Chaos, Fractals, and Tom Stoppard's *Arcadia*

Robert L. Devaney

Department of Mathematics
Boston University
Boston, MA 02215

Tom Stoppard's wonderful play, *Arcadia*, offers teachers of both mathematics and the humanities the opportunity to join forces in a unique and rewarding way. The play features not one but two mathematicians, and the mathematical ideas they are involved with form one of the main subthemes of the play. Such contemporary topics as chaos and fractals form an integral part of the plot, and even Fermat's Last Theorem and the Second Law of Thermodynamics play important roles.

The play is set in two time periods, the early nineteenth century and the present, in the same room in an English estate, Sidley Park. As the play opens, we meet Thomasina, a young thirteen year old girl who struggles with her algebra and geometry under the watchful eye of her tutor, Septimus Hodge. But Thomasina is not your typical mathematics student; as becomes clear as the play unfolds, she is a prodigy who not only questions the very foundations of her mathematical subjects, but also sets about to change the direction of countless centuries of mathematical thought. In the process, she invents "Thomasina's geometry of irregular forms" (aka fractal geometry), discovers the second law of thermodynamics, and lays the foundation for what is now called chaos theory.

In the modern period, we meet Valentine, a contemporary mathematical biologist who is attempting to understand the rise and fall of grouse populations using iteration. As luck would have it, Valentine is heir to Sidley Park and part of his inheritance is a complete set of game books that go back to Thomasina's time. These books detail the precise number of grouse shot at the estate each year. Gradually, he becomes aware of some of the old mysteries surrounding Sidley Park, including Thomasina's discoveries, and this sets the stage for a unique series of scenes that hop back and forth between the nineteenth century and the present.

At the same time, Valentine's friend, Hannah Jarvis, is attempting to understand some of the mysteries surrounding some of the historical events that occurred around the period that Thomasina lived at Sidley Park. Thomasina had died in a tragic fire the night before her seventeenth birthday, and right about that time, a hermit moved into a cottage on the estate and lived there for many years. As part of her research, Hannah finds out that this hermit had spent his entire life working out what appeared to her to be incomprehensible mathematical equations. With Valentine's help, Hannah comes to realize that the hermit turns out to be Septimus, who, after Thomasina's death, spent the rest of his life trying to push her ideas forward.

Mathematics is not the only theme of this play, but the ideas of regular versus irregular geometry or order versus chaos seem to pervade all of the other events occurring at Sidley Park. We are thrust into a debate about emerging British landscape styles featuring the orderly classical style versus the irregular, "picturesque" style. Valentine and Hannah methodically proceed to uncover Sidley Park's secrets, in stark contrast to her nemesis, Bernard Nightingale, who jumps from one theory to another with reckless abandon. Indeed, the entire play pits the rationalism of Newton against the romanticism of Lord Byron.

1 Thomasina's Geometry of Irregular Forms

Thomasina does not like Euclidean geometry. Early in the play she chides her tutor, Septimus, "Each week I plot your equations dot for dot, xs against ys in all manner of algebraical relation, and every week they draw themselves as commonplace geometry, as if the world of forms were nothing but arcs and angles. God's truth, Septimus, if there is an equation for a bell, then there must be an equation for a bluebell, and if a bluebell, why not a rose?" So she decides to abandon classical Euclidean geometry in order to discover the equation of a leaf.

Years later, Hannah discovers Thomasina's workbooks in which she has written, "I, Thomasina Coverly, have found a truly wonderful method whereby all the forms of nature must give up their numerical secrets and draw themselves through number alone." Hannah asks Valentine how she does this. Val explains that she uses "an iterated algorithm."

"What's that?" Hannah inquires. With the precision that only a mathematician can muster, Val responds "It's an algorithm that's been....iterated."

Then the fun begins. Val goes on to explain that an algorithm is a recipe, that if you knew the recipe to produce a leaf, you could then easily iterate the algorithm to draw a picture of the leaf. "The math isn't difficult. It's what you did at school. You have an x and y equation. Any value for x gives you a value for y. So you put a dot where it's right for both x and y. Then you take the next value for x which gives you another value for y......what she's doing is, every time she works out a value for y, she's using *that* as her next value for x. And so on. Like a feedback.... If you knew the algorithm, and fed it back say ten thousand times, each time there'd be a dot somewhere on the screen. You'd never know where to expect the next dot. But gradually you'd start to see this shape, because every dot will be inside the shape of this leaf."

2 The Chaos Game

What Thomasina has discovered and what Val is trying to explain is what is now commonly called the "chaos game," or, more precisely, an iterated function system. The game proceeds in its simplest formulation as follows. Place three dots at the vertices of a triangle. Color one vertex red, one green, and one blue. Then take a die and color two faces red, two green, and two blue.

