Chaotic Systems as Simple (But Complex) Compositional Algorithms
Author(s): Rick Bidlack
Source: *Computer Music Journal*, Vol. 16, No. 3 (Autumn, 1992), pp. 33-47
Published by: The MIT Press
Stable URL: http://www.jstor.org/stable/3680849
Accessed: 01-04-2017 20:59 UTC

Rick Bidlack
Department of Music
State University of New York at Buffalo
Buffalo, New York 14260 USA
rb@acsu.buffalo.edu

Chaotic Systems as Simple (but Complex) Compositional Algorithms

Chaos is the generic term used to describe the output, under certain conditions, of nonlinear dynamical systems. It is generally believed that what we call chaos is a behavioral characteristic found in a vast number of real-world phenomena. Chaos, in other words, describes important aspects of the way things in the real world move and change with time. As such, chaos is of potential interest to composers who work with computers, because it offers a means of endowing computer-generated music with certain natural qualities not attainable by other means. Several composers and researchers in the computer music community have recently described their work using chaos and its mathematical cousin, fractal geometry. Applications of chaotic dissipative maps to sound synthesis have been explored by Truax (1990) and Waschka and Kurepa (1989). The use of chaotic systems as generators of higher level events, such as note sequences, has been described by Pressing (1988), Di Scipio (1990), and Gogins (1991).

This article is concerned exclusively with the use of chaotic systems as generators of note events. In such a context, chaos is employed as an algorithm for making choices having to do with note parameters such as pitch, dynamic level, rhythm, and instrumentation. Rather than viewing the output of chaotic systems as music in its own right, however, it is probably best to consider such output as raw material of a certain inherent and potentially useful musicality. Clearly there will be as many ways to apply chaos to musical decision-making as there are composers interested in doing so. In light of this, the discussion below emphasizes a breadth of behaviors that may be elicited from nonlinear dynamical systems, while making use of fairly neutral mappings of chaotic sequences into a generic musical space. One hopes that these examples will be sufficiently compelling in their own right to inspire other composers to develop their own more extensive and idiosyncratic manipulations of the output of the systems.

Computer Music Journal, Vol. 16, No. 3, Fall 1992,

Classifying Chaotic Systems

Nonlinear dynamical systems are modeled as systems of mathematical equations. These equations are usually iterated such that the solutions calculated from one iteration are fed back into the equations to become the input values for the next iteration. The sequence of values produced in this manner is called an *orbit.* A particular orbit of a nonlinear dynamical system represents the behavior of that system only for a particular set of initial conditions (i.e., the values assigned to the variables and constants of the system at the beginning of the calculation sequence). Since the set of initial conditions of any chaotic system is infinite (it equals the set of real numbers), the number of orbits attainable from any such system is also infinite.

Chaotic systems may be classified into two basic types, *dissipative* and *conservative*. Dissipative systems correspond to those phenomena in which friction plays a role and energy is dissipated to the surrounding environment. The overwhelming majority of natural phenomena on Earth are of this type. Conservative systems are best exemplified by the dynamics of celestial bodies, in which energy is conserved.

Since dissipative systems lose energy over time, their phase space—that is, the area or volume in which the dynamics of the system take place—shrinks with time. It is this shrinkage that leads to the formation of an *attractor* after an initial transient phase. An attractor may consist of a single point (a single-point attractor), a number of points among which the orbit oscillates (a periodic attractor), or a complex and fractally arranged folding of space (a chaotic attractor). Conservative systems, on the other hand, maintain a constant phase space. Orbits of conservative systems do not go through a transient phase, nor are they drawn to an attractor. Nonetheless, conservative systems still produce orbits that remain on a single point, or cycle among a narrowly restricted set of points, or wander erratically throughout the phase space.

Chaotic systems are formulated either as *iterated maps* in one or two dimensions (notated mathematically with difference equations), or *continuous flows* in three or more dimensions (notated with differential equations). Most of the work up to the present involving musical applications of chaotic systems has made use of dissipative iterated maps. Examples of four distinctly different systems are discussed here: two iterated maps, one dissipative and one conservative; and two continuous flows, again one dissipative, the other conservative. More detailed explanations of nonlinear dynamical systems have been presented by Thompson and Stewart (1986) and Moon (1987) and are recommended. Devaney (1988) also has provided a very clear overview of the fundamental concepts of chaotic orbits and iterated systems.

Each of the systems discussed below is accompanied by a sufficient amount of C language program to demystify the translation of the mathematical equations into a working computer program. The imaginary *output()* function call in each of the program examples is located at the place in the iteration loop where the output values of the system should be tapped. Typically, a routine like *output()* would either plot its arguments on a graph or map them into some musical space. (Real-time mappings into the MIDI domain are also feasible on most computers.) Each code fragment includes variable initializations guaranteed to produce a stable orbit of the system. The macro *ITERATION_LOOP* would typically be defined as a *while()* or *for()* expression.

Fig. 1. C language fragment demonstrating the calculation of an orbit of the Hénon map. Note that with each iteration of the equations, the value pair (x, y) is plotted or mapped to some appropriate musical space, the equations are solved anew, and the new (x, y) values are passed on to the next iteration.

```
/* Initialize variables and constants
 * with values which will produce a
 * chaotic orbit. */
float x = 0,        /* variable */
      y = 0,        /* variable */
      new_x,
      new_y,
      A = 1.4,      /* constant */
      B = 0.3;      /* constant */

/* Iterate through the equations an
 * arbitrary number of times. */
ITERATION_LOOP {
      /* Plot or produce sound. */
   output (x, y);
      /* Solve the equations. */
   new_x = y + 1.0 - (A * (x * x));
   new_y = B * x;
      /* Assign values for iteration. */
   x = new_x;
   y = new_y;
}
```

Chaos in Two Dimensions: the Hénon Map

The Hénon map was originally introduced as a simple and efficient model of dissipative chaotic systems in general, and is not derived from any particular natural phenomenon (Hénon 1976; Eckmann and Ruelle 1985). It is an iterated map of the Cartesian plane expressed as a system of two difference equations:

