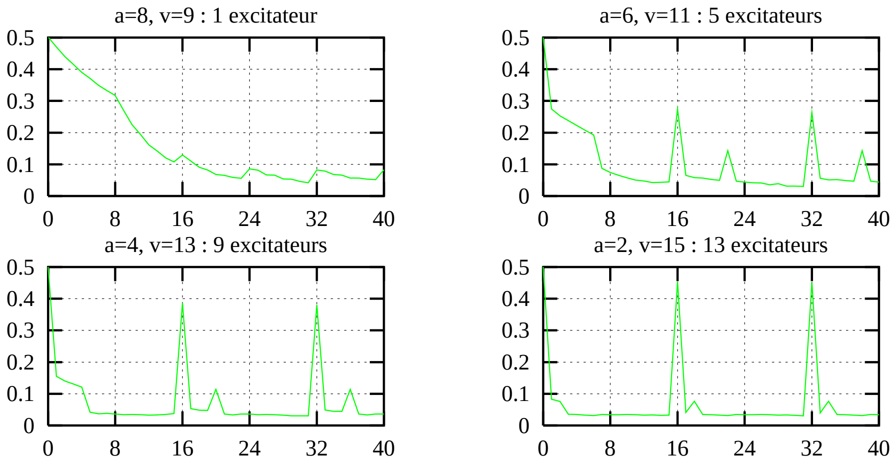


FIG. 5.7 – *Proportion initiale de cellules en zéros : 0.5*

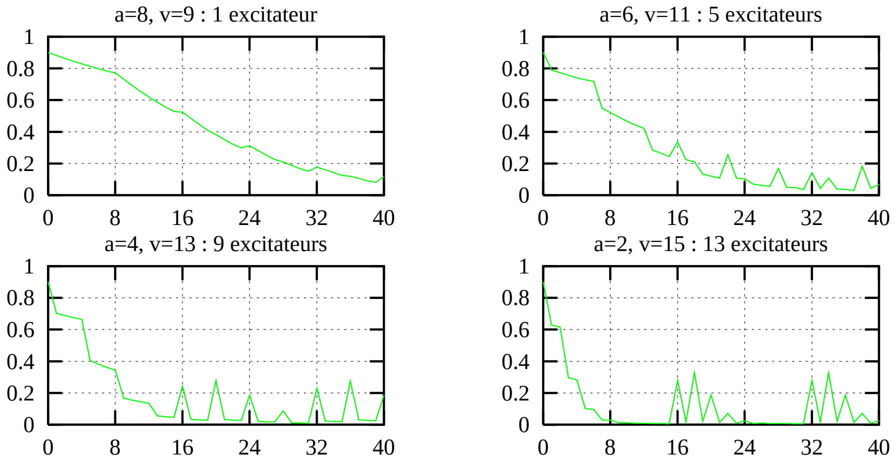


FIG. 5.8 – *Proportion initiale de cellules en zéros : 0.9*

Afin de comprendre ce phénomène, envisageons notre automate sur un terrain réduit à un couple de cellules. Si on représente le nombre de cellules en l'état neutre présentes dans ce système (0, 1 ou 2), en prenant la condition initiale (0,1), avec les quatre jeux de paramètres utilisés précédemment, on obtient les courbes de la figure 5.9.

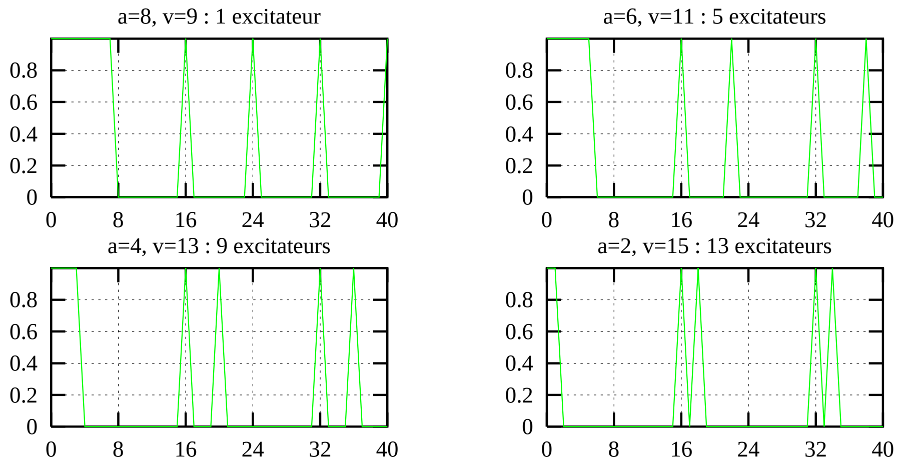


FIG. 5.9 – *Nombre de zéros dans un couple de cellules.*

On comprend donc pourquoi les courbes que nous avons vu jusqu'ici présentent des pics toutes les n itérations, séparés par la valeur de l'état du premier excitateur. En fait, les paramètres sont choisis de telle manière que chaque cellule excitée forme avec chacune de ses voisines excitatrices un couple n-périodique imperturbable. Celle-ci présentera donc l'état nul à partir de cet instant, toutes les 16 itérations. Les voisines excitatrices, elles, présenteront leur état excitateur e aux mêmes instants, et présenterons donc l'état nul e itérations avant. L'état e peut-être n'importe lequel des états excitateurs, mais plus le nombre d'excitateurs sera faible, plus nous observerons de cellules excitée par l'état a, le premier état excitateur, ceci se comprend à l'aide de la remarque suivante.

Remarque 5.2.1. *Soit une cellule C_0, et un temps t_0 tel que $C_0(t_0) = 0$ et $\forall t < t_0, C_0(t) \neq 0$. Si à t_0, C_0 ne possède pas de voisine adulte, alors, si elle est excitée à un moment donné, elle le sera par une (ou plusieurs) cellule(s) dans l'état a, le premier état excitateur.*

Preuve. En effet, si une cellule n'est pas excitée lors de son premier zéro, elle restera en cet état jusqu'à ce qu'une de ses voisines soit adulte. Si elle est excitée par une cellule en l'état e à l'instant t, alors elle aurait pu l'être à l'instant $t-1$, par la même cellule en l'état $e-1$. □

De plus, plus le nombre de cellules en zéros est important, plus nous observons de pics secondaires, car une cellules excitée excitera à son tour, a itérations plus tard, une cellule voisine, et ainsi de suite, jusqu'à ce qu'il n'y ait plus de cellule en l'état neutre. Ainsi, dans la figure 5.7, on observe un seul pic secondaire, alors que dans la figure 5.8, nous pouvons en compter trois. L'automate cellulaire garde donc une certaine mémoire de la distribution initiale des états, ce qui explique le fait qu'à partir de conditions initiales de même type, on obtienne des proportion de zéros dans le temps pratiquement similaires.

5.3 Transition Aléatoire

Nous allons maintenant introduire un aspect probabiliste à notre automate. Supposons que la présence d'un adulte dans le voisinage d'une cellule vide provoque son passage en l'état 1 non pas à coup sur, mais avec une probabilité p. Précisons en pseudo langage de programmation ce que nous entendons par là. On note W la variable aléatoire de Bernoulli de probabilité p ($P(W=1)=p; P(W=0)=1-p$) :

`si` $C_{i,j}(t)=0$
`alors`
  `si {`“Un voisin au moins ce $C_{i,j}$ est adulte” `et` $W=1$`}`
  `alors` $C_{i,j}(t+1)=1$
  `sinon` $C_{i,j}(t+1)=0$

On remarque que cette règle est différente de la règle ci-dessous que nous n'avons pas considéré. On désigne par G une mémoire (compteur de graines ou d'adultes).

`si` $C_{i,j}(t)=0$
`alors {`
  `si` $C_{i+1,j}(t)$ est adulte `et` $W=1$ `alors` $G:=G+1$;
  `si` $C_{i-1,j}(t)$ est adulte `et` $W=1$ `alors` $G:=G+1$;
  `si` $C_{i,j+1}(t)$ est adulte `et` $W=1$ `alors` $G:=G+1$;
  `si` $C_{i,j-1}(t)$ est adulte `et` $W=1$ `alors` $G:=G+1$;
  `si` $G \neq 0$
  `alors` $C_{i,j}(t+1)=1$
  `sinon` $C_{i,j}(t+1)=0$
    `}`

Dans la première procédure, dès qu'un voisin au moins est adulte, la probabilité de passage en 1 est p, alors que dans la seconde, elle dépend de G, le nombre de voisin adulte, et vaut $1-(1-p)^G$.

Mais revenons à la première règle, qui est celle que nous allons considérer pour la suite. Si p est égal à 1, le modèle est le même que le précédent. Si p est nul, le système meurt très rapidement, faute de pouvoir se reproduire. Mais que se passe-t-il entre ces deux extrêmes? Pour en avoir une idée, nous effectuons tout d'abord trois séries d'expériences avec une probabilité de passage p variant de 0.1 à 1 et trois densités initiales de zéros ($p_z=0$: figure 5.10, $p_z=0.5$: figure 5.11 et $p_z=0.9$: figure 5.12).

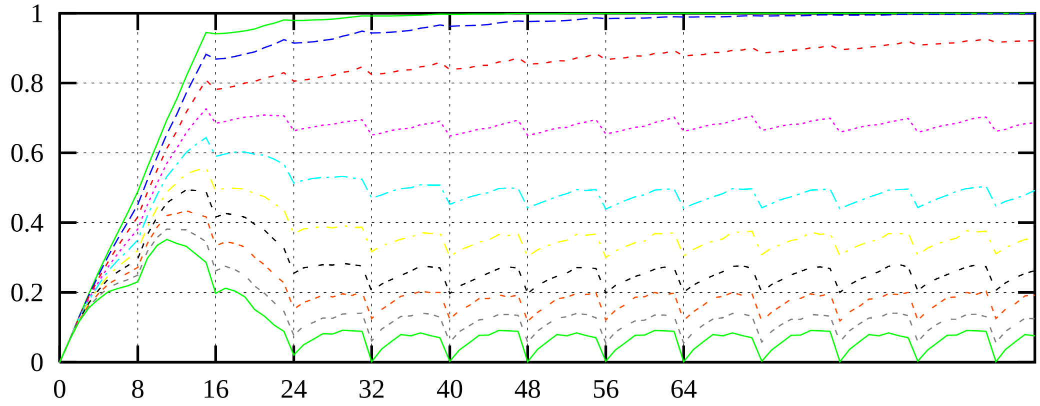


FIG. 5.10 – $p_z = 0$. *Les probabilités de passage varient de* 0.1 *à* 1.

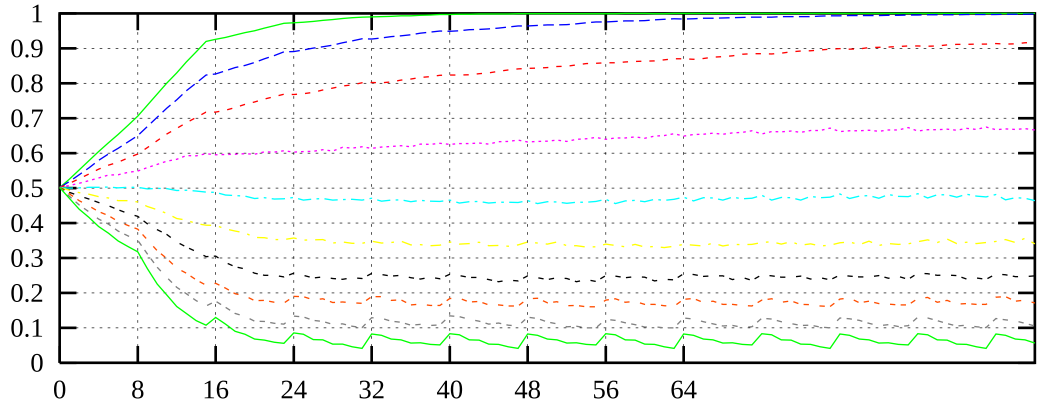


FIG. 5.11 – $p_z = 0.5$. *Les probabilités de passage varient de* 0.1 *à* 1.

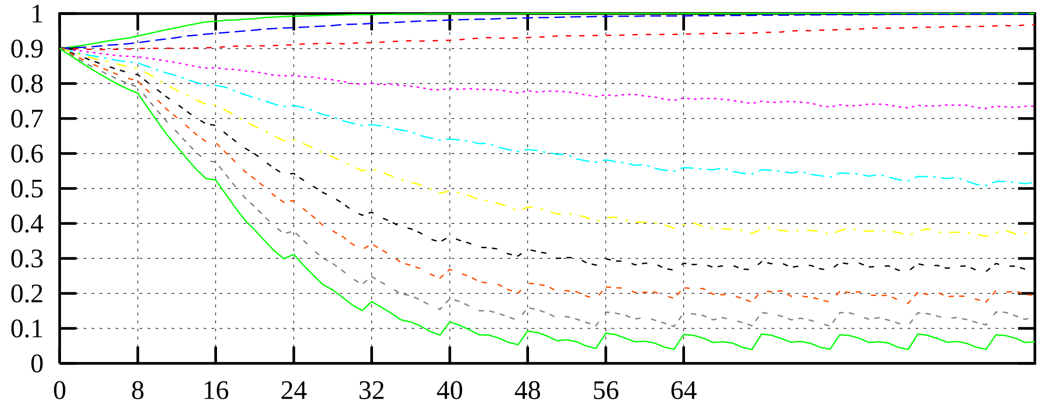


FIG. 5.12 – $p_z = 0.9$. *Les probabilités de passage varient de* 0.1 *à* 1.

Nous constatons que le nombre de zéros augmente quand la probabilité diminue, comme nous pouvions nous y attendre. Le nombre moyen de cellules vides à l'air de se stabiliser après une centaine d'itérations, et les moyennes atteintes semblent être les mêmes pour chaque jeu de paramètres, quelque soit la densité initiale de zéros. Pour nous en convaincre, nous allons calculer la proportion moyenne de cellules vides pour les itérations 200 à 216, pour les trois densités initiales habituelles, en fonction de la probabilité de passage, celle-ci variant de 0 à 1 par pas de 0.01 (figure 5.13).

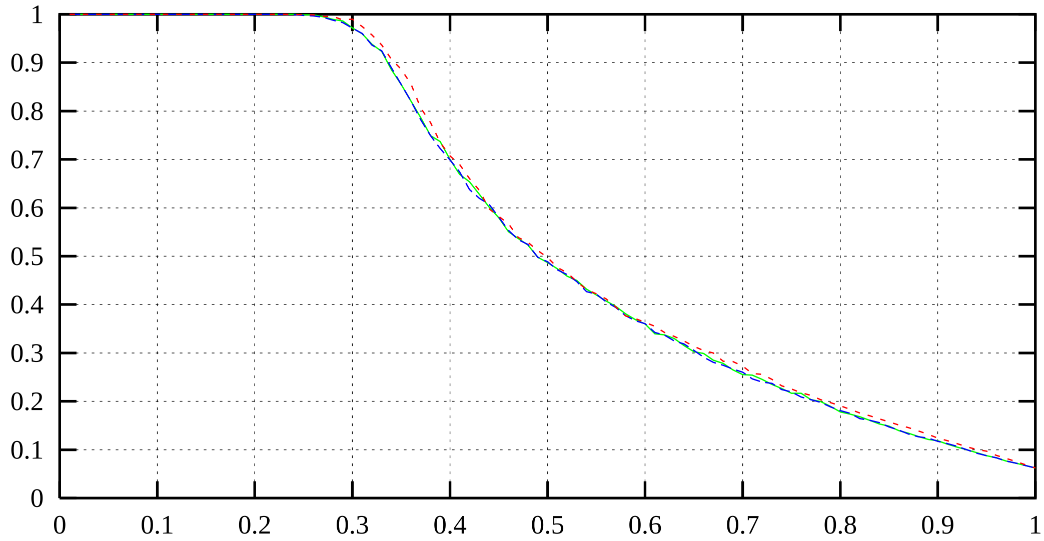


FIG. 5.13 – $p_z \in \{0.0{,}0.5{,}0.9\}$. *Moyenne du nombre de cellule vides sur 16 itérations après 200 itérations, en fonction de p,la probabilité de passage.*

Ici aussi, les courbes se superposent presque parfaitement. Notre automate semble tendre vers une proportion moyenne de zéros dépendante de la probabilité de passage, et non de la densité initiale. La figure 5.14 illustre la même expérience, avec cette fois-ci $p_z = 0.5$, mais pour tous les couples $\{a,v\}$ étudiés précédemment: $\{8{,}9\}$, $\{7{,}10\}$, $\{6{,}11\}$, $\{5{,}12\}$, $\{4{,}13\}$, $\{3{,}14\}$, $\{2{,}15\}$ et $\{1{,}16\}$.

