

Distributing the Heat Equation

Marc CHEVALIER
Thomas PELLISSIER TANON

November 13, 2014

1 Heat equation

$$\frac{\partial x}{\partial t} = \nabla^2 x$$

2 Cellular automata

Question 1. There is N^2 cells in $\llbracket 0, N - 1 \rrbracket^2$. Therefore, tN^2 applications of the function δ are necessary to compute X^t on $\llbracket 0, N - 1 \rrbracket^2$.

Question 2. We use a processor per cell. Each cell store its own state and the state of its 8 neighbours. Each cell compute its next state and send this information to its neighbours.

But, the diagonal neighbours are not linked directly. To communicate the new state to these cells, we resend each state we receive from below or above to the left and right neighbours. At the end of the communications, each cell has updated the states of its neighbours.

Si tu n'es pas d'accord mineurement sens toi libre de rectifier la chose(même si c'est en vrac à coté, je m'en occuperai). Si c'est majeur... ben faudra voir.

Question 3. Implem bloquante. Je l'analyserai dans le détail dès que possible.

3 Average automata

Question 4. In this case, we don't need the state of the diagonal cells. So, we have no need of the second step of communication.

Question 5. Let $A = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{pmatrix}$ and $B = \begin{pmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \end{pmatrix}$.

$$\begin{aligned} \delta(A) + \lambda \delta(B) &= (1-p)a_{2,2} + p \frac{a_{2,1} + a_{1,2} + a_{3,2} + a_{2,3}}{4} + \lambda \left((1-p)b_{2,2} + p \frac{b_{2,1} + b_{1,2} + b_{3,2} + b_{2,3}}{4} \right) \\ &= (1-p)(a_{2,2} + \lambda b_{2,2}) + p \frac{a_{2,1} + a_{1,2} + a_{3,2} + a_{2,3} + \lambda(b_{2,1} + b_{1,2} + b_{3,2} + b_{2,3})}{4} \\ &= \delta(A + \lambda B) \end{aligned}$$

Therefore $\delta \in \mathcal{L}(\mathcal{M}_3(\mathbb{R}), \mathbb{R})$

Consequently, δ^\dagger is a linear mapping, since it's just N^2 simultaneous application of δ .

More formally, let $(\mathcal{G}, \mathcal{H}, \lambda) \in \mathcal{M}_N(\mathbb{R})^2 \times \mathbb{R}$.

$$\begin{aligned}\delta^\dagger(\mathcal{G} + \lambda \cdot \mathcal{H}) &= (\delta(v_{i,j}(\mathcal{G} + \lambda \cdot \mathcal{H})))_{i,j} \\ &= (\delta(v_{i,j}(\mathcal{G}) + \lambda v_{i,j}(\mathcal{H})))_{i,j} \\ &= \delta^\dagger(\mathcal{G}) + \lambda \delta^\dagger(\mathcal{H})\end{aligned}$$

where $v_{i,j}(\mathcal{G}) \in \mathcal{M}_3(\mathbb{R})$ is the neighbours of the cell (i, j) in \mathcal{G} . This function is clearly linear. Therefore δ^\dagger is linear.

Let \mathcal{G} an initial state and $t \in \mathbb{N}^*$

As δ^\dagger is linear, we can decompose \mathcal{G} as a sum of N^2 matrices where all coefficient except one is equal to 0.

$$\mathcal{G} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \mathcal{G}^{(i,j)}$$

$$\text{where } (\mathcal{G}^{(i,j)})_{k,l} = \mathcal{G}_{i,j} E_{i,j} = \begin{cases} \mathcal{G}_{i,j} & \text{if } (i, j) = (k, l) \\ 0 & \text{else} \end{cases}$$

$$\begin{aligned}\delta^\dagger(\mathcal{G}) &= \sum_{i=0}^{N-1} \sum_{j=1}^{N-1} \delta^{+2t}(\mathcal{G}^{(i,j)}) \\ &= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \mathcal{G}_{i,j} \delta^{+2t}(E_{i,j})\end{aligned}$$

Moreover, we just have to compute $\delta^{+2t}(E_{1,1})$ and we obtain all $\delta^{+2t}(E_{i,j})$ by toric translation. These transformation only depend on N which is assumed to be constant. So, we just need $\delta^{+2t}(E_{1,1})$ in time $\log(t)$. Particularly, if we have $\delta^{+t}(E_{1,1})$, we can compute $\delta^{+2t}(E_{1,1})$ in time $\log(t)$ in constant time. We conclude that

$$T(2t) = T(t) + \mathcal{O}(1)$$

where $T(t)$ is the computation time of $\delta^{+t}(E_{1,1})$. According with the master theorem

$$T(t) = \log(t)$$

Finally, when we have $\delta^{+2t}(E_{1,1})$, we can compute $\delta^{+2t}(\mathcal{G})$ in constant time.

J'aimerais aussi analyser l'implem plutôt qu'un modèle mathématique légèrement abstrait