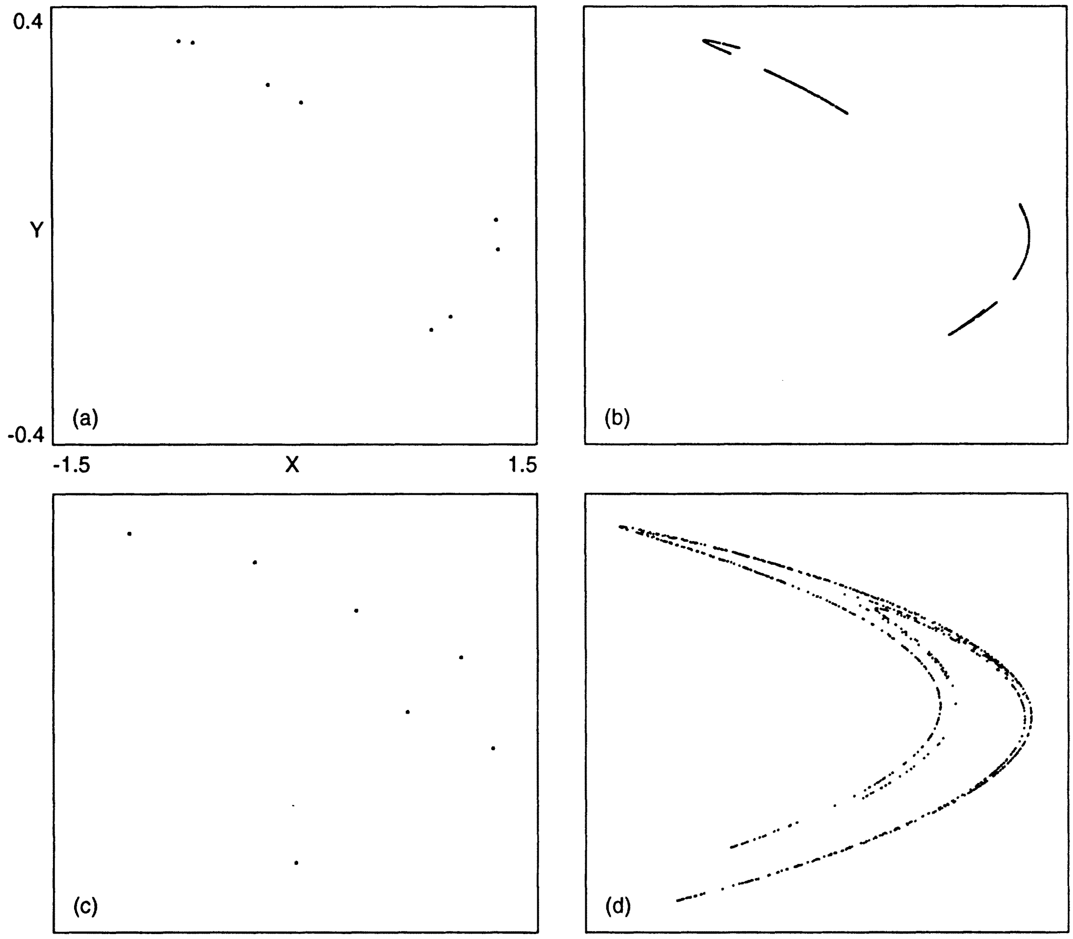
$$x_{n+1}=y_n+1-Ax_n^2,$$
$$y_{n+1}=Bx_n,$$

where A and B are positive constant parameters, and the sequence of output values (x_0,y_0), (x_1,y_1),...(x_n,y_n) are points on the plane. Figure 1 shows a C language implementation of the Hénon equations.

The physical shape of the attractor, hence the global behavior of the orbit, is determined by the values assigned to the constant parameters A and B. Figure 2 shows examples of four attractors. The initial conditions of each of these four orbits differ only in the value assigned to A. For visual clarity, the first 200 points in each orbit, which encompass the initial transient portion of each orbit, have not been included in these plots, because the points that comprise the transient portion by definition do not lie on the attractor. The exact length of the transient depends on the initial conditions, and may be in excess of several hundred iterations for some orbits.

Overall behavior of the Hénon system is controlled primarily by adjusting the value of A. For all A below a certain transition value (the exact value of which

Fig. 2. Four orbits of the Hénon map, plotted to 1,000 iterations; (a) period-8 attractor at A = 1.046; (b) chaotic attractor in four bands at A = 1.076; (c) period-7 attractor at A = 1.302; (d) chaotic attractor at A = 1.4. The coordinate system of each graph is identical. For each orbit, B = 0.3, and the initial point is (0,0).

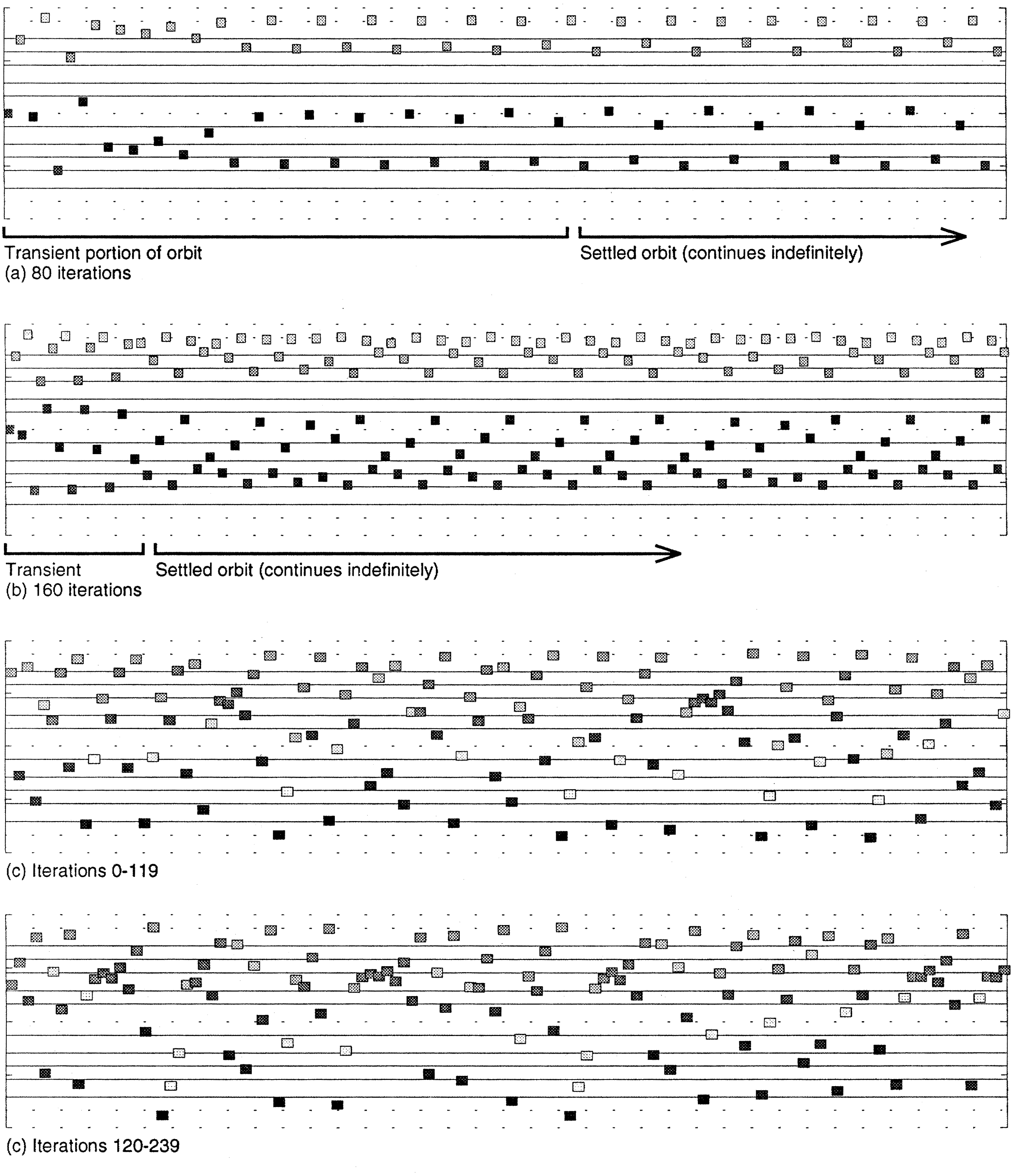


depends on the value of B), all orbits are periodic. At the lowest values of A, this period is 1.0 (the orbit goes to a single-point attractor). The period jumps by powers of 2 as A increases (Fig. 2a). Above the transition value, *most* (not all) orbits are chaotic. If A is assigned too high a value, the orbit escapes quickly to infinity. The region between the transition value and the maximum value of A is called the *chaotic regime.* The distribution of (x, y) values in chaotic orbits may be either separated into narrow bands (Fig. 2b), or continuous throughout the overall region of the attractor (Fig. 2d). Within the chaotic regime exist an infinite number of periodic windows (most of them extremely narrow), that is, regions which produce periodic orbits. Many of the orbits in this category are of an odd-numbered period (Fig. 2c).

