The Chaos game rules in 'plain English'

We want to translate the mathematical chaos game, namely the rule $z_{k+1} = w_{s_i}(z_k)$ for going from one game point to the next, into 'plain English'. To do this we first find a convenient expression for z_{k+1}. Let $w_i = A_i + v_i$. If q_i is the fixed point of w_i, then $q_i = w_i(q_i) = A_i q_i + v_i \ \rightarrow v_i = q_i - A_i q_i$. So we can write $z_{k+1} = w_i(z_k) = A_i z_k + v_i = A_i z_k + q_i - A_i q_i$. Thus, $\boxed{z_{k+1} = q_i + A_i(z_k - q_i)}$, or $\boxed{z_{k+1} = z_k + (\mathbb{I} - A_i)(q_i - z_k) = z_k + (A_i - \mathbb{I})(z_k - q_i)}$. Note that $z_k - q_i$ is the vector pointing in the direction *from* q_i *towards* z_k, and if v is any vector, $q_i + v$ is that vector translated so that its tail is at the point q_i; see Figure 1 below. Thus, the formula $z_{k+1} = q_i + A_i(z_k - q_i)$ says,"Stand on the fixed point q_i and apply the transformation A_i to the vector pointing from you to the current game point z_k. The end point of this new vector is the next game point z_{k+1}". Here, A_i is applied to vectors whose tails are at q_i (if $B_i(v) = A_i(v - q_i)$, then $B_i(q_i) = 0$). Alternatively, $z_{k+1} = z_k + (\mathbb{I} - A_i)(q_i - z_k)$ says, " Stand on the current game point z_k and apply $(\mathbb{I} - A_i)$ to the vector $q_i - z_k$ that points from you to the fixed point q_i. The end point of this new vector is the next game point z_{k+1}".

For example, if $A_i = \begin{bmatrix} r & 0 \\ 0 & r \end{bmatrix} = D_r$ is dilation by r, then $z_{k+1} = z_k + (A_i - \mathbb{I})(z_k - q_i)$ says, "Standing on the game point z_k, to obtain the next game point move towards the fixed point q_i a distance $r - 1$ times the distance from the current game point to q_i". If A_i is rotation counterclockwise by θ; $R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$, then the rule (if an i appears in the game sequence) $z_{k+1} = q_i + A_i(z_k - q_i)$ is, " Rotate the line joining the fixed point q_i to the current game point clockwise by θ about the fixed point to obtain the next game point". And if A_i is dilation by r followed by a rotation by θ, $A_i = R_\theta \circ D_r$, then the rule is, "Move towards the fixed point q_i a distance $1 - r$ times the distance from the current game point to q_i and then rotate by θ counterclockwise about the fixed point". In general, the game rule is the action of the transformation A_i *with respect to the fixed point* q_i. Unless the transformation A_i is a 'simple' one, the game rule may be rather complicated to describe.

