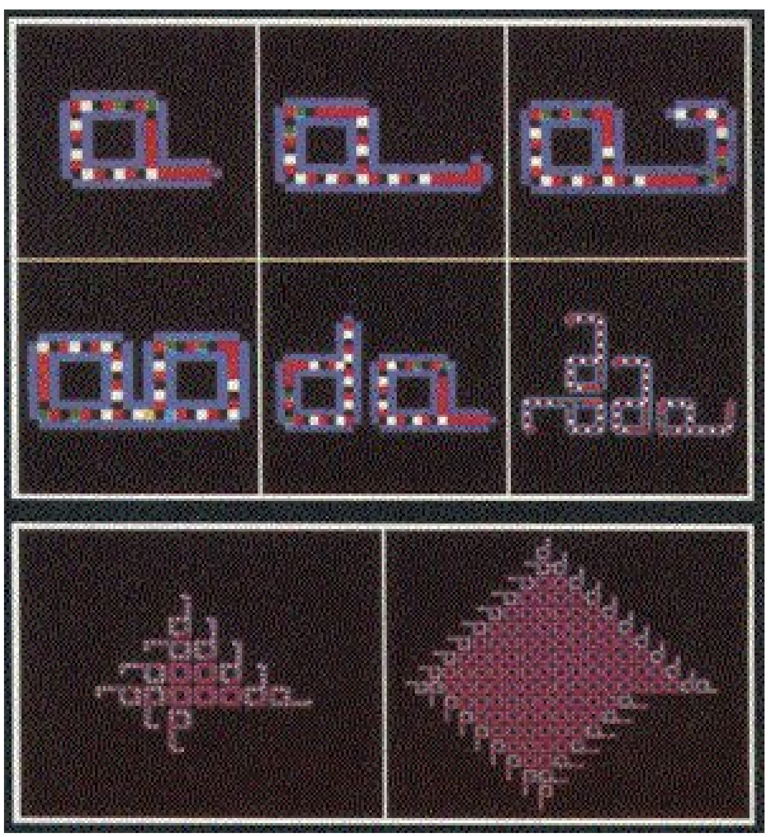
PITRAT Colin

Automates cellulaires et fractales



I – Présentation des automates cellulaires

1- Introduction

L'idée des automates cellulaires est à peu près contemporaine de celle des ordinateurs. Les premières recherches sont dues à John Von Neumann et Stanislas Ulam au début des années 1950. Von Neumann et Ulam voulaient réaliser un système simple, capable de se reproduire à la manière d'un organisme vivant, ce qui donna naissance aux automates cellulaires.

Un automate cellulaire est une zone d'espace de dimension quelconque (1 ou 2 en général pour des raisons évidentes de visualisation, parfois 3), quadrillé et régit par un temps discret. Chaque case représente une cellule, et est, à un moment donné, dans un certain état. Le nombre d'état est généralement fini. L'évolution de l'automate du temps t au temps $t+1$ est déterminée par un certain nombre de règles simples en nombre finit qui tiennent compte des états des cellules voisines et de l'état de la cellule en question au temps t.

Un automate cellulaire répond donc à quatre propriétés:

1 – **Localité** : l'évolution d'une cellule sur un certain intervalle de temps ne dépend que d'un voisinage limité de cette cellule. L'état des cellules lointaines n'intervient qu'au bout d'un temps suffisament long

2 – **Invariance temporelle** : les rêgles d'évolution ne dépendent pas du temps. Si l'automate est dans un certain état au temps T_1, et qu'il retrouve cet état au temps T_2, les états $T_1 + t$ et $T_2 + t$ seront identiques quel que soit t.

3 – **Invariance spatiale** : les rêgles d'évolution ne dépendent pas de la position de la cellule dans l'automate.

4 – **Déterminisme** : L'évolution d'un automate cellulaire est entièrement déterminée par la donnée de son état initial. Le hasard n'intervient pas dans cette évolution.

2- Propriétés supplémentaires

De l'invariance temporelle peut découler la notion de cycle temporelle. Ainsi, si une configuration donnée de cellule à l'instant T se retrouve à l'instant T + t, elle se retrouvera aussi à tous les instants T + nt, avec n > 0.

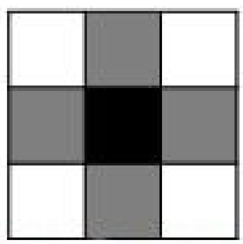
De l'invariance temporelle et spatiale peut découler la notion de “projectile” : une configuration de cellule au temps T évolue pour se retrouver à l'identique à un temps T + t décalée d'un vecteur déplacement **D**. On retrouvera cette même configuration à tous les temps T + nt décalée d'un vecteur déplacement n**D**, avec n > 0.