The most straightforward mapping of orbit into the musical domain is simply to assign each dimension of the phase space to a different musical parameter. Figure 3 illustrates musical textures made from the orbits plotted in Fig. 2. In each case, values on the x-axis are mapped into a four-octave pitch space symmetrical around middle-C, while values on the y-axis are mapped to an unspecified gray-scale value that could be interpreted variously as a rhythmic value, a dynamic level, or a timbral factor of some sort. In contrast to the plots shown in Fig. 2, the note-event sequences of Fig. 3 do include the initial transient phases of the orbits. The distinction between the transient and the attractor is thus readily observed in each case.

Periodic attractors produce simple oscillating pat-

Fig. 3. Three orbits of the Hénon system, mapped to frequency and an unspecified gray-scale value; (a) period-8 attractor at A = 1.046; (b) chaotic banded attractor at A = 1.076; (c) chaotic attractor at A = 1.4.



(a) 80 iterations

(b) 160 iterations

(c) Iterations 0-119

(c) Iterations 120-239

terns somewhat reminiscent of classic Alberti bass figuration or of contemporary minimalist sequences; see Fig. 3a. Banded chaotic attractors produce textures that at first glance appear to be composed of a regular pattern of note-events. On closer inspection, a pattern of variation within regularity is revealed, with some flexibility in the composition of the repeating sequence; see Fig. 3b. Full-blown chaotic orbits produce the most complex sequences. The distribution of notes within Fig. 3c is clearly not random; rather, it verges on falling into a pattern without actually doing so. The nascent patterned quality of this texture is as easily heard as it is seen. Note in particular the small clumps of notes that occur with *almost* periodic regularity in the upper middle register. The overall impression is that of a complexly hocketed texture, in which there is no clear boundary between the component voices.

The Standard Map: Chaos on a Torus

The standard map is a conservative iterated map. The equations are derived from a model of a pendulum that receives external periodic perturbations (Chirikov 1979; Lichtenberg and Lieberman 1983). The phase space of the standard map is the two-dimensional surface of the unit torus, that is, a torus whose diameter and girth are both 2π. The system is described by the difference equations

$$I_{n+1} = I_n + K \sin \Theta_n,$$

$$\Theta_{n+1} = \Theta_n + I_{n+1},$$

where K is the perturbation parameter, and I and Θ are variables taken modulo 2π in the interval from 0 to 2π. Figure 4, a C language implementation of the standard map, demonstrates the method by which the variables are constrained to the periodic phase space.

Plots of the standard map, such as those shown in Fig. 5, are made by cutting the toroidal phase space both latitudinally and longitudinally to allow its depiction as a planar surface. Keep in mind that the planar surface is periodic along both dimensions; the left and right edges of the plot are identical, as are the top and bottom edges.

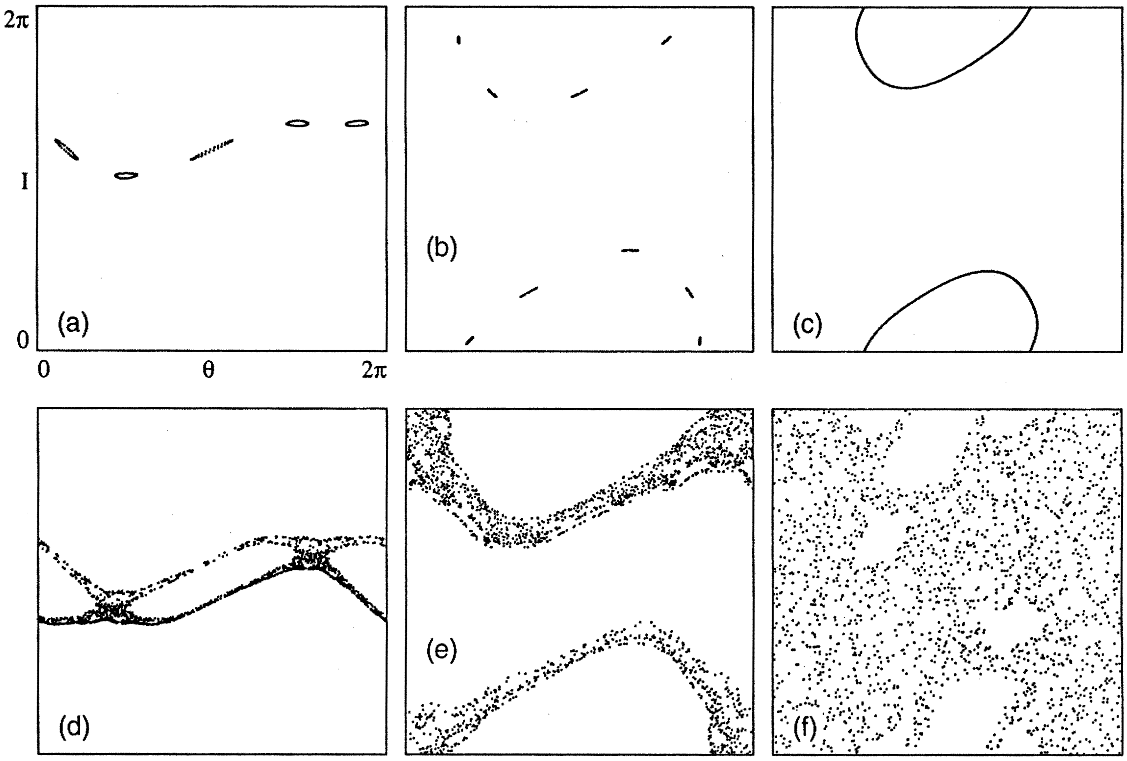
Fig. 4. C language fragment demonstrating an implementation of the standard map. The procedure followed parallels that of Fig. 1, with the addition of the periodic phase space constraint.

```
/* Initialize variables and the single constant
 * parameter K with values which will just produce
 * a chaotic orbit. */
float iota = 0.3,       /* variable */
      theta = 0.3,      /* variable */
      new_iota,
      new_theta,
      K = 1.1,          /* perturbation parameter */
      TWO_PI = 6.283185;

/* Iterate through the equations taking the
 * periodic phase space into account. */
ITERATION_LOOP {
      /* Plot or produce sound. */
   output (theta, iota);
      /* Solve the equations. */
   new_iota = iota + (K * sin (theta));
   new_theta = theta + new_iota;
      /* Assign values for next iteration */
   iota = new_iota;
   theta = new_theta;
      /* Constrain values to the interval between
       * zero and two pi (phase space is a torus).*/
   while (iota < 0.0) iota = iota + TWO_PI;
   while (iota >= TWO_PI) iota = iota - TWO_PI;
   while (theta < 0.0) theta = theta + TWO_PI;
   while (theta >= TWO_PI) theta = theta - TWO_PI;
}
```

An understanding of the effect of the parameter K is critical to understanding the behavior of the standard map. In general, orbits computed from values of K greater than 1 tend to be chaotic, while those computed with K less than 1 tend to follow regular cyclical paths. The transition value $K = 1$ is only an approximation, however, and the precise behavior of a given orbit depends as much on its initial starting point as on the value of K. One crucial difference between cyclical and chaotic orbits is that a chaotic orbit will, given enough time, visit the entire range of values from 0 to 2π along the I-axis, while a cyclical orbit remains confined forever to a smaller range of values along this dimension. The higher the value of K is, the fewer the number of iterations required for a chaotic orbit to wander over the entire range of I. For values of K that are only minimally sufficient to drive the orbit into chaos, the number of iterations required to take an orbit through the entire range of I-values can be enormous—on the order of 100,000 or more. The manner in which the orbit wanders over the range of I is also interesting; it will typically resonate within a confined region for a certain num-

Fig. 5. Six orbits of the standard map. Cyclical orbits (a), (b), and (c) for K = 0.97; chaotic orbits (d), (e), and (f) for K = 1.0, K = 1.05, and K = 2.6, respectively. Initial values of I and Θ differ for each orbit shown. Each orbit is plotted to 2,000 iterations.

