1 Finding cyclic points

Or however these points are called that is, idk.

At the center of each cardioid or disk in the mandelbrot set sits one point which ends up exactly at zero after a finite iteration count $n-1$, and therefore at its initial value z_0 the iteration after that; this means that the point immediately gets stuck in a periodic cycle of length n, hence the name.

1.1 Motivation

Finding these points can be useful for a number of reasons; for one, they always indicate the presence of a minibrot cardioid or disk around them, meaning they can be used to automatically find and zoom towards the nearest minibrot. If the type and size of the cardioid or disk is known, they can also be used to clip parts of that disk, so these typically expensive to compute points can be discarded as inside the mandelbrot set in a matter of milliseconds.

Also, knowing the cycle length of a specific minibrot or disk can be useful as well; it can be used to estimate the iteration count required to render a reasonably detailed image of that minibrot, as each iteration of that minibrot or its julia sets requires following along one cycle of that minibrot's main cardioid's orbit, taking n actual iterations to complete.

Finally, being able to automatically analyze the location, cycle length and orientation of minibrot sets may eventually enable us to build an algorithm that can automatically skip iterating through the full cycle and just compute iterations on the minibrot directly; this would allow rendering nested minibrots and julia sets in $O(log(n))$-complexity.

1.2 Computation

So, given all of that, how can these points be computed?

The first key insight here is that just like the function values z_n after n iterations, their derivatives $\frac{dz_n}{dz_0}$ can also be computed iteratively:

$$\begin{aligned} z_{n+1} &= z_n^2 + z_0 \\ \frac{dz_{n+1}}{dz_0} &= \frac{d(z_n^2 + z_0)}{dz_0} \\ &= \frac{dz_n^2}{dz_0} + 1 \\ &= 2\frac{dz_n}{dz_0} z_n + 1. \end{aligned}$$

$$\begin{aligned}
\frac{d^2 z_{n+1}}{dz_0^2} &= \frac{d(\frac{dz_{n+1}}{dz_0})}{dz_0} \\
&= \frac{d(2\frac{dz_n}{dz_0}z_n + 1)}{dz_0} \\
&= 2\frac{d(\frac{dz_n}{dz_0}z_n)}{dz_0} \\
&= 2 \cdot \frac{d^2 z_n}{dz_0^2} z_n + 2 \cdot (\frac{dz_n}{dz_0})^2. \\
\frac{d^3 z_{n+1}}{dz_0^3} &= 2 \cdot \frac{d^3 z_n}{dz_0^3} z_n + 6 \cdot \frac{d^2 z_n}{dz_0^2} \cdot \frac{dz_n}{dz_0} \\
\frac{d^4 z_{n+1}}{dz_0^4} &= 2 \cdot \frac{d^4 z_n}{dz_0^4} z_n + 8 \cdot \frac{d^3 z_n}{dz_0^3} \cdot \frac{dz_n}{dz_0} + 6 \cdot (\frac{d^2 z_n}{dz_0^2}) \\
\frac{d^5 z_{n+1}}{dz_0^5} &= 2 \cdot \frac{d^5 z_n}{dz_0^5} z_n + 10 \cdot \frac{d^4 z_n}{dz_0^4} \cdot \frac{dz_n}{dz_0} + 20 \cdot \frac{d^3 z_n}{dz_0^3} \cdot \frac{d^2 z_n}{d^2 z_0} \\
&\ldots
\end{aligned}$$

Generally, the kth derivative can be expressed as

$$\frac{d^k z_{n+1}}{dz_0^k} = \sum_{i=0}^{k} \binom{k}{i} \cdot \frac{d^{k-i} z_n}{dz_0^{k-i}} \cdot \frac{d^i z_n}{dz_0^i}$$

for all $k \geq 2$; however, anything beyond the third derivative is as far as I know rarely ever needed, and therefore largely just a curiosity.

The first few derivatives are useful however as they can be used to approximate the nth iteration of points close to the one the derivatives where computed for:

$$x_n \approx z_n + (x_0 - z_0)\frac{dz_n}{dz_0}.$$

By calculating the root of this approximation, one can also generate an estimate for a root of the nth iteration of the set. This can then be used as the new base point for the next approximation to generate an even better estimate, basically using Newton's method; that way, a root of the nth iteration of the mandelbrot set and therefore a point with cycle length $n + 1$ can be computed recursively as long as the estimate actually converges.

