Tree methods with OpenMP

Exercice 1. We aim at computing the price of an Amercian put option using a tree method. Let T be the maturity time of the continuous time model and $N > 0$ the number of discretization steps. We consider a regular time step $h = \frac{T}{N}$.
Let $u = \mathrm{e}^{\sigma\sqrt{h}}$, $d = \mathrm{e}^{-\sigma\sqrt{h}}$ and $1 + R = \mathrm{e}^{rh}$. The CRR model is defined by

$$X_{n+1}^N = U_{n+1} X_n^N \text{ pour } n \geq 1, \quad X_0^N = x. \tag{1}$$

where the sequence $(U_n, n \geq 1)$ is iid with valued in $\{d, u\}$ and

$$\mathbb{P}(U_1 = u) = 1 - \mathbb{P}(U_1 = d) = \frac{d - \mathrm{e}^{rh}}{d - u} \triangleq p.$$

The time-0 price is given by $P(0, X_0)$ where the function P solves the backward induction

$$\begin{cases} P(N, y) & = (K - y)_+ \\ P(n-1, y) & = \max\Big((K-y)_+, \mathrm{e}^{-rh}(P(n, yu)p + P(n, yd)(1-p))\Big) \quad \forall 1 \leq n \leq N. \end{cases}$$

Solving this backward equation boils down to traversing a binary and recombining tree starting from the leaves.

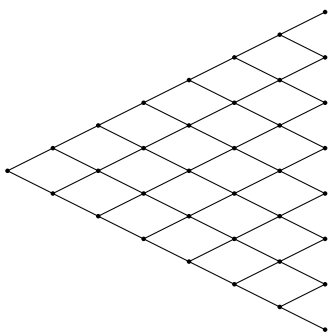

FIGURE 1 – Tree with depth 7

As an example, one can take : $X_0 = 100$, $K = 100$, $r = 0.05$, $\sigma = 0.2$, and $T = 3$. Then, the price is around 8.65.

1. Understand and analyze the sequential code **arbre-omp/arbre.c**.
2. Propose a way of parallelizing this code using `parallel for` directives, maybe split in two separated pragmas.
3. Compare the prices obtained by the sequential and parallel codes. Explain the difference.
4. Without modifying the pragams used in the parallel code, fix the code of question 2.
5. Study the performance of the parallel code.
6. Modify the parallel code to reduce the number of calls to the function `pow`.
7. Propose a new way of traversing the tree to identify independent subtrees.
8. Analyze the potential performance improvements.
9. Implement this new parallel approach to solve the dynamic programming equation using the functions `solve_crr_tree` and `solve_crr_tree_upsidedown`.