To play the game, you need a *seed*, an arbitrary starting point in the plane. The algorithm is: Roll the die, then depending upon which color comes up, move your point half the distance toward the appropriate colored vertex. Then iterate, i.e., repeat this process, using the terminal point of the previous move as the seed for the next. Do not plot the first 15 points generated by this algorithm, but after these few initial moves, begin to record the location of each and every point.

Students who have not seen this game before are always surprised and amazed at the result. Most expect the algorithm to yield a blur of points in the middle of the triangle. Some expect the moving point to fill the whole triangle. But the fact is, the result is anything but a random mess: the resulting picture is one of the most famous of all fractals, the Sierpinski triangle. See Figure 1.

The important point about this object being a fractal is that it is a self-similar set. Then, using this self-similarity, we can "go backwards." That is, just by looking as the self-similar features of this set, we can read off the rules of the algorithm that allowed us to generate this set. For the Sierpinski triangle basically consists of three self-similar pieces in which each length is half the length of the corresponding length in the original triangle. And these are precisely the numbers that came up when we generated this image: three vertices, and we moved half one-half the distance to each vertex at each iteration.

For leaf-making purposes (which we will describe later), it is best to rework this algorithm in a slightly different form. Begin with a square in the plane, and put the red vertex in the center of the top side of the square, and the other two vertices at the lower vertices. The algorithm then linearly contracts all points in the original square into one of three smaller subsquares

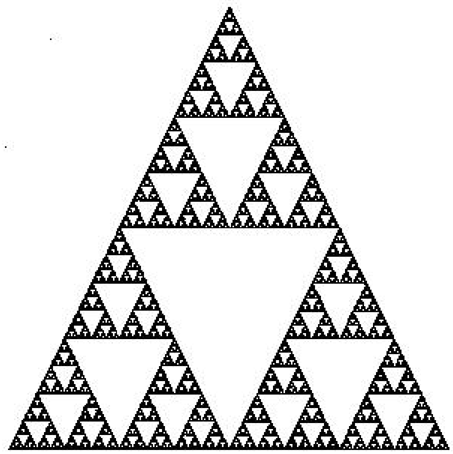


Figure 1: The Sierpinski triangle.

by moving half the distance toward the vertex in each subsquare as in Figure 2. For example, if the origin in the plane is located at the lower left vertex, then the contraction that takes the square into the region A is simply $(x, y) \mapsto (0.5x, 0.5y)$.

3 Other Chaos Games

As another example of Thomasina's algorithm, start with six points arranged at the vertices of a regular hexagon. Number them from one to six and erase the colors on the die. Beginning with an initial seed in the hexagon, roll the die and then move the starting point two-thirds of the distance toward the appropriate vertex. For what comes later, we think of this as contracting the original distance to the chosen vertex by a factor of three. That is, three is the contraction ratio for this game rather than two as in the case of the Sierpinski triangle.

Then iterate this procedure. As before, do not record the first 15 or so iterations, but plot the rest. The result is again anything but a random mess: It is the Sierpinski hexagon. See Figure 3. Note that this fractal image is not quite the lifelike image that Thomasina promised us, but there is a hint of what is to come. Look at the boundary of the innermost white region in the Sierpinski hexagon (or, in fact, the boundary of any internal white region).

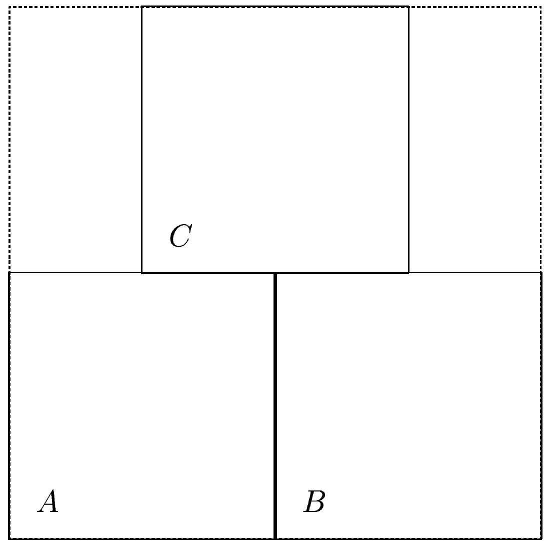


Figure 2: The three contractions to produce the Sierpinski triangle. Each move of the chaos game contracts the large outermost square linearly into one of the subsquares.