Comme prévu, la proportion moyenne de cellules vides atteinte diminue lorsque le nombre d'excitateurs augmente. On observe dans chaque cas un **seuil** pour la probabilité de passage en dessous duquel le système s'éteint. On remarque que le **phénomène de seuil** est plus accentué lorsque le nombre d'excitateurs augmente.

5.4 Modèle non-spatialisé

Nous n'avons pas du tout étudié ici l'automate cellulaire sous son aspect spatial, et nous avons même mis en évidence une relative indépendance de certaines variables de l'automate vis à vis du caractère spatial. Cela nous amène à nous demander si certains aspects de ce modèle ne pourraient pas être reproduits par un modèle non spatialisé de type Usher

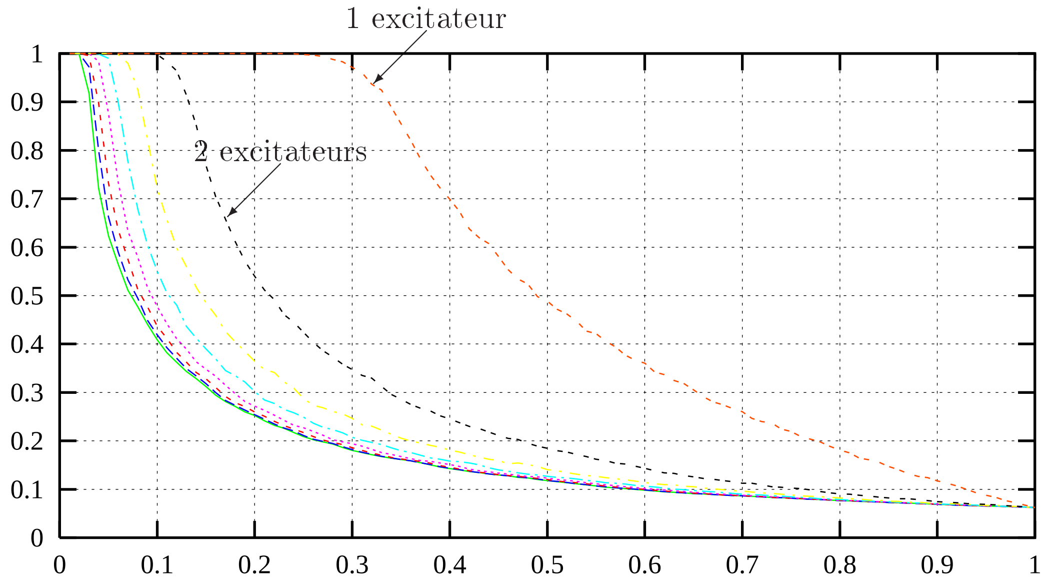


FIG. 5.14 – $p_z = 0.5$. *De 1 à 15 excitateurs. Moyenne du nombre de cellules vides sur 16 itérations après 200 itérations, en fonction de p,la probabilité de passage.*

([Fav95]). Considérons donc que nous ne connaissons du système étudié précédemment que la proportion d'individus dans chaque classe d'âge, et que nous ignorons leurs positions. Pour ce faire, nous appellerons $x_i(t)$ la proportion des individus dans l'état i à l'instant t. Il découle de là une première égalité :

$$\sum_i x_i(t) = 1 \tag{5.3}$$

Le seul endroit ou l'aspect spatial intervient dans notre automate cellulaire est le passage d'une cellule de l'état neutre à l'état 1, tout les autres passages étant totalement déterministes. Si l'on considère que les différents états sont répartis uniformément et indépendemment dans l'espace, ce qui est évidemment faux, la proportion de cellules en l'état neutre qui possèdent au moins une voisine adulte est :

$$P = 1 - \left(\sum_{i \notin Ad} x_i(t) \right)^4 \tag{5.4}$$

Ce qui nous permet d'écrire le système suivant :

$$\left\{\begin{array}{lcl} x_0(t+1) & = & x_{n-1}(t) + x_0(t)\left(\sum_{i \notin Ad} x_i(t)\right)^4 \\ x_1(t+1) & = & x_0(t)\left(1 - \left(\sum_{i \notin Ad} x_i(t)\right)^4\right) \\ x_2(t+1) & = & x_1(t) \\ \vdots & & \\ x_{n-1}(t+1) & = & x_{n-2}(t) \end{array}\right. \tag{5.5}$$

où la première équation exprime que le nombre de cellules en zéro à l'instant $t+1$ est la somme du nombre de cellules en $n-1$ à l'instant t, et du nombre de cellules en zéro qui ne possédaient pas de voisine adulte à l'instant t; ces cellules restent donc dans l'état neutre.

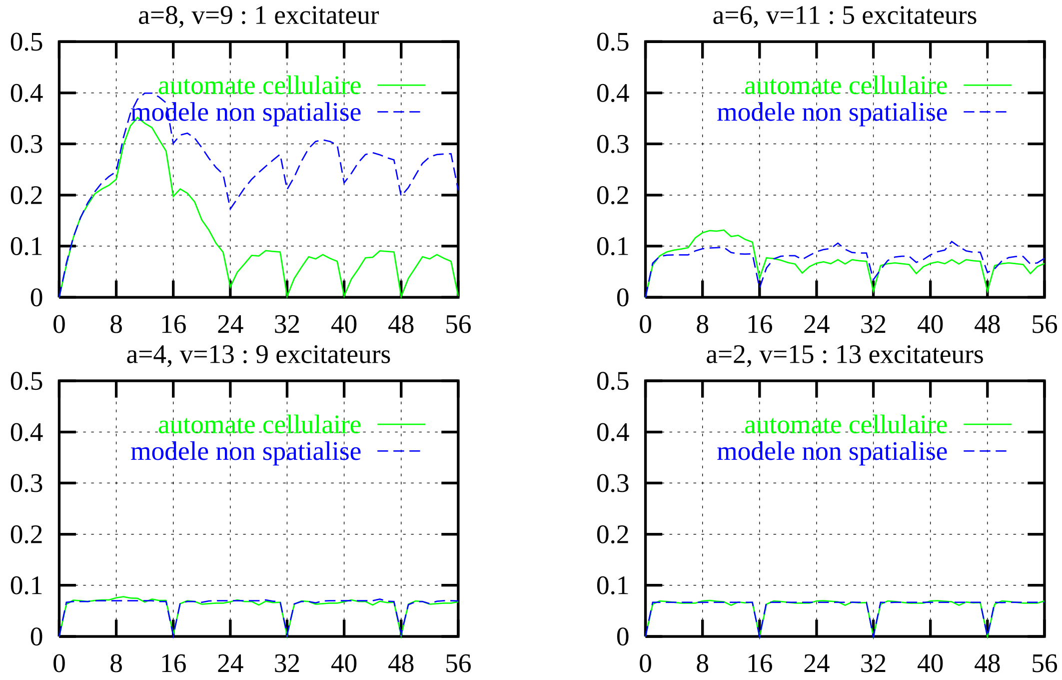


FIG. 5.15 – $p_z = 0.0$ *: on augmente le nombre d'états excitateurs*

Les figures 5.15, 5.16 et 5.17 nous permettent de comparer les résultats du modèle non spatialisé à ceux de l'automate cellulaire lui-même. Nous avons effectué trois séries de calculs ($p_z = 0$, $p_z = 0.5$ et $p_z = 0.9$) en prenant les couples $\{a,v\}$ utilisés précédemment ($\{8,9\}$, $\{6,11\}$, $\{4,13\}$ et $\{2,15\}$). Nous pouvons constater que le modèle non spatialisé est meilleur (c'est-à-dire que son comportement ressemble le plus à celui de l'automate

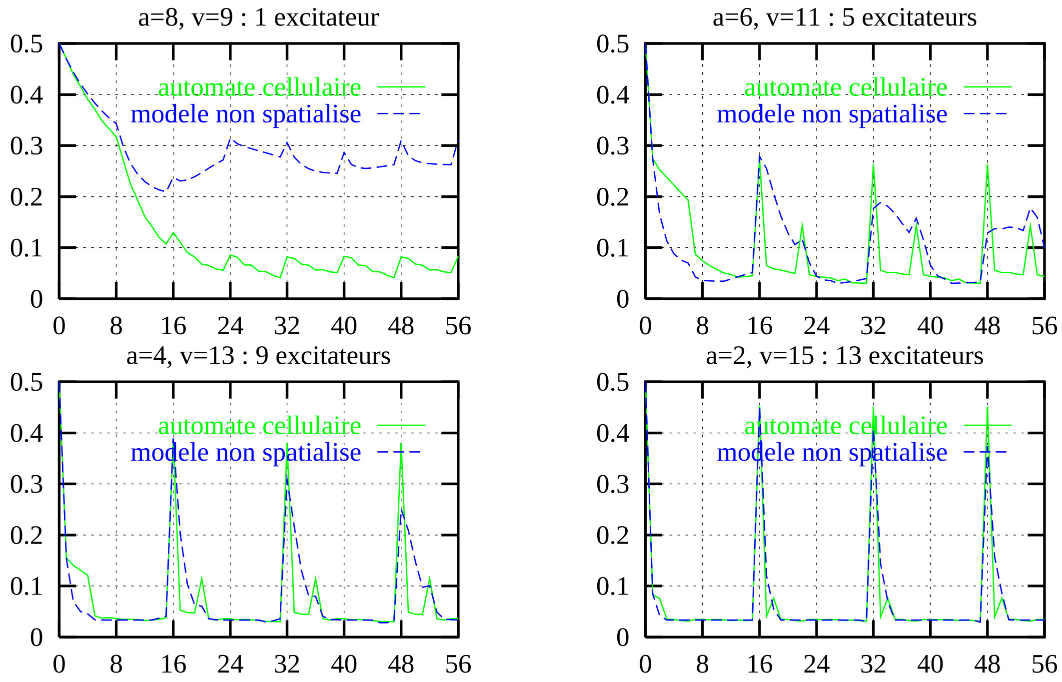


FIG. 5.16 – $p_z = 0.5$: *on augmente le nombre d'états excitateurs*

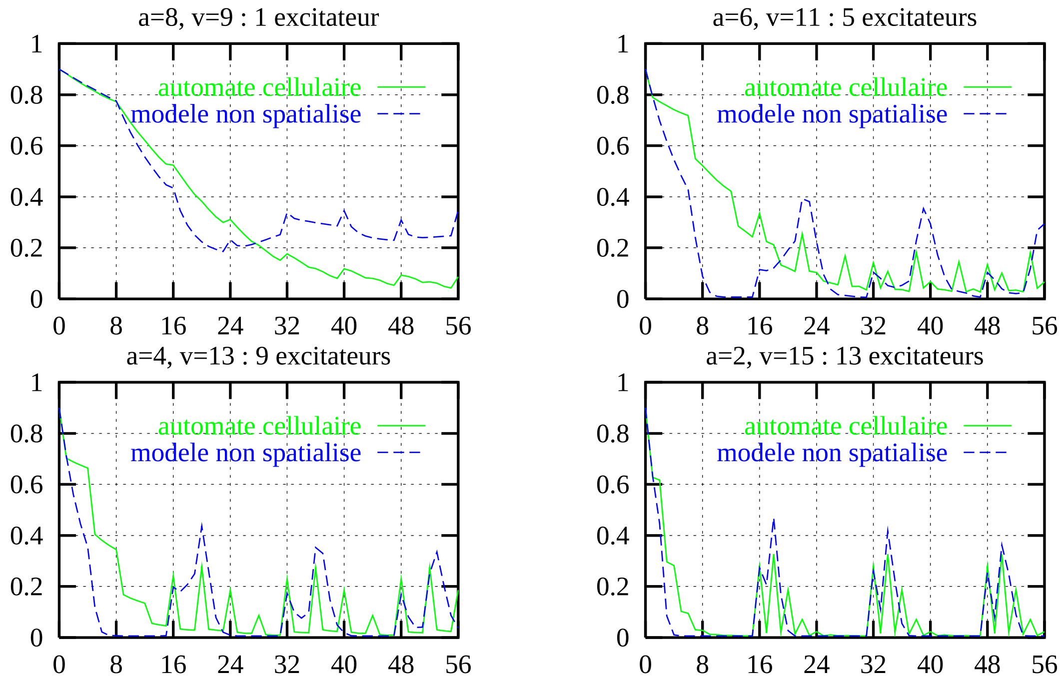


FIG. 5.17 – $p_z = 0.9$: *on augmente le nombre d'états excitateurs*

cellulaire) dans les cas suivants :

- Au commencement de la trajectoire. En effet, pendant les premières itérations, on peut considérer que les état présents sont repartis uniformément dans l'espace, ce qui n'est bien sûr plus du tout le cas par la suite.
- Lorsque le nombre d'excitateurs est grand. En effet, plus le nombre d'excitateurs est grand, moins l'aspect spatial de l'automate est présent, la présence d'un adulte dans le voisinage de chaque cellule en l'état neutre devenant de plus en plus probable.

Si l'on étudie les éventuels équilibres de ce modèle, nous parvenons au système d'équations suivant :

$$\begin{cases} x_0 = \dfrac{x_1}{1-(1-(v-a)x_1)^4} \\ x_0 = 1-(n-1)x_1 \end{cases} \tag{5.6}$$

Le schéma de la figure 5.18 montre qu'il existe une et une seule solution à ce système d'équations, quelque soient les valeurs de a, v et n.