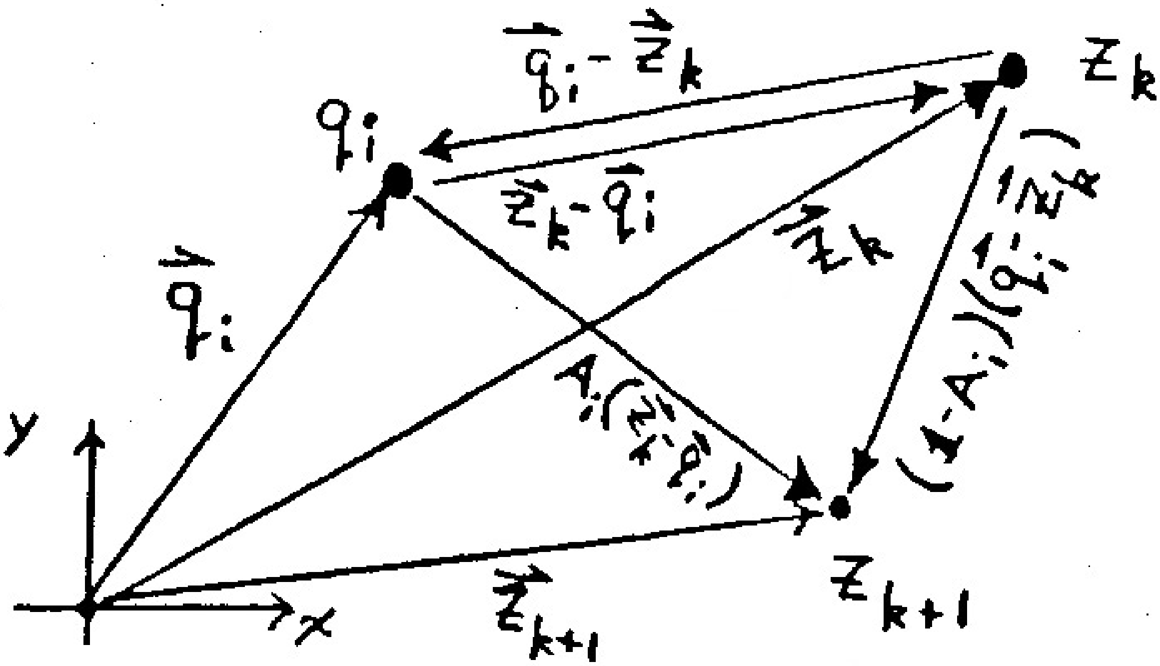

Figure 1

Some chaos games

1. Sierpinski (Triangle)
   - three black pins 1, 2, 3, arranged at vertices of equilateral triangle
   - choose random number s_i from $\{1, 2, 3\}$
   - actions; move 1/2 distance from current game point to black pin labelled s_i
2. Square
   - four black pins at the corners of a square

- choose random number s_i from $\{1, 2, 3, 4\}$
- actions; move 2/3 distance to pin labelled s_i

3. Pentagon

    - five black pins at the corners of a pentagon
    - choose random number s_i from $\{1, 2, 3, 4, 5\}$
    - actions; move 1/2 distance to pin labelled s_i

4. Full Square

    - four black pins at the corners of a square
    - choose random number s_i from $\{1, 2, 3, 4\}$
    - actions; move 1/2 distance to pin labelled s_i

5. Sierpinski variation #1

    - three black pins 1, 2, 3, arranged as in Sierpinski, but <u>place pin 3 directly above pin 2</u>
    - choose random number s_i from $\{1, 2, 3\}$
    - actions; move 1/2 distance to black pin labelled s_i

6. Sierpinski variation #2

    - three black pins 1, 2, 3, arranged as in variation #1
    - choose random number s_i from $\{1, 2, 3\}$
    - actions;
        - $s_i = 1$; move 1/2 distance to black pin labelled 1
        - $s_i = 2$; move 1/2 distance to black pin labelled 2
        - $s_i = 3$; move 1/2 distance to pin 3 <u>and then rotate counterclockwise about pin 3 by 90 degrees</u>

7. Spiral

- two black pins, aligned vertically
- choose random number s_i from $\{1, 2\}$
- actions;
  - $s_i = 1$; move $8/10$ distance to pin 1 and then rotate counterclockwise about pin 1 by 20 degrees
  - $s_i = 2$; move $1/2$ distance to pin 2 and then rotate counterclockwise about pin 2 by 30 degrees

8. Christmas Tree

- three black pins, 1 and 2 along a horizontal line, 3 directly above midpoint between 1 and 2
- choose random number s_i from $\{1, 2, 3\}$
- actions;
  - $s_i = 1$; move $1/2$ distance to pin 1 and then rotate counterclockwise about pin 1 by 90 degrees
  - $s_i = 2$; move $1/2$ distance to pin 1 and then rotate clockwise about pin 2 by 90 degrees
  - $s_i = 3$; move $1/2$ distance to pin 3

