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 s indicating the scale and orientation of the minibrot set can be computed as

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

1.4 Skipping iterations - very WIP indeed

[just image some explanatory text here for now, gotta add that later]. Let z_0 again be a point with an n-cyclic orbit, x_0 a point very close to that, and $d_i := x_i - z_i$ for all $i \in \mathbb{N}_0$:

$$
\begin{aligned}
d_{i+1} &= x_{i+1} - z_{i+1} \\
&= x_i^2 + x_0 - z_i^2 - z_0 \\
&= (x_i + z_i)(x_i - z_i) + d_0 \\
&= d_i^2 + 2z_i d_i + d_0.
\end{aligned}
$$

As long as d_i is way smaller than z_i, the term d_i^2 is also way smaller than $2z_i d_i$; in that case, it can just be ignored, so d_{i+1} can be approximated as $2z_i d_i + d_0$. Let's use this to approximate the points following d_{kn} for an $k \in \mathbb{N}$:

$$
d_{kn+1} \approx 2z_{kn} d_{kn} + d_0.
$$

Since the orbit of z_0 is n-cyclic, z_{kn} must be equal to z_0; likewise, z_{kn+1} would be equal to z_1 and so on:

$$
\begin{aligned}
d_{kn+1} &\approx 2z_0 d_{kn} + d_0 \\
d_{kn+2} &\approx 2z_{kn+1} d_{kn+1} + d_0 \\
&\approx 2z_1(2z_0 d_{kn} + d_0) + d_0 \\
&\approx 4z_0 z_1 d_{kn} + (2z_1 + 1)d_0
\end{aligned}
$$

If the last term here already looks kind of familiar, that's because it is similar to $2z_1(2z_0+1)+1$ or $2z_1\frac{dz_1}{dz_0}$, which we've previously shown to be $\frac{dz_2}{dz_0}$; as such, we can simplify that term by writing it in terms of that:

$$
\begin{aligned}
d_{kn+2} &\approx 4z_0 z_1 d_{kn} + (2z_1 + 1)d_0 \\
&= 4z_0 z_1 d_{kn} + (2z_1(1 + 2z_0 - 2z_0) + 1)d_0 \\
&= 4z_0 z_1 d_{kn} + (2z_1(2z_0 + 1) + 1 - 4z_0 z_1)d_0 \\
&= 4z_0 z_1 d_{kn} + (\frac{dz_2}{dz_0} - 4z_0 z_1)d_0 \\
d_{kn+3} &\approx 2z_{kn+2} d_{kn+2} + d_0 \\
&\approx 2z_2(4z_0 z_1 d_{kn} + (\frac{dz_2}{dz_0} - 4z_0 z_1)d_0) + d_0 \\
&= 8z_0 z_1 z_2 d_{kn} + (2z_2\frac{dz_2}{dz_0} - 8z_0 z_1 z_2 + 1)d_0 \\
&= 8z_0 z_1 z_2 d_{kn} + (\frac{dz_3}{dz_0} - 8z_0 z_1 z_2)d_0 \\
&\ldots \\
d_{(k+1)n-1} &\approx 2^{n-1} z_0 z_1 \ldots z_{n-2} d_{kn} + (\frac{dz_{n-1}}{dz_0} - 2^{n-1} z_0 z_1 \ldots z_{n-2})d_0.
\end{aligned}
$$

At this point, the pattern breaks; since $z_{(k+1)n-1}$ must be equal to zero, our assumption that d_i is smaller than z_i is no longer true, so our previous approximation $2z_i d_i + d_0$ is no longer that accurate. Instead, the term $2z_i d_i$ is now equal to zero:

$$
\begin{aligned}
d_{(k+1)n} &= x_{(k+1)n} - z_{(k+1)n} \\
&= x_{(k+1)n-1}^2 + x_0 - z_{(k+1)n-1}^2 - z_0 \\
&= (z_{(k+1)n-1} + d_{(k+1)n-1})^2 + d_0 \\
&= d_{(k+1)n-1}^2 + d_0 \\
&\approx (2^{n-1} z_0 z_1 \ldots z_{n-2} d_{kn} + (\frac{dz_{n-1}}{dz_0} - 2^{n-1} z_0 z_1 \ldots z_{n-2})d_0)^2 + d_0.
\end{aligned}
$$

