CAS 520 Peter Dresslar Final Project: 1-Dimensional Life

Introduction

I converted the NetLogo model, “Life” (Uri Wilensky, 1997) into two implementations of one-dimensional Cellular Automata (CA). Life is based on the well-known two dimensional CA, Conway’s Game of Life (GOL), and ships with NetLogo with a 10201-cell domain with wrapping (toroidal behavior) on by default.

The three models I am delivering are:

- 1d-2d-Life
- 1d-Life
- Life

Life’s CA ruleset evaluates the current state of any given cell and the summed states of its eight two-dimensionally-adjacent neighbors. If the current cell value being evaluated has state s over the state space [0, 1], s(t+1) equals 1 if the sum is 3, s(t) if the sum is 2, and 0 if the sum is <2 or >3.

Building from an initial observation that a finite version of GOL could be re-represented in one dimension by taking advantage of a finite “side-length” (n) and the related n*n=N, the supplied 1d-2d-Life model illustrates the simultaneous and (nearly) equivalent processing of GOL in both dimensionalities. Crucially, 1d-2d-Life departs from Life by using two one-dimensional state arrays, rather than NetLogo patches, to process state: representing (t, t+1), each array has length N. Neighbors are located using additive combinations of s, n, and N as appropriate (see source code for details.) When operated, the model demonstrates the familiar phenomena of GOL such as “still-lifes” and “spaceships,” and 1D analogous primitive forms such as gliders can be seen to survive, move, and even wrap around the screen.

The 1d-2d-Life model would not work with a theoretically infinite system, since there would in that case be no discrete point at which the two state arrays could be separated.

The second model, 1d-Life, takes this approach further by investigating the abandonment of the connection to the 2D version altogether. The model accomplishes dimensional-indifference by shifting what was a “side-length” scalar into an independent model parameter called λ (lambda), so named due to its use in the model as a sequence frequency. The model implements a one-dimensional patch “field” of 10201 cells—this can also be thought of as a 1D “tape—and in this implementation uses NetLogo native patch agents to store and operate state. This aspect of the model is similar to the Life model, and very little code from the base model has been altered outside of the addition of λ. The neighborhood evaluations resolve to simple additive algebra with s and λ in this setup:

```
{s-λ-1,s-λ,s-λ+1,s-1,s+1,s+λ-1,s+λ,s+λ+1}.
```

The third provided model, Life, is a slightly modified version of the “base” Wilensky 2D GOL model used in experimentation.

Analysis and Results

In order to evaluate the models we can first run them and visually inspect their outputs. As mentioned above, the outcome of running produces pattern development and movement in the 1D visualizations evocative—if only through evolutionary growth—of their 2D analogues.

While it can be difficult to visually resolve patterns in the 1D versions, running a step in the 1d-2d-Life model and stopping processing can be useful in order to compare the results in the 1d output with the appropriate (bottommost) row in the 2D output. Special setup buttons are provided in both the 1D versions that allow the user to set up and watch the processing of a preset “glider” form in both dimensions. Note that these buttons in the 1d-Life version incorporate the setting of lambda into their “seeding” of the gliders. Toggling to different values for λ between button presses can be instructional.

The base Life model includes a simple “density” monitor that calculates the proportion of living (s=1) cells in the model. This monitor has been copied/adapted into the 1D versions. It can be seen through visual inspection that for different starting densities of random grid states, the model tends to collapse eventually to a static final density that is dependent in part on the starting density. Exploring the hypothesis that the nearly-identical operation of the models we should see a similar relationship between starting and final densities in each model, BehaviorSpace experiments were developed and run for each of the models, taking the Wrapping-On and Wrapping-Off settings as an additional parameter to explore.

