

# Introduction aux automates cellulaires\*

Antoine Delignat-Lavaud

## 1 Introduction aux automates cellulaires

**Définition 1 (Automate cellulaire)** Une automate cellulaire  $\mathcal{A} = (d, \mathbb{S}, \mathcal{V}, \delta)$  est la donnée de :

1. Sa dimension  $d \in \mathbb{N}^*$ , qui est la dimension de son réseau  $\mathbb{Z}^d$
2. Un ensemble fini d'états  $\mathbb{S}$ , appelé alphabet
3. Son voisinage  $\mathcal{V} = (v_1, v_2, \dots, v_n) \in (\mathbb{Z}^d)^n$
4. Sa règle locale de transition  $\delta : \mathbb{S}^n \rightarrow \mathbb{S}$

Une *configuration* de l'automate  $\mathcal{A}$  est une application de  $\mathbb{Z}^d$  dans  $\mathbb{S}$  : on associe à chaque cellule de l'automate un état. On note  $\mathfrak{C}$  l'ensemble des configurations de  $\mathcal{A}$ .

Un automate cellulaire peut être considéré comme un système dynamique à temps et espace discrets en munissant  $\mathfrak{C}$  de la fonction globale de transition de l'automate  $\mathcal{A}$ , c'est-à-dire la fonction d'évolution temporelle de la configuration. Cette fonction  $\mathfrak{F} : \mathfrak{C} \rightarrow \mathfrak{C}$  est définie cellule par cellule par :

$$\forall \mathbf{c} \in \mathfrak{C}, \forall \gamma \in \mathbb{Z}^d, \mathfrak{F}(\mathbf{c})(\gamma) = \delta(\mathbf{c}(\gamma + v_1), \dots, \mathbf{c}(\gamma + v_n))$$

L'automate cellulaire décrit alors le système dynamique

$$\begin{cases} \mathbf{c}_0 \in \mathfrak{C} \\ \forall n \in \mathbb{N}, \mathbf{c}_{n+1} = \mathfrak{F}(\mathbf{c}_n) \end{cases}$$

**Définition 2** On appelle *orbite* d'une configuration initiale  $\mathbf{c} \in \mathfrak{C}$  l'ensemble des configurations prises par l'automate depuis l'état initial  $\mathbf{c}$  :

$$\mathcal{O}(\mathbf{c}) = \{\mathfrak{F}^k(\mathbf{c}), k \in \mathbb{N}\}$$

## 2 Automates cellulaires élémentaires

**Exemple 1 (ACE)** On appelle *automate cellulaire élémentaire* tout automate  $\mathcal{A} = (d, \mathbb{S}, \mathcal{V}, \delta)$  tel que :

1.  $d = 1$  : Les cellules de l'automates correspondent à des entiers
2.  $\mathbb{S} = \{\square, \blacksquare\}$  : les cellules n'ont que deux états possibles
3.  $\mathcal{V} = \{-1, 0, 1\}$  : l'état au temps  $t + 1$  d'une cellule  $\gamma$  dépend de l'état au temps  $t$  de  $\gamma$  et de ses deux voisines immédiates  $\gamma + 1$  et  $\gamma - 1$

---

\*Présentation donnée au lycée Camille Jullian

La règle locale de transition d'un automate élémentaire est une fonction de  $\mathbb{S}^3$  dans  $\mathbb{S}$ . Comme  $\mathbb{S}$  n'a que deux éléments, il n'existe que  $2^8 = 256$  règles locales de transition possibles. On peut donner  $\delta$  sous forme d'un tableau :

■■■	■■□	■□■	■□□	□■■	□■□	□□■	□□□
□	■	□	■	■	□	■	□

Chaque règle  $\delta$  peut être vue comme l'écriture binaire d'un entier entre 0 et 255. Ainsi, le tableau précédent correspond à la règle élémentaire  $\overline{01011010}^2 = 90$ .