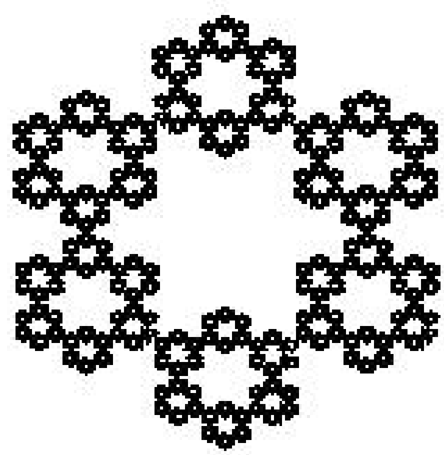


Figure 3: The Sierpinski hexagon.

Note how this curve resembles a snowflake. Indeed, this boundary curve is the von Koch snowflake curve, another very famous fractal.

This particular algorithm can also be expressed in terms of linear contractions of a given square. Start with a square and place six points in the square so that they form the boundary of a regular hexagon. Geometry exercise: Where should these points be placed if vertex number one lies in the middle of the left side of the square? At each iteration, contract all points in the original square toward the appropriate vertex. The image is, in each case, a square exactly one-third the size of the original square and located as shown in Figure 4.

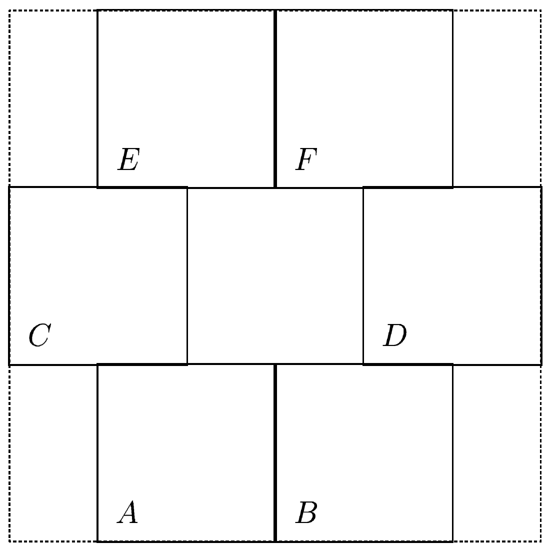


Figure 4: The six contractions to produce the Sierpinski hexagon.

And again we can use the self-similar nature of the Sierpinski hexagon to go backwards. For this set consists of six self-similar pieces, each of which is one-third the size of the original hexagon. And those were the numbers that comprised the rules of this game: six vertices and we contracted the distance each time we rolled by one-third.

As another example, suppose we play the chaos game with eight vertices arranged as follows in a square: four lie at the corners of the square and the other four at the midpoints of the sides of the square. With a contraction ratio of three as in the case of the hexagon, we obtain another famous fractal, the Sierpinski carpet depicted in Figure 5.

We can complicate things a bit by allowing rotations (and reflections) in the rules of the game. For example, suppose we start with the original

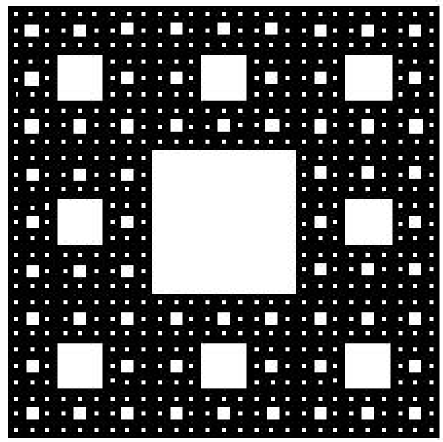
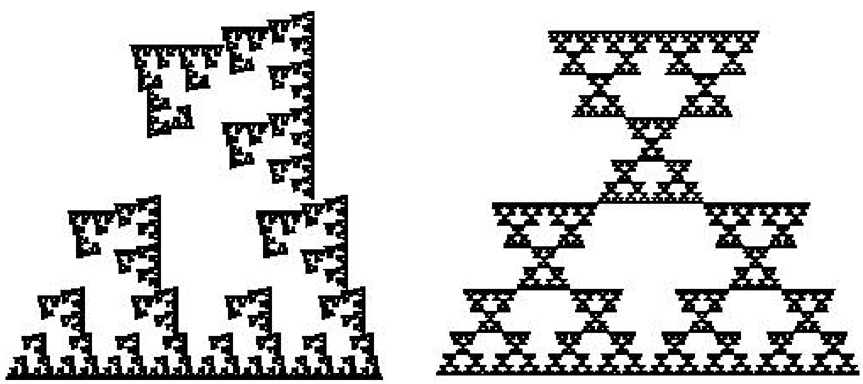


Figure 5: The Sierpinski carpet.



