Algorithmic Art with Discrete Dynamical Systems

Sinan Kapçak

American University of the Middle East, Kuwait; sinan.kapcak@aum.edu.kw

Abstract

Discrete dynamical systems feature a recurrent structure and rich qualitative behavior, including chaos, making them suitable for algorithmic art. After defining an orbit of a system, we discuss techniques for aesthetically enhancing their renderings. We use additional functions to control drawing attributes such as the point size, color, opacity and more.

Introduction

A *discrete dynamical system* (DDS), or an *iterative map*, is a function with a fixed rule that determines the future states by iterating on previous states. This fact alone makes DDS suitable for algorithmic art. Some well-known examples of the use of DDS in algorithmic art include the work of Joel [3], the Bridges papers by Krawczyk (e.g. [5, 6]), and some of the references in both papers. These works mainly focus on the generation of drawings of strange attractors by exploiting the orbits of iterative maps. In addition to the orbits, in this paper, we will use iterative maps to generate many other attributes of algorithmic art such as geometric shape, color and opacity. Moreover, we will use a chaotic map to obtain pseudorandom numbers, which will determine the geometry of one of our examples.

Planar Maps

A *planar map*, or a *two-dimensional DDS*, can be represented by the following system of equations:

$$\begin{aligned} x_{n+1} &= f(x_n, y_n), \\ y_{n+1} &= g(x_n, y_n), \end{aligned} \tag{1}$$

where $f : \mathbb{R}^2 \to \mathbb{R}$ and $g : \mathbb{R}^2 \to \mathbb{R}$. Starting with the initial point (x_0, y_0), from the system defined in (1), we have $x_1 = f(x_0, y_0)$ and $y_1 = g(x_0, y_0)$. Similarly, x_2 and y_2 can be obtained via the equations $x_2 = f(x_1, y_1)$ and $y_2 = g(x_1, y_1)$, respectively. Hence, for the initial point (x_0, y_0), system (1) can be used to obtain the following (finite) orbit in the plane:

$$P_0 = (x_0, y_0),\ P_1 = (x_1, y_1), \ldots,\ P_k = (x_k, y_k).$$

As an example, consider the following planar map:

$$\begin{aligned} x_{n+1} &= \frac{1}{2}(x_n - y_n), \\ y_{n+1} &= \frac{1}{2}(x_n + y_n). \end{aligned} \tag{2}$$

Let $P_0 = (3, 1)$. Applying this point to system (2), we obtain $P_1 = (1, 2)$, $P_2 = (-\frac{1}{2}, \frac{3}{2})$, $P_3 = (-1, \frac{1}{2})$, and so on. Figure 1, which presents a *phase diagram* of the system, shows the points for 10 iterations. The line segments connect the points so that the consecutive states can be tracked. Clearly, the points are approaching

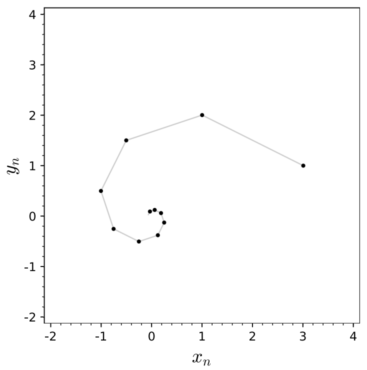


Figure 1: *Phase diagram of discrete system (2): The orbit for the initial condition* $P_0 = (3, 1)$ *is shown.*

the origin. Phase diagrams are graphical representations of the states of dynamical systems. For each initial point, a phase diagram gives an ordered list of discrete points in the plane, called an *orbit* of the system.

A point (x^*, y^*) is said to be a *fixed point* of system (1) if $f(x^*, y^*) = x^*$ and $g(x^*, y^*) = y^*$. Clearly, $(0, 0)$ is a fixed point of system (2). There are three types of fixed points that a system may possess. A fixed point (x^*, y^*) is called *asymptotically stable* if the following holds: For all starting values (x_0, y_0) near (x^*, y^*), not only does the system remain near (x^*, y^*) but also $(x_n, y_n) \longrightarrow (x^*, y^*)$ as $n \longrightarrow \infty$. A fixed point (x^*, y^*) is called *stable* if for all starting values (x_0, y_0) near (x^*, y^*), the system remains near (x^*, y^*) but does not converge to (x^*, y^*). A fixed point (x^*, y^*) is called *unstable* if it is neither asymptotically stable nor stable.