Les cellules d'un ACE se répartissent sur  $\mathbb{Z}$ . On peut donc représenter l'évolution des configurations de l'automate dans  $\mathbb{Z}^2$  :

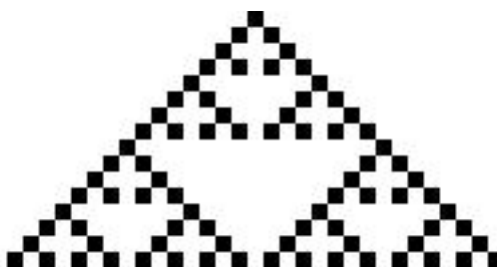


Diagramme temporel de l'ACE 90 ( $t = \downarrow$ )

### 3 Implémentation des ACE sous Maple

```

1 delta := proc(num, a, b, c)
2   local bin, i;
3   bin := convert(num, 'base', 2);
4   i := c+2*b+4*a+1;
5   return 'if'(i <= nops(bin), bin[i], 0);
6 end;
7
8 ac_elementaire := proc(num, e_initial, t)
9   local i, j, n, l, n_l, reseau;
10
11   n := nops(e_initial);
12   l := e_initial;
13   n_l := array(1..n);
14   reseau := array(1..t, 1..n);
15
16   for j to t do
17     for i to n do
18       reseau[j, i] := ligne[i];
19     end;
20
21     n_l[1] := delta(num, 0, l[1], l[2]);
22     for i from 2 to n-1 do
23       n_l[i] := delta(num, l[i-1], l[i], l[i+1]);
24     end;
25     n_l[n] := delta(num, l[n-1], l[n], 0);
26
27     l := [seq(n_l[i], i=1..n)];
28   end;
29   return reseau;
30 end;
31

```

### 4 Topologie de l'espace des configurations

**Théorème 1** *L'ensemble  $\mathcal{C}$  des configurations d'un automate cellulaire, muni de la distance*

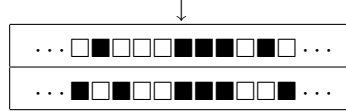
$$d(c, c') = 2^{-k} \quad \text{avec} \quad k = \min_{z \in \mathbb{Z}^d} \{ \|z\|_\infty, c(z) \neq c'(z) \}$$

où  $\|z\|_\infty = \|(z_1, \dots, z_d)\|_\infty = \max |z_i|$  est un espace métrique.

**Preuve :** Il est clair que  $\mathfrak{d}(\mathfrak{c}, \mathfrak{c}') = \mathfrak{d}(\mathfrak{c}', \mathfrak{c})$ .  
 $\mathfrak{d}(\mathfrak{c}, \mathfrak{c}') = 0 \iff k = \infty$  c'est à dire lorsque  $\forall z \in \mathbb{Z}^d, \mathfrak{c}(z) = \mathfrak{c}'(z)$  soit  $\mathfrak{c} = \mathfrak{c}'$ .  
 Enfin, en posant  $\Delta(\mathfrak{c}, \mathfrak{c}') = \{z \in \mathbb{Z}^d, \mathfrak{c}(z) \neq \mathfrak{c}'(z)\}$ , on a  $\Delta(\mathfrak{c}, \mathfrak{c}'') \subset \Delta(\mathfrak{c}, \mathfrak{c}') \cup \Delta(\mathfrak{c}', \mathfrak{c}'')$ , d'où l'inégalité triangulaire.  $\diamond$

En particulier, pour un automate de dimension 1, la distance de Cantor est donnée par :

$$\mathfrak{d}(\mathfrak{c}, \mathfrak{c}') = 2^{-\min\{|i|, \mathfrak{c}(i) \neq \mathfrak{c}'(i)\}}$$



Sur cette exemple, on  $\mathfrak{d}(\mathfrak{c}_1, \mathfrak{c}_2) = 2^{-3}$ .

À l'aide de cette topologie, on va pouvoir étudier les comportements dynamiques de  $\mathfrak{F}$ .

**Définition 3 (Equicontinuité)** On dit que  $\mathfrak{F}$  est équicontinue en  $\mathfrak{c} \in \mathfrak{C}$  si

$$\forall \varepsilon > 0, \exists \eta > 0, \forall \mathfrak{c}' \in B(\mathfrak{c}, \eta), \forall n \in \mathbb{N}^*, \mathfrak{d}(\mathfrak{F}^n(\mathfrak{c}), \mathfrak{F}^n(\mathfrak{c}')) < \varepsilon$$

On dit de plus que  $\mathfrak{F}$  est uniformément équicontinue si elle est équicontinue pour tout  $x \in \mathfrak{C}$  et que  $\eta$  convient pour tous les  $x$ .