A. B.

Figure 6: The two fractals obtained by adding rotations to the rules.

configuration that produced the Sierpinski triangle, but we now change one of the rules as follows. When we roll the color corresponding to the topmost vertex, we first move the given point half the distance toward that vertex as before, but then we rotate the image point 90 degrees in the counterclockwise direction about the top vertex. (Here we choose a larger square around the chosen vertices than we did originally.) The fractal that emerges is shown in Figure 6A. Note again that we can go backwards, since the top self-similar piece of the resulting image is exactly half the size of the entire fractal, but it is now rotated by 90 degrees, while the other two self-similar pieces are also half the size but are not rotated. If we change this rule so that the rotation is 180 degrees about the top vertex, we obtain the image displayed in Figure 6B. This fractal consists of three self-similar pieces, each of which is half the size of the entire set, but the top piece is rotated by 180 degrees while the bottom two pieces are not rotated. Again, we can go backwards.

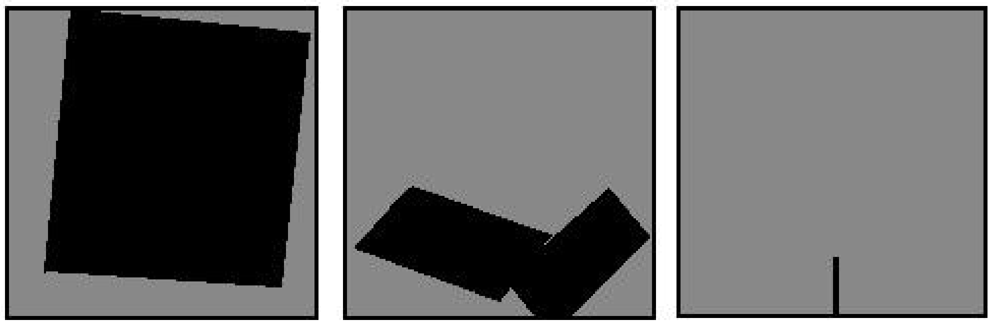
4 Thomasina's geometry

How did Thomasina produce an algorithm that yields a natural form? We will illustrate this with the simplest case, a fern. A leaf is a little more difficult and a little less spectacular than a fern. We need to describe an algorithm that, when iterated as in the chaos game, yields an image of a fern. The fern we will produce is often called the *Barnsley fern* after the mathematician who popularized this procedure [B].

To do this we start with a square as before. We will describe four linear contractions on this square. Unlike the previous two examples, these contractions will involve more than just simple contractions and rotations; they will involve more general linear transformations. Here is the first operation: squeeze and distort the square linearly so that its image appears as in Figure 7A. Note that the square is compressed a bit from the bottom and from both sides, and then rotated a little. Figure 7B displays the effects of the next two contractions. The left hand parallelogram is obtained by first shrinking the square to a rectangle, then shearing and rotating to the left. The second is obtained in similar fashion, except that the square is first flipped along its vertical axis, and then contracted, sheared, and rotated to the right. In Figure 7C we see the final contraction: the entire square is crushed to a line segment in the horizontal direction, then compressed again in the vertical direction to yield the short vertical line segment indicated.

Each of these rules can be described concisely using some matrix algebra. In section 6, we give exact formulas for each of these transformations as well as a brief discussion of where they come from.

Now we play the chaos game with these rules as the four constituent moves. However, instead of randomly choosing a particular contraction with equal probabilities, we will choose the rules to apply with differing probabilities. We will apply the first contraction with the highest probability, namely



A. B. C.

Figure 7: The four contractions that generate the Barnsley fern.

85% of the time. Contractions 2 and 3 will be invoked with probability .07, and the final contraction will be called with probability only .01. When this algorithm is carried out using a computer, the dots slowly fill the screen to reveal a lifelike image of a fern, as illustrated in Figure 8.

5 Why Thomasina's algorithm works

The fact that the above algorithm works is no mystery; it is exactly the same as in the previous examples of chaos games. For we can decompose a fern into "self-similar" copies of itself. Look closely at Figure 8: Do you see several pieces of the fern that resemble the entire fern, only in miniature? If you remove the two lowest fronds of the fern and a piece of the stem, then what remains is more or less an exact copy of the original fern, only slightly smaller. Indeed, we can obtain this smaller piece of the fern by taking the entire fern and applying our first contraction to it. That is, we compress the fern from both sides and the bottom and rotate a bit to move it onto the slightly smaller upper piece.