As before, the term $2^{n-1}z_0z_1...z_{n-2}$ appears in this equation, this time even twice; let's call it a for now to clean things up:

$$d_{(k+1)n} \approx (ad_{kn} + (\frac{dz_{n-1}}{dz_0} - a)d_0)^2 + d_0.$$

With this, we have again found an approximation that computes n iterations at once. The sequence $(d_{kn})_{k\in\mathbb{N}_0}$ produced by this is essentially the same as an orbit in the mandelbrot set, just scaled, rotated and translated; to show this, let's take a look at the sequence $\Delta_k := a^2d_{kn} + a(\frac{dz_{n-1}}{dz_0} - a)d_0$:

$$\begin{aligned}
\Delta_{k+1} &= a^2d_{(k+1)n} + a(\frac{dz_{n-1}}{dz_0} - a)d_0 \\
&\approx a^2((ad_{kn} + (\frac{dz_{n-1}}{dz_0} - a)d_0)^2 + d_0) + a(\frac{dz_{n-1}}{dz_0} - a)d_0 \\
&= (a^2d_{kn} + a(\frac{dz_{n-1}}{dz_0} - a)d_0)^2 + a^2d_0 + a(\frac{dz_{n-1}}{dz_0} - a)d_0 \\
&= \Delta_k^2 + \Delta_0.
\end{aligned}$$

In fact, all subsequences of the form $(d_{kn+i})k \in \mathbb{N}_0$ for $0 \leq i \leq n-1$ can be approximate as scaled, rotated and translated versions of this sequence Δ; this can be shown by rearranging and then substituting in our previously shown formula $d_{kn+i} \approx 2^iz_0z_1...z_{i-1}d_{kn} + (\frac{dz_i}{dz_0} - 2^iz_0z_1...z_{i-1})d_0$:

$$\begin{aligned}
d_{kn+i} &\approx 2^iz_0z_1...z_{i-1}d_{kn} + (\frac{dz_i}{dz_0} - 2^iz_0z_1...z_{i-1})d_0 \\
\Rightarrow d_{kn} &\approx \frac{d_{kn+i} - (\frac{dz_i}{dz_0} - 2^iz_0z_1...z_{i-1})d_0}{2^iz_0z_1...z_{i-1}}
\end{aligned}$$

$$\begin{aligned}
\Delta_k &= a^2d_{kn} + a(d\frac{z_{n-1}}{dz_0} - a)d_0 \\
&\approx a^2(\frac{d_{kn+i} - (\frac{dz_i}{dz_0} - 2^iz_0z_1...z_{i-1})d_0}{2^iz_0z_1...z_{i-1}}) + a(\frac{dz_{n-1}}{dz_0} - a)d_0 \\
&= a^2\frac{d_{kn+i}}{2^iz_0z_1...z_{i-1}} - a^2\frac{\frac{dz_i}{dz_0}d_0}{2^iz_0z_1...z_{i-1}} + a\frac{dz_{n-1}}{dz_0}d_0 \\
&= a(2^{n-i-1}z_iz_{i+1}...z_{n-2})d_{kn+i} + a(\frac{dz_{n-1}}{dz_0} - (2^{n-i-1}z_iz_{i+1}...z_{n-2})\frac{dz_i}{dz_0})d_0.
\end{aligned}$$

For $i = 0$, this produces exactly the equation we used to define Δ_k; for $i = n-1$, this produces the equation $\Delta_k \approx ad_{(k+1)n-1}$, which as we've shown in the previous section, indeed behaves like a normal orbit in the mandelbrot set too.

