N° d'ordre : 1514

THÈSE

PRÉSENTÉE À

L'UNIVERSITÉ BORDEAUX I

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET D'INFORMATIQUE

Par **Jérôme Olivier DURAND-LOSE**

POUR OBTENIR LE GRADE DE

DOCTEUR

SPÉCIALITÉ : INFORMATIQUE

Automates Cellulaires,
Automates à Partitions et
Tas de Sable.

Soutenue le : lundi 17 juin 1996

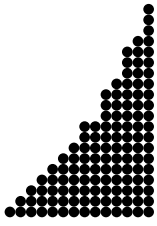
Après avis des rapporteurs :

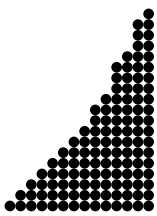
Jean-Paul ALLOUCHE	Directeur de recherche, CNRS
Serge GRIGORIEFF	Professeur

Devant la commission d'examen composée de :

Michel MENDÈS-FRANCE	Professeur	Président
Robert CORI	Professeur	Rapporteur
Eric GOLES	Professeur	Co-directeur de thèse
Serge GRIGORIEFF	Professeur	Examinateur
Jacques MAZOYER	Professeur	Co-directeur de thèse
Yves MÉTIVIER	Professeur	Co-directeur de thèse

– 1996 –





Je tiens à remercier vivement :

Le Professeur Michel MENDÈS-FRANCE, de l'UFR de Mathématiques de l'Université Bordeaux I, qui m'a fait l'honneur de présider ce jury.

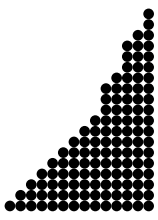
Le Professeur Robert CORI, du Laboratoire Bordelais de Recherche en Informatique de l'Université Bordeaux I, qui a bien voulu rapporter ma soutenance.

MM. les rapporteurs, Jean-Paul ALLOUCHE, Directeur de recherche CNRS, et Serge GRIGORIEFF, Professeur de l'Université Paris VII, dont les critiques constructives me furent fort utiles.

Le Professeur Jacques MAZOYER, du Laboratoire d'Informatique du Parallélisme de l'ENS de Lyon, avec qui j'ai commencé cette thèse, et plus généralement, les membres du LIP.

Le Professeur Eric GOLES, du Departamento de Ingeniería Matemática de la Facultad de Ciencias Físicas y Matemáticas de l'U. de Chile à Santiago, qui m'a accueilli en coopération au Chili, et l'équipe du DIM, ainsi que la Coopération française.

Le Professeur Yves MÉTIVIER, du Laboratoire Bordelais de Recherche en Informatique de l'Université Bordeaux I, qui m'a encadré en fin de thèse, et l'ensemble du LaBRI.



A ma famille,

A mes amis,

et

A tous ceux qui m'ont fait me sentir chez moi,
Alors que je n'étais qu'un intrus chez eux.

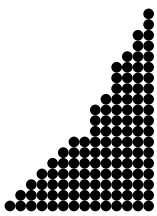
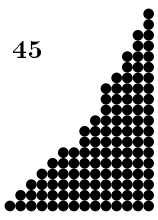
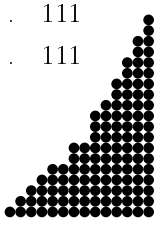
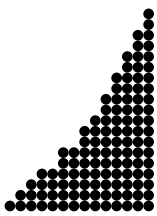
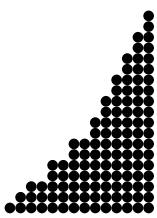


Table des matières

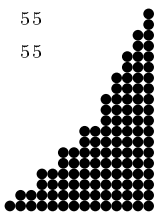


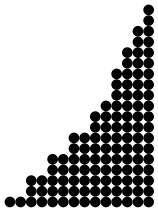






Liste des figures





Introduction

Cette thèse se compose de deux parties distinctes correspondant à mon parcours. J'aurais aimé qu'elles ne forment qu'un tout, mais le chemin que j'ai emprunté ne l'a pas permis.

J'ai commencé ma thèse avec le Professeur Jacques Mazoyer au Laboratoire de l'Informatique du parallélisme de l'ens Lyon. Je suis ensuite parti au Chili pour effectuer mon service militaire comme coopérant à l'Universidad de Chile, encadré par le Professeur Eric Goles, co-directeur de thèse. Depuis septembre 1995, je suis au Laboratoire Bordelais de Recherche en Informatique où j'ai fini cette thèse sous la houlette du Professeur Yves Métivier, également co-directeur.

La première partie, commencée à Lyon, a été terminée au Chili. Nous y démontrons des résultats théoriques sur les automates cellulaires.

La seconde partie entreprise au Chili s'est achevée à Bordeaux. Elle est consacrée à l'étude d'un automate cellulaire particulier : le tas de sable linéaire.

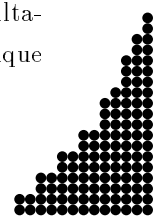
Depuis plusieurs décennies, le parallélisme intéresse les chercheurs. A cela, il y a deux raisons principales : d'une part mieux faire ce que l'on sait déjà faire en séquentiel et exploiter au maximum le parallélisme, et d'autre part mieux comprendre et modéliser les phénomènes physiques parallèles. De nos jours, il est le sujet de beaucoup d'études, tant théoriques que pratiques car les progrès en termes de puissance et de rapidité des microprocesseurs sont de plus en plus difficiles et coûteux, alors que des ordinateurs parallèles et massivement parallèles sont disponibles sur le marché.

Cette thèse se situe dans ce vaste domaine. Les automates cellulaires sont les modèles canoniques du calcul à fine granularité, et leurs capacités en termes de calcul et de simulation intéressent un grand nombre de personnes.

Les automates cellulaires (ac) sont connus et utilisés depuis les années 50 et les travaux d'Ulam [139] et de Von Neumann [110].

L'ac le plus connu du grand public est le « Jeu de la vie » de J. Conway [22]. Entre deux instants t et $t+1$, une case du plan est allumée, ou éteinte, en fonction de son état (allumée/éteinte) et de ceux de ses huit plus proches voisines. On le retrouve dans des revues de vulgarisation scientifique ([79], La Recherche), et dans le livre Winning-Ways de Berlekamp *et al.* [14, chap. 25]. Il sert d'économiseur d'écran de stations de travail et est utilisé, à titre d'exemple, dans la sous-section 3.2.

Les ac travaillent sur des grilles régulières où chaque point, appelé cellule, possède une valeur dans un ensemble fini d'états. A chaque itération, toutes les cellules sont remises à jour simultanément grâce à une unique fonction locale dépendant des mêmes cellules voisines. La dynamique



est uniforme : elle est la même partout. Elle est purement locale : il n'y a ni interaction à longue distance, ni référence à une donnée globale. Elle est parallèle et synchrone : toutes les cellules sont mises à jour en même temps. L'uniformité, la localité, le parallélisme et le synchronisme sont les aspects fondamentaux des AC. Ils se rencontrent aussi sous le nom de « Tessellation automata » ou structures dans la littérature. Il existe des versions asynchrones des AC, mais la finalité est tout à fait différente [24].

Les AC peuvent aussi être définis comme les fonctions globales commutant avec toutes les translations et continues pour la topologie produit. Dans ce cadre, on les appelle « Shift dynamical systems » [87, 118]. Ce point de vue concerne plus les mathématiques discrètes, alors que le précédent se rattache à l'informatique. Hedlund [60] a démontré l'équivalence de ces deux définitions de manière implicite en 1969. Richardson [117] l'a redémontrée de manière explicite en 1972.

En biologie, les AC servent à étudier l'organisation d'éléments biologiques entre eux [46, 62, 83].

En physique, ils sont utilisés pour modéliser des phénomènes non-linéaires à interaction locale (écoulements, avalanches, propagations ...). Dans l'univers réel, les interactions sont locales (limitées par la célérité de la lumière) et uniformes dans l'espace. Les AC réversibles sont les pendants de la réversibilité macroscopique. Différentes notions ont été ainsi appliquées aux AC : thermodynamique, isentropisme, énergie potentielle, mouvement perpétuel ... [25, 98, 141].

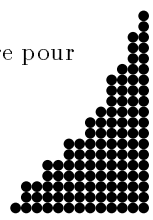
En informatique, les AC sont un outil de modélisation très important pour les architectures massivement parallèles ainsi que pour beaucoup de recherches théoriques [47, 55, 69, 92, 130], notamment la synchronisation [91, 96, 97]. Ils sont utilisés pour le traitement d'images (lissage) ou en traitement du signal (suppression du bruit). Et plus généralement dans tout problème où l'on doit faire subir le même traitement partout dans un immense tableau.

Les différentes classes de langages et de fonctions calculables définies par des AC ont été étudiées. Il existe encore beaucoup de problèmes ouverts concernant les égalités de classes de langages reconnus par des AC sur une ligne finie, en temps réel ou linéaire en fonction de la taille de l'entrée. On parle dans ce cas-là d'« Iterative array » et parfois d'architectures systoliques. Les questions que l'on se pose sont les fermetures par concaténation, par réflexion, si les inclusions sont strictes ... [61, 67, 131, 132] aussi bien que la puissance de reconnaissance de différents voisinages pour des AC de dimension 1 [61, 114, 140].

En cryptographie, la réversibilité est primordiale pour récupérer les données originelles [145]. Pour assurer la confidentialité, l'inversion du codage doit être la plus difficile possible. On sait que la complexité de l'automate inverse ne peut être bornée en fonction de l'AC (sinon la réversibilité serait décidable, ce qui n'est pas le cas en dimension supérieure à 2, *cf.* section 2.5). Mais malheureusement, on ne sait obtenir systématiquement que des AC réversibles dont les inverses sont de la même taille. De plus, ne travaillant que localement, les AC ne mélangent pas assez rapidement les données.

En mathématiques et en informatique théorique, ils sont étudiés en rapport avec les structures fractales [2, 3, 28, 111, 127], l'algèbre [18, 112], la complexité [124–126], les statistiques [19] ... Leurs propriétés fonctionnelles, notamment la réversibilité, ont aussi beaucoup été étudiées.

La réversibilité est une notion importante, elle couvre à la fois le désir de revenir en arrière pour



remonter aux origines des phénomènes, et l'espoir de ne plus perdre d'énergie avec des ordinateurs réversibles (isentropiques). Par exemple, la recherche d'un inverse peut correspondre à la solution d'un problème codé par une configuration [123]. Toffoli a étudié la réversibilité des AC [133–135] et rédigé un large tour d'horizon sur ce sujet avec Margolus [138].

La recherche sur la réversibilité des AC commença en 1962 quand Moore [100] demanda s'il existait des AC pour lesquels certaines configurations n'avaient pas d'antécédents. Il donna comme condition nécessaire à cela : la non-injectivité. L'année suivante, Myhill [107] démontra que cette condition était suffisante. Ce résultat est connu sous le nom de « Garden-of-eden theorem ». La réversibilité ne fut nommée comme telle qu'à partir du début des années 70 et des articles de Richardson [117] et d'Amoroso [4–7].

A partir de là, la recherche sur les AC réversibles (AC-R) et leurs propriétés fonctionnelles [94, 95, 108, 150, 151] commença véritablement. Les AC-R sont relativement rares [118]. Burks [20] émit l'hypothèse qu'il n'en existait aucun d'universel pour le calcul, mais Toffoli [133] prouva le contraire en construisant un tel AC-R (de dimension 2) à partir des travaux de Bennett [12, 13] sur les machines de Turing réversibles.

La réversibilité est aussi reliée aux problèmes d'ensembles limites : ensembles des configurations qui ont des antécédents de tout ordre [29, 53, 74, 86]. Toutes les configurations possibles sont dans l'ensemble limite si, et seulement si, l'automate est surjectif, ce qui est impliqué par l'injectivité global et est équivalent à l'injectivité sur les configurations finies [100, 107].

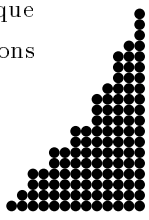
Des considérations physiques ont abouti à de nouveaux AC-R : la grille de gaz de Hardy, Pazzis et Pomeau [59] issue de modélisations hydrodynamiques, ou celle de Toffoli et Margolus [23, 45, 90, 137], qui donna le modèle de la boule de billard BBM étudié dans le chapitre 4 de la première partie.

La dimension d'un AC est celle de la grille sur laquelle il opère. Il est à signaler que les modèles des tas de sable (« sand pile model » ou SPM) et du « chip firing game » (CFG) linéaires, étudiés dans la seconde partie, sont des AC de dimension 1.

Les AC de dimension 1 (AC-1) sont à part. Amoroso et Patt [7] ont démontré qu'il est possible de déterminer algorithmiquement si un AC-1 est réversible ou non, c'est un problème décidable. Kari [72] a démontré que cela est impossible en dimension supérieure ainsi que l'indécidabilité de l'appartenance à toute classe non triviale [76]. La question de savoir si oui ou non il existait un AC-1 réversible universel pour le calcul fut résolue affirmativement par Morita et Harao [105].

La recherche sur la puissance de calcul des AC-R et leurs possibilités de simulation a généré de nouvelles techniques, et a indiqué les limites des machines en termes de calcul. Toffoli [133] a montré qu'un AC pouvait toujours être simulé par un AC-R, mais ce dernier a une dimension de plus. De même, Morita [102, 104] a démontré que cela était possible avec un AC-R de même dimension, mais cette fois, la simulation ne marche que sur des configurations finies (*i.e.*, constantes en dehors d'une partie finie de la grille).

La recherche sur les AC prit un nouvel élan au début des années 80 avec, entre autres, les conférences de l'île Mosquito et les travaux de Wolfram [142–144, 146, 147] proches de la physique statistique. C'est l'époque où apparurent le BBM de Margolus [89, 90] ainsi que des modélisations



du spin Ising [25, 98, 141]. Čulik publia différents articles sur les AC [27–29] dont [26] sur les réversibles, Morita s'intéressa à la simulation [101–106] et Kari s'intéressa à la décidabilité de différentes propriétés des AC [71–73,75,76]. Récemment, des sélections d'articles sur les AC ont été publiées, par Wolfram en 1986 [148] et en 1994 [149], et par Gutowitz en 1991 [58].

Il est aussi à signaler qu'il existe des réalisations d'AC génériques. Toffoli et Margolus ont fait une carte pour PC permettant d'effectuer et d'observer les itérations d'AC relativement simples [137]. Actuellement, il se fait en Allemagne [63, 64] de la recherche sur les langages et les architectures pour les AC. Parallèlement, une équipe italienne a fait un langage et une implantation sur un réseau de transputers avec récupération des données sous Maple [21, 57]. Ces deux dernières réalisations sont utilisées pour des applications pratiques comme l'évolution d'un volcan, des couches de terrain en géologie, de la circulation sur un réseau d'autoroutes ou de la contamination des sols.

La première partie de la thèse s'intéresse aux AC et aux capacités qu'ils ont de se simuler entre eux. L'étude porte plus particulièrement sur les AC réversibles et sur la simulation.

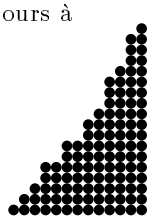
Nous utilisons les automates (cellulaires) à partitions de Toffoli et Margolus dont la réversibilité est décidable. Ils sont aussi connus sous le mon de « voisinage de Margolus », et bien que pratiquement homonymes, ce ne sont pas les automates cellulaires « partitionnés » de Morita. Nous construisons diverses simulations les reliant aux AC. Leurs sous-classes réversibles sont aussi reliées.

D'autres constructions sont ensuite faites sur un automate à partitions particulier, le « Billiard ball model » de Toffoli et Margolus. Nous montrons son universalité au sens du calcul, ainsi que sa capacité à simuler tous les automates à partitions réversibles grâce à des constructions de plus en plus abstraites.

Nous prouvons ainsi l'existence d'AC capables de simuler tous les AC de même dimension. Il en est de même si l'on se restreint aux AC-R. Il s'agit de notre principal résultat, c'est un pas de plus pour savoir s'il existe ou non un AC-R capable de simuler tous les AC de même dimension, réversibles ou non.

Le tas de sable, ou « Sand pile model » (SPM), est un modèle simple d'écoulement de grains dans un espace modélisé par un graphe. SPM sert en informatique à modéliser la migration des tâches dans un réseau de processeurs et à concevoir des algorithmes d'équilibrage de charges [85, 121, 122, 128] et dans ce cas il rejoint les réseaux de Pétri [116].

En physique granulaire [68], SPM est utilisé pour modéliser les phénomènes du type avalanche, tremblement de terre, plasma ... qui sont appelés « $1/f$ phenomena », car ils vérifient statistiquement un rapport inverse entre les intensités des perturbations et leurs probabilités. On parle aussi de « Self-organized criticality », car après un bouleversement, le système est toujours dans un état pseudo-stable, et peut tout de suite se re-bouleverser [9–11,32,33,70,129,152]. La cuillerée de sucre en est un exemple : on a beau la cogner contre les bords du bocal, on arrive toujours à faire tomber quelques grains. Le fait de tasser n'a pas vraiment amélioré la stabilité.



En ligne, SPM est un automate cellulaire équivalent au « Chip firing game » (CFG) où un nœud donne un grain à chacun de ses voisins dès qu'il a au moins autant de grains que de voisins. La dynamique du CFG repose sur la valeur alors que celle du SPM se base sur le gradient.

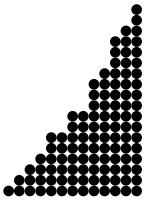
Ce modèle qui lui aussi conserve le nombre de grains, est une autre forme de réseau de Pétri. Il a été étudié par : Anderson *et al.* [8], Bitar et Goles [15], Björner [16, 17] et Prisner sur des graphes orientés [113]. Les conditions d'arrêt et de non-arrêt ont été étudiées sur différents types de graphes.

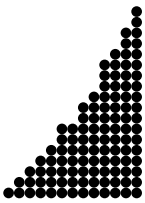
La seconde partie de cette thèse est consacrée à l'écroulement d'une pile de sable dans un espace linéaire. La dynamique du modèle est très simple : si une pile de sable a au moins deux grains de plus qu'une de ses voisines, elle lui en donne un. Le nombre de grains reste constant.

Le phénomène a déjà été étudié dans le cas séquentiel et mis en bijection avec CFG par Goles et Kiwi [48–52]. Nous nous intéressons au cas parallèle.

Dans un premier temps, nous étudions l'évolution d'un système, initialement vide, dont la première pile reçoit un grain à chaque itération. Des signaux apparaissent sur les configurations de CFG correspondantes. L'étude de la dynamique ainsi que la position finale des grains, en fonction de leurs parités, permet de comprendre l'évolution du système.

A partir de là, le cas initial s'étudie simplement. Nous démontrons le principal résultat de la seconde partie : le temps de stabilisation d'une pile seule est linéaire en fonction de son nombre de grains. Le facteur d'accélération par rapport au cas séquentiel est de l'ordre du nombre de piles utilisées. Nous appliquons ces résultats et techniques à divers cas bornés ainsi qu'au contrefort.





...
donnez-moi de nouvelles données
donnez-moi de nouvelles données

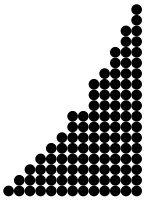
à perte de vue des lacs gelés
qu'un jour j'ai juré d'enjamber

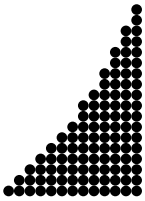
à perte de vue des défilés
des filles à lever
des défis à relever
...

A. Bashung - J. Fauque,
perte de vue.

Partie A

Automates à Partitions





Chapitre 1

Introduction aux Automates à Partitions

Dans cette partie, nous étudions les automates (cellulaires) à partitions (AP) et les relions par différentes simulations aux automates cellulaires (AC). Par une suite de simulations en cascade, nous démontrons un résultat de simulation concernant la sous-classe des AC réversibles (AC-R).

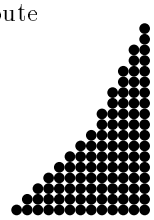
Toute cette partie se base sur les travaux de tous ceux qui se sont intéressés à la réversibilité des AC depuis les années 60, en particulier dans les deux dernières décennies : Kari, Margolus, Morita, Toffoli, Wolfram ...

La première partie de la thèse s'articule comme suit :

Le chapitre 2 est consacré aux définitions des automates cellulaires (AC) et des automates à partitions (AP), de leurs espaces de travail ainsi que des notions de réversibilité, de simulation et d'universalité. Ces trois dernières notions sont les clefs de cette partie. Des lemmes sur la réversibilité et l'universalité des AP sont démontrés. Pour chaque machine de Turing, nous construisons un AP de dimension 1 la simulant.

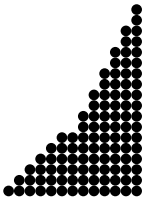
Dans le chapitre 3, nous construisons des simulations entre AC et AP, dans les deux sens. Nous faisons de même avec leurs sous-classes réversibles. Cela nous permet d'affirmer que les puissances de calcul et de simulation sont identiques dans chacune des classes. Il en est de même dans les deux sous-classes réversibles. Nous démontrons ainsi la conjecture 8.1 de l'article [138] de Toffoli et Margolus sur la simulation des AC-R par les AP-R, partiellement démontrée par Kari [77]. Nous donnons des exemples de constructions avec le jeu de la vie et une règle majoritaire. Nous commentons la complexité des automates obtenus par ces simulations.

Le chapitre 4 est centré sur l'étude d'un AP-R particulier : le modèle de la boule de billard (BBM) de Toffoli et Margolus [137, 138]. Nous rappelons les résultats et méthodes de Fredkin, Margolus et Toffoli sur cet automate, sur la logique conservatrice et sur la capacité de BBM à simuler toute fonction de cette logique.



Nous utilisons leur codage pour construire un nouveau codage que nous nommons dual. Celui-ci permet de simuler simplement toute fonction de la logique réversible. A partir de là, grâce à plusieurs constructions de plus en plus abstraites, nous démontrons que BBM est capable de simuler toutes les machines de Turing, puis tous les AP-R, sur n'importe quelle configuration. Nous ajoutons un état, '$\star$' à BBM pour obtenir BBM*, un AP capable de simuler tous les AP, réversibles ou non . L'automate BBM* n'est malheureusement pas réversible.

Le chapitre 5 termine cette partie en combinant les résultats des chapitres 3 et 4. Nous donnons un AC-R capable de simuler tout AC-R et un non-réversible capable de simuler tous les AC.



Chapitre 2

Définitions et Propriétés

Dans ce chapitre nous rappelons les définitions des automates cellulaires, des automates à partitions et des espaces où ils opèrent. Nous donnons également la définition et des résultats sur la réversibilité, la simulation et l'universalité. Ce sont les trois notions clefs de cette partie.

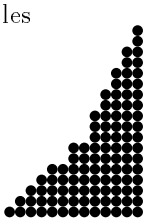
2.1 Espace de Travail

Les définitions et vocabulaires suivants correspondent aux automates à partitions. Pour les automates cellulaires, les objets manipulés restent les mêmes, mais avec un vocabulaire légèrement différent, pour respecter l'usage et pour bien les différencier. Tant que faire se peut nous essayons d'utiliser des lettres majuscules droites pour les automates cellulaires et cursives pour ceux à partitions.

L'espace de travail est une *grille* $\mathcal{L}$ de type $\mathbb{Z}^d$. La *dimension* est d. Par convention, quand rien n'est précisé, $d = 2$: la grille est un plan discret. C'est le cadre général de la première partie de la thèse. Pour ne pas alourdir inutilement ; les définitions, les démonstrations et les figures sont toutes en dimension 2. La plupart des résultats et démonstrations s'étendent sans problèmes aux autres dimensions. Les généralisations sont faites dans le chapitre 5.

Définition 1 Les points de la grille sont appelés *nœuds*. Chaque nœud possède un *état* appartenant à un ensemble fini $\mathcal{Q}$. Une *configuration* est une valuation de la grille, c'est-à-dire une grille où l'état de chaque nœud est défini.

Chaque automate utilise son propre ensemble d'états. L'ensemble de toutes les configurations est noté : $\mathcal{C}$ pour les AP ($\mathcal{C} = \mathcal{Q}^{\mathcal{L}} = \{\mathcal{L} \to \mathcal{Q}\}$). Tous les automates décrits opèrent sur les configurations. Ils définissent des règles de transition ou d'itération.



2.2 Automates Cellulaires

Les automates cellulaires étant beaucoup plus connus et utilisés que ceux à partitions, nous rappelons leur définition en premier.

Les automates cellulaires (AC) opèrent sur le même espace que les automates à partitions. Leurs grilles sont notées L ($= \mathbb{Z}^2$). Ce sont les mêmes, mais pour les constructions des chapitres suivants, il est nécessaire de bien différencier les deux types. Les points de cette grille sont appelés *cellules* et prennent une valeur dans un ensemble fini d'*états* Q. L'ensemble de toutes les configurations est noté : C ($\mathsf{C} = \mathsf{Q}^{\mathsf{L}} = \{\mathsf{L} \to \mathsf{Q}\}$).

Définition 2 Un *Automate Cellulaire* (AC), A, est défini par le triplet suivant :

$$\mathsf{A} \;=\; (\,\mathsf{Q},\,\mathsf{r},\,\mathsf{f}\,)\;,$$

où Q est un ensemble fini d'*états*, r un entier strictement positif appelé *rayon* et f une fonction de $\mathsf{Q}^{(2\mathsf{r}+1)\times(2\mathsf{r}+1)}$ dans Q appelée *fonction locale*.

Les configurations sont itérées de la manière suivante. L'image d'une cellule est déterminée par les valeurs des cellules de son *voisinage*, c'est à dire, distantes d'au plus r selon chaque coordonnée.

Définition 3 La *fonction globale* $\mathsf{F} : \mathsf{C} \to \mathsf{C}$ d'un automate cellulaire, $(\mathsf{Q}, \mathsf{r}, \mathsf{f})$, opère sur l'ensemble des configurations comme suit : $\forall \mathsf{c} \in \mathsf{C}$, $\forall (i,j) \in \mathsf{L}$,

$$\mathsf{F}(\mathsf{c})_{i,j} \;=\; \mathsf{f}\left(\begin{array}{ccccccc}
\mathsf{c}_{i-\mathsf{r},j+\mathsf{r}} & \mathsf{c}_{i-\mathsf{r}+1,j+\mathsf{r}} & \cdots & \mathsf{c}_{i,j+\mathsf{r}} & \cdots & \mathsf{c}_{i+\mathsf{r}-1,j+\mathsf{r}} & \mathsf{c}_{i+\mathsf{r},j+\mathsf{r}} \\
\mathsf{c}_{i-\mathsf{r},j+\mathsf{r}-1} & \mathsf{c}_{i-\mathsf{r}+1,j+\mathsf{r}-1} & \cdots & \mathsf{c}_{i,j+\mathsf{r}-1} & \cdots & \mathsf{c}_{i+\mathsf{r}-1,j+\mathsf{r}-1} & \mathsf{c}_{i+\mathsf{r},j+\mathsf{r}-1} \\
\vdots & \vdots & & \vdots & & \vdots & \vdots \\
\mathsf{c}_{i-\mathsf{r},j} & \mathsf{c}_{i-\mathsf{r}+1,j} & \cdots & \mathsf{c}_{i,j} & \cdots & \mathsf{c}_{i+\mathsf{r}-1,j} & \mathsf{c}_{i+\mathsf{r},j} \\
\vdots & \vdots & & \vdots & & \vdots & \vdots \\
\mathsf{c}_{i-\mathsf{r},j-\mathsf{r}+1} & \mathsf{c}_{i-\mathsf{r}+1,j-\mathsf{r}+1} & \cdots & \mathsf{c}_{i,j-\mathsf{r}+1} & \cdots & \mathsf{c}_{i+\mathsf{r}-1,j-\mathsf{r}+1} & \mathsf{c}_{i+\mathsf{r},j-\mathsf{r}+1} \\
\mathsf{c}_{i-\mathsf{r},j-\mathsf{r}} & \mathsf{c}_{i-\mathsf{r}+1,j-\mathsf{r}} & \cdots & \mathsf{c}_{i,j-\mathsf{r}} & \cdots & \mathsf{c}_{i+\mathsf{r}-1,j-\mathsf{r}} & \mathsf{c}_{i+\mathsf{r},j-\mathsf{r}}
\end{array}\right)\;.$$

L'image d'une configuration est une configuration, ce qui permet l'utilisation itérative des AC. Nous nous référons de temps en temps à une configuration et à son image comme respectivement ancienne (ou actuelle) et nouvelle.

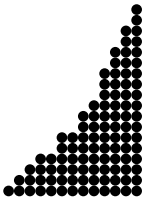
2.3 Automates à Partitions

Maintenant un voisinage ne sert plus à calculer la valeur nouvelle d'une cellule, mais celle de tout le voisinage.

Soient h et v deux entiers strictement positifs, et α, β, x et y quatre entiers tels que $0 \leq x < h$ et $0 \leq y < v$.

Définition 4 La *tuile* de coordonnée $(\,\alpha,\,\beta\,)$ dans la *partition* de *taille* $(\,h,\,v\,)$ et d'*origine* $(\,x,\,y\,)$ de la configuration c est la partie rectangulaire de la grille suivante :

$$b_{\alpha,\beta} \;=\; {}^{c}|_{(\,x+\alpha.h,\;y+\beta.v\,)} \;+\; [0,h-1]\times[0,v-1]\;.$$



Les tuiles sont des éléments de $\mathcal{Q}^{h\times v}$. Nous les notons b pour « block » qui est le vocabulaire de Kari dans [77].

Définition 5 La *partition* de *taille* (h, v) et d'*origine* (x, y) est le découpage régulier de toute la grille en tuiles, comme illustré par la figure 1.

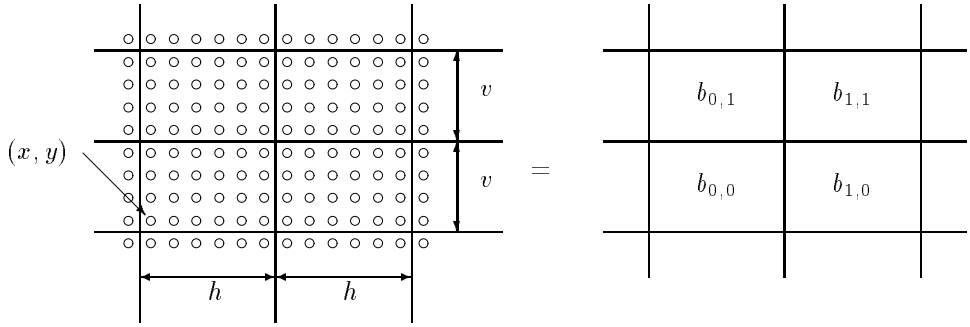


Figure 1 : Partition d'origine (x, y) et de taille (h, v) .

Le couple (h, v) est une constante pour un automate à partitions donné. Pour cela, on l'appelle aussi la taille de l'automate.

Définition 6 La *transition locale*, t, est une fonction opérant sur l'ensemble des tuiles ($t : \mathcal{Q}^{h\times v} \longrightarrow \mathcal{Q}^{h\times v}$).

Cette fonction permet d'agir sur les partitions de la façon suivante :

Définition 7 Une *transition (élémentaire)* T, est le remplacement simultané de toutes les tuiles d'une partition par leurs images par la transition locale, t, comme illustré par la figure 2.

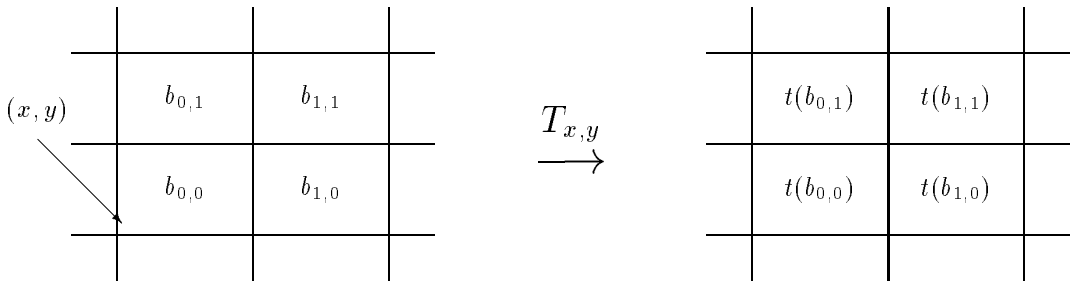


Figure 2 : $T_{x,y}$: transition d'origine (x, y) .

Définition 8 Un *Automate à Partitions* (AP), est défini par le quintuplet suivant :

$$(\ \mathcal{Q},\ (h, v),\ n,\ \mathcal{O},\ t\ )\ ,$$

où $\mathcal{O} = (x_i, y_i)_{1\leq i\leq n}$ est le n-uplet des origines des n partitions.

Les (x_i, y_i) sont les origines des partitions utilisées pour les n transitions de la fonction de mise à jour. Les origines sont toutes prises modulo la taille des tuiles, *i.e.*, $\forall i$, $0 \leq x_i < h$ et $0 \leq y_i < v$.

Définition 9 La *fonction globale* ou *transition globale* d'un automate à partitions, $\mathcal{T}$, est la composition séquentielle des transitions d'origines $(x_i, y_i)_{1 \leq i \leq n}$:

$$\mathcal{T} : \mathcal{C} \to \mathcal{C} = T_{x_n,y_n} \circ T_{x_{n-1},y_{n-1}} \circ \cdots \circ T_{x_1,y_1} \ .$$

Plusieurs partitions doivent être utilisées pour permettre à l'information de se promener de tuile en tuile, comme illustré par la figure 3. Le nombre de partitions et de transitions utilisées par l'AP est noté n. Il ne doit pas être déraisonnablement grand pour des raisons de cohérences. S'il n'y avait pas de limite, il suffirait de prendre à chaque fois k fois toutes les partitions pour avoir un facteur d'accélération de k. Cette possibilité ne nous plaît guère car elle permet de mettre en fait beaucoup d'itérations en une seule. Nous pensons qu'il devrait y avoir un rapport du type $n \leq h.v$ ou $n \leq \ln(h.v)$. Derrière ceci se profilent des questions de définitions de complexité en temps d'une résolution d'un problème par un AP.

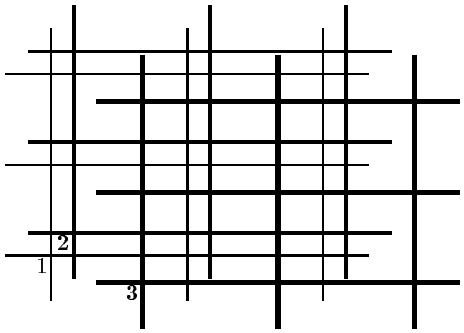


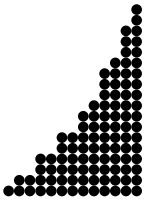
Figure 3 : Trois partitions.

Il y a un exemple d'AP de dimension 1 à la fin de ce chapitre (proposition 21), et de dimension 2 dans le chapitre 3. Le Billiard ball model décrit au chapitre 4 est un exemple en dimension 2.

Les AP sont aussi connus sous le nom d'AC utilisant le « voisinage de Margolus ». Il existe aussi des automates utilisant des partitions au sens de Morita (partitioned cellular automata), mais c'est autre chose : au lieu de faire une partition de l'espace, ont le fait des cellules (leurs états sont des produits d'états). Sans doute, pour ces derniers, le terme « AC partitionné » devrait-il être utilisé en Français, mais la confusion reste facile. Nous avons choisi le terme automate à partitions d'abord parce qu'il utilise des partitions, et ensuite parce que Toffoli et Margolus les nomment « Partitioning cellular automata » dans [138]. Peut-être aurait-il mieux fallu choisir « AC par bloc » comme le semble sous-entendre Kari [77] quand il parle de « Block Permutation ».

2.4 Simulation

Soient f et g deux fonctions opérant sur deux ensembles F et G ($f : F \to F$ et $g : G \to G$).