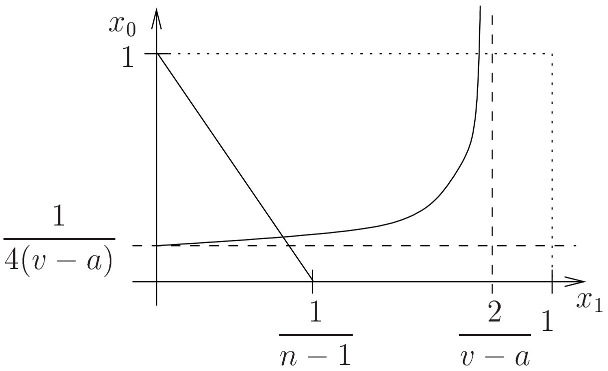


FIG. 5.18 – *Schéma du système d'équations 5.6*

Nous pouvons maintenant adapter le système d'équations pour qu'il modélise l'automate cellulaire avec transition probabiliste. Une cellule en l'état neutre disposant d'un voisin adulte ne passe plus en 1 à coup sur, mais avec une probabilité p. La probabilité qu'une cellule en zéro passe en 1 devient donc :

$$P = p.\left(1-\left(\sum_{i \notin Ad} x_i(t)\right)^4\right) \tag{5.7}$$

Ce qui nous donne le système suivant :

$$
\left\{
\begin{array}{lcl}
x_0(t+1) & = & x_{n-1}(t) + x_0(t)\left(1 - p.\left(1 - \left(\sum_{i \notin Ad} x_i(t)\right)^4\right)\right) \\
x_1(t+1) & = & x_0(t).p.\left(1 - \left(\sum_{i \notin Ad} x_i(t)\right)^4\right) \\
x_2(t+1) & = & x_1(t) \\
\vdots & & \\
x_{n-1}(t+1) & = & x_{n-2}(t)
\end{array}
\right. \tag{5.8}
$$

que nous avons simulé, mais dont nous n'avons pas encore étudié les propriétés mathématiques. Les figures 5.19, 5.20 et 5.21 nous permettent de comparer les résultats du modèle non spatialisé à ceux de l'automate cellulaire, pour le jeux de paramètres $\{a = 8, v = 9, n = 16\}$.

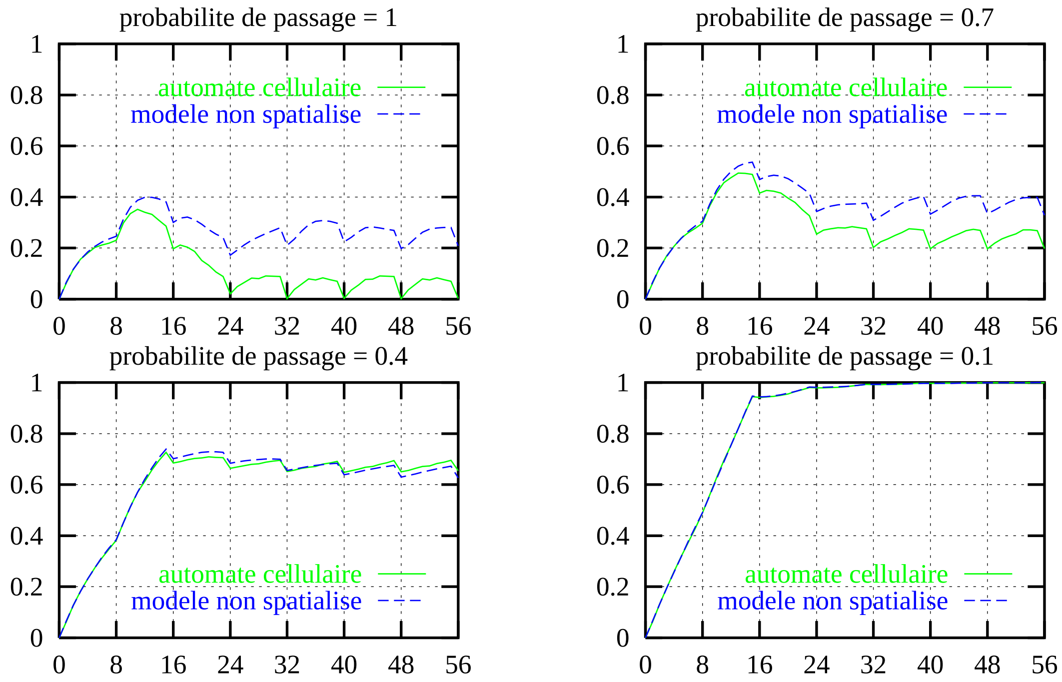


FIG. 5.19 – $\{a = 8, v = 9, n = 16\}$. $p_z = 0.0$: *plusieurs probabilités de passage.*

Nous pouvons constater que les résultats sont d'autant meilleurs que la probabilité de passage est faible.

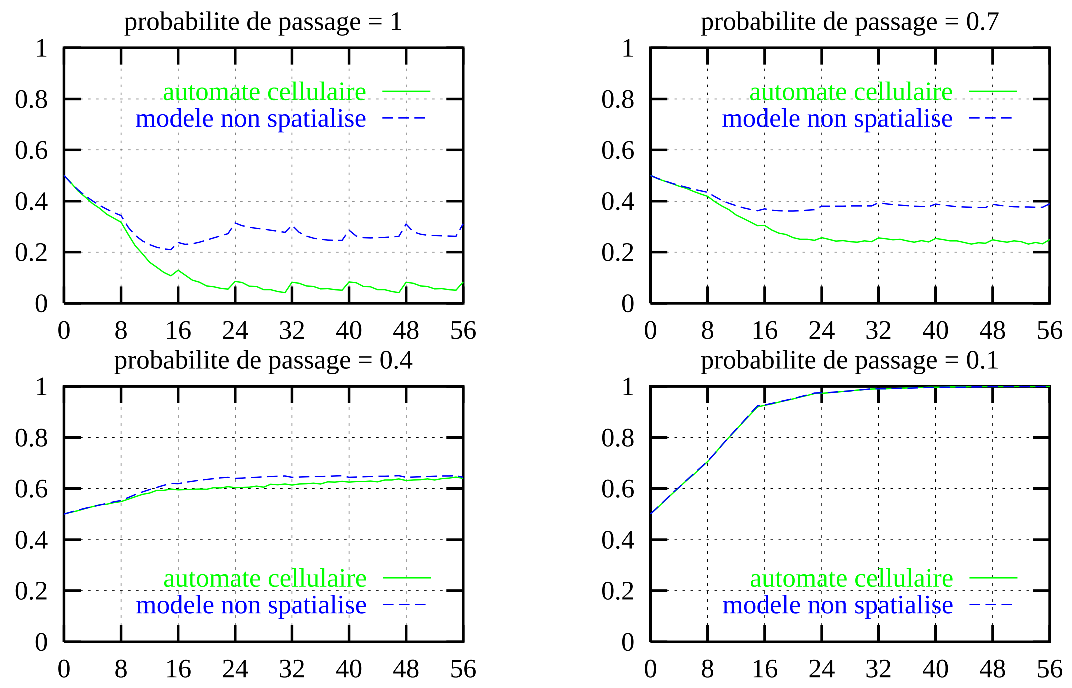


FIG. 5.20 – $\{a = 8, v = 9, n = 16\}$. $p_z = 0.5$: *plusieurs probabilités de passage.*

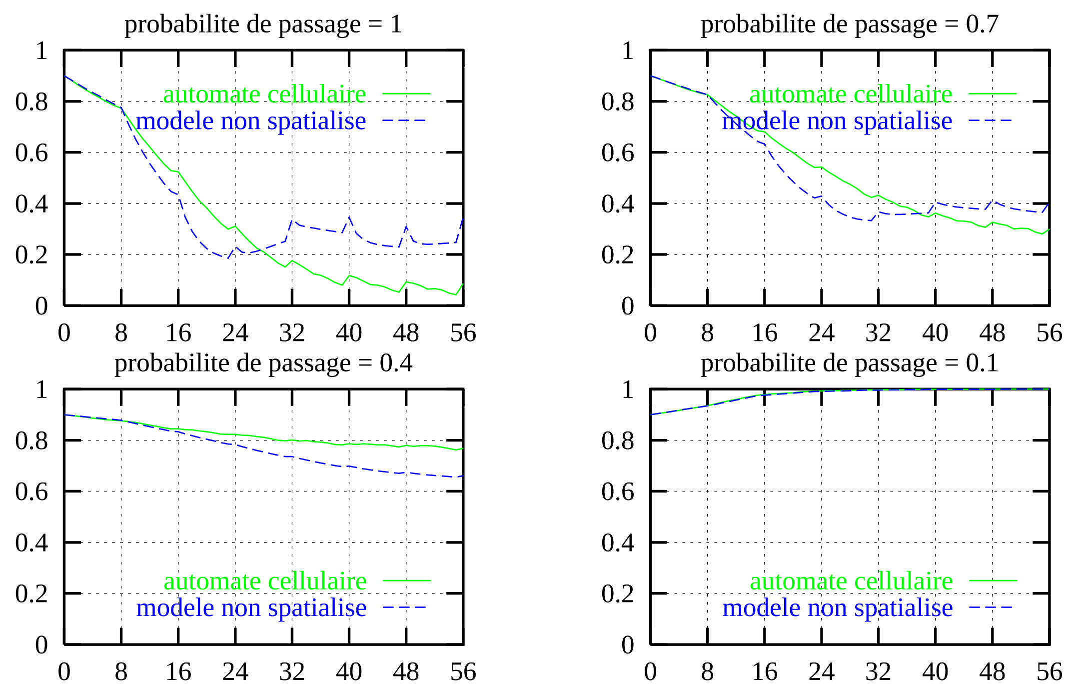


FIG. 5.21 – $\{a = 8, v = 9, n = 16\}$. $p_z = 0.9$: *plusieurs probabilités de passage.*

5.5 Conclusion

Nous avons montré, pour les paramètres choisis (tels que la période de fécondité soit symétrique par rapport au milieu du cycle de vie), que dans le cas déterministe, la probabilité pour qu'il se présente une condition initiale telle que la forêt disparaisse est extrêmement faible dès que la proportion de zéros dans la condition initiale (c'est à dire la place libre) n'est ni trop forte, ni trop faible. L'évolution de la proportion de zéros est de période 16, l'amplitude des oscillations étant d'autant plus grande que la proportion de zéros est forte dans la condition initiale, mais les oscillations ne dépendent pas de la répartition (au hasard) de la condition initiale.

L'introduction d'une probabilité de passage à l'état 1 (plus petite que 1) lorsqu'une cellule en l'état zéro a au moins une cellule excitatrice voisine produit deux effets :

- Diminuer la probabilité de passage diminue l'amplitude des oscillations, c'est un effet régularisant du hasard.
- Pour tous les paramètres considérés, il apparaît nettement un seuil en dessous duquel la proportion de zéros devient 1 (la forêt disparaît).

Le modèle non spatialisé que nous proposons, s'il reproduit qualitativement les oscillations, est encore loin d'une bonne adéquation. Nous risquerons cependant la conclusion suivante :

Lorsque le nombre d'excitateurs est grand, lorsque le passage de 0 à 1 est aléatoire, pour un terrain assez vaste, un modèle non spatialisé densité dépendant peut rendre compte efficacement de l'évolution du système. En effet, c'est à ces conditions que l'on peut espérer voir disparaître *"l'effet géométrique"* décrit dans le Chapitre 1, c'est à dire le caractère propre de l'aspect spatial du modèle. Ce sont des conditions dans lesquelles les caractéristiques spatiales se "moyennisent".

Chapitre 6

Modèle de type "accélération"

6.1 Présentation

Dans son article "A Model for the Mosaic-Cycle Concept" ([Wis91]), Ch. Wissel construit un modèle de forêt simple de type automate cellulaire, basé sur le concept de "Mosaïque cyclique" : La forêt est divisée en cellules, et chacune d'elles voit la succession d'états (figure 6.1), influencée par les états des cellules voisines. Les états des cellules sont représentatifs de la végétation présente à l'intérieur de la cellule. Par exemple, le modèle de Wissel s'applique à une forêt de hêtres d'Europe centrale. La succession des états est la suivante :

- 1 et 2 : Pas de végétation.
- 3 à 7 : Présence de bouleaux.
- 8 à 22 : Forêt mélangée.
- 23 à 25 : Un bosquet de hêtres.
- 26 à 50 : des Hêtres adultes.
- 51 à 55 : des vieux Hêtres.

En ce qui concerne le temps, une itération représente une décennie. Wissel veut son modèle le plus simple possible, mimant les "facteurs clés" de la dynamique spatiale étudiée :

- La colonisation d'une ouverture par les bouleaux, s'ils sont présents dans une cellules voisine.

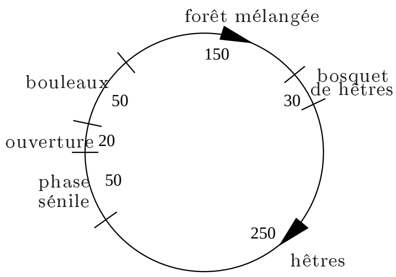


FIG. 6.1 – *Cycle de base d'une cellule du modèle de Wissel*

– La colonisation d'une ouverture par la "forêt mélangée" si elle est présente dans une cellule voisine.
– L'effet des radiations solaires sur les hêtres exposés brutalement à la suite d'une ouverture au Nord, entraînant la mort des arbres à court terme.

Ce sont les trois seuls facteurs pouvant influencer le cycle de chaque cellule. Mais Wissel estime dans son article que le modèle ainsi constitué est trop compliqué, et décide de ne conserver que le troisième facteur, afin de mieux comprendre les résultats obtenus par simulation :

> *In order to obtain a better understanding of results, several parts of the model have been changed. The early colonisation by birch and by mixed wood have been omitted without essentially changing the results[...] They may be realistic, but they are not essential for the results of the model[...]*

Le facteur "radiation" retenu par Wissel, est de type probabiliste, ce qui fait de son modèle un automate cellulaire stochastique, c'est-à-dire qu'une partie des "décisions" prises par l'automate sera due au hasard. Notre travail portant plutôt sur les automates cellulaires déterministes, nous nous sommes plutôt intéressés à l'autre aspect du modèle, que nous appellerons "accélération".

Pour cela, nous allons étudier un automate cellulaire très simple, de type "mosaïque cyclique", intégrant seulement le facteur accélération.

– L'ensemble des états $E = [0; n[$, sera divisés de la même manière que pour l'automate précédent :
  – l'état neutre : 0
  – l'ensemble des états jeunes : $J = [1; a[$.
  – l'ensemble des états adultes : $A = [a; v[$.
  – l'ensemble des états vieux : $V = [v; n[$.
– Le voisinage considéré sera ici aussi le voisinage de Von Neumann (figure 6.2).

	c4	
c3	c	**c1**
	c2	

FIG. 6.2 – *Le voisinage de Von Neumann*

– La fonction de transition sera définie comme suit:
  Soit c une cellule du plan et t un entier positif:

`si` $\mathcal{A}(c,t) \neq 0$
`alors` $\mathcal{A}(c,t+1) = \mathcal{A}(c,t) + 1 \mod n$
`sinon`

`si` $\exists\ d \ \in\ \mathcal{V}(c)$ telle que $\mathcal{A}(d,t)\ \in\ A$
`alors` $\mathcal{A}(c,t+1) = a$
`sinon` $\mathcal{A}(c,t+1) = \mathcal{A}(c,t) = 1$

Ici, chaque cellule passe à chaque pas de temps à l'étape suivante, l'état suivant $(n-1)$ étant toujours l'état neutre. Mais une cellule dans l'état neutre peut passer directement à l'état a si l'état de l'une de ses voisines appartient à l'ensemble des états adultes, $[a,v[$. Son processus de vie est accéléré, et elle n'a pas à passer par les états jeunes.

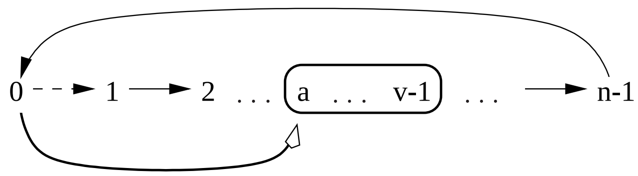


FIG. 6.3 – *Automate de type "accélération"*

6.2 Remarques à caractère théorique

Nous pouvons tout d'abord rappeler une propriété générale des automates cellulaires en disant que la trajectoire de l'ensemble des cellules finit par devenir périodique. Nous pouvons ensuite remarquer qu'une cellule isolée sera automatiquement de période n, puisqu'aucune voisine ne fera passer son état directement de zéro à a. Existe-t-il d'autres périodes acceptables par les règles de cet automate.