Also, note how the two lowest fronds of the fern are arranged. We may obtain the left hand frond by compressing the original fern by a much larger amount, and then rotating to the left. The right hand frond is obtained by first flipping the entire fern along its vertical axis, then contracting and rotating to the right. Finally, the piece of the stem that was removed can be obtained by squashing the entire fern to the center and then down, exactly

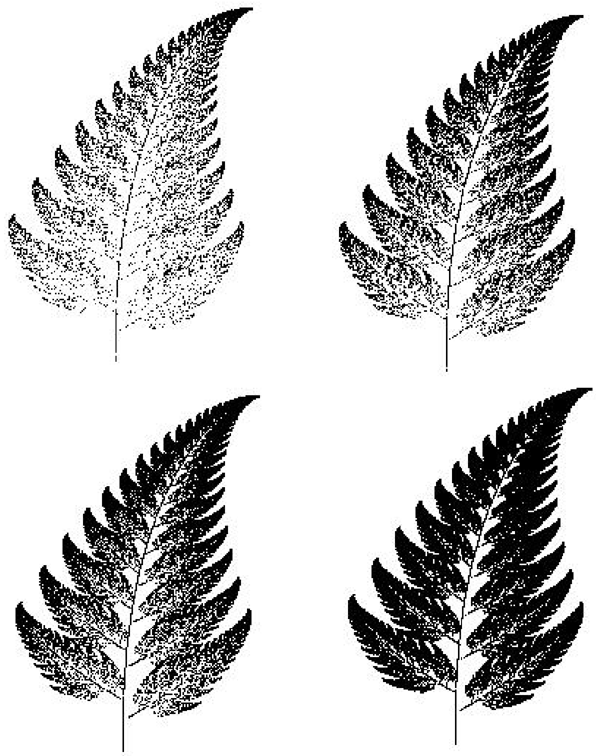


Figure 8: The results of the chaos game played with 10,000, 30,000, 50,000, and 150,000 iterations. As Valentine said to Hannah: “If you knew the algorithm, and fed it back say ten thousand times, each time there'd be a dot somewhere on the screen. You'd never know where to expect the next dot. But gradually you'd start to see this shape, because every dot will be inside the shape of this (leaf).”

our contraction number 4.

So we have divided the fern into four “self-similar” copies of itself, i.e., copies that can be obtained by applying the linear rules above. The mathematical theorem behind all of this says that, when we iterate the above rules, the resulting image is exactly what we started with, the fern. We use different probabilities here so that the density of points will be more or less even when the iteration is complete (there are many more points that need to be drawn in the image region given by contraction number 1 than that given by contraction number 4).

6 The Formulas

Since the four contractions that produce the fern are affine transformations of the plane, they may be encoded using matrices. Each transformation is of the form

$$V_{\text{new}} = A \cdot V_{\text{old}} + W$$

where V_{old} is the vector representing the seed, V_{new} is the new position of the point, A is a 2×2 matrix, and W is a constant vector. For example, contraction 1 is given by

$$\begin{pmatrix} x_{\text{new}} \\ x_{\text{old}} \end{pmatrix} = \begin{pmatrix} 0.85 & 0.04 \\ -0.04 & 0.85 \end{pmatrix} \cdot \begin{pmatrix} x_{\text{old}} \\ y_{\text{old}} \end{pmatrix} + \begin{pmatrix} 0 \\ 1.60 \end{pmatrix}.$$

Contractions 2 and 3 take the form

$$\begin{pmatrix} x_{\text{new}} \\ x_{\text{old}} \end{pmatrix} = \begin{pmatrix} 0.20 & -0.26 \\ -0.23 & 0.22 \end{pmatrix} \cdot \begin{pmatrix} x_{\text{old}} \\ y_{\text{old}} \end{pmatrix} + \begin{pmatrix} 0 \\ 1.60 \end{pmatrix}$$

and

$$\begin{pmatrix} x_{\text{new}} \\ x_{\text{old}} \end{pmatrix} = \begin{pmatrix} -0.15 & 0.28 \\ 0.26 & 0.24 \end{pmatrix} \cdot \begin{pmatrix} x_{\text{old}} \\ y_{\text{old}} \end{pmatrix} + \begin{pmatrix} 0 \\ 0.44 \end{pmatrix}.$$

Finally, contraction 4 is given by

$$\begin{pmatrix} x_{\text{new}} \\ x_{\text{old}} \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0.16 \end{pmatrix} \cdot \begin{pmatrix} x_{\text{old}} \\ y_{\text{old}} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

The square involved is given by $-5 \leq x \leq 5$ and $0 \leq y \leq 10$.

7 Valentine's Grouse

As Thomasina struggles with her new geometry, there is a parallel mathematical development taking place in the play. Valentine is trying to use ideas from chaos theory to explain the rise and fall of the population of grouse on the Sidley Park Estate. He knows the data about grouse kills on the estate for the past two hundred years, and he would like to extrapolate from this to predict the populations in the future. Curiously, he is using the exact same technique that Thomasina had experimented with years before. Well, not quite. As Valentine explains, "Actually I'm doing it from the other end. She started with an equation and turned it into a graph. I've got a graph — real data — and I'm trying to find the equation which would give you the graph if you used it the way she used hers. Iterated it. It's how you look at population changes in biology. Goldfish in a pond, say. This year there are x goldfish. Next year there'll be y goldfish. Some get born, some get eaten by herons, whatever. Nature manipulates the x and turns it into y. Then y goldfish is your starting population for the following year. Just like Thomasina. Your value for y becomes your next value for x. The question is: what is being done to x? What is the manipulation? Whatever it is, it can be written down in mathematics. It's called an algorithm."