The following well-known theorem gives the stability condition of a fixed point of a planar map. Let us first recall the Jacobi matrix J^* for a planar map of the form given in (1), which is defined as

$$J^* = \begin{pmatrix} f_x & f_y \\ g_x & g_y \end{pmatrix},$$

where the partial derivatives are evaluated at the fixed point (x^*, y^*).

Theorem 1 [3, p. 200]**.** *Let* J^* *be the Jacobi matrix of system (1) at the fixed point* (x^*, y^*)*. Then,* (x^*, y^*) *is an asymptotically stable fixed point if the trace-determinant condition* $|\operatorname{tr} J^*| - 1 < \det J^* < 1$ *holds true.*

Theorem 1 gives the stability condition of the fixed point (x^*, y^*). For equation (2), the Jacobi matrix at the fixed point $(0, 0)$ becomes

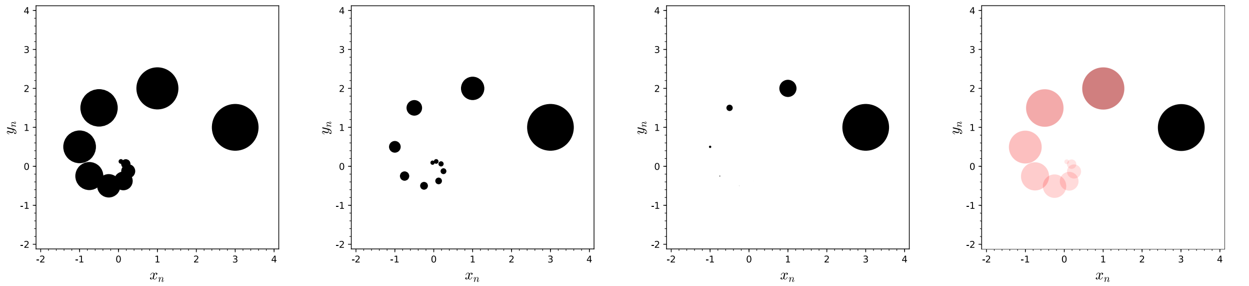
$$J^* = \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}.$$

Since $\operatorname{tr} J^* = 1$ and $\det J^* = \frac{1}{2}$ for system (2), the origin is asymptotically stable by Theorem 1. For this stable fixed point, the orbit is spiraling inward (see Figure 1). There are many other qualitative behaviors of a discrete system. References for the theory of DDSs include [2, 1]. Some interactive applets for exploring one- and two-dimensional iterative maps and the corresponding details can be found in [4].

Example 1: Playing with Circles

In this example, we restrict the geometric objects of interest to circles/disks and generate some corresponding outputs. We control five parameters: the point size (radius of a circle/disk), the red-green-blue (RGB) color and the opacity (alpha).

In a DDS, the positions of the points depend on the iteration. To generate artistic dynamics, we use the iteration number n as the independent variable, and compute the point size, RGB color, and opacity using functions of n. One way to control the point size is to use an increasing (resp. decreasing) function to increase (resp. decrease) the point size in each iteration. First, let $L(n)$ be the size of the n-th point (radius of the n-th disk) of the orbit, and let it be defined by the linear function $L(n) = \frac{(b-a)n}{k} + a$, where a and b are non-negative real numbers and k, a non-negative integer, is the number of iterations. For $n = 0$, we have $L(0) = a$ as the size of the initial point, while the k-th iteration gives the size of the terminal point, $L(k) = b$. To obtain a hyperbolic change, we define the following hyperbolic function of n: $H(n) = \frac{a+bn}{1+n}$, where a and b are non-negative real numbers. For the initial point, when $n = 0$, the point size is $H(0) = a$ whereas for a very large n, the point size is very close to the limiting value $\lim_{n\to\infty} H(n) = b$. By assigning the function H to define the point size in each iteration, we obtain point size values that vary hyperbolically from a to b. If $a > b$ (resp. $a < b$), the point size is decreasing (resp. increasing). A sharper change in the point size can also be defined by adopting the following function, which yields an exponential variation: $E(n) = 2^{-n}a + (1 - 2^{-n})\, b$, where a and b are non-negative real numbers. Similarly, when $n = 0$ and $n \longrightarrow \infty$, we have a and b, respectively. Moreover, in place of the functions H, E or L, various other functions, e.g., trigonometric, polynomial, or constant, can be chosen to obtain different series of point sizes.