Based on this, a naive approach to find points with cyclic orbits close to a given base point z_0 would be to compute z_n and $\frac{dz_n}{dz_0}$ for each iteration n and each time simply try if Newton's method converges; however, as computing the values and derivatives of other points as required by Newton's method has linear complexity regarding iteration count, doing this for each iteration has a complexity of $O(n^2)$, making it very slow for high iteration counts.

One way to speed this up is therefore to perform some simple test to see if Newtons method could converge before actually performing it; another is to improve the initial estimate, so that Newton's method converges quicker when you do actually perform it. One simple way to do the latter is to use a quadratic approximation instead of a linear one and compute the first estimate based on that using the quadratic formula:

$$x_n \approx z_n + (x_0 - z_0)\frac{dz_n}{dz_0} + \frac{1}{2}(x_0 - z_0)^2 \frac{d^2 z_n}{dz_0^2}.$$

$$x_0' = z_0 + \frac{-\frac{dz_n}{dz_0} \pm \sqrt{(\frac{dz_n}{dz_0})^2 - 2\frac{d^2 z_n}{dz_0^2} z_n}}{\frac{d^2 z_n}{dz_0^2}}.$$

Out of the two roots produced by this, one can then simply choose the one that is closer to the base point to get an already slightly better first estimate. It might be possible to get an even better estimate by including even more higher derivatives in the polynomial approximation and using Newton's method on that first; however, I haven't tried that yet so I'm not sure if it is actually faster.

Regardless of which method is used to compute this estimate, as long as the derivative $\frac{dx_0'}{dz_0}$ of that estimate with respect to z_0 can also be computed, a simple test to decide whether to use Newton's method this iteration or not would be to test if the absolute value of that derivative is less than one (or some other constant, for that matter); that way, estimates which are relatively unstable and dependent on z_0 can be quickly discarded. As an example, for the quadratic estimate described above this derivative could be computed like this:

$$\begin{aligned}
\frac{dx_0'}{dz_0} &= \frac{d\left(z_0 + \frac{(x_0'-z_0)\cdot\frac{d^2z_n}{dz_0^2}}{\frac{d^2z_n}{dz_0^2}}\right)}{dz_0} \\
&= 1 + \frac{\frac{d\left((x_0'-z_0)\cdot\frac{d^2z_n}{dz_0^2}\right)}{dz_0}\cdot\frac{d^2z_n}{dz_0^2} - \left((x_0'-z_0)\cdot\frac{d^2z_n}{dz_0^2}\right)\cdot\frac{d^3z_n}{dz_0^3}}{\left(\frac{d^2z_n}{dz_0^2}\right)^2} \\
&= 1 + \frac{\frac{d\left(-\frac{dz_n}{dz_0}\pm\sqrt{(\frac{dz_n}{dz_0})^2-2\frac{d^2z_n}{dz_0^2}z_n}\right)}{dz_0} - (x_0'-z_0)\cdot\frac{d^3z_n}{dz_0^3}}{\frac{d^2z_n}{dz_0^2}} \\
&= 1 + \frac{-\frac{d^2z_n}{dz_0^2} \pm \left(\frac{\frac{d((\frac{dz_n}{dz_0})^2-2\frac{d^2z_n}{dz_0^2}z_n)}{dz_0}}{2\sqrt{(\frac{dz_n}{dz_0})^2-2\frac{d^2z_n}{dz_0^2}z_n}}\right) - (x_0'-z_0)\cdot\frac{d^3z_n}{dz_0^3}}{\frac{d^2z_n}{dz_0^2}} \\
&= \frac{\pm\frac{2\cdot\frac{dz_n}{dz_0}\cdot\frac{d^2z_n}{dz_0^2}-2(\frac{d^3z_n}{dz_0^3}z_n+\frac{d^2z_n}{dz_0^2}\cdot\frac{dz_n}{dz_0})}{2\sqrt{(\frac{dz_n}{dz_0})^2-2\frac{d^2z_n}{dz_0^2}z_n}} - (x_0'-z_0)\cdot\frac{d^3z_n}{dz_0^3}}{\frac{d^2z_n}{dz_0^2}} \\
&= \frac{\frac{-\frac{d^3z_n}{dz_0^3}z_n}{\pm\sqrt{(\frac{dz_n}{dz_0})^2-2\frac{d^2z_n}{dz_0^2}z_n}} - (x_0'-z_0)\cdot\frac{d^3z_n}{dz_0^3}}{\frac{d^2z_n}{dz_0^2}} \\
&= \frac{\frac{-\frac{d^3z_n}{dz_0^3}z_n}{(x_0'-z_0)\cdot\frac{d^2z_n}{dz_0^2}+\frac{dz_n}{dz_0}} - (x_0'-z_0)\cdot\frac{d^3z_n}{dz_0^3}}{\frac{d^2z_n}{dz_0^2}} \\
&= -\frac{\frac{d^3z_n}{dz_0^3}}{\frac{d^2z_n}{dz_0^2}}\cdot\left(\frac{z_n}{(x_0'-z_0)\cdot\frac{d^2z_n}{dz_0^2}+\frac{dz_n}{dz_0}} + (x_0'-z_0)\right).
\end{aligned}$$