One of the simplest such algorithms used by population biologists is the logistic equation given by $F_k(x) = kx(1-x)$. Here x represents the percentage of some maximal population so that x lies between 0 and 1. The constant k is a parameter; we would use one value of k for grouse, another for rabbits, and a third for elephants. Given an initial population x_0 and a particular value of k, we can then iterate F_k to find the populations in successive years. For example, if $k = 1.5$ and $x_0 = 0.123$, then we find in succession

$$\begin{aligned}
x_0 &= 0.123 \\
x_1 &= 0.161\ldots \\
x_2 &= 0.203\ldots \\
x_3 &= 0.243\ldots\ldots \\
x_4 &= 0.275\ldots \\
&\vdots \\
x_{20} &= 0.3333\ldots \\
x_{21} &= 0.3333\ldots
\end{aligned}$$

so that the population has eventually stabilized at $0.3333\ldots$. If, on the other hand, we select $k = 3.2$ and $x_0 = 0.123$, then after several iterations we find that the population begins to cycle back and forth. One year the population is high; the next year it is low:

$$\begin{aligned} x_0 &= 0.123 \\ &\vdots \\ x_{20} &= 0.7994\ldots \\ x_{21} &= 0.5130\ldots \\ x_{22} &= 0.7994\ldots \\ x_{23} &= 0.5130\ldots. \end{aligned}$$

Like Thomasina, we can turn this data into a graph by plotting the *time series* corresponding to this iteration. This is a plot of the iteration count versus the actual numerical values. The cycling behavior above is illustrated in Figure 9.

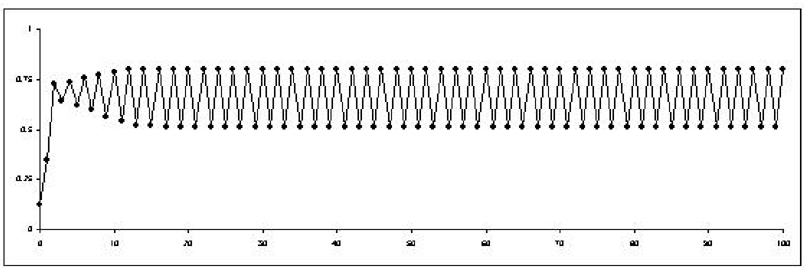


Figure 9: A time series indicating cyclic behavior.

When we choose $k = 4$ and $x_0 = 0.123$, the results of iteration are anything but predictable (Figure 10). The time series for this iteration shows no pattern whatsoever. More importantly, when we choose a nearby initial population, say 0.124, the output of the iteration is vastly different. A small change in the initial population has produced a major change in the eventual behavior. This is the phenomenon of chaos. Here we see that a very simple iterative scheme can yield results that are totally unpredictable.

8 The Orbit Diagram

The logistic equation would seem to be a rather simple mathematical object; after all, it is only a quadratic function. How hard could iteration of a

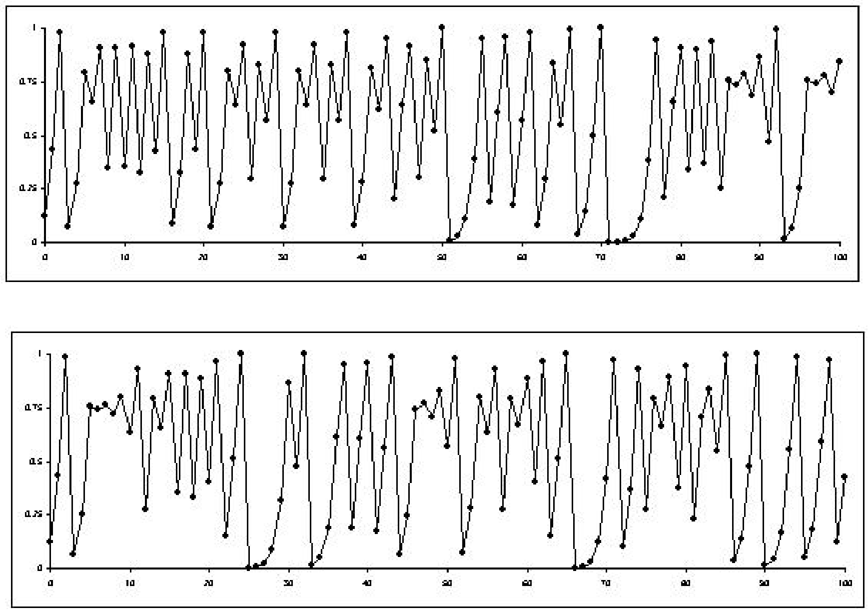


Figure 10: Several time series for $k = 4$ indicating chaotic behavior. The first is for the initial value $x_0 = 0.123$, the second for $x_0 = 0.124$.

quadratic function be? Well, it's pretty hard! Indeed, mathematicians finally understood the entire picture for the logistic equation in the late 1990's.

