# Computing the Mandelbrot Set, Reliably*

Jihoon Hyun, Hyunwoo Lee, and Martin Ziegler

{qawbecrdtey,lhw941,ziegler}@kaist.ac.kr,

KAIST, Daejeon, Republic of Korea

**Abstract.** We present and empirically evaluate the (arguably first) C++ program computing the Mandelbrot Set *reliably*, namely using the iRRAM library to implement the abstract proof [doi:10.1002/malq.200310124] of its Turing-computability subject to the Hyperbolicity Conjecture.

Since the 80ies, computers have helped popularize fractals [5,1], enabling a hands-on experience and experimental approach to their otherwise highly abstract mathematics. The Mandelbrot Set for example is commonly "computed" by iterating, for each screen pixel $c$ in the complex plane, the quadratic sequence $M_{c,n+1} = c + M_{c,n}^2$ it induces: $c$ does not belong to the Mandelbrot Set iff $M_{c,n}$ diverges iff it exceeds 2 in magnitude for some $n$.

This approach however constitutes a mere heuristic, for two reasons: First the iteration is usually conducted in floating-point arithmetic with rounding and truncation errors that propagate and make the computed sequence differ unreliably from the mathematical one. Secondly, the number $n$ of iterations after which $M_{c,n}$ is considered to not diverge is usually some fixed 'sufficiently' large integer, such as 100 or 1000, lacking rigorous justification.

We present what seems to be the first program computing the Mandelbrot Set reliably, namely avoiding both the aforementioned deficiencies. The first one is conveniently taken care of using the `iRRAM` C++ library [3] with the *Exact Real Computation* Paradigm [4]. Regarding the second deficiency, we have implemented in C++ the abstract proof from [2] that the Mandelbrot Set is Turing-computable—subject to the Hyperbolicity Conjecture.

Full reliability naturally comes at a penalty in efficiency: We empirically explore the asymptotic dependence of running time up to resolution $50 \times 50$. The source code is available at `http://github.com/realcomputation/MANDELBROT`. Future work will explore possibilities for further optimization.

## References

1. Mark Braverman and Michael Yampolsky. *Computability of Julia Sets*, volume 23 of *Algorithms and computation in mathematics*. Springer, 2009.
2. Peter Hertling. Is the mandelbrot set computable? *Math. Log. Q.*, 51(1):5–18, 2005.

* This work was supported by the National Research Foundation of Korea (grant 2017R1E1A1A03071032) and by the International Research & Development Program of the Korean Ministry of Science and ICT (grant 2016K1A3A7A03950702). We thank Jiman Hwang and Sewon Park for helpful discussions.

3. Norbert Th. Müller. The iRRAM: Exact arithmetic in C++. In Jens Blanck, Vasco Brattka, and Peter Hertling, editors, *Computability and Complexity in Analysis*, volume 2064 of *Lecture Notes in Computer Science*, pages 222–252, Berlin, 2001. Springer. 4th International Workshop, CCA 2000, Swansea, UK, September 2000.
4. Norbert Th. Müller and Martin Ziegler. From calculus to algorithms without errors. In Hoon Hong and Chee Yap, editors, *Mathematical Software - ICMS 2014 - 4th International Congress, Seoul, South Korea, August 5-9, 2014. Proceedings*, volume 8592 of *Lecture Notes in Computer Science*, pages 718–724. Springer, 2014.
5. Heinz-Otto Peitgen and Peter H. Richter. *The beauty of fractals - images of complex dynamical systems.* Springer, 1986.

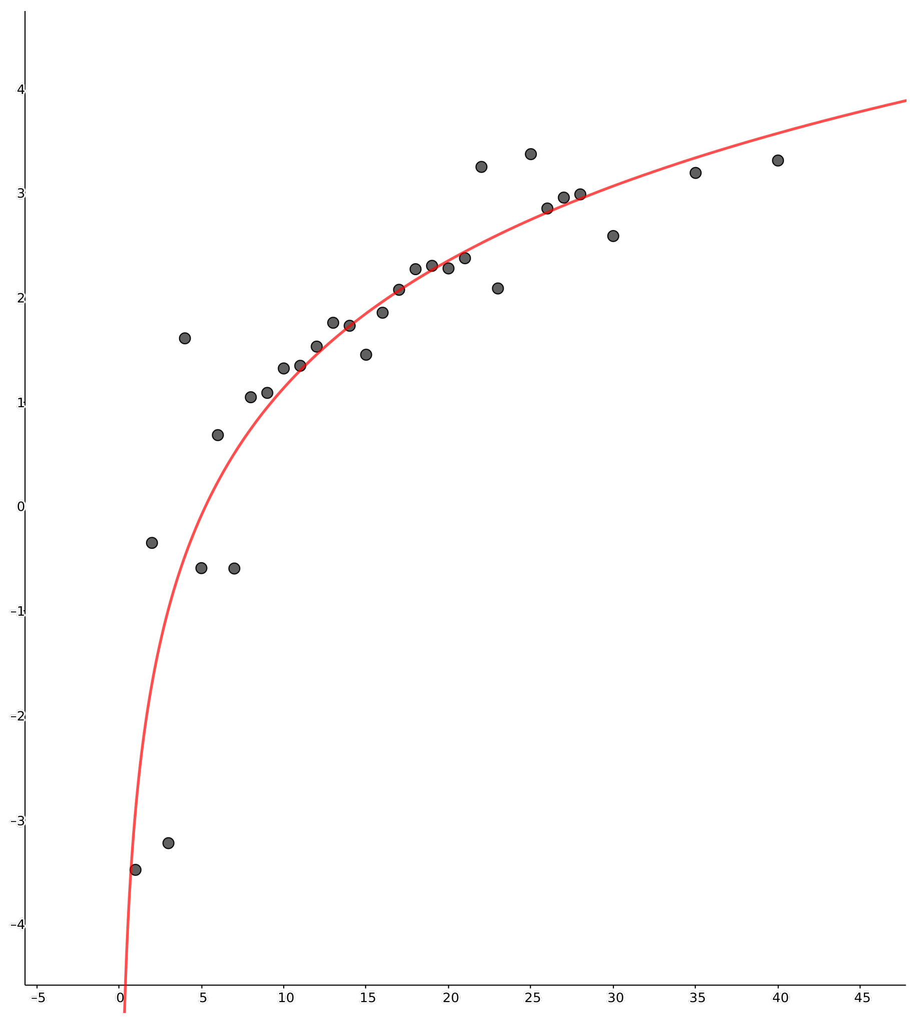


**Fig. 1.** Logarithm of CPU time in dependence on the resolution