Fractal Curves And Dynamical Systems

in 3D-XplorMath, a Visualization Program

Fractal Curves

1.) Dragon Curve
2.) Koch Snowflake
3.) Hilbert Square Filling Curve
4.) Sierpinski Curve
5.) Hilbert Cube Filling Curve

Dynamical Systems

6.) Henon Map
7.) Feigenbaum Tree
8.) User Defined Feigenbaum Iteration
9.) User Defined x y Iteration
10.) Mandelbrot Set And Its Julia Sets

About The Dragon Curve*

see also: Koch Snowflake, Hilbert SquareFillCurve
To speed up demos, press DELETE

The Dragon is constructed as a limit of polygonal approximations D_n. These are emphasized in the 3DXM default demo and can be described as follows:
1) D_1 is just a horizontal line segment.
2) D_{n+1} is obtained from D_n as follows:

a) Translate D_n, moving its end point to the origin.

b) Multiply the translated copy by $\sqrt{1/2}$.

c) Rotate the result of b) by $-45°$ degrees and call the result C_n.

d) Rotate C_n by $-90°$ degrees and join this rotated copy to the end of C_n to get D_{n+1}.

The fact that the **limit points** of a sequence of longer and longer polygons can form a two-dimensional set is not surprising. What makes the Dragon spectacular is that it is a **continuous curve** whose image has positive area—properties that it shares with Hilbert's square filling curve.

There is a second construction of the Dragon that makes it easier to view the limit as a continuous curve and is also similar to the constructions of the following curves. Select in the Action Menu: `Show With Previous Iteration`.

* This file is from the 3D-XplorMath project. Please see:

http://3D-XplorMath.org/

This demo shows a local construction of the Dragon: We obtain the next iteration D_{n+1} if we modify each segment of D_n by replacing it by an isocele $90°$ triangle, alternatingly one to the left of the segment, and the next to the right of the next segment. This description has two advantages:
(i) Every vertex of D_n is already a point on the limit curve. Therefore one gets a dense set of points, $c(j/2^n)$, on the limit curve c.
(ii) One can modify the construction by decreasing the height of the modifying triangles from $aa = 0.5$ to $aa = 0$. The polygonal curves are, for $aa < 0.5$, polygons without self-intersections. This makes it easier to imagine the limit as a curve. In fact, the *Default Morph* shows a deformation from a segment through continuous curves to the Dragon—more precisely, it shows the results of the $(ee = 11)$th iterations towards those continuous limit curves.

The Dragon is a fractal tile for the plane., see several versions at `http://en.wikipedia.org/wiki/Dragon_curve`. For two beautiful possibilities select from the Action Menu `Tile Plane With Dragon Pairs`, or: `Tile Plane With Dragon Quartetts`.

Finally, one can choose in the Action Menu to map any selected Fractal curve by either the complex map $z \to z^2$ or by the complex exponential. The program waits for a mouse click and then chooses the mouse point as origin.

R.S.P., H.K.

About the Koch Snowflake (or Island)*

The Koch Snowflake Curve (aka the Koch Island) is a fractal planar curve of infinite length and dimension approximately 1.262. It is defined as the limit of a sequence of polygonal curves defined recursively as follows:

1) The first polygon is an equilateral triangle.
2) The $(n+1)$-st polygon is created from the n-th polygon by applying the following rule to each edge: construct an equilateral triangle with base the middle third of the edge and pointing towards the outside of the polygon, then remove the base of this new triangle.

Note that at each step the number of segments increases by a factor 4 with the new segments being one third the length of the old ones. Since all end points of segments are already points on the limit curve we see that no part of the limit curve has finite length.
Actually this is true for a 1-parameter family of similar constructions: Vary the parameter aa (`Set Parameters` in the Settings Menu) in the interval $[0.25,\ 0.5]$ and watch how the iterations evolve or choose `Morph` in the Animation Menu and observe the deformation of the limit curves.

Hausdorff Dimension: Consider the union of those disks which have a segment of one polygonal approximation as a diameter, then this union covers all the further approximations. From one step to the next the diameter of the disks shrinks to one third while the number of disks is mul-

* This file is from the 3D-XplorMath project. Please see: http://3D-XplorMath.org/

tiplied by 4—so that the area of these covering disk unions converges to zero. The fractal Hausdorff d-measure is defined as the infimum (as the diameter goes to zero) of the quantity (diameter)$^d\times$ (number-of-disks), and the fractal Hausdorff dimension is the infimum of those d for which the d-measure is 0. This shows that the Hausdorff dimension of the Koch curve is less than or equal to $\log(4)/\log(3)$, and since the union of the disks of every second segment does not cover the limit curve one can conclude that the Hausdorff dimension is precisely $\log(4)/\log(3)$.

The artist Escher has made rather complicated fundamental domains for tilings of the plane by modifying the boundary between neighboring tiles. This idea can be used to illustrate the flexibility of fractal constructions: Select from the Action Menu of the Koch Snowflake `Choose Escher Version` and observe:
The new polygonal curves remain boundaries of tiles of the plane under the iteration steps that make them more and more complicated.

Finally, one can choose in the Action Menu to map any selected Fractal curve by either the complex map $z \to z^2$ or by the complex exponential. The program waits for a mouse click and then chooses the mouse point as origin.
Note that one gets the graph of a continuous function if one plots the x-coordinate of a continuous curve against the curve parameter. This can be viewed with the last Action Menu entry.
H.K.

About Hilbert's Square Filling Curve*

See also: Koch Snowflake, Dragon Curve
Speed up demos by pressing DELETE

In 1890—the year the German Mathematical Society was founded, David Hilbert published a construction of a continuous curve whose image completely fills a square. At the time, this was a contribution to the understanding of continuity, a notion that had become important for Analysis in the second half of the 19th century. Today, Hilbert's curve has become well-known for a very different reason—every computer science student learns about it because the algorithm has proved useful in image compression. In this application one has to enumerate a first square, its four half size subsquares, their sixteen quarter-size subsquares and so on, in such a way that squares whose numbers are close are also close to each other geometrically. In other words, the continuity of this space filling curve is now important, in contrast to the fact that the curve was considered a pathological example of continuity for many years after Hilbert's discovery.