Computing the number of points in address regions

Suppose the chaos game for the IFS $W = w_i \cup w_2 \cup \cdots \cup w_k$ is played with probabilities $p_1, p_2, \ldots, p_k$, and $\{s_1, s_2, \ldots s_N\}$ is a game sequence for these probabilities. Let's imagine that this finite game sequence is the beginning part of the infinite random sequence $\{s_1, s_2, \ldots\}$. If $s_j s_{j-1} \ldots s_{j-r}$ is a sequence of the integers $\{1, 2, \ldots, k\}$, then this sequence will appear with frequency $p_{s_j} p_{s_{j-1}} \cdots p_{s_{j-r}}$ in $\{s_1, s_2, \ldots\}$. So we expect approximately $(p_{s_j} p_{s_{j-1}} \cdots p_{s_{j-r}})N$ occurrences of this sequence in $\{s_1, s_2, \ldots, s_N\}$. Each time this sequence occurs in the game sequence, a game point lands in the region with address $s_{j-r} \ldots s_{j-1} s_j$. Therefore, we expect approximately $(p_{s_j} p_{s_{j-1}} \cdots p_{s_{j-r}}) \cdot N$ game points in this region.

This leads us to estimate how many game points would be needed to draw a fractal using the chaos game. Let's look again at the Sierpinski triangle. We know that to draw the fractal completely on our computer screen, we need to generate all detail down to the size of a pixel, and the length of the addresses of these address regions is 9. Thus, we would need at least one game point in every address regions whose address length is 9, which means we need a game sequence $\{s_1, s_2, \ldots, s_N\}$ that contains every subsequence of $1'^s, 2'^s$ and $3'^s$ of length 9. If the probabilities are all equal, $p_i = \frac{1}{3}$, then the frequency of appearance of any one of these sequences in a random sequence $\{s_1, s_2, \ldots\}$ generated with these probabilities is $(\frac{1}{3})^9 \approx \frac{1}{20{,}000}$, which means we would need a game sequence of length about $20,000$. You can check these numbers out for yourself. The sizes (in pixels) of the canvases that contain the image produced by the chaos game applet are;

version of applet	**canvas size (pixels)**
640 × 480 (VGA)	304 × 304
800 × 600 (SVGA)	380 × 380
1024 × 768 (XVGA)	500 × 500
1280 × 1024 (UVGA)	645 × 645
1400 × 1050	800 × 800

Varying the probabilities

Suppose we play the chaos game with the Sierpinski IFS, but instead of using equal probabilities $p_1 = p_2 = p_3 = \frac{1}{3}$, we use the probabilities $p_1 = p_2 = \frac{1}{5},\ p_3 = \frac{3}{5}$ to generate the game sequence. Then the digit 3 will occur in the game sequence 3 times as often as a 1 or a 2. Consequently, game points will land in the region with address 3 (the top part of the triangle) 3 times more often as they will land in the regions with address 2 or 3. So playing the chaos game with these probabilities will result in the top part of the Sierpinski triangle filling out more quickly than the bottom parts (try the applets). We know that since the probabilities are all non-zero, this chaos game *will* draw the Sierpinski triangle, but it will take longer (*a lot* longer) to draw it with these probabilities than with equal probabilities. For example, the string 121212121 will occur in the random sequence with frequency $(\frac{1}{5})^9 \approx (2 \times 10^6)^{-1}$ so you will need a game sequence with length approximately 2×10^6 to draw the fractal to level 9 (compare this to the $20,000$ game points we estimated when we use equal probabilities).

We can go on. The string 13 will occur 3 times as often in the game sequence as the string 12, so 3 times as many game points will land in the region will address 31 than in the region with address

21. The string 133 will occur 9 times as often as the string 122, so 9 times as many games points will land in the region with address 331 than in the region with address 221, etc. This explains why in the chaos game that is played with these probabilities, you see a greater density of game points in the upper parts of all the sub-triangles than in the lower parts.