Définition 10 La fonction f *simule (itérativement)* g si et seulement s'il existe deux fonctions $\alpha : G \to F$ et $\beta : F \to G$ « négligeables en temps et en espace » devant f et g, et une fonction $\varphi : G \times \mathbb{N} \to \mathbb{N}$, telles que :

$$\forall x \in G,\ \forall n \in \mathbb{N},\ \ g^n(x) \ = \ \beta \circ f^{\varphi(x,n)} \circ \alpha(x) \ . \tag{1}$$

Ceci correspond au schéma transitif de la figure 4. Cette définition se retrouve dans l'article de Toffoli [133]. Il impose à α d'être injective, mais le fait que l'égalité soit vraie pour $n = 0$ l'implique : $\beta \circ \alpha$ est l'identité. La fonction α doit être injective, et β surjective.

$$\forall n \in \mathbb{N}, \qquad \begin{array}{ccc} G & \xrightarrow{\ g^n\ } & G \\ {\scriptstyle \alpha}\downarrow & & \uparrow{\scriptstyle \beta} \\ F & \xrightarrow{\ f^{\varphi(.,n)}\ } & F \end{array}$$

Figure 4 : f simule g.

« Négligeables en temps et en espace » signifie, en deux mots, que la complexité en temps, comme celle en espace, de α et β sont faibles comparées à celles de g et f. Le calcul est fait par f et non par α ou β. Le rôle de α est uniquement de mettre en forme les données pour pouvoir appliquer f, et celui de β uniquement de récupérer le résultat. Dans la plupart des cas, ce sont des projections, des injections canoniques ou des remplacements terme à terme. Elles peuvent être plus complexes que cela, mais quand la taille du problème croît, leurs complexités deviennent négligeables par rapport à celles de f et g.

Soit τ un entier. La simulation est en *temps constant* τ quand φ ne dépend pas de x et est linéaire en n : $\varphi(x, n) = \tau.n$. Elle est en *temps réel* quand $\tau = 1$, *i.e.*, $\varphi(x, n) = n$. Toutes les simulations de cette thèse sont en temps constant. Certaines sont même en temps réel.

Si les fonctions f et g sont bijectives, par unicité des antécédents, il découle que l'égalité (1) est aussi vraie pour n négatif.

La simulation est *uniforme*, ou *réciproque*, si et seulement si α et β sont bijectives et $\beta = \alpha^{-1}$. Cela signifie que réciproquement g simule f.

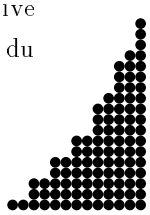
La définition s'étend logiquement aux automates comme suit :

Définition 11 Un automate en simule un autre si et seulement si sa fonction globale simule celle de l'autre.

Les automates étant généralement utilisés de manière itérative, il est naturel de vouloir que la simulation le soit également. De là vient la notion de simulation itérative énoncée.

2.5 Réversibilité

Définition 12 Un automate, A, est *réversible* si et seulement si sa fonction globale est bijective et son inverse est lui-même la fonction globale d'un automate du même type, appelé *inverse* du



premier et noté , A^{-1}.

Les automates cellulaires réversibles sont notés AC-R, ceux à partitions AP-R. Il y a un résultat très simple concernant les AP :

Lemme 13 *Un automate à partitions $\mathcal{P}$ est réversible si et seulement si sa transition locale t est réversible et son inverse est alors :*

$$\mathcal{P}^{-1} \;=\; \left(\; \mathcal{Q},\; (h,\, v),\; n,\; (x_{n+1-i},\, y_{n+1-i})_{1\leq i\leq n},\; t^{-1} \;\right) \quad .$$

Preuve. Si $\mathcal{P}$ est réversible, alors sa fonction globale $\mathcal{T}$ est bijective. Par construction (définition 9), la transition T_{x_1,y_1} doit être injective. Par construction (définition 7), la transition locale t doit être elle aussi injective. Opérant sur un ensemble fini, elle est bijective.

Réciproquement, si la transition locale t est bijective, par construction, la transition T_{x_1,y_1} l'est aussi. Son inverse est $T^{-1}_{x_1,y_1}$, la transition d'origine (x_1, y_1) et de transition locale t^{-1}. Il en est de même pour les autres transitions. Par composition, la fonction globale $\mathcal{T}$ est réversible et son inverse est :

$$\mathcal{T}^{-1} \;=\; T^{-1}_{x_1,y_1} \circ T^{-1}_{x_2,y_2} \circ \cdots \circ T^{-1}_{x_n,y_n} \quad .$$

L'AP inverse s'en déduit directement.

Q. E. D.

On peut programmer un ordinateur pour savoir si une fonction opérant sur un ensemble fini est bijective ou non. En calculabilité, on dit que le problème est *décidable*. Grâce au lemme précédent, la réversibilité des AP est décidable : il existe une méthode effective permettant de savoir si un AP quelconque est réversible ou non.

Pour les automates cellulaires, il n'en est pas de même. Travaillant sur des ensembles de cardinalités différentes, la fonction locale f ne peut être bijective. Néanmoins, il existe les résultats suivants :

Théorème 14 (Moore 1962 [100] et Myhill 1963 [107]) *Un* AC *est bijectif si et seulement si sa fonction globale* F *est injective.*

Théorème 15 (Hedlund 1969 [60] et Richardson 1972 [117]) *Un* AC *est réversible si et seulement si il est bijectif.*

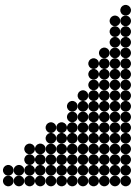
En fait, le résultat principal de chacun des articles [60] et [117] est que toute fonction opérant sur $\mathcal{C}$, continue pour la topologie produit est la fonction globale d'un AC.

Et pour ce qui est de la décidabilité :

Théorème 16 (Amoroso et Patt 1972 [7]) *La réversibilité d'un* AC *de dimension*1 *est décidable.*

Les dimensions supérieures sont le domaine de l'indécidable [66, 71, 76].

Théorème 17 (Kari 1990 [72]) *La réversibilité d'un* AC *de dimension supérieure ou égale à* 2 *n'est pas décidable.*



Il apparaît là une différence fondamentale entre ces deux modèles. Et pourtant, nous montrons dans le chapitre suivant qu'ils se simulent l'un l'autre, tout comme leurs sous-classes réversibles. L'indécidabilité se cache dans la méthode de simulation.

Il est aisé de savoir si un AC est ou non la fonction identité de $\mathcal{C}$ et composer deux AC se fait simplement. On peut donc tester si deux automates cellulaires sont l'inverse l'un de l'autre. Les AC étant dénombrables, il en découle que l'ensemble des AC-R est récursivement énumérable.

2.6 Universalité

Machine de Turing

Une machine de Turing (MT) opère sur une *bande* infinie où sont inscrits des symboles d'un alphabet fini Σ. Elle modifie cette bande grâce à une tête qui peut lire et écrire et se déplacer d'une case à droite ou à gauche. La tête est commandée par un automate fini qui change d'état en fonction de son état et de ce qui est lu par la tête. Cela est schématisé par la figure 5.

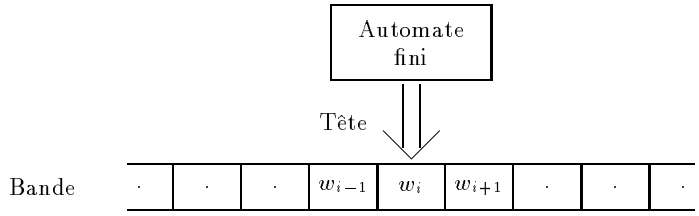


Figure 5 : Machine de Turing.

Une itération est la lecture d'un symbole sur la bande, le changement d'état de l'automate, l'écriture d'un symbole sur la bande et le déplacement de la tête.

Définition 18 Une *machine de Turing* (MT) est définie par :

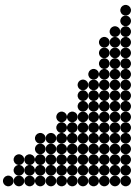
$$(\Sigma,\ Q,\ \delta,\ s_0) \ ,$$

où Σ est l'ensemble fini des *symboles* de la bande, Q l'ensemble fini des *états* de l'automate, δ est la *fonction de transition* et s_0 est l'*état initial* de l'automate. La fonction δ donne le symbole à écrire sur la bande, le nouvel état de l'automate et le sens de déplacement de la tête en fonction de l'état actuel et du symbole lu sur la bande. Elle est de type :

$$\delta \ :\ Q \times \Sigma \ \rightarrow \ Q \times \Sigma \times \{ -1,\, 1 \} \ \cup \ \{ \text{FIN} \} \ .$$

Le pseudo-état FIN signifie la fin du calcul. Il peut ne jamais être atteint. La bande est un mot bi-infini indexé par $\mathbb{Z}$ (un élément de $\Sigma^{\mathbb{Z}}$). Au départ, la tête est en position 0 et l'automate en position s_0. Le calcul se termine quand la fonction de transition δ renvoie FIN. Soit w la bande, q l'état de l'automate et i la position de la tête. La configuration de la machine de Turing est représentée par (q, w, i). La configuration suivante est :

$$(q', w', i+d) \quad \textbf{avec} \quad \delta(q, w_i) = (q', a, d) \quad \textbf{et} \quad \begin{cases} w'_x = w_x & \text{si } x \neq i \\ w'_i = a & \text{sinon} \end{cases} \ .$$



Les machines de Turing servent de base à la calculabilité. Une fonction f est *calculable* si et seulement s'il existe un codage et une machine de Turing, telle que démarrant avec le codage d'une valeur x écrite sur le ruban, elle s'arrêtera avec le codage de $f(x)$ écrit sur le ruban. Les machines de Turing permettent de calculer tout ce qui est calculable (thèse de Church). Pour plus de détails, consulter les chapitres 6 et 7 du livre de Minsky « Finite and infinite machines » [99].

Deux Notions d'Universalité

Définition 19 Un automate est *universel (au sens du calcul)* s'il est capable de simuler n'importe quelle machine de Turing.

Les AC et AP travaillent sur des données de taille infinie à chaque itération, ce que ne peuvent faire les machines de Turing en temps fini. La définition suivante permet de tenir compte de cette particularité, et apporte une autre notion d'universalité.

Définition 20 Un automate est *intrinsèquement universel* s'il est capable de simuler n'importe quel automate du même type.

Ces définitions sont très différentes, la première se réfère au fait de calculer tout ce qui est calculable dans le domaine du fini, alors que la seconde dépend du modèle. La seconde correspond au fait de pouvoir faire tout ce que peuvent faire les autres automates du même type, en ce sens cet automate peut le plus parmi ceux de sa classe.

L'universalité intrinsèque n'implique pas l'universalité (au premier sens) à moins qu'il n'existe un automate de la classe qui le soit (par transitivité de la relation de simulation).

Connaître un automate intrinsèquement universel permet de voir les limites du modèle. Ce qu'il ne peut pas faire, nul autre ne le peut.

Tout cela est indépendant de toutes notions d'efficacité, de vitesse, de simplicité, d'encombrement spatial, de complexité ...

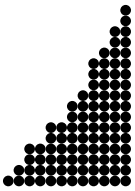
Proposition 21 *Il existe des* AP *universels au sens du calcul.*

Preuve. Pour montrer cela, il suffit de montrer que pour chaque machine de Turing, il existe un AP capable de la mimer. La proposition découle alors du fait qu'il existe des machines de Turing universelles.

Soit $M = (\Sigma, Q, \delta, s_0)$ une machine de Turing à m états et n symboles distincts ($m = |\Sigma|$, $n = |Q|$ et $\Sigma \cap Q = \emptyset$).

La bande étant de dimension 1 nous utilisons un AP de dimension 1. La grille est une droite infinie : $\mathbb{Z}$. Les tuiles sont de taille (2). Il y a deux partitions dont les origines sont (0) et (1). Les états sont $Q \cup \Sigma \cup \{\text{FIN}\}$ et l'AP est :

$$(Q \cup \Sigma \cup \{ \text{FIN} \}, (2), 2, ((0), (1)), t_M) \quad .$$



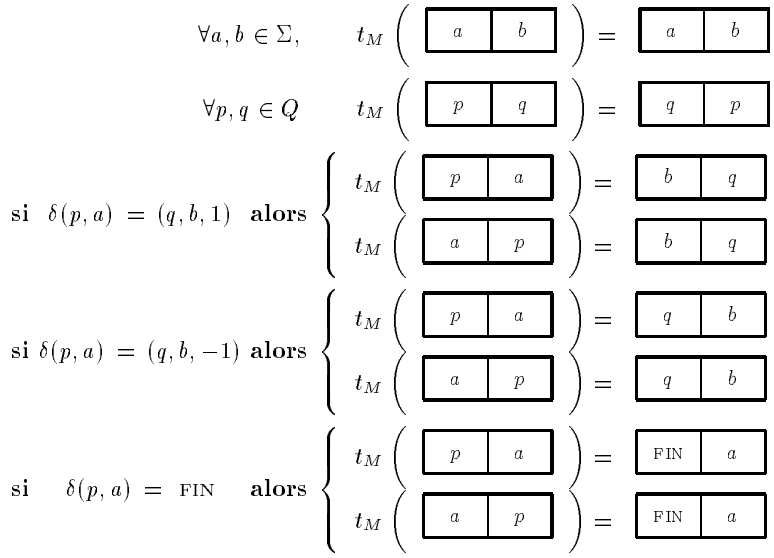
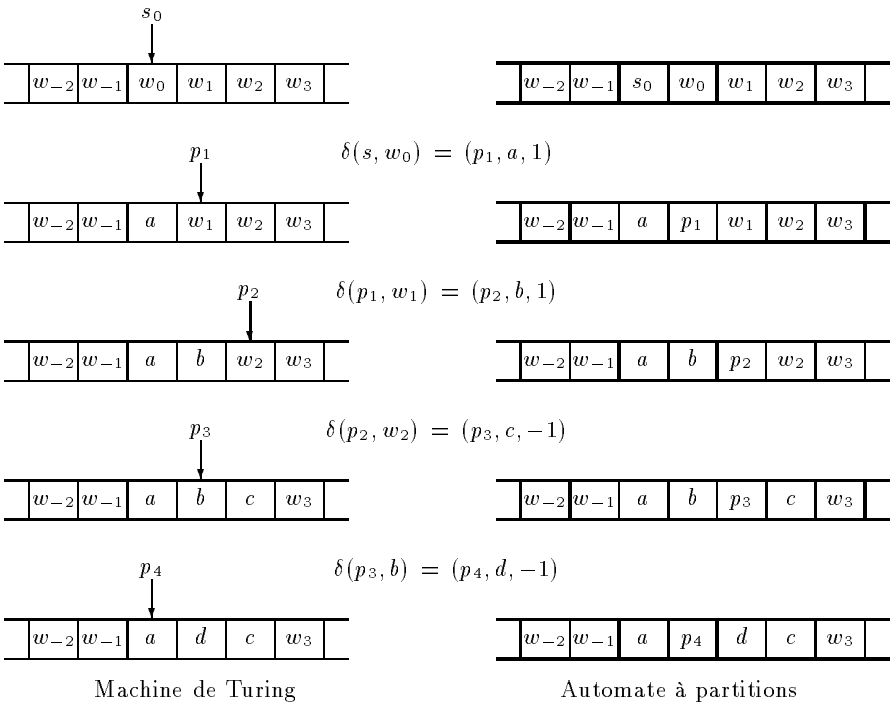


Figure 6 : Règles de transitions pour simuler une machine de Turing.

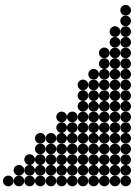
La transition locale se définit à partir de la fonction de transition de la machine de Turing comme indiqué sur la figure 6. La présence de la tête sur une case correspond à la présence de l'état et du symbole dans la même tuile.

Quelques itérations sont décrites sur la figure 7.



Les lignes grasses indiquent la partition itérée.

Figure 7 : Simulation d'une MT par un AP.



Chaque itération de l'AP fait deux itérations de la machine de Turing. La fin du calcul correspond au passage à l'état FIN.

Q. E. D.

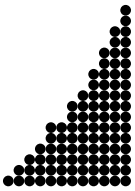
La simulation est en temps réel. Si l'on n'écrit pour l'AP que les transitions définies par la figure 6, l'écriture des tables est du même ordre de grandeur que la table : $\Omega(m.n.\ln(m.n))$.

Il est possible d'obtenir une seule itération de MT par itération de l'AP. Pour cela, il faut rajouter un temporisateur à chaque état représentant un état de la MT.

Il existe aussi des AC universels pour le calcul, il s'agit d'un résultat classique, *e.g.* [1, 53, 56, 84, 93, 103, 105, 106, 119, 120, 133] et [103] pour les AC-R. Il existe même une simulation directe de n'importe quelle MT par des AC-R de dimension 1 [34].

L'AP construit est de taille minimale (2) et a le moins de partitions possible (2) pour que l'information puisse circuler. Il a besoin de $m + n + 1$ états pour simuler une machine de Turing à m états et n symboles. Grâce aux travaux de Margenstern [88] regroupant des résultats de Rogozhin, écrits en russe, on sait qu'il existe une machine de Turing universelle avec 5 états et 5 symboles. Il existe donc un AP universel à 11 états, géométriquement minimal. Ce nombre d'états doit pouvoir être diminué.

En dimension 2, le Billiard ball model décrit dans le chapitre 4 est minimum géométriquement, ne possède que deux états et est réversible en plus.



Chapitre 3

Bi-Simulation entre Automates à Partitions et Automates Cellulaires

Dans ce chapitre, nous construisons des simulations en temps réel entre automates à partitions et automates cellulaires. Nous faisons de même avec leurs sous-classes réversibles. Nous nous intéressons à la complexité de ces simulations et en donnons des exemples.

3.1 Simulation d'Automates à Partitions par des Automates Cellulaires

Nous avons un théorème de simulation uniforme :

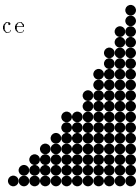
Théorème 22 *Tout automate à partitions s'identifie naturellement à un automate cellulaire dont les cellules sont des matrices de tuiles.*

Preuve. Soit $\mathcal{P} = (\ \mathcal{Q},\ (h, v),\ n,\ (x_i, y_i)_{1 \leq i \leq n},\ t\ )$, le plus général des AP. Nous construisons un AC A le simulant.

Les cellules sont les tuiles de la première partition. Ceci permet de s'affranchir du problème des positions relatives des nœuds dans les différentes partitions : elles sont les mêmes pour toutes les cellules. L'ensemble des états de l'AC A est l'ensemble des tuiles : $\mathsf{Q} = \mathcal{Q}^{h \times v}$.

Pour tout entier k, chaque matrice de $k \times k$ tuiles de la première partition –soit $kh \times vk$ nœuds– contient des sous-matrices de $(k-1) \times (k-1)$ tuiles de chacune des partitions. De plus ces sous-matrices contiennent la partie centrale de $(k-2) \times (k-2)$ tuiles de la première partition.

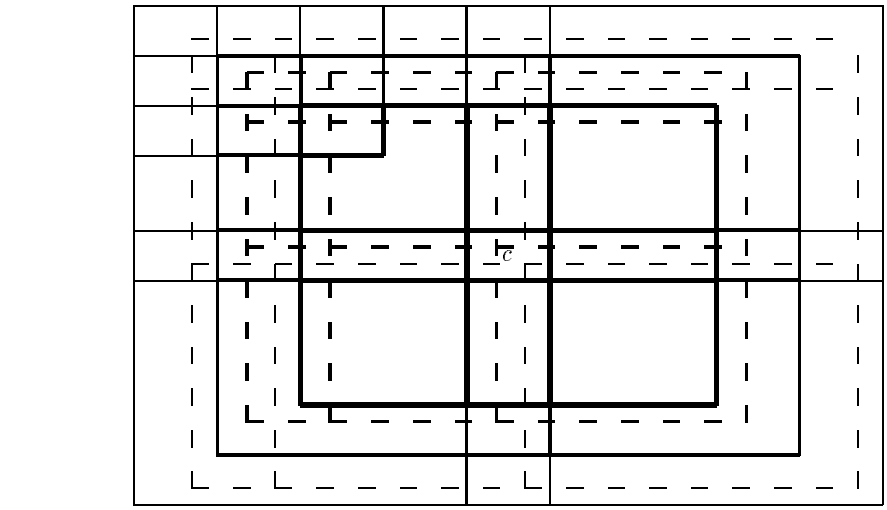
A partir de là, sur toute matrice de $k \times k$ tuiles, nous pouvons faire une transition sur une matrice extraite de taille $(k-1) \times (k-1)$ d'une partition quelconque et retrouver une sous-matrice



de $(k-2)\times(k-2)$ tuiles de la partition originelle. Nous pouvons réitérer ceci pour faire deux transitions et obtenir une matrice de $(k-4)\times(k-4)$ tuiles de la première partition au milieu.

En prenant une matrice de $(2n+1)\times(2n+1)$ tuiles, nous pouvons donc faire les n transitions. A la fin, nous obtenons l'image de la tuile centrale après toutes les transitions —la taille est devenue 1×1. Toute la transition globale a été effectuée. C'est ainsi que se définit la fonction locale f. Le rayon A est donc n, le nombre de partitions de $\mathcal{P}$. L'AC est $\mathsf{A} = \left(\mathcal{Q}^{h\times v},\ n,\ \mathsf{f}\right)$. Les premiers découpages sont représentés sur la figure 8.

Q. E. D.



———	$(2n+1)\times(2n+1)$	tuiles de la première partition (cellules $[-n,n]\times[-n,n]$).
– –	$2n\times 2n$	tuiles de la seconde partition.
———	$(2n-1)\times(2n-1)$	tuiles de la première partition (cellules $[-n+1,n-1]\times[-n+1,n-1]$).
– –	$(2n-2)\times(2n-2)$	tuiles de la troisième partition.
———	$(2n-3)\times(2n-3)$	tuiles de la première partition (cellules $[-n+2,n-2]\times[-n+2,n-2]$).

Figure 8 : Différents découpages.

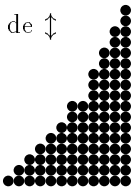
Il y a véritablement identification entre un AP et l'AC qui le simule. Les configurations sont en bijection par simple groupage – dégroupage. Il s'ensuit directement le théorème suivant :

Théorème 23 *Tout* AP *réversible est simulable par un* AC *réversible.*

La construction est la même, la fonction locale est la suite d'extractions et de transitions, qui vient d'être décrite. La réversibilité de l'AC est assurée par le fait que les fonctions de simulation et la transition globale de l'AP-R le sont.

Exemple : Simulation d'une Règle Majoritaire

Prenons l'AP de dimension 2 suivant : $(\ \{\updownarrow, \leftrightarrow\},\ (2,2),\ 2,\ ((0,0),(1,1)),\ t_1,\ )$. Les nœuds suivent localement la majorité : dans chaque tuile, s'il y a plus dans un état que dans l'autre, alors ils se mettent tous dans l'état majoritaire. Pour une tuile b, $\#_{\updownarrow}(b)$ ($\#_{\leftrightarrow}(b)$) est le nombre de $\updownarrow$



($\leftrightarrow$) qu'elle contient. La transition locale t_1 se définit comme suit :

$$\forall b \in \{\updownarrow, \leftrightarrow\}^4,\quad t_1(b) \;=\; \begin{cases} \begin{array}{|c|c|}\hline \updownarrow & \updownarrow \\ \hline \updownarrow & \updownarrow \\ \hline \end{array} & \textbf{si } \#_{\leftrightarrow}(b) < \#_{\updownarrow}(b)\ , \\ b & \textbf{si } \#_{\updownarrow}(b) = \#_{\leftrightarrow}(b)\ , \\ \begin{array}{|c|c|}\hline \leftrightarrow & \leftrightarrow \\ \hline \leftrightarrow & \leftrightarrow \\ \hline \end{array} & \textbf{si } \#_{\updownarrow}(b) < \#_{\leftrightarrow}(b)\ . \end{cases}$$

L'AC simulant cet AP est : $(\{\updownarrow, \leftrightarrow\}^4, 1, \mathsf{f}_1)$. Nous montrons sur l'exemple de la figure 9 comment se calcule f_1 : on fait les deux transitions sur le voisinage et on récupère la partie centrale.

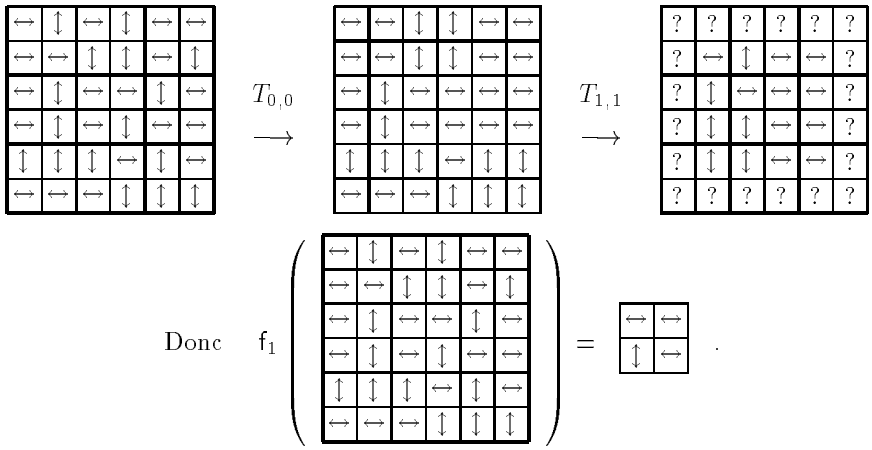


Figure 9 : Calcul d'une image de la fonction locale pour la règle majoritaire.

3.2 Simulation d'Automates Cellulaires par des Automates à Partitions

Nous démontrons la réciproque du théorème 22 :

Théorème 24 *Tout automate cellulaire peut être simulé par un automate à partitions.*

Preuve. Soit $\mathsf{A} = (\mathsf{Q}, \mathsf{r}, \mathsf{f})$ un AC quelconque. Nous construisons l'AP suivant :

$$\mathcal{P} \;=\; \big(\mathsf{Q} \cup \mathsf{Q}^2,\ (4\mathsf{r}, 4\mathsf{r}),\ 4,\ ((0,0), (2\mathsf{r}, 0), (0, 2\mathsf{r}), (2\mathsf{r}, 2\mathsf{r})),\ t\ \big)\ .$$

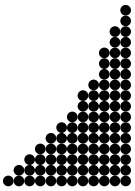
Ils travaillent sur la même grille ($\mathcal{L} = \mathsf{L}$), mais les états sont différents, $\mathcal{P}$ utilisant aussi les états doubles : $(p, q) \in \mathsf{Q}^2$. Les tuiles sont 2×2 plus grandes que le voisinage de A. Elles sont coupées en quatre matrices $2\mathsf{r} \times 2\mathsf{r}$ (M, H, V, C) :

M : cellules/nœuds du milieu ayant toutes leurs voisines dans la tuile,

V : cellules ayant toutes leurs voisines avec les tuiles du haut et du bas,

H : cellules ayant toutes leurs voisines avec les tuiles sur les côtés,

C : cellules ayant toutes leurs voisines avec les huit tuiles contiguës.



Ce découpage est illustré par la figure 10. Dans la suite de la démonstration, les tuiles sont notées (M, H, V, C). Chacune des sous-matrices $2\mathsf{r} \times 2\mathsf{r}$ a une valeur dans $\mathcal{Q}^{4\mathsf{r}^2}$. Nous en distinguons deux types : celles ayant toutes leurs valeurs dans Q (appartenant à $\Theta = \mathsf{Q}^{4\mathsf{r}^2}$) et celles ayant toutes leurs valeurs dans Q^2 (appartenant à $\Omega = \mathsf{Q}^{8\mathsf{r}^2}$).

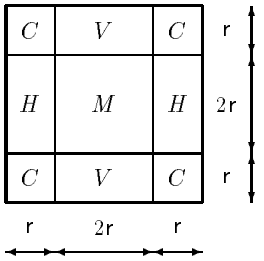


Figure 10 : Partition des tuiles.

Chaque configuration c de A est codée de manière canonique en une configuration d de $\mathcal{P}$ par identification : $\forall x \in \mathcal{L}, d_i = \mathsf{c}_i$. La simulation se fait en deux temps : ajouter à chaque cellule de la valeur de son prochain état, puis effacer toutes les anciennes valeurs.

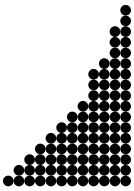
Pour chacune des cellules du milieu M par choix de la taille des tuiles, son voisinage se trouve intégralement dans la tuile. Il est possible de calculer les nouvelles valeurs et de les stocker comme second état sur les cellules. Leurs anciennes valeurs sont conservées comme premier état pour servir au calcul des nouvelles valeurs des autres cellules.

Une fois que les nouvelles valeurs ont été rajoutées pour M, il suffit de changer de partition pour faire de même pour H, puis V et finalement C. Quand toutes les nouvelles valeurs ont été rajoutées aux anciennes, il suffit d'effacer les anciennes pour obtenir la configuration-image par l'AC A.

$$
\begin{array}{l}
\textbf{si } (M,H,V,C) \in \Theta\times\Theta\times\Theta\times\Theta \\
\qquad\qquad\quad \cup \Theta\times\Omega\times\Theta\times\Theta \\
\qquad\qquad\quad \cup \Theta\times\Theta\times\Omega\times\Omega \\
\quad \textbf{alors } t(M,H,V,C) = (\gamma(M,H,V,C),H,V,C) \\
\textbf{sinon si } (M,H,V,C) \in \Theta\times\Omega\times\Omega\times\Omega \\
\quad \textbf{alors } t(M,H,V,C) = \pi_2^2(\gamma(M,H,V,C),H,V,C) \\
\textbf{sinon } t(A) = A
\end{array}
$$

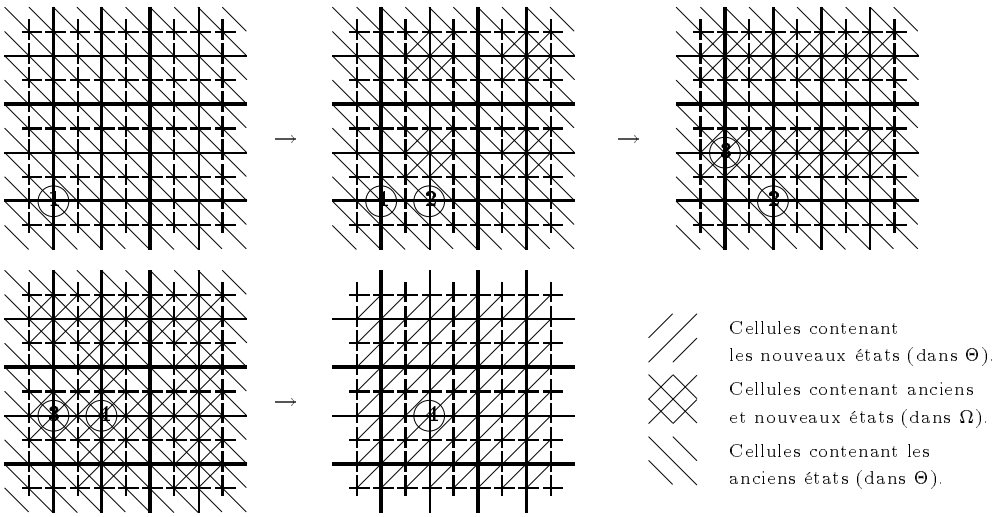
Figure 11 : Transition locale pour la simulation d'un AC par un AP.

Tout ceci se trouve synthétisé dans la transition locale de la figure 11, où γ est la fonction renvoyant la matrice du milieu avec les valeurs doublées par les nouvelles calculées grâce à l'ensemble de la tuile. Nous notons π_2^2 la seconde projection ($\pi_2^2(p,q) = q$) généralisée à la tuile entière. Elle sert à effacer les anciennes valeurs des cellules. A chaque fois la tuile (M, H, V, C) correspond à la partition courante. Avec les changements de partitions, M représente tour à tour M, puis H, puis V et finalement C des tuiles de la première partition.



Au début de la simulation, il n'y a que des états dans Q, soit $(M, H, V, C) \in \Theta \times \Theta \times \Theta \times \Theta$. Après la première transition, la tuile passe dans $\Omega \times \Theta \times \Theta \times \Theta$. Pour la seconde partition, la tuile sera dans $\Theta \times \Omega \times \Theta \times \Theta$ et passera à $\Omega \times \Omega \times \Theta \times \Theta$. Pour la troisième, de $\Theta \times \Theta \times \Omega \times \Omega$ à $\Omega \times \Theta \times \Omega \times \Omega$. Pour la quatrième, de $\Theta \times \Omega \times \Omega \times \Omega$ à $\Omega \times \Omega \times \Omega \times \Omega$ pour être projetée, en ne gardant que les secondes coordonnées, sur $\Theta \times \Theta \times \Theta \times \Theta$. Ceci est schématisé par les différents grillages de la figure 12.

Q. E. D.



Les numéros entourés correspondent aux origines des partitions.

Figure 12 : Ordre d'apparition des nouveaux états.

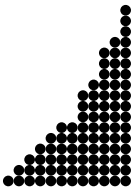
Exemple : Simulation du Jeu de la Vie

Le jeu de la vie est l'AC de dimension 2 suivant : $(\ \{\ ' \ ', \bullet\},\ 1,\ \mathsf{f}_2\ )$. Il y a beaucoup de littérature sur les caractéristiques de cet AC, de l'article de Conway [22] à la page web : `http://members.aol.com/life1ine/life/lifepage.htm`.

Sa fonction locale, f_2, est définie à partir de la valeur de la cellule et de celles ses huit voisines. Si la cellule vaut $\bullet$, elle le reste si et seulement si deux ou trois de ses voisines valent $\bullet$. Si la cellule vaut $' \ '$, elle devient $\bullet$ si et seulement si trois de ses voisines valent $\bullet$. La figure 13 donne quelques valeurs de f_2.

L'AP le simulant est $(\ \{\ ' \ ', \bullet\} \cup \{\ ' \ ', \bullet\}^2,\ (4,4),\ 4,\ ((0,0),(2,0)(0,2)(2,2)),\ t_2\ )$. Plutôt que de détailler complètement t_2, nous donnons sa valeur sur les quelques exemples de la figure 14.

A chaque fois, les nouveaux états apparaissent dans le centre. Le déplacement des partitions permet de changer la position des centres sur la grille. Dès que tous les nouveaux états sont présents, les anciens sont effacés.



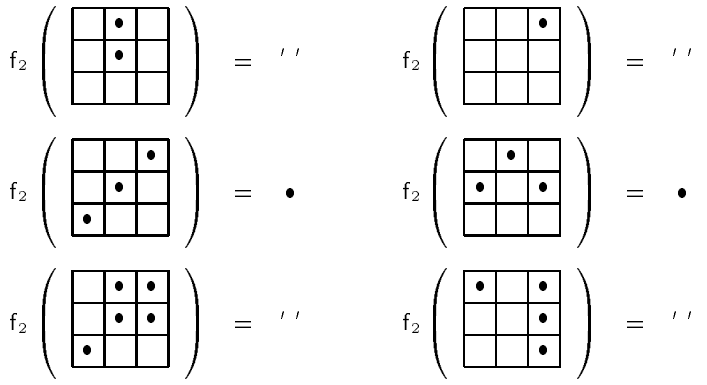


Figure 13 : Exemples de la fonction locale du jeu de la vie.

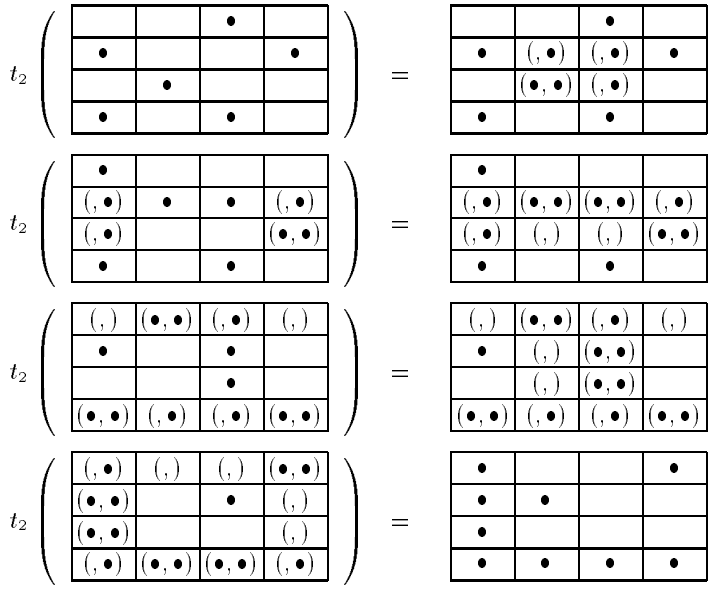


Figure 14 : Exemples de calcul de la transition locale simulant le jeu de la vie.

