Fast calculation of the Mandelbrot set with infinite resolution

Sergey Khashin
Department of Mathematics, Ivanovo State University, Ivanovo, Russia
khash2@gmail.com
http://math.ivanovo.ac.ru/dalgebra/Khashin/index.html

October 12, 2016

Abstract

A significant number of people in the world has been devoting a significant amount of time to the implementation of the Mandelbrot set. The main problem is that double precision gets exhausted very fast, while the arbitrary presion realizations are terribly slow.

The proposed idea can dramatically increase the speed of construction of the site of the Mandelbrot set when its size is small($< 10^{-14}$).

Below we demostrate our java implementation which works rather fast and with arbitrary precision.

On a standard home PC java computes a single picture with arbitary high resolution for some seconds. My program computes similar or the same series of pictures to those computed by others scientists about 1000 times faster! Proper benchmarking will appear hopefully soon. This significant accelation is obtained due to purely mathematical improvements of the existing algorithms.

1 Introduction

Consider the sequence of polynomials defined by the recurrence relation: $f_0(z) = 0$, $f_{n+1}(z) = f_n(z)^2 + z$, that is:

$$f_0(z) = 0,\ f_1(z) = z,\ f_2(z) = z^2 + z,\ f_3(z) = z^4 + 2z^3 + z^2 + z,$$
$$f_4(z) = z^8 + 4z^7 + 6z^6 + 6z^5 + 5z^4 + 2z^3 + z^2 + z, \ldots$$

The Mandelbrot set M is the set of comlex c for which the sequence $f_0(c),\ f_1(c),\ \ldots$ remains bounded.

Let $d_n(z, \delta)$ be a polynomial

$$d_n(z, \delta) = f_n(z + \delta) - f_n(z, \delta),$$

So

$$d_{n+1}(z, \delta) = f_{n+1}(z + \delta) - f_{n+1}(z) = f_n(z + \delta)^2 + z + \delta - f_n(z)^2 - z =$$
$$= (f_n(z + \delta) - f_n(z))(f_n(z + \delta) + f_n(z)) + \delta = d_n(z, \delta)(2f_n(z) + d_n(z, \delta)) + \delta.$$

Let

$$h_n(z, \delta) = \frac{d_n(z, \delta)}{\delta} = \frac{f_n(z + \delta) - f_n(z)}{\delta},$$

so

$$h_0(z, \delta) = 0,\ h_1(z, \delta) = 1,\ h_2(z, \delta) = 2z + \delta + 1,$$
$$h_3(z, \delta) = (4z^3 + 6z^2 + 1) + \delta(6z^2 + 6z + 1) + \delta^2(4z + 1) + \delta^3.$$
$$\ldots.$$

There exist recurrent formula for $h_n(z,\delta)$:

$$h_{n+1} = h_n(2f_n + \delta h_n) + 1\,. \tag{1}$$

From the definition, we can write:

$$f_n(z+\delta) = f_n(z) + \delta h_n(z,\delta). \tag{2}$$

Let $z \in M$, so for all n:

$$f_n(z) < 2.$$

Lets calculate with great precision the values

$$f_0(z), f_1(z), \ldots.$$

After that, for each small δ we can calculate the values $h_n(z,\delta)$ by formula (1) using usual double precision and calculate the values $f_n(z+\delta)$ by formula (2) again with double precision.

The implementation of the proposed algorithm in Java can be found in [1]. Do not forget to install the Java-plugin and let he performed (the default is not allowed!).

References

[1] S.I.Khashin, *Fast Mandelbrot set with infinite resolution, ver. 2.* http://math.ivanovo.ac.ru/dalgebra/Khashin/man2/man.html