An even more interesting pattern emerges when we play the chaos game for the full square IFS with non-equal probabilities. We know that if we play the chaos game with any non-zero probabilities the resulting image will (eventually) be a solid square. But you see vastly different patterns emerging (different ways of convergence to the solid square) with non-equal probabilities. For example, choosing the probabilities $p_1 = p_2 = \frac{2}{5},\ p_3 = p_4 = \frac{1}{10}$ will result in the appearance of a series of vertical lines; Figure 2. We can understand what causes this pattern when we realize that most of the time the game sequence is just $1^{'s}$ and $2^{'s}$ (in fact 80% of the time) so points will be attracted towards the fixed point of the IFS made with just lenses 1 and 2 which is the vertical line along the left edge of the square joining the corners that are the fixed points of w_1 and w_2. But whenever a 3 or a 4 appears in the game sequence, the game point (which is most likely lying near that vertical line) will jump towards one of the corners along the right edge of the square (the fixed points of w_3 and w_4). They will move $\frac{1}{2}$ the distance towards them, in fact. Thus, they will lie along the vertical line at $x = \frac{1}{2}$, which is clearly visible in the figure. The vertical line at $x = \frac{3}{4}$ is caused by game points near the line $x = \frac{1}{2}$ being moved under w_3 or w_4 (which again moves them $\frac{1}{2}$ the distance to one of the corners on the right side). This is how the vertical lines at $x = \frac{1}{2}, x = \frac{3}{4}, x = \frac{7}{8}, x = \frac{15}{16}, \ldots$ appear.

We can also estimate the relative density of points on the various vertical lines (and hence explain why they don't all appear equally dark). To do this it simplifies the problem if we consider the game beginning with not a single game point but a uniform distribution of points along the vertical line at the left edge of the square ($x = 0$) with density ρ_0 (this is reasonable because most of the time the game sequence will be $1^{'s}$ and $2^{'s}$ so the game points will just be moving very near this line back and forth between the two corners). Points on the vertical line at $x = \frac{1}{2}$ come from the line $x = 0$ under one of the transformations w_3 or w_4. A 3 or a 4 appear in the game sequence 20% of the time, so the density $\rho_{\frac{1}{2}}$ of points on the line $x = \frac{1}{2}$ is $\rho_{\frac{1}{2}} = 0.2\rho_0$. Similarly, the points on the line at $x = \frac{1}{4}$ come from points originating on the line $x = \frac{1}{2}$ and then moving towards either of the fixed points q_1 or q_2 by the transformations w_1 or w_2. Thus, these points have to first move to the line $x = \frac{1}{2}$ from the line $x = 0$, and then back to the line $x = \frac{1}{4}$. We know that there's a 20% probability of points moving from the line $x = 0$ to $x = \frac{1}{2}$. Then there's an 80% probability that once on $x = \frac{1}{2}$ they will move back to $x = \frac{1}{4}$, so the total probability for a point to move from $x = 0$ to $x = \frac{1}{2}$ and then to $x = \frac{1}{4}$ is $0.2 \times 0.8 = 0.16$. Thus, we expect the line at $x = \frac{1}{4}$ to be 16% as dark as the line $x = 0$, and 80% as dark as the line $x = \frac{1}{2}$; $\rho_{\frac{1}{4}} = 0.16\rho_0 = 0.8\rho_{\frac{1}{2}}$.

See Figure 3 for a variation of this problem (here the adjacent diagonal fixed points have the different probabilities).