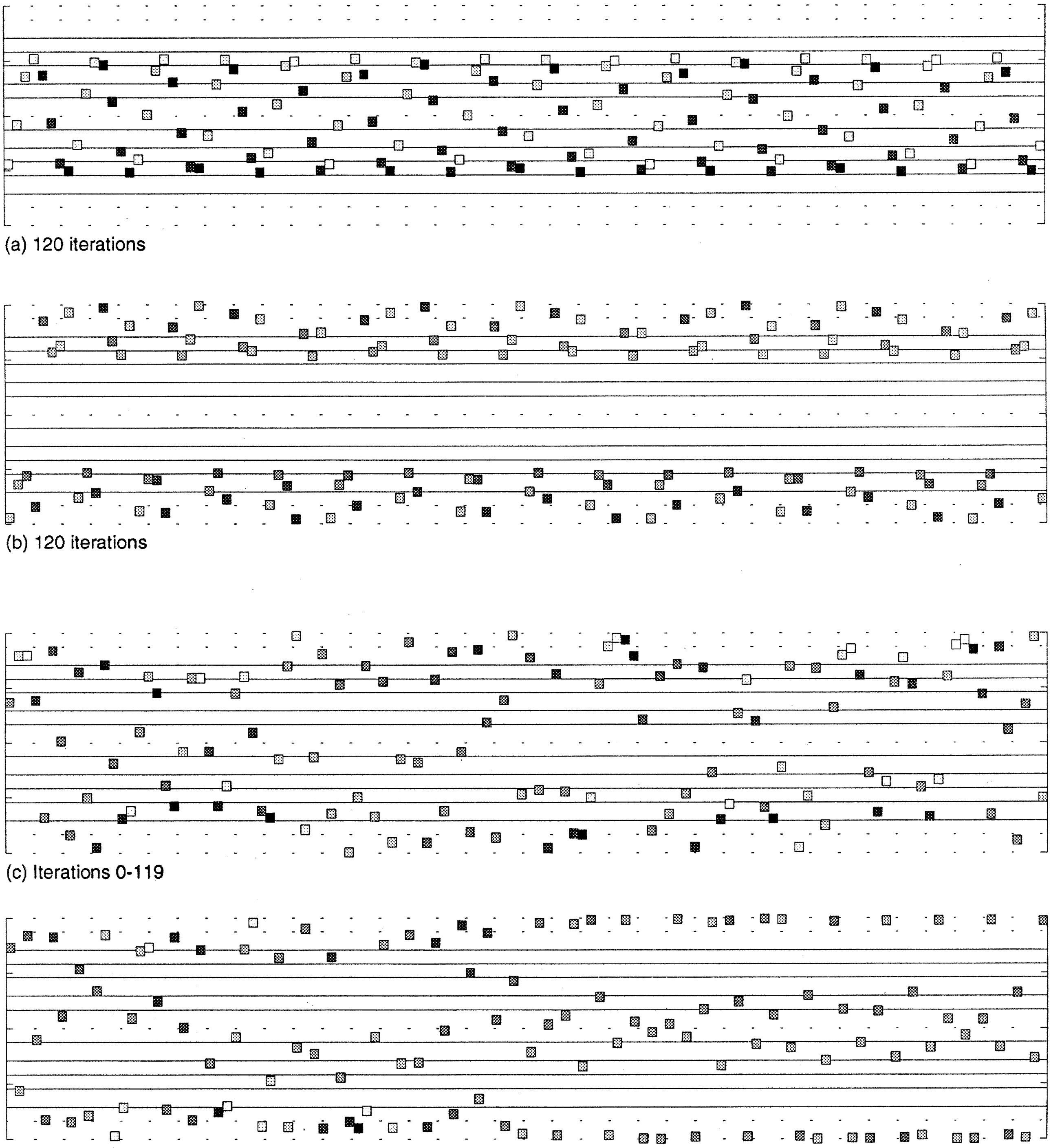


ber of iterations before embarking on a relatively rapid transition to a neighboring region of the phase space, where it again remains confined for an extended period of time. The transition time between these resonances is inversely proportional to the value of *K*. Figure 5 shows several orbits computed from different values of *K* as well as different initial values for *I* and Θ.

Mapping an orbit of the standard map into pitch space can produce interesting results owing to the incongruity of mapping a periodic space into one that is not. An example of this is illustrated in Fig. 6, which shows two complementary mappings of the orbit plotted in Fig. 5c. The frequency/gray-scale domain is the same that was used for the scores in Fig. 3. In Fig. 6a values along the Θ-axis are mapped to pitch, while the I-axis is mapped to gray-scale. In Fig. 6b this orientation is reversed. Keep in mind when comparing the graphic plot with the scores that the phase space wraps around from top to bottom—the two half-ovals in Fig. 5c, for example, actually form a single irregular oval centered at (π,0), (or (π,2π). As the equations are iterated, the succession of points that are generated circles around the oval in a clockwise direction. The large jumps in frequency and gray-scale values observed in Fig. 6a and 6b result when the orbit crosses a boundary at 0 or 2π.

Mappings of highly chaotic sequences of the standard map produce musical textures very different from those shown in Figs. 6a and 6b. Figure 6c is a score generated from an orbit with a relatively high value of *K* = 2.2. There is more internal variability in this sequence than in the textures generated from chaotic orbits of the Hénon map. One reason for this is the relative independence of the two variables, *I* and Θ, of the standard map, as compared to those of the Hénon map, in which the value of *y* is simply the last value of *x* scaled by parameter *B*. Primarily, however, the observed behavioral differences can be attributed to the basic difference between conservative and dissipative systems—the absence of a constraining attractor allows chaotic orbits of conservative

Fig. 6. Two orbits of the standard map, projected as musical scores; (a) cyclical orbit at K = 0.97, I mapped to gray-scale, Θ mapped to frequency; (b) the same orbit with I mapped to frequency and Θ mapped to gray-scale; (c) chaotic orbit at K = 2.2.



(a) 120 iterations

(b) 120 iterations

(c) Iterations 0-119

(c) Iterations 120-239

Fig. 7. C language implementation of the Lorenz equations.A suitably small value is chosen for Δt. Note that the integration procedure implicitly prepares the three variables (x, y, z) for the next iteration.

```
/* Initialize variables and constants with
 * values which produce a chaotic orbit. */
float x = 0.1,              /* variables */
      y = 0.1,
      z = 0.1,
      xdot,                 /* 1st derivatives */
      ydot,
      zdot,
      S = 10.0,             /* constants */
      R = 28.0,
      B = 8.0/3.0,
      delta_t = 0.01;/* for integration */

/* Iterate the equation an arbitrary length of
 * time. The orbit produced is a discrete form
 * of an underlying continuous trajectory. */
ITERATION_LOOP {
      /* Plot or produce sound. */
   output (x, y, z);
      /* Solve the equations. */
   xdot = S * (y - z);
   ydot = (R * x) - y - (x * z);
   zdot = (x * y) - (B * z);
      /* Euler's method of integration. */
   x = x + (xdot * delta_t);
   y = y + (ydot * delta_t);
   z = z + (zdot * delta_t);
}
```

systems to wander unrestricted throughout their phase spaces.

