CS 652 Final Project

Daniel Loman

6/18/2012

Abstract

Conway's Game of Life is the classic example of emergence in cellular automata. Presented here is a 2D and 3D expansion of the original version of Conway's Game of Life. Extension into 3 spatial dimensions as well as an extension of the 1 or 0 based neuron approach to a numerical scale ranging from 1 to 10. During this study we witnessed rapid trends toward equilibrium states dependent on the location transition function, initial conditions and boundary conditions.

1 Introduction

Conway's Game of Life, further referred to as GoL, is a cellular automata based game based upon the mathematical problems presented by John Von Neumann. The GoL provides a simple mechanism for the study of emergence in self-organization.[2] Extending the GoL to 3 spatial dimensions and increasing the complexity of each individual automata state allows a significant increase in the complexity of the interactions we are able to model , allowing for the simulation of diffusion, whether that be diffusion of chemical reactions or heat flow.

1.1 Cellular Automata

Cellular automata is a modeling techniques are based on modeling a manifold using a finite number of discrete states. Each automata has a set of additional cells commonly referred to as its neighbors.[1] This neighborhood of cells is the set of cells residing in some specific minimum spatial distance from the cell itself. Cellular automata state transformations are dependent on the state of their neighbors and a specific set of rules that the cellular automata is operating on commonly refeed to as a local transition function.[4] This respect is where cellular automata differ from coupled map lattices; state values are entirely dependent on initial conditions and neighborhood state values. Traditionally the structure of cellular automata states are constructed using the neuron model; the states can be either 'on' or 'off'. More complex automata can also be modeled with state complexity significantly more complex than 1s and 0s.[12]

1.2 Diffusion

Diffusion is a transportation mechanism that does not require bulk movement. Molecular diffusion is considered the result of the random walk powered by thermal energy. [3] Diffusion is most often modeled using vector wave equations. Vector calculus models of diffusion allow for the modeling of an infinite number of particles quickly and is computationally less intensive.[7] Discretization of diffusion allows for more a more fine grained approach at the cost of an increase in computational complexity.[5]

2 Experimental Background

This experiment updates the classic 2D binary state cellular automata or CA which is defined by the tuple $[F, \sum, f]$ where F is defined as the set of integers, each automata has 8 in the 2D case or 26 in the 3D case classical Moore Neighbors. $iu(x) = y \in F : x \neq y \& |x - y| \leq 1$. It also follows that $\sum = 0, 1$ in the classical cellular automata but in this experiments case is extended to be the range of double precision floating point number ranging from [0.0,10.0]. f is defined as the local transition function, in the classical cellular automata this is defined as[8][9]

$$x^{t+1} = f(u(x^t)) = \begin{cases} 1, & \text{if } x^t = 0 \text{ and } \sigma_x^t \in [\theta_1, \theta_2] \\ & \text{or } x^t = 1 \text{ and } \sigma_x^t \in [\delta_1, \delta_2] \\ 0, & \text{otherwise} \end{cases}$$

(1)

In this experiment the local transition function is the subtraction of a scaled value of the average value of the cell's 26 neighbors. It is defined as the following equation.

$$x^{t+1} = f(u(x^t)) = x^t - \sum_{-1 \leq x \geq 1} \sum_{-1 \leq y \geq 1} \sum_{-1 \leq z \geq 1} F(u(x^t(x, y, z))$$

(2)

3 Design

In this experiment a 3 dimensional cellular automata was created and leveraged to model heat or chemical

diffusion in a solid object. For each time step a plot is generated and the next time steps grid is generated. When generating the next grid all values are compared to their neighboring values and the local transition function is evaluated and applied. Once all time steps are completed an animation is made of all the plots.

4 Implementation

This experiment was implemented using the python programming language. Python was chosen for its flexibility, its combination of object oriented design principles, its rich feature set, as well as the power and flexibility of the open source graphing utilities and matrix processing modules, specifically the Numpy and Matplotlib modules.

