1 Finding orbit 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; since zero is also the starting value, this means that the point immediately gets stuck in a periodic cycle of that 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 orbit 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 the minibrot's main cardioid's orbit, taking n actual iterations to complete.

Finally, being able to automatically analyze the location, orbit 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:

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

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

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 z for 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 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 that 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:

$$\begin{aligned}
\frac{dx'_0}{dz_0} &= \frac{d\left(z_0 + \frac{-\frac{dz_n}{dz_0} \pm \sqrt{(\frac{dz_n}{dz_0})^2 - 2\frac{d^2z_n}{dz_0^2}z_n}}{\frac{d^2z_n}{dz_0^2}}\right)}{dz_0} \\
&= 1 + \frac{\frac{d\left(-\frac{dz_n}{dz_0} \pm \sqrt{(\frac{dz_n}{dz_0})^2 - 2\frac{d^2z_n}{dz_0^2}z_n}\right)}{dz_0} \cdot \frac{d^2z_n}{dz_0^2} - \left(-\frac{dz_n}{dz_0} \pm \sqrt{(\frac{dz_n}{dz_0})^2 - 2\frac{d^2z_n}{dz_0^2}z_n}\right) \cdot \frac{d^3z_n}{dz_0^3}}{(\frac{d^2z_n}{dz_0^2})^2} \\
&= 1 + \frac{\left(-\frac{d^2z_n}{dz_0^2} \pm \frac{\frac{d((\frac{dz_n}{dz_0})^2 - 2\frac{d^2z_n}{dz_0^2}z_n)}{dz_0}}{2\sqrt{(\frac{dz_n}{dz_0})^2 - 2\frac{d^2z_n}{dz_0^2}z_n}}\right) \cdot \frac{d^2z_n}{dz_0^2} - ((x'_0 - z_0) \cdot \frac{d^2z_n}{dz_0^2}) \cdot \frac{d^3z_n}{dz_0^3}}{(\frac{d^2z_n}{dz_0^2})^2} \\
&= 1 + \frac{\left(-\frac{d^2z_n}{dz_0^2} \pm \frac{2 \cdot \frac{dz_n}{dz_0} \cdot \frac{d^2z_n}{dz_0^2} - 2(\frac{d^3z_n}{dz_0^3}z_n + \frac{d^2z_n}{dz_0^2} \cdot \frac{dz_n}{dz_0})}{2\sqrt{(\frac{dz_n}{dz_0})^2 - 2\frac{d^2z_n}{dz_0^2}z_n}}\right) - (x'_0 - z_0) \cdot \frac{d^3z_n}{dz_0^3}}{\frac{d^2z_n}{dz_0^2}} \\
&= \frac{\frac{-\frac{d^3z_n}{dz_0^3}z_n}{\pm\sqrt{(\frac{dz_n}{dz_0})^2 - 2\frac{d^2z_n}{dz_0^2}z_n}} - (x'_0 - z_0) \cdot \frac{d^3z_n}{dz_0^3}}{\frac{d^2z_n}{dz_0^2}} \\
&= \frac{\frac{-\frac{d^3z_n}{dz_0^3}z_n}{(x'_0 - z_0) \cdot \frac{d^2z_n}{dz_0^2} + \frac{dz_n}{dz_0}} - (x'_0 - z_0) \cdot \frac{d^3z_n}{dz_0^3}}{\frac{d^2z_n}{dz_0^2}} \\
&= -\frac{\frac{d^3z_n}{dz_0^3}}{\frac{d^2z_n}{dz_0^2}} \cdot \left(\frac{z_n}{(x'_0 - z_0) \cdot \frac{d^2z_n}{dz_0^2} + \frac{dz_n}{dz_0}} + (x'_0 - z_0)\right).
\end{aligned}$$

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 $O(n\log(n))$).