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 towards equilibrum states dependent on the location transition function, initial conditions and boudnry conditions.

1 Introduction

Conway's Game of Life, further refereed to as GoL, is a cellular automata based game based upon the mathematical problems presented by John Von Neumann. The GoL provides an 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 modeling techniuqe based on modeling a manifold using a finite number of discrete states. Each automata has a set of additionally cells commonly refered to as its neighbors.[1] This neighborhood of cells are the set of cells residing some specific minimum spatial distance from the cell itself. Cellular automata state transformation are dependent on the state of thier neighbors and specific set of rules that the cellular automata is operating on.[4] This respect is where cellular automata differ from coupled map lattice, state values are entirley dependent on inital conditions, and neighborhood state values. Traditionally the structure of cellular automata states are constructed using the neuron model, or the states can be either 'on' or 'off'. More Complex automata can also be modeled with state complexity significantly more complex then 1s and 0s.

1.2 Diffusion

Diffusion is transportation mechanism that doesn not require bulk movement. Molecular diffusion is consider the result of the random walk powered by thermal energy. Diffusion is most often model using vector wave equations. Vector calculous models of diffusion allow for the modeling of an infinite number of particles qucikly and computatinally less intensive.[7] Discretization of diffusion allows for more a more fine grained approach, at the cost of an increase in compuational 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 intergers, 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 cellualar automata but in this experiments case is extended to be the range of double precision floating point number rangeing from [0.0,10.0]. f is defined as the local transition function, in the classical cellular automata this is defined as[8][9]

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

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.

$$x^{t+1} = f(u(x^t))) = x^t - \sum_{-1 \leq x \geq 1} \sum_{-1 \leq y \geq 1} \sum_{-1 \leq z \geq 1} F(u(x^t(x, y, z)) \tag{2}$$

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 generation grid all values are compared to thier 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 Implimentation

This experiment was implemented using the python programming language. Python was choosen for its flexibility, its combination of object oriented design princibles, its rich feature set, as well as the power and flexibility of the open source graphing utilties 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 mathimatical functions run extrmely slowly when compared to the run times of compiled software. Numpy solves this issue by implimenting most mathmatical algorithms as well as matrix and array mathamatics with runtimes nearly eqivalent to C or FORTRAN code.[13] Matplot lib on the other hand is a plotting library which is designed to quckly and easily take Numpy arrays and matricies and make powerful complex plots leveraging the power of OpenGL.[14]

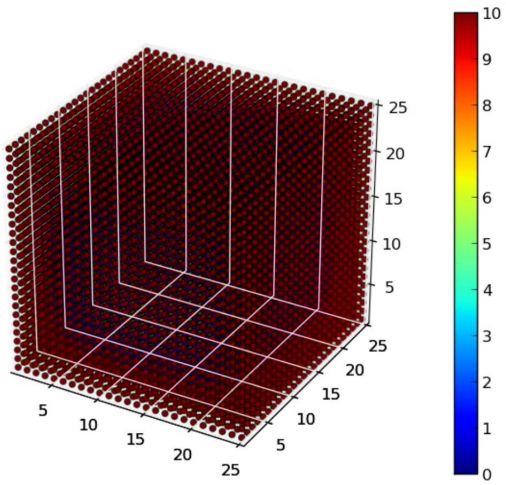
4.2 Multiprocessing Module

The compuational power require 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 mulitprocessing library was leveraged to give the python access to multiple cpu cores. Unfortuanlty given time constraints the multiprocessing implimentation is not yet fully implimented. Howerver work on it will continue and all parallel code is availble on the *para* git branch.

5 Results

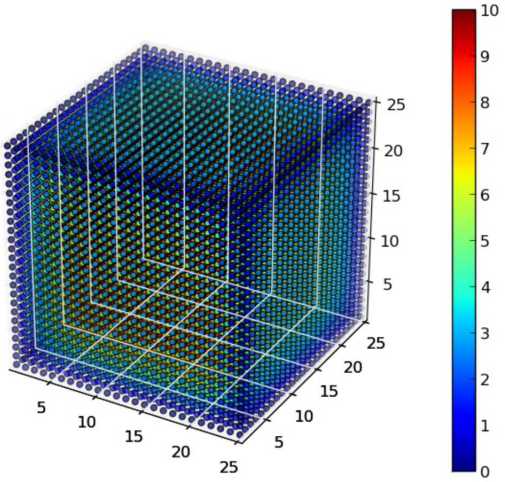
All experients shown were done using a boundry condition of 0 to ensure continuity in all tests. In each experiement we began all cells at a the maximum value of 10.0, other initial conditions are easily possible but the most interesting results came from these initial conditions. In the following figure you can see the grid at initialization.

Figure 1: CA when $t = 0$



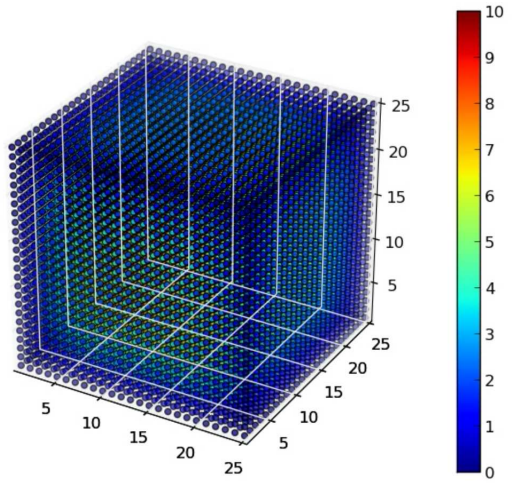
As the time step is increased the boundry condition starts efffecting, most significantly in the 8 corners of the cube, the CA's values.

Figure 2: CA when $t = 33$



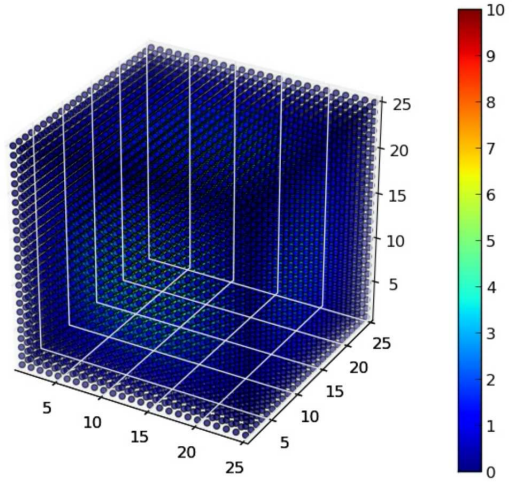
As time increases further the flow spreads further into the center of the cube as shown below:

Figure 3: CA when $t = 66$



until all values of the CA approach equlibrium.

Figure 4: CA when $t = 100$



6 Conclusion

Cellular automata are a powerful and interesting tool for use in the study of diffusive system. They allow fine grain control of system interactions and indepth view of real time systme properites[10]The evolution of the CA values is strictly controlled, like in all CA, by the local transition function and the boundry conditions[11]. The averageing aspect of the local transion function as well as the forced value boundry conditions implimented in this experiemnt lead to the CA treding rapidly towards eqiulibrium. With a different implimentation of local transion functions, such as avergaing plus randomness[15] or marchov chain random walks, as well as changes to the boundry condition system, such as boundries with no effect on the system would have significant impact on the behavior of the system. These types of behaviors would have 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] *Matplotlib* http://en.wikipedia.org/wiki/Matplotlib 2012.

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

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