4.1 Numpy & Matplotlib

Numpy and Matplotlib are extensions or modules to the python programming language. Because python is an interpreted language many mathematical functions run extremely slowly when compared to the speed of compiled software. Numpy solves this issue by implementing most mathematical algorithms as well as matrix and array mathematics efficiently with runtimes nearly equivalent to C or FORTRAN code.[14] Matplotlib on the other hand is a plotting library that is designed to quickly and easily take Numpy arrays and matrices and make powerful complex plots leveraging the power of OpenGL.[15]

4.2 Multiprocessing Module

The computational power required to compute large grid sizes of 3 dimensional cellular automata is not insignificant. Parallel processing, the utilization of multiple processors to carry out simultaneous calculations, could greatly increase the performance of our CA. The python multiprocessing library was leveraged to give the python access to multiple cpu cores. Unfortuanlty given time constraints the multiprocessing implementation is not yet fully implemented. However work on it will continue and all parallel code is available on the *para* git branch.

5 Source Code

All source code including the uncompleted parallel branch and experimental results are available via github using command: **$git clone https://github.com/dloman/Diffusion3d.git**

6 Results

All experiments shown were done using a boundary condition of 0 to ensure continuity in all tests. In each experiment we initialized all cells to a the maximum value of 10.0. Other initial conditions are easily possible but the most interesting results came from these initial conditions. In the following figure you can see the grid at initialization.
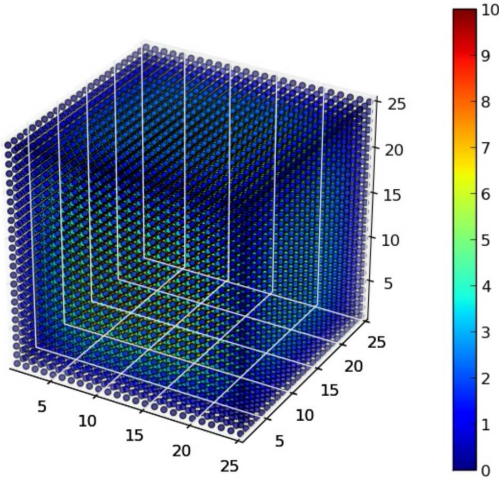
Figure 1: CA when $t = 0$
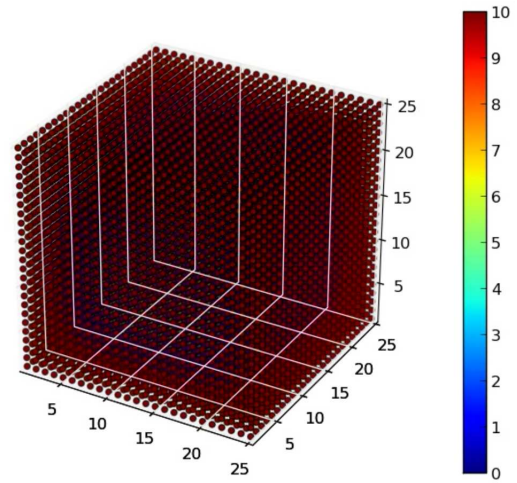
As the time step is increased the boundary condition starts effecting the CA's values, most significantly in the 8 corners of the cube.