Chaos in Three Dimensions: The Lorenz System

Chaotic systems in three or more dimensions generally take the form of continuous flows, the orbits of which are smooth, unbroken curves. The well-known Lorenz system, or "butterfly attractor" was developed from a simplified model of atmospheric turbulence (Lorenz 1963; Sparrow 1982). It is a dissipative system described by the three differential equations

$$\dot{x} = \sigma\,(y - x),$$

$$\dot{y} = Rx - y - xz,$$

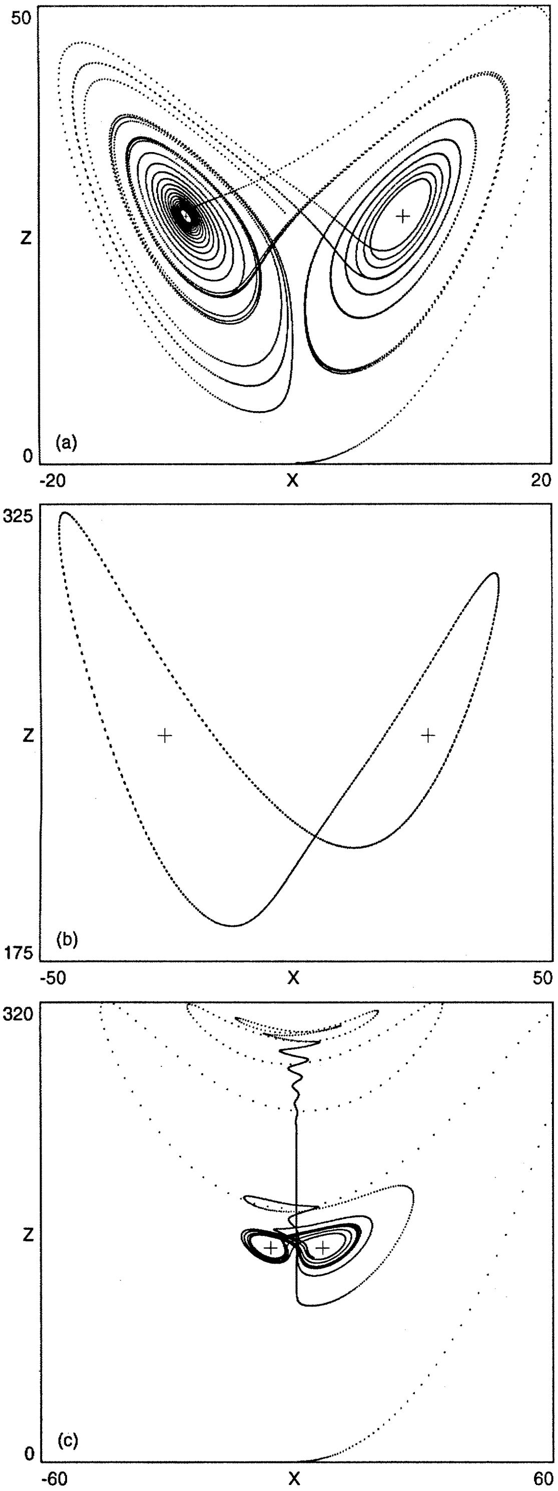
$$\dot{z} = xy - Bz,$$

where σ, R, and B are positive constant parameters, and $\dot{x}$, $\dot{y}$, and $\dot{z}$ are first derivatives with respect to time; that is, $\dot{x} = dx/dt$. The use of a digital computer to solve the equations means that they must be recast into a form that will allow the theoretically continuous orbit to be approximated as a sequence of discrete points separated in time by some small quantization interval, Δt. This process, known as integration, converts the differential equations into what are, essentially, difference equations. Figure 7 illustrates this procedure in C for the Lorenz equations, and employs Euler's method of integration, by far the simplest method available.

Like the Hénon and standard maps, the behavior of the Lorenz system can vary enormously as the values of the constant parameters are changed. For example, if R is between 0 and 1, then all orbits, no matter where they are initiated, are attracted to the origin (0,0,0) and remain there. For values of R between 1 and the *critical value* of approximately 27.74, orbits will be drawn to either of two *stationary points* located at $(\pm\sqrt{B(R\text{-}1)}, \pm\sqrt{B(R\text{-}1)}, (R\text{-}1))$ where they then remain. For supercritical values of R (above 27.74), the stationary points are nonstable, or repelling. This provides the necessary condition for the formation of chaotic orbits, which consist of a series of complexly interwoven sequence of outward spirals away from the stationary points; see Fig. 8a. However, like the chaotic regime of the Hénon map, not all orbits computed with a supercritical R are guaranteed to be chaotic; periodic windows exist at several values of R. One of these is shown in Fig. 8b. Lengthy transients may also be produced by initiating an orbit very far from its eventual attractor; Fig. 8c shows one example.

The presence in the equations of three variables, x, y, and z, allows one to map orbits of the Lorenz system into three musical dimensions simultaneously. If, for example, frequency, loudness, and duration (or the time between successive note attacks) are used, then the sequence generated will consist of an undulating line in which alternations in pitch, note iteration speed, and dynamic level are coordinated by the equations themselves. Much of the interest in this line lies in the details of the transitions of the orbit as it alternates revolutions from one stationary point to the other and back again. At times the motion of the line is subtle, while at other times transitions and shifts of register can be sudden and dramatic.

Fig. 8. Three orbits of the Lorenz system, projected onto the (x,z) plane along the y-axis; (a) chaotic orbit at σ = 10, R = 28, B = 8/3; (b) periodic orbit at R = 150; (c) chaotic orbit with long transient at σ = 10, R = 150, B = 0.25. Crosshairs in each plot show the locations of the stationary points.



Unfortunately, a great deal of space is required to demonstrate adequately the full range of behaviors exhibited by a chaotic orbit of the system. This exercise will therefore be left to the interested reader. In lieu of raw examples, two excerpts from the score of *Dodecanon I* (Bidlack 1990), a twelve-part canon made from a single orbit of the Lorenz system similar to that plotted in Fig. 8a, are shown in Fig. 9. Some idea of the character of the lines generated from the system can be grasped from these excerpts.