It was known in 1890 that such a curve, i.e., a continuous map c of $[0,1]$ onto $[0,1] \times [0,1]$, could not be one-to-one, i.e., *Certain pairs of points* t_1, t_2 *of the interval* $[0,1]$ *must*

* This file is from the 3D-XplorMath project. Please see: http://3D-XplorMath.org/

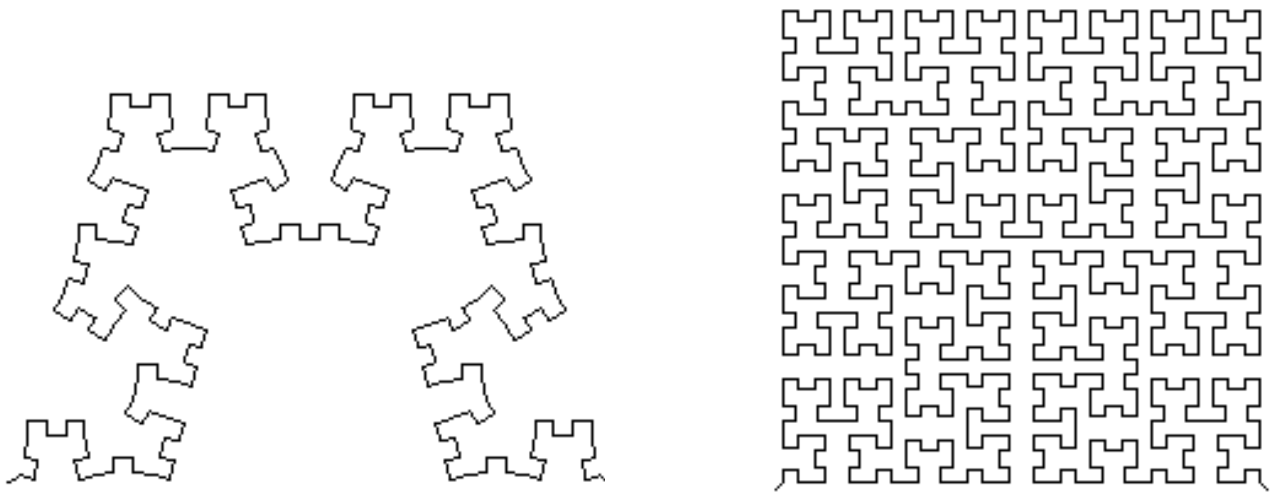
have the same image $c(t_1) = c(t_2) \in [0,1] \times [0,1]$. This led Hilbert to give a special twist to his construction: He gave a sequence of polygon approximations of the strange limit curve that, surprisingly, were all one-to-one! In retrospect it seems almost as if Hilbert foresaw what would be needed a century later in image compression; when people say that they are using Hilbert's square-filling curve, they mean more precisely that they are using Hilbert's approximations to that curve!

The basis of Hilbert's construction is a single step that is repeated over and over again. We first explain a simplified version, although this does not exactly give Hilbert's one-to-one approximations that made the construction so famous. Assume that we already have a curve inside the square and joining the left bottom corner to the right bottom corner. 3DXM offers four different initial such curves, leading to quite different pictures. The basic construction step is to scale the square and its curve by $\frac{1}{2}$ and put **four** copies of this smaller square side by side in the original square, in such a way that these four smaller copies of the curve fit together to form a new curve from the left bottom corner to the right bottom corner of the original square. But instead of reading more words, we suggest that you view the default approximations of the Hilbert curve in 3DXM. We use a rainbow coloration to emphasize the continuous parametrization, and we repeat the colors **four** times to emphasize that four copies of the previous approximation make up the new one.

The two end points of the curve (to which this basic iteration step is applied) play a special role, on the one hand they lead at each iteration step to more points that are already points on the limit curve, on the other hand exactly these easy points lead to double points on the approximations! Hilbert therefore removed small portions of the curve near its two end points before he applied the above iteration step. One can see how these Hilbert approximations manage to stay one-to-one and how they wander through all the little squares of the current subdivision of the original square—and these are just the properties used in image compression.
In 3DXM one can choose with the parameter cc between several initial curves. An even number and the following odd number choose the same curve, but for even cc the Hilbert iteration is done *without* the endpoints and for odd cc *including* the endpoints. In the Action Menu one can switch between Hilbert's approximation (cc=0) and one that emphasizes the iteration of the endpoints (cc=5).

Finally we add to the above descriptive part some more technical explanations, namely how to understand the limit as a continuous curve. Select the Action Menu entry "Emphasize Limit Points". The first shown step (for our default value cc $=$ 5) is a curve that is mostly a straight segment, but has also two little wiggles, that emphasize the initial point $c(0)$ and the end point $c(1)$. The second step is a curve with four straight segments that join five wiggles, the points $c(j/4),\ j := 0,\ldots,4$. These points are really points on the limit curve because they remain fixed under all further applications of Hilbert's basic construction step. In the third step we get 17 wiggles, the points $c(j/16),\ j := 0, \ldots, 16$ of the limit curve, and so on. The 3DXM demo shows six such iterations. One can deduce from this the continuity of the limit curve if one proves for these approximations:

$$|t_2 - t_1| \le \frac{1}{4^n} \Rightarrow |c(t_2) - c(t_1)| \le \frac{1}{2^n}.$$



Early iterations for $bb = 0.4$ (left), for $bb = 0.5$ (right, the Hilbert case).

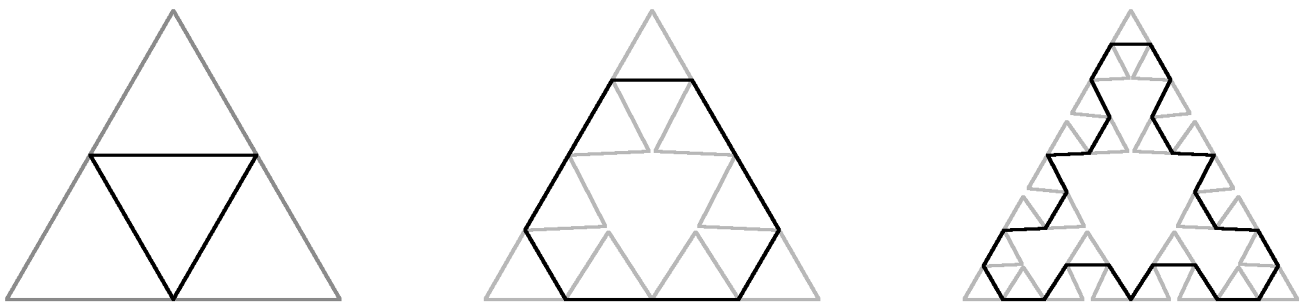
H.K.