Définition 6.2.1. *Nous appellerons* ***trajectoire périodique simple*** *une trajectoire périodique telle que pour chaque cellule c, il n'existe* ***qu'un seul*** *t tel que, pendant une période, $\mathcal{A}(c,t) = 0$.*

Exemple 2.
- *0, 1 ..., $n-1$, 0, 1 ... : période simple = n.*
- *0, a ..., $n-1$, 0, a ... : période simple = $n-a+1$.*
- *0, 1 ..., $n-1$, 0, a ..., $n-1$, 0, 1 ... : période complexe = $2n-a+1$.*

Puisque les deux seuls états qui peuvent suivre l'état zéro sont les états 1 et a, les deux seules périodes simples acceptables sont respectivement n et $r = (n-a+1)$. Nous pouvons nous demander ici à quelles conditions l'état d'une cellule peut être r-périodique.

6.2.1 Les cycles auto-excitateurs

La définition est un peu différente que celle du chapitre précédent (définition 4.3.8), car la règle de transition est différente, mais basée sur le même principe :

Définition 6.2.2. *Un cycle auto-excitateur est un cycle de cellules $(c_1,c_2,\ldots,c_q)$ (défini-*

tion 2.3.2) tel que :

$$\left.\begin{array}{l}\mathcal{A}(c_i,t)=0\\ \mathcal{A}(c_i,t+1)=a\end{array}\right\}\Rightarrow \mathcal{A}(c_{i+1},t)\in A$$

Une condition suffisante pour que l'état d'une cellule soit r-périodique est qu'elle appartienne à un **cycle auto-excitateur r-périodique**.

Proposition 6.2.3. *Un cycle de cellules $(c_1,\ldots,c_q)$ est auto-excitateur de période r si et seulement si il vérifie pour tout t les conditions suivantes :*

$$\left\{\begin{array}{ll}\mathcal{A}(c_i,t)\notin\{1,\ldots,a-1\} & \\ \mathcal{A}(c_i,t)=0 & \Rightarrow a\leq \mathcal{A}(c_{i+1},t)\leq v-1\\ \mathcal{A}(c_i,t)=e\geq a & \Rightarrow e+1\leq \mathcal{A}(c_{i+1},t)\leq e+v-a\end{array}\right.$$

Exemple 3. *$a=5$, $v=8$, $n=16$:*

$$\begin{array}{|c|c|}\hline 0 & 7\\ \hline 13 & 10\\ \hline\end{array}\rightarrow\begin{array}{|c|c|}\hline 5 & 8\\ \hline 14 & 11\\ \hline\end{array}\rightarrow\begin{array}{|c|c|}\hline 6 & 9\\ \hline 15 & 12\\ \hline\end{array}\rightarrow\begin{array}{|c|c|}\hline 7 & 10\\ \hline 0 & 13\\ \hline\end{array}\ldots : \textit{période } r=n-a+1=12$$

Nous pouvons nous demander maintenant comment interagissent entre elles les deux périodes simples n et $r=n-a+1$.

6.2.2 Cohabitation entre deux périodes

Un premier résultat nous montre ce qu'il advient d'une cellule excitée par une cellule r-périodique.

Proposition 6.2.4. *Une cellule excitée par une cellule r-périodique devient à son tour r-périodique.*

Pour démontrer ce résultat, il suffit de suivre ces deux cellules à la trace.

$$\begin{array}{|c|c|}\hline 0 & e\\ \hline\end{array}\rightarrow\begin{array}{|c|c|}\hline a & e+1\\ \hline\end{array}\xrightarrow{n-a}\begin{array}{|c|c|}\hline 0 & e\\ \hline\end{array}\rightarrow \text{etc}\ldots$$

On peut se demander maintenant comment deux cellules de périodes simples différentes peuvent cohabiter.

Théorème 6.2.5. *Les trajectoires de deux cellules voisines c et d ne peuvent être périodiques respectivement de périodes n et r que si :*

$$v-a<PGCD(n,r)$$

Preuve. Pour que ces deux cellules voisines adoptent des périodes différentes, il est nécessaire que la cellule de trajectoire r-périodique (d) n'excite jamais l'autre cellule (c) qui adopterais dans ce cas une trajectoire r-périodique (proposition 6.2.4).

Nous allons donc considérer l'état de la cellule d à chaque fois que la cellule c est en l'état neutre, afin de savoir s'il apparaît un état adulte. Supposons qu'à l'état initial, la

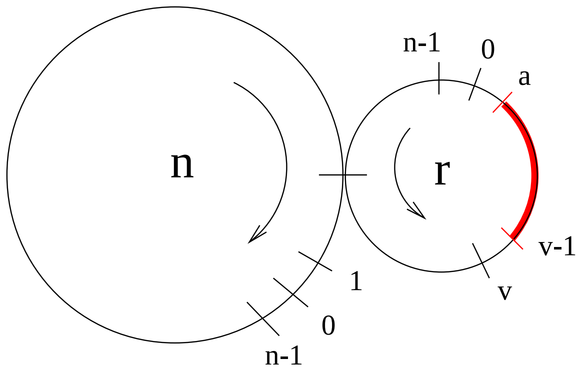


FIG. 6.4 – *Deux cellules voisines de périodes n et r.*

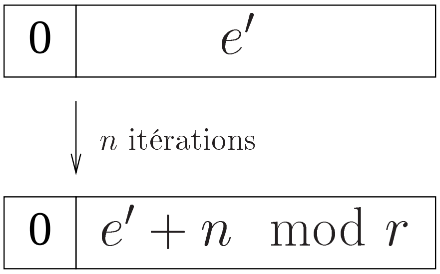


FIG. 6.5 – *Quel sera l'état de d aprés n itérations ?*

cellule c soit en 0, et la cellule d en l'état e. Aprés n itérations, la cellule c sera à nouveaux en 0. Quel sera alors l'état de la cellule d. Définissons une bijection B de $\{0,a,a+1\ldots,n-1\}$ vers $\{0,1,2\ldots,n-a+1\}$.

$$
\begin{array}{cccccc}
 & 0 & a & a+1 & \ldots & n-1 \\
B & \downarrow & \downarrow & \downarrow & & \downarrow \\
 & 1 & 2 & 3 & \ldots & n-a+1
\end{array}
$$

Si nous appelons e' l'image de e par B, aprés n itération l'image par B de l'état de la cellule d sera donc $e'+n \mod r$ (figure 6.5). L'ensemble des image par B des états adoptés par la cellule d quand la cellule c est en 0 est donc :

$$\{e' + i.n \mod r, i \in \mathbb{N}\} \tag{6.1}$$

Cet ensemble est le translaté par e' du sous-groupe de $\mathbb{Z}/r\mathbb{Z}$ engendré par n. Un résultat bien connu nous dit que le plus petit entier engendrant ce sous-groupe est

$$p = PGCD(n,r)$$

L'ensemble (6.1) peut donc s'écrire

$$\{e',e'+p,e'+2p,\dots\}$$

L'entier p étant la différence entre deux éléments consécutifs de (6.1), si $p \leq v-a$, l'ensemble (6.1) contient forcemment au moins l'image par B d'un état adulte. Il existe alors un instant t tel que la cellule c soit en 0, et sa voisine d dans un état adulte. □

Nous pouvons maintenant appliquer ce résultat à un domaine bi-dimentionnel quelconque :

Résultat 6.2.6. *Soit un automate de type accélération, dont la règle de passage est définie par les paramètres a, v et n. Si*

$$v-a \geq PGCD(n \mod (n-a+1), n-a+1)$$

et qu'il existe au moins une cellule $(n-a+1)$-périodique, alors le réseau entier sera $(n-a+1)$-périodique.

6.3 Expériences

Nous avons réalisé une série d'expériences, consistant en la modélisation de l'automate de type Accélération sur un domaine carré de 128×128 cellules. Pour la condition initiale, l'état de chaque cellule est tiré au hasard (uniforme) parmi les n états, sauf les cellules du bord, qui sont dans l'état neutre. On modélise dix trajectoires différentes, pour chaque jeu de paramètres $\{a,v,n\}$ tel que n soit inférieur ou égal à 16. Cela nous donne $\sum_{n=4}^{16} \frac{(n-3)(n-2)}{2} = 455$ jeux de paramètres, soit 4550 expériences. Pour chacune d'entre elles, on note la période atteinte. La table 6.1 résume ces informations. Chaque tableau correspond à une valeur de n. Lorsque, pour un jeu de paramètres, plusieurs périodes différentes sont trouvées, on note dans la case correspondante la plus petite et la plus grande de ces périodes. Quand la seule période obtenue est n, on note n. Nous avons de plus ajouté une information à ces tableaux. Les cases qui apparaissent en blanc sur fond noir correspondent aux cas tels que $v-a < PGCD(n,r)$, c'est-à-dire lorsque des cellules de périodes n et $r = n-a+1$ peuvent cohabiter. On constate que les résultats obtenus sont en accord avec les remarques théoriques :

- Pour les cases blanches : $v-a \geq PGCD(n,r)$, ce qui signifie que deux cellules de périodes n et r ne peuvent pas cohabiter (voir théorème 6.2.5) et s'il existe au moins une cellule de période r, alors tout le réseau adopte la période r (résultat 6.2.6). C'est ce qu'il se produit généralement, sauf parfois lorsqu'il n'y a qu'un seul excitateur (les cases sur les diagonales) : en effet, s'il y a trop peu d'excitateurs, il peut ne pas se former de cycle de période r. On trouve alors la période n.
- Pour les cases noires : $v-a < PGCD(n,r)$, les cellules de périodes n et r peuvent cohabiter. On constate donc généralement comme période le $PPCM$ de n et r, voir un multiple de cette valeur, dans le cas où il apparaît une période différente de n et r (période complexe, voir définition 6.2.1). Lorsqu'il n'y a qu'un seul excitateur,

pour les même raisons que dans le cas précédent, il peut ne pas apparaître de cycle de période r. On note alors la période n.

6.4 Conclusion

L'étude théorique de l'automate cellulaire du type Accélération nous donne une bonne compréhension de son comportement. Mais il reste quelques points qu'il faudrait appronfondir.

- Au niveau théorique, il faudrait étudier les périodes complexes.
- De plus, il faudrait soumettre l'automate à différents tests (voir section 4) comme par exemple étudier l'influence de la taille du terrain (terrain plus grand $\Rightarrow$ moins de périodes n?), ou de l'introduction de caractères stochastiques.

$n = 4$	
v \a	2
3	3

$n = 5$		
v \a	2	3
3	4	
4	4	3

$n = 6$			
v \a	2	3	4
3	5		
4	5	12:60	
5	5	4	n

$n = 7$				
v \a	2	3	4	5
3	6			
4	6	5		
5	6	5	4	
6	6	5	4	3

$n = 8$					
v \a	2	3	4	5	6
3	7				
4	7	24:168			
5	7	6	5:8		
6	7	6	5	n	
7	7	6	5	n	3

$n = 9$						
v \a	2	3	4	5	6	7
3	8					
4	8	7				
5	8	7	18:36			
6	8	7	18	5		
7	8	7	6	5	4	
8	8	7	6	5	4	9:54

$n = 10$							
v \a	2	3	4	5	6	7	8
3	n						
4	9	40					
5	9	8	n				
6	9	8	7	30			
7	9	8	7	6	n		
8	9	8	7	6	n	20	
9	9	8	7	6	n	4	3

$n = 11$								
v \a	2	3	4	5	6	7	8	9
3	10:11							
4	10	n						
5	10	9	8					
6	10	9	8	n				
7	10	9	8	7	6			
8	10	9	8	7	6	5:11		
9	10	9	8	7	6	5	4	
10	10	9	8	7	6	5	4	3

$n = 12$									
v \a	2	3	4	5	6	7	8	9	10
3	n								
4	11	12:60							
5	11	10	36:180						
6	11	10	36	24:72					
7	11	10	9	24:240	n				
8	11	10	9	24	7	n			
9	11	10	9	8	7	n	5:12		
10	11	10	9	8	7	n	5	n	
11	11	10	9	8	7	n	5	n	n

$n = 13$										
v \a	2	3	4	5	6	7	8	9	10	11
3	n									
4	12	n								
5	12	11	n							
6	12	11	10	n						
7	12	11	10	9	8					
8	12	11	10	9	8	n				
9	12	11	10	9	8	7	6			
10	12	11	10	9	8	7	6	5:13		
11	12	11	10	9	8	7	6	5	4	
12	12	11	10	9	8	7	6	5	4	3

$n = 14$											
v \a	2	3	4	5	6	7	8	9	10	11	12
3	n										
4	13	n									
5	13	12	n								
6	13	12	11	n							
7	13	12	11	10	n						
8	13	12	11	10	9	14:56					
9	13	12	11	10	9	8	n				
10	13	12	11	10	9	8	n	42			
11	13	12	11	10	9	8	n	6	5:14		
12	13	12	11	10	9	8	n	6	5	28	
13	13	12	11	10	9	8	n	6	5	4	3

$n = 15$												
v \a	2	3	4	5	6	7	8	9	10	11	12	13
3	n											
4	14	n										
5	14	13	n									
6	14	13	60	n								
7	14	13	12	11	30							
8	14	13	12	11	90:1260	n						
9	14	13	12	11	30:90	45	8:15					
10	14	13	12	11	30	9	8	n				
11	14	13	12	11	10	9	8	7	30			
12	14	13	12	11	10	9	8	7	30	n		
13	14	13	12	11	10	9	8	7	6	15:30	4	
14	14	13	12	11	10	9	8	7	6	n	4	n

$n = 16$													
v \a	2	3	4	5	6	7	8	9	10	11	12	13	14
3	n												
4	15	n											
5	15	14	n										
6	15	14	13	48									
7	15	14	13	48:960	n								
8	15	14	13	48	11	16:80							
9	15	14	13	12	11	10	n						
10	15	14	13	12	11	10	9	n					
11	15	14	13	12	11	10	9	n	n				
12	15	14	13	12	11	10	9	n	7	48			
13	15	14	13	12	11	10	9	n	7	6	5:16		
14	15	14	13	12	11	10	9	n	7	6	5	n	
15	15	14	13	12	11	10	9	n	7	6	5	n	3

TAB. 6.1 – *Résumé des expériences pour l'automate de type Accélération.*

Conclusion

L'automate cellulaire nous semble être un outil intéressant pour la modélisation des mosaïques végétales de part sa simplicité, et à plusieurs titres. Tout d'abord, l'automate cellulaire est parfaitement adapté à une utilisation informatique. Ses règles se prêtent naturellement à la programmation. Quelques "modules" écrit en langage C forment un ensemble simple à utiliser, à transporter (sur différents systèmes), et à faire évoluer; en effet, une programmation propre et commentée permet d'ajouter ses propres règles très rapidement, et d'utiliser les outils de représentation existant. Il ne nous semble donc pas nécessaire de développer des outils plus compliqués, plus "conviviaux", ce qui nécessiterait beaucoup de travail, tout en restreignant immanquablement le champs des possibilité, et au prix d'une perte de maîtrise de l'objet programmé : le langage C est sans ambiguïté, et nous permet de maîtriser toutes les étapes du modèle. De plus, la simplicité des règles et le caractère immédiat de sa représentation font de l'automate cellulaire un objet informatique très rapide, ce qui permet de conduire une expérimentation immédiate et abondante, afin de favoriser la construction d'une théorie adaptée au modèle (l'automate AVN étudié précédemment tourne à environ 200 itérations par seconde sur un PC/Pentium sous Linux).