[I'll add more text here later, for now I'm still experimenting how this can best be implemented in the form of an algorithm to skip iterations efficiently. Just wanted to write down these equations here for now, before I forget them again.]

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't simply be expressed as a single complex number.

Instead, we'll have to treat the real and imaginary component as separate real numbers, here referred to as $\mathrm{re}(z)$ and $\mathrm{im}(z)$, and use a matrix of partial derivatives:

$$\frac{\partial z_n}{\partial z_0} = \begin{bmatrix} \frac{d\,\mathrm{re}(z_n)}{d\,\mathrm{re}(z_0)} & \frac{d\,\mathrm{re}(z_n)}{d\,\mathrm{im}(z_0)} \\ \frac{d\,\mathrm{im}(z_n)}{d\,\mathrm{re}(z_0)} & \frac{d\,\mathrm{im}(z_n)}{d\,\mathrm{im}(z_0)} \end{bmatrix}.$$

Similar a normal derivative, this Jacobian matrix tells us how the region of points close to z_0 is transformed relative to z; however, while a normal derivative can only encode the scale and rotation part of this transformation, the Jacobian encodes precisely how much both components of z_n change with respect to both components of z_0.

In the case of the mandelbar set for example, this is needed because its formula involves the complex conjugate $\bar{z}$, which flips the sign of the imaginary but not the real component of z; similarly, the burning ship fractal makes use of the component-wise *abs* function, which too flips regions in the second and fourth quadrant in such a way that a single scale and rotation value is no longer enough to fully encode the resulting transformation. Hence, the Jacobian matrix is needed.

Even worse though, there are now 8 second-order partial derivatives required to accurately describe the change rate of the Jacobian - and 16 third-order partial derivatives, 32 fourth-order ones and so on. While these can all still be iteratively computed, doing so quite quickly becomes infeasible as the number of required operations per iteration grows exponentially. As such, we are mostly limited to working with the first- and second-order partial derivatives for these fractals.

Now, before we start actually doing anything with those, a few other quick definitions: I'll from here on use $M(z)$ to refer to the matrix $\begin{bmatrix} re(z) & -im(z) \\ im(z) & re(z) \end{bmatrix}$; that way, multiplying z with another number a now has the effect of multiplying the Jacobian $\frac{\partial z}{\partial z_0}$ with $M(a)$, similar to how it previously would have had the effect of multiplying the derivative by a. Also, I'll from here on use the concept of complex numbers and vectors interchangeably; for example, the product $\begin{bmatrix} a & b \\ c & d \end{bmatrix} \cdot z$ would then be equal to $a\,\mathrm{re}(z) + b\,\mathrm{im}(z) + (c\,\mathrm{re}(z) + d\,\mathrm{im}(z))i$. I know this is probably a terrible way to write this down, but I sadly couldn't find a better one, so please excuse the mess of equations you're about to see.

2.1 Finding cyclic points

With that being said, let's now start by examining how we can use this to apply our previous equations to the mandelbar set. We'll begin with the first derivative - or now, Jacobian - of

the sequence once again:

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

While similar equations could also be found for the second-order partial derivatives, there's at least as far as I know no simple way of representing those using a matrix; so, we'll just ignore those for now and see what we can do using the Jacobian alone.

Similar to what we did with the derivative before, we can use the Jacobian to get a linear approximation of x_n for points x_0 close to the point the Jacobian was computed for:

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

Based on that, we can again compute the root of this approximation and use that as a new base point; repeating this a number of times, we can again find a $n+1$-cyclic point using Newton's method:

$$x_0' = z_0 - \left(\frac{\partial z_n}{\partial z_0}\right)^{-1} z_n.$$

The derivative $\frac{dx_0'}{z_0}$ can then also be computed based on the second-order partial derivatives, so we can use that as a quick check whether Newton's Method might converge or not; however, this also limits us to the linear approximation, as calculating this derivative for an estimate that's based on a quadratic approximation as we did before would require third-order partial derivatives, which are due to their sheer number expensive to compute.