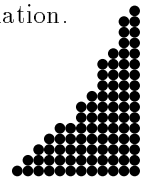
3.3 Simulation AC Réversible par AP Réversible

Nous démontrons la réciproque du théorème 23 :

Théorème 25 *Tout automate cellulaire réversible est simulable par un automate à partitions réversible.*

Preuve. Soit A un automate cellulaire réversible et A^{-1} son inverse. Prenons r assez grand pour convenir aux deux automates —augmenter un rayon ne fait que demander plus de valeurs pour calculer, même si celles-ci ne servent à rien.

La transition locale t opère sur un ensemble fini. Si elle est définie partiellement mais injectivement, elle peut toujours être complétée de manière bijective. Dans dans le premier cas de l'algorithme de la figure 11, elle est injective : elle ne fait que rajouter des valeurs, de l'information.



Celles-ci sont redondantes car elles sont définies à partir des autres. Par contre, la partie projective n'est pas injective : les premières coordonnées sont effacées quelles qu'elles soient. Deux tuiles ne différant que par leurs premières coordonnées ont la même image.

Pour pallier cela, il faut faire un effacement sélectif : n'effacer que les valeurs qui sont effectivement redondantes. C'est là où interviennent l'automate inverse et la taille suffisante du rayon. Les anciennes valeurs peuvent être calculées à partir des nouvelles grâce à A^{-1}, de la même façon que les nouvelles valeurs ont été calculées à partir des anciennes grâce à A.

Pour effacer les données sans pour autant perdre l'injectivité, les anciennes valeurs sont testées avant d'être effacées. Si elles correspondent à celles obtenues avec les nouvelles, elles passent à la trappe. Bien sûr, comme pour la génération des nouvelles valeurs, le calcul n'est possible que pour les cellules du milieu M.

Il faut donc visiter à nouveau, toutes les partitions pour effacer, morceau par morceau les anciennes valeurs. Le nouvel AP est le suivant :

$$\mathcal{P}' \;=\; \left(\mathsf{Q} \cup \mathsf{Q}^2,\ (4\mathsf{r}, 4\mathsf{r}),\ 7,\ \mathcal{O},\ t'\ \right) \quad ,$$

avec

$$\mathcal{O} \;=\; \left((0,0), (2\mathsf{r},0), (0,2\mathsf{r}), (2\mathsf{r},2\mathsf{r}), (0,0), (2\mathsf{r},0), (0,2\mathsf{r})\right) \quad .$$

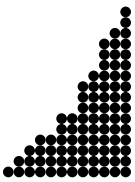
La figure 15 donne le détail de t'. Le prédicat $\mathcal{P}(\mathcal{M},\mathcal{H},\mathcal{V},\mathcal{C})$ est vrai si et seulement s'il y a des états doubles dans M et si les anciennes valeurs dans M (les premières) correspondent à celles calculées avec les nouvelles dans la tuile (la valeur est la seconde s'il y en a deux) et l'automate inverse A^{-1}. C'est le prédicat $\mathcal{P}$ qui assure l'injectivité de l'effacement.

Q. E. D.

si $(M,H,V,C) \in \Theta\times\Theta\times\Theta\times\Theta$
$\cup\ \Theta\times\Omega\times\Theta\times\Theta$
$\cup\ \Theta\times\Theta\times\Omega\times\Omega$
alors $t'(M,H,V,C) = (\gamma(M,H,V,C),H,V,C)$
sinon si $\mathcal{P}(\gamma(\mathcal{M},\mathcal{H},\mathcal{V},\mathcal{C}),\mathcal{H},\mathcal{V},\mathcal{C})$ **et** $(M,H,V,C) \in \Theta\times\Omega\times\Omega\times\Omega$
alors $t'(M,H,V,C) = (\pi_2^2(\gamma(M,H,V,C)),H,V,C)$
sinon si $\mathcal{P}(\mathcal{M},\mathcal{H},\mathcal{V},\mathcal{C})$ **et** $(M,H,V,C) \in \Omega\times\Omega\times\Omega\times\Theta$
$\cup\ \Omega\times\Theta\times\Theta\times\Omega$
$\cup\ \Omega\times\Theta\times\Theta\times\Theta$
alors $t'(M,H,V,C) = (\pi_2^2(M),H,V,C)$
sinon { complété bijectivement }

Figure 15 : Transition locale pour la simulation d'un AC-R par un AP-R.

Remarque 26 Le fait qu'il y ait simulation entre AC et AP d'une part, et AC-R et AP-R d'autre part, n'est pas contradictoire avec le lemme 13 et le théorème 17 du chapitre 2, en dimension supérieure à 2, la réversibilité des AP est décidable alors que celle des AC ne l'est pas. Ceci ne gêne en rien les équivalences de classes car la simulation d'un AC par un AP se fait dans un ensemble de configurations plus grand où la réversibilité se perd. Pour la simulation des sous-classes réversibles, la connaissance de l'AP inverse est nécessaire.



Ce théorème est la démonstration de la conjecture 8.1 de l'article [138] de Toffoli et Margolus.

Kari [77] a démontré que les AC-R pouvaient être représentés grâce à des permutations de blocs en dimensions 1 et 2. Dans cet article, il a conjecturé (conjecture 5.3) que cela était vrai pour toute dimension d avec 2^d permutations de blocs. Ce qu'il nomme « Block permutation » est une transition pour nous. Notre résultat est que cela est toujours possible avec $2^{d+1}-1$ transitions. Nous avons fait une nouvelle version [37] de notre construction avec seulement 2^d et un effaçage progressif des anciennes valeurs, comme il le fait dans son article. Le coût n'est pas négligeable: la taille est $(6\mathsf{r}, 6\mathsf{r})$ au lieu de $(4\mathsf{r}, 4\mathsf{r})$.

3.4 Complexité

Toutes les simulations sont en temps réel, c'est-à-dire qu'une itération de l'un correspond à une itération de l'autre. Ce n'est pas un résultat très impressionnant car on prend un automate particulier pour chaque automate à simuler. Avec BBM, dans le chapitre suivant, il en sera tout autrement.

Nous définissons la complexité d'un automate comme la place qu'il faut pour le définir. Ce qui prend de la place est la définition de la fonction locale (ou de la transition locale). L'alphabet se définit par son nombre d'états, tout comme le rayon, le nombre et les origines des partitions qui se définissent par des entiers. Ils occupent une place négligeable en comparaison avec celle de la table, à moins d'être dans un cas dégénéré où il n'y aurait qu'un état ou un nombre déraisonnable de partitions.

La complexité d'un AC est donc :

$$\log(|\mathsf{Q}|)\,|\mathsf{Q}|^{(2\mathsf{r}+1)^2} \quad (+\log|\mathsf{Q}| + \log \mathsf{r}) \quad .$$

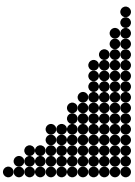
Et pour un AP, elle vaut :

$$\log(|\mathcal{Q}|)\,h\,v\,|\mathcal{Q}|^{h\,v} \quad (+\log|\mathcal{Q}| + \log(h\,v) + \log n + n\log(h\,v)) \quad .$$

Avec notre construction, l'AC simulant un AP avec m états, n partitions et une taille (h, v) possède un nombre exponentiel d'états : $m^{h\,v}$ et un rayon égal à n. On passe donc de $\log(m)\,h\,v\,m^{h\,v}$ à $h\,v\,\log(m)\,m^{h\,v(2n+1)^2}$, soit pratiquement à la puissance $(2\,n+1)^2$.

L'AP simulant un AC à m états et de rayon r possède $\mathsf{m}+\mathsf{m}^2$ états et 4 partitions de taille $(4\mathsf{r}, 4\mathsf{r})$. La complexité passe de $\log(\mathsf{m})\,\mathsf{m}^{(2\mathsf{r}+1)^2}$ à $\log(\mathsf{m}+\mathsf{m}^2)\,16\mathsf{r}^2\,(\mathsf{m}+\mathsf{m}^2)^{16\mathsf{r}^2}$, soit pratiquement mise à la puissance 8 et multipliée par $32\mathsf{r}^2$.

Dans le cas où l'on fait une simulation réversible, il y a alors 7 partitions mais le rayon est le plus grand de celui de l'AC et de celui de son inverse. A cause des résultats d'indécidabilité (théorème 17), il n'existe pas de fonction récursive bornant le rayon de l'inverse à partir du nombre d'états et du rayon d'un AC. Si l'on considère l'AC-R défini sur le grand rayon, l'accroissement de la complexité est le même puisque le nombre de partitions n'intervient pas.



Complexité de Kolmogorov

On définit la complexité de Kolmogorov d'un automate A $\mathcal{K}_A$, comme la taille de la plus petite machine de Turing qui, partant sur un ruban entièrement blanc y écrit le code de A. Consulter le chapitre 4 de [80] et [78, 82] pour plus d'information. Toutes les simulations réalisées plus haut correspondent à des algorithmes. Il existe une MT qui, partant avec le code de l'automate à simuler, écrit le code de l'automate simulant. Soit k la taille de cette machine, par composition des machines de Turing, on peut en construire une qui écrit le code de l'automate simulant B. Sa taille est $\mathcal{K}_A + k$.

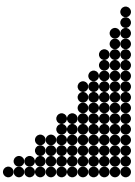
Il existe une MT qui donne l'inverse d'un AC-R. Quand on sait qu'un AC est réversible, pour trouver son inverse, il suffit d'essayer tous les AC et ils se dénombrent facilement. Par contre, sur un AC non réversible, cette MT ne s'arrêtera jamais. Dans le cas où l'on veut simuler un AC-R par un AP-R, il faut aussi composer avec cette MT, car la simulation nécessite les deux AC.

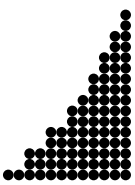
Théorème 27 *Pour chacune des simulations, il existe une constante k telle que, si B est l'automate construit pour simuler A alors :*

$$\mathcal{K}_B \leq \mathcal{K}_A + k \ .$$

Il n'y a pas vraiment d'augmentation de la complexité de Kolmogorov. Le raisonnement que nous venons de faire peut être refait pour bon nombre de simulations, où c'est toute une classe qui simule une autre classe, et non un seul automate.

Ce théorème signifie que les automates simulant construits sont relativement simples. Leur définition est beaucoup plus succinte que leur taille pourrait le laisser croire.





Chapitre 4

Modèle de la Boule de Billard

Pour modéliser des gaz et des plasmas, Toffoli et Margolus ont introduit et étudié le modèle de la boule de billard, ou « Billiard ball model » (BBM) [45,90,134,137]. C'est un automate à partitions réversible. Son nom vient de l'observation de signaux se déplaçant en ligne droite (suivant les diagonales) et rebondissant perpendiculairement à leurs trajectoires en cas de choc.

Toffoli et Margolus ont prouvé l'universalité de cet automate, mais d'une manière qui ne permet pas de distinguer entre l'absence de signal et le signal zéro. Ceci est gênant pour effectuer des calculs récursifs car la présence du signal zéro n'est pas discernable.

Dans ce chapitre, nous rappelons leurs définitions et techniques. Nous doublons leurs signaux faisant des constructions de plus en plus abstraites. Nous redémontrons l'universalité pour le calcul et finissons ce chapitre en démontrant son universalité intrinsèque (BBM est capable de simuler tous les AP réversibles).

4.1 Définition

Il n'y a que deux états : le vide et le plein, la matière étant symbolisée par la *boule* $\bullet$.

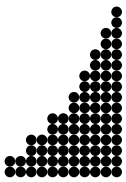
$$\text{BBM} = (\ \{\_, \bullet\},\ (2, 2),\ 2,\ ((0,0),(1,1)),\ t_{\text{BBM}}\ )\quad .$$

La transition locale t_{BBM} est partiellement définie par la figure 16. Pour la définir totalement, il faut la compléter par rotations et symétries.

Elle se comprend comme suit.

- Le vide, comme le plein, se conserve.
- S'il n'y a qu'une boule, elle passe dans le coin opposé.
- S'il y a deux boules disposées selon la diagonale, elles changent de diagonale. Par contre, si elle sont sur une même moitié, elles restent en place.
- S'il y a trois boules, rien ne se passe.

La transition locale de BBM, t_{BBM} conserve le nombre de $\bullet$. Elle est bijective, invariante par symétrie, par rotation. Il n'y a aucune orientation de l'espace.



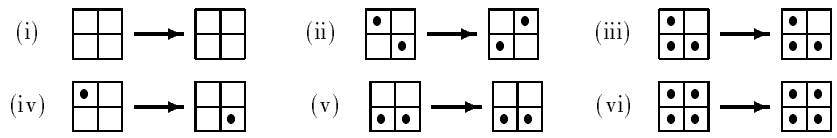


Figure 16 : Règles de base de t_{BBM}.

Proposition 28 BBM *est réversible.*

Preuve. c'est une simple application du lemme 13 car t_{BBM} est bijective.

Q. E. D.

4.2 Architecture de Base

Nous utilisons deux niveaux de codage pour coder des signaux logiques : le codage *basique* de Toffoli et Margolus et le codage *dual* que nous introduisons ici.

La figure 17 montre un exemple d'itération de BBM. Les deux règles (i) et (iv) de la figure 16, suffisent à créer un signal : une boule qui se déplace. Cela sert de base aux constructions suivantes.

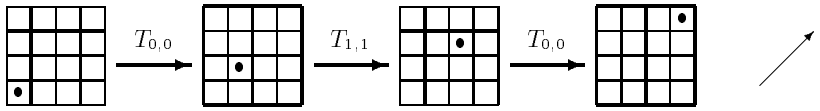


Figure 17 : Mouvement d'une boule.

Codage Basique

Toute cette sous-section est un rappel du codage donné par Toffoli et Margolus dans [137].

Nous avons vu qu'une boule seule se déplace en ligne droite (figure 17). Elle peut servir de signal, mais pour cela, il faut pouvoir la faire changer de direction et interagir avec d'autres signaux.

Pour résoudre ce premier problème: le changement de direction, il suffit de faire suivre la boule par une autre boule, et de les envoyer vers un rectangle inamovible. Elle rebondit en quatre temps comme le détaille la figure 18. La règle clef est la règle (ii) de la figure 16.

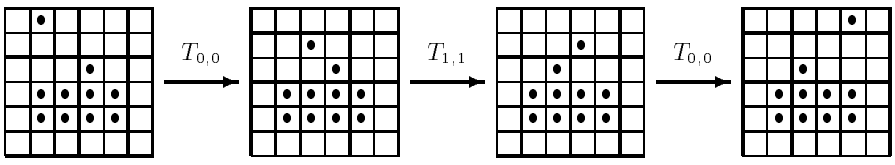
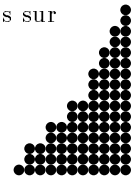


Figure 18 : Réflexion d'un signal.

Un signal est dorénavant une séquence de deux boules. Les signaux se déplacent selon les diagonales, dans les deux sens. Leurs directions peuvent être changées en mettant des blocs sur



leurs routes pour les faire rebondir. Par exemple, pour fabriquer un délai, il suffit de rallonger le chemin à parcourir, comme illustré par la figure 19.

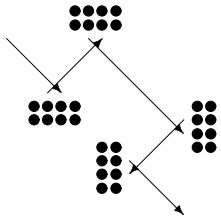


Figure 19 : Délai.

Pour démontrer l'universalité, Toffoli et Margolus ont utilisé une approche binaire. Leur choix a été de prendre deux boules comme signal *1* et rien comme signal *0*, et de travailler avec la logique suivante :

Définition 29 La logique *conservatrice* est la logique restreinte aux fonction binaires réversibles conservant le nombre de 1.

Par exemple, les seules fonctions conservatrices de $\{0,1\}^2$ sont l'identité et la permutation $(s(a,b) = (b,a))$. Pour trouver une fonction ayant un minimum d'intérêt, il faut travailler sur $\{0,1\}^3$. Cette fonction décrite sur la figure 20 laisse passer le premier signal et en fonction de sa valeur permute ou non les deux autres, c'est la *porte de Fredkin*.

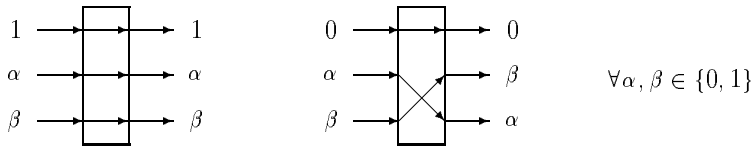


Figure 20 : Porte de Fredkin.

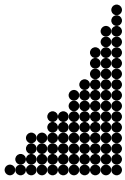
L'étude de cette logique et de cette porte a abouti aux résultats suivants :

Lemme 30 (Fredkin et Toffoli 1982 [45]) *Toute fonction binaire peut être construite à partir de la porte de Fredkin.*

Le problème est que cela nécessite le passage par une fonction sur un plus grand domaine, avec l'apport de constantes et l'évacuation de sorties non nécessaires. Cela se comprend bien, comment obtenir une fonction non conservatrice avec un constructeur qui l'est. Morita a affiné ce lemme :

Lemme 31 (Morita 1990 [101]) *Il est possible de construire n'importe quelle fonction conservatrice avec seulement des portes de Fredkin et des constantes* 0 *en entrée. Ces* 0 *étant régénérés à la fin.*

Toffoli et Margolus ont démontré qu'il était possible de simuler une porte de Fredkin avec le codage de base. Les *0* étant représentés par rien du tout, il en découle :



Théorème 32 (Toffoli et Margolus 87 [137]) BBM *est capable de simuler n'importe quelle fonction binaire conservatrice (avec le codage de base) sans apport de constantes.*

Remarquons au passage que bien que cet automate soit très simple il nécessite beaucoup de temps et d'espace pour faire le calcul d'une unique porte de Fredkin. A moins d'une extrême rapidité et compacité, la réalisation de BBM ne serait que de peu d'utilité.

Codage Dual

La construction précédente est intéressante, tant que l'on manipule des portes logiques et des fonctions dont le temps de calcul est fixe. Mais si l'on ignore quand doit arriver la réponse, on ne peut savoir si la réponse vaut *0* et est arrivée, ou bien si la réponse n'est pas encore arrivée. Ceci est particulièrement gênant quand on simule une machine de Turing, car celle-ci peut s'arrêter n'importe quand, ou même ne jamais s'arrêter.

Pour remédier à cela, nous avons changé l'implantation des signaux en les doublant. Ceci permet aussi de passer de la logique conservatrice à la logique réversible. C'est une technique que l'on retrouve dans [54] pour prouver l'universalité du « Chip firing game ».

s^+	s^-	s
0	0	Pas de signal
0	1	**0**
1	0	**1**
1	1	Erreur

Figure 21 : Du codage basique au codage dual.

Nous doublons chaque signal pour passer du codage basique au dual comme indiqué sur la figure 21. Un signal est toujours composé de deux boules, et leurs trajectoires indiquent la valeur de ce signal. Avec le précédent codage, la présence ou non de boules donne la valeur, l'existence des signaux est explicite.

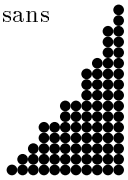
Avec ce codage, nous construisons une porte de Fredkin comme illustré par la figure 22. Des délais nécessaires n'ont pas été indiqués par souci de lisibilité. Il n'y a pas de risque de collision de signaux car il n'y a qu'un et un seul signal réel pour un signal dual.

Il est toujours possible de simuler n'importe quelle fonction de logique conservatrice. Il est maintenant possible de faire une porte `non` (figure 23), ce qui permet de dépasser le cadre de la logique conservatrice pour passer à celui de la logique réversible.

Nous appelons *logique réversible* celle qui ne s'intéresse qu'aux fonctions réversibles, et pas seulement celles conservant le nombre de 1.

Lemme 33 *Il est possible de simuler n'importe quelle fonction binaire réversible grâce au* BBM *sans apport de constantes et sans générer d'autres signaux.*

Preuve. En effet, toute fonction bijective en codage dual correspond à une restriction de fonction conservatrice en codage basique. Or BBM est capable de simuler toute fonction conservatrice sans



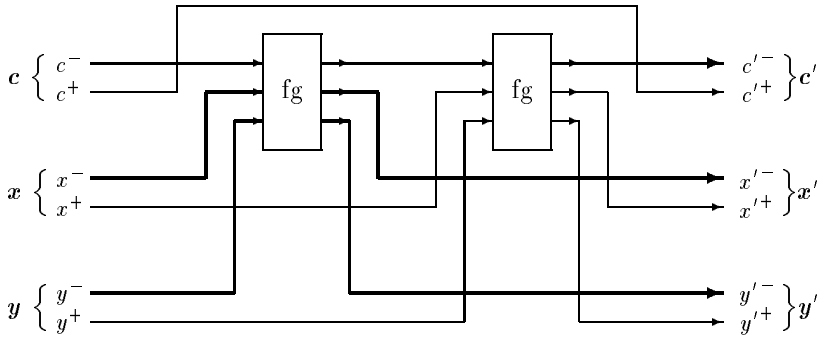


Figure 22 : Porte de Fredkin duale.

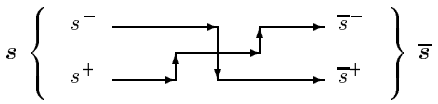


Figure 23 : Porte `non` en codage dual.

apport de constantes (théorème 32).

Q. E. D.

4.3 Universalité au Sens du Calcul

Pour prouver que BBM est capable de simuler n'importe quelle fonction calculable, il suffit de montrer qu'il est capable de simuler n'importe quel automate à deux registres.

Pour plus de précisions sur l'universalité, la calculabilité, les machines de Turing, les automates à deux registres et les liens qui les unissent, se référer à la sous-section 2.6 et au chapitre 14 du livre de Minsky 'Finite and Infinite Machines' [99].

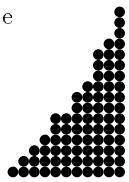
Définition 34 Un *automate à deux registres* est un automate fini relié à deux registres pouvant contenir n'importe quelle valeur entière positive. L'automate peut effectuer, sur chacun des registres, les opérations suivantes : additionner 1, soustraire 1 (ou rien si déjà à 0) et tester la nullité.

Théorème 35 BBM *est capable de simuler n'importe quel automate à deux registres.*

Preuve. La construction repose d'une part sur l'automate, et d'autre part, sur une ligne infinie d'unités logiques entre lesquelles sont coincés les signaux qui codent les registres en unaire.

L'automate est une grosse fonction binaire qui envoie des ordres par des signaux, puis attend un signal de retour d'exécution. Il se réinjecte son état à chaque itération. Il est schématisé sur la figure 24.

Les constantes et signaux poubelles sont nécessaires car la fonction de l'automate n'a vraisemblablement que peu de chance d'être réversible. Ces deux flots se déplaçant perpendiculairement à la demi-ligne de l'automate et des unités. Ces dernières étant totalement autonomes, ils ne



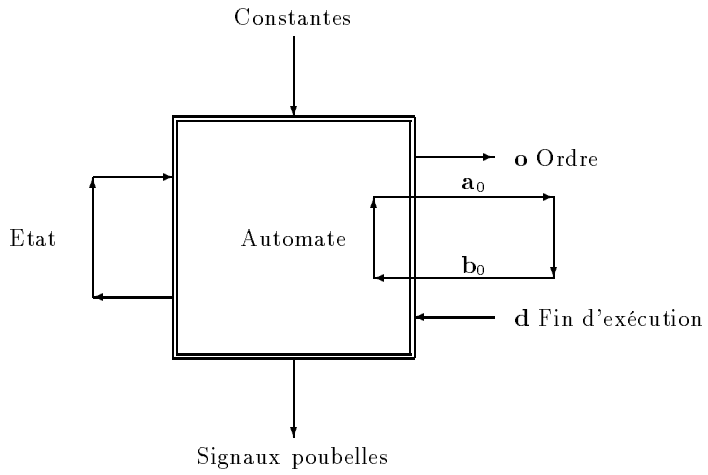


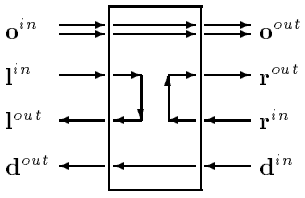
Figure 24 : Automate contrôlant les deux registres.

provoquent aucune perturbation. La ligne des constantes est infinie car on ne peut présumer du temps de calcul (problème de la halte).

L'ordre est codé au moyen d'un signal dual double $\mathbf{o} = (\mathbf{o}_0, \mathbf{o}_1)$. Le signal $\mathbf{o}_0$ est utilisé pour dire qu'il y a un ordre et $\mathbf{o}_1$ pour le déterminer. Le retour d'exécution est un signal $\mathbf{d}$ qui ne vaut $\mathbf{1}$ que lorsqu'un ordre a été exécuté.

Les registres sont représentés en unaire au moyen de signaux duaux : $n \equiv \mathbf{1}^n\mathbf{0}^\omega$. Les deux registres sont notés : $\mathbf{A} = \mathbf{a}_0\mathbf{a}_1\ldots$ et $\mathbf{B} = \mathbf{b}_0\mathbf{b}_1\ldots$ Ces signaux sont stockés dans une ligne infinie. Ils sont coincés entre des unités identiques à celle décrite dans la figure 25. Les signaux $\mathbf{a}_0$ et $\mathbf{b}_0$ ne font que tourner en entrée. Ceci permet à l'automate de tester lui-même la nullité de chacun des registres.

La partie difficile est la gestion des registres. Ceux-ci sont définis comme de petites unités logiques toutes identiques communiquant entre elles avec les signaux $\mathbf{o}$, $\mathbf{l}$, $\mathbf{r}$ et $\mathbf{d}$. Elles correspondent à la fonction de la figure 25. Il est à signaler qu'à défaut d'être conservatrices, ces unités sont réversibles.

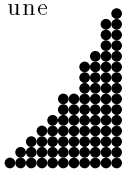


$\mathbf{o}^{in}$	$\mathbf{l}^{in}$	$\mathbf{r}^{in}$	$\mathbf{d}^{in}$	$\mathbf{o}^{out}$	$\mathbf{l}^{out}$	$\mathbf{r}^{out}$	$\mathbf{d}^{out}$
0,0	*	*	*	0,0	*	*	*
1,*	1	1	0	1,*	1	1	0
1,0	1	0	0	0,1	0	0	1
1,1	1	0	0	0,1	1	1	1
0,1	*	*	0	0,1	*	*	0

$*$: inchangé

Figure 25 : Unité de registre et fonction binaire associée.

Les unités fonctionnent en modifiant $\mathbf{l}$ ou $\mathbf{r}$ selon leurs valeurs et celles de l'ordre $\mathbf{o}$. Il n'y a modification que si l'on est en fin de registre ($\mathbf{l} = \mathbf{1}$ et $\mathbf{r} = \mathbf{0}$) et s'il y a un ordre à effectuer ($\mathbf{o}_0 = \mathbf{1}$). La modification à faire est alors indiquée par $\mathbf{o}_1$: $\mathbf{0}$ pour une soustraction et $\mathbf{1}$ pour une



addition. Pour soustraire 1 on met $\mathbf{l}$ à $\mathbf{0}$. Pour additionner 1 on met $\mathbf{r}$ à $\mathbf{1}$.

Le signal $\mathbf{d}$ vaut toujours $\mathbf{0}$ sauf lorsqu'il convoie le message que l'opération a été exécutée, auquel cas il vaut $\mathbf{1}$. Il est mis à $\mathbf{1}$ quand un élément effectue un ordre. Il n'y a jamais plus d'un ordre actif ($\mathbf{o}_t = (\mathbf{1}, .)$) ou signal de fin d'exécution ($\mathbf{d}_t = \mathbf{1}$) en même temps dans l'ensemble des unités.

L'automate et les unités sont connectés entre eux comme sur la figure 26 où l'on voit clairement que les unités ne contiennent que des $\mathbf{a}$, ou que des $\mathbf{b}$, successifs. A chaque instant, $(\mathbf{l}^{in}, \mathbf{r}^{in})$ vaut soit $(\mathbf{a}_k, \mathbf{a}_{k+1})$ (et $(\mathbf{l}^{out}, \mathbf{r}^{out})$ $(\mathbf{b}_k, \mathbf{b}_{k+1})$), soit $(\mathbf{b}_k, \mathbf{b}_{k+1})$ (et $(\mathbf{l}^{out}, \mathbf{r}^{out})$ $(\mathbf{a}_k, \mathbf{a}_{k+1})$), selon la parité du temps.

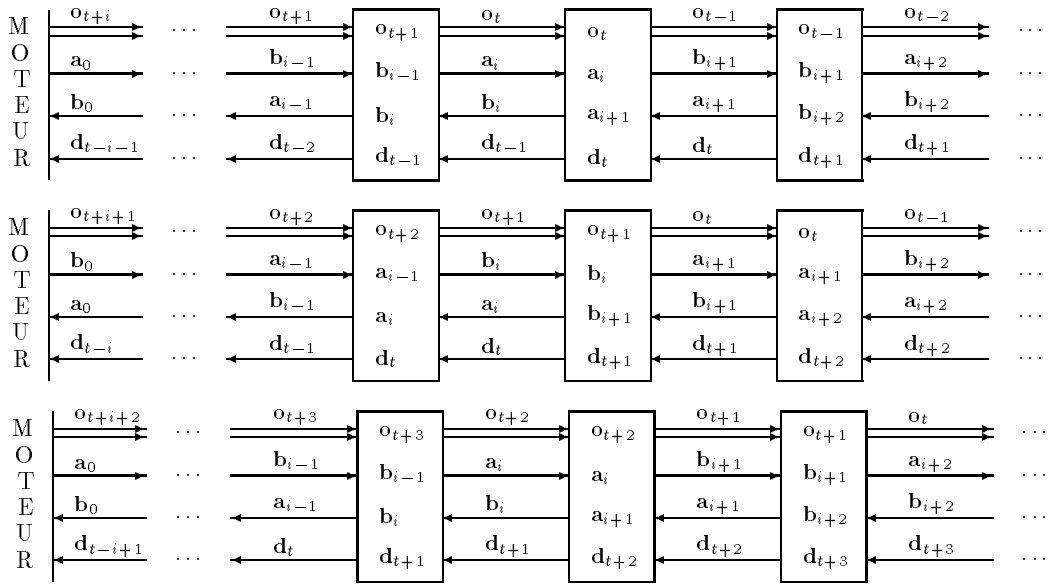


Figure 26 : Automate, unités et connexions pour trois itérations successives.

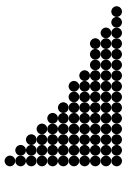
De même, selon la parité de t, $\mathbf{o}_t$ ne rencontre que des $\mathbf{a}$ ou des $\mathbf{b}$, mais il les rencontre tous.

Décomposons. Pour effectuer une opération $\mathbf{op}$ sur $\mathbf{a}$ (ou $\mathbf{b}$), l'automate envoie un signal $\mathbf{o} = (\mathbf{1}, \mathbf{op})$ synchronisé avec $\mathbf{a}_0$ (ou $\mathbf{b}_0$). Il attend ensuite de recevoir un $\mathbf{d}$ valant $\mathbf{1}$ pour savoir que l'opération a été effectuée et passer à l'opération suivante.

Ce $\mathbf{o}$ transite et rencontre successivement tous les couples $(\mathbf{a}_i, \mathbf{a}_{i+1})$ valant $(\mathbf{1}, \mathbf{1})$, jusqu'à atteindre la fin du registre qui vaut $(\mathbf{1}, \mathbf{0})$. A ce moment-là, $(\mathbf{a}_i, \mathbf{a}_{i+1})$ est modifié en fonction de l'opération et $\mathbf{d}$ est mis à $\mathbf{1}$. Si $\mathbf{op}$ vaut $\mathbf{1}$ (addition) alors il est mis à $(\mathbf{1}, \mathbf{1})$, sinon (soustraction) il est mis à $(\mathbf{0}, \mathbf{0})$. Le signal $\mathbf{d}$ transite jusqu'à l'automate lui indiquant que l'opération a bien été effectuée et qu'il peut continuer.

Le temps de réponse est proportionnel à $\mathbf{a}$ ($\mathbf{b}$).

Q. E. D.



4.4 Universalité Intrinsèque

Théorème 36 BBM *est capable de simuler n'importe quel automate à partitions réversible.*

Preuve. Soit $\mathcal{P} = (\mathcal{Q}, (h, v), n, (x_i, y_i)_{1 \leq i \leq n}, t)$ un AP-R. l'ensemble $\mathcal{Q}$ est codé en binaire dual. D'après le lemme 13, la transition locale t, est réversible. D'après le lemme 33, il est possible de construire une fonction en logique réversible qui calcule t avec $\mathcal{Q}$ codé dualement.

Pour chaque tuile de chaque partition, la transition locale t est codée par un circuit binaire, comme sur la figure 27.

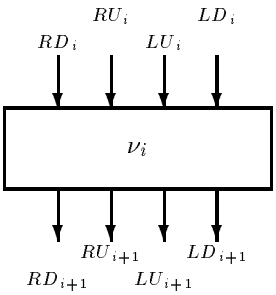


Figure 27 : Circuit pour la i^e partition.

Ses entrées viennent des quatre tuiles de la partition précédente avec lesquelles elle partage des nœuds. De même, les valeurs calculées sont dirigées vers les tuiles de la partition suivante à laquelle appartiennent les nœuds correspondants. Ce découpage est indiqué sur la figure 28.

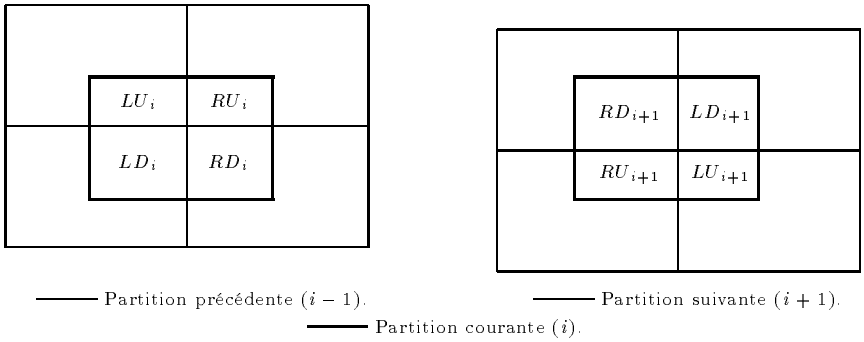
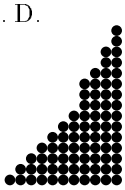


Figure 28 : Intersection des partitions.

Il ne reste plus qu'à connecter les circuits. Les AP étant itératifs, la première partition est celle qui suit la dernière. La figure 29 montre les connexions dans le cas où il y a deux partitions.

Les valeurs initiales sont placées à l'entrée des tuiles correspondantes de la première partition, dans les petits rectangles fins de la figure 29. La première transition s'effectue dans les circuits ν_1. Les valeurs intermédiaires circulent jusqu'aux circuits ν_2 de la seconde partition où s'effectue la seconde transition. Les valeurs finales reviennent ensuite à leur position d'origine. Le tour de manège suivant peut alors commencer. Un nombre quelconque de partitions peut être rajouté dans le cycle.

