Hardware acceleration of continuous cellular automata through parallel computing on GPUs

Jehiel López Grande
Advisors: Dr. René Armando Cumplido Parra, Dr. Julio César Pérez Sansalvador, Dr. Ignacio Algredo Badillo
Instituto Nacional de Astrofísica, Óptica y Electrónica

Abstract

Continuous Cellular Automata (CCA) are extensions of traditional cellular automata that allow each cell to hold real-valued states, enabling the simulation of more complex, dynamic, and realistic systems. These models are applied in biology, physics, and social sciences. However, CCA introduce additional computational challenges due to the use of floating-point arithmetic, complex transition functions, and extended neighborhoods. These characteristics increase the computational cost of simulations, especially when dealing with large grids. To address these challenges, this work proposes the acceleration of CCA through parallel execution on Graphics Processing Units (GPUs). We explore the design and implementation of GPU-based parallel computing framework tailored to the structure and requirements of continuous models.

Keywords: Acceleration, GPGPU, Continuous Cellular Automata, Parallel Computing.

Introduction

Cellular Automata (CA) are discrete computational models where each cell updates its state based on local rules and its neighbors [1]. Classical CA use binary or discrete states.

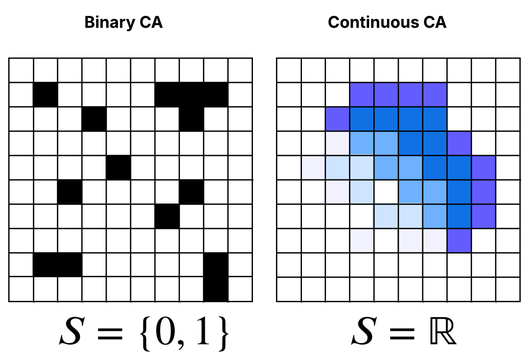

Figure 1. Cellular automaton

Continuous Cellular Automata (CCA) extend this concept by allowing real-valued states, enabling more realistic modeling of complex systems [2]. Fast and efficient execution is crucial to enable large-scale simulations.

Objectives

General Objective:
Design and implement a GPU-based parallel computing framework to improve the execution throughput of continuous cellular automata, considering the common features of models used to simulate biological, physical, and social phenomena.

Specific Objectives:

1. Analyze contemporary continuous cellular automata models to identify structural patterns and computational bottlenecks relevant to parallel execution.
2. Design and validate a parallelization strategy optimized for GPU architecture (CUDA) to accelerate continuous state updates, ensuring computational fidelity against reference models and demonstrating system scalability.
3. Quantify the performance gains and hardware resource efficiency achieved by the applied optimization strategies within the proposed framework.
4. Demonstrate the adaptability and generalizability of the framework by deploying and benchmarking diverse continuous cellular automata models.

Cellular Automata State Taxonomy

Classification Category	State Structure (S)	Criteria for Inclusion
Discrete Single State CA	$S = \{s\}$ where $s \in \{S_1, \ldots, S_k\}$	The cell state is a **single value** selected from a **finite, discrete set**
Continuous Single State CA	$S = \{x\}$ where $x \in \mathbb{R}$	The cell state is a **single value** that is a **real/continuous number**.
Discrete Tuple State CA	$S = (s_1, \ldots, s_m)$ where all s_i are discrete	The cell state is a **tuple**, and **all elements** are discrete values from finite sets.
Hybrid/Continuous Tuple State CA	$S = (s_1, \ldots, s_m)$ where at least one $s_i \in \mathbb{R}$	The cell state is a **tuple**, and **at least one** element is continuous or real-valued.

Table 1. Classification of Cellular Automata Models by State Representation

Case Study: Reaction–Diffusion Systems

Reaction–diffusion (RD) systems describe how the concentrations of interacting substances evolve in space and time through two coupled processes: local nonlinear reactions and spatial diffusion.

Accelerating RD models on GPUs presents additional challenges due to:

- **Floating-point data types:** RD models rely on real-valued concentrations, increasing arithmetic cost and memory bandwidth usage.
- **Nonlinear reaction terms:** The local reaction processes introduce nonlinear behavior that makes each update step computationally heavier than in simple discrete CA rules.
- **Diffusion stencils:** Implementing Laplacian operators requires accessing multiple neighbors, intensifying memory access pressure on the GPU.

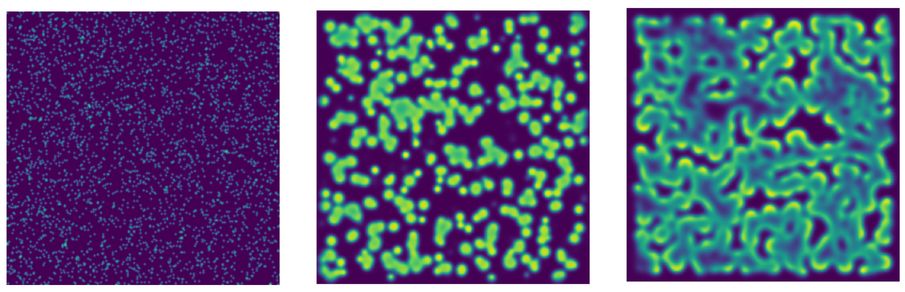

Figure 2. Gray–Scott reaction–diffusion model at different simulation steps.