The Sierpinski Triangle, The Sierpinski Curve*

The Sierpinski Triangle is a well known example of a "large" compact *set* without interior points. It is defined by the following construction:
Start with an equilateral triangle and subdivide it into four congruent equilateral triangles. Remove the middle one. Subdivide the remaining triangles again and remove in each the middle one. Repeat this procedure. Each step reduces the area by a factor 3/4. – But more is true:

Sierpinski's Triangle is the image of a continuous curve.

As in the other fractal curves in 3DXM we have to define an iteratively defined and uniformly convergent sequence of polygonal curves. As in the case of the Hilbert square filling curve there is an easier construction by *non-injective* curves which, however, can be modified to give better looking *injective* approximations. In the following illustration we have chosen the 3DXM parameter $bb = 0.49$, because for $bb < 0.5$ the easier construction also gives injective approximations. ($bb = 0.5$ gives Sierpinski's curve.)



* This file is from the 3D-XplorMath project. Please see: http://3D-XplorMath.org/

The starting polygonal curve has the vertices and the edge midpoints of an equilateral triangle as its vertices. The initial point is the midpoint of the bottom edge. The curve that joins every second vertex of the starting curve is the triangle in the middle. We view the starting curve as passing through two edges of each of the three outer triangles. We only have to describe for one of these triangles how the next iteration is obtained. We will obtain curves that always run through two edges of each triangle, and the basic iteration can always be applied. If we join every second vertex of the resulting curves then we obtain the injective approximations of the Sierpinski Curve.
The basic iteration step, for one triangle:
First add the two midpoints of the traversed edges of the triangle. Two more points are added, one over the first and one over the last of the four subsegments. The points lie in the inside of the traversed triangle and they are the tips of isocele triangles whose base is the first, resp. the last, of the four subsegments. In the case of the Sierpinski Curve these isocele triangles are in fact equilateral. If the parameter bb is smaller than 0.5 then the height of the isocele triangle is by the factor $bb/0.5$ smaller than the height of the equilateral triangle – thus avoiding the creation of double points of the approximation.
The iterated polygonal curve joins the initial point of the first edge to the first tip, continues to the first edge-midpoint, passes through the vertex of the original triangle to the second edge-midpoint, continues through the second tip

and ends at the final point of the last segment. The iterated polygonal curve traverses three triangles, two edges in each. Therefore the iteration step can be repeated.
The default `Morph` from the Animation Menu of 3DXM varies *bb* from 1/3 to 1/2 thus joining the first triangle contour by a family of continuous (and injective) curves to the Sierpinski Curve.
Finally, one can choose in the Action Menu to map any selected Fractal curve by either the complex map $z \to z^2$ or by the complex exponential. The program waits for a mouse click and then chooses the mouse point as origin.

H.K.

About Hilbert's Cube Filling Curve*

See also: The more famous Hilbert SquareFillCurve.

Hilbert's cube filling curve is a continuous curve whose image fills a cube. It is a straight forward generalization of the continuous square filling curve. It is shown in anaglyph stereo via a sequence of polygonal approximations. Each approximation is a polygon that joins two neighboring vertices of the cube.
The iteration step goes as follows:
The cube with the given (initial or a later) approximation is scaled with the factor 1/2. Eight of these smaller copies are put together so that they again make up the original cube, and this is done in such a way that the endpoint of the curve in the first cube and the initial point of the curve in the second cube fit together, and so on with all eight cubes. The result of one iteration therefore is a curve that is four times as long as the previous curve and that runs more densely through the cube. In 3DXM, if one rotates the cube with the mouse then the cube and its first subdividing eight cubes are shown together with one iteration of the initial curve.

To achieve a better feeling for the iteration step, one can set the parameter cc to integer values between 0 and 5.

---

* This file is from the 3D-XplorMath project. Please see: http://3D-XplorMath.org/

This will select different initial curves. An even value of cc and the following odd value give the same initial curve, but for even cc the Hilbert iteration is done *without* the endpoints, while for odd cc the endpoints are *included* in the iteration. (Using the Action Menu, one can switch between Hilbert's default (cc=0) and a case that emphasizes the iteration of the endpoints, cc=5.)
We have the same situation as in the two-dimensional case: The endpoints and their iterates are points that already lie on the limit curve because they are not changed under further iterations. One can say that the endpoints and their iterates are related to the limit curve in a very simple way. On the other hand, the approximating polygons develop double points at these iterates and the result is that the approximations look much more confusing if the endpoints and their iterates are included in the iteration. This is why we offer the choice between iterating with and without the endpoints.

H.K.

Henon Map*

The Henon Map visualization gives the orbit under iteration of the map $(x, y) \to (y + 1 - aa\ x^2, bb\ x)$.

The default values are $aa = 1.4$ and $bb = 0.3$. The initial point is $(x, y) = (cc, dd)$ with the defaults $cc = 1.0, dd = 1.0$. The number of iterations plotted is ee, but the first ff iterates are omitted. The defaults are $ee = 3000$ and $ff = 20$.

In 3DXM, to move the finished image, drag the image with the mouse. To zoom in our out, drag vertically with the Shift key pressed. (If you zoom in, you might want to increase parameter ee using Settings > Set Parameters.)

To zoom into a particular region, hold down Command and then drag a rectangle in the window, then the program will zoom into that region of the Henon attractor, allowing you to see it in greater detail.

(Morphing aa and bb works, but there is no default morph, so first select Set Morphing... from the Settings menu to set up the morph—be sure to click the Init To Current Values button, then change aa0 aa1, bb0 and bb1.)

H.K.

---

* This file is from the 3D-XplorMath project. Please see: http://3D-XplorMath.org/

About The Feigenbaum Tree*

See also: Julia Set of $z \to (z^2 - c)$

The Feigenbaum Tree is one of the earliest examples of parameter dependent behavior of a dynamical system. The dynamical system in question is called the *Logistic Map*:

$$f_\mu(y) := 4\mu \cdot y(1-y),\ y \in [0,1],\ \mu \in [1/4,1].$$

Since both the parameter space, $[1/4,1]$, and the dynamical space, $[0,1]$, are 1-dimensional, one can illustrate in a (μ, y)-plane how the dynamical behavior changes as the parameter μ varies. The usual experiment (and the one used in 3DXM) goes as follows: Starting with a set of initial values $\{y_k;\ y_k \in [0,1],\ k = 1, \ldots, K\}$ (and with as many parameter values μ as one wants to handle) one computes many iterations $f_\mu^{\circ n}(y_k), n = 1, \ldots, N$ with N large.