The fact is that, when the parameter k is varied, there are a tremendous number of different possibilities that may occur. The orbits of the function may tend to some attracting cycle (as in the case $k = 3.2$ displayed in Figure 9). Or the orbits may behave chaotically as in Figure 10. When the function is chaotic, lots of different behaviors are possible: there are infinitely many orbits that cycle and infinitely many other orbits that do not. In the latter case, some of these orbits may fill up a certain interval (or intervals) densely. And, as in the case $k = 4$, nearby orbits have vastly different fates.

To gain an appreciation of the complexity of the logistic function, we have plotted the *orbit diagram* of this function in Figure 11. In this figure, the parameter is plotted on the horizontal axis (here k runs from 2 to 4). Above each k-value, we plot the "eventual" behavior of the orbit of 0.5, the critical point for this function (i.e., the point where the derivative of the logistic function is zero). When we say eventual, we mean that we compute the first 400 points on the orbit of 0.5, but then only plot the last 300 points on this orbit.

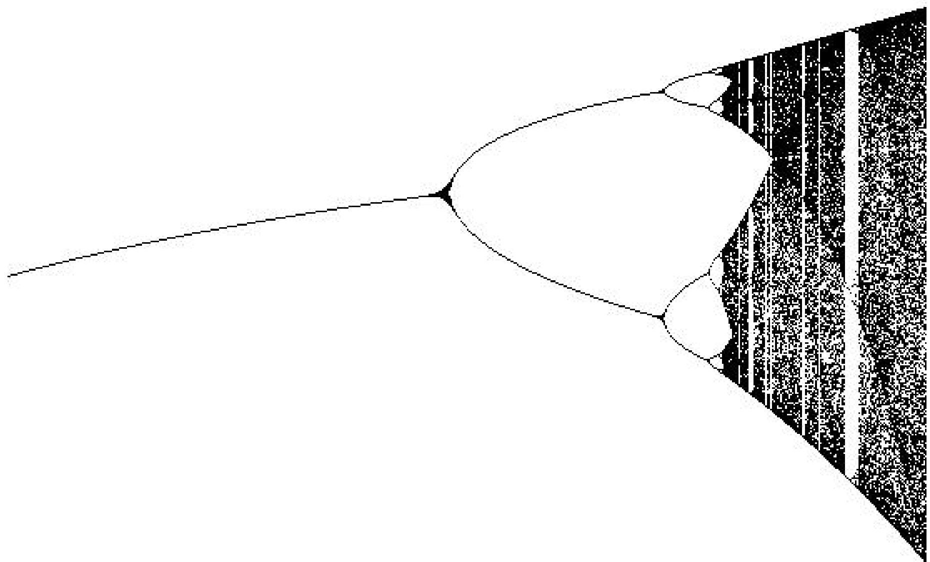


Figure 11: The orbit diagram for the logistic function.

As a remark, there is a reason why we use the critical point to plot this picture, for it is a fact that the critical point always finds any "attracting" cycle. Therefore there can be at most one attracting cycle for any chosen k-value; that is, for any specific logistic function, all its other cycles must be "repelling." In the orbit diagram, the "windows" that appear are regions in which the logistic function has an attracting cycle of some given period. There is usually much more going on in these windows. In any window except the largest left-hand window (where we see a fixed point, followed by a 2-cycle, then a 4-cycle, etc.), there is in fact an uncountable set of points on which the function behaves chaotically. For example, a famous theorem due to Sharkovsky states that, when a continuous function on the real line has a cycle of period three, then it must have a cycle of every other period as well! So, in the right of the orbit diagram, we see a period three window. For any parameter drawn from this region, the logistic map must therefore have cycles of all periods as well as a set on which there is chaotic behavior.