Calculating this, x_0' and all the derivatives necessary for that for every single iteration on the base point z_0 just to see if Newton's method could be used is obviously kind of slow itself; however, the crucial part here is that this doesn't require any derivative samples from other points each iteration, and as such runs with nearly linear complexity in regards to the iteration count n (though it still requires these samples to use Newton's method for iterations where it could converge according to the test - if you factor that in, the complexity is probably something like $O(n\log(n))$).

1.3 Finding minibrots - WIP

Finding points with cyclic orbits is all well and good, but to really take advantage of that we also need to be able to locate the minibrots around them; that is, detect whether the point belongs to a cardioid or disk, and what the orientation and size of that cardioid or disk is.

To do this, let's assume we already have a perfectly n-cyclic starting point z_0, meaning that z_{n-1} is exactly zero and is the first element in the sequence to do so. The $n-1$th iteration of the close vicinity around z_0 can as usual be approximated as $x_{n-1} \approx z_{n-1} + (x_0 - z_0)\frac{dz_{n-1}}{dz_0}$; but since z_{n-1} is 0, this then simplifies down to $x_{n-1} \approx (x_0 - z_0)\frac{dz_{n-1}}{dz_0}$. Let's call this value d_0. The next iteration can then be approximated as $x_n \approx x_0 + d_0^2$; the iteration after that as

$$\begin{aligned} x_{n+1} &\approx (x_0 + d_0^2)^2 + x_0 \\ &\approx x_0^2 + 2x_0 d_0^2 + d_0^4 + x_0 \\ &\approx x_1 + 2x_0 d_0^2 + d_0^4. \end{aligned}$$

However, as d_0^2 is already very, very close to zero, the term d_0^4 and any terms multiplied by it can be ignored as they are basically negligible in comparison; also, $x_0 d_0^2$ can be replaced with $z_0 d_0^2$ for similar reasons:

$$\begin{aligned} x_{n+1} &\approx x_1 + 2z_0 d_0^2 \\ x_{n+2} &\approx (x_1 + 2z_0 d_0^2)^2 + x_0 \\ &\approx x_1^2 + 4z_0 x_1 d_0^2 + 4z_0^2 d_0^4 + x_0 \\ &\approx x_2 + 4z_0 z_1 d_0^2 \\ x_{n+3} &\approx x_3 + 8z_0 z_1 z_2 d_0^2 \\ &... \end{aligned}$$

This pattern continues until we hit the next multiple of n, minus one; let's call this value d_1:

$$\begin{aligned} x_{2n-1} &\approx x_{n-1} + 2^{n-1}(z_0 z_1 ... z_{n-2}) d_0^2 \\ d_1 &\approx d_0 + 2^{n-1}(z_0 z_1 ... z_{n-2}) d_0^2. \end{aligned}$$

Now it should already become apparent why these minibrots even exist; x_{2n} will then be $x_0 + d_1^2$, x_{3n-1} will be $d_0 + 2^{n-1}(z_0 z_1 ... z_{n-2}) d_1^2$ and so on; generally, the sequence d_k as defined by the values of $x_{(k+1)n-1}$ will approximately follow the equation $d_{k+1} = 2^{n-1}(z_0 z_1 ... z_{n-2}) d_k^2 + d_0$ for as long as it stays small enough, completing one iteration every n iterations of x.