(a) $L(n)$ **(b)** $H(n)$ **(c)** $E(n)$ **(d)**

Figure 2: *Dynamic point sizes, color and opacity for system (2). An orbit for ten iterations is shown. The point size decreases (a) linearly, (b) hyperbolically and (c) exponentially. In (d), the point size, red color and opacity change linearly, exponentially and hyperbolically, respectively.*

Now, as an example, we apply the functions L, H and E, with $a = .6$ and $b = 0$, to the point size parameter of equation (2). Figure 2 shows the resulting linear, hyperbolic and exponential changes in the point size. Similarly, for red-green-blue-alpha (RGBA) values, we may use the functions L, H and E to generate dynamic colors and levels of opacity in the artwork. Since each of the red, green, blue and alpha values are drawn from the interval $[0, 1]$, in addition to the functions L, H and E, we may also use the trigonometric functions $S(n) = |\sin n|$ and $C(n) = |\cos n|$, where n may be expressed in degrees or radians. When RGB is $(0, 0, 0)$ and $(1, 1, 1)$, we obtain black and white values, respectively. When alpha (the opacity) is 1, we have a complete opaque color. As an example, we apply dynamic colors to Figure 2a. We fix the G and B values at zero and exponentially vary the red value, with a starting value of 0 and a terminal value of 1, while the opacity is varied hyperbolically from 1 to 0; see Figure 2d. Alternatively, instead of taking these functions of the iteration n, we may also define one-dimensional maps to control the point size and RGBA parameters, which usually yields much richer dynamics, as we will see in the next example.

Now, we will present three dynamical systems and their outputs obtained using this technique.

1. The following system with $\beta = 0.09$ yields the output shown in Figure 3a:

$$\begin{aligned} x_{n+1} &= \beta x_n - y_n, \\ y_{n+1} &= \frac{x_n}{1+\beta^2}. \end{aligned} \tag{3}$$

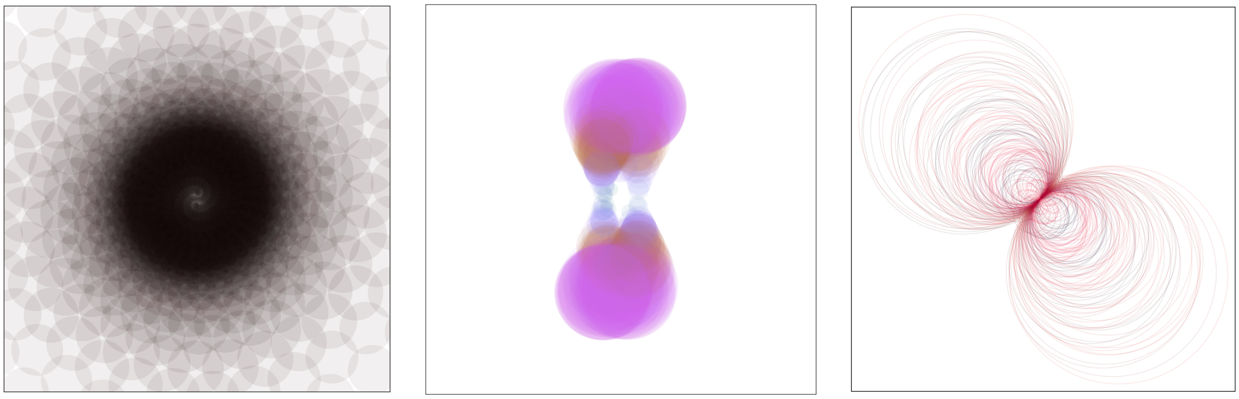
2. Figure 3b shows the output of the following single-parameter system with $a = 2.35$.

$$\begin{aligned} x_{n+1} &= \sin(x_n - a y_n), \\ y_{n+1} &= a^2 \sin\left(y_n^2 - x_n\right). \end{aligned} \tag{4}$$

3. For the following system, we exclude the interior of the disks. In this way, Figure 3c is obtained.

$$\begin{aligned} x_{n+1} &= -1.9 x_n - y_n, \\ y_{n+1} &= 0.99 x_n. \end{aligned} \tag{5}$$

Details about the initial points and iteration numbers can be found in Table 1. The quantity d_n in the table is defined as $d_n = \sqrt{x_n^2 + y_n^2}$, which is the distance from (x_n, y_n) to the origin. The starting and terminal values as well as the types of functions used for the point size, color and opacity are also given. For example, it can be seen in the table that for equation (4), the opacity (alpha) varies from 0.2 to 0.01 in accordance with the function $H(n)$. Similarly, for equation (5), the red color value starts at zero and ends at 1 in accordance with the sine function $S(d_n)$.



