Non-Platonic Autopoiesis of a Cellular Automaton Glider in Asymptotic Lenia

Q. Tyrell Davis[1],
[1]Independent Researcher
Boulder, Colorado, United States
qtd.science.wrought049@passmail.net

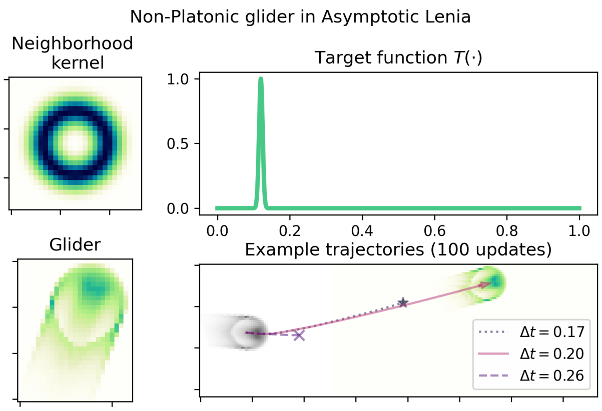

Figure 1: **A non-Platonic glider in asymptotic Lenia.** With step size $\Delta t = 0.2$, the glider persists for 100 update steps, but disappears earlier for step sizes $\Delta t = 0.17$ or 0.26. The rules supporting this glider are defined by a target function (Equation 5) with $\mu_T = 0.12$ and $\sigma_T = 0.005$, and neighborhood kernel K with $\mu_K = 0.5$ and $\sigma_K = 0.15$ and native radius $k_0 = 13$ pixels ($r = 1.0$ at 13 pixels from kernel center). Code available at https://github.com/riveSunder/fractal_persistence. Instructions for replicating results at commit 09437415...

Introduction

Like Life, Lenia CA support a range of patterns that move, interact with their environment, and/or are modified by said interactions. These patterns maintain a cohesive, self-organizing morphology, *i.e.* they exemplify autopoiesis, the self-organization principle of a network of components and processes maintaining themselves (Varela et al., 1974), and a framework for investigating the life-like properties of CA (Beer, 2004, 2015).

Lenia dynamics are described by PDEs. This description lends itself to simulation with computers, as in classical physics, albeit incurring the 'necessary evil' of discretization. Unlike simulating classical physics, where a trade-off of precision and accuracy in exchange for keeping time and computational costs feasibly low is made, a subset of patterns in Lenia and similar systems have been shown to lose their autopoietic competence when simulated with discretization parameters too fine (Davis, 2023; Kojima and Ikegami, 2023; Davis and Bongard, 2022). For these patterns in the context of their respective rules, simulation appears part and parcel to their ability to self-organize: they exhibit non-Platonic autopoiesis.

Lenia has been expanded with a number of variations on the original. In this work I examine a glider in asymptotic Lenia (abbreviated 'ALenia' in this work) (Kawaguchi et al., 2021). ALenia replaces the growth function with a target function: the update rule is then defined by the difference between the output of the target function and the current cell states.

Standard Lenia uses a clipping function to constrain cell values to a range from 0 to 1, but ALenia's update scheme does away with the need. Recent work implementing ALenia as a reaction-diffusion system reported that non-Platonic behavior in standard Lenia may depend on the clipping function, and that ALenia gliders are likely not affected by non-Platonic dissolution (Kojima and Ikegami, 2023). In this work I show an example of a glider in ALenia that depends on a certain simulation coarseness for autopoietic competence: when simulated with too fine spatial or temporal resolution the glider no longer maintains its morphology or dynamics.

Lenia and Asymptotic Lenia

An instance of Lenia has states A defined as cell (or pixel) values at coordinates $x = (x_h, x_w)$ that change over time t. The update rule has a form similar to the Euler method, adding the product of the result of differential equation $\frac{\partial A}{\partial t}$ and step size Δt:

$$A(x, t + \Delta t) = \left[A(x, t) + \Delta t \cdot \frac{\partial A}{\partial t} \right]_0^1 \quad (1)$$

Unlike the Euler method, Equation 1 constrains cell values $A(x, t)$ in the range $\mathcal{R}[0, 1]$. Hard truncation, represented by $[\cdot]_0^1$, is typically used, but many pattern-rule pairs work equally well with a smooth squashing function.

Lenia dynamics are defined by a differential equation, aka the growth function $G(\cdot)$.

$$\frac{\partial A}{\partial t} = G(K \circledast A(x,t)) \tag{2}$$

Where K is a neighborhood kernel (*e.g.* top left in Figure 1) and $\circledast$ represents convolution. The growth function takes a similar form to that of a Gaussian, stretched to output values from -1 to 1:

$$G(x) = 2 \cdot e^{-\left(\frac{x-\mu_G}{\sigma_G}\right)^2} - 1 \tag{3}$$

Where μ_G and σ_G set the peak and width of the curve and $x = K \circledast A(x,t)$ represents cell neighborhoods.

The neighborhood kernel $K(r) = e^{-\left(\frac{r-\mu_K}{\sigma_K}\right)^2}$ is also Gaussian-like, and acts on radial coordinates r. Neighborhood kernels are normalized to sum to 1.0.

ALenia replaces the growth function G with a target function T. $\frac{\partial A}{\partial t}$ is then the difference of cell values $A(x,t)$ and target function applied to cell neighborhoods $K \circledast A(x,t)$

$$\frac{\partial A}{\partial t} = T(K \circledast A(x,t)) - A(x,t) \tag{4}$$

ALenia's target function is unstretched, yielding values from 0 to 1:

$$T(x) = e^{-\left(\frac{x-\mu_T}{\sigma_T}\right)^2} \tag{5}$$

As a result of the asymptotic target function, ALenia obviates the need to truncate cell values.