Q. E. D.



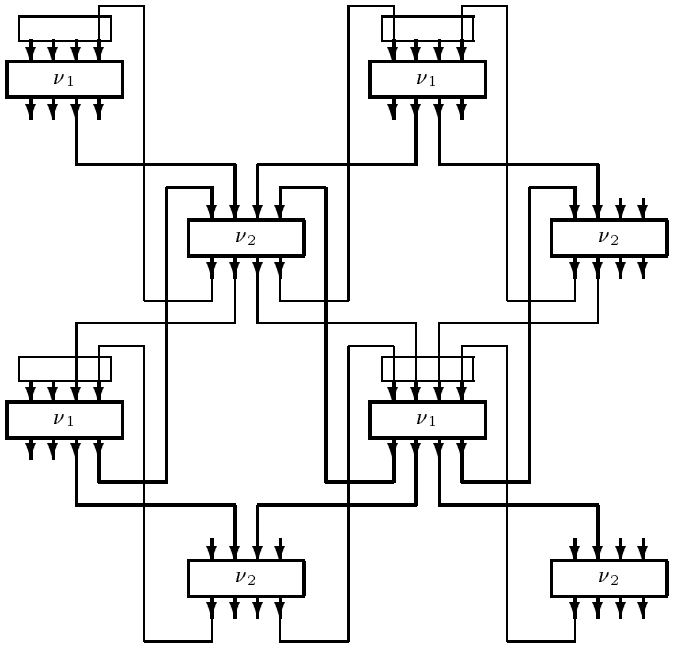


Figure 29 : Connexions pour la simulation des AP réversibles.

4.5 Simuler les Non Réversibles

Corollaire 37 *Il existe des automates à partitions (non réversibles) capables de simuler n'importe quel automate à partitions.*

Preuve. Nous modifions BBM en lui rajoutant un nouvel état : $\star$. Celui-ci sert à générer des constantes et à détruire les signaux parasitaires. Son action est définie par les règles supplémentaires de la figure 30.

Un signal arrivant en diagonale est doublé alors qu'en arrivant latéralement, il est détruit. Ces règles ne sont pas injectives, BBM* n'est pas réversible.

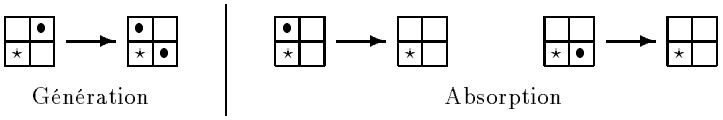
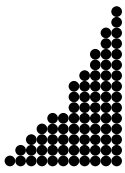


Figure 30 : Règles rajoutées pour $t_{\text{BBM}^\star}$.

La figure 31 montre comment on fait une horloge, *i.e.*, un nouveau signal généré tous les k temps. La longueur de la boucle doit être k. Elle montre aussi comment on peut détruire un signal.

Nous disposons maintenant de générateurs et de destructeurs de signaux. Les unités de la démonstration précédente peuvent simuler n'importe quelle fonction logique, réversible ou non, de manière autonome. Toute transition locale peut donc être simulée, donc tout AP par la même méthode que précédemment.



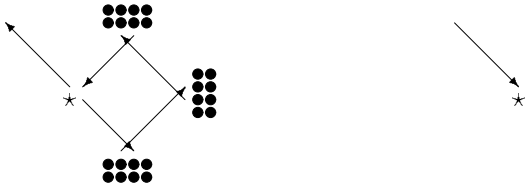


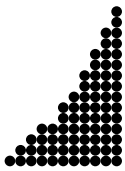
Figure 31 : Horloge et poubelle.

Q. E. D.

Tout automate cellulaire étant simulable par un automate à partitions (théorème 24), il découle naturellement :

Corollaire 38 BBM* *est capable de simuler n'importe quel automate cellulaire.*

Notre construction ne fonctionne qu'en dimension 2 et plus. Martin [93] a construit un AC de dimension 1 capable de simuler tous les AC de dimension 1. Sa construction repose sur une décomposition en couche des états et demande un nombre d'états très élevé. Elle peut être généralisée à toute dimension et permet un codage direct de la fonction locale d'un AC alors que la nôtre demande de passer par un AP.



Chapitre 5

Résultats et Conclusion

En appliquant les simulations réciproques du chapitre 3 aux résultats du chapitre 4, nous obtenons :

Théorème 39 *En dimension supérieure à 2, il existe des automates cellulaires (ou à partitions) réversibles capables de simuler tous les automates cellulaires (ou à partitions) réversibles.*

Pour atteindre ce résultat, nous avons démontré le théorème 25, à savoir la simulation de tout AC-R par un AP-R. C'est la conjecture 8.1 de l'article [138] de Toffoli et Margolus. C'est aussi la (conjecture 5.3) de l'article [77]de Kari qui démontrait la précédente conjecture en dimensions 1 et 2.

L'AP BBM du chapitre 4 est un exemple qui brille par sa minimalité : il ne nécessite que deux états et deux partitions 2×2. Avec la construction de la section 3.1, l'AC obtenu a 16 états ($\{\_, \bullet\}^4$) et un rayon de 1 (voisinage des huit plus proches voisins, dit de Moore).

En faisant de même avec BBM*, nous obtenons une version du théorème précédent pour les automates quelconques :

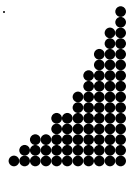
Corollaire 40 *Il existe des automates cellulaires (ou à partitions) capables de simuler tous les automates cellulaires (ou à partitions).*

L'AC associé à BBM* a 81 états ($\{\_, \bullet, \star\}^4$) et n'est pas réversible. La construction de simulation des AP du chapitre 4 peut aussi être appliquée à l'AC-R de Morita et de Ueno [106], à celui de Serizawa [119] et au jeu de la vie de Conway [22] pour montrer l'universalité intrinsèque comme au sens du calcul de BBM*. L'AP BBM* peut simuler BBM.

Corollaire 41 *Pour tout entier d non nul, il existe des* AC-R *(et des* AP-R*) de dimension $d+1$ capables de simuler tous les* AC *(et* AP*) de dimension d.*

Preuve. Toffoli a démontré que tout AC de dimension d pouvait être simulé par un AC-R de dimension $d+1$ [133]. Il y a des AC-R et de AP-R de dimension $d+1$ capables de simuler tous ceux de même dimension $(d+1)$ (théorème 39).

Q. E. D.



Tous les résultats de cette partie se généralisent facilement en dimension d plus grande que 2. Les seuls résultats valables en dimension 1 sont la décidabilité de la réversibilité des AP (lemme 13) et les différentes simulations du chapitre 3. Toutes ces simulations sont résumées sur la Figure 32.

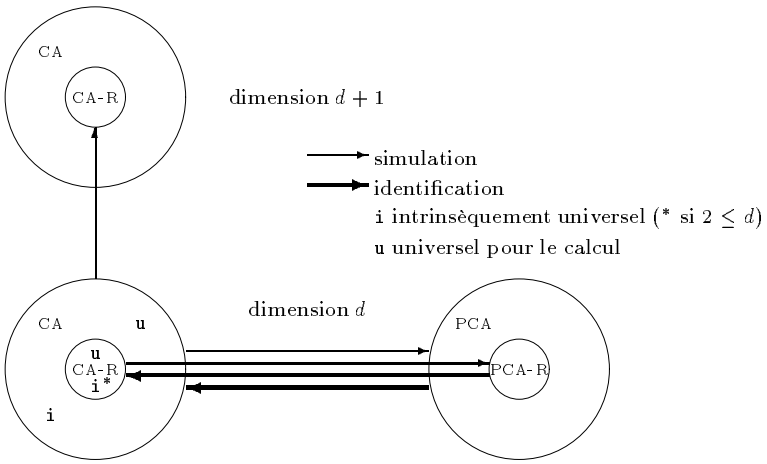


Figure 32 : Simulation et réversibilité.

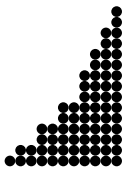
Nous pensons que les techniques mises en œuvre ici peuvent servir à montrer que tout AC-R est simulable par un automate cellulaire « partitionné » (au sens de Morita) réversible. La réversibilité est décidable dans ce modèle qui est équivalent aux AC. Malheureusement, la simulation directe des AC produit toujours un automate non-réversible.

Le problème de l'existence d'un réversible simulant tous les réversibles en dimension 1 reste ouvert. Il doit y avoir des pistes dans les travaux de Morita [103–105]. Malheureusement, il ne s'agit que de simulations valables uniquement sur des configurations finies ou pour des machines de Turing. Ce n'est pas un cadre assez large pour les AC sur des configurations quelconques.

Le problème ouvert point de départ de cette partie est celui de la simulation des automates non réversibles par des réversibles de même dimension. Cette possibilité nous gêne car il se perd partout de l'entropie, la réversibilité étant l'isentropisme. De plus, à l'inverse du cas où l'on passe en dimension supérieure, il n'y a pas d'espace illimité pour stocker les informations nécessaires au retour en arrière. Nous ne pouvons rien mettre sur les côtés comme pour la simulation de Morita avec des configurations finies [104].

Si cela était possible, il suffirait toutefois de démontrer que BBM est capable de simuler BBM* car le premier peut tout ce que peuvent les réversibles et si le second était simulable, tous le seraient par transitivité. Par contre si cela était impossible, la démonstration devrait utiliser d'autre techniques et notions que celles que nous avons utilisées, mais nous n'avons pas de piste sérieuse. Le problème reste ouvert.

Cette problématique nous fait penser aux automates classiques reconnaisseurs de langages. Selon le modèle, les automates déterministes ou non déterministes n'ont pas le même pouvoir de reconnaissance, voir les livres suivants [31, 65, 81, 99].

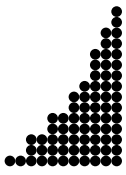


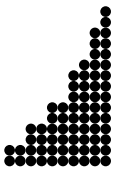
«Mompox no existe», dijo. «A veces, soñamos con ella, pero no existe».
«Por lo menos puedo dar fe de que existe la torre de Santa Bárbara» dijo José Palacio. «Desde aquí la estoy viendo».

Gabriel Garcia Marquez,
El general en su laberinto.

Partie B

Tas de Sable





Chapitre 6

Introduction aux Tas de Sable

La dynamique des piles de sable, ou « Sand pile model » (SPM), est basée sur la différence du nombre de grains entre deux sites. Si celle-ci est suffisante, un grain change de site.

Le système étudié dans cette partie est une ligne infinie de piles de sable mises à jour en parallèle. A chaque instant, une pile donne un grain de sable à chacune de ses voisines ayant au minimum deux grains de moins qu'elle. La stabilité est toujours atteinte. Ceci a déjà été étudié par Goles et Kiwi [48–52]. Ils ont décortiqué plusieurs cas séquentiels : écroulement d'une pile, grain à grain limité par une sortie, brusque changement de niveau ...

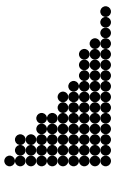
En ligne, SPM est un automate cellulaire équivalant au « Chip firing game » (CFG). Dans le CFG, un nœud donne un grain à chacun de ses voisins dès qu'il a au moins autant de grains que de voisins. La dynamique de SPM repose sur le gradient alors que celle de CFG est basée sur la valeur. C'est par là, en considérant les différences de hauteurs, que Goles et Kiwi ont montré l'équivalence de ces deux modèles.

La seconde partie s'articule comme suit :

Dans le chapitre 7 sont regroupées les définitions du modèle parallèle des tas de sable linéaires. L'équivalence avec le « Chip firing game » est rappelée. Ce sont tous deux des automates cellulaires.

Le chapitre 8 commence par l'étude d'un phénomène particulier : l'évolution d'un système au départ vide recevant un grain de sable sur sa première pile à chaque itération. Cette étude se termine au chapitre 10. Deux bandes de triangles apparaissent sur la représentation en trois dimensions de l'évolution du système. En différences de hauteurs, les configurations sont formées par la concaténation de quatre motifs : 00, 13, 02 et 11. Les frontières entre ceux-ci se comportent comme des signaux se coinçant les uns les autres. Une poignée de paramètres géométriques suffit à définir chaque configuration. Nous donnons deux algorithmes récursifs pour calculer ces paramètres.

Dans le chapitre 9, les grains sont numérotés par ordre d'apparition et suivis. Ils s'accumulent de part et d'autre d'une diagonale en fonction de leurs parités. Nous expliquons cela en regardant localement la transhumance des grains dans les deux premiers motifs et au passage des frontières. Cela permet de mieux comprendre les interactions entre les frontières et la migration des grains.

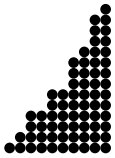


La ligne de séparation du chapitre 9 n'est qu'une observation, dans le chapitre 10 nous recherchons le comportement asymptotique du système du point de vue géométrique. Le rapport des tailles des deux moitiés de pentes 2 et 1 trouvé au chapitre 8 a pour limite $\sqrt{2}$. La racine s'explique par une égalité d'aires. Les configurations croissent homothétiquement en fonction de la racine du nombre de grains.

Le chapitre 11 est consacré à l'écroulement d'une pile de sable solitaire placée à l'extrémité de l'espace. Durant une première phase, elle donne un grain à sa voisine à chaque itération. Il se forme à son pied un tas qui suit exactement l'évolution du grain à grain des chapitres 8 à 10. Quand la pile originelle et le tas ont la même hauteur, de nouveaux signaux apparaissent et le système se stabilise. Les flux des grains se comprennent grâce à l'analyse faite au chapitre 9. En tout, nous trouvons un temps linéaire en fonction du nombre initial de grains. Ceci permet d'affirmer le parallélisme intrinsèque de cet automate, en comparaison avec les résultat de Goles et Kiwi [52] dans le cas linéaire séquentiel.

Dans le chapitre 12, nous étudions le comportement dans les cas où l'espace est borné par une sortie, un mur ou un autre grain à grain. Nous étudions aussi l'érosion d'un brusque changement de hauteur (contrefort).

Les résultats sont regroupés au chapitre 13 qui conclut la seconde partie et la thèse.



Chapitre 7

Définitions

Le système se compose d'une infinité de piles de sable alignées. Chacune d'entre elles peut accueillir un nombre non borné de grains mais il doit y avoir un nombre fini de grains dans le système. A chaque itération, si une pile a au minimum deux grains de plus qu'une pile voisine, elle lui en donne un.

7.1 Structure

Les piles sont indexées par l'ensemble des entiers, $\mathbb{N}$. Le système se représente par un mot fini d'entiers correspondants aux hauteurs des piles. Nous suivons les notations de Goles et Kiwi dans [52].

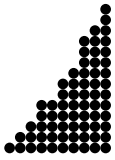
Définition 42 Une *configuration* est une suite d'entiers nuls après un certain rang. L'ensemble des configurations est noté $\mathcal{C}$ ($\mathcal{C} = \{ \nu \in \mathbb{N}^{\mathbb{N}} | \exists n_0, \forall n,\ n_0 < |n| \Rightarrow \nu_n = 0 \}$).

Les configurations son notées entre doubles crochets, par exemple : $\nu = [[\nu_0\, \nu_1\, \ldots\, \nu_k\,]]$, comme l'illustre la figure 33. Nous appelons *pente*, la différence entre deux piles voisines. Nous utilisons aussi pente pour la valeur moyenne des pentes entre deux piles éloignées.

La dynamique du « Sand pile model » (SPM) se définit comme suit : Un grain passe d'une pile à sa voisine si la pente entre les deux est au minimum de 2. Il passe au plus un seul grain entre deux piles à chaque itération, comme les grains `a`, `b`, `c` et `d` sur la figure 33.

Définition 43 Soit $\square(z)$ la fonction seuil suivante : $\forall z \in \mathbb{Z}$, $\square(z) = 1$ si et seulement si $0 \leq z$, et 0 sinon. Soient μ et ν deux configurations. La dynamique F, est définie de la façon suivante :

$$F(\mu) = \nu \iff \begin{cases} \nu_0 = \mu_0 + \square(\mu_1 - \mu_0 - 2) \\ \qquad\quad - \square(\mu_0 - \mu_1 - 2)\ , \\ 0 < i,\ \nu_i = \mu_i + \square(\mu_{i+1} - \mu_i - 2) + \square(\mu_{i-1} - \mu_i - 2) \\ \qquad\quad - \square(\mu_i - \mu_{i+1} - 2) - \square(\mu_i - \mu_{i-1} - 2)\ . \end{cases}$$



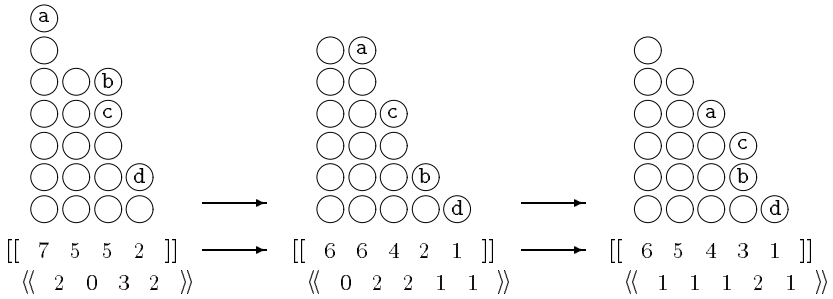


Figure 33 : Exemple d'itérations.

Les termes positifs représentent la possibilité de recevoir un grain de chaque voisine, les négatifs, au contraire, de leur en donner. Toutes les piles sont mises à jour en même temps, c'est un processus parallèle et synchrone comme sur la figure 33.

A part dans les chapitres 8 à 10 où il y a une action extérieure au système, le nombre de grains reste constant.

Les configurations peuvent être codées par la liste de leurs différences de hauteurs grâce à la fonction suivante φ :

$$\forall \mu \in \mathcal{C},\quad x = \varphi(\mu) \;=\; \langle\!\langle\ (\mu_0 - \mu_1)\ (\mu_1 - \mu_2)\ (\mu_2 - \mu_3)\ \ldots \rangle\!\rangle\ .$$

Nous utilisons $\langle\!\langle\ \ldots\ \rangle\!\rangle$ pour noter les configurations en différences de hauteurs. Avec ce code, la dynamique devient :

$$\Theta(x) = y \iff \begin{cases} y_0 = x_0 & -2\,\square(x_0 - 2) + \square(x_1 - 2) \\ \forall i,\ 0 < i, & y_i = x_i + \square(x_{i-1} - 2) - 2\,\square(x_i - 2) + \square(x_{i+1} - 2)\ . \end{cases}$$

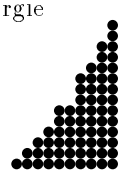
Ceci est la définition du *chip firing game* (CFG) pour un graphe ligne. Ce jeu est défini comme suit : tous les sommets d'un graphe ont des jetons et dès un sommet a plus de jetons que de voisins, il en envoie un à chacun de ses voisins.

En dimension 1, SPM et CFG sont équivalents grâce au codage φ, comme l'ont déjà démontré Goles et Kiwi dans [52]. Ce n'est pas le cas sur un graphe possédant un cycle.

Ces deux dynamiques sont invariantes par translation et ne reposent que sur une action strictement locale. Ce sont donc des automates cellulaires, leur fonction locale s'identifiant avec la dynamique. En fait, SPM n'est pas vraiment un automate cellulaire car son ensemble d'états n'est pas fini. Néanmoins, celui-ci est discret et dans tous les cas étudiés, le nombre d'états de chaque configuration est fini. Pour le CFG, seuls les états 0, 1, 2, 3 et 4 apparaissent (dans les extensions du chapitre 12, on ne va que jusqu'à -4). Qui plus est, nulle part dans cette partie les résultats, preuves, observations, ... ne font appel à des résultats sur les AC.

7.2 Convergence

Dans les cas étudiés ici, sauf pour les considérations asymptotiques des chapitres 8 à 10, il y a toujours convergence du système vers un point fixe. Pour démontrer cela nous utilisons l'énergie



potentielle E définie par :

$$\forall \mu \in \mathcal{C},\ E(\mu) \ = \ \sum_{i \in \mathbb{Z}} \mu_i^2 \ . \tag{2}$$

Cette énergie est bien définie car il y a toujours un nombre fini de grains. Chaque grain possède une énergie potentielle de l'ordre de sa hauteur.

La somme a été étendue sur $\mathbb{Z}$ pour ne pas avoir à alourdir les notations avec les bornes. Dans certains chapitres, nous nous intéressons seulement à une configuration définie sur un nombre limité de sites. Nous considérons la configuration nulle et les échanges nuls en dehors.

Théorème 44 *Un tas de sable, μ_0, se stabilise en au plus $E(\mu_0)$ coups.*

Preuve. Soient μ et ν deux configurations telles que : $F(\mu) = \nu$. Soit $\delta_{i \to j}$, l'échange de grains du sommet i vers le sommet j. Il est défini comme suit :

$$\delta_{i \to j} = \begin{cases} \square(\,\mu_j - \mu_i - 2\,) - \square(\,\mu_i - \mu_j - 2\,) & \text{si} \quad |i-j| = 1 \ , \\ 0 & \text{sinon} \ . \end{cases}$$

Remarquons que $\delta_{i \to j} = -\delta_{j \to i}$ et $\nu_i = \mu_i + \delta_{i \to i+1} + \delta_{i \to i-1}$.

Calculons la différence d'énergie potentielle :

$$\begin{aligned} E(\mu) - E(\nu) &= \sum_{i \in \mathbb{Z}} (\mu_i^2 - \nu_i^2) \\ &= \sum_{i \in \mathbb{Z}} (\mu_i + \nu_i)(-\delta_{i \to i+1} - \delta_{i \to i-1}) \\ &= \sum_{i \in \mathbb{Z}} \delta_{i \to i+1}(\mu_{i+1} - \mu_i + \nu_{i+1} - \nu_i) \ . \end{aligned}$$

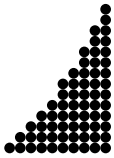
Si $\delta_{i \to i+1} = 1$, alors $\mu_i + 2 \leq \mu_{i+1}$. Comme chaque pile peut au plus gagner ou perdre 2 grains : $\nu_i \leq \mu_i + 2$ et $\mu_{i+1} \leq \nu_{i+1} + 2$. Ce qui fait $2 \leq \mu_{i+1} - \mu_i$ et $-2 \leq \nu_{i+1} - \nu_i$. Donc $0 \leq \mu_{i+1} - \mu_i + \nu_{i+1} - \nu_i$. Il n'y a égalité que si $\nu_i = \mu_i + 2 = \mu_{i+1} = \nu_{i+1} + 2$. Et dans ce cas, il faut que $\delta_{i+1 \to i+2} = -1$ pour avoir $\mu_{i+1} = \nu_{i+1} + 2$.

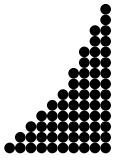
Si $\delta_{i \to i+1} = -1$, on démontre de même que $\mu_{i+1} - \mu_i + \nu_{i+1} - \nu_i \leq 0$. Il n'y a égalité que si $\nu_i + 2 = \mu_i = \mu_{i+1} + 2 = \nu_{i+1}$ et alors $\delta_{i+1 \to i+2} = 1$.

$\delta_{i \to i+1}$ ne pouvant prendre que les valeurs $\{-1, 0, 1\}$, l'énergie est décroissante. Il n'y a conservation de l'énergie que si tous les $\delta_{i \to i+1}$ sont nuls. Dans le cas contraire, s'il existe i tel que $\delta_{i \to i+1} = 1$, alors $\delta_{i+1 \to i+2} = -1$ puis $\delta_{i+2 \to i+3} = 1$. Ceci se continuerait à l'infini, mais c'est impossible car il y a un nombre fini de grains.

Donc, soit la configuration est stable, soit son énergie potentielle est strictement décroissante. Comme l'énergie est entière et positive, le théorème s'en trouve démontré.

Q. E. D.





Chapitre 8

Grain à Grain

Dans ce chapitre, nous étudions l'évolution d'un système au départ vide recevant un grain sur sa première pile à chaque itération. Nous commençons par faire quelques observations puis démontrons leur véracité et leur pérennité.

8.1 Système

L'espace est maintenant seulement infini à droite et nous travaillons à l'origine. Les grains apparaissent au début et tant que les configurations sont décroissantes, elles le restent. La configuration initiale étant vide, le système reste sur des configurations décroissantes. Tant que l'on ne considère que les suites décroissantes, les termes $\square(\mu_1 - \mu_0 - 2)$, $\square(\mu_{i+1} - \mu_i - 2)$ et $\square(\mu_i - \mu_{i-1} - 2)$ de la définition 43 sont toujours nuls. Les grains ne peuvent se déplacer que de i vers $i+1$.

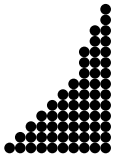
La suite des configurations ξ, est engendrée par :

$$\forall t,\ 0 \leq t,\ \left\{ \begin{array}{rcl} \xi^0 & = & [[\ 0\ ]] \\ \chi^{t+1} & = & F(\ \xi^t\ ) \\ \xi^{t+1} & = & [[\ (\chi_0^{t+1}+1)\ \chi_1^{t+1}\ \chi_2^{t+1}\ \chi_3^{t+1}\ \ \ldots\ ]]\ . \end{array} \right. \tag{3}$$

Les configurations sont codées en différences de hauteurs. La fonction F est celle de la définition 43. Les 50 premières itérations sont représentées en 3 dimensions sur la figure 34; celles de 100 à 150 sur la figure 35. Le temps augmente selon la profondeur. Les pentes semblent régulières. Les longueurs et hauteurs des configurations croissent tranquillement. Sur la surface de la figure 35 apparaissent des formes géométriques : deux bandes de triangles. Nous allons expliquer tout ceci.

8.2 Passage en Différences de Hauteurs

Les configurations représentées dans les figures 34 et 35 sont codées sur la figure 36 en différences de hauteurs. Les triangles observés à la surface se retrouvent avec les motifs suivants : ..222.., ..1313.., ..0202.. et ..111.. Ces motifs sont stables comme le montre la figure 37. Les motifs ..1313.. et ..0202.. forment des damiers, 1 et 3, tout comme 0 et 2, alternent dans le temps et l'espace.



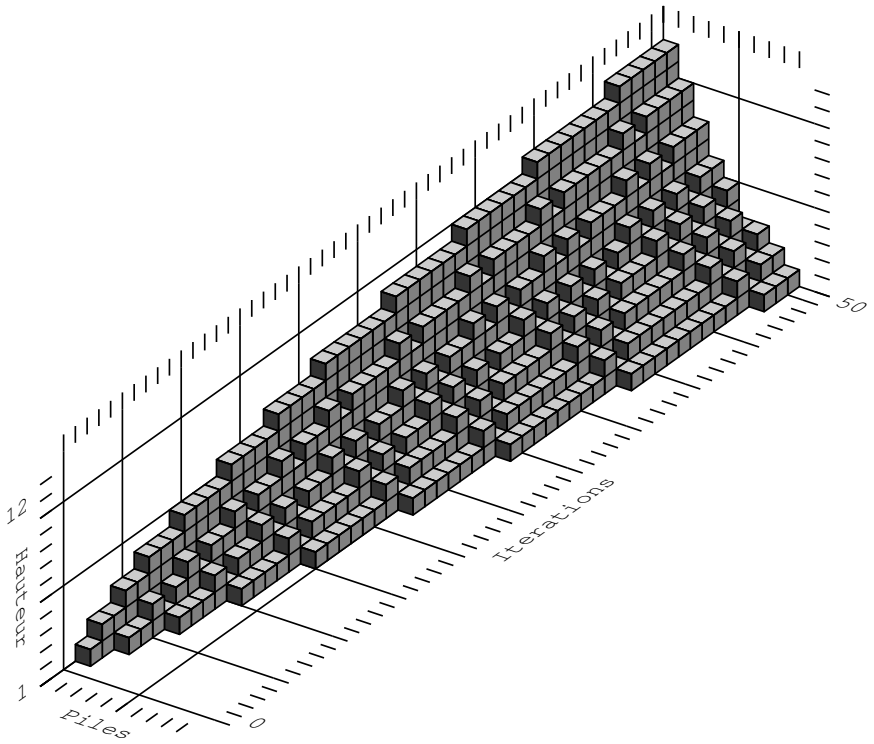


Figure 34 : Les 50 premières itérations du grain à grain.

En observant la figure 36, il est naturel d'énoncer la proposition suivante :

Proposition 45 *Toute configuration est de la forme :*

$$2^* \ (\varepsilon|3) \ (13)^* \ (\varepsilon|12) \ (02)^* \ (\varepsilon|0) \ 1^* \ . \tag{4}$$

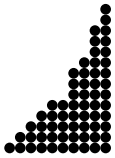
C'est la concaténation de quatre *portions* de *motifs* : ..222.., ..1313.., ..0202.. et ..111.. respectivement. Chacune de ces portions peut être vide. La jonction entre les seconde et troisième portions est soit 3|0, soit 1|2.

Nous démontrons la proposition 45 par induction. Elle est vraie pour les 150 premières itérations comme le prouve la figure 36.

Définition 46 Les *frontières* entre respectivement les deux premiers, les deux centraux et les deux derniers motifs sont notées L (pour left), M (middle) et R (right). Nous appelons *bordures*, les extrémités des configurations et *parités*, les premières valeurs des second et troisième motifs.

Ces paramètres sont déjà représentés sur la figure 36 et définis géométriquement sur la figure 41.

Les parités définissent la première valeur des motifs centraux. Elles sont nécessaires avec les longueurs pour définir exactement ceux-ci. Grâce à la proposition 45, une seule parité suffit avec la longueur des quatre motifs pour connaître toute la configuration.



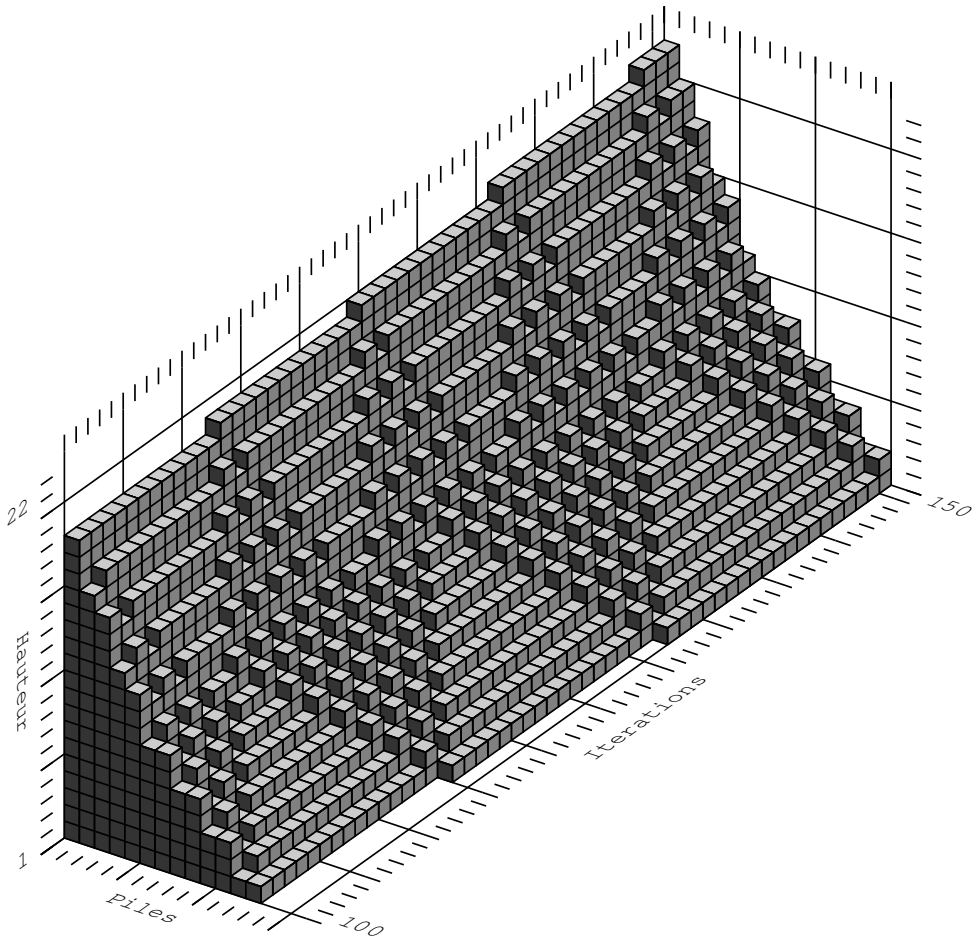


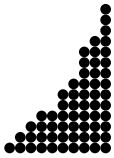
Figure 35 : Représentations géométriques des itérations 100 à 150.

CFG est un automate cellulaire de rayon 1. L'interaction est purement locale. Tout ce qui se trouve à plus d'une pile de distance n'a pas d'influence. La figure 37 montre que tant que les seconde et troisième portions sont assez larges, M vaut 1|2 ou 3|0. Le problème est de savoir ce qui se passe quand une de ces portions disparaît puis réapparaît. Cela correspond aux figures 38 à 40.

Sur la figure 36, L et R se comportent comme des *signaux* toujours en mouvement de part et d'autre de M. La figure 37 montre que les bordures ne posent pas de problèmes pour ce qui est de la cohésion des portions externes et que M n'a pas de mouvement propre. Nous appelons aussi M *séparation médiane*. Après observations des figures 37 à 40, il apparaît que seule l'action de L ou R fait bouger M.

L'étude se restreint donc à l'action des frontières L et R. Nous les étudions d'abord séparément. Nous nous occupons ensuite des cas où elles entrent en contact.

Tant que la troisième portion reste de longueur supérieure à 2, les deux premières peuvent être étudiées séparément. Si la première frontière L vaut 2|1, le comportement de la moitié gauche du



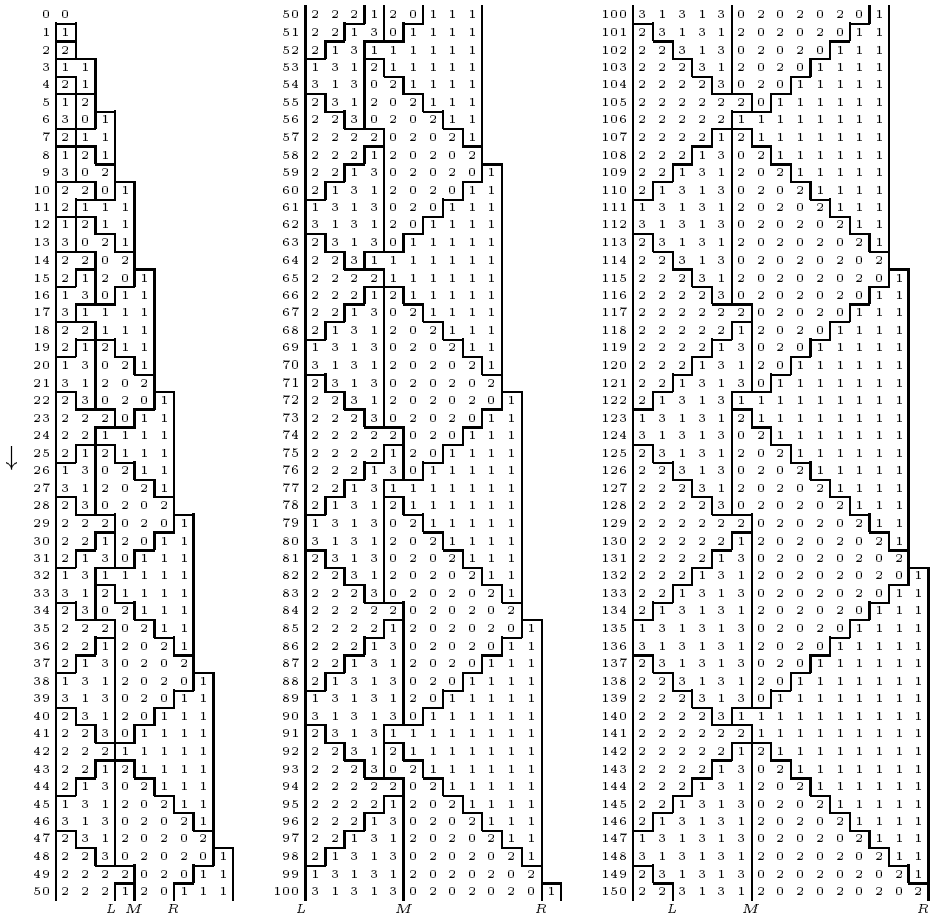


Figure 36 : Observations géométriques sur les différences de hauteurs.