Toutefois, cet argument ne doit pas nous faire mettre tous nos espoirs dans le développement d'une technologie qui permet chaque jours de faire fonctionner plus vite des modèles plus complexes. Un modèle n'a d'intérêt que si nous en avons une certaine compréhension. CH. Wissel écrit à ce sujet, dans [Wis91] :

> *High complex models which try to give a realistic description of whole ecosystems are not discussed here. They contain so many factors with hundreds of parameters that is impossible to appreciate all the model assumptions. These models cannot give an understanding of the system because it is impossible to access which detail of the model is responsible for a particular result. In addition, they are so developed to describe a very specific situation only. Therefore, they cannot provide general results. Thus, I do not consider these complex models in theoretical ecology.*
>
> *The aim of theoretical ecology is to provide an understanding of the mechanisms and functional relation in ecology, to find general statements. Therefore, simple models are used wich do not contain all the details of the system. Only key factors are included wich are essential for the particular question under consideration.*

Les automates cellulaires que nous avons étudié sont des modèles disposant de très peu de paramètres (a, v, n). Cela nous laisse à priori l'espoir de bien comprendre leur dynamique. C'était déjà le cas pour les paramètres $a = 1$ et $v = 2$, où, par les travaux de J.P. Allouche et C. Reder ([AR84]), nous disposions d'un certain nombre de théorèmes mathématiques. Malheureusement, le cas $a \geq 2$, c'est-à-dire le cas où il y a des états *jeunes*, semble plus difficile. Nous n'avons pas été capable d'établir de théorèmes... Faut-il pour autant attendre que cet automate AVN soit parfaitement compris au plan mathématique pour l'utiliser? Nous pensons avoir montré que non. Nous avons, à défaut de théorèmes, construit une "théorie", celle des *cycle auto-excitateurs de période* n, qui possède un assez bon pouvoir explicatif et prédictif. Ainsi, nous pensons avoir maintenant une bonne connaissance pratique de l'automate de type AVN. Une fois que le modèle est bien compris, on peut, comme nous avons commencé à le faire, le compliquer, y ajouter un caractère stochastique, petit à petit, ce qui permet de garder la maîtrise du modèle.

Forts de cette connaissance et des enseignements méthodologiques de notre étude, il pourrait être tentant d'aborder de façon systématique l'étude de tous les automates de type "*Mosaic Cycle*". Cette voie ne nous semble cependant pas devoir être empruntée. En effet, l'expérience montre que malgré la simplicité de l'automate cellulaire, on ne peut pas se contenter de quelques simulations. Ainsi, pour l'automate AVN, les cas non n-périodiques sont rares, et avant de tomber sur l'un d'entre eux, nous avions effectué de nombreuses simulations nous confortant dans l'idée que toutes les trajectoires issues de cet automate seraient de période n. De plus, pour un jeu de paramètres, on doit tester non seulement plusieurs conditions initiales, mais plusieurs types de conditions initiales, ce qui introduit de nouveaux paramètres, par exemple, dans notre cas, le pourcentage de cellules en 0 dans la condition initiale. Plus généralement, même si l'on se contente de "théories" du type de celles que nous avons développé pour l'automate AVN, on ne peut pas envisager de faire une étude exhaustive de tous les automates cellulaires de type "*Mosaic Cycle*", et une étude satisfaisante d'un seul d'entre eux prendra déjà beaucoup de temps.

De plus, les règles particulières que nous avons étudié se concentrent essentiellement sur l'aspect recrutement d'une dynamique forestière. Mais leur valeur tient plus dans l'apprentissage suscité par leur étude que dans leurs significations propres. C'est pourquoi il nous parait nécessaire, maintenant que nous avons acquis un certain "savoir faire" face à ce type de modèles, de concevoir des règles d'automate en accord avec les préoccupations de "*terrain*". Par exemple, dans l'article [RPH98], les auteurs étudient un périmètre de forêt naturelle de Guyane française partagé en quadrats élémentaires de 5×5 mètres, dont la structure est supposée homogène, caractérisés par des types structuraux, dix au total. Le diagramme de flux (figure 6.6) indique des probabilités plus ou moins fortes de passer d'un type à l'autre. Une telle étude pourrait donner lieu à l'élaboration de règles de type automate cellulaire, à condition de "*spatialiser*" ces données. En effet, dans cet article, on comptabilise le nombre de quadrats qui passent d'un type structural à un autre. Il serait intéressant d'essayer de mettre en relation ces passages avec la nature du voisinage de chaque quadrat. Les auteurs écrivent d'ailleurs :

> *Ce type de modèle de dynamique forestière (matrice de transition) présente cependant l'inconvénient de reposer sur un postulat d'indépendance des quadrats,*

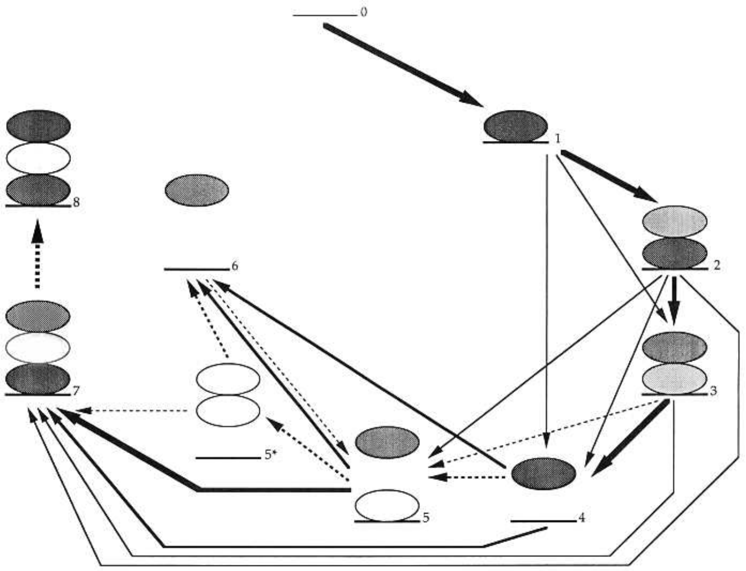


FIG. 6.6 – *Diagramme des flux entre types structuraux observés entre 1981 et 1991. L'épaisseur des flèches indique la probabilité de transition et les pointillés, des transitions possibles mais non observées. Les transitions associées aux chablis ou à la mort des grands arbres sur pied n'ont pas été figurées.*

> *alors que sur le transect étudié, il apparaît assez nettement que les chablis induisent des couplages spatiaux forts entre quadrats contigus. C'est donc assez naturellement vers des modèles résolument spatiaux et temporels qu'il convient de s'orienter pour décrire la dynamique de la mosaïque forestière : les automates cellulaires semblent une bonne piste.*

Nous pensons que notre travail montre que l'utilisation d'automates cellulaires pour la modélisation de mosaïques végétales est effectivement envisageable, à condition qu'elle soit considérée avec méthode, dans le cadre d'un programme scientifique d'une certaine durée. Une utilisation anecdotique risquerait d'être plus nuisible qu'utile, laissant croire à l'existence de propriétés qui pourraient n'être que des artefacts de simulation. L'ensemble formé par la classe d'Automates Cellulaires *Mosaic Cycle*, les différents "modules" écrit en C à l'occasion de ce travail, ainsi que les méthodes abordées ici, nous paraissent être un bon point de départ pour ce genre d'étude.

Annexe A

Étude statistique de la structuration spatiale de l'Eperua falcata

Nous aborderons ici l'étude statistique de la structure spatiale d'une espèce d'arbre (Eperua Falcata) dans son contexte forestier. Les données sur lesquelles s'appuient cette étude ont été fournies par Bernard Riéra, et se présentent comme suit:

Liste des Eperua Falcata		
diamètre en cm	X en cm	Y en cm
22.6	2	26
50.3	97	240
17.5	26	306
etc...		

La littérature abordée [Rip81, UF85, Cre93] propose plusieurs méthodes, chacune adaptée à un certain type de données. Mais la méthode qui semble ici la plus appropriée (ainsi que la plus utilisée) est la fonction de Ripley. En effet, c'est une des rares approches qui permette d'intégrer toutes les échelles de structurations spatiales présentes. Elle nous permettra dans un premier temps de comparer nos données à un processus de Poisson, puis d'avoir une idée plus précise quand à la structure spatiale des données.

A.1 La fonction de Ripley

La fonction de Ripley, que l'on nommera K est très simple : c'est le nombre moyen d'individus à une distance h de chaque individu, divisé par la densité :

$K(h) = \lambda^{-1}E(\text{nombre d'individus distants de } h \text{ d'un individu arbitraire}), h \geq 0$

Nous prendrons $\hat{\lambda} = \frac{\text{nombre d'individus}}{\text{Surface considérée}}$ comme estimateur de la densité. Mais il reste le problème du bord : en effet, nous ne sommes pas certains de connaître tous les voisins des points trop prés du bord de la zone considérée. Ici aussi, plusieurs méthodes sont proposées :

– éliminer les points trop prés du bord : perte de données.

– travailler sur un tore : cette méthode ne fonctionne que si la zone considérée est un rectangle.

Nous retiendrons la méthode proposée par Ripley [Rip81] :

$$\hat{K}(h) = \hat{\lambda}^{-1}\sum_{i=1}^{N}\sum_{j=1\,i\neq j}^{N} w(s_i,s_j)^{-1}I(||s_i - s_j|| \leq h)/N,h > 0 \tag{A.1}$$

avec $I(P)$ égal à un si P est vrai, et $w(s_i,s_j)$ la proportion de la circonférence du cercle centré en s_i, passant par s_j, à l'intérieur de la zone considérée. En fait, w est une pondération qui permet en quelque sorte de *"deviner"* les points qui sont en dehors de la zone étudiée. Si l'on suppose la stationnarité et l'isotropie de nos données, $\hat{K}$ est un estimateur non-biaisé de K quand λ est connu. Il semble que nous pouvons utiliser sans crainte $\hat{\lambda}$ ([Rip81], p.159).

Si nous avons à faire à un processus de Poisson (hasard pur), alors $K(h) = \pi h^2,h \geq 0$. En présence de régularité, $K(h)$ sera inférieur à πh^2, et en présence d'agrégats (clusters), $K(h)$ sera supérieur à πh^2 ([Cre93] p.616). La méthode étant choisie, il nous reste à la programmer.

A.2 Programmation

La programmation de la fonction de Ripley ne pose pas de problème technique particulier, si ce n'est que pour éviter de calculer plusieurs fois les même distances, il vaut mieux calculer dans un premier temps la distance de chaque point à tous les autres ([Gor95]). La difficulté réside en la fonction w de pondération : comment calculer la proportion d'un cercle intérieure à la zone étudiée? Quelque approches analytiques ont été effectuées ([Gor95]), mais elles nécessitent des zones d'études simples (rectangles, voir triangles) : il est difficile de parler alors de polygones quelconques (non-convexes ou à trous). Mais sur le terrain, les relevés coûtent chers, et n'étudier qu'un rectangle inscrit dans la zone de données peut nous emmener à disqualifier trop de points : c'est le cas pour notre jeu de données. Nous allons donc adopter une approche discrète, qui consiste à poser sur la circonférence du cercle un certain nombre de points (figure A.1). En comptant les points à l'intérieur de la zone étudiée, puis en divisant par le nombre total de points posés, nous aurons une approximation de w, d'autant meilleure que le nombre de points posés est grand. Il suffit maintenant de pouvoir déterminer si un points est à l'intérieur de la zone étudiée, ce qui ne posera pas trop de problèmes.

A.3 Calculs

Le jeu de données en question comporte plusieurs *"paquets"* d'arbres très distincts (voir Figure A.2). Nous allons donc, afin de respecter les contraintes d'isotropie et de stationnarité, travailler sur l'un d'entre eux, en l'occurrence le plus important : celui qui ce situe

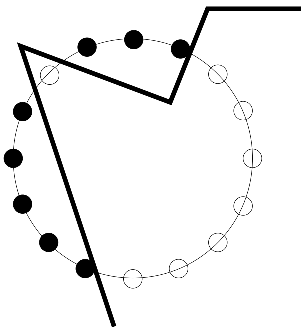


FIG. A.1 – *On pose un certain nombre de points sur la circonférence du cercle.*

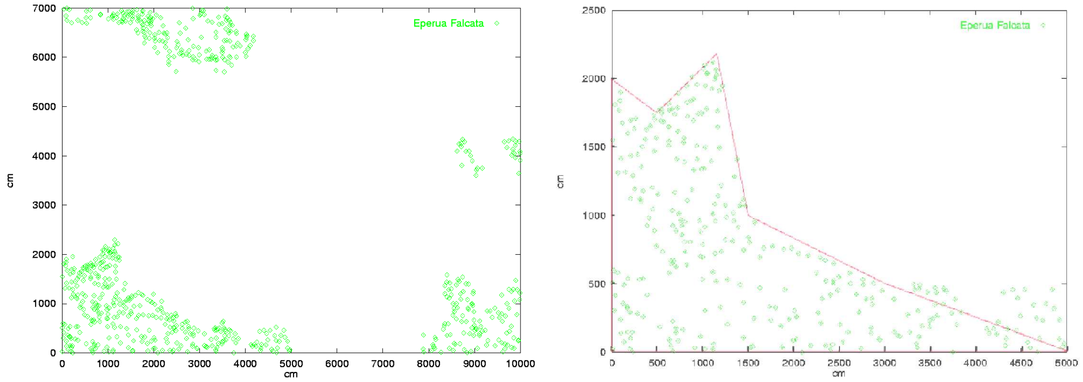


FIG. A.2 – *Notre forêt d'Eperua Falcata, puis un zoom sur la région considérée*

en-bas à gauche de la carte (voir Figure A.2). Les contours de cette zone ont été déterminés "à l'œil".

Nous allons calculer la fonction $\hat{L}(h) = (\hat{K}(h)/\pi)^{1/2} - h$, qui à l'avantage d'être plus facilement interprétable que la fonction de Ripley, et qui de plus permet la comparaison de processus de densités différentes. La fonction $\hat{L}(h)$ s'annule pour un processus de Poisson infini, est négative en présence de régularité et positive en présence d'agrégats. Cette fonction sera comparée à l'enveloppe supérieure (-5%) et inférieure (+5%) issues de 1000 simulation d'un processus de Poisson de densité égale, sur la même surface.

En ce qui concerne la correction aux bords, nous calculerons w avec 80 points sur le cercle, ce qui concilie apparemment le temps de calcul, et la précision nécessaire.

A.4 Résultats

La courbe obtenue (figure A.3) sort clairement de l'intervalle de confiance calculé, ce qui nous permet de rejeter avec un risque de 10% l'hypothèse que nos arbres sont distribués aux hasard. De plus la sortie se fait par le haut, entre $h = 100$ et $h = 400$, signe d'agrégats de rayon compris entre 1 et 4 mètres.