To get rid of the factor $2^{n-1}(z_0 z_1 ... z_{n-2})$ and turn this into an actual mandelbrot set equation, we can just multiply both sides of the equation with it:

$$2^{n-1}(z_0 z_1 ... z_{n-2}) d_{k+1} = (2^{n-1}(z_0 z_1 ... z_{n-2}) d_k)^2 + 2^{n-1}(z_0 z_1 ... z_{n-2}) d_0$$

It is therefore the term $2^{n-1}(z_0 z_1 ... z_{n-2}) d_k$ that actually iterates just like the mandelbrot set, and the term $2^{n-1}(z_0 z_1 ... z_{n-2}) d_0$ or $2^{n-1}(z_0 z_1 ... z_{n-2})(x_0 - z_0)\frac{dz_{n-1}}{dz_0}$ that acts as the starting value for it; as such, a number a indicating the scale and orientation of the minibrot set can be computed as

$$a = \frac{1}{2^{n-1}(z_0 z_1 ... z_{n-2})\frac{dz_{n-1}}{dz_0}}.$$

2 Generalizing for other fractals - WIP

Applying these same methods to other popular fractals such as the mandelbar set (also commonly referred to as tricorn) or the burning ship fractal is slightly complicated by the fact that these are based on non-conformal mappings (which can even be seen visually; they contain heavily skewed parts), meaning that their derivative can only be expressed as a Jacobian matrix.

Oh, and sorry in advance for all the bad notation coming up in the following part; I'm not a mathematician, and the internet wasn't really helpful for finding out how to denote Jacobian matrices of complex functions either.

To see why we actually need the Jacobian, let's take a look at, for example, the mandelbar set first. It is defined by the sequence $z_{n+1} = \bar{z}_n^2 + z_0$, where $\bar{z}$ denotes the complex conjugate of z. This is already not differentiable in the usual sense; when looking at any small region of points, it is not only scaled, rotated and translated here, but also flipped, meaning that a complex number representing the scaling factor and rotation alone is no longer enough to describe the applied transformation. Instead, a 2×2 matrix is needed that tells us how both the real and imaginary component of z_n change with respect to both the real and imaginary component of z_0; This matrix is the Jacobian.

In the following, we'll use $\frac{\partial z_n}{\partial z_0}$ to refer to the Jacobian matrix $\begin{bmatrix} \frac{d\,\mathrm{re}(z_n)}{d\,\mathrm{re}(z_0)} & \frac{d\,\mathrm{re}(z_n)}{d\,\mathrm{im}(z_0)} \\ \frac{d\,\mathrm{im}(z_n)}{d\,\mathrm{re}(z_0)} & \frac{d\,\mathrm{im}(z_n)}{d\,\mathrm{im}(z_0)} \end{bmatrix}$, where $\mathrm{re}(z)$ and $\mathrm{im}(z)$ are the real and imaginary components of z, and $(\frac{\partial z_n}{\partial z_0})_{rr}$, $(\frac{\partial z_n}{\partial z_0})_{ri}$, $(\frac{\partial z_n}{\partial z_0})_{ir}$ and $(\frac{\partial z_n}{\partial z_0})_{ii}$ to refer to the individual entries of this matrix. Also, we'll use $M(z)$ to refer to the matrix $\begin{bmatrix} re(z) & -im(z) \\ im(z) & re(z) \end{bmatrix}$; that way, multiplying the derivative with a number z_n as we did earlier can now be done by multiplying $M(z_n)$ with the Jacobian matrix.

With all of that said, let's start by first figuring out how to compute these derivatives again:

$$\begin{aligned}
z_{n+1} &= \bar{z}_n^2 + z_0 \\
\frac{\partial z_{n+1}}{\partial z_0} &= \frac{\partial(\bar{z}_n^2 + z_0)}{\partial z_0} \\
&= \frac{\partial \bar{z}_n^2}{\partial z_0} + \frac{\partial z_0}{\partial z_0} \\
&= \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \cdot \frac{\partial z_n^2}{\partial z_0} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix} \cdot M(z_n) \cdot \frac{\partial z_n}{\partial z_0} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\
\frac{\partial^2 z_{n+1}}{\partial z_0^2} &= \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix} \cdot \frac{\partial(M(z_n) \cdot \frac{\partial z_n}{\partial z_0})}{\partial z_0} \\
&= \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix} \cdot \left(M(z_n) \cdot \frac{\partial^2 z_n}{\partial z_0^2} + 2 \cdot (\frac{\partial z_n}{\partial z_0})^2 \right)
\end{aligned}$$