If one plots only the iterations with say $n \geq 1000$, then one observes in the (μ, y)-plane the *Feigenbaum Tree*: for small μ the iterated points $f_\mu^{\circ n}(y_k)$ converge to a stable fixed point of the map f_μ, $y_f = f_\mu(y_f)$, $y_f := 1 - 1/4\mu$. Observe that the derivative f' at the fixed point is $2 - 4\mu \leq 0$. At $\mu = 3/4$ the derivative at the fixed point is -1, so that the fixed point stops being attractive. It turns out that for larger μ the orbit of period 2 is attractive for a while

* This file is from the 3D-XplorMath project. Please see:

http://3D-XplorMath.org/

– until μ reaches another bifurcation point after which an orbit of period 4 becomes attractive.

This period doubling "cascade" continues up to a certain μ-value, past which there is for a while no longer an attractive orbit. All this is clearly visible in the 3DXM demo. One should use the Action Menu entry: `Iterate Mouse Point Forward` to watch how arbitrary initial points are iterated and how these iterations converge to the attracting orbits of period 2^d in the left, period doubling, part of the Feigenbaum Tree. —*Speed-Up Note:* If one presses DELETE either during the default iterations or during the iteration of a point chosen by mouse, then all delays are skipped and the result of the iteration is reached quickly.

After the period doubling in the left part has been observed one wants to look at the right part of the Feigenbaum Tree more closely. The μ-interval which the illustration uses is the interval $[bb, cc]$. It can be changed in the Parameter entry of the Settings Menu. Since the attractive orbit of period 2 appears after $\mu = 0.75$, one loses only the simple attractors if one increases bb from 0.25 to 0.75, and one gains that the remaining part of the Tree is stretched by a factor of 3. In the same way one can magnify any part of the parameter space. Of course the dynamical space is always fully shown—unless one decides to use SHIFT+MOUSE to scale the image to see part of the dynamical space magnified. In this case translation using CONTROL+MOUSE-DRAG may be useful.

The most obvious feature in the right part of the Feigenbaum Tree are gaps, three fairly large ones and any number of thinner ones. The three large ones belong to parameter intervals where the map f_μ has attractive orbits of period 6, period 5, resp. period 3. If one magnifies a gap enough, one can experimentally check that the gaps belong to attractive orbits (use in the Action Menu `Iterate Mouse Point Forward`). One also observes that at the right end of these intervals the periods double again, and again. In other words, the Feigenbaum Tree illuminates, almost at the first glimpse, many properties of this 1-parameter family of iterated maps.

The Action Menu has been expanded by four entries `Iteration Invariant Density` (either with mouse choice of $aa = \mu$ or previous aa) and `Density Function` (again with mouse choice of aa or previous value). Before one chooses any of these one should look at `Iterate Mouse Point Forward`, where one sees how the iterated point, given by the vertical coordinate y, jumps around with fixed μ. The `Iteration Invariant Density` expands this: 1000 different y-values are chosen and represented in the left-most column on the screen. These points are iterated and shown in the second column, iterated again and shown in the third column, and so on, 400 times. Except for the first few columns one clearly sees a density pattern develop: all the vertical columns look essentially alike. This can be studied further with the entry `Density Function`: Here we count how often each pixel-sized interval of the dynamical (=vertical) interval is visited during the iterations and we plot the counting result (normalized to fit on the screen). We observe a function that describes the probability density with which each pixel interval is visited. – These demos explain why the curves that represent attractors do extend into the chaotic regions.

Finally we remark that the Feigenbaum Tree is related to the real part of the Mandelbrot set because the Mandelbrot set also parametrizes quadratic maps $z \to f_c(z) := (z^2 - c)$ according to their dynamical properties. If c is chosen from the big bottom apple then f_c has an attractive fixed point. As one passes on the real axis from the apple to the disk above it, the fixed point changes from attractive through indifferent to unstable and the orbit of period 2 becomes attractive. As one moves (always along the real axis) towards the top of the Mandelbrot set one continues to meet exactly the same kind of dynamical behavior as one sees in the Feigenbaum Tree. For more details see the documentation for *Julia Set of* $z \to (z^2 - c)$.

H.K.

User Defined Feigenbaum Iteration*

Please see first: About The Feigenbaum Tree

The question "*How can one find periodic attractors of, say, a family of Newton Iterations?*" led to the development of this exhibit. The user can input an iteration function that depends on one parameter aa (which is, as in the Feigenbaum case, represented horizontally). The dynamical space consists of some interval of arguments y of the function. We can view them as starting values of the iteration. The default iteration in 3DXM is the Newton iteration for the zeros of the polynomial $y \to (y^2-3)^2+3aa$.

The Feigenbaum picture is ideally suited to watch how the aa-family of iterations behaves: One quickly spots attractive fixed points or attractive orbits with small period; but one also observes the density curves in a seemingly chaotic region. If one expands the scale, i.e. stretches a very small aa-interval over the whole screen, then one sees easily whether there are in this interval periodic attractors, or whether still only chaos is visible (then choose a different aa-interval or expand the current interval further).

The remaining details are the same as for the classical Feigenbaum Tree and are explained in detail in the documentation for that exhibit.
H.K.

---

* This file is from the 3D-XplorMath project. Please see: http://3D-XplorMath.org/

The Area Preserving Henon Twist Map*

User Defined Example

The User Defined entry is designed to study the behaviour of 2-dimensional maps under forward iteration near an isolated, neutral fixed point. (We want a fixed point inside the window since otherwise most of the iterated points will move out of sight.) Our example is Henon's quadratic, area preserving twist map F:

$$F(x,y) := \begin{pmatrix} \cos(aa)\cdot x - \sin(aa)\cdot(y - e^{bb}\cdot x|x|) \\ \sin(aa)\cdot x + \cos(aa)\cdot(y - e^{bb}\cdot x|x|) \end{pmatrix}.$$

Henon used x^2 instead of $x|x|$ for the perturbation term. See below.

The main parameter aa controls the derivative of F at the fixed point $(0,0)$; $\mathrm{d}F|_{(0,0)}$ is the rotation matrix with angle aa. The behaviour of the iterations changes strongly with aa. Try also $-aa$. F is area preserving since the Jacobian determinant $\det(\mathrm{d}F) = 1$ everywhere.

By default $e^{bb} = 1$. This parameter serves to choose the size of the neighborhood of the fixed point, because of the scaling property

$$F(\vec{x}; e^{bb}) = e^{-bb}\cdot F(e^{bb}\cdot\vec{x}; 1).$$

We use $\exp(bb)$ instead of bb, because the scaling parameter is a multiplicative rather than an additive parameter.