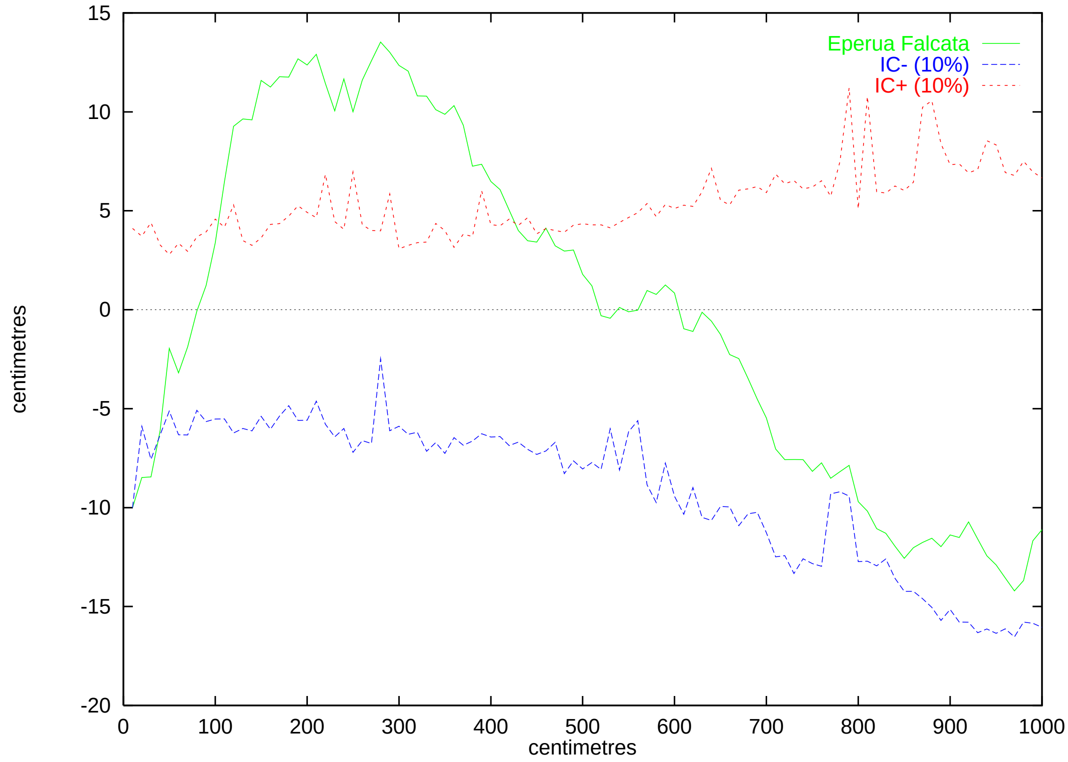
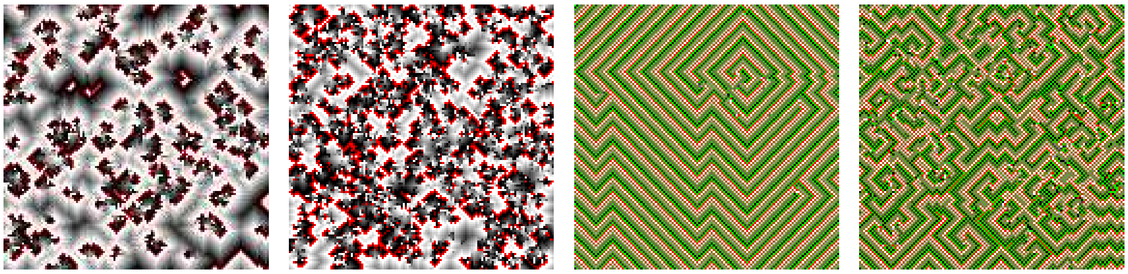


Fig. A.3 – *Fonction de Ripley*

Annexe B

Les ponts cassés

Nous avons pu constater que les figures produites par l'automate AVN ont en général un caractère très *géométrique*. Lorsqu'une structuration spatiale apparaît, elle est souvent contituée de *spirales en losanges* ou de *losanges concenriques*. La figure B.1 représente quelques exemples de structurations spatiales obtenues avec l'automate AVN, avec n, le nombre d'états égal à 16. La condition initiale est tirée au hasard, tous les états étant équi-probables pour chaque cellule.

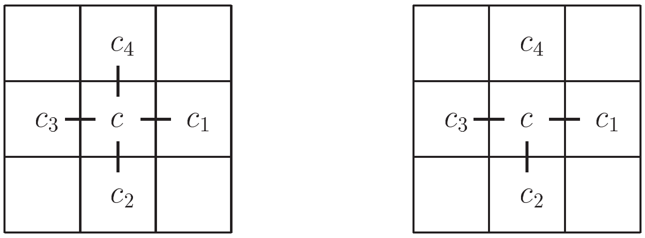


(a) $a = 1$; $v = 3$ (b) $a = 1$; $v = 4$ (c) $a = 9$; $v = 10$ (d) $a = 9$; $v = 11$

FIG. B.1 – *Des structures spatiales "en losanges".*

Dans cette section, nous tentons de briser cette régularité, tout en préservant l'aspect déterministe de l'automate. En effet, un tel effet peut être obtenu en introduisant du *hasard* dans les règles même du modèle (4.4.5), mais l'automate obtenu est alors stochastique.

Les règles utilisées seront donc celles de l'automate AVN, et nous agiront sur le voisinage. Considérons que la notion de voisinage soit matérialisée par un pont entre deux cellules voisines. Imaginons que nous "brisions" un certain nombre de ces ponts au hasard. Deux cellules précédemment voisines pourraient alors ne plus l'être (figure B.2).



(a) $\mathrm{V}(c) \;=\; \{c_1,c_2,c_3,c_4\}$ (b) $\mathrm{V}(c) \;=\; \{c_1,c_2,c_3\}$

FIG. B.2 – *Les ponts cassés*

Oservons maintenant l'effet des "ponts cassés" sur la propagation d'une onde de proche en proche (figures B.3), simulée par l'automate suivant :

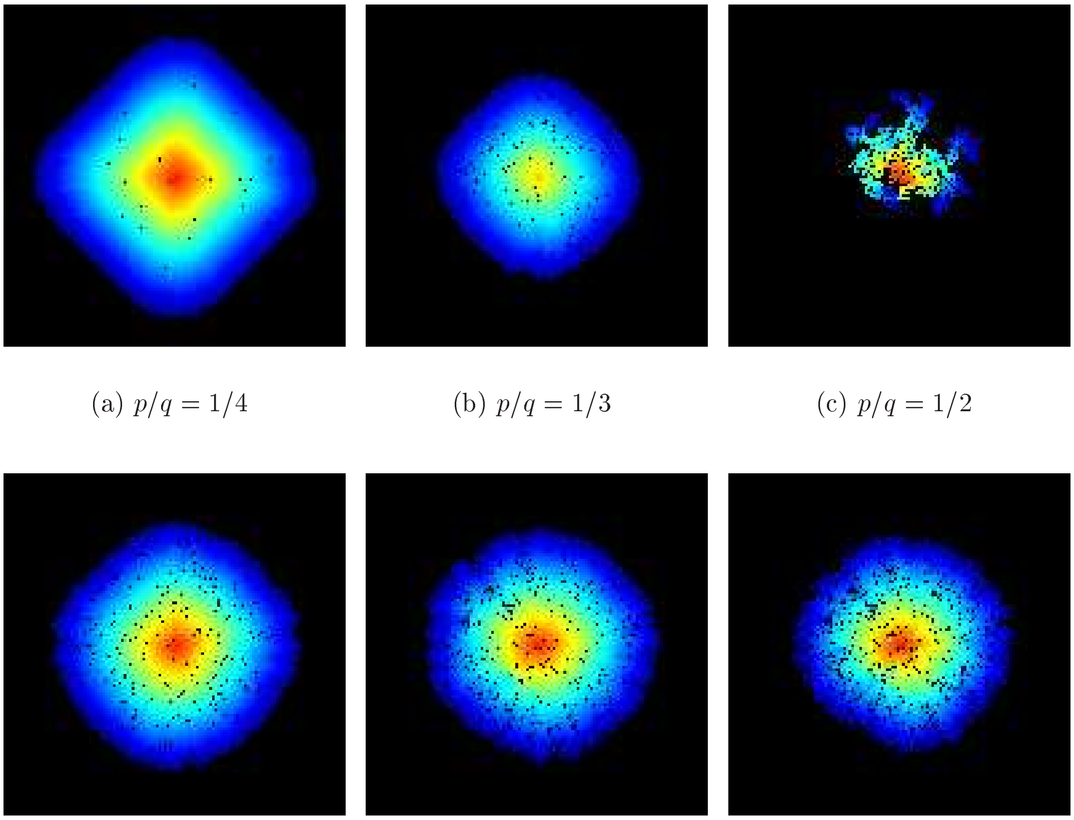
`si` $\mathcal{A}(c,t) \;\neq\; 0$
`alors` $\mathcal{A}(c,t+1) = \mathcal{A}(c,t) + 1$
`sinon`
  `si` $\exists\ d \;\in\; \mathcal{V}(c)$ telle que $\mathcal{A}(d,t) \;=\; 1$
  `alors` $\mathcal{A}(c,t+1) = 1$
  `sinon` $\mathcal{A}(c,t+1) = \mathcal{A}(c,t) = 0$

---

avec comme condition initiale une cellule en 1 au centre du domaine. Nous ferons varier la proportion de ponts cassés, que nous appelerons p.

Le but de cette expérience étant d'obtenir des "ondes" d'aspect circulaire, nous pouvons constater que $p = 12/30 = 0{,}4$ semble donner un bon résultat. Cependant, il apparaît sur les images des "trous" : il sont dûs au fait que certaines cellules (ou eventuellement groupes de cellules) se trouvent isolées du fait que tous les ponts vers leurs voisines sont coupés. Mais globalement, on obtient des ondes de forme circulaire. La figure B.4 représente les mêmes expériences que celles illustrées par la figure B.1, en appliquant une proportion $p = 0{,}4$ de ponts cassés.

Les structures obtenues avec les ponts cassés sont semblables aux autres, mais adoptent des formes plus circulaires. On peut cependant remarquer des différences significatives. Par exemple, pour le cas $\{a = 9;\, v = 10\}$, l'expérience "avec les ponts cassés" montre clairement une spirale, dont la largeur est supérieure à celle observée pour l'expérience initiale. De même, par le fait que le nombre de voisins diminue, dans le cas $\{a = 1;\, v = 2\}$, la trajectoire produite avec les ponts cassés meurt, ce qui n'est pas le cas dans l'expérience originale. Malgrès ces petites différences, il semble que la technique des "ponts cassés" soit efficace pour briser l'effet "losange", dû à la nature même de l'objet automate cellulaire, tout en en

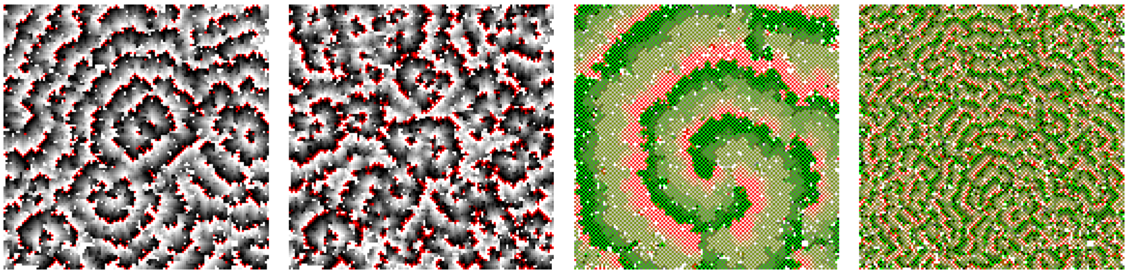


(a) $p/q = 1/4$ (b) $p/q = 1/3$ (c) $p/q = 1/2$

(d) $p/q = 11/30$ (e) $p/q = 12/30$ (f) $p/q = 13/30$

FIG. B.3 – *Illustration des ponts cassés: p/q est la proportion de ponts cassés.*

conservant l'aspect déterministe. Pour finir, les figures B.5, B.6 et B.7 montrent l'intégralité des expériences effectuées avec $n = 16$ et $p = 0,4$, avec une condition initiale au hasard telle que tous les états sont équi-probables. Cette série d'expériences peut être comparée avec les figures 4.18, 4.19 et 4.20, qui représentent la même série, sans pont cassé.



(a) $a = 1$; $v = 3$ (b) $a = 1$; $v = 4$ (c) $a = 9$; $v = 10$ (d) $a = 9$; $v = 11$

FIG. B.4 – *Mêmes expériences que pour la figure B.1, avec* 40% *de ponts cassés*