Curiously, the way mathematicians finally understood the logistic function was by moving to iteration in the complex plane. There we have many more tools available (the Schwarz Lemma, the Riemann Mapping Theorem, etc.). Using these tools together with computer graphics enabled mathematicians to describe precisely the order of events that occurred as the parameter k-varies. Indeed, it is the Mandelbrot set that provides this explanation (technically, the Mandelbrot set is a record of all that occurs for another quadratic function, namely $z^2 + c$, but this function and the logistic function behave essentially the same from a dynamical systems point of view). And, of course, the Mandelbrot set (see Figure 12) makes a brief appearance in the play. For Hannah glances over Valentine's shoulder and catches a glimpse of what is on the computer scren. "Oh, but... How beautiful!" she exclaims. Val responds, "The Coverly set. Lend me a finger. (He takes her finger and presses one of the computer keys several times.) See? In an ocean of ashes, islands of order. Patterns making themselves out of nothing. I can't show you how deep it goes. Each picture is a detail of the previous one, blown up. And so on. Forever. Pretty nice, eh?" Hannah: "Is it important?" Val responds, "Interesting. Publishable." "Well done!" says Hannah. "Not me. It's Thomasina's. I just pushed her equations through the computer a few million times further than she managed to do with her pencil."

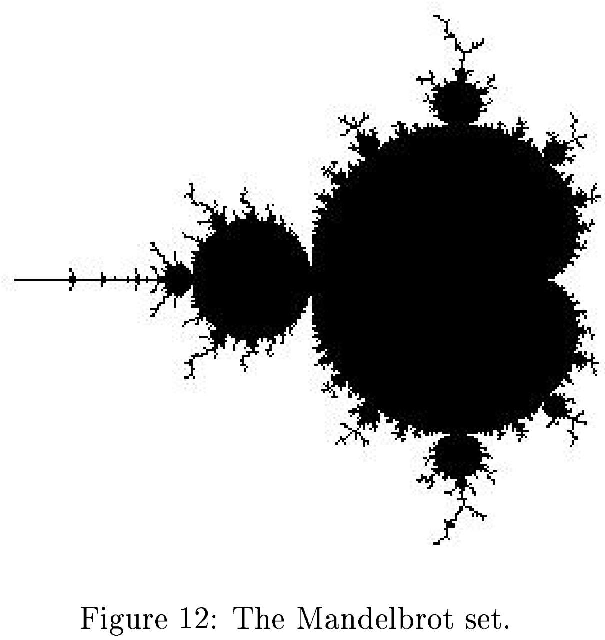


Figure 12: The Mandelbrot set.

9 Summary

All of the different mathematical ideas in the play are tremendously exciting, since they have, for the most part, arisen in the past quarter century. Moreover, the images with which they are associated are extremely beautiful and captivating. Valentine himself is ecstatic about this stuff. He summarizes what he is seeing in chaos and fractals: "The unpredictable and the predetermined unfold together to make everything the way it is. It's how nature creates itself, on every scale, the snowflake and the snowstorm. It makes me so happy. To be at the beginning again, knowing almost nothing... A door like this has cracked open five or six times since we got up on our hind legs. It's the best possible time to be alive, when almost everything you thought you knew is wrong."

The play *Arcadia* is a wonderful experience for all students. It is fast-paced, witty, and thoroughly enjoyable. Best of all, it can be combined with some wonderful mathematical ideas to give students a truly interdisciplinary experience. I have worked over the years with several high school, college, and professional productions of *Arcadia*. Often, in the high schools, the mathematics, science, and humanities teachers team up to give introductory classes on the topics of the play to many or all of the students in the school. These various interdisciplinary experiences for the students are then brought to a wonderful conclusion with the staging of the play.

For more information on the history of the subject of chaos, consult [6]. Some of the mathematical and biological underpinnings of the subject may be found in [7]. Fractals are described in [1], [3], and [8]. A primer on chaos can be found in [2] for high school students and in [4] for undergraduates. An interactive version of this paper plus some java applets to play the chaos game may be found at the Dynamical Systems and Technology Project website at `http://math.bu.edu/DYSYS`.

References

[1] Barnsley, M. *Fractals Everywhere* Academic Press, Boston, 1988.

[2] Choate, J. and Devaney, R. L. *Chaos*. Key Curriculum Press, 1999.

[3] Choate, J., Devaney, R. L., and Foster, A. *Fractals*. Key Curriculum Press, 1999.

[4] Devaney, R. L. *A First Course in Chaotic Dynamical Systems.* Westview Press, 1992.

[5] Devaney, R. L. *Chaos Rules!* Math Horizons, November (2004), 11-14.

[6] Gleick, J. *Chaos: Making a New Science*, Viking, New York, 1987.

[7] May, R. B. Simple mathematical models with complicated dynamics. *Nature.* **261**, 459-467, (1976).

[8] Mandelbrot, B. *The Fractal Geometry of Nature.* San Francisco: Freeman, 1983.

[9] Stoppard, T. *Arcadia.* London: Faber and Faber, 1993.