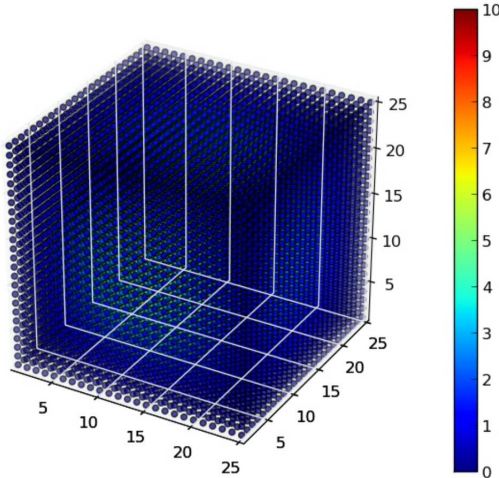Figure 2: CA when $t = 33$
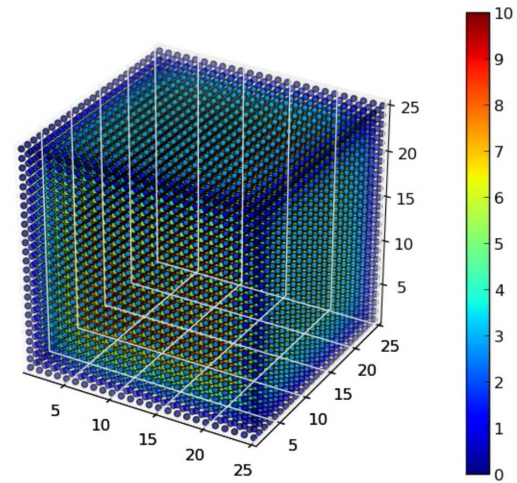
As time increases further the flow spreads into the center of the cube as shown below:

Figure 3: CA when $t = 66$

until all values of the CA approach equilibrium.

Figure 4: CA when $t = 100$

7 Conclusion

Cellular automata are powerful and interesting tools for use in the study of diffusive systems. They allow fine grain control of system interactions and in-depth view of real time system properties[10]. The evolution of the CA values is strictly controlled, like in all CA, by the local transition function and the boundary conditions[11]. The averaging aspect of the local transition function as well as the forced value boundary conditions implemented in this experiment lead to the CA trending rapidly towards equilibrium. With a different implementation of local transition functions, such as averaging plus randomness[13] or markov chain random walks, as well as changes to the boundry condition system, such as boundaries with no effect on the system, would have significant impact on the behavior of the system. These types of behaviors would be of interest to be studied in further detail in the future.

References

[1] Boccara N. and M. Roger, Some properties of local and nonlocal site exchange deterministic cellular automata, *Int. J. Modern Phys.*, **C5**,581-588 (1994).

[2] De Arcaneglis, L. and Herrmann, H. J., Self-organized criticality on small world networks, *Physica A*, **308**, 545-549 (2002).

[3] Appert C. and Zaleski S., Lattice gas with a liquid-gas transition, *Phys. Rev. Lett.*, **64**, 1-4 (1990).

[4] Guinot V., Modelling using stochastic, finite state cellular automata: rule inference from continuum model, *Appl. Math. Model.*, **26**, 701-714(2002).

[5] Newman M. E. J., Watts D. J., Scaling and percolation in the small-world network model, *Phys. Rev. E*, **60**,7332-7342 (1999).

[6] Tomassini M., *Generalized automata networks*, 7th Int. Conference on Cellular Automata for Research and Industry, ACRI 2006, France, *Lecture Notes in Computer Sciences*, **4173**, 14-28 (2006).

[7] Tomassini M., Giacobini M., Darabos C., Evolution and dynamics of small-world cellular automata, *Complex Systems*, **15**, 261-284 (2005).

[8] Watts D. J., *Small worlds: The dynamics of networks between order and randomness*, Princeton Univ. Press, (1999).

[9] Watts D. J., Strogatz S. H., Collective dynamics of small-world networks, *Nature* (London), **393**, 440-442 (1998).

[10] Weimar J. R., Cellular automata for reaction-diffusion systems, *Parallel computing*, **23**, 1699-1715 (1997).

[11] Wolfram, S., *Cellular automata and complexity*, Reading, Mass: Addison-Wesley (1994).

[12] S. Wolfram, *A New Kind of Science, Champaign, Illinois*, 2002.

[13] Yang X. S., Chaos in small-world networks, *Phys. Rev. E.*, **63**, 046206 (2001).

[14] *Matplotlib* http://en.wikipedia.org/wiki/Matplotlib 2012.

[15] *Numpy* http://en.wikipedia.org/wiki/Numpy 2012.