2.2 Finding Minibrots

As before, we of course also want to know what structure any given cyclic point belongs to - whether it is a disk, ellipse, minibrot or another miniature fractal, and what the scale and orientation of that structure is.

Again, we start using a perfectly n-cyclic starting point z_0 and some other point x_0 around it; the $n-1$th iteration of this point can then again be approximated as $x_{n-1} \approx z_{n-1} + \frac{\partial z_{n-1}}{\partial z_0}(x_0 - z_0)$, which simplifies down to just $x_{n-1} \approx \frac{\partial z_{n-1}}{\partial z_0}(x_0 - z_0)$ since z_{n-1} is exactly zero. Let's call this value d_0 again, as we did before.

The next iteration can then be approximated as $x_n \approx x_0 + \bar{d}_0^2$; the iteration after that as

$$\begin{aligned}
x_{n+1} &\approx (\overline{x_0 + \overline{d_0^2}})^2 + x_0 \\
&\approx \overline{x_0^2} + 2\overline{x}_0 d_0^2 + d_0^4 + x_0 \\
&\approx x_1 + 2\overline{z}_0 d_0^2 + 2(\overline{x}_0 - \overline{z}_0) d_0^2 + d_0^4.
\end{aligned}$$

And again, the terms $2(\overline{x}_0 - \overline{z}_0)d_0^2$ and d_0^4 can be simply discarded as they are basically negligible in comparison to the other terms:

$$\begin{aligned}
x_{n+1} &\approx x_1 + 2\overline{z}_0 d_0^2 \\
x_{n+2} &\approx (\overline{x_1 + 2\overline{z}_0 d_0^2})^2 + x_0 \\
&\approx \overline{x_1^2} + 4z_0\overline{x}_1\overline{d_0^2} + 4z_0\overline{d_0^4} + x_0 \\
&\approx x_2 + 4z_0\overline{z}_1\overline{d_0^2} \\
x_{n+3} &\approx x_3 + 8\overline{z}_0 z_1 \overline{z}_2 d_0^2 \\
&\dots \\
x_{2n-1} &\approx \begin{cases} x_{n-1} + 2^{n-1}\overline{z}_0 z_1 \dots \overline{z}_{n-2} d_0^2 & \text{for } n \equiv 0 \mod 2 \\ x_{n-1} + 2^{n-1} z_0 \overline{z}_1 \dots \overline{z}_{n-2}\overline{d_0^2} & \text{for } n \equiv 1 \mod 2 \end{cases}
\end{aligned}$$

By substituting d_1 for x_{2n-1} and repeating this, we can again find a general equation for the sequence of points d_k approximating $x_{(k+1)n-1}$:

$$d_{k+1} = \begin{cases} 2^{n-1}\overline{z}_0 z_1 \dots \overline{z}_{n-2} d_k^2 + d_0 & \text{for } n \equiv 0 \mod 2 \\ 2^{n-1} z_0 \overline{z}_1 \dots \overline{z}_{n-2}\overline{d_k^2} + d_0 & \text{for } n \equiv 1 \mod 2 \end{cases}$$

Notice how the sequence behaves differently this time depending on whether the cycle length n is even or odd; if it is even, all of the conjugates applied to the previous element of the sequence cancel each other out, if it is odd they don't. This is the reason that the mandelbar set contains both smaller copies of itself and of the mandelbrot set; depending on the cycle length, the points close to the cyclic point undergo different transformations that can't even be made equivalent through a shift of reference (for example by multiplying by a certain constant, as we did earlier with the mandelbrot set); as such, they produce different miniature fractals. This is generally one of the biggest challenges when trying to apply these methods to other fractals; most of them contain not just one or two different kinds of miniature fractals, but a whole spectrum of them plus an often infinite number of hard to generalize edge-cases.