Si l'on étudie l'évolution à long terme de divers automates cellulaires, on peut alors définir différentes catégories selon l'état vers lequel ils tendent :

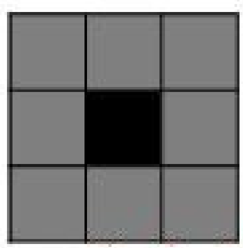
- Extinction / Envahissement : l'ensemble des cellules prend le même état.
- Cyclique : L'état évolue selon un cycle de longueur variable. Ce cycle peut-être très long si il y'a plusieurs cycles locaux de longueur différentes, car la longueur du cycle global est alors le PPCM de toutes les longueurs de cycle. Il existe un grand nombre de sous catégories pour les automates cycliques (glissants, localement cycliques ...)
- Instable : L'automate évolue sans connaître de cycle, et sans se stabiliser. Cet état n'est pas facilement déterminable : il faudrait faire évoluer le système sur un temps infini en enregistrant l'ensemble des évolutions pour s'assurer qu'il n'existe pas de cycle.
- Chaotique : L'évolution de l'automate tend, selon l'état initial, vers l'un des états précédents. Cet catégorie est bien sûr la plus interessante à étudier.

3- Différents voisinages

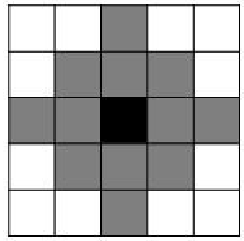
- Voisinage de Von Neumann



- Voisinage de Moore



- Voisinage de Moore étendu



II – Programme utilisé

1- Automates cellulaires concernés

Le programme utilisé pour simuler les automates cellulaires a été écrit en C à l'aide de la librairie SDL. Il permet de simuler un grand nombre d'automates cellulaires, et apporte même certains suppléments à la définition d'automate cellulaire présentée ci-dessus. Les automates pouvant être simulés sont à 1 ou 2 dimensions.

Restrictions / Caractéristiques du programme :
- Le nombre d'états possible dans un automate n'est pas limité
- Un état n'a qu'un seul successeur possible autre que lui-même : si une cellule évolue, ce ne peut être que dans l'état suivant
- Les transitions peuvent être dotés d'une probabilité propre à chaque état du voisinage
- On ne considère qu'un type d'état dans le voisinage pour décider de la transition

1 Dimension :
- Le voisinage considéré est constitué de la cellule elle même et de ses deux voisines.
- Le voisinage est orienté : il y a une différenciation entre la droite et la gauche
- L'évolution temporelle est représentée verticalement

2 Dimensions :
- Le voisinage considéré peut-être de Von-Neumann, de Moore ou de Moore étendu.
- Le voisinage n'est pas orienté : on ne fait pas la différence entre les cellules du voisinage, on ne fait que compter le nombre de cellules dans l'état considéré.

2- Utilisation du programme

Le programme s'utilise en ligne de commande. On utilise la syntaxe suivante :

```
> ./automates vie -f -i vie.world
```

ou “vie” est le fichier décrivant l'automate, “-f “ indique d'executer l'automate en plein écran, “-i” permet de spécifier un état initial (ici le fichier vie.world). Si l'on ne spécifie pas de fichier, l'état initial est généré aléatoirement à partir des probabilités fournies dans le fichier décrivant l'automate.

La touche F1 permet de sauvegarder l'état courant de l'automate. La touche Espace permet de faire avancer l'automate d'une étape. La touche Entrée permet de lancer l'automate, tandis que la touche backspace l'arrête. Les flêches droite et gauche permettent de modifier l'état sélectionné, et la touche Alt de montrer / cacher quel est l'état sélectionné. On peut, à l'aide de la souris, changer l'état d'une cellule vers l'état sélectionné.