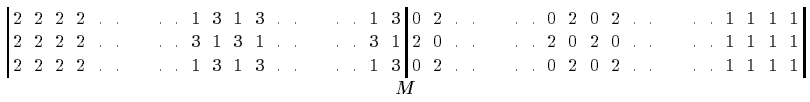
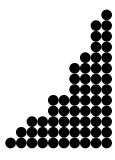


Figure 37 : Motifs et stabilité de la frontière M.

système suit celui de la figure 38. Le signal L va à gauche, rebondit sur le début de la configuration, repart à droite jusqu'à la séparation M, la déplace puis recommence ce cycle.

Plus formellement, si la frontière gauche L vaut $2|1$, elle se déplace vers la gauche, sinon elle vaut $2|3$ et se déplace vers la droite. Si elle atteint le début de la configuration, elle fait demi-tour en deux temps. Si elle atteint la séparation médiane M, elle fait aussi demi-tour en deux temps, mais elle la déplace de 1 cran sur la droite. Après ce rebond, la proposition 45 est de nouveau vérifiée.

Considérons maintenant que c'est la seconde portion qui reste de longueur supérieure à 2. Les deux derniers motifs peuvent être étudiés séparément. Si la troisième frontière R vaut $2|1$, l'évolution de la moitié droite suit celle de la figure 39. Le signal R va à droite jusqu'à atteindre l'extrémité de la configuration et étend la longueur de celle-ci de 1. Elle revient ensuite sur la



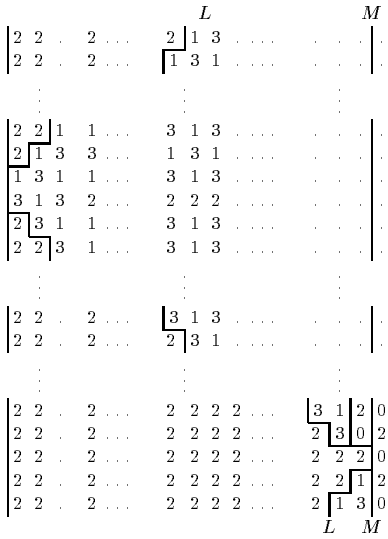


Figure 38 : Comportement de L et M dans la moitié gauche.

gauche pour rebondir sur la séparation médiane en la déplaçant. Elle recommence ensuite ce cycle.

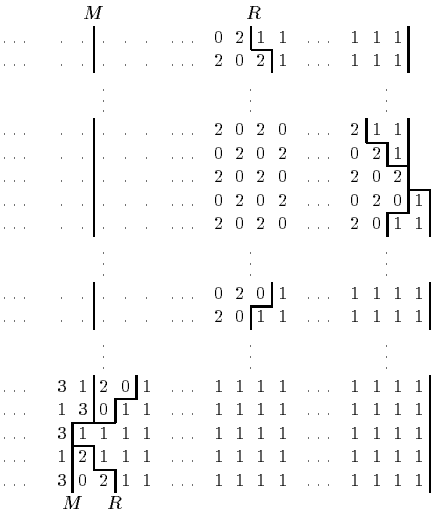


Figure 39 : Comportement de M et R dans la moitié droite.

Plus formellement, si la frontière droite R vaut $2|1$, celle-ci se déplace vers la droite, sinon, elle vaut $0|1$ et se déplace vers la gauche. Si elle atteint l'extrémité de la configuration, elle l'allonge de 1 et fait demi-tour en deux temps. Si elle atteint le frontière médiane M, elle la déplace de 1 vers la gauche et fait demi-tour en deux temps.

Les frontières L et R restent toujours d'un même côté de M. Tant que la seconde ou la troisième portion reste assez longue, L et R ne s'influencent nullement. La figure 40 montre ce qui se passe lorsque L et M arrivent pratiquement en même temps. Tant que la proposition 45 reste vérifiée, il ne peut y avoir d'autres rencontres. Si L et R sont plus distantes, l'interaction locale ayant pour rayon 1, il n'y a pas du tout de rencontre. En fait, les première et dernière rencontres de la

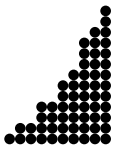


figure 40 ne sont que la succession des deux passages, sans aucune interaction.

A chaque fois, la cohérence au niveau de M est restaurée. Après les collisions, les frontières L et R reprennent leurs cycles, de part et d'autre de M.

Les damiers des seconde et troisième portions se continuent toujours au travers d'une collision de L et R, même lorsque ces portions disparaissent. Ceci n'est faux que dans le cas où les signaux L et R sont exactement synchronisés (diagramme du milieu de la figure 40).

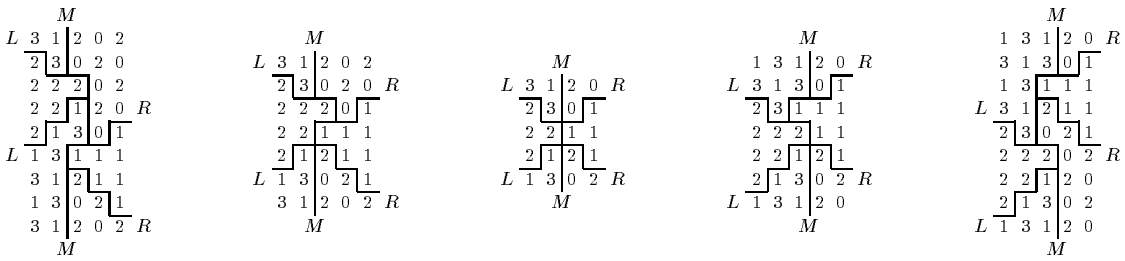


Figure 40 : Rencontres de L et R sur M.

Nous avons étudié tous les cas possibles d'interactions entre les signaux. Par induction, la proposition 45 est vraie.

Les différences de hauteurs sont bornées et ne peuvent prendre que les valeurs 0, 1, 2 et 3. Cela permet de faire une première approximation des configurations : elles se composent de deux moitiés de pentes respectives 2 et 1. Soient i et j des indices de piles non vides :

$$\nu_j \neq 0 \quad \text{et} \quad i < j \quad \Longrightarrow \quad j - i - 1 \;\leq\; \nu_i - \nu_j \;\leq\; 2\,(j-i) + 1 \;\; . \tag{5}$$

Le système tout entier peut être décrit par sa taille, les positions des différentes frontières et le début du motif de l'une des deux portions centrales. Connaître les directions de L et R est équivalent à connaître le début, ou la fin, des motifs centraux connaissant leurs longueurs.

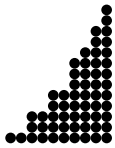
Soient respectivement H et T, la *hauteur maximum* et le *nombre de piles utilisées* (ou *taille*) de la configuration courante. La hauteur H est en fait celle de la première pile. Ces paramètres sont définis sur la figure 41.

L'équation (5) devient sur la configuration entière :

$$T - 1 \;\leq\; H \leq 2T + 1 \;\; . \tag{6}$$

8.3 Algorithmes

Dans cette section, nous donnons des algorithmes rapides pour calculer les différents paramètres en fonction du temps. Ne connaissant pas de formule directe, nous itérons un calcul récursif beaucoup plus rapide que la simulation de l'automate cellulaire. Comme indice d'itérations, nous utilisons n le nombre de grains tombés puisqu'il en tombe un à chaque itération. Les algorithmes sont basés sur le déplacement des signaux L et R.



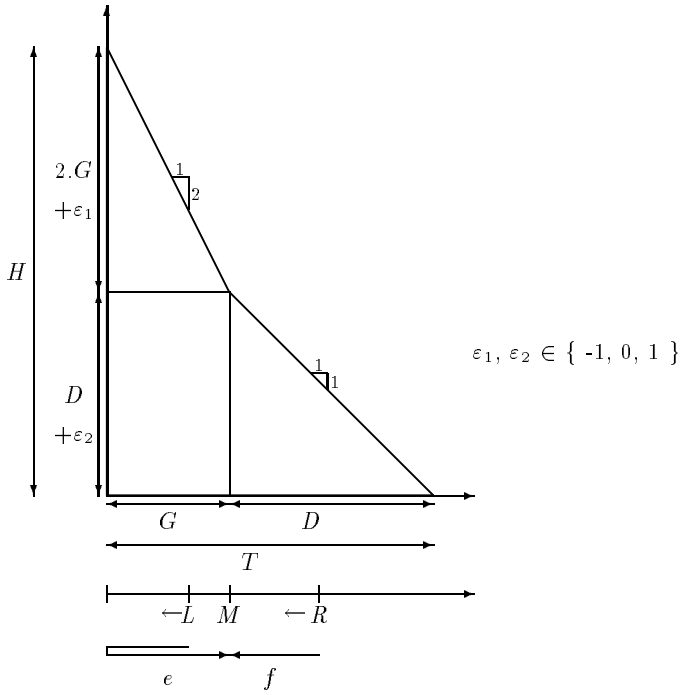


Figure 41 : Définitions géométriques de G, D, T, H, L, M, R, e et f.

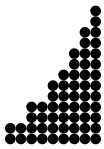
A chaque itération, les signaux L et R se déplacent de 1 cran. Leurs directions sont dl (pour L) et dr (pour R). Ils valent -1 pour la gauche et 1 pour la droite. Si un signal arrive sur une bordure, les modifications idoines sont faites, puis il repart. La figure 42 contient l'algorithme et des résultats.

Définition 47 Soient respectivement G et D, les tailles des moitiés gauche et droite. Soit e (f) le temps que L (R) mettra à revenir à la séparation médiane M.

Ces paramètres sont aussi définis géométriquement sur la figure 41. Une configuration est maintenant définie par (G, D, e, f). Grâce aux inégalités de cette figure, et les précédentes (5) et (6) :

$$\begin{aligned} T &= G + D \ , \\ H &= 2.G + \varepsilon_1 + D + \varepsilon_2 \ , \\ n &= (G + \frac{D}{2})(D + \varepsilon_2) + \frac{G(2.G + \varepsilon_1)}{2} + \varepsilon_3 \ , \\ \text{avec} \quad & |\varepsilon_1| \leq 1 \ ; \ |\varepsilon_2| \leq 1 \ \text{ et } \ |\varepsilon_3| \leq T \ . \end{aligned} \tag{7}$$

Il y a plusieurs ε car 1 et 2 ne sont que des approximations des pentes de motifs ..1313.. et ..0202..



```
L := L + dl;
R := R + dr;
si        ( L == M )
alors     dl := -1;
          si        ( M == R )
          alors     dr := 1;
          sinon     M := M + 1;
                    L := M;
sinon si  ( R == M )
          alors     M := M - 1;
                    R := M;
                    dr := 1;
si        ( -1 == L )
alors     dl := 1;
          L := 0;
si        ( R == T + 1 )
alors     R := T;
          T := T + 1;
          dr := -1;
n = n + 1;
```

n	dl	L	M	R	dr	T
110	-1	1	5	9	1	13
111	-1	0	5	10	1	13
112	1	0	5	11	1	13
113	1	1	5	12	1	13
114	1	2	5	13	1	13
115	1	3	5	13	-1	14
116	1	4	5	12	-1	14
117	-1	6	6	11	-1	14
118	-1	5	6	10	-1	14
119	-1	4	6	9	-1	14
120	-1	3	6	8	-1	14
121	-1	2	6	7	-1	14
122	-1	1	5	5	1	14
123	-1	0	5	6	1	14
124	1	0	5	7	1	14
125	1	1	5	8	1	14
126	1	2	5	9	1	14
127	1	3	5	10	1	14
128	1	4	5	11	1	14
129	-1	6	6	12	1	14
130	-1	5	6	13	1	14
131	-1	4	6	14	1	14
132	-1	3	6	14	-1	15
133	-1	2	6	13	-1	15
134	-1	1	6	12	-1	15
135	-1	0	6	11	-1	15
136	1	0	6	10	-1	15
137	1	1	6	9	-1	15
138	1	2	6	8	-1	15

L, M et R sont les positions des frontières, dl et dr leurs directions,
T la longueur totale et n est le nombre de grains tombés.

Figure 42 : Algorithme pas à pas et résultats.

Tant que ni L ni R ne rencontrent M, la dynamique est simple : L et R se déplacent à vitesse constante. Nous nous servons de cette remarque pour optimiser l'algorithme de la figure 42. L'algorithme de la figure 43 est plus rapide que le précédent car il ne considère que les instants où L ou R revient en M. Les nouvelles variables ne sont plus les sens des déplacements et les positions, mais le temps que mettront les signaux pour revenir au milieu : e et f. A ce moment là, tout est remis à jour.

```
si          ( e < f )
alors       n := n + e;
            G := G + 1;
            D := D - 1;
            f := f - e - 1;
            e := 2 * G + 1;
sinon si    ( f < e )
alors       n := n + f;
            G := G - 1;
            D := D + 2;
            e := e - f - 1;
            f := 2 * D + 1;
sinon       n := n + e;
            D := D + 1;
            e := 2 * G + 1;
            f := 2 * D + 1;
```

n	G	D	e	f
29	3	3	7	3
32	2	5	3	11
35	3	4	7	7
42	3	5	7	11
49	4	4	9	3
52	3	6	5	13
57	4	5	9	7
64	3	7	1	15
65	4	6	9	13
74	5	5	11	3
77	4	7	7	15
84	5	6	11	7
91	4	8	3	17
94	5	7	11	13
105	6	6	13	1
106	5	8	11	17
117	6	7	13	5
122	5	9	7	19
129	6	8	13	11
140	5	10	1	21

n est le nombre de grains tombés, G et D sont les longueurs des deux moitiés, et e et f sont les temps de retour au centre des signaux L et G.

Figure 43 : Algorithme à grands pas et résultats.

Chapitre 9

Etiquetage des Grains

Dans ce chapitre, nous étudions toujours le grain à grain. Les grains sont numérotés par ordre d'apparition. Ceci permet de suivre leurs migrations et d'observer une singulière répartition des grains pairs et impairs. En en expliquant localement les raisons, on comprend que le positionnement des grains se fait par couche, grâce aux signaux. Tout ceci nous aide à appréhender les mécanismes aussi bien locaux que globaux du système.

9.1 Premières Observations

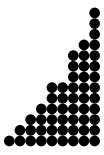
Sur la figure 44, les grains impairs sont représentés en noir. Leur positionnement est vraiment singulier.

Nous observons une ligne issue de l'origine qui sépare ces deux damiers. Pendant un temps, tous les grains pairs vont s'accumuler d'un côté de cette diagonale alors que les impairs vont tous de l'autre. De temps à autre, les pairs et les impairs échangent leurs rôles, formant ce damier. Les lignes de séparation autres que la diagonale sont de pente 2 au-dessus et 1 au-dessous. La silhouette générale des configurations se retrouve. Les silhouettes passées sont comme mémorisées, fossilisées. Dans la suite, nous montrons que cette diagonale correspond bien aux positions de la séparation médiane M.

Observons plus finement la figure 44, les pairs et les impairs forment des trapèzes de part et d'autre de la diagonale. Ils se correspondent et ont même aire. Pour chacun des trapèzes du dessus, deux sommets opposés sont alignés horizontalement, alors qu'au-dessous, c'est verticalement. Y aurait-il un 'méta-signal' se déplaçant pour indiquer quand inverser les pairs et les impairs, une forme de synchronisation ou un équilibre arithmétique? Nous ne savons pas. Qui plus est, ceci n'est qu'un ensemble d'observations sur les 10 000 premiers grains, pas une démonstration.

La répartition de tous les pairs d'un côté et des impairs de l'autre n'implique pas une inégalité de longueurs, mais d'aires (*i.e.*, de nombre de grains) au-dessus et au-dessous de la diagonale.

Nous appelons *nexus* un sommet de la diagonale appartenant à deux trapèzes. Ce sont les lieux du changement pair / impair. Soit $(\mathtt{a}, \mathtt{b})$ un nexus de la figure 44, pour les raisons géométriques



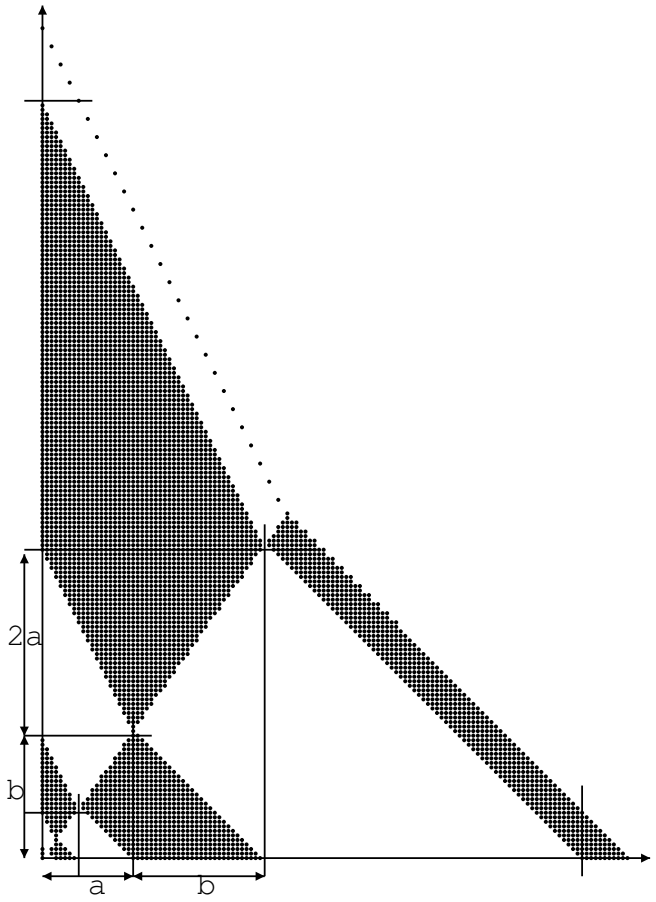


Figure 44 : Position des grains impairs (en noir).

expliquées au-dessus et la construction de la figure, le suivant devrait être proche de $(\mathtt{a}+\mathtt{b}, \mathtt{b}+2\mathtt{a})$ (pentes 1 et 2). nous pouvons donc calculer approximativement la pente de la diagonale (si c'est bien une droite) :

$$\frac{\mathtt{b}}{\mathtt{a}} = \frac{\mathtt{b}+2\mathtt{a}}{\mathtt{a}+\mathtt{b}} .$$

Ce qui amène directement à $\mathtt{b}^2 = 2\,\mathtt{a}^2$. Résultat que l'on retrouve en posant que l'aire du triangle du dessus $(\mathtt{b}(2\mathtt{a}+\mathtt{b})/2)$ est la même que celle du triangle du dessous $(\mathtt{a}(\mathtt{a}+\mathtt{b})/2)$. Si l'observation que nous venons de faire se pérennise et que la séparation est bien une droite, alors sa pente ne peut être que $\sqrt{2}$. Ceci n'est qu'une observation, mais elle donne l'intuition des résultats du chapitre suivant.

9.2 Migration des Grains

L'étrange répartition des grains selon leur parité s'explique plus ou moins naturellement en regardant ce qui se passe localement. Cette étude permet aussi de bien comprendre la migration des grains.

Dans les figures 45 à 50 les grains sont représentés en blanc ou en noir selon leur parité, sans se soucier de savoir si les pairs sont les blancs ou les noirs. Si la parité d'un grain n'est pas connue, il est marqué d'un petit cercle. Les flèches verticales indiquent le prochain grain à tomber sur la première pile. Celles horizontales indiquent les grains qui vont passer d'une pile à la suivante. Les grains déjà fixés son représentés par une silhouette en « gratte-ciel ».

La figure 45 montre l'évolution des premières piles quand le motif est ..222.. Les grains pairs et impairs viennent les uns après les autres. Ils tombent et dévalent l'escalier en conservant leur ordre. Ils sont comme une vague de billes descendant un escalier.

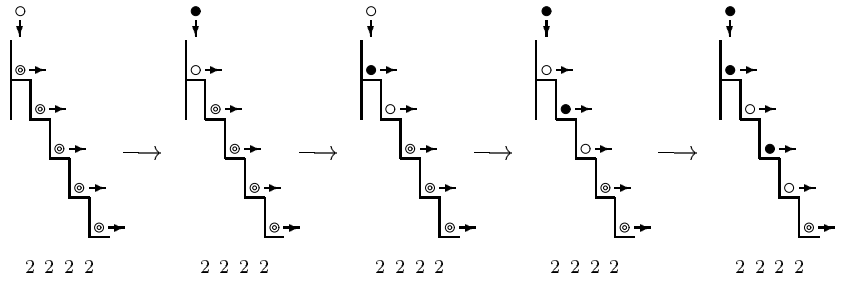


Figure 45 : Nouveaux grains arrivant sur la première pile.

La figure 46 montre le passage de la frontière L quand celle-ci va à droite (la figure 48 s'occupe du déplacement à gauche). La frontière L vaut 2|3 comme nous l'avons vu dans le chapitre précédent. La vague alternée de grains pairs-impairs s'avance alors que des grains solitaires fuient devant.

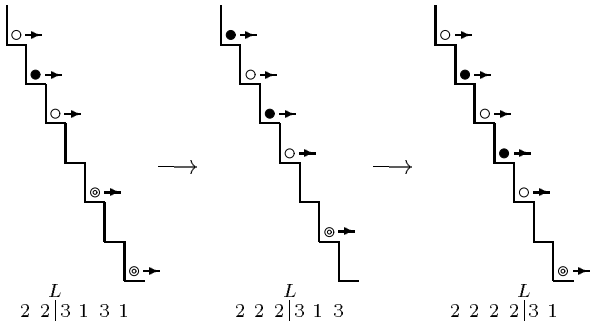


Figure 46 : Signal L se déplaçant vers la droite.

Quand la vague arrive à la séparation médiane M, si les signaux L et R ne sont pas strictement synchronisés, la dynamique suit celle de la figure 47 (sinon, c'est celle de la figure 50). Le premier grain passe, le second se fixe définitivement sur la structure. Le troisième passe au-dessus du second et le quatrième se fixe. Le troisième passe à son tour dans la moitié droite ...

Alors que le signal L repart à droite, les grains de rang impair de la vague restent bloqués alors que ceux de rang pair leur passent dessus et vont se fixer quelque part dans la partie droite.

La destination finale des grains pairs et impairs dépend de la parité du premier grain de la vague.

Quand le signal L revient à gauche, un grain sur deux se fixe et le suivant continue sa route une diagonale plus haut, comme l'illustre la figure 48. Les nouveaux grains fixés forment une nouvelle

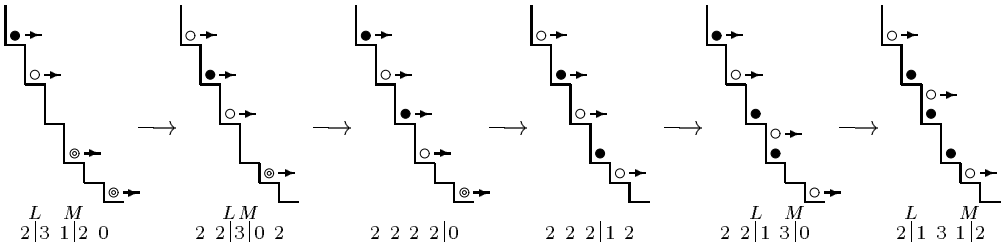


Figure 47 : Signal L atteignant le milieu seul.

couche. Ils sont tous de la même parité.

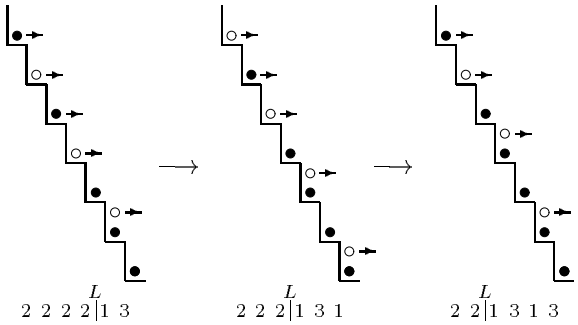


Figure 48 : Signal L revenant vers la gauche.

Quand le signal L arrive à l'extrémité de gauche, une nouvelle vague commence, comme sur la figure 49. La parité du premier grain est la même que celle de la vague précédente et des grains qui sont allés à droite. Ce cycle va se répéter et les grains seront répartis selon la même parité.

Comme ce sont seulement les grains d'une même parité qui passent à droite, le signal R n'a aucune influence sur le phénomène. Sauf dans le cas où L et R sont exactement synchronisés

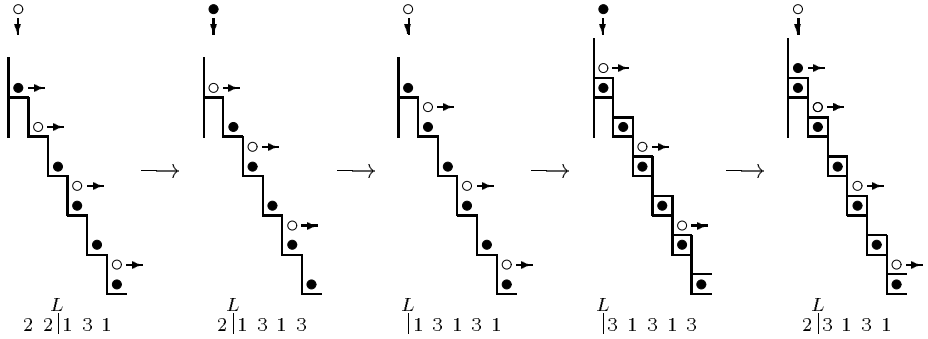


Figure 49 : Signal L atteignant l'extrémité gauche.

Synchronisation Exacte

Dans le cas où L et R atteignent le milieu en même temps, si les deux signaux ne sont pas exactement synchronisés, cela se passe comme s'ils ne l'étaient pas du tout. Par contre, s'ils le

sont, c'est le premier grain qui se retrouve bloqué et le second qui lui passe dessus, comme l'illustre la figure 50. Les destinations finales des grains pairs et impairs sont interverties.

Cela est à mettre en liaison avec le fait que les triangles de motifs ..1313.. et ..0202.. restent toujours en phase sauf dans le cas où la rencontre est exactement synchronisée, comme nous l'avions remarqué sur la figure 40 au chapitre 8. Cela correspond aussi à un cas spécial dans les algorithmes des figures 42 et 41. Ces nexus ont donc un rôle important. Il serait intéressant de pouvoir les positionner directement. Cela permettrait de calculer les configurations sans itérer et de mieux appréhender le système asymptotiquement.

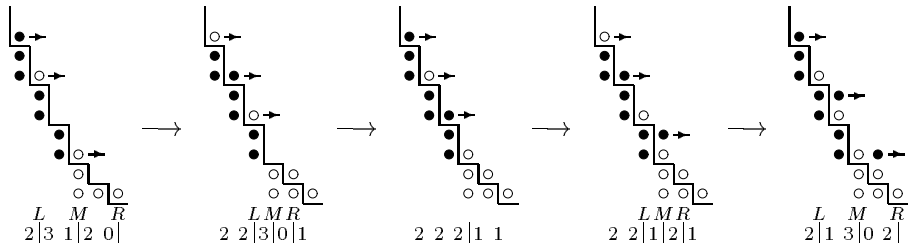


Figure 50 : Signaux L et R exactement synchronisés en M.

9.3 Partie de droite

A droite, il n'arrive que des grains de même parité. Néanmoins, nous montrons comment se passe le remplissage pour bien comprendre l'action de R. Le signal R fait des allers et retours entre la séparation médiane M et la bordure droite.

Commençons par le cas où R va à droite. Les grains arrivent espacés de 1, ce qui est normal car il n'y a plus que ceux d'une même parité. Ils dévalent tranquillement la pente sans rien devant eux comme l'illustre la figure 51.

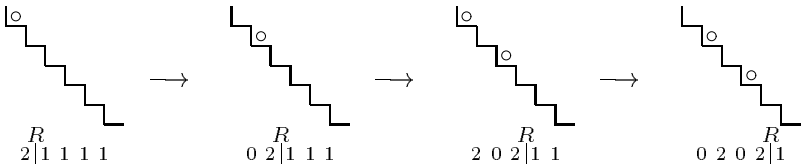


Figure 51 : Signal R allant à droite.

Arrivés en bas les premiers grains commencent à se fixer pour former une nouvelle couche sur la silhouette, comme l'illustre la figure 52.

En remontant, R fait se compresser les grains. Ceux-ci s'arrêtent définitivement les uns après les autres, comme un embouteillage qui se propage. Ceci est illustré par la figure 53.

L'embouteillage remonte jusqu'en haut de la seconde partie. Là, comme la couche est terminée, les nouveaux arrivants se retrouvent à circuler librement sur la hauteur suivante. Ceci est illustré par la figure 54.

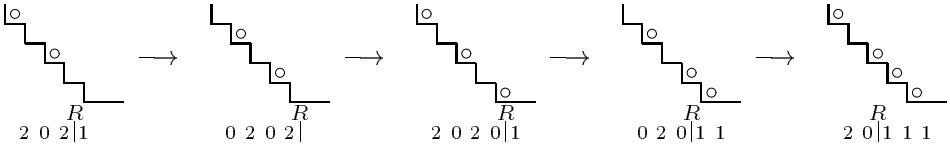


Figure 52 : Signal R atteignant l'extrémité droite.

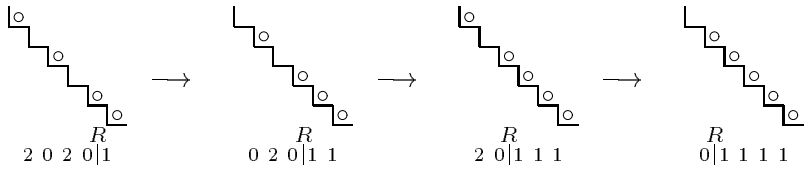


Figure 53 : Signal R allant à gauche.

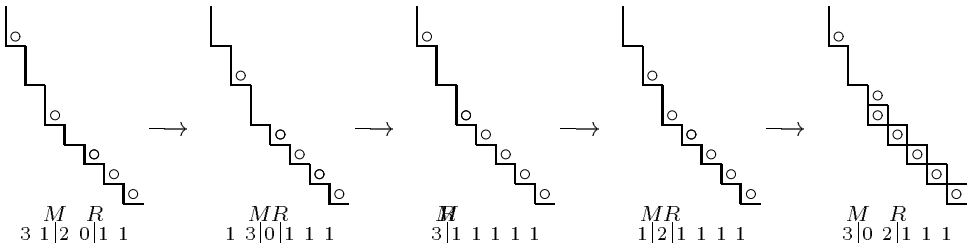


Figure 54 : Signal R rejoignant M seul.

Les deux parties ont beaucoup de similitudes. A chaque fois, il y a un grain sur deux qui se fait coincer et les signaux L et M remontent en formant une couche et redescendent avec les grains qui vont former la suivante.

Quand les signaux vont à droite, ils étalent un nouvel arrivage de grains. Quand ils reviennent à gauche, ils en fixent un grain sur deux.

Chapitre 10

Comportement asymptotique

Nous nous intéressons ici au comportement asymptotique du rapport D/G dans le cas du grain à grain. Dans un premier temps, nous faisons un encadrement simple de D/G que nous réinjectons ensuite. Relier ces deux paramètres permet d'avoir un équivalent de tous les paramètres.

10.1 Premier Encadrement

Dans tout ce chapitre, nous indexons le système par k, le k^e retour de R au milieu M. Soient T_k, D_k et G_k les valeurs de T, D et G au moment de ces retours.

Entre chaque retour, la longueur totale a été augmentée de 1. La moitié de droite est augmentée de 1 pour la collision de R avec M et est diminuée de 1 pour chaque passage de L au centre. C'est l'inverse pour la moitié gauche. La moitié droite est en plus augmentée de 1 par le passage de R à l'extrémité de la configuration.

Soit γ_k le nombre de collisions du signal de gauche L avec le milieu M entre les k^e et $k{+}1^e$ retours de R au centre. Les relations de récurrence suivantes sont vérifiées :

$$\left\{ \begin{array}{lcl} T_{k+1} & = & T_k \ + \ 1 \ , \\ D_{k+1} & = & D_k \ - \ 1 \ + \ \gamma_k \ \ , \\ G_{k+1} & = & G_k \ + \ 1 \ - \ \gamma_k \ + \ 1 \ . \end{array} \right. \tag{8}$$

Lemme 48 *Chaque fois que R retourne au milieu, soit G soit D a été augmentée de 1 et l'autre est restée inchangée. Les inégalités $D_k \leq G_k \leq 2D_k$ et $1 \leq \gamma_k \leq 2$ sont toujours vérifiées.*

Preuve. Montrons que $D_k \leq G_k \leq 2D_k$. Les valeurs calculées par les algorithmes des figures 42 et 43 ou géométriquement sur les figures 34, 35 et 36 montrent que cet encadrement est vrai dès le début.

Si $D_k < G_k < 2D_k$, alors il y a une ou deux rencontres entre L et M avant $k{+}1$: $1 \leq \gamma_k \leq 2$. Les valeurs D_k et G_k ne varient pas plus de 1. Au pire l'égalité est atteinte ($D_{k+1} \leq G_{k+1} \leq 2D_{k+1}$).

Si $D_k{=}G_k$, alors il y a une rencontre entre L et M avant $k{+}1$: $\gamma_k = 1$. Seule D augmente et $D_{k+1} < G_{k+1} < 2D_{k+1}$.

Inversement, si $G_k = 2D_k$, alors il y a deux rencontres : $\gamma_k = 2$. Seule G augmente et $D_{k+1} < G_{k+1} < 2D_{k+1}$.

Une rapide récurrence fournit l'encadrement du rapport D/G.

Q. E. D.

A partir de (7) et de ce lemme, nous obtenons :

$$2G^2 \leq n \leq 4G^2 \ .$$

Ce qui donne :

$$G = \Omega(\sqrt{n}) \ . \tag{9}$$

Donc G tend vers l'infini, tout comme D.

10.2 Recherche de la Limite

Théorème 49 *Le rapport G_k/D_k converge vers $\sqrt{2}$ quand k tend vers l'infini ($D \approx \sqrt{2}.G$).*

Preuve. Considérons l'encadrement suivant :

$$1 \leq \frac{p}{q} \leq \frac{G_k}{D_k} \leq \frac{p+1}{q} \leq 2 \ , \tag{10}$$

Avec l'hypothèse suivante sur les entiers p et q :

$$1 < q^2 \leq p^2 < D_k < G_k \quad \text{et} \quad \frac{2q^2+q}{2D_k} \leq 1 \ , \tag{11}$$

rappelons que $1 \leq G_k/D_k \leq 2$.

Le temps pour un signal d'aller et de revenir est deux fois la longueur de la moitié correspondante (plus 1 si les deux signaux n'étaient pas exactement synchronisés). Soit Δt le temps pour que R revienne q fois au centre (à partir de k). Grâce au lemme 48, la valeur minimale de G_{k+q} est G_k et la maximale G_k+q. De là :

$$q.2G_k \leq \Delta t \leq q(2(G_k+q)+1) \ .$$

De même, les valeurs minimales et maximales de D_{k+q} sont D_k et D_k+q. Soit α le nombre de passages de L au milieu M entre k et k+q. Il vérifie :

$$\frac{2q.G_k}{2(D_k+q)+1} \leq \frac{\Delta t}{2(D_k+q)+1} \leq \alpha \leq \frac{\Delta t}{2D_k}+1 \leq \frac{q(2(G_k+q)+1)}{2D_k}+1 \ .$$

Le dernier +1 vient du fait que le signal L peut être sur le point d'atteindre le milieu au début du laps de temps considéré.