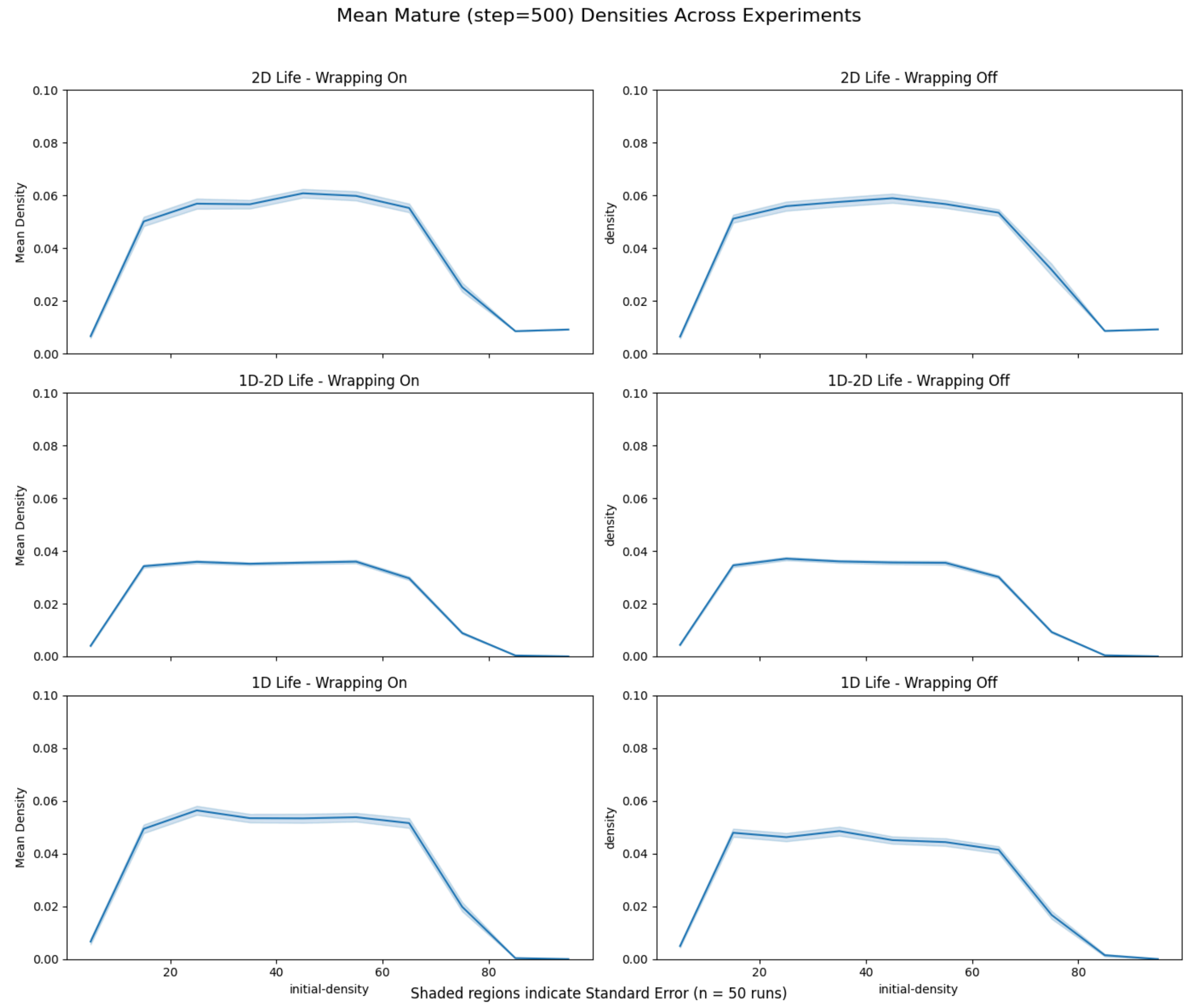
The BehaviorSpace experiments are included with the supplied models. Detailed data can be found in the attached Python notebook. The BehaviorSpace experiments are `initial-density` parameter sweeps using living-cell density at a mature model state (500 steps) for their analytical currency, and scan across the range of initial densities from 5 percent to 95 percent (by tens).

For reference, the following table (Table 1) lists the mature model state means and standard deviations for the 35-percent initial-density runs, selecting this value since it is the base model’s default:

Model variant	**Wrapping On**	**Wrapping Off**
	Mature density (mean ± stdev)	**Mature density (mean ± stdev)**
2-D Life (base)	0.053 ± 0.011	0.049 ± 0.012
1D-2D Life	0.057 ± 0.011	0.058 ± 0.012
1-D Life	0.035 ± 0.004	0.036 ± 0.004

Table 1 – Step 500 density at 35 % initial-density; runs = 50 (per experiment)

The following figure (Figure 1) displays the outcomes in comparable plots.



Mean Mature (step=500) Densities Across Experiments

2D Life - Wrapping On

2D Life - Wrapping Off

1D-2D Life - Wrapping On

1D-2D Life - Wrapping Off

1D Life - Wrapping On

1D Life - Wrapping Off

Shaded regions indicate Standard Error (n = 50 runs)

Figure 1 – Experiment visualizations

Discussion

From the results, we can observe that it appears that the *two models supply successful 1-dimensional recapitulations of the 2D GOL CA.* We can also observe that the fidelity between the models is not absolute. This is not a surprise: the relationship of the boundary conditions between our 1D, 1D-2D, and 2D models is topologically complex, yielding slightly different results over large numbers of runs due to senescent-cell adjacency shapes and/or wrapping conditions. It seems likely that improvements could be made to the 1D implementations to better cover these boundary cases, but, in the end, it seems that the most accurate implementations would always yield differences between 1D and 2D implementations.