L'équicontinuité ponctuelle affirme l'existence d'un voisinage où les orbites sont stables par perturbation de l'état initial. La notion d'équicontinuité uniforme traduit donc une très forte stabilité dynamique de  $\mathfrak{F}$ . Le théorème suivant va nous permettre de caractériser l'équicontinuité uniforme d'un AC.

## 5 Caractérisation de l'équicontinuité

**Définition 4** On dit que la fonction globale de transition  $\mathfrak{F}$  d'un automate cellulaire est ultimement périodique si :

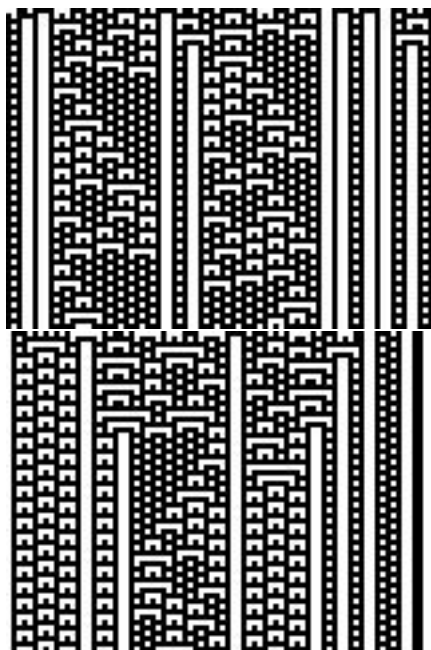
$$\exists n_0 \in \mathbb{N}, \exists p \in \mathbb{N}^*, \forall n \geq n_0, \mathfrak{F}^{n+p} = \mathfrak{F}^n$$

**Théorème 2 (Caractérisation des automates équicontinus)**  $\mathfrak{F}$  est équicontinue si, et seulement si,  $\mathfrak{F}$  est ultimement périodique.

**Preuve :** Pour simplifier, l'automate est supposé de dimension 1. Comme  $\mathfrak{F}$  est uniformément équicontinu, il existe des entiers  $k$  et  $i$  tels que

$$\forall n \in \mathbb{N}, \forall \mathfrak{c} \in \mathfrak{C}, \forall \mathfrak{c}' \in \mathcal{B}(\mathfrak{c}, 2^{-k}), (\mathfrak{F}^n \mathfrak{c})_i = \mathfrak{F}^n(\mathfrak{c}')_i$$

Or, remarquons que toute configuration  $\mathfrak{r}$  dont les  $k+1$  premiers états sont fixés peut se prolonger en une configuration périodique dans l'espace (en répétant le motif initial). Les configuration  $(\mathfrak{F}^n(\mathfrak{r}))_{n \in \mathbb{N}}$  alors aussi des états périodiques sur  $\mathbb{Z}$ , de période inférieure. Or, les configurations  $k+1$ -périodiques dans l'espace sont en nombre fini, car l'alphabet des états est fini. Nécessairement,  $(\mathfrak{F}^n(\mathfrak{r}))_{n \in \mathbb{N}}$  est ultimement périodique. Ceci est vrai en particulier pour  $(\mathfrak{F}^n(\mathfrak{c}'))_n$  en prenant la plus grande période (temporelle, sur  $\mathfrak{F}$ ) pour les  $\hat{\mathfrak{c}}$  ainsi prolongés.  $\diamond$



ACE 109, deux états initiaux aléatoires ( $t = \downarrow$ ).  
Le système évolue rapidement vers des configurations périodiques.

## 6 Classification dynamique

**Définition 5**  $\mathfrak{F}$  est dite sensible aux conditions initiales si :

$$\exists \varepsilon > 0, \forall c \in \mathcal{C}, \forall \eta > 0, \exists c' \in B(c, \eta), \exists n \in \mathbb{N}^*, \mathfrak{d}(\mathfrak{F}^n(c), \mathfrak{F}^n(c')) \geq \varepsilon$$

Il est important de noter que les propriétés dynamiques dépendent de la métrique choisie pour les configurations. La métrique de CANTOR a pour inconvénient de privilégier une cellule particulière de l'automate, l'origine. En particulier, certains automates qui semblent stables, comme le décalage de toutes les cellules, est sensible aux conditions initiales.