Etudions ces bornes, pour l'inférieure :

$$\begin{aligned}
\frac{q.G_k}{D_k}\left(\frac{1}{1+\frac{2q+1}{2D_k}}\right) &= \frac{2q.G_k}{2(D_k+q)+1} \\
\frac{q.G_k}{D_k}\left(1-\frac{2q+1}{2D_k}\right) &\leq \\
\frac{q.G_k}{D_k}-\frac{G_k}{D_k}\frac{2q^2+q}{2D_k} &= \\
p-1 &\leq \frac{2q.G_k}{2(D_k+q)+1} \quad . \qquad (12)
\end{aligned}$$

et pour la supérieure :

$$\begin{aligned}
\frac{q(2(G_k+q)+1)}{2D_k} &= \frac{2q.G_k+2q^2+q}{2D_k} \\
&= \frac{q.G_k}{D_k}+\frac{2q^2+q}{2D_k} \\
\frac{q(2(G_k+q)+1)}{2D_k} &\leq (p+1)+1 \quad , \qquad (13)
\end{aligned}$$

En rassemblant les deux inégalités précédentes (13) et (12) :

$$p-1 \;\leq\; \alpha \;\leq\; p+3 \quad . \qquad (14)$$

Les tailles des parties, à la fin des q allers et retours de R, sont :

$$\left\{\begin{array}{lcl} D_{k+q} &=& D_k+\alpha-q \quad , \\ G_{k+q} &=& G_k-\alpha+q+q \;=\; G_k-\alpha+2q \quad , \end{array}\right.$$

Le second $+q$ vient de l'extension de la taille des configurations. L'équation (10) s'écrit aussi :

$$p.D_k \;\leq\; q.G_k \;\leq\; (p+1).D_k \quad . \qquad (15)$$

Avec les nouvelles valeurs :

$$\begin{aligned}
q(G_k-\alpha+2q) &= q.G_{k+q} \\
p.D_k-q.\alpha+2q^2 &\leq \\
p.(D_{k+q}-\alpha+q)-q.\alpha+2q^2 &= \\
p.D_{k+q}-\alpha(p+q)+p.q+2q^2 &= \\
p.D_{k+q}-(p+3)(p+q)+p.q+2q^2 &\leq \\
p.D_{k+q}-p^2-3.p-3.q+2q^2 &= \\
(p-1).D_{k+q} &\leq q.G_{k+q} \quad , \qquad (16)
\end{aligned}$$

Et pour la partie droite :

$$\begin{aligned} q.G_{k+q} &= q(G_k - \alpha + 2q) \\ &\leq (p+1).D_k - q.\alpha + 2q^2 \\ &= (p+1).(D_{k+q} - \alpha + q) - q.\alpha + 2q^2 \\ &= (p+1).D_{k+q} - \alpha(p+q+1) + p.q + q + 2q^2 \\ &\leq (p+1).D_{k+q} - (p-1)(p+q+1) + p.q + q + 2q^2 \\ &= (p+1).D_{k+q} + 2q^2 + 2q + 1 - p^2 \\ q.G_{k+q} &\leq (p+2).D_{k+q} \quad . \end{aligned} \tag{17}$$

A chaque fois, la dernière inéquation de la série est obtenue en considérant p^2 et q^2 petits devant D_k (cela correspond à l'hypothèse (11) faite sur p et q dans (10)). En rassemblant les inéquations (16) et (17) :

$$(p-1).D_{k+q} \;\leq\; q.G_{k+q} \;\leq\; (p+2).D_{k+q} \;,$$

ou

$$\frac{p-1}{q} \;\leq\; \frac{G_{k+q}}{D_{k+q}} \;\leq\; \frac{p+2}{q} \;. \tag{18}$$

Cela signifie que le rapport ne variera pas plus que de $2/q$. Regardons dans quel sens celui-ci évolue :

$$\begin{aligned} \frac{G_{k+q}}{D_{k+q}} - \frac{G_k}{D_k} &= \frac{G_k - \alpha + 2q}{D_k + \alpha - q} - \frac{G_k}{D_k} \\ &= \frac{G_k}{D_k\left(1 + \dfrac{\alpha - q}{D_k}\right)} \left(\frac{2q-\alpha}{G_k} - \frac{\alpha - q}{D_k}\right) \quad . \end{aligned} \tag{19}$$

Les valeurs de G_k et D_k sont positives, ainsi que celles de p et q (donc α), et sont plus petites que D_k, de là :

$$\begin{aligned} \operatorname{sgn}\left(\frac{G_{k+q}}{D_{k+q}} - \frac{G_k}{D_k}\right) &= \operatorname{sgn}\left(\frac{2q-\alpha}{G_k} - \frac{\alpha - q}{D_k}\right) \\ &= \operatorname{sgn}\left(2q - \alpha - \frac{G_k}{D_k}(\alpha - q)\right) \quad . \end{aligned}$$

Comme G ne peut que croître entre k et $k+q$: $0 \leq \alpha - q$.

Pour la lisibilité des équations suivantes, nous posons : $A = 2q - \alpha - \dfrac{G_k}{D_k}(\alpha - q)$, la valeur dont on cherche le signe.

$$\begin{aligned} 2q - \alpha - \frac{p+1}{q}(\alpha - q) \;\leq\; A \;&\leq\; 2q - \alpha - \frac{p}{q}(\alpha - q) \\ q(2q-\alpha) - (p+1)(\alpha - q) \;\leq\; q.A \;&\leq\; q.(2q-\alpha) - p(\alpha - q) \\ 2q^2 - \alpha(p+q+1) + p.q + q \;\leq\; q.A \;&\leq\; 2q^2 - \alpha(p+q) + p.q \\ 2q^2 - (p+3)(p+q+1) + p.q + q \;\leq\; q.A \;&\leq\; 2q^2 - (p-1)(p+q) + p.q \\ 2q^2 - p^2 - 4p - 2q - 3 \;\leq\; q.A \;&\leq\; 2q^2 - p^2 + p + q \\ 2q^2 - p^2 - 4p - 2p - 3 \;\leq\; q.A \;&\leq\; 2q^2 - p^2 + p + p \\ 2q^2 - (p+3)^2 \;\leq\; q.A \;&\leq\; 2q^2 - (p-2)^2 \quad . \end{aligned}$$

Rappelons que $2 \leq q \leq p \leq 2q$ (du lemme 48 et de (10)).
Si $\frac{p+3}{q} < \sqrt{2}$ alors $0 < 2q^2 - (p+3)^2$, $0 < \frac{1}{q} \leq A$ et G_k/D_k est croissant.
Si $\sqrt{2} < \frac{p-2}{q}$ alors $2q^2 - (p+3)^2 < 0$, $A \leq -\frac{1}{q} < 0$ et G_k/D_k est décroissant.

Finalement, le rapport ne varie pas plus que de $2/q$ et se rapproche de $\sqrt{2}$ s'il en est éloigné de plus de $4/q$. Et dans ce cas, grâce à (19) :

$$\frac{1}{q.D_k + q.(\alpha - q)} \leq \frac{1}{q.D_k.(1 + \frac{\alpha - q}{D_k})} \leq \left| \frac{G_{k+q}}{D_{k+q}} - \frac{G_k}{D_k} \right| .$$

Comme D_k croît au plus linéairement (en k), q étant fixé et α borné, la série ayant pour terme $1/q.D_k + q.(\alpha - q)$ diverge. Ceci assure que le rapport restera à moins de $4/q$ de $\sqrt{2}$. A partir de là, il ne s'en éloignera pas de plus de $6/q$.

Quand le nombre de grains n tend vers l'infini, k aussi. Il en est de même pour D_k et G_k d'après (9). Les p et q vérifiant (10) et (11) peuvent être choisis arbitrairement grands, il suffit d'aller assez loin. La valeur de $1/q$ tend vers zéro. Le rapport converge vers $\sqrt{2}$.

Q. E. D.

10.3 Conséquences

Le théorème 49, ainsi que le fait que tous les paramètres divergent vers l'infini, permet de relier asymptotiquement ceux-ci au nombre de grains tombés, n. L'aire totale de la figure 41 vaut aussi n, *i.e.*, l'aire des deux triangles et du rectangle :

$$\left\{ \begin{array}{lcccl} n & \approx & \frac{D^2}{2} + G.D + G^2 & \approx & (2+\sqrt{2})\,G^2 \quad , \\ G & \approx & \sqrt{\frac{n}{2+\sqrt{2}}} \quad , & & \\ D & \approx & \sqrt{2}\,G & \approx & \sqrt{\frac{n}{\sqrt{2}+1}} \quad , \\ H & \approx & \sqrt{(2+\sqrt{2})\,n} \quad , & & \\ T & \approx & (1+\sqrt{2}).G & \approx & \sqrt{\frac{2+\sqrt{2}}{2}n} \quad . \end{array} \right. \tag{20}$$

Notons au passage que les deux triangles de la figure 41 ont pratiquement la même taille : G^2.

Nous aurions aussi pu faire une résolution en passant par une approximation continue du système comme ont fait Anderson *et al.* pour une variation du CFG dans [8]. Ils ont aussi observé des partitions géométriques et des signaux.

Chapitre 11

Ecroulement d'une Pile

Ce chapitre est consacré à l'évolution d'une configuration où seule la première pile contient des grains. Ceci correspond à la configuration originelle $[[N]]$ où N est le nombre de grains de la première pile, ainsi que le nombre total de grains dans le système. La figure 55 illustre l'évolution à partir d'une pile de 50 grains.

Nous observons sur la figure 55, que l'écroulement se déroule en deux phases. Pendant la première phase, à chaque instant, il tombe un grain de la première à la seconde pile. Après, l'ensemble s'équilibre. Nous étudions ces deux phases séparément.

11.1 Première Phase

La première pile (numéro 0) donne un grain à chaque instant à la seconde. Les piles $1, 2, 3 \ldots$ sont au départ vides. Il tombe un grain dans la pile numéro 1 à chaque instant. Nous sommes dans le cas étudié au cours des chapitres 8 à 10. Nous utilisons t, le numéro d'itération comme indice de temps.

Il y a trois signaux qui se déplacent : L, M et R. La taille de la configuration ainsi que la hauteur de la pile numéro 1 croissent en $O(\sqrt{t})$. La hauteur de la pile numéro 0 décroît linéairement en t, elle vaut $N - t$.

Soit t_1 le moment où se finit la première phase et où commence la seconde. La première phase se termine quand les piles 0 et 1 ont, à deux grains près, la même hauteur :

$$N - t_1 \;\approx\; \sqrt{(2+\sqrt{2})\,t_1} \quad ,$$

qui se résout en :

$$t_1 \;=\; N - \sqrt{(2+\sqrt{2})\,N} + o(\sqrt{N}) \quad . \tag{21}$$

11.2 Seconde Phase

Regardons maintenant ce qui se passe après que la pile 0 arrive au niveau de la pile 1.

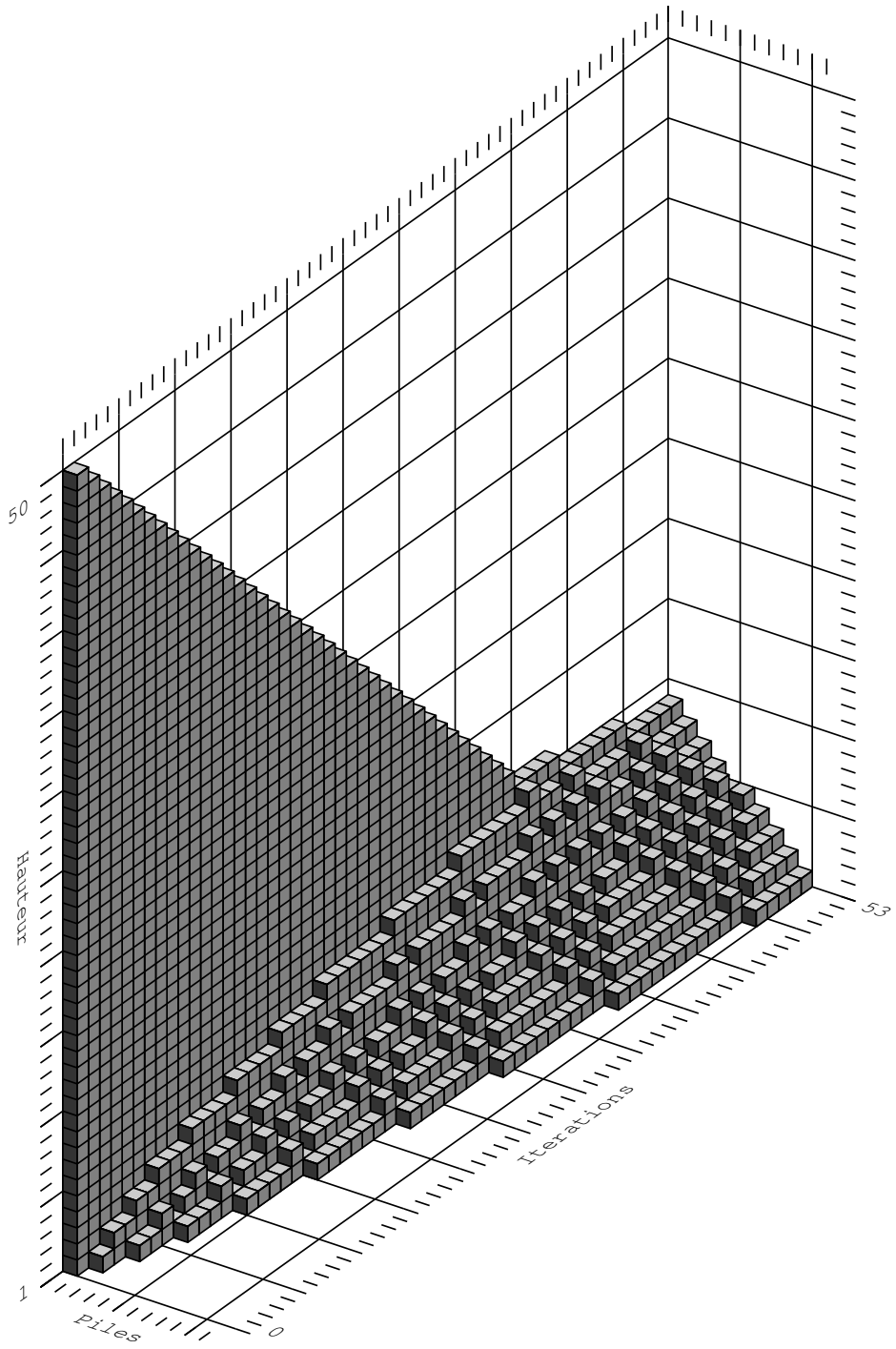


Figure 55 : Ecroulement d'une pile de 50 grains.

La figure 56 décrit le début de la seconde phase en différences de hauteurs. Dans la première colonne, L est éloigné du bord gauche. Dans les suivantes L est en train de faire demi-tour.

Nous verrons plus loin que la présence de L au début ne fait que sauter une étape de la stabilisation.

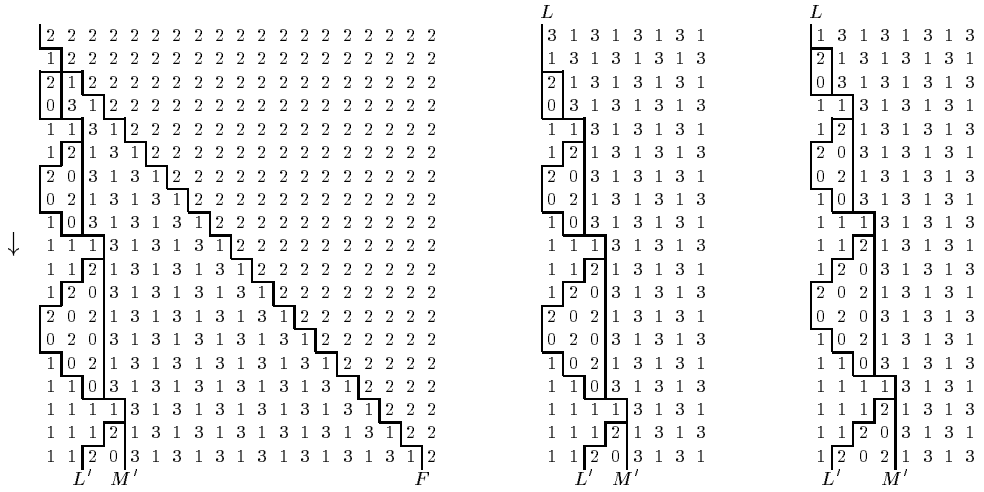


Figure 56 : Début de la seconde phase de l'écroulement.

Trois nouveaux signaux sont apparus, de gauche à droite : L' qui fait des allers et retours, M' une nouvelle séparation poussée par L' et F (pour fin) qui part sur la droite. Ils ont des comportements similaires à ceux étudiés dans le chapitre précédent. Ce qui est tout à fait normal puisqu'ils séparent les mêmes motifs que les signaux précédents.

Le signal F n'apparaît que dans le cas où L n'est pas présent au début de la configuration. Il continue à droite jusqu'à rencontrer L comme l'illustre la figure 57. A leur rencontre, ils fusionnent pour former une barrière statique B. Celle-ci est large d'une ou deux piles selon la parité de la distance entre F et L.

Figure 57 : Rencontre de F et de L et génération de B.

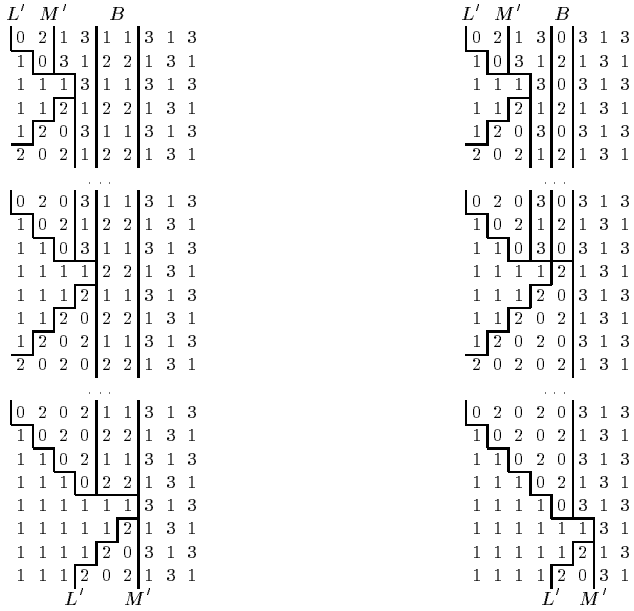
La barrière B est encadrée par M' sur la gauche et M sur la droite. Ces dernières sont poussées par L' et R respectivement. La figure 58 montre ce qui se passe si M', poussée par L' est la première à rencontrer B.

L'automate cellulaire CFG commute avec la symétrie de coordonnée $(x \to -x)$. Il se passe donc symétriquement la même chose quand M, poussée par R, rencontre B la première. Les motifs sont les mêmes.

Dans les deux cas, B est absorbée par la frontière, M' ou M, qui l'a rencontrée la première. Ces dernières continuent ensuite leur routes toujours poussées par L' et R.

Comme le montrent les deux dernières colonnes de la figure 56, si L était présent à l'origine au moment où la seconde phase a commencé, F n'apparaît pas et L est directement absorbé. La bordure B n'est donc pas créée. Le système se retrouve directement dans la situation où B est déjà absorbée.

Le cas où M et M' rencontrent simultanément B est étudié plus loin (figure 60).



Il se passe symétriquement la même chose avec M et L.

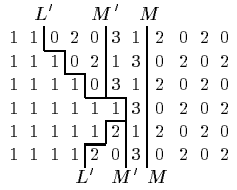
Figure 58 : Frontière M' atteignant la bordure statique B.

La figure 59 montre ce qui se passe après que B disparaît : les frontières M' et M se rencontrent enfin. Elles s'annulent et il ne reste plus que les signaux L' et R. Par symétrie du CFG, seul le cas où M' arrive sur M est représenté sur la figure 59 (cas (a) et (c)). L'autre cas, l'arrivée de M sur M' est symétrique.

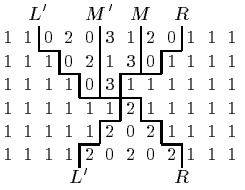
Il y a encore deux cas particuliers : quand M' et M se rencontrent pratiquement simultanément. Si L' et R ne sont pas exactement synchronisés (cas (b) de la figure 59), les signaux L' et R continuent leurs courses comme dans le premier cas. S'ils le sont (cas (d) de la figure 59), les signaux L' et R sont aussitôt absorbés. La stabilité est atteinte, c'est un des deux seuls cas où la configuration finale ne contient que des 1.

Etudions le cas particulier où la dynamique de la figure 58 n'est pas suivie. C'est-à-dire que ni M ni M' n'arrivent seules sur la bordure statique B. La figure 60 montre ce qui se passe alors.

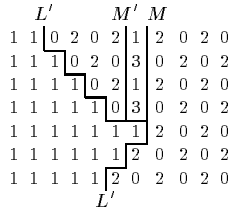
Après le second cas de la figure 60 (M' et M sont poussées simultanément vers B qui est large de 2), il ne reste plus que L' et R en course. La bordure B est alors détruite par l'un d'eux, comme



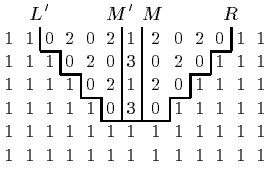
(a) Réduction du motif ..1313...



(b) Rencontre quasi simultanée des deux frontières.

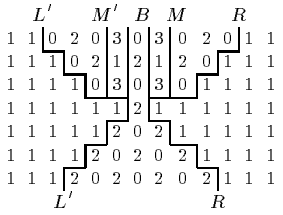


(c) Une frontière efface le motif ..1313...



(d) Les deux frontières effacent le motif ..1313...

Figure 59 : Rencontre des deux frontières M' et M.



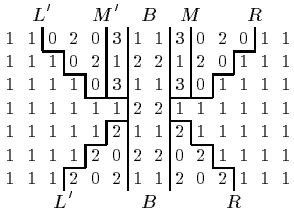
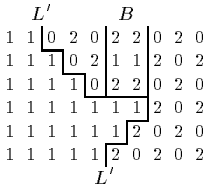


Figure 60 : Rencontre simultanée de L' et R avec B.

sur la figure 60. Il est à signaler que si L' et R arrivent simultanément sur B, il ne reste que des 1. C'est le second et unique autre cas où il ne reste pas de 0 à la fin.



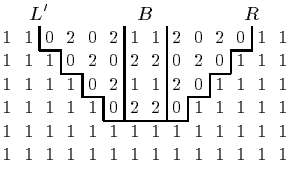


Figure 61 : Destruction finale de B par L' et/ou R.

Quand il ne reste plus que L' et R en piste, ils se rencontrent et se détruisent en créant un unique 0 au milieu des 1 comme l'illustre la figure 62. Le système est stabilisé.

Grâce à l'approche du chapitre 9, les signaux, les frontières et les motifs peuvent s'interpréter en terme de flux de grains. Le signal L' sort les grains de sa zone, la partie haute, couche diagonale après couche diagonale, rabotant le haut du système. Le signal R met les grains en place à l'autre

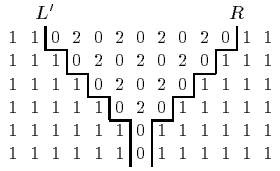


Figure 62 : L' et R se rencontrent finalement.

bout du système, diagonale par diagonale. La partie médiane de motif (..1313..) les convoie entre temps. La figure 63 résume tout ceci. Les flèches indiquent le mouvement des grains.

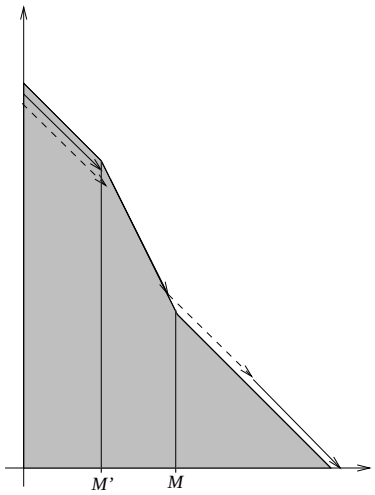


Figure 63 : Mouvement des grains au cours de la seconde phase.

La façon dont les grains partent sous les balayages des signaux fait penser à un pull en laine que l'on défait en tirant un fil.

Temps Asymptotique

Résumons ce qui se passe au cours de la seconde phase. Rapidement F et L se fondent en une nouvelle bordure B. Les signaux L' et R poussent progressivement M' et M respectivement, jusqu'à se qu'elles absorbent B et se fondent, puis elles disparaissent. Tout cela est synthétisé sur la figure 64.

La bordure statique B n'est pas très importante pour la dynamique, elle aide juste une des frontières M' ou M à avancer plus vite. De même, les cas particuliers étudiés précédemment (non-apparition en B, rencontres synchronisées) ne retardent pas la stabilisation.

Le signal M est poussé à gauche par R. Le signal M' est poussé à droite par R. Nous faisons une approximation en négligeant le fait que M va en direction de M'. Le signal M' va plus vite puisque L' a une plus courte distance à parcourir que L.

Soit d_0 la distance que M' doit parcourir pour atteindre M. Elle est inférieure à M_0, la position originelle de M plus 1 (L pourrait la déplacer avant de disparaître). Le signal M' se déplace de 1 à

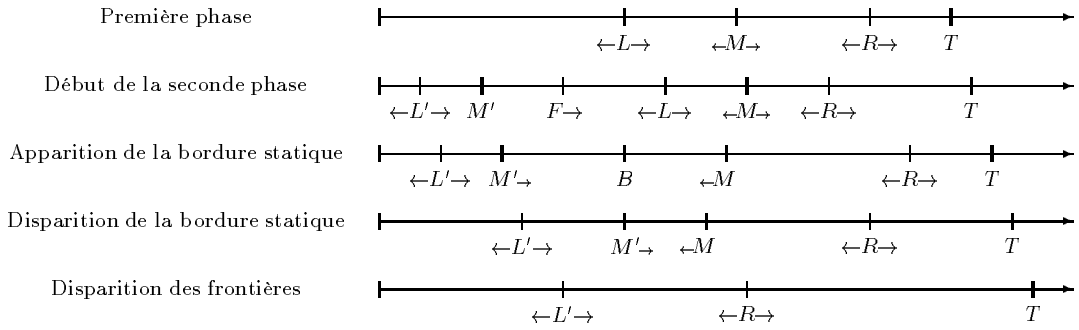


Figure 64 : Les différentes étapes de l'écroulement.

droite chaque fois que L' revient. Pour atteindre M_0, il lui faut $\sum_{i=0}^{M_0} 2i = M_0(M_0+1)$ itérations. D'après les équations asymptotiques (20) du chapitre 10,

$$M_0 \;\approx\; \sqrt{\frac{N}{2+\sqrt{2}}} \;\; .$$

Il faut donc au plus $N/(2+\sqrt{2}) + o(N)$ itérations. Les signaux L' et R mettront au plus $2\sqrt{N}$ itérations pour se rejoindre. En sommant cela avec le temps de la première phase donnée par (21), nous trouvons finalement :

Théorème 50 *Le temps d'écroulement d'une pile de sable au début d'un espace linéaire $T_{par}(N)$ est linéaire. De plus :*

$$N + o(N) \;<\; T_{par}(N) \;<\; \left(1 + \frac{1}{2+\sqrt{2}}\right) N + o(N) \;\; .$$

Le temps parallèle calculé par Goles et Kiwi [52, Lemma 3.2] est :

$$T_{\mathrm{sec}}(N) \;=\; \left(\begin{array}{c} k+1 \\ 3 \end{array}\right) + kk' - \left(\begin{array}{c} k' \\ 2 \end{array}\right) \;\approx\; \frac{N^{\frac{3}{2}}}{3\sqrt{2}} \;\; ,$$

avec $N = k(k+1)/2 + k'$ et $0 \leq k' \leq k$. Le gain de la parallélisation est de l'ordre de $\sqrt{N}$, de l'ordre du nombre de piles utilisées. Ceci prouve la nature parallèle du phénomène.

11.3 Pile au Milieu de l'Espace

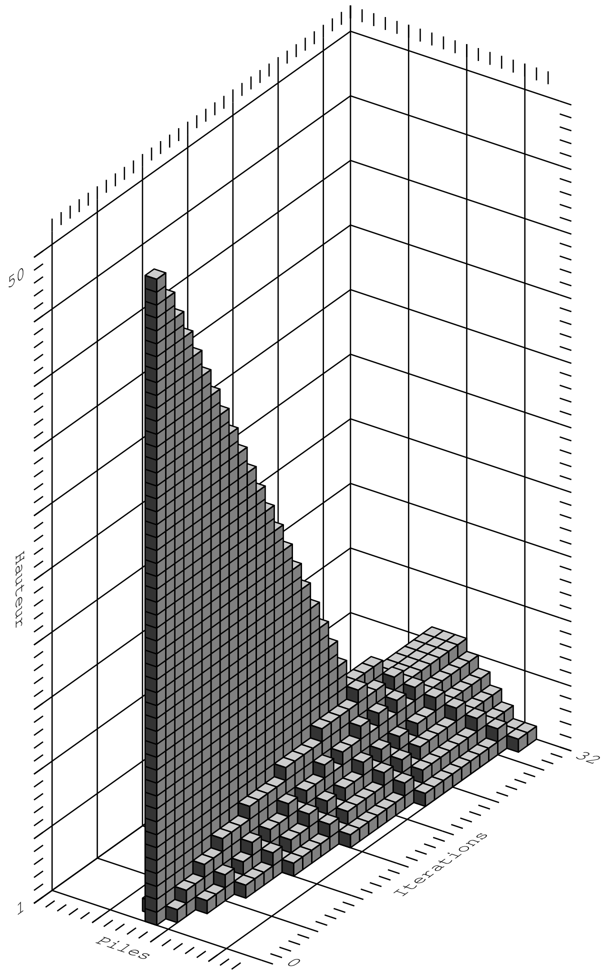
Si la pile est toute seule au milieu de l'espace, pendant la première phase, à chaque itération, elle donne deux piles, une de chaque côté. Et de chaque côté, croissent symétriquement deux tas. Cela est illustré par les figures 65.

La relation (21) devient :

$$N - 2\,t_1 \;\approx\; \sqrt{(2+\sqrt{2})\,t_1} \;\; ,$$

Il en découle :

$$t_1 \;=\; \frac{N - \sqrt{(2+\sqrt{2})\,N}}{2} + o(\sqrt{N}) \;\; .$$



0								-50	50							
1							-1	-47	47	1						
2						0	-2	-44	44	2	0					
3						-1	-1	-42	42	1	1					
4						-1	-2	-39	39	2	1					
5						-2	-1	-37	37	1	2					
6					-1	0	-3	-34	34	3	0	1				
7					-1	-1	-2	-32	32	2	1	1				
8					-1	-2	-1	-30	30	1	2	1				
9					-2	0	-3	-27	27	3	0	2				
10				-1	0	-2	-2	-25	25	2	2	0	1			
11				-1	-1	-1	-2	-23	23	2	1	1	1			
12				-1	-1	-2	-1	-21	21	1	2	1	1			
13				-1	-2	0	-3	-18	18	3	0	2	1			
14				-2	0	-2	-2	-16	16	2	2	0	2			
15			-1	0	-2	-1	-2	-14	14	2	1	2	0	1		
16			-1	-1	0	-3	-1	-12	12	1	3	0	1	1		
17			-1	-1	-1	-1	-3	-9	9	3	1	1	1	1		
18			-1	-1	-1	-2	-2	-7	7	2	2	1	1	1		
19			-1	-1	-2	-1	-2	-5	5	2	1	2	1	1		
20			-1	-2	0	-3	-1	-3	3	1	3	0	2	1		
21			-2	0	-2	-1	-3	0	0	3	1	2	0	2		
22		-1	0	-2	0	-3	-1	-1	1	1	3	0	2	0	1	
23		-1	-1	0	-2	-1	-2	-1	1	2	1	2	0	1	1	
24		-1	-1	-1	0	-3	0	-2	2	0	3	0	1	1	1	
25		-1	-1	-1	-1	-1	-2	1	-1	2	1	1	1	1	1	
26		-1	-1	-1	-1	-2	0	0	0	0	2	1	1	1	1	
27		-1	-1	-1	-2	0	-1	0	0	1	0	2	1	1	1	
28		-1	-1	-2	0	-1	-1	0	0	1	1	0	2	1	1	
29		-1	-2	0	-1	-1	-1	0	0	1	1	1	0	2	1	
30		-2	0	-1	-1	-1	-1	0	0	1	1	1	1	0	2	
31	-1	0	-1	-1	-1	-1	-1	0	0	1	1	1	1	1	0	1
32	-1	0	-1	-1	-1	-1	-1	0	0	1	1	1	1	1	0	1

Figure 65 : Ecroulement d'une pile isolée de 50 grains en 3−D et différences de hauteurs.

La seconde phase est alors légèrement différente à cause du fait que la pile originelle ne peut fournir autant à chaque fois. Cela se traduit par des sauts de crans pour les deux signaux symétriques L'. Mais en tout, le temps de stabilisation reste linéaire.

Chapitre 12

Applications à d'Autres Cas

Dans les trois premières sections de ce chapitre, nous considérons que le tas ne peut plus s'étendre infiniment : l'espace est limité. Dans un premier temps, nous considérons qu'il y a une sortie absorbant l'excès, dans un second temps qu'il y a un mur de hauteur infinie qui arrête sa progression, et finalement qu'un autre grain à grain est placé à l'autre extrémité.

La dernière section de ce chapitre est consacrée au contrefort : un brusque changement de niveau.

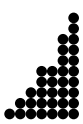
Nous donnons rapidement les résultats, les techniques sont les mêmes que précédemment. Seules les vues en différences de hauteurs sont données quand les représentations en trois dimensions ne sont que peu explicites. Les démonstrations ne sont qu'esquissées pour éviter que cette thèse ne soit un suite de cas particuliers. Ce qui suit n'est qu'une extension des résultats et techniques déjà mis en œuvre, mais complète des travaux déjà existants..

Ils y a des différences de hauteurs négatives car les configurations ne sont plus monotones. Dès que l'espace est fini, sachant que les différences de hauteurs sont bornées, il y a forcément convergence vers un point fixe ou un cycle.

Nous appelons *temps de latence* la durée du régime transitoire et notons l la largeur de l'espace.

12.1 Limité par une Sortie

A une distance l du goutte à goutte, on met en place une sortie : un site qui absorbe tous les grains qui tombent dedans et reste de hauteur nulle. Goles et Kiwi ont étudié ce cas, mais en séquentiel et en laissant tout le temps au système de complètement se stabiliser entre chaque chute de grains [49]. La figure 66 montre plusieurs exemples de cette dynamique. La figure 67 donne l'évolution des silhouettes des configurations.



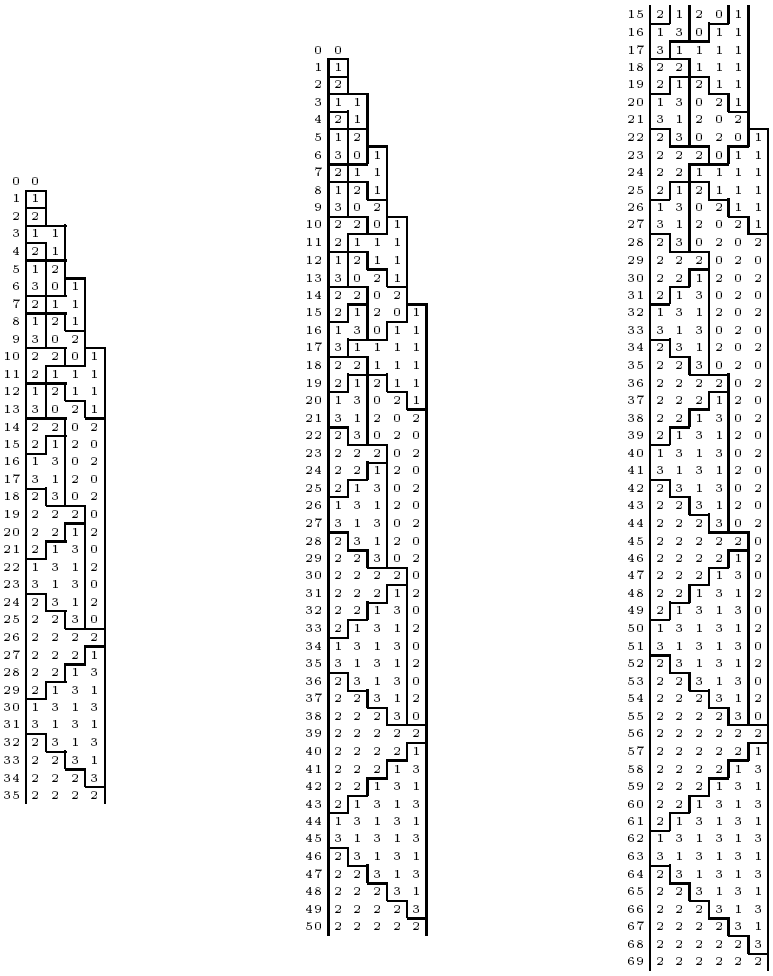


Figure 66 : Grain à grain limité par une sortie, largeurs de 5, 6 et 7.