Autopoiesis and Discretization

I simulated the glider in Figure 1 across a range of spatial and temporal resolutions defined by Δt and k_r, respectively.

I modified temporal resolution Δt by simply using a different value for the step size `dt`. In pseudocode: `A = A + dt * (T(conv(A,K)) - A)`. To smoothly vary spatial resolution, I change the size of the neighborhood kernel $K(r)$ by normalizing to kernel radius kr (in pixels): `xx, yy = meshgrid(x, y); r = sqrt(xx**2 + yy**2) / kr`. I used scikit-image (Van der Walt et al., 2014) to rescale patterns[1] and define glider stability by homeostasis of glider 'mass', *i.e.* the sum of cell values. I chose normalized (relative to starting) cell-sum thresholds of 0.9 to 1.3 based on empirical observations of typical variation for a stable glider.

Results are shown in Figure 2. Gain of stability when moving from fine to coarse simulation parameters (Δt, k_r) is consistent with non-Platonic autopoiesis, as is a loss of stability when increasing k_r and/or decreasing Δt.

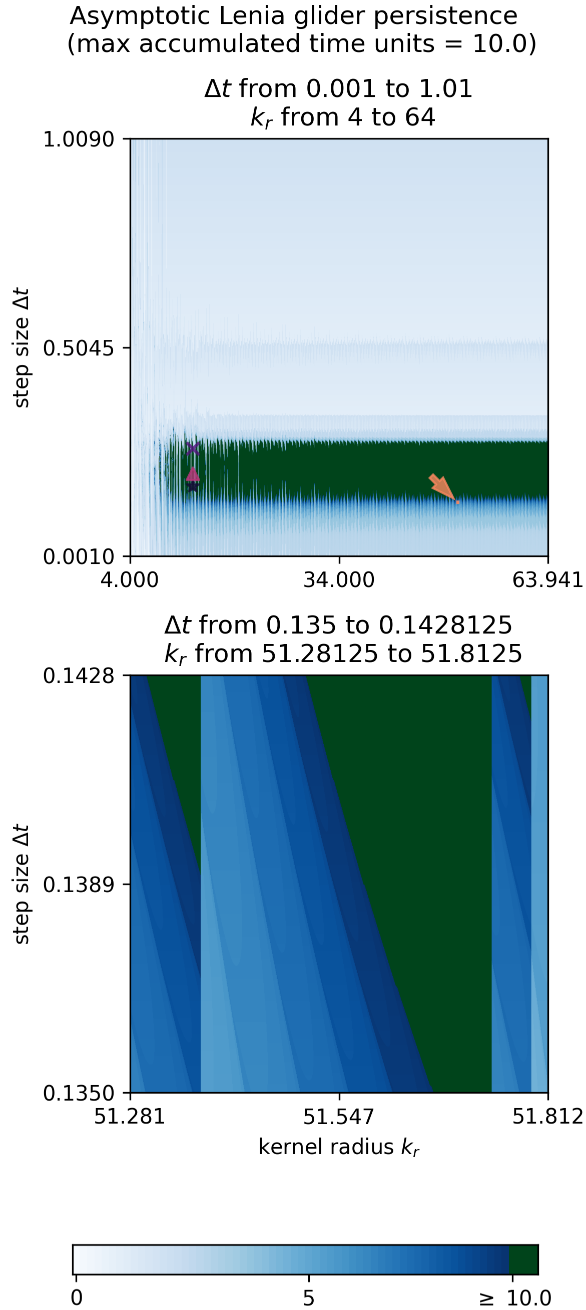

Figure 2: **ALenia glider persistence with respect to Δt and k_r shows non-Platonic loss of autopoiesis**. Persistence in accumulated time units is indicated by dark colors, with a transition from blue to green at the arbitrary threshold of 10 time units, the maximum simulation time. 'X', triangle, and star symbols mark Δt = 0.26, 0.20, and 0.17 from Figure 1, respectively. The lower persistence map is a zoom to the region in the small orange rectangle indicated by the arrow, showing retained details of apparent fractal structure.

[1] `skimage.transform.rescale` with `scale_factor = kr/k0` and `order=5` spline interpolation, and `anti_aliasing=True` when downscaling.

References

Beer, R. D. (2004). Autopoiesis and cognition in the game of life. *Artificial Life*, 10(3):309–326.

Beer, R. D. (2015). Characterizing autopoiesis in the game of life. *Artificial life*, 21(1):1–19.

Davis, Q. T. (2023). Discretization-dependent dissolution of gliders in (dis)continuous systems: Non-platonic self-organization in complex systems. *Innovations in Machine Intelligence (IMI)*, 3:1–23.

Davis, Q. T. and Bongard, J. (2022). Step size is a consequential parameter in continuous cellular automata. *ALIFE 2022: The 2022 Conference on Artificial Life*, 43.

Kawaguchi, T., Suzuki, R., Arita, T., and Chan, B. (2021). Introducing asymptotics to the state-updating rule in Lenia. In *ALIFE 2021: The 2021 Conference on Artificial Life*. MIT Press.

Kojima, H. and Ikegami, T. (2023). Implementation of Lenia as a reaction-diffusion system. In *ALIFE 2023: Ghost in the Machine: Proceedings of the 2023 Artificial Life Conference*. MIT Press.

Van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., and Yu, T. (2014). scikit-image: image processing in python. *PeerJ*, 2:e453.

Varela, F. G., Maturana, H. R., and Uribe, R. (1974). Autopoiesis: The organization of living systems, its characterization and a model. *Biosystems*, 5(4):187–196.