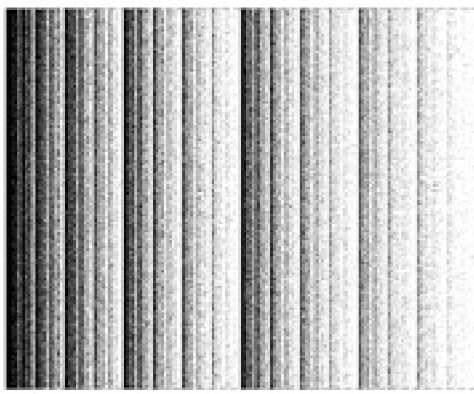

Figure 2

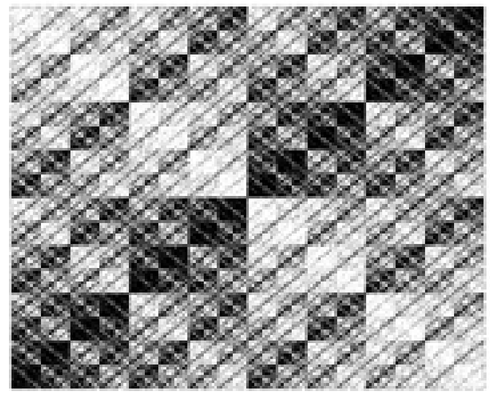

Figure 3

Finding the best probabilities

Let's consider the Fern IFS $W = w_1 \cup w_2 \cup w_3 \cup w_4$. If we play the chaos game using equal probabilities $p_i = \frac{1}{4}$, we see that it is not possible to obtain a good image of the fractal, no matter how long we play the game. We notice that the ends of the fern and all the sub-ferns do not have many points in them.

The tops or ends of the sub-ferns have addresses that begin with a 3 and end with a tail of 1's (see Figure 30). The address of the n^{th} branch up along the main stem is 11....13 where there are $(n-1)$ $1^{'s}$. Say you want to see game points in the 15^{th} branch of the fern. The probability of seeing a point here is $p_1^{14} p_3 = (\frac{1}{4})^{15} \approx 10^{-9}$, so you need 10^9 steps in the game to get a point up here! I think it would take a *very* long time for your computer to plot a billion game points. So if we only have 100,000 points, we wouldn't expect any points to lie above the eighth branch. This is why the fern looks incomplete when we play the chaos game with equal probabilities (Figure 31)

We can obtain a much better result (i.e., a much better image of the fern) by adjusting the probabilities. Roughly speaking, the problem with uniform probabilities is that too many of the game points end up on the lower branches of the fern. After a while those lower branches get saturated with points while the upper parts of the branches are lacking points. What we need to do is to move some of the points that go to the lower branches up to higher branches. This can be accomplished if we assign a greater probability to those transformations that move game points to the higher branches. To this end we change the probabilities so that p_1 is much greater than the other probabilities (recall that w_1 is responsible for building the higher branches of the fern). In fact, if $p_1 = .85$ and $p_2 = p_3 = p_4 = .05$, then we will expect about 500 game points to fall in the 15th branch of the fern. Thus, with these (nonuniform) probabilities we obtain a good image of the fern using a modest number of game points (see Figure 31.