We might note that additional probing could be made to ascertain more precise correlations between models. However, it is my assessment that this analysis would be premature. Part of the reason for this is that the 1d-Life model was a late, unplanned, and thought-provoking addition to the project.

Instead of deeper analysis into the models as they exist, for now the logical next step would likely be to formalize the general relationships between the models—including the boundary conditions for each—using descriptive mathematical models. From that point, it would then be possible to assure that each of the (new) models maintains the best possible fidelity to the formal definitions. From this perspective, the current state of the models and associated analysis is sufficient for the next task at hand.

The following is a list of additional observations raised as open questions to potentially explore further.

1. The parameter λ successfully transforms the two-dimensional CA into a one-dimensional CA that is indifferent to any other dimensional representation. It operates as a “tuner” on the cells of the tape; when it is changed mid-process, patterns quickly decohere and vanish from the operation of the automaton.
2. By allowing this indifference, λ also relaxes the finite constraint on the 1D version of GOL. While computing is finite, there appears to be no theoretical reason why the 1D version could not work in an infinite space.
3. λ only processes correctly in one dimension with a setting of 2 or greater. At a value of 1, the model immediately collapses, since the defined neighborhood overlaps. There does not appear to be an upper bound on lambda, apart from any limitations derived from the amount of space available to the implementation. We might observe that a setting for lambda of n + any positive multiple of N cells in a finite tape might work as a stand-in (for instance, 2 and 10203 might work equivalently with a theoretical 10201 cell tape, though this approach is not provided for in the supplied implementation.
4. It is natural to contemplate the relationship between lambda and 2D implementations of Life. One possibility is that lambda could be seen as present and part of the 2D implementation, but invisible due to an inherent setting to an additive identity (0) or a

multiplicative identity (1). One possible way to investigate this would be by “skipping” a dimension and evaluating three dimensional implementations instead.

5. The supplied glider templates point the way toward a relatively novel mechanism for defining “important” life primitives from GOL. For instance, offsets {0, ±(λ+1), ±(2λ−1), ±2λ, ±(2λ+1)} for an origin point, o, define a left or right moving “glider” in 1D. This definition appears more fundamental than corresponding higher dimensional descriptions of CA “life.”
6. We can observe that the 1-D version of GOL appears to represent a more fundamentally compact data processing and storage approach to operating the process of the CA, which could be useful in applications where performance is critical.
7. The approach taken seems likely to be easily applicable to other 2D CA, and possibly more generally applicable.

Finally, it should be observed that while the Universality and Turing Completeness of the 2D CA is well known, the question of whether one or both of the 1D implementations would share those characteristics is not immediately clear. To be clear, no evidence is presented here indicating any such qualities. The question seems worth further investigation.

I hope to continue to pursue these and other aspects of the project beyond the scope of this class, and would welcome the opportunity to discuss any of them further in the appropriate context.

Real World Applications

While Conway’s Game of Life has existed over the past fifty years largely as a foray into abstract theoretical grounds, a cursory search through online sources yields some recent acceleration in the application of this particular model and CAs in general to real-world research. These applications include image processing and artificial intelligence.

License

I am obligated to use the CC BY-NC-SA 3.0 for the models, as they are adaptive works according to the terms of the original model here: https://creativecommons.org/licenses/by-nc-sa/3.0/. The rest of my repository uses the MIT license. I prefer it as an open source standard. For more information, please see: https://github.com/peterdresslar/GOL-1d-nl/blob/main/LICENSE.

References

Conway, J. H. (1970). "The Game of Life." *Scientific American*, 223(4), 120–123.

Wilensky, U. (1997). *NetLogo Life model*. Center for Connected Learning and Computer-Based Modeling, Northwestern University. https://ccl.northwestern.edu/netlogo/models/Life

Related Works

Schiff, J. L. (2011). Cellular Automata: A Discrete View of the World. Wiley.

Wolfram, S. (2002). A New Kind of Science. Wolfram Media.

Adamatzky, A. (Ed.). (2010). Game of Life Cellular Automata. Springer.

Ilachinski, A. (2001). Cellular Automata: A Discrete Universe. World Scientific.

Mitchell, M. (2009). Complexity: A Guided Tour. Oxford University Press.

Golly (Cellular Automata simulation software): http://golly.sourceforge.net/

Evans, K. M. (1994). "Larger than Life: Digital Creatures in a Family of Two-Dimensional Cellular Automata." https://www.ibiblio.org/lifepatterns/largerthanlife.html

Game of Life Lexicon. ConwayLife.com Community. https://conwaylife.com/ref/lexicon/lex_home.htm