---

* This file is from the 3D-XplorMath project. Please see:

http://3D-XplorMath.org/

The iteration is applied to the segment $[0,1] \cdot (cc, dd)$. The number of points on this segment is *tResolution*. The default number of iterations is $ee = 2000$. The next 2000 iterations are obtained from the Action Menu Entry: `Continue Curve Iteration`.
Since the graphic rendering is much slower than the the computation of iterations one can increase the parameter hh from its default value $hh = 1$ and then only one out of hh iterations is shown on the screen. This is useful if one needs to see the result of a large number of iterations. (For example $hh = 4 \cdot n$ in the case $aa = \pi/2$.)

The Action Menu Entry `Iterate Mouse Point Forward` allows to iterate a single point. During the selection the point coordinates appear on the screen. If DELETE is pressed during the iteration then the waiting time at each step is cancelled so that the point races through its orbit.

The Action Menu Entry `Choose Iteration Segment By Mouse` allows to Mouse-select initial and final point of a segment on which ff points will be distributed and iterated (by default $ff = 16$). The parameter hh speeds up the iteration as above. After the first ee iterations an Action Menu Entry is activated and allows to iterate further.

As ususal one can `translate` the image by dragging or one can `scale` it by depressing SHIFT and dragging vertically.

One can also `morph` the images. They change rather drastically with aa. As default morph bb is decreased so that the neighborhood of the fixed point gets expanded. One observes that most of the iterated points travel on *invariant curves* around the fixed point. Occasional periodic points clearly show up in the image. If aa is an irrational multiple of π then the visible periods do increase as the neighborhood of the fixed point expands with decreasing bb. (For the default morph the number ee of iterations is restricted to 500 to reduce waiting times.)

The Henon twist map can be written as a rotation plus a quadratic perturbation:

$$F(x,y) := \begin{pmatrix} \cos(aa) & -\sin(aa) \\ \sin(aa) & +\cos(aa) \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \text{perturb},$$

$$\text{perturb} := e^{bb} \cdot x|x| \cdot \begin{pmatrix} +\sin(aa) \\ -\cos(aa) \end{pmatrix}.$$

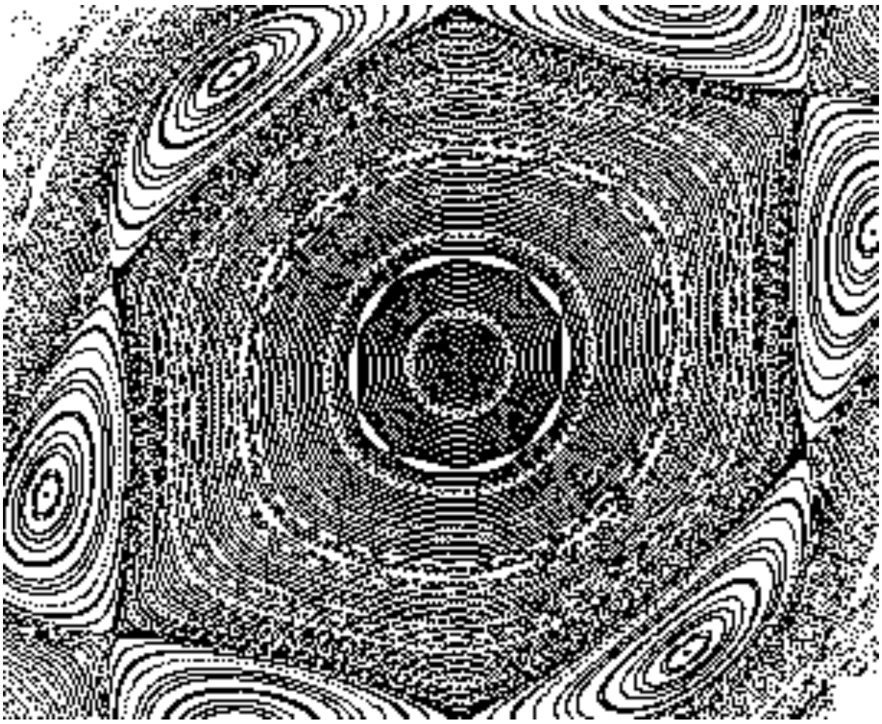
The scalar product between the perturbation and the tangent to the rotation circles is the

$$\begin{aligned} \text{Forward Perturbation} &= \\ &-e^{bb}|x|^3 \cdot (\sin^2(aa) + \cos^2(aa)). \end{aligned}$$

This explains why we changed the Henon map. Our negative forward perturbation means that the images under F stay behind the rotation image, and more so the larger $|x|$. This is the usual behavior of a monotone twist map.

Henon's perturbation has the factor x^3 instead of $|x|^3$, so that the twist in the left half plane partially cancels the twist in the right half plane. In our definition do the elliptical islands around periodic points appear more easily, while with Henon's definition the behaviour near the fixed point, in the case when aa is a rational multiple of π (e.g. $aa = \pi/2$), is much more complicated.

We recommend that users try out also Henon's definition and definitions of their own.



H.K.

The Mandelbrot Set And Its Julia Sets*

If one wants to study iterations of functions or mappings, $f^{\circ n} = f \circ \ldots \circ f$, as n becomes arbitrarily large then *Julia sets* are an important tool. They show up as the boundaries of those sets of points p whose iteration sequences $f^{\circ n}(p)$ converge to a selected *fixed point* $p_f = f(p_f)$. One of the best studied cases is the study of iterations in the complex plane given by the family of quadratic maps

$$z \longrightarrow f_c(z) := z^2 - c.$$

The *Mandelbrot set* will be defined as a set of parameter values c. It provides us with some classification of the different 'dynamical' behaviour of the functions f_c in the following sense: If one chooses a c-value from some specific part of the Mandelbrot set then one can predict rather well how the iteration sequences $z_{n+1} := f_c(z_n)$ behave.

1) Infinity is always an attractor. Or, more precisely, for each parameter value c we can define a Radius $R_c \geq 1$ such that for $|z| > R_c$ the iteration sequences $f^{\circ n}(z)$ converge to infinity. Proof: The triangle inequality shows that $|f_c(z)| \geq |z|^2 - |c|$ and then $|f_c(z)| > |z|$ is certainly true if $|z^2| - |c| > |z|$. Therefore it is sufficient to

* This file is from the 3D-XplorMath project. Please see: http://3D-XplorMath.org/

define $R_c := 1/2 + \sqrt{1/4 + |c|}$, which is the positive solution of $R^2 - R - |c| = 0$.
This implies: if we start the iteration with $z_1 > R_c$ then the absolute values $|z_n|$ increase monotonically—and indeed faster and faster to infinity. Moreover, any starting value z_1 whose iteration sequence converges to infinity will end up after *finitely many iterations* in this neighborhood of infinity, $U_\infty := \{z \in \mathbb{C} \mid |z| > R_c\}$. The set of all points whose iteration sequence converges to infinity is therefore an open set, called the attractor basin $A_\infty(c)$ of infinity.

