Probabilistic Calculation

LUIS RODRIGUEZ DOMINGO

May 2024

1 Introduction

In order to achieve that each cell in the grid is processed individually by a GPU-thread, we have to define the grid and the set of rules of the CA that will provide it.

We have used two grids, both of them of the same size and they have being initialized at the beginning as a squared grid. One of them saved the current state and the second saved the state of the next step. I mean, to avoid race conditions where a thread modify the neighbourhood of its surroundings, all the changes are saved on a separated grid.

The rules are defined so the state of the cell in the next step is decided from the current state of the cell and the current state of the neighbours cells. We only consider the 8 neighbours surrounding the cell in a 3x3 box with the cell in the center. The rules are defined based on probabilistic calculation.

We start by thinking what options could each box in the grid have and what is the probability that a tumor cell is in the box the next iteration. First we establish if there is a tumor cell currently in the box and then we determine if any of its neighbours contains a tumor cell. For each tumor cell in one of their neighbours, the probability that in the next iteration a tumor cell will occupy the box increases.

Let α be the box where we make the probability calculation. And we called P_α the probability that a tumor cell will occupy α at the next iteration. Let see what are the options for α:

1. If α doesn't have a tumor cell and none of their neighbours have a tumor cell, then $P_\alpha = 0$. No tumor cell will be in α the next iteration

2. If there is a tumor cell in α and there aren't tumor cells in its neighbourhood. We will use the probability of reproduction described in [**?**], call it p_{re}, and $P_\alpha = p_{re}$.

3. If there is a tumor cell in α and all of his neighbours are occupied by tumor cells, then the tumor cell in the box is considered to be in quiescence and will remain unchanged to the next iteration, so $P_\alpha = 1$.

4. If there isn't a tumor cell in α and some of its neighbours are occupied by tumor cells, we will have to calculate what is the probability of one of those tumor cells to migrate or reproduce into the α cell.

5. If there is a tumor cell in α and some of its neighbours are occupied by tumor cells we have to think first what are the odds that the actual tumor cell keeps its position, and if the tumor cell migrates or dies, we will have to calculate what is the probability of one of the neighbour tumor cells will migrate or reproduce into α.

With this approach we can assure that the calculation of each cell's state in the next iteration is independent of the calculation of the other cells. Using this we can assign a unique GPU-thread to each cell and run them concurrently without risk of race conditions, unlike the deterministic approaches where you had to prevent mutual exclusion when a cell pass to its next state.

1.1 Neighbour probability

We will first establish the elements that will remain unchanged during this section. Let α be the box of the grid where we make the probability calculation that a tumor cell could occupy it in the next iteration. Let then be $v \in \mathbb{N}, 1 \leq v \leq 8$ the number of tumor cells in the neighbouring boxes of α, and let $\mathcal{V} = \{1, ..., v\}$ the numbered list of α's neighbours, following the next distribution. As we are considering the $3x3$ box with α in the center as the neighbourhood, each tumor cell in the surrounding boxes of α will have an index, starting at the top left corner and, skipping boxes without tumor cells, going clockwise increasing the index as in Figure 1.

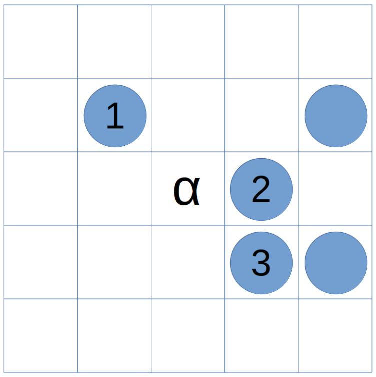

Figure 1: Labeling of α's neighbourhood

Using combinatorial formulas, we know that there are $v!$ ways to order $\mathcal{V}$. Lets call $\sigma_i \in \{\sigma_1, ..., \sigma_{v!}\} : \mathcal{V} \rightarrow \{\sigma_1, ..., \sigma_{v!}\} : \mathcal{V}$ with σ_i being a bijection for each $i \in \{1, ..., v!\}$, all the different permutations that give us all the different ways to order $\mathcal{V}$.

For each neighbor, we have to first calculate the probability that a tumor cell is generated in one of its own neighbors based on the Polesczuk-Enderling algorithm. We will use the probabilities of migration and reproduction of a cell define in the Polesczuk-Enderling algorithm, lets call them p_{mi} and p_{re}, and we will define $l \in \{0, ..., 8\}$ the number of boxes without a tumor cell in the neighbourhood of the cell. We decide that $l \neq 0$ because α will count as a box without tumor cell at the time we calculate the probabilities. We define $K = \frac{p_{mi}+p_{re}}{l}$ as the probability that one tumor cell makes one of its neighbours to be occupied by a tumor cell in the next iteration. We will label the probability K_λ for each possible cell λ of the grid.

To calculate the probability that a tumor cell could occupy the cell α in the next iteration we cannot simply take the probabilities K_{lambda} of its neighbors and sum them together. On the contrary, we have to consider all possible ordering sequences in which the neighbors of α could be processed by the Polesczuk-Enderling algorithm, since there is only room for one cell in α and that means that only one neighbor can be chosen to produce the tumor cell occupying α. Therefore, let us individually calculate the probability that a given cell is the one that produces a tumor cell in α. Lets call $a_0 \in \mathcal{V}$ one of the neighboring boxes of α with a tumor cell. We have to consider all different orders of $\mathcal{V}$ and ask ourselves how can a_0 be the neighbour that produces the tumor cell in α. Let us take a permutation σ_i, and see that $\sigma_i(j) = a_0, j \in \mathcal{V}$, which means that a_0 will be the j-th neighbour to be accessed. The probability that a_0 is the neighbour producing a tumor cell in α is them:

$$P = (1 - K_{\sigma_i(1)})...(1 - K_{\sigma_i(j-1)})K_{a_0} \tag{1}$$

Knowing that there are $v!$ different options to sum, we are going to group them by the position of a_0 in the order. Let start taking all $\sigma_\tau(1) = a_0$. Because a_0 is the first neighbour to be accessed the probability that is the one producing a tumor cell in α is based only in his own chances. Taking into account that there are v different σ_τ fulfilling that $\sigma_\tau(1) = a_0$, we can say:

$$P_1 = \frac{v}{v!}K_{a_0} = \frac{1}{(v-1)!}K_{a_0} \tag{2}$$

Let us continue with the calculation, now for the second neighbor, $\sigma_\tau(2) = a_0$. Knowing that a_0 is in the second position, we have that in the first position can be all the other $v-1$ neighbours, each of them being in a number of equally distributed occasions for each to be in first in $v(v-1)$ choices. Summing up we have that

$$P_2 = \frac{v(v-1)}{v!}(1 - K_1)K_{a_0} + ... + \frac{v(v-1)}{v!}(1 - K_v)K_{a_0} \tag{3}$$

$$P_2 = \frac{1}{(v-2)!}K_{a_0} \sum_{i \in \mathcal{V} \setminus \{a_0\}} (1 - K_i) \tag{4}$$

Let us generalize to the n-th position. Using the same logic, we have $v(v-1)...(v-(n-1))$ possibilities for the first n positions, so each option have $(v-n)!$

different σ, where the first n positions are in the same order. That will make the sum

$$\begin{aligned} P_n = &\frac{v(v-1)...(v-(n-1))}{v!}(1-K_1)...(1-K_n)K_{a_0}+\dots \\ &+\frac{v(v-1)...(v-(n-1))}{v!}(1-K_t)\dots(1-K_v)K_{a_0} \end{aligned}$$

But we can do better, seeing that the different probabilities K calculated are the different factors of a product, we can reduce the possibilities by using another combinatorial formula. This makes that there are $\binom{n-1}{v-1}$ combinations with respect to the order of the first n-1 position. Taking that the number of combinations is $v!$, but we have that a_0 is in the n-th positions so that $(v-1)!$ remains, each of these combinations has the probability given by the following equation

$$\frac{(v-1)!}{\binom{v-1}{n-1}} = \frac{(v-1)!(v-n)!(n-1)!}{(v-1)!} = (v-n)!(n-1)! \tag{5}$$

represents different σ permutations. So the sum total will be,

$$\begin{aligned} P_n = &\frac{1}{(v-n)!(n-1)!}K_{a_0}[(1-K_1)\dots(1-K_{n-1})+\dots \\ &+(1-K_1)...(1-K_v)+(1-K_2)(1-K_3)\dots(1-K_v)+\dots] \end{aligned}$$

Since we already know the formula for the probability of selecting each position where a_0 could be, we can now add up all the options to get the final probability that a_0 is the neighbor that produces a tumor cell in α.

$$\begin{aligned} P = &\frac{1}{(v-1)!}K_{a_0} + \frac{1}{(v-2)!}K_{a_0}[(1-K_1)+\dots+(1-K_v)]+\dots+ \\ &\frac{1}{(v-n)!(n-1)!}K_{a_0}[(1-K_1)\dots(1-K_{n-1})+\dots+(1-K_1) \\ &\dots(1-K_v)+(1-K_2)\dots(1-K_n)+\dots] \\ &+\dots+\frac{1}{(1)!(v-1)!}K_{a_0}[(1-K_1)(1-K_2)\dots(1-K_v)] \\ =&K_{a_0}[\frac{1}{(v-1)!}+\frac{1}{(v-2)!}[(1-K_1)+\dots+(1-K_v)]+\dots+ \\ &\frac{1}{(v-n)!(n-1)!}[(1-K_1)\dots(1-K_{n-1})+\dots+(1-K_1)\dots \\ &(1-K_v)+(1-K_2)\dots(1-K_v)+\dots]+\dots+ \\ &\frac{1}{(v-1)!}[(1-K_1)(1-K_2)\dots(1-K_v)]] \end{aligned} \tag{6}$$

Finally, considering the computational cost of calculating the above probability, we will factor, in order to reduce the number of products, in the following way

$$
\begin{aligned}
P =& K_{a_0}[\frac{1}{(v-1)!} + \frac{1}{(v-2)!}(1-K_1) + ...+ \\
& \frac{1}{(v-n)!(n-1)!}[(1-K_1)(1-K_2)...(1-K_{n-1}) + ... + (1-K_1)...(1-K_v)] + ...+ \\
& \frac{1}{(v-1)!}[(1-K_1)(1-K_2)...(1-K_v)]+ \\
& \frac{1}{(v-2)!}(1-K_2) + ...+ \\
& \frac{1}{(v-n)!(n-1)!}[(1-K_2)(1-K_3)... + ... + (1-K_2)...(1-K_v)] + ...+ \\
& \frac{1}{(v-2)!(1)!}[(1-K_2)...(1-K_v)] + ...+ \\
& \frac{1}{(v-2)!}(1-K_v)] \\
=& K_{a_0}[\frac{1}{(v-1)!} + (1-K_1)[\frac{1}{(v-2)!} + (1-K_2)[\frac{1}{(v-3)!(3-1)!} + ...] + ...] + ... \\
& + (1-K_2)[\frac{1}{(v-2)!} + (1-K_3)[\frac{1}{(v-3)!(3-1)!} + ...] + ...] + ... \\
& + (1-K_v)\frac{1}{(v-2)!}]
\end{aligned}
\tag{7}
$$