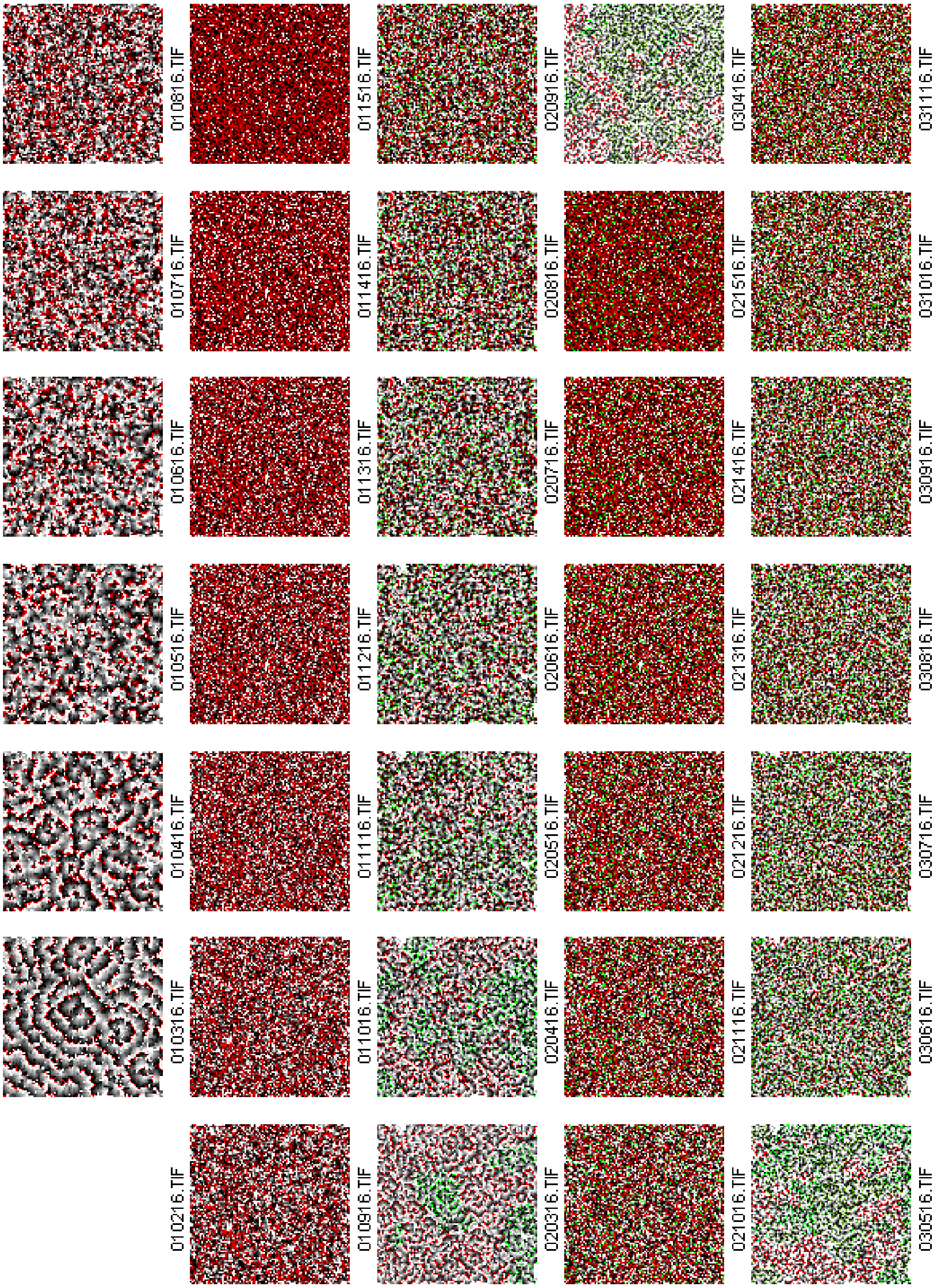


FIG. B.5 – $n = 16$ *et* $z = 9/10$ *: première partie*

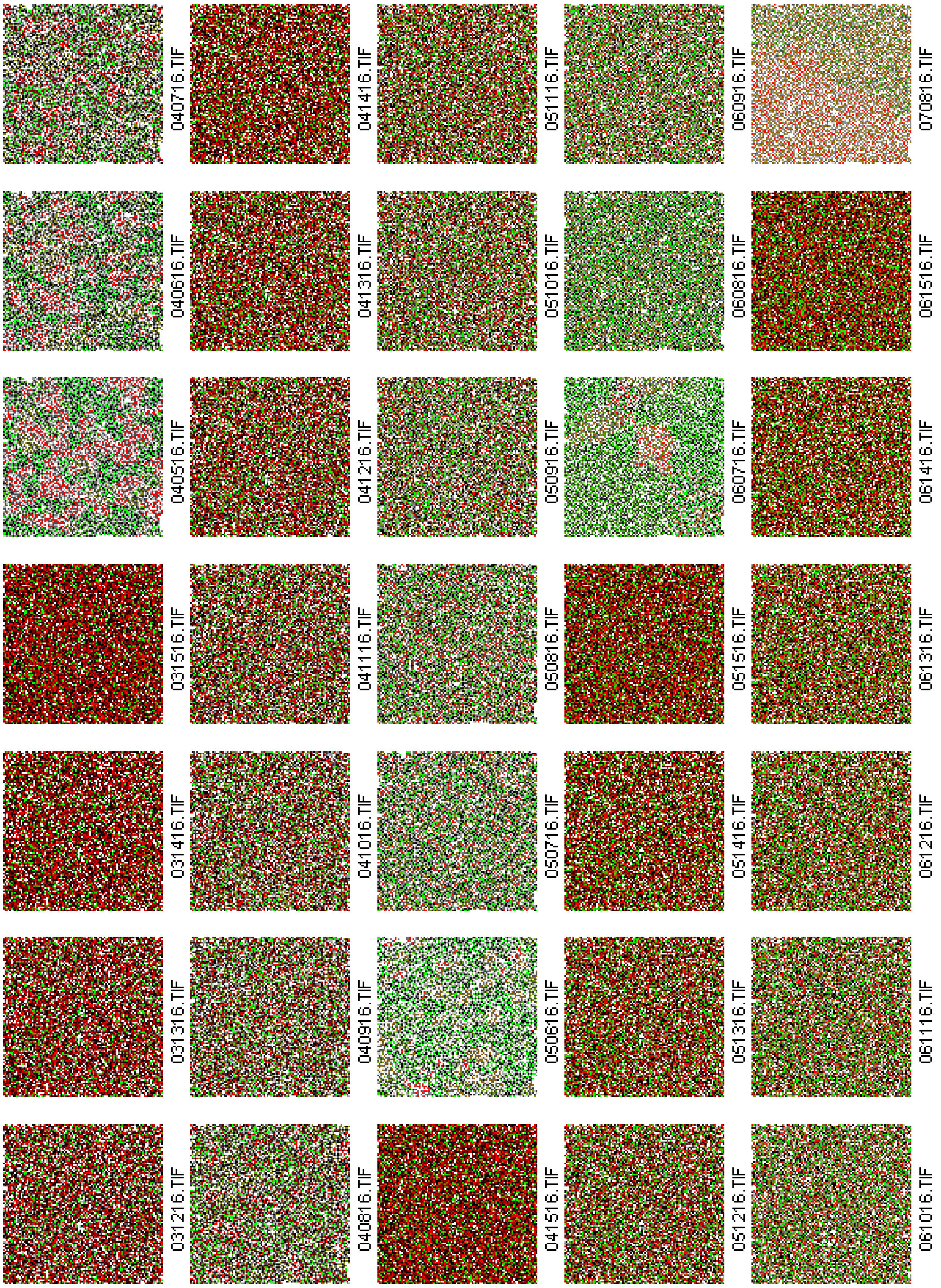


FIG. B.6 – $n = 16$ *et* $z = 9/10$ *: deuxième partie*

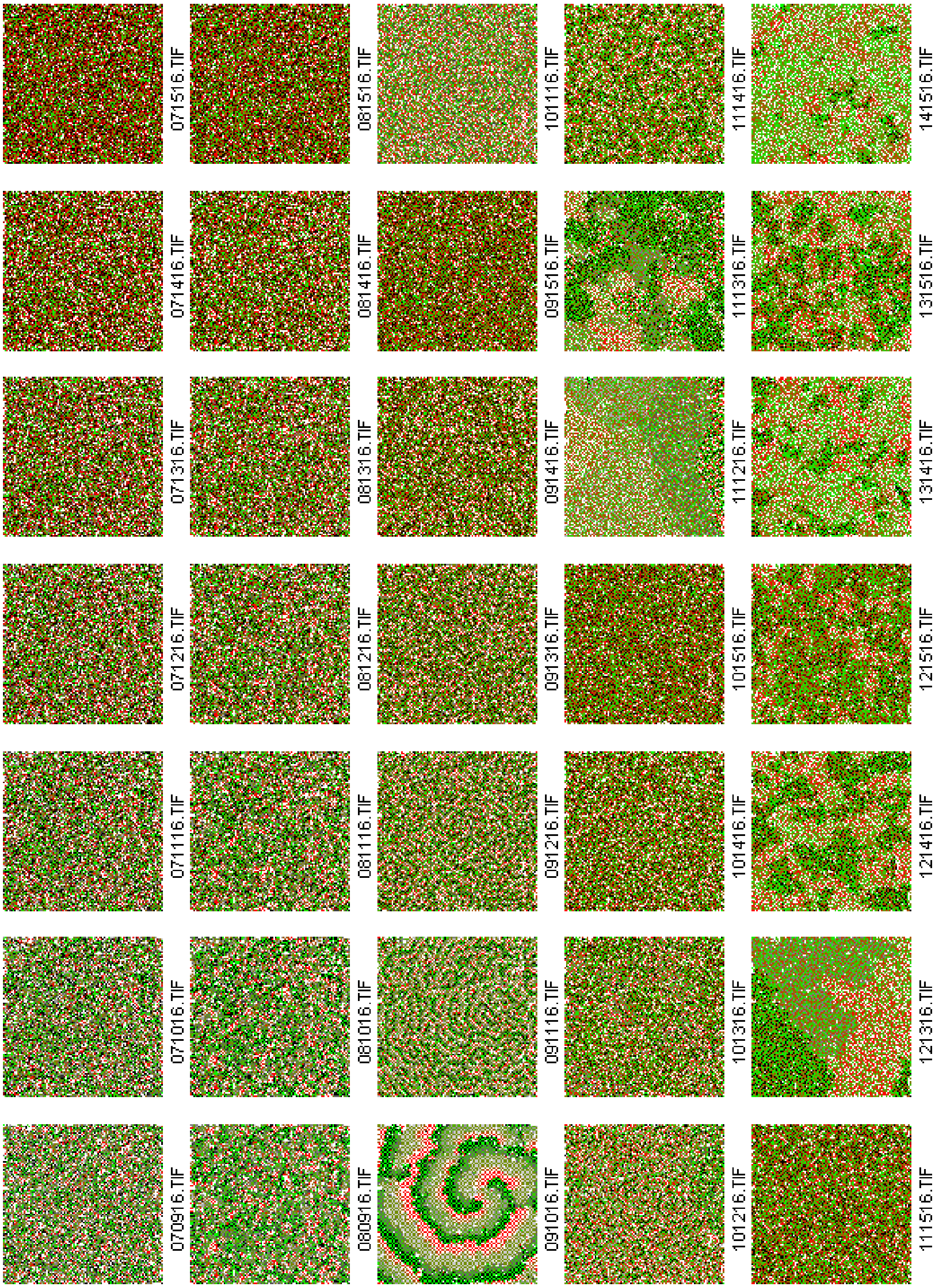


FIG. B.7 – $n = 16$ *et* $z = 9/10$ *: troisième partie*

Annexe C

Programmation d'un Automate Cellulaire en C

L'automate cellulaire est un objet mathématique qui se prête très bien à la programmation informatique. Il a pour support un tableaux d'entiers, qui s'actualise à chaque itération, en fonction du contenu du tableau à l'itération précédente. Mais encore faut-il prendre quelques précautions, d'une part pour que le programme obtenu modélise bien l'objet désiré, d'autre part, pour que le programme soit "efficace". Le langage utilisé sera le langage C.

C.1 Un programme juste

L'automate cellulaire aura comme support un tableau d'entiers :

```
unsigned char TAB[HMAX][WMAX];
```

Supposons que nous ayons déjà écrit une fonction `Voisin(i,j)` qui renvoie `TRUE` si la cellule `TAB[i][j]` possède au moins un voisin adulte, `FALSE` sinon. Nous pourrions être tentés de programmer l'automate de la façon suivante :

```
  for (i=1; i<=w; i++)
    for (j=1; j<=h; j++) {
      if (TAB[i][j]==0) {
        if (Voisin(i,j)) TAB[i][j]=1;
      }
      else TAB[i][j]=(TAB[i][j]+1)%n;
   }
```

Ce serait une erreur ! En effet, toutes les cellules doivent être actualisées "en même temps", car l'état de chaque cellule à l'étape $t+1$ dépend de son propre état et de celui de ses voisines à l'étape t. Pour se faire, on utilisera deux tableaux, qui contiendront l'ancien et le nouvel état.

```
unsigned char OLD[HMAX][WMAX], NEW[HMAX][WMAX];
```

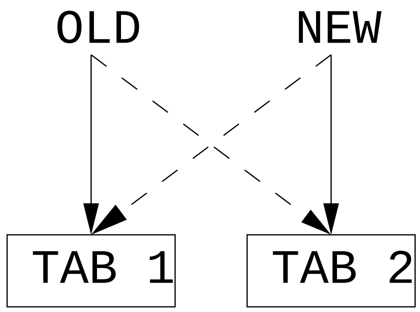


FIG. C.1 – *La technique des pointeurs nous permet d'intervertir les adresses.*

Ce qui nous donne l'algorithme suivant :

```
for (i=1; i<=w; i++)      /*---------------------------*/
  for (j=1; j<=h; j++)    /* Transfert de NEW dans OLD */
    OLD[i][j]=NEW[i][j]; /*---------------------------*/
for (i=1; i<=w; i++) {
  for (j=1; j<=h; j++)
    if (OLD[i][j]==0) {
      if (Voisin(i,j)) NEW[i][j]=1;
    }
    else NEW[i][j]=(OLD[i][j]+1)%n;
}
```

Cet algorithme modélise bien l'objet considéré, mais il n'est pas efficace.

C.2 Un programme efficace

Tout d'abord, la phase de transfert du contenu du tableau `NEW` dans le tableau `OLD` peut être évitée. En effet, la technique des pointeurs (manipulations des adresses des objets en mémoires), nous permet d'intervertir les adresses des deux tableaux (figure C.1). Ainsi, le tableau pointé par `OLD` sera alternativement `TAB1` ou `TAB2`, et on remplace le code correspondant par :

```
TMP=OLD;
OLD=NEW;
NEW=TMP;
```

De plus, en C, lorsqu'on déclare un tableau à deux dimensions `int TAB[HMAX][WMAX]`, on alloue en fait en mémoire $HMAX \times WMAX$ emplacements de la taille d'un entier. Ainsi, pour accéder à l'élément `TAB[i][j]`, on effectue une multiplication et une addition : $i \times WMAX + j$. Puisque nous traitons chaque cellule successivement, nous pouvons éviter ces opérations simplement en incrémentant l'adresse d'une cellule, pour passer à la suivante. En prenant quelques précautions pour le passage à la ligne suivante, on obtient le code suivant :

```
TMP=NEW;
```

```
  NEW=OLD;
  OLD=TMP;
  newaux=NEW;oldaux=OLD;
  for (i=1; i<=w; i++) {
    for (j=1; j<=h; j++){
      if ((*oldaux)==0) {
        if (Voisin(oldaux,a,v)!=0)
          *newaux=1;
        else *newaux=0;
      }
      else *newaux=((*oldaux+1)%n);
      ++oldaux; ++newaux;
    }
    oldaux+=(WMAX-w); newaux+=(WMAX-w);
  }
```

Afin de les comparer, nous avons testé l'algorithme naïf et l'algorithme définitif. Pour 10000 itérations, l'algorithme naïf prend 83 secondes, pour seulement 46 secondes avec l'algorithme définitif, soit pratiquement deux fois moins de temps, ce qui n'est pas négligeable lorsqu'il s'agit d'observer la dynamique du modèle "en direct".