2) Definition of the Julia set J_c. On the other hand, the attractor basin of infinity is never all of $\mathbb{C}$, since f_c has fixed points $z_f = 1/2 \pm \sqrt{1/4 + c}$ (and also points of period n, that satisfy a polynomial equation of degree 2^n, namely $f^{\circ n}(z) = z$).

Definition. The nonempty, compact boundary of the attractor basin of infinity is called the *Julia set of f_c*,

$$J_c := \partial A_\infty(c).$$

Example. If $c = 0$ then the exterior of the unit circle is the attractor basin of infinity, its boundary, the unit circle, is the Julia set J_0. The open unit disk is the attractor basin of the fixed point 0 of f_c. The other fixed point 1 lies on the Julia set; 1 is an expanding fixed point since $f_c'(1) = 2$; its iterated preimages $-1, \pm\mathbf{i}, \ldots$ all lie on the Julia set.

Qualitatively this picture persists for parameter values c near 0 because the smaller fixed point remains attractive.

However, the Julia set immediately stops being a smooth curve—it becomes a continuous curve that oscillates so wildly that no segment of it has finite length. Its image is one of those sets called a *fractal* for which a fractional dimension between 1 and 2 can be defined. Our rainbow coloration is intended to show J_c as a continuously parametrized curve. We next take a more carefull look at attractive fixed points.

3) c-values for which one fixed point of f_c is attractive.

There is a simple criterion for this: if the derivative at the fixed point satisfies $|f_c'(z_f)| = |2z_f| < 1$ then z_f is a linearly attractive fixed point; if $|2z_f| > 1$ then z_f is an expanding fixed point; if the derivative has absolute value 1 then no general statement is true (but interesting phenomena occur for special values of the derivative).

Since the sum of the two fixed points is 1, the derivative f_c' can have absolute value < 1 at most at one of them. Let w_c be that square root of $1 + 4c$ having a positive real part. Then $|1 - w_c|$ is the smaller of the absolute values (of the derivatives of f_c at the fixed points). The set of parameter values c with a (linearly) attractive fixed point of f_c is therefore the set $\{c \mid \ |1 - w_c| < 1\}$, or $\{c = (w^2 - 1)/4 \mid \ |1 - w| < 1\}$. In other words, the numbers $1 + 4c$ are the squares of numbers w that lie in a disk of radius one with 0 on its boundary. The apple shaped boundary is therefore the square of a circle through 0. It is called a *cardioid*.

4) The definition of the Mandelbrot set in the parameter plane.

The behavior of the iteration sequence $z_{n+1} := f_c(z_n)$ in the z-plane depends strongly on the value of the parameter c. It turns out that for those c satisfying $|c| > R_c$, the set of points z whose iteration sequences do *not* converge to infinity has area $= 0$. Such points are too rare to be found by trial and error, but one can still compute many as iterated preimages of an unstable fixed point. It follows from $|c| > R_c$ that only the points of the Julia set J_c do not converge to infinity. Moreover, the Julia set is no longer a curve, but is a totally disconnected set: no two points of the Julia set can be joined by a curve inside the Julia set. (In this case our coloration of J_c has no significance.)

The Mandelbrot set is defined by the opposite behaviour of the Julia sets:

$$\text{Mandelbrot Set} : \mathbf{M} := \{c \mid \; J_c \;\text{ is a connected set}\}$$

There is an 80 year old theorem by Julia or Fatou that says:

$$\begin{aligned}\mathbf{M} &= \{c \; ; \quad f_c^{\circ n}(0) \quad \text{stays bounded}\} \\ &= \{c \; ; \quad |f_c^{\circ n}(0)| < R_c \;\text{ for all } n\}.\end{aligned}$$

This provides us with an algorithm for determining the complement of $\mathbf{M}$; namely $c \notin \mathbf{M}$ if and only if the iteration sequence $\{f_c^{\circ n}(0)\}$ reaches an absolute value $> R_c$ for some positive integer n. (But, the closer c is to $\mathbf{M}$, the larger this termination number n becomes).

On the other hand, if f_c has an attractive fixed point, then it is also known that $\{f_c^{\circ n}(0)\}$ converges towards that fixed point. The interior of the cardioid described above is therefore part of the Mandelbrot set, and in fact it is a large part of it.

As *experiments* we suggest to choose c-values from the apple-shaped belly of the Mandelbrot set and observe how the Julia sets deform as c varies from 0 to the cardioid boundary. For an actual animation, choose the deformation interval with the mouse (Action Menu) and then select 'Morph' in the Animation Menu. To see how the derivative at the fixed point controls the iteration near the fixed point, choose 'Iterate Forward' (Action Menu) and watch how chosen points converge to the fixed point. This is very different for c from different parts of the Mandelbrot belly.

5) Attractive periodic orbits. As introduction let us determine the orbits of period 2, i.e., the fixed points of $f_c \circ f_c$ that are not also fixed points of f_c. Observe that:

$$\begin{aligned} f_c \circ f_c(z) - z &= z^4 - 2cz^2 - z + c^2 - c \\ &= (z^2 - z - c)(z^2 + z - c + 1). \end{aligned}$$

The roots of the first quadratic factor are the fixed points of f_c, the roots of the other quadratic factor are a pair of points that are not fixed points of f_c, but are fixed points of $f_c \circ f_c$, which means, they are an orbit of period 2, clearly the only one. Such an orbit is (linearly) attractive if the product of the derivatives at the points of the orbit has absolute value < 1. The constant coefficient in the quadratic equation is the product of its roots, i.e. the product of the points of period 2 is $1 - c$. Therefore:

The set of c-values for which the orbit of period 2 is attractive is the disk $\{c\ ; \quad |1 - c| < 1/4\}$.