(a) *Equation (3)* **(b)** *Equation (4)* **(c)** *Equation (5)*

Figure 3: *The outputs of the discrete systems given by equations (3), (4) and (5).*

Example 2: Neimark-Sacker Bifurcation and Controlling Chaos with Logistic Map

We now focus on an example of the attracting closed curve that appears when Neimark-Sacker bifurcation [2] takes place. We introduce the following DDS, which exhibits this behavior when $\alpha = 0.571$, $\beta = 1.749$, $\mu = 0.754$, $k = 8.625$, and the initial point $(x_0, y_0) = (0.5, 0.5)$.

$$\begin{aligned} x_{n+1} &= x_n \left(\alpha + \beta e^{-k y_n}\right), \\ y_{n+1} &= \mu x_n \left(1 - e^{-k y_n}\right). \end{aligned} \tag{6}$$

Table 1: *Details on the Dynamical Systems in Example 1.*

DDS	Equation (3)	Equation (4)	Equation (5)
Initial point	$(-1.1, -0.6)$	$(3, 1)$	$(3, 1)$
Point size	$(0.125, 0) - H(d_n)$	$(4, 0.1) - \frac{d_n}{2}$	$(2, 0.1) - d_n$
Red	$(0, 1) - E(d_n)$	$(0, 1) - H(d_n)$	$(0, 1) - S(d_n)$
Green	$(0, 1) - H(d_n)$	$(0.9, 0.3) - H(d_n)$	$(0, 0.039) - H(d_n)$
Blue	$(0, 1) - H(d_n)$	$(1, 0) - S(d_n)$	$(0, 0.26) - L(n)$
Alpha	$(0.2, 0) - H(d_n)$	$(0.2, 0.01) - H(n)$	$(0.2, 0.1) - H(n)$
Number of iterations	600	300	275
Figure	3a	3b	3c

Consider also the logistic map $u_{n+1} = ru_n(1 - u_n)$, where $0 \leq r \leq 4$. Note that $0 \leq u_i \leq 1$ for every positive integer i when $0 \leq u_0 \leq 1$. We set $u_0 = 0.3$ as the initial condition.

The main goal here is to construct a series of line segments $\overline{Q_iT_i}$. The endpoints Q_i and T_i are determined by system (6) and the logistic map together. Points $P_i = (x_i, y_i)$ are generated by system (6). We take Q_i as the point that satisfies the vector equation $\overrightarrow{OQ_i} = (1 - u_i)\overrightarrow{OP_i} + u_i\overrightarrow{OP_{i+1}}$, where O is the origin and u_i is the number generated by the logistic map. Now consider the vector $\mathbf{v}_i = \overrightarrow{P_iP_{i+1}}$. We rotate the vector $\mathbf{v}_i$ by the rotation matrix $M_i(\theta) = \left(\begin{smallmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{smallmatrix}\right)$, where the rotation angle is $\theta = 360u_i$ degrees, and we obtain the vector $\mathbf{w}_i = M_i\mathbf{v}_i$. The point T_i that satisfies $\overrightarrow{OT_i} = \overrightarrow{OQ_i} + \mathbf{w}_i$ is now the other point of line segment; we thus obtain the series of line segments $\overline{Q_iT_i}$. See Figure 4.

