

# 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 (just some rough notes of mine right now - WIP)

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

Since the polynomials generated by repeated iteration don't seem to contain any double zeros in case of the mandelbrot set (though I haven't managed to verify that yet), the values close to these zeros can be modeled as a simple linear gradient; to find exact zeros, you can just iterate a nearby point until it is close to a zero, compute an estimate for the zero using the current value and the direction of the gradient, use this estimation as the next starting point and repeat until you've reached the desired accuracy.

Doing this requires keeping track of the first derivative as you iterate a point:

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

However, computing the second derivative is also required to make sure that the linear approximation  $z_n + \Delta z_0 \cdot \frac{dz_n}{dz_0}(z_0)$  is actually good enough for the orbit point approximation to converge:

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

Higher derivatives can be calculated in a similar fashion:

$$\begin{aligned}
\frac{d^3 z_{n+1}}{dz_0^3} &= \frac{d\left(\frac{d^2 z_{n+1}}{dz_0^2}\right)}{dz_0} \\
&= \frac{d\left(2\left(\frac{dz_n}{dz_0}\right)^2 + 2\frac{d^2 z_n}{dz_0^2}z_n\right)}{dz_0} \\
&= 2 \cdot \frac{d\left(\left(\frac{dz_n}{dz_0}\right)^2\right)}{dz_0} + 2 \cdot \frac{d\left(\frac{d^2 z_n}{dz_0^2}z_n\right)}{dz_0} \\
&= 4 \cdot \frac{d^2 z_n}{dz_0^2} \cdot \frac{dz_n}{dz_0} + 2 \cdot \frac{d^3 z_n}{dz_0^3} z_n + 2 \cdot \frac{d^2 z_n}{dz_0^2} \cdot \frac{dz_n}{dz_0} \\
&= 6 \cdot \frac{d^2 z_n}{dz_0^2} \cdot \frac{dz_n}{dz_0} + 2 \cdot \frac{d^3 z_n}{dz_0^3} z_n
\end{aligned}$$