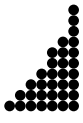
La configuration s'étend. Quand R atteint la sortie, il y 'tombe' et disparaît. Ensuite, il ne reste plus que L qui pousse M vers la sortie avant de s'y jeter lui-même.

Le fait que 22...2 apparaissent avant la fin est trompeur car la représentation est en différences de hauteurs. Le dernier aller et retour correspond à faire que la valeur de la dernière pile soit 2 et non pas 1.

Le système se stabilise, la configuration finale est de la forme $\langle\!\langle\ 2\ 2\ \ldots\ 2\ \rangle\!\rangle$ en CFG, soit $[\![\ 2l\ \ldots\ 6\ 4\ 2\ ]\!]$ pour SPM. Le tas de sable est en équilibre : à chaque itération, il entre et il sort un grain du système. Les nouveaux grains progressent d'une case vers la sortie à chaque itération. Le temps de présence d'un nouveau grain dans le système, est donc égal à la largeur de la configuration. A chaque itération, le grain en haut de chaque pile tombe et il en tombe un pour le remplacer. Tous les autres grains sont stables.

La configuration n'est stable que parce qu'elle est alimentée en grains à la plus haute énergie potentielle. Dès que cela finit, tout s'en va avec l'apparition d'une partie de pente 1 en haut.

Dans le cas étudié par Goles et Kiwi, le système converge vers $\langle\!\langle\ 1\ 1\ \ldots\ 1\ \rangle\!\rangle$, soit $[\![\ l\ \ldots\ 3\ 2\ 1\ ]\!]$



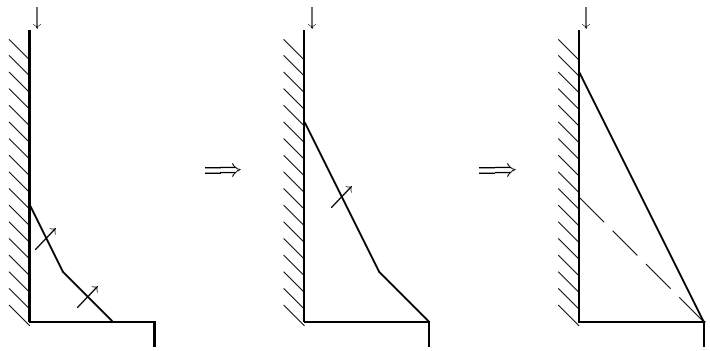


Figure 67 : Silhouette de l'évolution du grain à grain limité par une sortie.

qui est indiqué avec des tirets sur la figure 67. En régime stationnaire, à chaque itération, le nouveau grain tombe en l relaxations dans la sortie. Il en est de même pour le nôtre, à ceci près qu'un grain sort en l itérations et non relaxations.

Le système est de nouveau dans une forme qui lui permet d'évacuer un grain à chaque fois.

Temps de Stabilisation

Nous notons t_l le nombre d'itérations du régime transitoire ou temps de latence, et t_c le temps de convergence.

Le phénomène se déroule en deux temps : tout d'abord extension de la configuration comme si l'espace était infini. Le signal R disparaît. La configuration s'étend ensuite vers le haut.

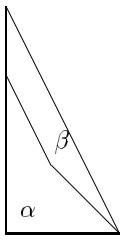


Figure 68 : Différentes aires.

La durée de la première phase est facile à trouver : il tombe un grain à chaque itération et aucun grain n'est encore sorti. En découpant les configurations par la diagonale qui a pour pente $\sqrt{2}$ (théorème 49), on trouve :

$$t_1 \;=\; \left(\sqrt{2}-1\right)\, l^2 \;+o(l^2)\;\;.$$

Pour la seconde phase, on sait, grâce à l'étude du chapitre 9, qu'au-dessus de la courbe ce sont seulement les grains d'une même parité. Et tous ceux de l'autre parité passent 'à la trappe'. Le temps de cette phase est donc le double de l'aire β de la figure 68, qui se calcule par différence entre la précédente et le triangle final.

$$t_2 \;=\; \left(4-2\sqrt{2}\right)\, l^2 \;+o(l^2)\;\;.$$

Les deux temps sont proportionnels. La seconde phase est la plus longue. Le temps total est donc :

$$t_c \;=\; \left(3-\sqrt{2}\right)\, l^2 \;+o(l^2)\;\;.$$

Signalons que Goles et Kiwi trouvent un temps de convergence de $(l-1)l/2$ pour leur cas particulier. Nous restons dans le domaine du quadratique.

12.2 Limité par un Mur

Quand l'espace est limité par un mur, le tas commence par s'étaler jusqu'à l'atteindre. Après, M se stabilise au milieu et le CFG devient périodique. Tout cela est illustré par les figures 70 et 71.

Sur la figure 71 on peut lire des temps de latence et des périodes. Ils sont regroupés sur la figure 69. Les autres valeurs ont été obtenues grâce au logiciel de simulation décrit dans l'annexe A.

Largeur	5	6	7	8	9	10	15	40	70	100
Temps de latence	17	17	40	41	72	64	237	1 649	5 934	12 574
Période	5	6	7	8	9	10	15	40	70	100

Figure 69 : Résultats numériques pour une limitation par un mur.

Le temps de latence correspond au temps nécessaire pour que la configuration prenne toute la place, que M se stabilise au milieu et que les signaux L et R se rencontrent et se synchronisent.

La période est égale à la longueur. Toute la configuration monte progressivement, une période correspond au temps de mettre un grain sur chaque pile.

Pendant une période, les signaux L et R font chacun un aller et retour sur leur moitié de configuration respectives. D'après l'étude du chapitre 9, ils ont ensemble rajouté un grain sur chaque pile. De plus, si l est impaire, il y a synchronisation exacte de L et R à chaque fois, donc tantôt une couche de grains de numéro pair, tantôt une couche d'impair. Par contre, si l est paire, ce sont les grains pairs d'un coté et les impairs de l'autre.

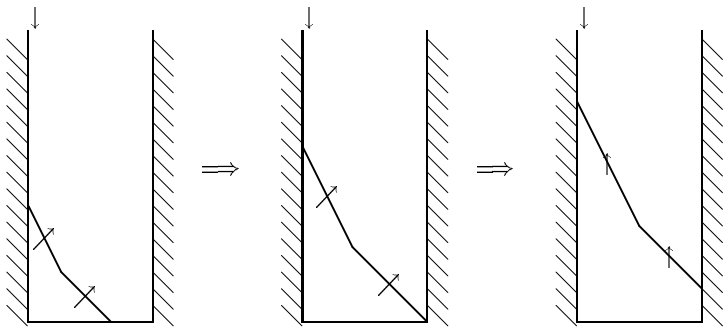
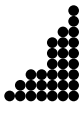


Figure 70 : Silhouette de l'évolution du grain à grain limité par un mur.

Rappelons que l est la largeur de l'espace borné. Des inégalités d'aires se déduisent :



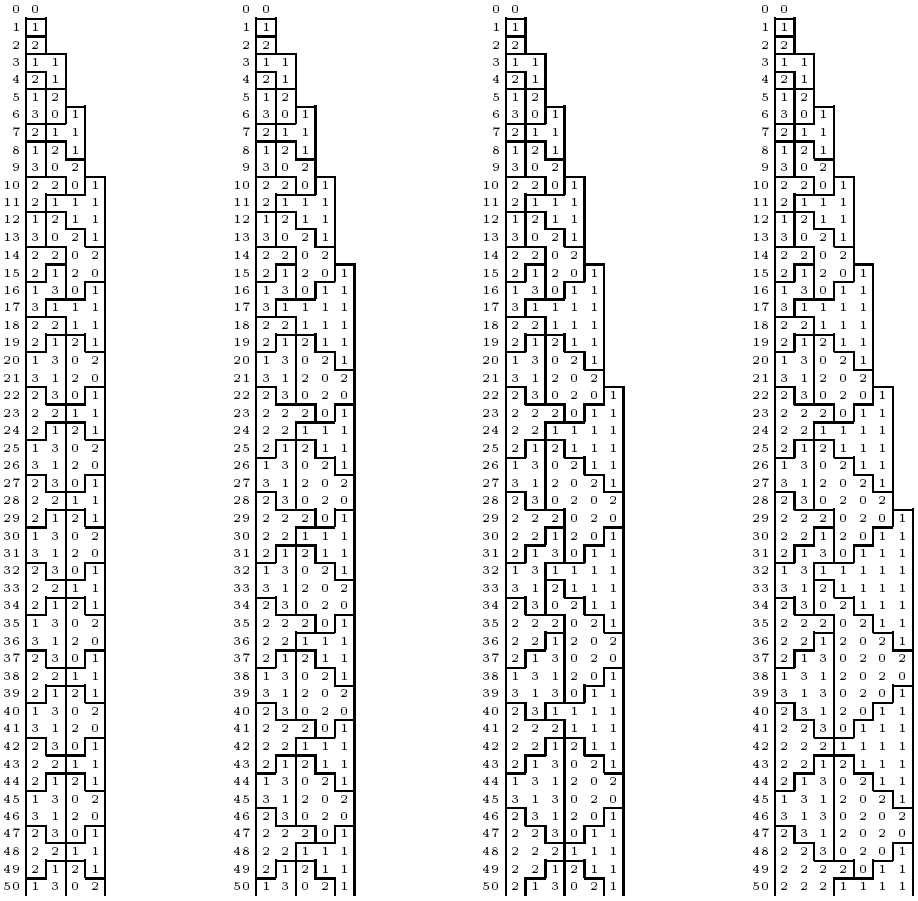
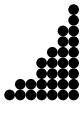


Figure 71 : Grain à grain limité par un mur pour une largeur de 5, 6, 7 et 8.

$$n \approx \frac{1}{2}\left(\frac{l}{2}\right)^2 + \left(\frac{l}{2}\right)^2 + lH \quad .$$

La croissance est cette fois-ci linéaire. Ce qui est normal car au lieu de remplir une surface en expansion (dimension 2), on remplit une bande de largeur constante (dimension 1).

$$H \approx \frac{n}{l} \quad .$$



12.3 Double Grain à Grain

L'espace est toujours borné, mais il y a un grain à grain à chaque extrémité. Diverses évolutions sont indiquées sur la figure 73.

Il y a maintenant des valeurs négatives, elles se traitent comme les valeurs positives correspondantes. Rappelons que pour le CFG, les valeurs, et non leurs différences dirigent la dynamique. La convergence du système est toujours assurée, car le nombre d'états est fini.

Il y a symétrie par rapport au milieu de la configuration. Cela découle directement de la symétrie du CFG et du fait qu'il travaille identiquement avec les valeurs positives et négatives.

Les deux tas commencent par s'étendre pour se limiter l'un l'autre. Après, toute la configuration monte progressivement, en fait, les périodes correspondent au temps de mettre une ou plusieurs couches de grains. Ceci est illustré par la figure 74 où sont représentées les silhouettes des configurations.

Sur la figure 73 on peut lire des temps de latence et des périodes. Ils sont regroupés sur la figure 72. Ils ont été obtenus grâce au logiciel de simulation décrit dans l'annexe A.

Largeur	8	9	10	11	12	13	14	15	16	17
Temps de latence	6	17	17	17	17	46	40	37	41	88
Période	4	9	5	11	6	13	7	14	8	17

Largeur	20	21	22	30	31	50	75	76	100	150
Temps de latence	64	143	113	237	209	777	1 586	1 481	2 724	8 577
Période	10	21	11	15	31	25	75	38	50	75

Figure 72 : Résultats numériques pour une limitation par un autre grain à grain.

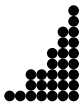
La parité joue un rôle important. Si l est paire, alors il se passe exactement comme si l'axe de symétrie, au milieu, était un mur. Chacun des deux grain à grain bloque l'autre. La période est alors égale à la moitié de la largeur. Le bloc de $-1|1$ central n'apparaît que si la largeur est paire.

Par symétrie et étude du centre, il se démontre que la période est égale à la largeur quand elle est impaire.

De même la croissance est linéaire et :

$$H \approx \frac{2n}{l} \ .$$

Le système possède énormément de symétries. Vu à la renverse, le double grain à grain représente l'érosion d'un plateau de hauteur infinie.



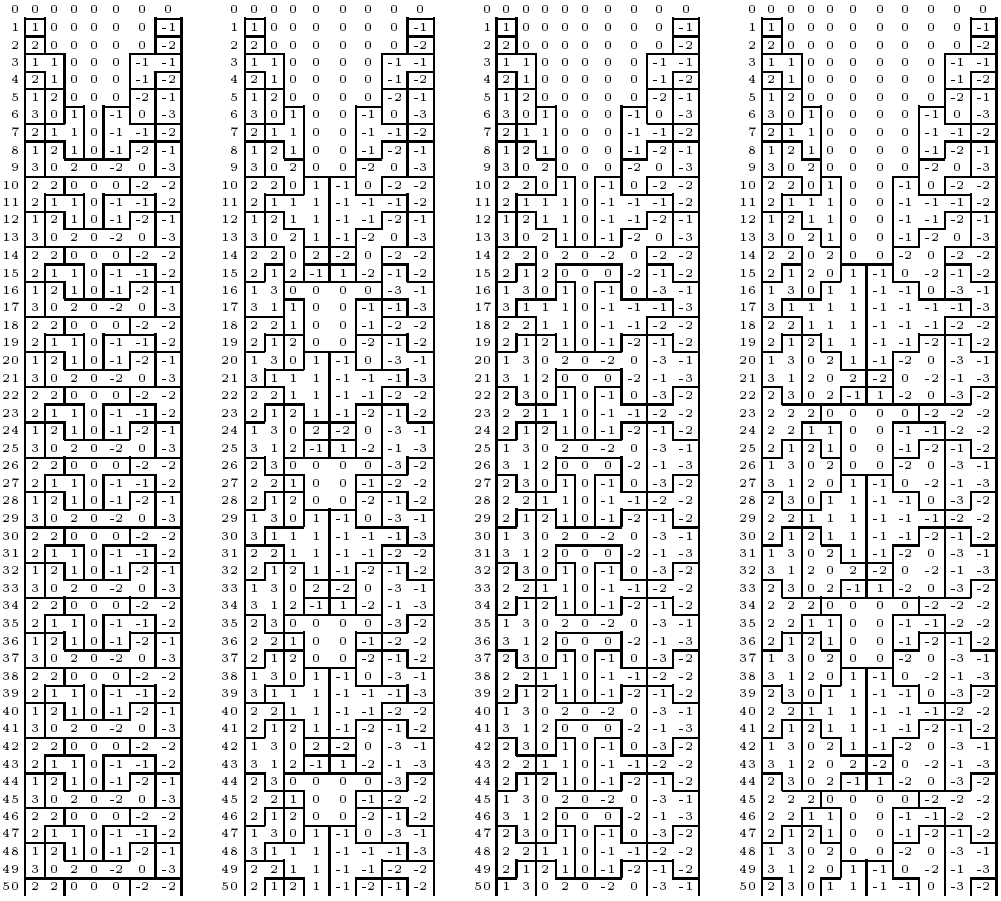


Figure 73 : Deux grain à grain distants de 8, 9, 10 et 11.

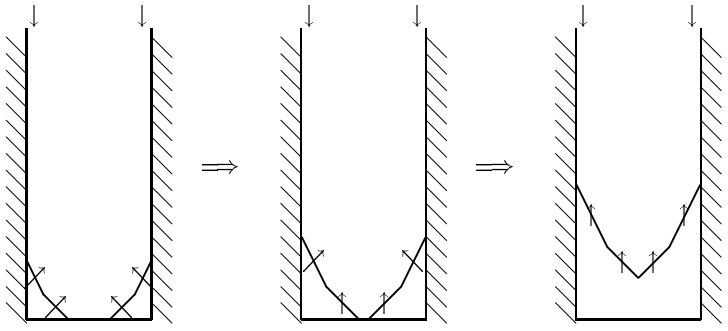
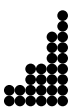


Figure 74 : Silhouette de l'évolution du grain à grain limité par un autre grain à grain.

12.4 Erosion d'un Contrefort

Dans cette section, nous trouvons des résultats proches de ceux énoncés par Anderson *et al.* dans [8]. Par d'autres méthodes et sur un modèle légèrement différent, ils trouvent les mêmes configurations finales et une borne asymptotique quadratique meilleure que la nôtre.

Leurs preuves sont basées sur le passage nécessaire par certaines positions. Ils s'occupent



ensuite des frontières et leur donnent un sens physique. De notre côté nous sommes parti de ces frontières pour aboutir à une dynamique des signaux dont nous tirons nos résultats.

Définition 51 Nous appelons contrefort de *hauteur* h, un tas de sable du type :

$$\begin{array}{lll} \text{SPM}: & [[\quad \ldots h\ h\ h\ 0\ 0\ 0 \ldots &]] \quad , \\ \text{CFG}: & \langle\!\langle \quad \ldots 0\ 0\ 0\ h\ 0\ 0\ 0 \ldots & \rangle\!\rangle \quad . \end{array}$$

Nous appelons *falaise* l'extrémité du contrefort et la notons O car elle correspond à une origine sur l'axe des piles. Les figures 75 et 76 montrent l'évolution à partir de contreforts de hauteurs de 10 et 15 respectivement.

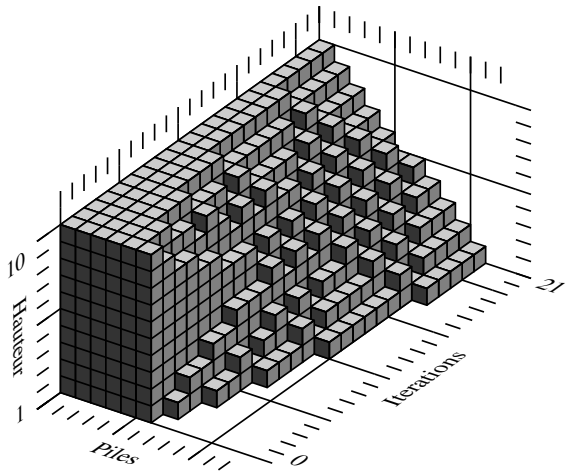


Figure 75 : Erosion d'un contrefort de hauteur 10.

Comme avant, nous représentons ces configurations en différences de hauteurs sur la figure 77. Il y a symétrie par rapport à la falaise : ce qui est tombé est exactement dans la même position que ce qui manque. Cela découle directement de la symétrie du CFG et du fait que la configuration initiale admette la symétrie.

Au début nous reconnaissons le grain à grain déjà étudié. Ce qui permet de dire directement que la hauteur du tas en bas monte proportionnellement à la racine du nombre d'itérations. Il en est de même pour la profondeur du trou du haut. La figure 78 montre la silhouette de cette évolution symétrique.

Tas et trous se rencontrent quand ils atteignent tous deux une hauteur de $h/2$. D'après les équations (20) de la section 10.3, ceci s'effectue en un nombre quadratique d'itérations. Ce premier temps est :

$$t_1 \ = \ \frac{h^2}{8+4\sqrt{2}} \ + \ o(h^2) \ .$$

Ensuite, tas et trous se stabilisent entre eux. Les parties de pente 2 n'en forment plus qu'une. Celle-ci charrie directement les grains en bas sans leur faire subir l'ancienne chute centrale. Le glissement de terrain suit la dynamique observée au chapitre 9.

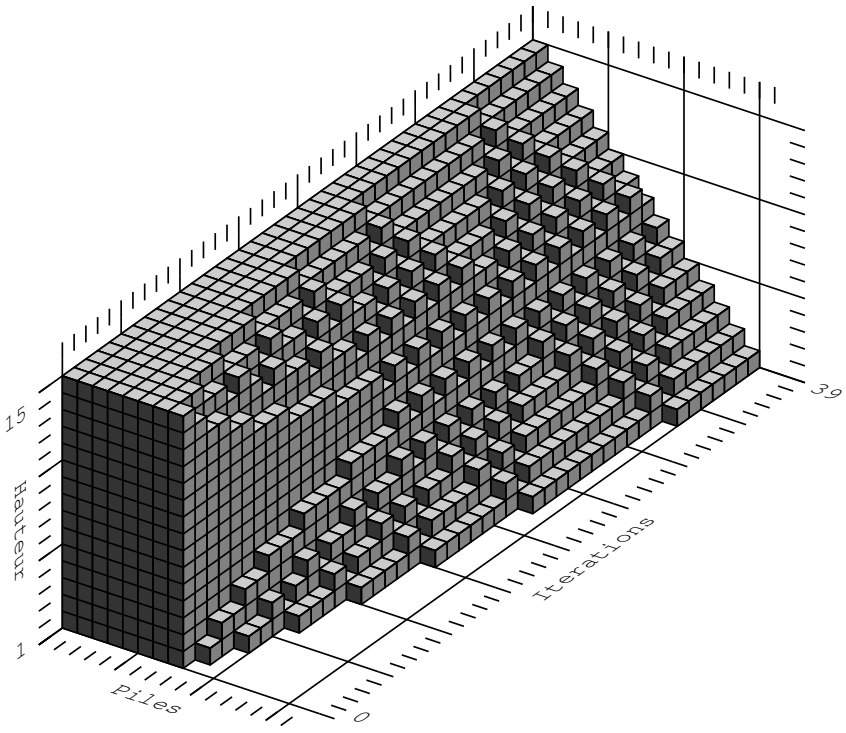


Figure 76 : Erosion d'un contrefort de hauteur 15.

Continuons notre étude en différences de hauteurs. Les configurations sont de la forme 11..1 puis ..0202.. d'un côté, et symétriquement, les mêmes motifs de l'autre.

Si h est paire, alors les deux ..0202.. se rejoignent et se recollent parfaitement. Ils se retrouvent à la fin sous forme d'un 0 qui se place au milieu des motifs ..11..

Par contre, si h est impaire, alors il y a un motif ..1313.. de largeur impaire entre les deux. Il finit par ne plus former qu'une marche de largeur 1. Cette marche se retrouve à la fin sous forme 1 qui se place parfaitement entre des motifs ..11..

Nous retrouvons l'écroulement d'une pile à ceci près que ce n'est pas la hauteur totale qui s'effondre, mais la partie haute qui est creusée.

La partie finale de la stabilisation est l'extension des parties de pente 1 aux dépens de celle de pente 2 au centre. Chaque M doit parvenir à la falaise. A chaque balayage de R, la valeur de D est augmentée de 1 et il n'y a plus de signal L pour la diminuer. Si l'on partait de D nul, il faudrait alors un nombre d'itérations de :

$$\sum_{i=1}^{\lfloor h/2 \rfloor} i \;=\; \frac{\lfloor h/2 \rfloor (\lfloor h/2 \rfloor + 1)}{2} \;=\; \frac{h^2}{8} \;+\; o(h^2) \;\; .$$

Ce temps est borné par une quantité proportionnelle au carré du nombre d'itérations.

En tout, le temps de stabilisation est quadratique et la configuration finale est une suite de 1

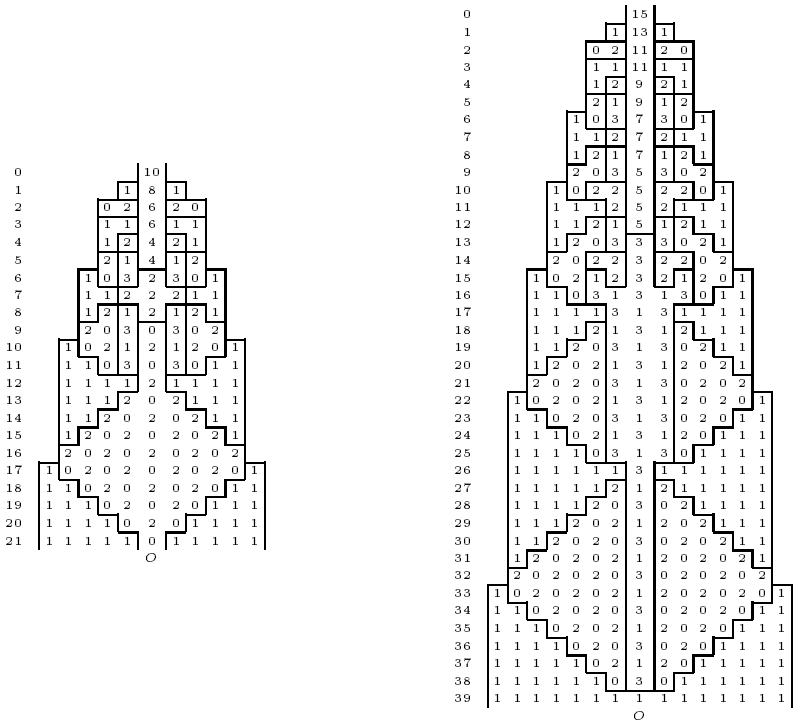


Figure 77 : Contreforts de hauteur 10 et 15 en différences de hauteurs.

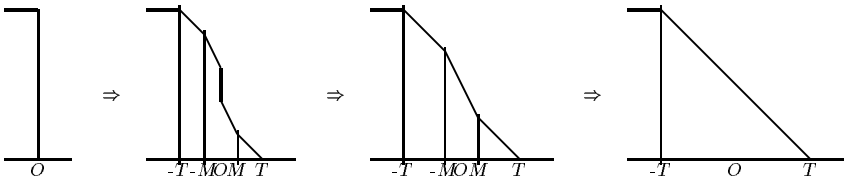


Figure 78 : Forme générale de l'évolution de l'érosion d'un contrefort.

centrée sur O, si h est impaire. Si h est paire, il y a un zéro au milieu.

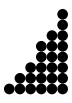
$$\frac{h^2}{8+4\sqrt{2}} + o(h^2) \leq t_c \leq \frac{h^2}{4} + o(h^2) \ .$$

Nous retrouvons les résultats d'Anderson *et al.* dans [8]. Par d'autres méthodes, ils trouvent les mêmes configurations finales et une borne asymptotique quadratique meilleure que la nôtre. Mais leur étude concerne le cas où un nombre quelconque de grains peut transiter entre deux piles :

Théorème 52 (Anderson et al. 1989 [8]) *Si le saut initial vaut* $2m+1$ *alors le système se stabilise en au plus* $(\frac{\pi^2}{6}-1)m^2 + O(m)$ *coups.*

La configuration finale est unique quel que soit le mode d'itération séquentiel ou parallèle, grâce aux résultats de Goles et Kiwi [52] :

$$\begin{aligned} \text{SPM} &: [[\ldots\ h\ h\ (h-1)\ (h-2)\ldots\ 2\ 1\ 0\ 0\ 0\ldots]] \ , \\ \text{CFG} &: \langle\!\langle\ \ \ldots 1\ 1 \qquad 1 \qquad \ldots\ 1\ 1\ 1\ 0\ 0\ldots \rangle\!\rangle \ , \end{aligned}$$



si h est impaire, et :

$$\begin{aligned} \text{SPM} &: [[\ldots h\ h\ (h-1)\ (h-2)\ldots (h/2)\ (h/2)\ldots 2\ 1\ 0\ 0\ 0\ldots]] \ , \\ \text{CFG} &: \langle\!\langle\ \ldots 1\ 1 \qquad 1 \qquad \ldots\ 1 \qquad 0 \qquad 1\ldots 1\ 1\ 0\ 0\ldots \rangle\!\rangle \ . \end{aligned}$$

si h est paire.

Temps Séquentiel

Tous les grains vont dans la même direction (à droite) sans jamais faire de pas en arrière.

Dans le cas de l'écroulement d'une pile, Goles et Kiwi [52] ont utilisé une énergie pour calculer le temps séquentiel. Elle revient à sommer tous les déplacements élémentaires des grains, chaque itération séquentielle correspondant à un seul déplacement élémentaire.

L'hypothèse qu'ils utilisent est que tous les grains vont toujours dans le même sens ce qui leur permet de faire la différence entre ce qui est tombé (uniquement situé sur les piles originelles) et jusqu'où sont arrivés les grains en bas. L'hypothèse reste valable ici. La seule différence est que maintenant tous les grains ne partent pas de la même pile.

Calculons cette différence d'énergie pour trouver le temps de relaxation séquentiel. Celle-ci est la somme des déplacements des grains, à chaque instant, un grain étant déplacé d'une case, soit la différence entre les sommes des coordonnées.

Tous les grains dans la partie haute doivent en sortir. Une fois la falaise franchie, ils doivent parcourir la distance jusqu'à leurs destinations. L'énergie est la somme de tous ces déplacements.

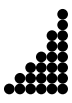
Commençons par le cas où h est impaire, posons $h = 2m + 1$. La configuration finale est régulière. En comptant par hauteur, ce qui manque et ce qui est apparu, nous trouvons la formule suivante :

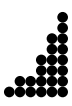
$$\Delta E = \sum_{i=m+2}^{2m+1} (i-m-1)^2 + \sum_{i=1}^{m} (m+1-i)^2 = \frac{m(m+1)(2m+1)}{3} \ . \tag{22}$$

Pour le cas pair, posons $h{=}2m$. Il y a une irrégularité au milieu. Nous nous retrouvons avec des opérations à peine plus compliquées. En comptant toujours par hauteur, nous trouvons la formule suivante :

$$\Delta E = \sum_{i=m+1}^{2m} (i-m)^2 + \sum_{i=1}^{m} (m+1-i)^2 = \frac{m(m+1)(2m+1)}{3} \ . \tag{23}$$

Le temps de relaxation séquentiel est donc cubique dans tous les cas. Il est à signaler que des contreforts de hauteur $2m$ et $2m+1$ mettent exactement autant de temps à se stabiliser.





Chapitre 13

Conclusion

Nous avons démontré que l'écroulement d'une pile se fait en temps linéaire.

Au cours de notre étude, nous avons observé différents cas : arrivée continuelle de grains sur la pile placée à l'origine dans un espace non borné puis limité par une sortie ou un mur, de deux grain à grain, d'un contrefort. Dans chacun des cas, nous avons trouvé un facteur d'accélération de l'ordre du nombre de piles utilisées. Par comparaison avec le temps séquentiel, ceci prouve la nature intrinsèquement parallèle de la dynamique des tas de sable. Ceci complète les travaux de Goles et Kiwi, et Anderson *et al.*

Au cours de notre étude, des signaux ont été mis à nu dans leurs évolutions et leurs interactions. Ceux-ci permettent de mieux comprendre le transit des grains et expliquent les différents temps de relaxation découverts.

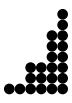
Nous avons développé des techniques et acquis des connaissances de nature locale et globale pour le système. Sans doute cela doit-il pouvoir être utilisé sur d'autres automates cellulaires, ou réseaux de Pétri.

Il serait intéressant de voir dans quelle mesure ces résultats se retrouveraient en dimension 2, ou sur certains types de graphes. Ceci devrait aboutir à une meilleure modélisation et à une meilleure compréhension des mécanismes d'équilibrage de charge et de certains phénomènes physiques.

D'un autre côté, nous avons exploré des configurations de départ relativement simples, comme des rectangles, mais comment se généraliseraient ces résultats à des formes plus complexes comme par exemple une rangée de gratte-ciel?

Le « Chip firing game » décrit par Anderson *et al.* dans [8] faisait en fait tomber la moitié de chaque côté. Nous nous sommes restreints à un grain seulement. Que se passe-t-il si l'on se place dans leur modèle? La configuration finale reste la même et le temps devrait être plus court. Il l'est dans l'écroulement d'une pile, le cas traité dans leur article. Mais au regard d'écoulements, comme des flux d'informations dans un réseau, il nous semble nécessaire de donner une borne au nombre de grains pouvant transiter entre deux piles à chaque itération. Remplacer la borne par k ferait-il accélérer le système d'un facteur k?

Anderson *et al.* observent des signaux de même nature que les nôtres. Notre méthode



d'exploration devrait pouvoir être généralisée à leur modèle pour trouver l'ordre de grandeur du temps de stabilisation. Peut-on y gagner un facteur N ou $\sqrt{N}$?

De même, changer le seuil pour la chute de grains, le nombre maximum de grains tombant et le nombre de grains arrivant doivent permettre d'obtenir divers comportements.

Nous avons uniquement considéré le transit des tâches, il reste à ajouter un processus faisant disparaître les grains de temps à autre, ce qui correspond à la réalisation de la tâche.

Table des Symboles

Automates à Partitions

$\mathcal{L}$ Grille, p. 11.

d Dimension de la grille, p. 11.

$\mathcal{Q}$ Ensemble des états, p. 11.

$\mathcal{C}$ Ensemble des configurations, p. 11.

AC Automate Cellulaire, p. 12.

L Grille d'un AC, p. 12.

Q Ensemble des états d'un AC, p. 12.

f Fonction locale d'un AC, p. 12.

F Fonction globale d'un AC, p. 12.

r Rayon d'un AC, p. 12.

C Ensemble des configurations d'un AC, p. 12.

AP Automate à partitions, p. 12.

(h, v) Taille des tuiles, p. 12.

t Transition locale, d'un AP, p. 13.

T Transition d'un AP, p. 13.

n Nombre de Partitions, p. 13.

$\mathcal{O}$ n-uplet des partitions, p. 13.

$\mathcal{T}$ Fonction globale ou transition globale, p. 14.

AC-R Automates cellulaires réversibles, p. 16.

AP-R Automates à partitions réversibles, p. 16.

MT Machine de Turing, p. 17.

Σ Alphabet d'une MT, p. 17.

Q Ensemble des états d'une MT, p. 17.

s_0 Etat initial d'une MT, p. 17.

FIN Arrêt d'une MT, p. 17.

δ Fonction de transition d'une MT, p. 17.

BBM Billiard ball model, p. 31.

t_{BBM} Transition locale de BBM, p. 31.

$\star$ Etat ajouté pour le simulateur universel, p. 39.

BBM* Automate à partitions BBM modifié, p. 39.

$t_{\text{BBM}^\star}$ Transition locale de BBM*, p. 39.

Tas de Sable

$\mathcal{C}$ Ensemble des configurations des piles de sable, p. 47.
$[[\ldots]]$ Configuration, p. 47.
F Fonction de transition, p. 47.
SPM Modèle tas de sable, p. 47.
$\square(.)$ Fonction seuil, p. 47.
$\langle\langle\ldots\rangle\rangle$ Configuration en différences de hauteurs, p. 48.
Θ Transition en différences de hauteurs, p. 48.
CFG Chip firing game, p. 48.
ξ Suite des configurations du grain à grain, p. 51.
L,M,R Signal de gauche, frontière médiane et signal de droite, p. 52.
H Hauteur maximale de la configuration, p. 56.
T Longueur totale de la configuration, p. 56.
n Nombre d'itérations ou de grains tombés, p. 56.
dl, dr Direction de L et de R, p. 56.
G, D Tailles des moitiés gauche et droite, p. 57.
e, f Temps avant que L (R) ne revienne au centre, p. 57.
k Numéro du retour de R au milieu M, p. 67.
γ_k Nombre de collisions de L avec M entre k et k+1 , p. 67.
Δt Temps entre q passages de R, p. 68.
α Nombre de fois que L atteint le milieu M entre k et k+q, p. 68.
N Nombre total de grains, p. 73.
t_1 Durée de la première phase de l'écroulement d'une pile, p. 73.
F Signal de fin de la première phase, p. 75.
M' Nouvelle frontière médiane de la seconde phase, p. 75.
L' Nouveau signal gauche de la seconde phase, p. 75.
B Bordure statique, p. 75.
d_0 Distance que M' doit parcourir, p. 78.
$T_{\text{par}}(N)$ Temps parallèle d'écroulement d'une unique pile, p. 79.
l Largeur de l'espace borné, p. 81.
t_l Temps de latence ou durée de la période transitoire, p. 83.
t_c Temps de convergence, p. 83.
h Hauteur d'un contrefort, p. 88.
O Origine du contrefort, p. 88.
E Energie de Goles et Kiwi pour les tas de sable, p. 91.

Look I gotta go
Yeah I'm running outta change
There's a lot of things
If I could I rearrange.