Again, this disk is part of the Mandelbrot set since $\{f_c^{\circ n}(0)\}$ has the two points of period 2 as its only limit points.
The interior of the Mandelbrot set has only two components that are explicitly computable. These are the c-values giving attractive fixed points or attractive orbits of period 2. For example, the points of period 3 are the zeros of a polynomial of degree 6, namely:

$$\begin{aligned}
&\big(f_c \circ f_c \circ f_c(z) - z\big)/(z^2 - z - c)\\
&= z^6 + z^5 + (1 - 3c)z^4 + (1 - 2c)z^3 +\\
&\quad + (1 - 3c + 3c^2)z^2 + (c - 1)^2 z + 1 - c(c - 1)^2.
\end{aligned}$$

But since this polynomial cannot be factored (with c a parameter) into two polynomials of degree 3 it does not provide us with a description of the attractive orbits of period 3. However, it does give those c-values for which the period 3 orbits are superattractive (i.e. $(f^{\circ 3})'(\mathit{orbit\ point}) = 0$), since in this case the constant term must vanish. Approximate solutions of $1 - c(c - 1)^2 = 0$ are $c = 1.7549, c = 0.12256 \pm 0.74486i$. One can navigate the Mandelbrot set and observe that the complex solutions are between the two biggest blobs that touch the primary apple from either side.

Linearly attractive orbits always have c-values which belong to open subsets of the Mandelbrot set (in particular all the blobs touching the two explicit components), but the closure of these open subsets does not exhaust the Mandelbrot set. For example for $c = i$ the orbit of 0 is $0 \mapsto -i \mapsto -1-i \mapsto i \mapsto -1-i \ldots$, i.e., after two preliminary steps it reaches an orbit of period 2. Since this orbit stays clearly bounded we have $i \in \mathbf{M}$ (by the criterium quoted before). On the other hand, if the iteration $z \mapsto z^2 - i$ had any attractor (besides ∞), then the orbit of 0 would have to converge to the attracting orbit. Therefore there is no attractor and no attractor basin. In fact, the complement of the Julia set is the (simply connected) attractor basin of ∞. Because of its appearance, this Julia set is called a dendrite.

To generalize this observation, consider, for any c, the orbit of 0: $0 \mapsto -c \mapsto c^2 - c \mapsto c^4 - 2c^3 + c^2 - c \mapsto (c^4 - 2c^3 + c^2 - c)^2 - c \mapsto \ldots$. If 0 is on a periodic orbit for some c, then this orbit is superattractive. If the periodicity starts later then this periodic orbit may not be an attractor even though the orbit of 0 reaches it in finitely many steps. For example $c^2 - c$ is periodic of period 3, if $c^3 \cdot (c-2) \cdot (c^3 - 2c^2 + c - 1)^2 \cdot (c^6 - 2c^5 + 2c^4 - 2c^3 + c^2 + 1) = 0$; $c = 2$ is the largest point on the Mandelbrot set, the third factor has as roots the three c-values (mentioned before) for which the iteration has superattractive orbits of period 3. The last factor has the root $c = 1.239225555 + 0.4126021816 \cdot i$, its Julia set is another dendrite. A third dendrite is obtained, for example, if the 4^{th} point $c^4 - 2c^3 + c^2 - c$ in the orbit of 0 is a fixed point, which is the case if $c^4(c-2)(c^3 - 2c^2 + 2c^2 - 2) = 0$; here the last factor has the numerical solutions $c = 1.543689$ and $c = 0.2281555 \pm 1.1151425 \cdot i$.

6) Suggestions for experiments. The final entry in the Action Menu for the Julia set fractal is a hierachical menu with five submenus, each of which lists a number of related c-values that you may select. The c-values in these menus were selected because they typify either some special topological property of the associated Julia set or some dynamical property of the iteration dynamics of $z \mapsto z^2 - c$, and these properties are referenced by special abbreviations added to the menu item. (In addition some menu items also list a "name" that is in common use to refer to the Julia set, usually deriving from its shape). For convenience we will list in the next couple of pages all the items from these five menus, but first we explain the abbreviations used to describe them.

Abbreviations used in the following lists of interesting C-values. 'FP' means 'fixed point', the corresponding c-values are from the belly of the Mandelbrot set. 'cyc k' means 'cyclic of period k', the corresponding c-values are from the blobs directly attached to the belly; its Julia sets have a fixed point which is a common boundary point of k components of the attractor basin and the attractive orbit wanders cyclicly through these k components. 'per $2 \cdot 3$' means: this c-value has an attractor of period 6 and the

c-value is from a blob which is attached to the disk in $\mathbf{M}$ (which gives the attractive orbits of period 2). By contrast, 'per $3 \cdot 2$' means that the c-value is from the biggest blob which is attached to a period-3 blob (attached to the belly); its attractor has also period 6, but the open sets through which the attractive orbit travels are arranged quite differently in the two cases. One should compare both of them with the cyclic attractors of period 2 resp. 3. The abbreviation 'tch 1-2' means that the c-value is in the Mandelbrot set a common boundary point between the belly (i.e. the component of attractive fixed points) and the component of attractors of period 2. For the 'Siegel disks' see Nr. 8 of this ATO first; the column entry in the list gives the rotation number of the derivative (of the iteration map) at the fixed point. In the dendrite section of the list we mean by 'ev per 2' that the orbit of 0 is 'eventually periodic with period 2', as explained in Nr5 of this ATO. Finally, if $c \notin \mathbf{M}$ then the Julia set is a totally disconnected Cantor set and there are no such easy distinctions between different kinds of behaviour of the iteration on the Julia set (all other points are iterated to ∞).

Interesting C-values From the Action Submenus

Attractors Menu

C - values				Popular Name	Behaviour
0.0	+	0.0	· i	Circle	FP
0.0	+	0.1	· i	Rough Circle	FP
0.127	+	0.6435	· i	Near-Rabbit	FP
-0.353	-	0.1025	· i	Near-Dragon	FP
0.7455	+	0.0	· i	Near San Marco	FP
1.0	+	0.0	· i		cyc 2
1.0	+	0.2	· i		cyc 2
0.1227	+	0.7545	· i	Rabbit	cyc 3
1.756	+	0.0	· i	Airplane	cyc 3
-0.2818	+	0.5341	· i		cyc 4
1.3136	+	0.0	· i		per 2·2
-0.3795	+	0.3386	· i		cyc 5
0.5045	+	0.5659	· i		cyc 5
-0.3909	+	0.2159	· i		cyc 6
0.1136	+	0.8636	· i		per 3·2
1.1409	+	0.2409	· i	Rabbit's Shadow	per 2·3
-0.3773	+	0.1455	· i		cyc 7
-0.1205	+	0.6114	· i		cyc 7
-0.36	-	0.1	· i	Dragon	cyc 8
0.3614	+	0.6182	· i		cyc 8
-0.3273	+	0.5659	· i		per 4·2
1.0	+	0.2659	· i		per 2·4
1.3795	+	0.0	· i		per 2·2·2
0.0318	+	0.7932	· i	Rabbit Triplets	per 3·3
-0.0500	+	0.6318	· i		cyc 10
-0.4068	+	0.3409	· i		per 5·2
0.5341	+	0.6023	· i		per 5·2
0.9205	+	0.2477	· i		per 2·5
1.2114	+	0.1545	· i		per 2·5
0.6977	+	0.2818	· i		cyc 11
0.4864	+	0.6023	· i	Quintuple Rabbits	per 5·3
0.65842566307252	-	0.44980525145595	· i	Super Attractor	per 21

Interesting C-values From the Action Submenus.