The Hénon-Heiles System: Chaos in Four Dimensions

The Hénon-Heiles system was originally introduced as a simplified model of the motion of an individual star within a galactic gravitational field (Hénon and Heiles 1964; Gustavson 1966). It is a conservative system described by the differential equations of motion:

$$\ddot{x} = -x = 2xy,$$

$$\ddot{y} = -y = x^2 + y^2,$$

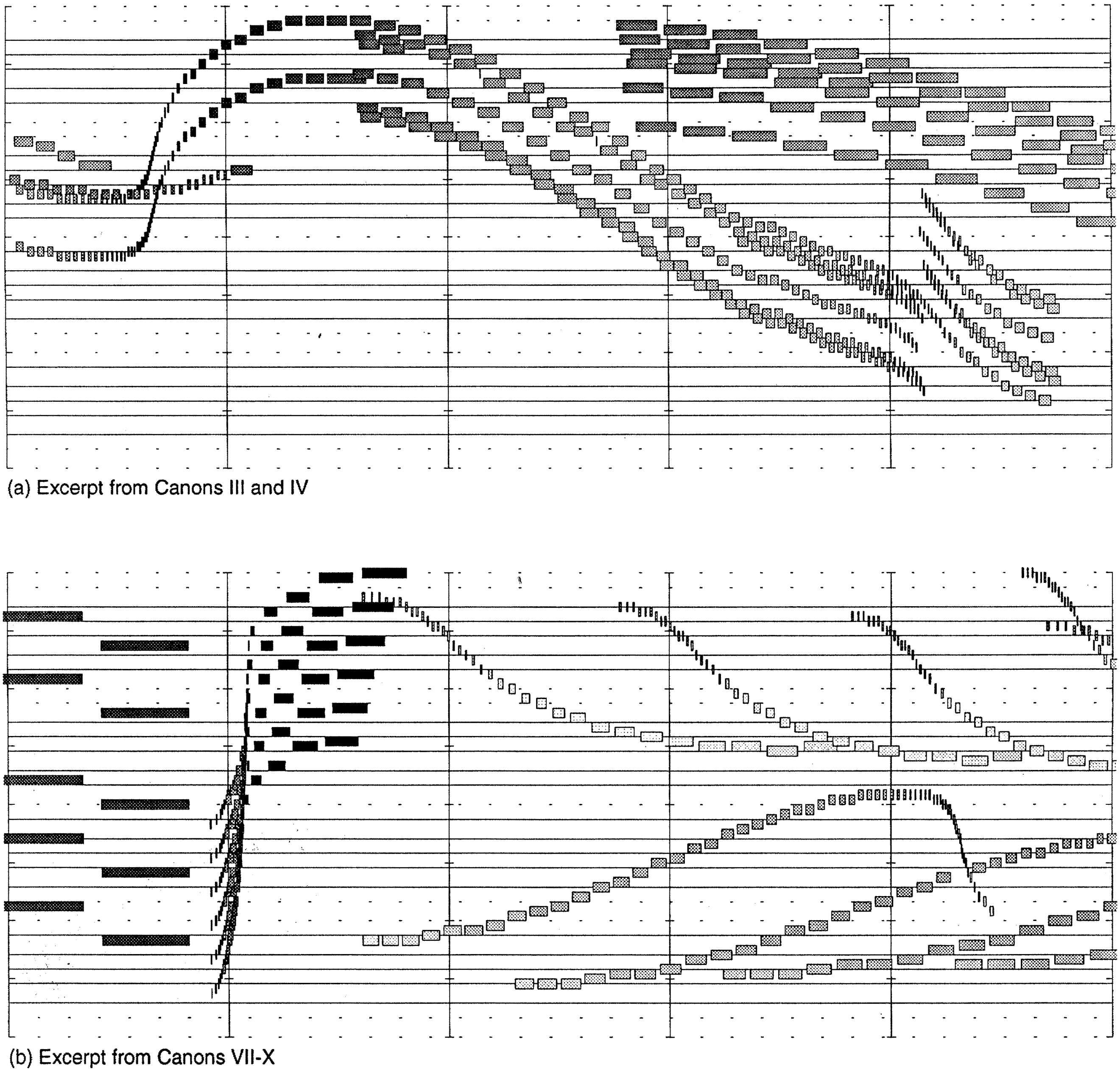
where $\ddot{x}$ and $\ddot{y}$ are second derivatives of x and y with respect to time. Since first derivatives can be inferred from the presence of second derivatives in the equations, one may think of the system as having four variables: x, y, $\dot{x}$, and $\dot{y}$. These variables are related to one another by another equation describing the total energy, E, of the system:

$$E = \frac{1}{2}\left(x^2 + y^2 + 2x^2y - \frac{2}{3}y^3\right) + \frac{1}{2}(\dot{x}^2 + \dot{y}^2).$$

Typically, the energy equation is used to set the initial values of the variables, and an orbit is then calculated using the two equations of motion.

The value chosen for E provides a means by which the overall behavior of the system can be roughly predicted. The maximum allowable value of E is ⅙. At this energy level, almost all orbits of the system are highly chaotic. (Above this value, all orbits escape to infinity—the star leaves the galaxy!) As E is decreased, the behavior of the system as a whole becomes less chaotic, until, at energy levels of 1⁄12 and below, all orbits follow smooth, cyclical trajectories.

Fig. 9. Two excerpts from Dodecanon I, *for MIDI synthesizers. The bar lines mark 5-second intervals, with rhythm and duration indicated proportionally. Velocity is indicated by the gray- scale value of each note (the darker, the louder). The canons are constructed from a single orbit of the Lorenz system.*