C.3 Le Code C

Pour clore cette annexe, nous y joignons le fichier contenant les différents automates étudiés, et un exemple de fichier principal.

```
#include <math.h>
#include <stdlib.h>

#define HMAX 1026
#define WMAX 1026
#define NMAX 256

float Myrand(void);
void Swap(void);
void InitCells(int,int,float);
int Voisin(unsigned char *,int,int);
void AVN(int,int,int);
void ProbaRien(int,int,int,float);
void ProbaSaufZero(int,int,int,float);
void ProbaZero(int,int,int,float);
void ProbaMort(int,int,int,float);
void Accel(int,int,int);
```

```
int Cpt(int, int, int, int, int);
void WriteCells(void);

/*-----------------------------------------------------------------------------\
| Les automates ont pour support les tableaux appelles TAB1 et TAB2. Pour      |
| regler les problemes dus a la programmation des bords, TAB1 et TAB2 sont au  |
| moins de dimensions (h+2)X(w+2). Les bords sont utilises comme tampons et    |
| contiennent des valeurs immuables. Les variables OLD et NEW contiennent les  |
| adresses des premieres cellules (TAB[1][1])                                  |
\-----------------------------------------------------------------------------*/
unsigned char TAB1[HMAX][WMAX];
unsigned char TAB2[HMAX][WMAX];
unsigned char *NEW,*OLD;
int h,w;
float rcolor[NMAX], gcolor[NMAX], bcolor[NMAX];

#include "printac.c"

/*-----------------------------------------------------------------------------\
| Retourne un flottant entre 0 et 1.                                           |
\-----------------------------------------------------------------------------*/
float Myrand()
{
  return(rand()*1.0/RAND_MAX);
}


void Swap(void)
{
  unsigned char *TMP;

  TMP=NEW;
  NEW=OLD;
  OLD=TMP;
}

/*-----------------------------------------------------------------------------\
| Initialise TAB1 et TAB2. OLD pointe sur TAB1 et NEW sur TAB2.                |
\-----------------------------------------------------------------------------*/
void InitCells(int n, int seed, float proba)
{
  int i,j;
```

```
  srand(seed);

  for (i=1; i<=w; i++)
    for (j=1; j<=h; j++)
      TAB2[i][j]=0;

  for (j=2; j<h; j++)  for (i=2; i<w; i++)
    if (Myrand() < proba)
      TAB2[i][j]=0;
    else TAB2[i][j]=Myrand()*(n-1)+1;

  for (i=0; i<=w+1; i++) {
    TAB2[i][0]=n+1;
    TAB2[i][h+1]=n+1;
  }
  for (j=1; j<=h; j++) {
    TAB2[0][j]=n+1;
    TAB2[w+1][j]=n+1;
  }
  for (i=0; i<=w+1; i++) {
    TAB1[i][0]=n+1;
    TAB1[i][h+1]=n+1;
  }
  for (j=1; j<=h; j++) {
    TAB1[0][j]=n+1;
    TAB1[w+1][j]=n+1;
  }

  OLD=&(TAB1[1][1]);
  NEW=&(TAB2[1][1]);
}


/*----------------------------------------------------------------------------\
| Retourne le nombre de voisin de *c compris entre a et v.                    |
\----------------------------------------------------------------------------*/
int Voisin(unsigned char *c, int a, int v)
{
  int res;
  unsigned char aux;

  aux = *(c+1);
  res = (aux>=a)&&(aux<v);
```

```
  aux = *(c-1);
  res += (aux>=a)&&(aux<v);
  aux = *(c+WMAX);
  res += (aux>=a)&&(aux<v);
  aux = *(c-WMAX);
  res += (aux>=a)&&(aux<v);
  return(res);
}


/*------------------------------------------------------------------------------\
| C'est l'auotmate AVN. Une cellule reste en zero jusqu'a qu'une de ses         |
| voisines soit excitatrice (comprise entre a et v-1); elle passe alors en 1.  |
| Pour les autres etats, on passe a l'etat suivant a chaque iteration et       |
| n-1 -> 0.                                                                     |
\------------------------------------------------------------------------------*/
void AVN(int a, int v, int n)
{
  int i,j;
  unsigned char *newaux, *oldaux;
  unsigned char *TMP;

  TMP=NEW;
  NEW=OLD;
  OLD=TMP;
  newaux=NEW;oldaux=OLD;
  for (i=1; i<=w; i++) {
    for (j=1; j<=h; j++){
      if ((*oldaux)==0) {
        if (Voisin(oldaux,a,v)!=0)
          *newaux=1;
        else *newaux=0;
      }
      else *newaux=((*oldaux+1)%n);
      ++oldaux; ++newaux;
    }
    oldaux+=(WMAX-w); newaux+=(WMAX-w);
  }
}



/*------------------------------------------------------------------------------\
```

```
| Regle les couleurs pour l'automate AVN.                                      |
| 0     : blanc.                                                               |
| 1..a-1: de vert a rouge.                                                     |
| a..v-1: de rouge a noir.                                                     |
| v..n-1: de noir a blanc.                                                     |
\-----------------------------------------------------------------------------*/
void AVNColor(int a, int v, int n)
{
  int i;

  rcolor[0]=1;
  gcolor[0]=1;
  bcolor[0]=1;
  for (i=1; i<a; i++) {
    rcolor[i]=1.0*(i-1)/(a-1);
      gcolor[i]=1-1.0*(i-1)/(a-1);
    bcolor[i]=0;
  }
  for (i=a; i<v; i++) {
    rcolor[i]=1-1.0*(i-a)/(v-a);
    gcolor[i]=0;
    bcolor[i]=0;
  }
  for (i=v; i<n; i++) {
    rcolor[i]=1.0*(i-v)/(n-v);
    gcolor[i]=1.0*(i-v)/(n-v);
    bcolor[i]=1.0*(i-v)/(n-v);
  }
  for (i=n; i<NMAX; i++) {
    rcolor[i]=0;
    gcolor[i]=0;
    bcolor[i]=1;
  }
}


/*-----------------------------------------------------------------------------\
| C'est le meme automate que AVN, sauf que chaque cellule a, a chaque          |
| iteration une probabilite (1-proba) de ne rien faire, quoi qu'il arrive.     |
\-----------------------------------------------------------------------------*/
void ProbaRien(int a, int v, int n, float proba)
{
  int i,j;
```

```
  unsigned char *newaux, *oldaux;

  Swap();
  newaux=NEW;oldaux=OLD;
  for (i=1; i<=w; i++) {
    for (j=1; j<=h; j++){
      if (Myrand() < proba) {
        if ((*oldaux)==0) {
          if (Voisin(oldaux,a,v)!=0)
            *newaux=1;
          else *newaux=0;
        }
        else *newaux=((*oldaux+1)%n);
      }
      else *newaux=*oldaux;
      ++oldaux; ++newaux;
    }
    oldaux+=(WMAX-w); newaux+=(WMAX-w);
  }
}


/*-----------------------------------------------------------------------------\
| C'est le meme automate que AVN, sauf que chaque cellule dans un etat non nul |
| a, a chaque iteration, une probabilite (1-proba) de ne rien faire.           |
\-----------------------------------------------------------------------------*/
void ProbaSaufZero(int a, int v, int n, float proba)
{
  int i,j;
  unsigned char *newaux, *oldaux;

  Swap();
  newaux=NEW;oldaux=OLD;
  for (i=1; i<=w; i++) {
    for (j=1; j<=h; j++){
      if ((*oldaux)==0) {
        if (Voisin(oldaux,a,v)!=0)
          *newaux=1;
        else *newaux=0;}
      else
        if (Myrand() < proba)
          *newaux=((*oldaux+1)%n);
        else *newaux=*oldaux;
```

```
      ++oldaux; ++newaux;
    }
    oldaux+=(WMAX-w); newaux+=(WMAX-w);
  }
}


/*-------------------------------------------------------------------------------\
| C'est le meme que AVN, sauf que chaque cellule en 0 a une chance (1-proba) de|
| ne rien faire, quoi qu'il arrive.                                              |
\-------------------------------------------------------------------------------*/
void ProbaZero(int a, int v, int n, float proba)
{
  int i,j;
  unsigned char *newaux, *oldaux;

  Swap();
  newaux=NEW;oldaux=OLD;
  for (i=1; i<=w; i++) {
    for (j=1; j<=h; j++){
      if ((*oldaux)==0) {
        if ((Myrand() < proba) && (Voisin(oldaux,a,v)!=0))
          *newaux=1;
        else *newaux=0;
      }
      else *newaux=((*oldaux+1)%n);
      ++oldaux; ++newaux;
    }
    oldaux+=(WMAX-w); newaux+=(WMAX-w);
  }
}


/*-------------------------------------------------------------------------------\
| C'est le meme que AVN, sauf que chaque cellule a une chance (1-proba) de      |
| passer a l'etat 0, quoi qu'il arrive.                                          |
\-------------------------------------------------------------------------------*/
void ProbaMort(int a, int v, int n, float proba)
{
  int i,j;
  unsigned char *newaux, *oldaux;

  Swap();
```

```
  newaux=NEW;oldaux=OLD;
  for (i=1; i<=w; i++) {
    for (j=1; j<=h; j++){
      if (Myrand() < proba) {
        if ((*oldaux)==0) {
          if (Voisin(oldaux,a,v)!=0)
            *newaux=1;
          else *newaux=0;
        }
        else *newaux=((*oldaux+1)%n);
      }
      else *newaux=0;
      ++oldaux; ++newaux;
    }
    oldaux+=(WMAX-w); newaux+=(WMAX-w);
  }
}


/*-------------------------------------------------------------------------------\
| Cet automate est une simplification de l'automate de Wissel.                  |
| Une cellule passe a chaque iteration a l'etat suivant (mod n) sauf si elle    |
| est dans l'etat neutre et qu'elle a au moins une voisine excitatrice : elle   |
| devient alors excitatrice a son tour.                                         |
| Parametres:                                                                   |
|  .a : premier excitateur.                                                     |
|  .v : dernnier excitateur.                                                    |
|  .n : le nombre d'etats.                                                      |
\-------------------------------------------------------------------------------*/
void Accel(int a, int v, int n)
{
  int i,j;
  unsigned char *newaux, *oldaux;

  Swap();
  newaux=NEW;oldaux=OLD;
  for (i=1; i<=w; i++) {
    for (j=1; j<=h; j++){
      if ((*oldaux)==0) {
        if (Voisin(oldaux,a,v)!=0)
          *newaux=a;
        else *newaux=1;
      }
```

```
      else *newaux=((*oldaux+1)%n);
      ++oldaux; ++newaux;
    }
    oldaux+=(WMAX-w); newaux+=(WMAX-w);
  }
}


/*---------------------------------------------------------------------------\
| Compte le nombre de cellules dans l'etat e dans le rectangle definit par les |
| points (x1,y1) et (x2,y2).                                                   |
\---------------------------------------------------------------------------*/
int Cpt(int e, int x1, int y1, int x2, int y2)
{
  int i,j,cpt;
  unsigned char *newaux;

  newaux=NEW+WMAX*(y1-1)+(x1-1);
  cpt=0;
  for (j=y1; j<=y2; j++){
    for (i=x1; i<=x2; i++) {
      if ((*newaux)==e) ++cpt;
      ++newaux;
    }
    newaux+=WMAX-x2+x1-1;
  }
  return(cpt);
}
```

Et voici un exemple de fichier principal :

```
#include "ac.c"


int main (int argc, char** argv)
{
  int a,v,n,seed,pv,zoom;
  float z;
  int i;

  /* Check usage. */
  if (argc < 8) {
    fprintf(stderr, "usage: %s a v n zero seed pv zoom\n",argv[0]);
```

```
    return 1;
  } else if (argc > 8)
    fprintf(stderr, "%s: extra options ignored.\n", argv[0]);

  a = atoi(argv[1]);
  v = atoi(argv[2]);
  n = atoi(argv[3]);
  z = atof(argv[4]);
  seed = atoi(argv[5]);
  pv = atoi(argv[6]);
  zoom = atoi(argv[7]);
  w=h=128;
  AVNColor(a,v,n);
  Open(zoom);
  InitCells(n,seed,z);
  PrintCells(zoom);

  Wait4Mouse();

  while (1) {
    for (i=0; i<pv; ++i)
      AVN(a,v,n);
    PrintCells(zoom);
    if (XCheckMaskEvent(dpy, ButtonPressMask, &event)) {
      if (event.xbutton.button == Button1)
        Wait4Mouse();
      else if (event.xbutton.button == Button3) {
        SaveCells();
        break;
      }
    }
  }
  Close();
  return 0;
}
```

Bibliographie

[ACC+93] Auger (P.), Cardinal (R.), Chesnais (J.M.), Govaere (M.C.) et Bardou (A.). – Computer Modeling of Arythmias: simulations of ventricular mappings. *Math. Modeling and Sci. Computing*, vol. 2, 1993, pp. 574–779.

[AR84] Allouche (J.P.) et Reder (C.). – Oscillation spatio-temporelles engendrées par un automate cellulaire. *Discr. Appl. Math.*, vol. 8, 1984, pp. 215–254.

[BAC+93] Bardou (A.), Auger (P.), Cardinal (R.), Dumee (P.), Birkvi (P.), Von Euw (D.) et Govaere (M.C.). – Theoretical study by means of computer simulation of conditions in wich extrasystoles can trigger ventricular fibrillation. Validation by epicardial mappings. *Journal of Biological Systems*, vol. 1, n 2, 1993, pp. 147–158.

[Ber95] Bernard (O.). – *Etude expérimentale et théorique de la croissance de Dunaliella tertiolecta (chlorophyceae) soumise à une limitation variable de nitrate : utilisation de la dynamique transitoire pour la conception et la validation des modèles.* – Thèse de PhD, University Pierre et Marie Curie, Paris VI, 1995, 183p.

[Cre93] Cressie (N.). – *Statistics for Spatial Data.* – New York, Wiley, 1993.

[Del96] Delahaye (J.P.). – Le Jeu de la vie, toujours vivant... *Pour la Science*, no221, 1996.

[EL97] Elmoznino (H.) et Lobry (C.). – Automates Cellulaires et Modélisation de la Dynamique Forestière. *Ecologie*, vol. 28, n4, 1997.

[Fav95] Favrichon (V.). – *Modèle Matriciel déterministe en temps discret. Application à l'étude de la dynamique d'un peuplement forestier tropical humide (Guyane Française).* – Thèse, Université Claude Bernard - Lyon 1, 1995.

[Fel50] Feller (W.). – *Probability Theory and its Applications, Volume 1.* – New York, Wiley, 1950.

[GGH80] Greenberg (J.M.), Greene (C.) et Hastings (S.P.). – A combinatorial problem arising in the study of reaction diffusion equations. *SIAM J. Alg. Disc. Math.*, vol. 1, 1980.

[GH78] Greenberg (J.M.) et Hastings (S.P.). – Spatial patterns for discret models of diffusion in excitable media. *SIAM J. Appl. Math.*, vol. 34, 1978, pp. 515–523.

[GHH78] Greenberg (J.M.), Hassard (B.) et Hastings (S.P.). – Pattern formation and periodic structures in systems modeled by reaction diffusion equations. *Bull. Amer. Math. Soc.*, vol. 84, 1978, pp. 1296–1327.

[Gor95] Goreaud (F.). – *Etude et modélisation des peuplements hétérogènes.* – Rapport technique, ENGREF, 1995.

[Hou] Houllier (F.). – Modélisation de la croissance de l'Epicéa dans les futaies irregulières de montagne. *In : réunion de travail 27 Novembre 1992, IFREMER, Paris.*

[KI93] Kazunori (S.) et Iwasa (Y.). – Modelling of wave regeneration in subalpine ABIES forest: Population dynamic with spatial structure. *Ecology*, vol. 74, n5, 1993, pp. 1538–1550.

[Lob93] Lobry (C.). – Automates Cellulaires et Modélisation de la forêt. *In : Actes du Colloque Forêt Tropicale, Kourou (France), Juillet 1993.*

[MS93] Markus (M.) et Schepers (H.). – Turing structures in a semi-random cellular automaton. *In : Mathematics applied to biology and medicine*, éd. par Wuertz. – Demongeot J., Capasso V., 1993.

[Rem91] Remmert (H.) (édité par). – *The Mosaic-Cycle Concept of Ecosystems.* – Springer-Verlag, 1991, *Ecological studies*, volume 85.

[Rip81] Ripley (B. D.). – *Spatial Statistics.* – New york, Wiley, 1981.

[RPH98] Riéra (B.), Pélissier (R.) et Houllier (F.). – Caractérisation d'un Mosaïque Forestière et de sa Dynamique en Forêt Tropicale Humide Sempervirente. *Biotropica*, vol. 30, n2, 1998, pp. 251–260.

[SHJD92] Silvertown (J.), Holtier (S.), Johnson (J.) et Dale (P.). – Cellular automaton model of interspecific competition for space - effect of pattern process. *Journal of Ecology*, vol. 80, 1992, pp. 527–534.

[UF85] Upton (G. J. G.) et Fingleton (B.). – *Spatial Data Analysis by Example, Volume 1.* – Chichester, Wiley, 1985.

[VR46] Viener (N.) et Rosenblueth (A.). – The mathematical formulation of the problem of conduction of impulses in a network of connected excitable elements, specially in the cardiac muscle. *Arch. Inst. Cardiol. Mex.*, vol. 16, 1946, pp. 205–265.

[Win80] Winfree (A. T.). – The Geometry of biological Time. *Biomathematics*, vol. 8, 1980. – Springer-Verlag.

[Wis91] Wissel (Ch.). – A Model for the Mosaic-Cycle Concept. *In : The Mosaic-Cycle Concept of Ecosystems*, éd. par H. (Remmert). – Springer-Verlag, 1991.

[Wol86] Wolfram (S.). – *Theory and Applications of Cellular Automata.* – World Scientific, 1986.