C - values			Popular Name	Behaviour
Between Attractors Menu:				
0.75	+ 0.0	· i	San Marco	tch 1-2
1.25	+ 0.0	· i	S.Marco's Shadow	tch 2-2·2
0.125	+ 0.64952	· i	Balloon Rabbit	tch 1-3
-0.35676	+ 0.32858	· i		tch 1-5
Siegel Disks Menu:				
0.390540870218	+ 0.586787907347	· i		$2\pi \cdot i\cdot$ gold
-0.08142637539	+ 0.61027336571	· i		$2\pi \cdot i/\sqrt{2}$
0.66973645476	- 0.316746426417	· i		$2\pi \cdot i/\sqrt{5}$
One Simply Connected Open Component Menu:				
0.0	+ 1.0	· i	Dendrite	ev per 2
0.2281554936539	+ 1.1151425080399	· i	Dendrite	FP after 3
1.2392255553895	- 0.4126021816020	· i	Dendrite	ev per 3
-0.4245127190500	- 0.2075302281667	· i		FP after 7
1.1623415998840	+ 0.2923689338965	· i		per 2 after 7
Outside Mandelbrot set Menu:				
0.765	+ 0.12	· i	Cantor set	
-0.4	- 0.25	· i	Cantor set	
-0.4253	- 0.2078	· i	Cantor set	

An *experiment* which one should always make after one has computed a Julia set for some c from the Mandelbrot set: Remember from which part of $\mathbf{M}$ c came and then 'Iterate Forward' (Action Menu) mouse selected points until they visually converge to a periodic attractor. Observe how the shape of the Julia set lets one guess the period of its attractor and how this relates to the position of c in $\mathbf{M}$.

7) Computation of the Julia set. In addition to the attractor at infinity there is at most one further attractor in the $z \to (z^2 - c)$ systems. All preimages of non-attractive fixed points or non-attractive periodic orbits are points on the Julia set. Since $|f_c'| > 1$ along the Julia set (with some exceptions), the preimage computation is numerically stable. This is a common method for computing Julia sets.

In our program we compute preimages starting from the circle $\{z;\ |z| = R_c\}$ around the wanted Julia set. Under inverse images these curves converge from outside to the Julia set. Such an approximation by curves allows us to color the Julia set in a continuous way and thus emphasize that, despite its wild looks it is the image of a continuous curve—at least for $c \in M$, otherwise we recall that the Julia set is totally disconnected, so in particular is not the image of a curve. Our computation works also for $c \notin \mathbf{M}$, since our 'curves' of course consist of only finitely many points, and the inverse images of each of these points have their limit points on the Julia set.

8) Self-similarity of a Julia set. A well advertised property of these Julia sets is their so called 'self-similarity'. By this one means: Take a small piece of the Julia set and enlarge it; the result looks very much like a larger piece of that same Julia set. For the Julia sets of the present quadratic iterations, this self-similarity is easily understood from the definitions: The iteration map f_c is a *conformal* map that stretches its Julia set 1:2 onto itsself. In other words, the iteration map itself maps any small piece of its Julia set to roughly twice as large a piece, and it does so in an angle preserving way. From this point of view self-similarity should come as no surprise.

9) Siegel Disks. We next would like to explain an experimentally observable phenomenon that mathematicians find truly surprising, but this needs a little preparation.

Simplifying Mappings. Imagine that we want to describe something on the surface of the earth, for example a walk. For a long time, people have been more comfortable giving the description on a map of the earth rather than on the earth itself. Mathematicians view a map of the earth more precisely as a mapping F from the earth to a piece of paper and they describe (or even prove) properties of the map by properties of the mapping F. An example of a useful property is 'conformality': angles between curves on the earth are the same as the angles between the corresponding curves on the map.

Conjugation by simplifying mappings. Let us consider one of the above iteration maps f_c and assume that it has an attractive fixed point z_f with derivative $q := f_c'(z_f)$, $|q| < 1$. The simplest map with the same derivative is the linear map $L(z) := q \cdot z$. It is the definition of derivative that the behaviour of f_c near the fixed point looks 'almost' like the behaviour of L near its fixed point 0, and 'almost' means: the smaller the neighborhoods of the fixed points (on which the maps are compared) the more the maps look alike. But more is true for f_c because of the assumption $|q| < 1$, we have the *theorem*: There exists on a fixed(!) neighborhood of the fixed point z_f a simplifying map F to a neighborhood of $0 \in \mathbb{C}$ that makes f_c look *exactly* like its linear approximation L, by which we mean: $f_c = F^{-1} \circ L \circ F$. In particular, this tells us everything about the iterations of f_c in terms of the iterations of L because they also look the same when compared using ('conjugation' by) F: $\quad f_c^{\circ n} = F^{-1} \circ L^{\circ n} \circ F$.

Siegel's Theorem. The previous result cannot be true in general if $|q| = 1$. For example if $q = \exp(2\pi i/k)$, then $L^{\circ k} = \mathrm{id}$, but $f_c^{\circ k} \neq \mathrm{id}$. Therefore they cannot look alike under a simplifying (i.e., 'conjugating') mapping F. But if $z \to q \cdot z$ is an irrational rotation and if some further condition is satisfied, for example if $q := \exp(2\pi i/\sqrt{2})$, then there is again such a simplifying mapping F such that f_c looks near that fixed point exactly like its linearization, namely: $\quad f_c = F^{-1} \circ L \circ F$.

Experiment. While Siegel's proof insures only *very* small neighborhoods on which the simplifying mapping F exists, these neighborhoods are surprisingly large in the present case. One can ‘observe’ Siegel’s theorem by first choosing $c = ((1-q)^2 - 1)/4$ such that $f_c'(z_f) = q$ with $q = \exp(2\pi i \cdot k/\sqrt{p})$, p prime (or square free), then one chooses points on a fairly straight radial curve from the fixed point almost out to the Julia set. Under repeated iterations these points travel on closed curves around the fixed point (’circles’ when viewed with F) and all of them travel with the same angular velocity, i.e., one observes that they remain on non-intersecting radial curves.

H.K.