How can we find the *optimal* probabilities? That is, a set of probabilities that allow us to draw the fractal with as few game points as possible. One can find a *good* set of probabilities by arguing as follows. What we want in a well-drawn fractal is an *equal density* of points in each address region, not (necessarily) an equal number of *points* in each address region. Let a_o be the area of the outline $\mathcal{F}_o$ of the fern, $c_i = |\det A_i|$ (so $c_2 = 0.096, c_3 = 0.078, c_4 = 0.578$), and F_i denote the address region with address i; $F_i = w_i(\mathcal{F}_o)$. Then the density of game points ρ_i in region F_i is (approximately) $\rho_i = \frac{\text{\# of game points}}{\text{area of } F_i} = \frac{p_i \cdot N}{c_1 \cdot a_o}$, where N is the total number of game points (length of the game sequence). As a first approximation to a better set of probabilities, we ask that the density of game points in each address region with address length 1 be equal; $\rho_1 = \rho_2 = \rho_3 = \rho_4$. Since the factor $\frac{N}{a_o}$ occurs in each ρ_i, these equalities become $\frac{p_i}{c_1} = \frac{p_2}{c_2} = \frac{p_3}{c_3} = \frac{p_4}{c_4}$. In addition to these equalities, we require $p_1 + p_2 + p_3 + p_4 = 1$. For the fern, $c_1 = 0$, so we set (arbitrarily) $p_1 = 0.02$ (so that we get *some* game points in those regions; we can change the value of p_1 later and see if we obtain a better set of probabilities). Now we solve $\frac{p_2}{c_2} = \frac{p_3}{c_3} = \frac{p_4}{c_4}$, $p_2 + p_3 + p_4 = 0.98$ (three equations in three unknowns). Taking the first two we obtain $p_2 = \frac{c_2 p_3}{c_3}$, so that $p_4 = 0.98 - p_2 - p_3 = 0.98 - (\frac{c_2}{c_3} + 1)p_3 = 0.98 - 2.23p_3$. From $\frac{p_3}{c_3} = \frac{p_4}{c_4}$ we obtain $p_3 = \frac{c_3 p_4}{c_4}$, and substituting this into $p_4 = 0.98 - 2.2p_3$ we find $p_4 = 0.75$. And so $p_2 = 0.12$ and $p_3 = 0.11$ (rounded off). One can check that playing the chaos game with these probabilities for the fern results in a good image of the fern fairly quickly; $\approx 50,000$ game points.

'Pseudo Fractals'

Let's take a random sequence of $1^{'s}, 2^{'s}$ and $3^{'s}$ with probabilities $p_1 = p_2 = p_3 = \frac{1}{3}$. Call this sequence s. We know this game sequence will draw the Sierpinski triangle in the chaos game very quickly. Now let's remove all the $1^{'s}$ that appear in this sequence, and call this new game sequence s' What image will result if we use this new game sequence in the chaos game? Well, since no 1 appears in this sequence, no 1 will appear in the address of any game point, so no game point will lie in region D_1, nor will there be any game points in region D_{21}, nor in D_{331}, etc. So there will be many more 'holes' in the fractal compared to the Sierpinski triangle, and all the holes correspond to the small triangles in the lower right hand corners of every triangle in the Sierpinski triangle. If you think about it, this will result in just a line joining the lower right and top verticies of the large triangle. We can also predict the outcome by realizing that the new game sequence s' is simply a random sequence of $1^{'s}$ and $2^{'s}$ with equal probabilities. If you use the Sierpinski IFS with this sequence, the transformation w_1 is never used, so in effect you are using the IFS $W = w_2 \cap w_3$, whose fixed point is the line joinging the fixed points of w_2 and w_3, which are the two vertices of the triangle.

Consider the Full Triangle IFS - this is the IFS which has the three transformations as the Sierpinski triangle, but with a fourth 'filling' the missing centre triangle of the Sierpinski IFS. Thus, the fixed point of this IFS is simply a solid triangle. Now take a random sequence s of $1^{'s}, 2^{'s}, 3^{'s}$ and $4^{'s}$, and this time remove all occurrences of the string 12. Call this new game sequence s''. Notice that s'' is *not* a random sequence. That is, it is not generated by randomly picking the digits 1,2,3 and 4 according to some probabilities p_1, p_2, p_3 and p_4 (because, for example, the string 12 never appears!). If we play the chaos game for the Full Triangle IFS with this sequence, then a game point will never land in an address region with an address containing the string 21. So for example, the sub-triangle D_{21} will have no points in it. Also the sub-triangle D_{121}, etc. Since all the transformations w_i are used in this chaos game, the final image is not a trivial line. In fact it looks like a fractal, but technically it is not because it is not self-similar; see Figure 33. In general, if we remove strings of length > 1 from any random sequence and use this to play the chaos game, a fractal-like image will result. 'Pseudo fractals' such as this one can be generated with the *Modified Chaos Game Applet*.