Le programme est de plus doté d'un calcul de la dimension fractale de l'état sélectionné de l'automate par la méthode des boites. Il faut pour celà appuyer sur F2. On affiche alors le résultat pour des boites de largeur de 1 à 32 cellules dans un diagramme Log/Log. La touche espace permet d'effectuer une régression linéaire. La dimension calculée est alors affichée dans la ligne de commande

3- Fonctionnement du programme

Fichier de description de l'automate :

Le format du fichier décrivant l'automate est le suivant :

largeur hauteur
nombre d'états
voisinage
couleur de l'état 1 (triplet)
...
couleur de l'état n (triplet)
nom de l'état 1 (chaine de caractère)
...
nom de l'état n (chaine de caractère)
probabilité initiale de l'état 1
...
probabilité initiale de l'état n
loi de transition pour l'état 1
...
loi de transition pour l'état n

Une loi de transition est décrite sous la forme suivante :
e_1 e_2 p_1 p_2 p_3 p_4 p_5 p_6 p_7 p_8 p_9 p_{10} p_{11} p_{12}

ou
e_1 représente l'état considéré dans le voisinage
e_2 représente l'état suivant si la transition à lieu
p_i représente la probabilité (en centièmes de %) de transition si il y a *i* cellules dans l'état e_1 dans le voisinage

Calcul de la dimension :

Pour calculer la dimension, on divise l'automate en zones carrées de tailles égales, puis on compte le nombre de zones comprenant une cellule dans l'état sélectionné. On reproduit l'opération pour des largeures de coté variant de 1 à 32 cellules. On note alors x_k le logarithme de k, et y_k le logarithme du nombre de zones actives pour la largeure de zone k. On a alors :

$$b=\frac{\sum_{k=1}^{n} x_k^2 \sum_{k=1}^{n} y_k - \sum_{k=1}^{n} x_k \sum_{k=1}^{n} x_k y_k}{n \sum_{k=1}^{n} x_k^2 - (\sum_{k=1}^{n} x_k)^2}$$

$$a=\frac{n \sum_{k=1}^{n} x_k y_k - \sum_{k=1}^{n} x_k \sum_{k=1}^{n} y_k}{n \sum_{k=1}^{n} x_k^2 - (\sum_{k=1}^{n} x_k)^2}$$

On a alors la dimension fractale D :

$$D = -a$$

La dimension calculée dépend donc de l'état sélectionné.

III – Exemples d'automates

1- Tamis de Sierpinski

C'est un automate à une dimension à deux états. On part d'une seule cellule vivante entourée de cellules mortes. Une cellule devient vivante si la cellule à sa gauche ou la cellule à sa droite est vivante, mais pas les deux. Elle devient/reste morte sinon.

Cet automate génère un tamis de Sierpinski. Le calcul de la dimension sur cet automate donne un résultat faux dû à la trop importante régularité. La méthode des boites n'est pas adaptée dans ce cas là.

2- Le jeu de la vie

Le but est ici de montrer que la propriété de localité des automates cellulaires n'empêche pas l'apparition de phénomènes chaotiques.

Le jeu de la vie est sans doute l'automate cellulaire le plus connu. C'est un automate en deux dimensions utilisant un voisinage de Moore et possédant deux états : morte ou vivante. Une cellule morte devient vivante si elle est entourée par exactement 3 cellules vivantes. Une cellule vivante survie si elle est entourée par 2 ou 3 cellules vivantes. Un moyen simple de visualiser une évolution chaotique est de partir d'un motif simple (par exemple une ligne verticale). L'évolution est alors très simple, et reste “verticale” (à un temps donné, toutes les lignes sont identiques). Mais il suffit de placer une cellule vivante près de cette ligne pour que l'évolution soit grandement perturbée. Ce n'est cependant pas le cas si la cellule est placée loin de la ligne : elle meurt rapidement. Le phénomène chaotique est donc limité à quelques états critiques.

On a ici à faire à un système discret, aussi bien au niveau spatial que temporel, ce qui explique que l'évolution puisse différer autant pour des états pourtant proches.

3 – Le feu de foret

C'est un automate à 2 dimensions, comportant 4 états : terre, arbre, feu, cendre. Il utilise un voisinage de Von Neumann. Un arbre prend feu si il y'a une cellule en feu dans son voisinage. Un feu se transforme en cendre. Le but est d'étudier la propagation d'un feu et d'étudier la zone de cendre (arbres ayant brulés). Ce phénomène se rapproche de la percolation, mais est un peu différent. Dans la percolation, on étudie généralement la propagation dans une direction, en partant d'une ligne perpendiculaire à cette direction. Dans le cas présent, le départ se fait d'un seul pixel. Les chances que l'on se trouve dans une zone isolée tout en étant au dessus du seuil de percolation sont grandes. Il faut donc écarter ces cas là de l'étude.
Malgré ce risque, on retrouve assez facilement le seuil de percolation attendu (0,59). On peut alors calculer la dimension de l'amas de cendre aux alentours du seuil de percolation, et trouver le résultat attendu de environ 1,8 – 1,9.