ACE 118, deux états initiaux aléatoires ( $t = \downarrow$ )  
Automate ultimement périodique mais seulement équicontinu « à décalage près ».



ACE 18, deux états initiaux aléatoires ( $t = \downarrow$ )  
Exemple d'automate non équicontinu : les configurations initiales sont identiques à 90%.

## 7 Modélisation par automates cellulaires

Les automates cellulaires permettent de modéliser des phénomènes complexes à partir de règles locales simples, ce qui en fait un outil de choix pour la physique.

### 7.1 Equation de la chaleur

Pour cette première application, on généralise la notion d'automate cellulaire à un alphabet discret. On va ainsi transformer l'équation locale de la chaleur en règle locale d'automate cellulaire pour la température. Le but est de comparer la simplicité du modèle obtenu avec la méthode de résolution de l'équation aux dérivées partielles.

On rappelle que pour un faible gradient de températures dans un milieu homogène de conductivité  $\lambda$ , de masse volumique  $\rho$  et de capacité thermique massique  $c$ , l'équation de la chaleur s'écrit :

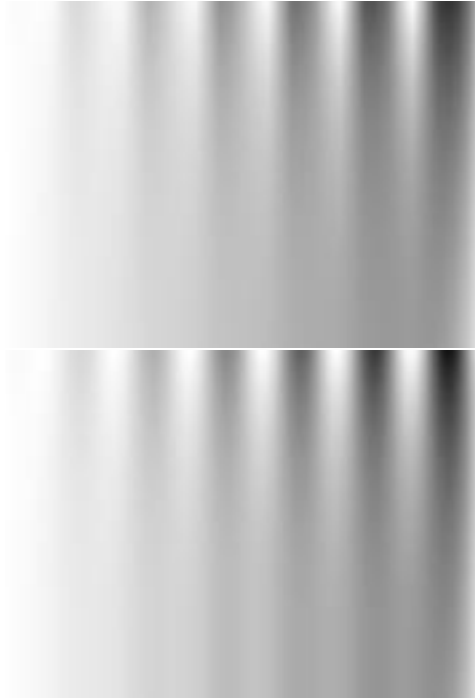
$$\frac{\partial T(x,t)}{\partial t} = \frac{\lambda}{\rho c} \Delta T$$

ce qui par la méthode des différences finies se discrétise en :

$$\frac{T_i^{k+1} - T_i^k}{\tau} = \frac{\lambda}{\rho c} \frac{T_{i-1}^k - 2T_i^k + T_{i+1}^k}{\delta^2}$$

où  $\tau$  est le pas temporel et  $\delta$  le pas spacial du modèle, pour un milieu à une dimension modélisée par le réseau  $(T_i)_{i \in \mathbb{Z}}$ .

Le profil de l'état initial du matériau est  $T(x) = x \cos(\alpha x)^2$ , avec pour conditions aux limites une température nulle aux deux extrêmités du milieu.



Solution numérique (à gauche), modèle automate cellulaire.

```

1  edp := pdsolve(
2    diff(T(x,t),t)=diff(T(x,t),x,x),
3    {T(x,1)=x/180*cos(13*Pi*x/360)^2,
4    T(0,t)=0, T(180,t)=0},
5    numeric);
6
7  img := ImageTools[Create](130,180,channels=1);
8  for i from 1 to 130 do
9    f := edp:-value(t=0.7*i);
10   for j from 1 to 180 do
11     img[i,j] := 1-rhs(f(j)[3]);
12   end;
13 end;
14
15 img2 := ImageTools[Create](130,180,channels=1);
16 img2[1] := [seq(evalf(
17   i/180*cos(13*Pi*i/360)^2),
18   i=1..n)];
19 for j from 2 to p do
20   img2[j,1] := 0; img2[j,n] := 0;
21   for i from 2 to n-1 do
22     img2[j,i] := (img2[j-1,i-1]+img2[j-1,i+1])/2;
23   end;
24 end;

```