Bono - U2,
The Fly.

Annexe

Annexe A

Outil de Représentation des Tas de Sable

La seconde partie de la thèse a été l'occasion de développer un petit logiciel en C++. Il nous a permis de faire les nombreuses figures en trois dimensions, ainsi que les schémas en différences de hauteurs, directement récupérables sous LaTeX.

C'est un programme qui s'exécute sous n'importe quel shell Unix. Il s'appelle `spm`, son utilisation est :

```
spm [options] [model]   .
```

Les options définissent quelles sorties doivent être générées et éventuellement les itérations à considérer. Elles sont définies sur la figure 79.

Option	Signification	Fichier résultant
`-e`	Légendes des figures en anglais	
`-d <de>`	Première itération : `<de>`	
`-f <fi>`	Dernière itération : `<fi>`	
`-i`	Informations sur les paramètres et recherche de périodes et temps de latence	`*.info`
`-p`	Vue 3-D en Postscript	`*.ps`
`-t`	Sortie LaTeX	`*.cfg.tex`
`-% <he>`	Met à l'échelle `<he>`/100 la figure postscript	

Figure 79 : Options pour `spm`

Par exemple, `-p` a été utilisé pour faire les différentes vues en trois dimensions. De même, c'est avec `-t` que nous avons obtenu les différentes figures en différences de hauteurs.

C'est grâce à l'option `-i` que les temps des figures 69 et 72 ont été calculés.

Les modèles définissent ce qui doit être itéré et avec quels paramètres. Ils sont définis sur la

figure 80.

Modèle	Signification, exemple	Préfixe des fichiers
b <la> <fi>	Grain à grain borné par un mur, largeur <la>, dernière itération <fi>, figure 71	borne_<la>_<fi>
c <ha>	Contrefort de hauteur <ha>, figures 76, 77	contrefort_<ha>
d <la> <fi>	Double grain à grain, largeur <la>, dernière itération <fi>, figure 73	double_<la>_<fi>
e <ha>	Effondrement d'une pile de hauteur <ha>, figure 55	effondrement_<ha>
f <fi[.conf]>	Configuration initiale dans <fi>.conf	<fi>
g <de> <fi>	Grain à grain, itérations <de> à <fi>, figures 34, 35, 36	grain_<de>_<fi>
i <lo> <fi>	Plateau avec hauteur infinie, de longueur <lo>, dernière itération<fi>	infinie_<lo>_<fi>
m <de> <fi>	Grain à grain seul, itérations <de> à <fi>, figure 82	milieu_<de>_<fi>
p <ha> <lo>	Plateau, de hauteur <ha> et de longueur <lo>, figure 81	plateau_<ha>_<lo>
v <la>	Grain à grain avec une sortie, largeur <la>, figure 66	vide_<la>

Figure 80 : Modèles pour `spm`

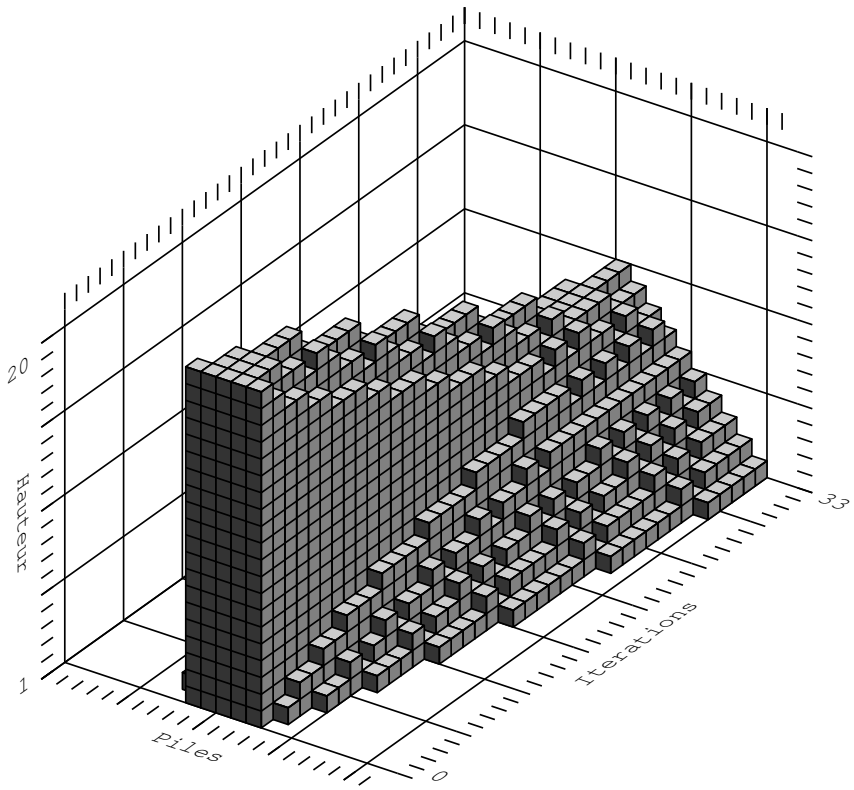
Par exemple, l'effondrement d'un pile seule, figure 65, est un plateau de largeur 1 :

`spm -p p 1 50`.

En cas de mauvaise formulation de l'ordre, la syntaxe est ré-affichée.

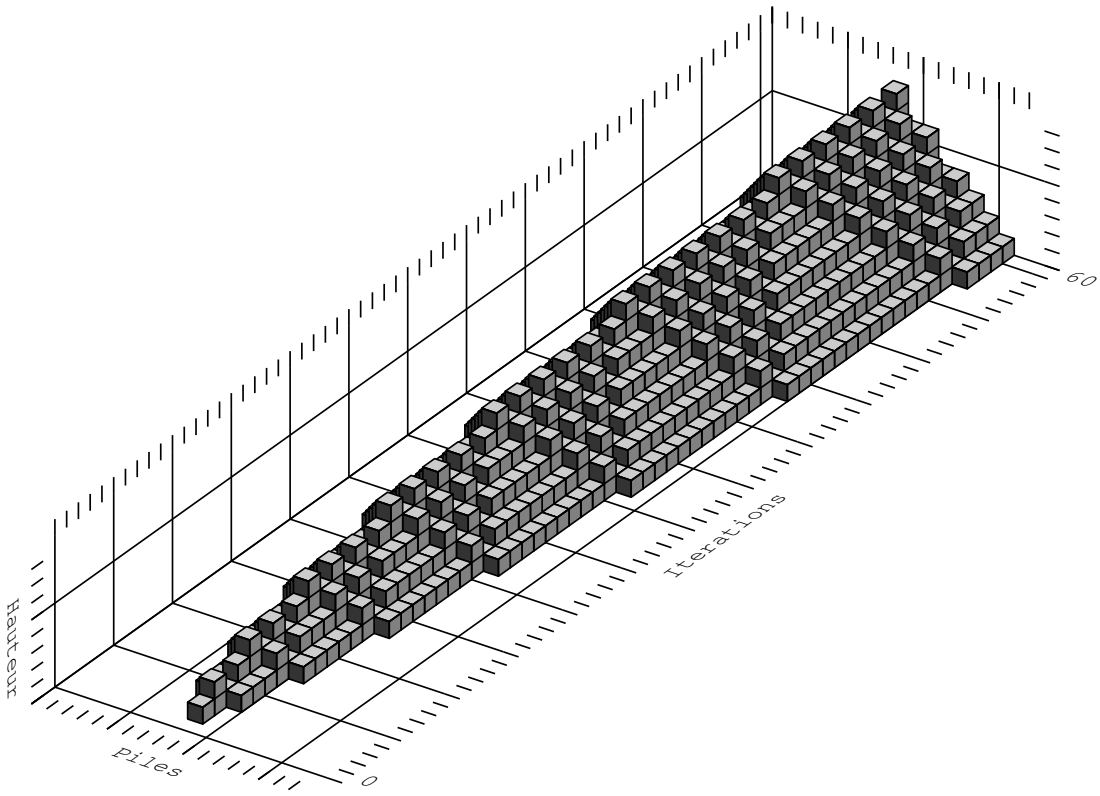
Les sources du programme sont disponibles sur :

`http://dept-info.labri.u-bordeaux.fr/~jdurand/recherche.html` .



spm -p p 20 5

Figure 81 : Erosion d'un plateau de hauteur 20 et de largeur 5.




```
spm -p m 0 60
```

Figure 82 : Grain à grain sans mur ni sortie.

Bibliographie

[1] J. Albert et K. Čulik II. A simple universal cellular automaton and its one-way and totalistic version. *Complex Systems*, 1:1–16, 1987.

[2] J.-P. Allouche. Finite automata in 1-d and 2-d physics. Dans *Number Theory and Physics*. Springer-Verlag, 1990.

[3] J.-P. Allouche, F. von Heaseler, H.-O. Peitgen, et G. Skordev. Linear cellular automata, finite automata and pascal's triangle. *Discrete Applied Mathematics*, 66:1–22, 1996.

[4] S. Amoroso et G. Cooper. The garden-of-eden theorem for finite configurations. *Proceeding of the American Mathematical Society*, 26:158–164, 1970.

[5] S. Amoroso et G. Cooper. Tessellation structure for reproduction of arbitrary patterns. *Journal of Computer and System Sciences*, 5:455–464, 1971.

[6] S. Amoroso, G. Cooper, et Y. Patt. Some clarifications of the concept of a garden-of-eden configuration. *Journal of Computer and System Sciences*, 10:77–82, 1975.

[7] S. Amoroso et Y. Patt. Decision procedure for surjectivity and injectivity of parallel maps for tessellation structure. *Journal of Computer and System Sciences*, 6:448–464, 1972.

[8] R. Anderson, L. Lovász, P. Shor, J. Spencer, E. Tardos, et S. Winograd. Disks, balls and walls: Analysis of a combinatorial game. *American Mathematical Monthly*, 96:481–493, 1989.

[9] P. Bak et K. Chen. Self-organized criticality. Distributed at the 2nd Latinamerican Conf. on Nonlinear Phenomena, Santiago, Chile, 1990.

[10] P. Bak, C. Tang, et K. Wiesenfeld. Self-organized criticality. *Physical Review A*, 38(1):364–374, 1988.

[11] P. Bak, T. Tang, et K. Wiesenfeld. Self-organized criticality: An explanation of $1/f$ noise. *Physical Review Letters*, 59(4):381–384, 1987.

[12] C. H. Bennett. Logical reversibility of computation. IBM *Journal of Research and Development*, 6:525–532, 1973.

[13] C. H. Bennett. Notes on the history of reversible computation. IBM *Journal of Research and Development*, 32(1):16–23, 1988.

[14] E. Berlekamp, J. Conway, et R. Guy. *Winning Ways for your Mathematical Plays (games in particular)*, volume 2. Academic Press, 1982.

[15] J. Bitar et E. Goles. Parallel chip firing games on graphs. *Theoretical Computer Science*, 92:291–300, 1992.

[16] A. Björner et L. Lovász. Chip-firing games on directed graphs. *Journal of Algebraic Combinatorics*, 1:305–328, 1992.

[17] A. Björner, L. Lovász, et P. W. Shor. Chip-firing games on graphs. *European Journal of Combinatorics*, 12:283–291, 1991.

[18] T. Boykett. Combinatorial constructor of one dimensional reversible cellular. *Contribution to General Algebra*, 9, 1995.

[19] M. Bramson et C. Neuhauser. Survival of one-dimensional cellular automata under random perturbations. *Annals of Probability*, 22(1):244–263, 1994.

[20] A. Burks. *Essays on Cellular Automata*. Univ. of Illinois Press, 1970.

[21] M. Cannatora, S. Di Gregorio, B. Rongo, W. Spataro, G. Spezzano, et D. Talia. A parrallel cellular automata environment on multicomputers for multicomputational science. *Parallel Computing*, 21:803–823, 1995.

[22] J. Conway. Mathematical games. *Scientific American*, pages 120–123, Octobre 1970.

[23] P. Cordero, E. Goles, et G. Hernández. Q2R + Q2R as a universal billiard. *International Journal of Modern Physics C*, 3,2-2:251–266, 1992.

[24] R. Cori, Y. Métivier, et W. Zielonka. Asynchronous mapping and asynchronous cellular automata. *Information and Computation*, 106:159–202, 1993.

[25] M. Creutz. Deterministic Ising model. *Annals of Physics*, 167:62–76, 1986.

[26] K. Čulik II. On invertible cellular automata. *Complex Systems*, 1:1035–1044, 1987.

[27] K. Čulik II. How to fire almost any arbitrary pattern on a cellular automaton. Dans Boccara, Goles, Martinez, et Picco, éditeurs, *Cellular Automata and Cooperative Systems*, pages 101–109. Kluwer, 1991.

[28] K. Čulik II et S. Dube. Fractal and recurrent behavior of cellular automata. *Complex Systems*, 3:253–267, 1989.

[29] K. Čulik II, J. Pachl, et S. Yu. On the limit sets of cellular automata. SIAM *Journal Computation*, 18(4):831–842, Août 1989.

[30] M. Daumas, J. O. Durand-Lose, et L.-P. Tock. High speed implementation of a cellular automaton. Dans XIV *International Conference of the Chilean Computer Science Society*, pages 283–294, 1994.

[31] M. Davis et E. Weyuker. *Computability, Complexity and Languages. Fundamentals of Theoretical Computer Science*. Academic Press, 1983.

[32] D. Dhar. Self-organized critical state of sand pile automaton models. *Physical Review Letters*, 64(14):1613–1616, 1990.

[33] D. Dhar et R. Ramaswamy. Exactly solved model of self-organized critical phenomena. *Physical Review Letters*, 63(16):1659–1662, 1989.

[34] J.-C. Dubacq. How to simulate Turing machines by invertible 1d cellular automata. *International Journal of Foundations of Computer Science*, 6(4):395–402, 1995.

[35] J. O. Durand-Lose. Partitioning automata, cellular automata, simulation and reversibility. Rapport technique 95–01, LIP, ENS Lyon, 46 allée d'Italie, 69 364 LYON 7, 1995.

[36] J. O. Durand-Lose. Reversible cellular automaton able to simulate any other reversible one using partitioning automata. Dans LATIN'*95*, numéro 911 dans Lecture Notes in Computer Science, pages 230–244. Springer-Verlag, 1995.

[37] J. O. Durand-Lose. Any reversible cellular automaton can be represented with block permutations. Rapport technique 1135-96, LaBRI, Université Bordeaux I, 33 405 Talence Cedex, France, 1996.

[38] J. O. Durand-Lose. *Automates Cellulaires, Automates à Partitions et Tas de Sable.* Thèse de doctorat, LaBRI, 1996. In French.

[39] J. O. Durand-Lose. Parallel transcient time of 1 dimensional sand pile. Rapport technique 1148-96, LaBRI, 1996. to appear in TCS A.

[40] J. O. Durand-Lose. Sand dripping in linear space. Dans *Cellular Automata Workshop 1996 - Schloß Rauischholzhauen*, pages 14–16, 1996.

[41] J. O. Durand-Lose. Sand piles in digraphs. Rapport technique 1113-96, LaBRI, Université Bordeaux I, 33 405 Talence Cedex, France, 1996.

[42] J. O. Durand-Lose. Grain sorting in the one dimensional sand pile model. Rapport technique 11457-97, LaBRI, Université Bordeaux I, 33 405 Talence Cedex, France, 1997. To appear in Complex systems.

[43] J. O. Durand-Lose. Intrinsic universality of a 1-dimensional reversible cellular automaton. Dans STACS'*97*, numéro 1200 dans Lecture Notes in Computer Science, pages 439–450. Springer-Verlag, 1997.

[44] K. Eriksson. Reachability is decidable in the numbers game. *Theoretical Computer Science*, 131:431–439, 1994.

[45] E. Fredkin et T. Toffoli. Conservative logic. *International Journal of Theoretical Physics*, 21, 3/4:219–253, 1982.

[46] M. Gardner. The fantastic combinations of John Conway's new solitaire game 'Life'. *Scientific American*, 223(4):120–123, 1970.

[47] M. Garzon et F. Botelho. Real computation with cellular automata. Dans Boccara, Goles, Martinez, et Picco, éditeurs, *Cellular Automata and Cooperative Systems*, pages 191–202. Kluwer, 1991.

[48] E. Goles. Sand pile automata. *Annales Institut Henri Poincaré, Physique Théorique*, 56(1):75–90, 1992.

[49] E. Goles et M. Kiwi. One-dimensional sand-pile, cellular automata and related models. Dans *Nonlinear Phenomena in Fluids, Solids and Other Complex Systems*, pages 169–186. North-Holland, 1991.

[50] E. Goles et M. Kiwi. Sand-pile dynamics in one-dimensional bounded lattice. Dans Boccara, Goles, Martinez, et Picco, éditeurs, *Cellular Automata and Cooperative Systems*, pages 211–225. Kluwer, 1991.

[51] E. Goles et M. Kiwi. Dynamics of sand-piles games on graphs. Dans LATIN'*92*, numéro 583 dans Lecture Notes in Computer Science, pages 219–230. Springer-Verlag, 1992.

[52] E. Goles et M. Kiwi. Games on line graphs and sand piles. *Theoretical Computer Science*, 115:321–349, 1993.

[53] E. Goles, A. Maass, et S. Martinez. On the limit set of some universal cellular automata. *Theoretical Computer Science*, 110:53–78, 1993.

[54] E. Goles et M. Margenstern. Universality of the chip firing game. Rapport technique 95–33, LITP, IBP, Paris 7, 1995. To appear in Theoretical Computer Science.

[55] E. Goles et S. Martínez. *Neural and Automata Networks, Dynamics Behavior and Applications.* Mathematics and Its Applications. Kluwer, 1991.

[56] D. Gordon. On the computation power of totalistic cellular automata. *Mathematical System Theory*, 20:43–52, 1987.

[57] S. di Gregorio, R. Rongo, W. Spataro, G. Spezzano, et D. Talia. A parallel cellular environment for high performance scientific computing. Dans *High Performance Computing and Networking*, numéro 1067 dans Lecture Notes in Computer Science, pages 514–521, 1996.

[58] H. Gutowitz, éditeur. *Cellular Automata, Theory and Experimentation.* MIT/North-Holland, 1991.

[59] J. Hardy, de Pazzis O., et Y. Pomeau. Molecular dynamics of a classical lattice gas: Transport properties and time correlation functions. *Physical Review*, A(13):1949–1960, 1976.

[60] G. A. Hedlund. Endomorphism and automorphism of the shift dynamical system. *Mathematical System Theory*, 3:320–375, 1969.

[61] O. Heen. *Economie de Ressources sur Automates Cellulaires.* Thèse de doctorat, LITP, IBP, Université Paris 7, 1996. In French.

[62] G. T. Herman et G. Rozenberg. *Developmental Systems and Languages.* North-Holland, 1975.

[63] C. Hochberger, R. Hoffmann, et S. Waldschmidt. Compilation of CDL for different target architectures. Dans *Parallel Computing Technologies*, numéro 964 dans Lecture Notes in Computer Science, 1995.

[64] R. Hoffmann, K.-P. Völkmann, et M. Sobolewski. The cellular processing machine CEPRA-8L. Dans *Parcella 94*, numéro 81 dans Mathematical Research, pages 179–188, 1994.

[65] J. Hopcroft et J. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison Wesley, 1979.

[66] L. Hurd, J. Kari, et K. Čulik II. The topological entropy of cellular automata is uncomputable. *Ergodic Theory and Dynamical System*, 12:255–265, 1992.

[67] O. Ibarra et T. Jiang. Relating the power of cellular arrays to their closure properties. *Theoretical Computer Science*, 57:225–238, 1988.

[68] H. Jeager, S. Nagel, et R. Behringer. The physics of granular materials. *Physics Today*, pages 32–38, april 1996.

[69] E. Jen. Transience and dislocations in one-dimensional cellular automata. Dans Boccara, Goles, Martinez, et Picco, éditeurs, *Cellular Automata and Cooperative Systems*, pages 299–310. Kluwer, 1991.

[70] L. P. Kadanoff, S. R. Nagel, L. Wu, et S. Zhou. Scaling and universality in avalanches. *Physical Review A*, 39(12):6514–6537, 1989.

[71] J. Kari. *Decision Problems Concerning Cellular Automata.* Thèse de doctorat, University of Turku (Finland), 1989.

[72] J. Kari. Reversibility of 2D cellular automata is undecidable. *Physica D*, 45:379–385, 1990.

[73] J. Kari. Properties of limit sets of cellular automata. Dans Boccara, Goles, Martinez, et Picco, éditeurs, *Cellular Automata and Cooperative Systems*, pages 311–321. Kluwer, 1991.

[74] J. Kari. The nilpotency problem of one-dimensional cellular automata. SIAM *Journal on Computing*, 21(3):571–586, 1992.

[75] J. Kari. Reversibility and surjectivity problems of cellular automata. *Journal of Computer and System Sciences*, 48(1):149–182, 1994.

[76] J. Kari. Rice's theorem for the limit sets of cellular automata. *Theoretical Computer Science*, 127:229–254, 1994.

[77] J. Kari. Representation of reversible cellular automata with block permutations. *Mathematical System Theory*, 29:47–61, 1996.

[78] A. Kolmogorov. Three approaches to the quantitative definition of information. *International Journal of Computer Mathematics*, 2:157–168, 1968.

[79] Le jeu de la Vie. *Jeux et Stratégies*, 21, 1983.

[80] J. van Leeuwen, éditeur. *Handbook of Theoretical Computer Science*, volume A. MIT Press, 1990.

[81] J. van Leeuwen, éditeur. *Handbook of Theoretical Computer Science*, volume B. MIT Press, 1990.

[82] M. Li et P. Vitanyi. *An Introduction to Kolmogorov Complexity and its Applications.* Texts and Monographs in Computer Science. Springer-Verlag, 1993.

[83] A. Lindenmayer et G. Rozenberg, éditeurs. *Automata, Language and Development.* North-Holland, 1976.

[84] K. Lindgren et M. G. Nordahl. Universal computation in simple one-dimensional cellular automata. *Complex Systems*, 4:299–318, 1990.

[85] B. Litow, S. Hosseini, K. Vairavan, et G. S. Wolffe. Performance characteristics of a load balancing algorithm. *Journal of Parallel and Distributed Computing*, 31:159–165, 1995.

[86] A. Maass. Some coded systems that are not unstable limit sets of cellular automata. Dans Boccara, Goles, Martinez, et Picco, éditeurs, *Cellular Automata and Cooperative Systems*, pages 433–449. Kluwer, 1991.

[87] A. Maass. On sofic limit sets of cellular automata. *Ergodic Theory and Dynamic Systems*, 15:663–684, 1995.

[88] M Margenstern. Non-erasing Turing machines: A new frontier between a decidable halting problem and universality. Dans LATIN'*95*, numéro 911 dans Lecture Notes in Computer Science, pages 386–397. Springer-Verlag, 1995.

[89] N. Margolus. Physics-like models of computation. *Physica D*, 10:81–95, 1984.

[90] N. Margolus. *Physics and Computation.* Thèse de doctorat, MIT, 1988.

[91] B. Martin. *Construction Modulaire d'Automates Cellulaires.* Thèse de doctorat, LIP, Ecole Normale Supérieure de Lyon-Université Claude Bernard Lyon 1 (France), 1993. In French.

[92] B. Martin. Self-similar fractals can be generated by cellular automata. Dans Boccara, Goles, Martinez, et Picco, éditeurs, *Cellular Automata and Cooperative Systems*, pages 463–471. Kluwer Academic Publishers, 1993.

[93] B. Martin. A universal cellular automaton in quasi-linear time and its S-n-m form. *Theoretical Computer Science*, 123:199–237, 1994.

[94] A. Maruoka et M. Kimura. Conditions for injectivity of global maps for tessellation automata. *Information and Control*, 32:158–162, 1976.

[95] A. Maruoka et M. Kimura. Strong surjectivity is equivalent to C-injectivity. *Theoretical Computer Science*, 18:269–277, 1982.

[96] J. Mazoyer. A 6-states minimal-time solution to the firing squad synchronisation problem. *Theoretical Computer Science*, 50:183–237, 1986.

[97] J. Mazoyer et N. Reimen. A linear speed-up theorem for cellular automata. *Theoretical Computer Science*, 101(1):59–98, 1992.

[98] N. Menyhárd. Kinetic Ising cellular automata models in one dimension. *Journal of Physics A*, 23(11):2147–2156, 1990.

[99] M. Minsky. *Finite and Infinite Machines*. Prentice Hall, 1967.

[100] E. Moore. Machine models of self-reproduction. Dans *Proceeding of Symposium on Applied Mathematics*, volume 14, pages 17–33, 1962.

[101] K. Morita. A simple construction method of a reversible finite automaton out of Fredkin gates, and its related problem. *Transactions of the* IEICE, E 73(6):978–984, Juin 1990.

[102] K. Morita. Any irreversible cellular automaton can be simulated by a reversible one having the same dimension (on finite configurations). *Technical Report of the* IEICE, COMP 92-45 (1992-10):55–64, 1992.

[103] K. Morita. Computation-universality of one-dimensional one-way reversible cellular automata. *Information Processing Letters*, 42:325–329, 1992.

[104] K. Morita. Reversible simulation of one-dimensional irreversible cellular automata. *Theoretical Computer Science*, 148:157–163, 1995.

[105] K. Morita et M. Harao. Computation universality of one-dimensional reversible (injective) cellular automata. *Transactions of the* IEICE, E 72(6):758–762, 1989.

[106] K. Morita et S. Ueno. Computation-universal models of two-dimensional 16-state reversible automata. IEICE *Transactions on Informations and Systems*, E75-D(1):141–147, Janvier 1992.

[107] J. Myhill. The converse of Moore's garden-of-eden theorem. Dans *Proceedings of the Symposium of Applied Mathematics*, numéro 14, pages 685–686, 1963.

[108] M. Nasu. Local maps inducing surjective global of one-dimensional tesselation automata. *Mathematical System Theory*, 11:327–351, 1978.

[109] M. Nasu. Indecomposable local maps of tessalisation automata. *Mathematical System Theory*, 13:81–93, 1979.

[110] J. von Neumann. *Theory of Self-Reproducing Automata*. Dans [20], 1966.

[111] J. Pedersen. Cellular automata as algebraic systems. *Complex Systems*, 6:237–250, 1992.

[112] D. Perrin. Local maps. Dans C. Choffrut, éditeur, *Automata Networks*, numéro 316 dans Lecture Notes in Computer Science, pages 29–41. Springer-Verlag, 1986.

[113] E. Prisner. Parallel chip firing on digraphs. *Complex Systems*, 1993?

[114] S. Rao Kosaraju. On some open problems in theory of cellular automata. IEEE *Transactions on Computers*, C-23(6):561–565, 1974.

[115] H. Rappetout. *De l'Effraction et du Vol de Vélos dans les Campus*. Cour des Miracles, 1995-96.

[116] C. Reutenauer. *Aspects Mathématiques des Réseaux de Pétri*. Masson, 1989.

[117] D. Richardson. Tessellations with local transformations. *Journal of Computer and System Sciences*, 6:373–388, 1972.

[118] M. Sears. The automorphisms of the shift dynamical systems are relatively scarce. *Mathematical System Theory*, 5:228–231, 1971.

[119] T. Serizawa. Three-state Neumann neighbor cellular automata capable of constructing self-reproducing machines. *Systems and Computers in Japan*, 18(4):33–40, 1987.

[120] A. R. Smith III. Simple computation-universal cellular spaces. *Journal of the Association for Computing Machinery*, 18(3):339–353, 1971.

[121] J. Spencer. Balancing vectors in the max norm. *Combinatorica*, 6(1):55–65, 1986.

[122] R. Subramanian et I. Scherson. An analysis of diffusive load-balancing. Dans ACM *Symposium on Parallel Algorithms and Architecture*, pages 220–225, 1994.

[123] K. Sutner. Linear cellular automata and the garden-of-eden. *Mathematical intelligencer*, 11(2):49–53, 1989.

[124] K. Sutner. A note on Culik-Yu classes. *Complex Systems*, 3:107–115, 1989.

[125] K. Sutner. De Bruijn graphs and linear cellular automata. *Complex Systems*, 5:19–30, 1991.

[126] K. Sutner. On the complexity of finite cellular automata. *Journal of Computer and System Sciences*, 50:87–97, 1995.

[127] S. Takahashi. Self-similarity of linear cellular automata. *Journal of Computer and System Sciences*, 44(1):114–140, 1992.

[128] E. Talbi. Allocation dynamique de processus dans les systèmes distribués et parallèles : État de l'art. Rapport technique 162, LIFL, 1995. In French.

[129] C. Tang et P. Bak. Critical exponents and scaling relations for self-organized critical phenomena. *Physical Review Letters*, 60(23):2347–2350, 1988.

[130] M. Tchuente. Modèles de calcul sur les réseaux d'automates. In French, 1983.

[131] V. Terrier. On real time one-way cellular array. *Theoretical Computer Science*, 141:331–335, 1995.

[132] V. Terrier. Language not recognizable in real time by one-way cellular automata. *Theoretical Computer Science*, 156:281–287, 1996.

[133] T. Toffoli. Computation and construction universality of reversible cellular automata. *Journal of Computer and System Sciences*, 15:213–231, 1977.

[134] T. Toffoli. Reversible computing. Rapport technique MIT/LCS/TM-151, MIT Laboratory for Computer Science, 1980.

[135] T. Toffoli. Bicontinuous extentions of inversible combinatorial functions. *Mathematical System Theory*, 14:13–23, 1981.

[136] T. Toffoli. Informations transport obeying the continuity equation. IBM *Journal of Research and Development*, 32(1):29–36, 1988.

[137] T. Toffoli et N. Margolus. *Cellular Automata Machine - A New Environment for Modeling.* MIT press, Cambridge, MA, 1987.

[138] T. Toffoli et N. Margolus. Invertible cellular automata: A review. *Physica D*, 45:229–253, 1990.

[139] S. Ulam. Random process and transformations. Dans *Proceedings of the International Congress of Mathematics*, numéro 2, pages 264–275, 1952.

[140] H. Umeo, K. Morita, et K. Sugata. Deterministic one-way simulation of two-way real-time cellular automata and its related problems. *Information Processing Letters*, 14(4):158–161, 1982.

[141] G. Vichniac. Simulating physics with cellular automata. *Physica*, 10D:96–115, 1984.

[142] S. Wolfram. Statistical mechanics of cellular automata. *Review of Modern Physics*, 55:601, 1983.

[143] S. Wolfram. Computation theory of cellular automata. *Communications in Mathematical Physics*, 96:15–57, 1984.

[144] S. Wolfram. Universality and complexity in cellular automata. *Physica*, 10D:1–35, 1984.

[145] S. Wolfram. Cryptography with cellular automata. *Proceedings of Crypto '85*, pages 429–432, 1985.

[146] S. Wolfram. Twenty problems in the theory of cellular automata. *Physica Scripta*, T9:170–183, 1985.

[147] S. Wolfram. Cellular automaton fluids 1: Basic theory. *Journal Statistical Physics*, 45:471–526, 1986.

[148] S. Wolfram. *Theory and Applications of Cellular Automata.* World Scientific, 1986.

[149] S. Wolfram. *Cellular Automata and Complexity.* Addison Wesley, 1994.

[150] T. Yaku. The constructibility of a configuration in a cellular automata. *Journal of Computer and System Sciences*, 7:481–496, 1973.

[151] T. Yaku. Inverse and injectivity of parallel relations induced by cellular automata. *American Mathematical Society*, 76:216–220, 1976.

[152] Y.-C. Zhang. Scaling theory of self-organized criticality. *Physical Review Letters*, 63(5):470–473, 1989.

Annexe B

Après la thèse

Dans ce chapitre sont regroupés les développements postérieures de la thèse.

2.1 Automates cellulaires réversibles

Le théorème 39 à été étendu à la dimension 1 et une simulation de tout AP-R par un AC partitionné (au sens de MORITA) réversible. Ceci a été publié dans [43].

La démonstration de la conjecture de TOFFOLI et MARGOLUS sur la simulation des AC-R par les AP-R a été améliorée. En particulier, des simulations en temps 2^d et $d+1$, au lieu de 2^{d+1}, ont été réalisées avec des pavés de taille 6r et $3(d+1)$r au lieu de 4r. Ceci fait l'objet d'un rapport de recherche du labri et d'une soumission [37].

2.2 Tas de sable

Les résultas de cette partie ont fait l'objet de publication [39] et de soumission [42].

Résumé

Cette thèse s'intéresse dans un premier temps aux automates cellulaires réversibles, et dans un second temps aux tas de sable linéaires.

Nous construisons diverses simulations reliant les automates cellulaires aux automates à partitions, en particulier celle des automates cellulaires réversibles par les automates à partitions réversibles, ce qui était une conjecture depuis 1990. Par des constructions successives, nous montrons que le « Billiard ball model » de Toffoli et Margolus est capable de simuler tous les automates à partitions réversibles de dimension 2. En rassemblant ces résultats, nous montrons qu'il existe des automates cellulaires réversibles capables de simuler tous les automates cellulaires réversibles de même dimension.

Dans un espace linéaire, « Tas de sable » et « Chip firing game » sont équivalents. Nous portons notre attention sur le cas où les grains tombent un à un. Des motifs délimités par des signaux apparaissent au sein des configurations engendrées. Nous étudions la dynamique du système et démontrons un équivalent asymptotique. Nous étendons nos méthodes et nos résultats à d'autres types de configurations initiales. Dans chaque cas étudié, le temps parallèle est inférieur au temps séquentiel dans un rapport de l'ordre du nombre de piles mises en œuvre.

Mots-Clés

Automates Cellulaires, Automates à Partitions, Réversibilité, Simulation, Tas de Sable Linéaires, Chip Firing Game et Billiard Ball Model.

Abstract

This thesis deals with two topics. The first is the reversible simulation of cellular automata. The second is the evolution in linear space of the Sand Pile Model and of the Firing Chip Game.

We make various simulations that link cellular automata and partitioning cellular automata. In particular, we prove a 1990 conjecture that reversible cellular automata can be simulated by reversible partitioning automata. We show that the Billiard Ball Model of Toffoli and Margolus is able to simulate any reversible partitioning automaton of dimension two. It follows that there are reversible cellular automata able to simulate any reversible cellular automaton of the same dimension.

In linear space, the Sand Pile Model and the Chip Firing Game are equivalent. We especially study the case of grain dripping. Patterns delimited by signals appear. We study their interactions and make asymptotic approximations. Our methods and results are applied to other cases. Over the cases studied, massive parallelism is achieved: sequential and parallel times differ by a factor corresponding to the number of active piles.

Key-Words

Cellular Automata, Partitioning Automata, Reversibility, Simulation, Sand Pile Model, Chip Firing Game and Billiard Ball Model.

Resumen

Esta tesis toca dos temas : en primer lugar, se interesa por la simulación de los autómatas celulares reversibles, y en segundo lugar, por la dinámica de los modelos lineales de Pilas de arena y de Tiro de fichas.

Construimos diversas simulaciones que vinculan los autómatas celulares y los autómatas con particiones. En particular, se desmuestra una conjeatura de 1990 sobre la simulación de automatas celulares reversibles. Mediante construciones sucecivas, mostramos que el modelo de la bola de billar de Toffoli y Margolus es capaz de simular todos los autómates con particiones reversibles de dimensión dos. Con estos resultados, establecemos que existen autómates celulares reversibles capaces de simular cualquier autómata celular reversible de la misma dimensión.

En un espacio lineal, Pilas de arena y Tiro de fichas son equivalentes. Consideramos atentamente el caso de los granos de arena que caen uno por uno ; este goteo deja aparecer unos motivos delimitados por signos. Estudiamos la dinámica del systema y elaboramos unas aproximaciones asinptóticas. Aplicamos nuestros métododos y resultados a otros tipos de configuraciones iniciales. En cada caso estudiado, el tiempo paralelo es inferior al tiempo secuencial en una proporción del orden del número de pilas utilizadas.

Palabras Claves

Autómatas Celulares, Autómatas con Particiones, Reversibilidad, Simulación, Modelo de Pilas de Arena, Tiro de Fichas y Modelo de la Bola de Billar.