In case of the mandelbar set again, the first case just boils down to the classic mandelbrot set equation:

$$\begin{aligned}
d_{k+1} &= 2^{n-1}\overline{z}_0 z_1 \dots \overline{z}_{n-2} d_k^2 + d_0 \\
2^{n-1}\overline{z}_0 z_1 \dots \overline{z}_{n-2} d_{k+1} &= (2^{n-1}\overline{z}_0 z_1 \dots \overline{z}_{n-2} d_k)^2 + 2^{n-1}\overline{z}_0 z_1 \dots \overline{z}_{n-2} d_0.
\end{aligned}$$

Similar to what we had before, it is the sequence of values $2^{n-1}\overline{z}_0 z_1 \dots \overline{z}_{n-2} \cdot d_k$ that actually iterates based on the same formula as the mandelbrot set, and the value $2^{n-1}\overline{z}_0 z_1 \dots \overline{z}_{n-2} d_0$ that acts as the starting value for; thus, we can again compute the scale and orientation of that minibrot or disk as the ratio between this value and x_0:

$$\begin{aligned}
a &= \frac{\partial x_0}{\partial(2^{n-1}\overline{z}_0 z_1 \dots \overline{z}_{n-2} d_0)} \\
&= \left(\frac{\partial(2^{n-1}\overline{z}_0 z_1 \dots \overline{z}_{n-2} d_0)}{\partial x_0} \right)^{-1} \\
&= \left(M(2^{n-1}\overline{z}_0 z_1 \dots \overline{z}_{n-2}) \frac{\partial x_{n-1}}{\partial x_0} \right)^{-1}.
\end{aligned}$$

One important difference to the original equation though is that this value a is now a 2×2 matrix instead of a real number; this comes from the fact that most (if not all) disks and

minibrots in the mandelbar set are skewed to at least some degree, therefore requiring an entire matrix to fully encode their scale, rotation and stretch factor and direction.

The second case though is already somewhat more complicated. To show that it is equivalent to the mandelbar set equation for some sequence $b \cdot d_k$, we need to again turn in into an equation of the form $bd_k = (\overline{bd_k})^2 + bd_0$; however this time, it's not possible to do that by just multiplying both sides of the equation with the coefficient $c = 2^{n-1} z_0 \overline{z}_1 ... \overline{z}_{n-2}$. Instead, we need to actually solve those two forms of the equation for b:

$$\begin{aligned} d_{k+1} &= c \cdot \overline{d_k^2} + d_0 \\ bd_{k+1} &= bc \cdot \overline{d_k^2} + bd_0 \\ bc &= \overline{b^2} \\ c &= \frac{\overline{b^2}}{b} = \frac{\overline{b^3}}{|b^2|} \\ b &= \overline{\sqrt[3]{c}} \cdot |c|^{\frac{2}{3}} \end{aligned}$$

Of course the cube root is not uniquely defined for complex numbers, but that doesn't matter for our use case here; any one of the three roots of the polynomial $x^3 = c$ will work, since the mandelbar set is rotationally symmetric by exactly 120 degrees.

After choosing one of these possible values for b, we can again compute the matrix a as the Jacobian matrix $\frac{\partial x_0}{\partial (bd_0)}$:

$$\begin{aligned} a &= \frac{\partial x_0}{\partial (bd_0)} \\ &= \left(\frac{\partial (bd_0)}{\partial x_0} \right)^{-1} \\ &= \left(M(b) \frac{\partial x_{n-1}}{\partial x_0} \right)^{-1}. \end{aligned}$$

These two cases are of course unique to the mandelbar set, but the general process should be the same for all fractals; find an equation for the series d_k, find some way to transform it into an already known fractal equation, and then compute a as the Jacobian matrix indicating the ratio between x_0 and the starting value for the transformed equation.