

Distributing the Heat Equation

Marc CHEVALIER
Thomas PELLISSIER TANON

December 7, 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.

Question 3. There is $4N^2$ communication for each step and need a time $\mathcal{O}(1)$.

On a non-toric grid, we need to communicate the state of the cells on the edges to the other side. We need $(N - 2)N$ additional communications and a total time of $\mathcal{O}(N)$.

On a ring, we need a time $\mathcal{O}(N^2)$ and $\mathcal{O}(N^2)$ communications.

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.

To be closer to the abstract algorithm we assume we have one MPI process per cell.

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.

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

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

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

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 $\delta^{+t}(E_{1,1})$. If we want to compute the $2t + 1^{\text{th}}$ generation, we just compute a iteration as previously and we use linearity to compute the $2t^{\text{th}}$ generation.

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.

If we do not consider N as a constant, we have

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

So, we conclude

$$T(t) = N^2 \log(t)$$

While the complexity in the general case is

$$T(t) = N^2 \cdot t$$

Question 6. In this algorithm, each processor keep the value of its cells. Each value is communicated to the other by broadcast. We do N^2 broadcast each time we double the rank of computed generation ie. N^4 communications.

So, we have $\mathcal{O}(N^4 \log(t))$ communications.

Question 7. We use the sums and translations explained at the question 5. Each cell need the state of all other cells to update its state, we need N^4 communications. The computation space is always $\mathcal{O}(N^2)$ and the time is $\mathcal{O}(|X^0| N^2)$

Question 8. Instead of reusing question 4 to compute Z^t we have used the algorithm described in question 6. Again we assume we have one processor per cell.

Question 9. For all $c \in \mathbb{R}$,

$$U_c^b : \llbracket 0, N-1 \rrbracket^d \rightarrow \mathbb{R} \\ x \mapsto c$$

and

$$V_c : \mathbb{Z}^d \rightarrow \mathbb{R} \\ x \mapsto c$$

are fixed points of δ^+ since the weighted mean of 5 values equal to c is c .

4 Thermal reservoirs

Question 10. With one constant C_k , the function U_k is a fixed point.

With two constant, on a grid of size 2

C_k	$\frac{k+l}{2}$
$\frac{k+l}{2}$	C_l

is a fixed point. Indeed

$$(1-p)\frac{k+l}{2} + p\frac{2k+2l}{4} = (1-p)\frac{k+l}{2} + p\frac{k+l}{2} = \frac{k+l}{2}$$

When there is a constant C_k , the limit of (X^t) is U_k . Indeed, as we work on a compact space, there is a minimum. If this minimum is different to k , there is a cell which has at least a neighbour greater and the other have the same value. Then, at the next generation, the value of this cell increase. Then this configuration were not a fixed point.

When there is no constant, the limit of (X^t) is $U_{\frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} X_{i,j}}$. To do this proof, we use the previous prove and the conservation of the sum of the values.

Question 11. δ^+ is no more linear when we work with a grid with constants. For instance, it is obvious if each cell is a constant, there is no evolution.

Question 12.

Lemma 1. *For all grid X , there is a unique fixed point l_X with the same constants.*

Proof. This is the difficult part. We can probably use the continuity of the uniform norm between a generation and the next one. □

Lemma 2. *For all grid X^0 , $(X^t) \rightarrow l_{X^0}$.*

Proof. We just need to prove that $(X^t - l_{X^0}) \rightarrow 0$. Indeed, there are only constant C_0 .

We can easily show that $\|(X^t - l_{X^0})\|_\infty \rightarrow 0$. □

Question 14. The graphic interface is available for both `sparce` and `constants` executables using the `-g` option.