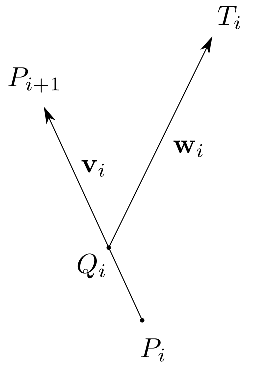


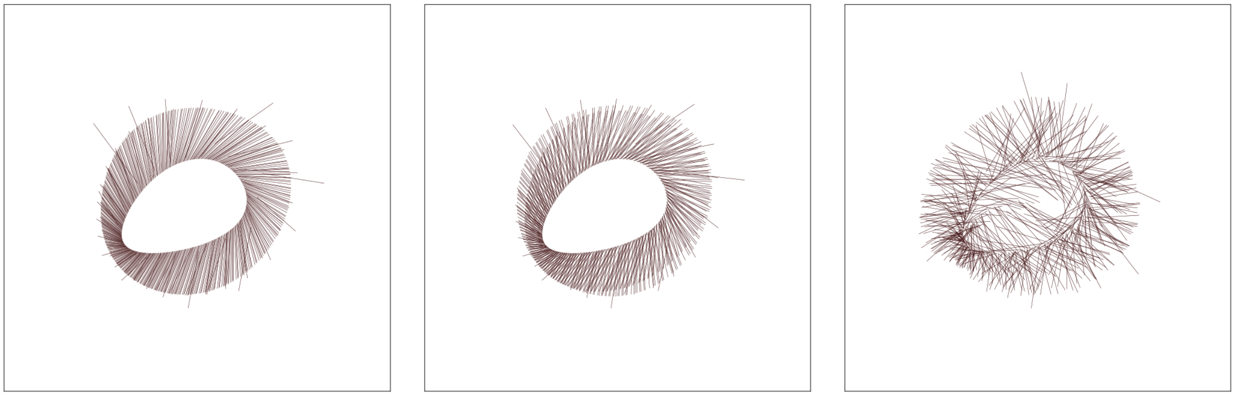
Figure 4: *Construction of the line segments $\overline{Q_iT_i}$ using equation (6) and the logistic map.*

When $0 < r < 1$, the fixed point origin of the logistic map is asymptotically stable, and when $1 < r < 3$, the positive fixed point is asymptotically stable, resulting in stable directions for the line segments. The logistic map also exhibits periodic orbits. For example, the case of $3 < r < 1 + \sqrt{6}$ yields a period 2 orbit. The system also exhibits chaos for some values of r. Figure 5 presents three different scenarios with various parameter values.

Conclusions

We have presented a means of using DDSs for creating algorithmic art. Through two examples, we have shown several possible techniques. In Example 1, although the object type was restricted to circles/disks, we obtained a rich variety of outputs. The reason is that, in addition to the numerous behaviors a discrete system can exhibit, we introduced dynamic point size, color and opacity parameter with values generated using functions of the iteration number.

One of the strengths of the use of a DDS for this purpose is the availability of equations that contain



(a) $r = 2.976$ **(b)** $r = 3.008$ **(c)** $r = 3.664$

Figure 5: *Line segments generated by Neimark-Sacker bifurcation and the logistic map after 500 iterations. Different parameter values lead to different directions for the sequence of line segments.*

free, or tuning, parameters that allow us to specify a generative art system whose parameter space can then be explored. As an alternative to a logistic map, as used in Example 2, one may apply another chaotic map such as a Ricker map or tent map. A two-dimensional map that exhibits periodic/chaotic orbits, such as a Hénon map, a baker's map, or a gingerbreadman map can also be used.

All code was written in the open, free computer algebra system *SageMath* [7], version 8.4, running on a PC with Linux (Ubuntu 18.04.1 LTS). Since the algorithm is iterative and we usually use up to 1500 iterations, it is extremely fast. In *SageMath*, as an output, a vector graphics image (.svg file) can be exported, which is usually important when one wishes to obtain a large printout of an image.

References

[1] R. Devaney. *An Introduction to Chaotic Dynamical Systems*. CRC Press, 2018.

[2] S. N. Elaydi. *Discrete Chaos: With Applications in Science and Engineering*. Chapman and Hall/CRC, 2008.

[3] W. J. Joel. "A simple attraction." *Computers & Graphics*, vol. 16, no. 1, 1992, pp. 41–43.

[4] S. Kapçak. "Discrete Dynamical Systems with SageMath." *Electronic Journal of Mathematics & Technology*, vol. 12, no. 2, 2018. https://php.radford.edu/~ejmt/ContentIndex.php#v12n2.

[5] R. J. Krawczyk. "The Ghostly Imagery of Strange Attractors." *Bridges Conference Proceedings*, Winfield, Kansas, USA, Jul. 30–Aug. 1, 2004, pp. 333–336. https://archive.bridgesmathart.org/2004/bridges2004-333.html.

[6] R. J. Krawczyk. "Dimension of Time in Strange Attractors." *Meeting Alhambra, ISAMA-BRIDGES Conference Proceedings*, Granada, Spain, Jul. 23-25, 2003, pp. 119–126. http://archive.bridgesmathart.org/2003/bridges2003-119.html.

[7] SageMath. The Sage Mathematics Software System (Version 8.4), The Sage Developers, 2018, https://www.sagemath.org.