Baseline Validation

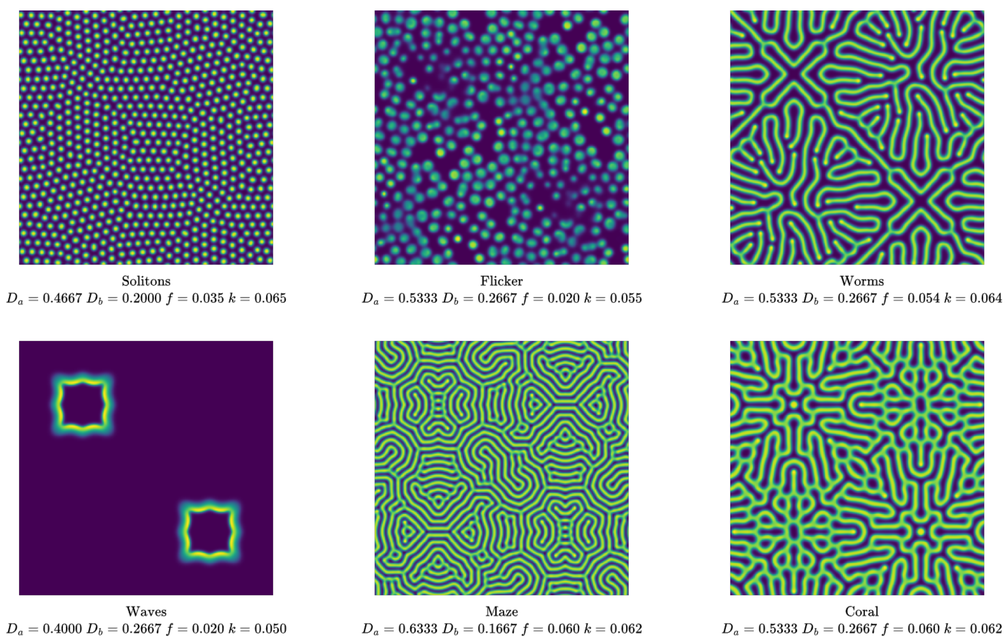

Figure 3. Baseline CUDA implementation showing patterns exhibited by the Gray-Scott reaction-diffusion system using fixed diffusion and reaction parameters (D_a, D_b, f, k).

Performance Evaluation

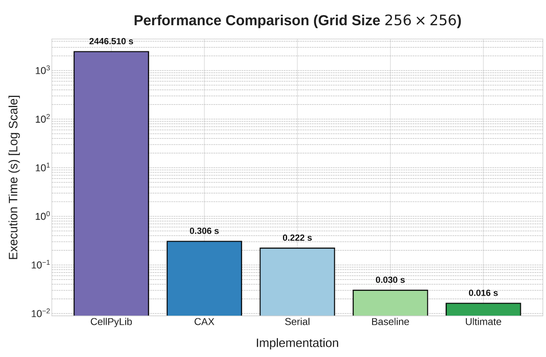

Figure 4. Execution time comparison on a 256×256 grid.

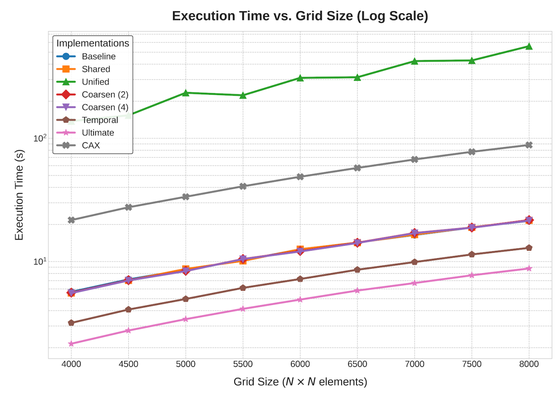

Figure 5. Execution time comparison of different memory access patterns and optimization techniques applied to the Gray-Scott model.

Framework Generalization

The following models will be implemented leveraging the previously validated acceleration techniques, extending their application to additional cellular automata models:

- **SmoothLife**: Continuous field artificial life model.
- **Brusselator**: Nonlinear theoretical oscillator.
- **Oregonator**: Chemical model of the Belousov-Zhabotinsky reaction.
- **FitzHugh-Nagumo**: Excitable medium neuron model.
- **Gierer-Meinhardt**: Activator-inhibitor morphogenesis model.

Schedule

Month	1	2	3	4	5	6	7	8	9	10	11	12
Literature review												
Review of GPU architecture												
Parallelization strategy design												
Prototype implementation												
Experimental evaluation												
Results analysis												
Final review and submission												

References

[1] Bhattacharjee, K., Naskar, N., Roy, S., & Das, S. (2020). A survey of cellular automata: Types, dynamics, non-uniformity and applications. Natural Computing, 19(2), 433–461. https://doi.org/10.1007/s11047-018-9696-8
[2] Rollier, M., Zielinski, K. M. C., Daly, A. J., Bruno, O. M., & Baetens, J. M. (2025). A comprehensive taxonomy of cellular automata. Communications in Nonlinear Science and Numerical Simulation, 140, 108362. https://doi.org/10.1016/j.cnsns.2024.108362
[3] Cagigas-Muñiz, D., Diaz-del-Rio, F., Sevillano-Ramos, J. L., & Guisado-Lizar, J.-L. (2022). Efficient simulation execution of cellular automata on GPU. Simulation Modelling Practice and Theory, 118, 102519. https://doi.org/10.1016/j.simpat.2022.102519
[4] Navarro, C. A., Quezada, F. A., Meneses, E., Ferrada, H., & Hitschfeld, N. (2025). CAT: Cellular Automata on Tensor Cores. IEEE Transactions on Parallel and Distributed Systems, 36(2), 341–355. https://doi.org/10.1109/TPDS.2024.3520395
[5] Compeau, P. (2022). Biological Modeling: A Short Tour. Philomath Press. ISBN 9798986819211. https://www.amazon.com/Biological-Modeling-Short-Phillip-Compeau/dp/B0BT6B2B8W