(a) Excerpt from Canons III and IV

(b) Excerpt from Canons VII-X

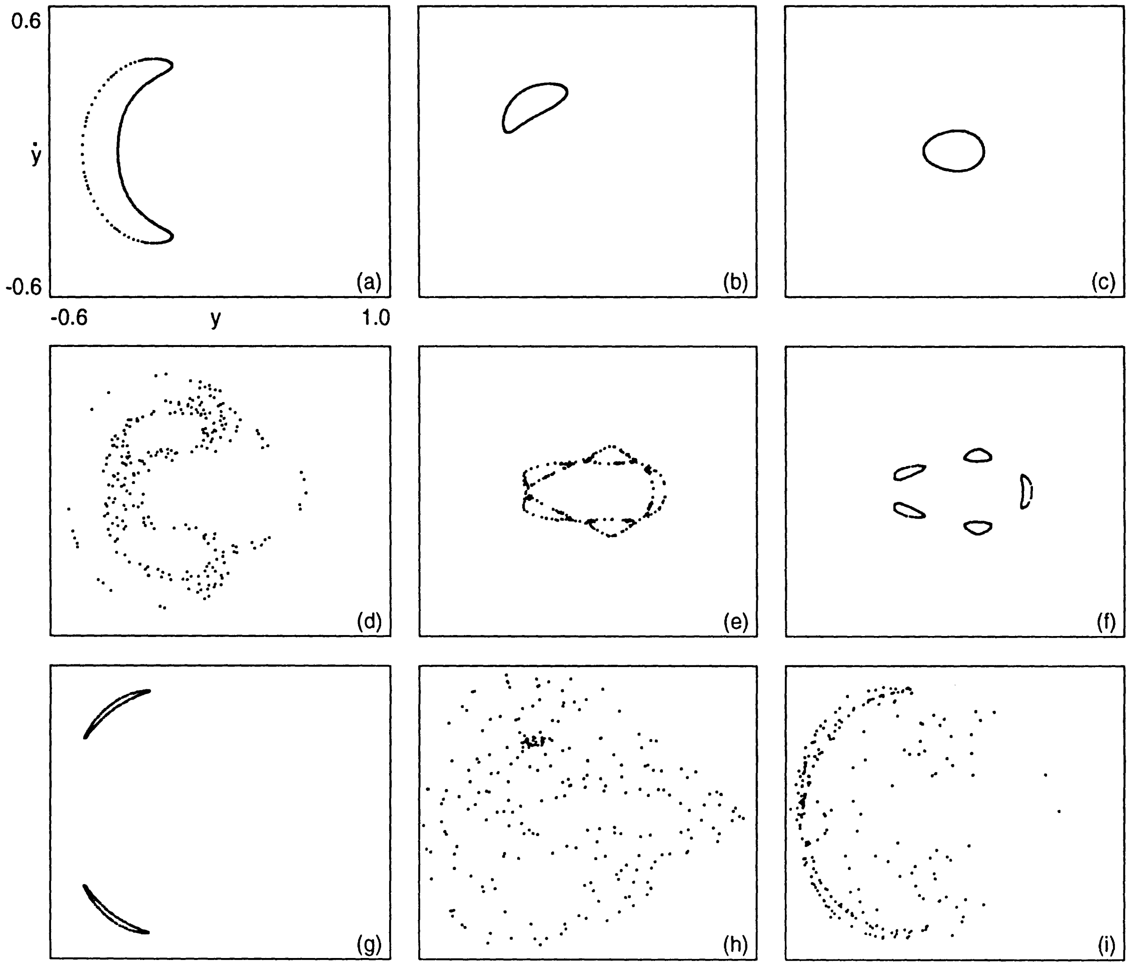
Fig. 10. C language implementation of the Hénon-Heiles system. To produce the Poincaré sections of Fig. 11, additional code must be added to the iteration loop to detect zero-crossings of xdot. The output() routine is then called only when such a zero-crossing occurs (xdot will be equal to or very close to zero in these cases, so its value may be ignored).

```
/* Initialize variables and the energy constant E to
 * produce a chaotic orbit. The first derivatives
 * xdot and ydot are also considered to be variables
 * which may be plotted or mapped into a musical
 * space. */
float x = 0,            /* variables */
      y = 0.1,
      xdot,             /* first derivatives */
      ydot = 0.2,
      E = 1.0/6.0,      /* max allowable energy */
      xdotdot,          /* second derivatives */
      ydotdot,
      xdot_squared,     /* for solution of xdot */
      delta_t = 0.01;   /* for integration */

/* The energy equation is solved for xdot. The first
 * stage of the solution gives its squared value. If
 * no real square root exists, then the initial values
 * chosen are rejected (they imply on an orbit which
 * cannot exist). Otherwise, an initial value for
 * xdot can be calculated. */
xdot_squared =
        (2.0*E)-(y*y)+((4.0/3.0)*y*y*y)-(ydot*ydot);
if (xdot_squared < 0.0) {
      printf(“Sorry, bad initial conditions!”);
      exit();
} else
      xdot = sqrt(xdot_squared);

/* Now that the initial conditions have been set, the
 * equations can be iterated. As with the Lorenz system,
 * the orbit produced is a discrete form of an underlying
 * continuous trajectory. */
ITERATION_LOOP {
      /* Plot or produce sound. */
   output (x, y, xdot, ydot);
      /* Solve the equations. */
   xdotdot = -x - (2.0 * x * y);
   ydotdot = -y - (x * x) + (y * y);
      /* Double integration as per Euler. */
   xdot = xdot + (xdotdot * delta_t);
   ydot = ydot + (ydotdot * delta_t);
   x = x + (xdot * delta_t);
   y = y + (ydot * delta_t);
}
```

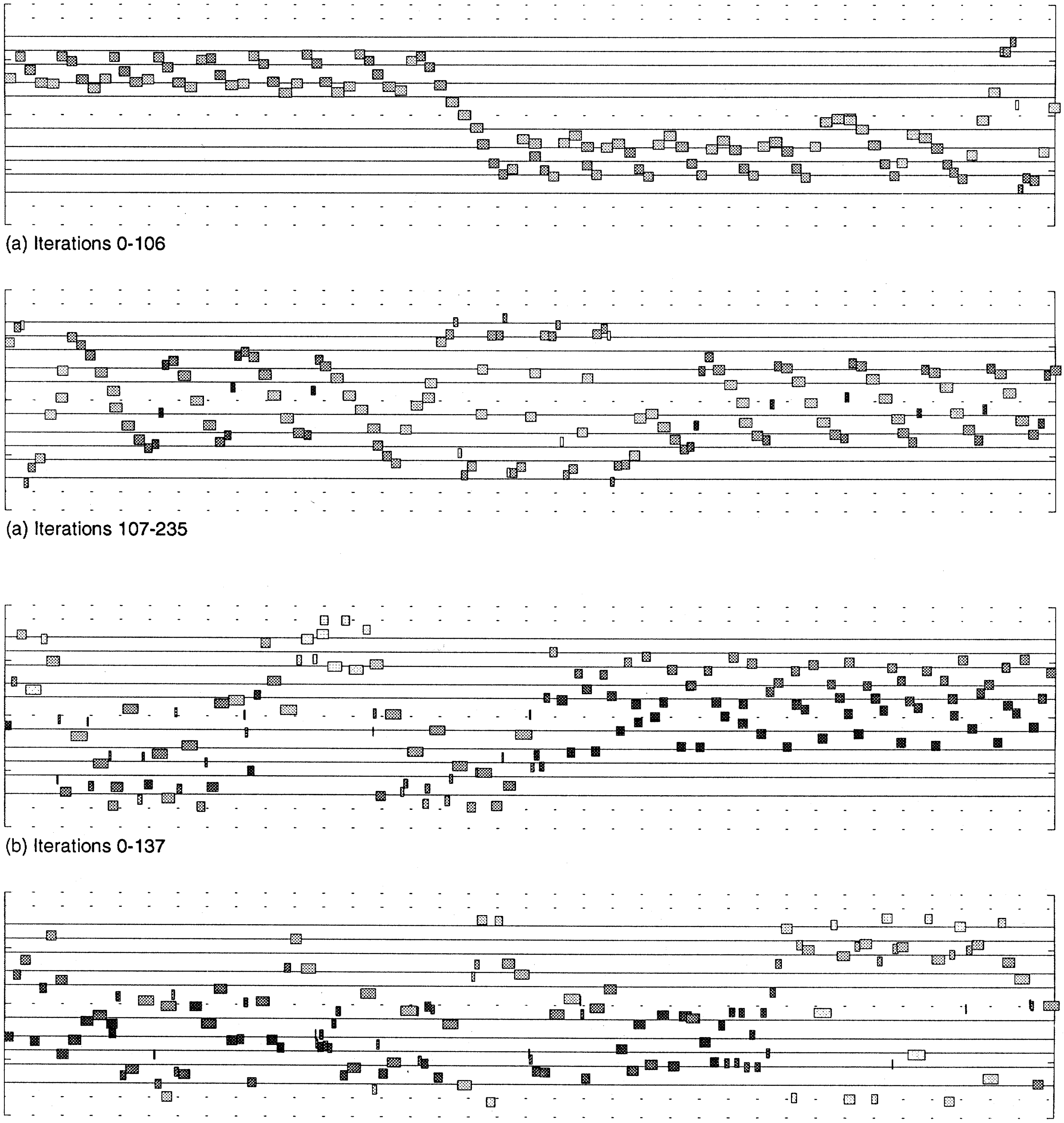
Fig. 11. Nine Poincaré sections of orbits of the Hénon-Heiles system, taken across the x= 0 plane and projected onto the (y, ẏ) plane. The energy level, E, is 1/12 for the top row, 1/8 for the middle row, and 1/6 for the bottom row. The initial values of y and ẏ differ for each orbit.



Initial conditions of the system are typically specified by choosing a value for E and for two of the variables, y and $\dot{y}$, and setting x to zero. The energy equation is then solved for the initial value of $\dot{x}$. It is possible that the values chosen will not allow a real solution for $\dot{x}$ to be found. This is an indication that the orbit specified either cannot exist, or will immediately escape to infinity. Figure 10 demonstrates this procedure in the C language, and uses Euler's simple method of integration as well.

Continuous orbits of the Hénon-Heiles system through the four-dimensional $(x, y, \dot{x}, \dot{y})$ space are not without interest; however, of perhaps greater musical utility are Poincaré sections taken from these orbits. A Poincaré section is made by plotting the points at which the continuous orbit intersects a plane bisecting the phase space of the system. The resulting surface-of-section or cross section reveals a great deal about the dynamics of a particular orbit. Typically, the bisecting plane is placed at $x = 0$. This is accomplished in software by adding a few lines of code in the iteration loop to track the value of x through the course of the orbit. A conditional statement is added to call the *output()* routine only when x changes from positive to negative (or vice versa). Since the x-value on each of these occasions is by definition zero (or very close to it), the number of dimensions of each point in the surface-of-section is essentially reduced from four to three: y, $\dot{y}$, and $\dot{x}$. The process of making a Poincaré section

Fig. 12. Two Poincaré sections of the Hénon-Heiles system, mapped into musical space; (a) chaotic orbit at E = 1/8, with variables y, ẏ, and ẋ mapped to gray-scale, frequency, and duration, respectively; (b) chaotic orbit at E = 1/6, with variables y, ẏ, and x mapped to frequency, duration, and gray-scale, respectively. An "iteration" is counted every time the orbit punctures the plane at x = 0.



(a) Iterations 0-106

(a) Iterations 107-235

(b) Iterations 0-137

(b) Iterations 138-272

transforms continuous orbits of the Hénon-Heiles system into sequences resembling those produced by an iterated map. An analogous process can, in fact, be applied to any continuous system (for a discussion of Poincaré sections of the Lorenz system, see Sparrow 1982 and Bidlack 1990).

Figure 11 shows several Poincaré sections of the Hénon-Heiles system projected onto the $(y, \dot{y})$ plane. In many ways, these sections resemble orbits of the standard map. Cyclical orbits may appear as unbroken ellipses (Figs. 11a, b, and c), or be broken into "islands" (Figs. 11f and g). Chaotic orbits (Figs. 11d, e, h, and i), if left to run long enough, will eventually fill in the entire phase space.

Although the Poincaré sections shown in Fig. 11 plot only two dimensions, it should be kept in mind that these sequences are really composed of triplets of values. Musical textures generated from these sections may take advantage of this to employ simultaneous mappings of, for example, pitch, dynamic level, and rhythm. Mappings generated from cyclical orbits of the Hénon-Heiles system strongly resemble those generated from cyclical orbits of the standard map, with the added interest of the third variable. Much more interesting are chaotic sequences. Figs. 12a and 12b are scores generated from the Poincaré sections plotted in Figs. 11d and 11h, respectively.

Aliquis utens hoc caveat

User beware! One of the most distinctive characteristics of nonlinear dynamical systems is a behavior known as *sensitive dependence on initial conditions.* It is because of this that relatively small quantitative differences in the input (the initial conditions) often produce very large qualitative differences in the output. This situation applies equally well to the ongoing calculation of a given orbit. The round-off errors that inevitably accumulate in the iteration of the equations act as minute perturbations on the calculated trajectory. The ramifications of this condition are both subtle and profound—an orbit computed with single-precision floating-point numbers will eventually diverge completely from an orbit calculated from the same initial conditions but computed with double-precision floating-point values. Alternatively, equivalent programs, compiled and run on different processors, are virtually guaranteed to produce different orbits. For continuous flows, the use of an alternative method of integration or a slightly different Δt in the calculations can drastically alter the output. One hopes this will not be considered a detraction, but part of the intrigue of nonlinear dynamical systems.

Conclusions

The raw musical sequences produced by chaotic orbits of dissipative and conservative systems exhibit marked differences. A dissipative system, once it has settled onto its attractor, maintains a relatively constant behavior (with nonetheless interesting details). An excerpt chosen at random from such a sequence will look very much like any other section of the same sequence. Conservative sequences, on the other hand, can exhibit much less internal consistency, and are marked by sudden changes in texture and range of values. From such a sequence, it is often easy to select excerpts that bear only familial resemblance to one another. This tends to give conservative textures a more localized focus, as well as a greater variation in the unfolding of their material.

The great attraction of nonlinear dynamical systems for compositional use is their natural affinity to the behaviors of phenomena in the real world, coupled with the mechanical efficiency of their computation and control. Chaotic systems offer a means of generating a variety of raw materials within a nonetheless globally consistent context. Chaotic sequences embody a process of transformation, the internal coherence of which is ensured by the rules encoded in the equations.

References

Bidlack, R. 1990. "Music from Chaos: Nonlinear Dynamical Systems as Generators of Musical Materials." Ph.D. diss. University of California, San Diego.

Chirikov, B. 1979. "A Universal Instability of Many-Dimensional Oscillator Systems." *Physics Reports (Review section of Physics Letters)* 52(5):263–379.

Devaney, R. L. 1988. "Fractal Patterns arising in Chaotic Dynamical Systems." In H.-O. Peitgen and P. H. Richter, eds. *The Science of Fractal Images.* New York: Springer-Verlag.

Di Scipio, A. 1990. "Composition by Exploration of Nonlinear Dynamical Systems." In *Proceedings of the 1990 International Computer Music Conference,* San Francisco: International Computer Music Association, pp. 324–327.

Eckmann, J.-P., and D. Ruelle. 1985. "Ergodic Theory of Chaos and Strange Attractors." *Reviews of Modern Physics* 57(3):617–656.

Gogins, M. 1991. "Iterated Function Systems Music." *Computer Music Journal* 15(1):40–48.

Gustavson, F. G. 1966. "On Constructing Formal Integrals of a Hamiltonian System Near Equilibrium Point." *The Astronomical Journal* 71(8):670–686.

Hénon, M. 1976. "A Two-Dimensional Mapping with a Strange Attractor." *Communications in Mathematical Physics* 50:69–77.

Hénon, M., and C. Heiles. 1964. "The Applicability of the Third Integral of Motion: Some Numerical Experiments." *The Astronomical Journal* 69(1):73–79.

Lichtenberg, A. J., and M. A. Lieberman. 1983. *Regular and Stochastic Motion.* Applied Mathematical Sciences, Volume 38. New York: Springer-Verlag.

Lorenz, E. N. 1963. "Deterministic Nonperiodic Flow." *Journal of the Atmospheric Sciences* 20:130–141.

Moon, F. C. 1987. *Chaotic Vibrations: An Introduction for Applied Scientists and Engineers.* New York: John Wiley and Sons.

Pressing, J. 1988. "Nonlinear Maps as Generators of Musical Design." *Computer Music Journal* 12(2):35–46.

Sparrow, C. 1982. *The Lorenz Equations: Bifurcations, Chaos and Strange Attractors.* Applied Mathematical Sciences, Volume 41. New York: Springer-Verlag.

Thompson, J. M. T., and H. B. Stewart. 1986. *Nonlinear Dynamics and Chaos.* Chichester: John Wiley and Sons.

Truax, B. 1990. "Chaotic Non-linear Systems and Digital Synthesis: An Exploratory Study." In *Proceedings of the 1990 International Computer Music Conference,* San Francisco: International Computer Music Association, pp. 100–103.

Waschka, R., and A. Kurepa. 1989. "Using Fractals in Timbre Construction: An Exploratory Study." In *Proceedings of the 1989 International Computer Music Conference,* San Francisco: International Computer Music Association, pp. 332–335.