Guaranteed Lower Area Bound for the Mandelbrot Set

Marc Meidlinger
marc.meidlinger@web.de

Summary

A method based on interval arithmetics and cell mapping is presented to compute a lower bound on the area of the Mandelbrot set that comes with a mathematical guarantee. The reliable computational steps are guided through point-sampled data towards positions in the complex plane where possible Mandelbrot interior resides. Simple connectivity of the set is exploited to flood-fill the inside of closed reliably analyzed region borders. Due to the highly parallelizable nature of the algorithm and as all computations were performed on a personal computer, tighter bounds are fairly easy to obtain using more equipped hardware.

1 Introduction

Visualization of the Mandelbrot set $z^2 + c$ routinely involves escape-time algorithms [1, 2]. While this method produces beautiful images, those are only approximations [3]. Rounding errors can accumulate and produce false results, points outside but close to the Mandelbrot set's boundary can take arbitrarily long to escape, and computer numbers can only represent rational real or imaginary values.

Addressing those drawbacks, Figueiredo et al. introduced a cell-mapping and interval arithmetics based algorithm to compute trustworthy Julia sets [4, 5]. They tile the dynamical plane into small squares and evaluate the iteration polynomial once for a full tile via interval arithmetics. Subsequently, tiles that intersect with this bounding box are identified and every one is handled the same way. The algorithm aims at identifying tiles that are part of an attracting cycle. This tiling process is repeated for undetermined tiles, termed refinement, into smaller squares. The color of a pixel in the final image then reflects the fate of all complex points within the square - interior to the Mandelbrot set, exterior or currently undetermined.

It is straightforward to use that method to compute a trustworthy Mandelbrot set and arrive at a guaranteed area bound by tiling the complex parameter plane into small squares as well and using those as interval seeds for the Julia set algorithm. Such an implementation is however very slow in nature. A first optimization step is obviously to not compute the entire interval Julia set but just the cell-mapped orbit of the critical point and then determine whether it is bounded or not.

To be able to optimize further and in a specific way, the construction of a trustworthy Mandelbrot set was split into two parts. Here an algorithm for computing a lower bound on the area of the set is presented. It uses point-sampled data to approximate the centers of hyperbolic components and the size of an inscribed square as an estimation of the size of the component itself. This data is then used to guide the reliable computation to regions where it is likely that Mandelbrot set interior can be found via CM/IA in general and at a given refinement level specifically. The trustworthy computation is based on analyzing parameter plane tiles on the border of larger squares and then exploiting simple connectivity of the Mandelbrot set to flood-fill the inside of found closed regions.

Förstemann gave a point-sampled derived numerical estimate of the Mandelbrot set's area of 1.5065 using a Monte-Carlo approach [6], however without rounding control. Recently a tesselation using distance estimation was conducted [7], that uses the well-known approximationof the actual distance function, for which no error-bound is given. A similar value of 1.52 was obtained by Ewing for the set's area, derived also by pixel-counting [3]. Another approach by approximating the bulbs attached at the main cardioid as circles was taken by Fowler et al. [8]. Fisher et. al gave a lower value of 1.50311 based on escape-time period checking with finite precision [11]. Round-off errors were not controlled, and period-detection could not be verified to not return a multiple of the actual period.

An upper bound on the set's area was estimated to 1.68288 [9]. It is based on computing a finite number of coefficients of a slowly converging series. Errors were probabilistically estimated to be in the range of machine-epsilon but not rigourously proven to arrive at a definite bound value [9, 10].

The presented algorithm here overcomes the escape-time problems. By using cell-mapping, all values in complex intervals are analyzed, and not only the ones with rational coordinates. Neither arbitrary precision nor rounding control are necessary, as calculations only work on dyadic fractions, so all intermediate and final values can be accurately represented with fast floating-types double and float128. The obtained area is therefore a bound in the mathematical sense.

Notation

Mtile, complex interval with dyadic end points in parameter space's 2-square [-2..2]+[-2..2]·i; Jtile, defined accordingly in the dynamical plane's 2-square; refinements are given as $\mathrm{M}_m\mathrm{J}_j$, where the M-plane has been partioned into $2^m \times 2^m$ identically sized Mtiles, and the dynamical plane into $2^j \times 2^j$ Jtiles; Mset, quadratic Mandelbrot set. Jset, Julia set; IA, interval arithmetics. CM, cell mapping.

2 Algorithmic Design in Detail

2.1 Julia Set Analysis

At the start, a queue contains those Jtiles that harbour the critical point. The dynamical plane itself is marked as unvisited. As both the Mset and the Jset are tiled with grid lines parallel to the axis and axis themselves are grid lines, one starts with four Jtiles that harbour the origin at one of their corners.

A Jtile is removed from the queue, its bounding box under the iteration function $z^2 + c$ with fixed c being the complex interval to be analyzed, is computed via interval arithmetics and the Jtiles the bounding box intersects with are identified. Jtiles that have not been visited before are added to the queue. This process iteratively constructs the orbit of the critical point's starting Jtiles until one of two termination criteria is met:

a) If a bounding box intersects with the outside of the 2-square, the critical orbit is not bounded in the current Mset and Jset resolutions, and the analyzed Mtile will remain undetermined.

b) If no more unvisited Jtiles are reachable and the orbit has never left the 2-square, it is bounded and the Mtile can be judged as interior to the Mandelbrot set.

2.2 Mandelbrot Set Analysis

The algorithm's principal means is to analyze via CM/IA only a fraction of the currently undetermined Mtiles of the parameter plane at a specific refinement level and exploiting the simple connectivity of the Mandelbrot set to flood-fill when a closed interior Mtile curve was established.

From the author's experience, the closer a Julia set seed gets to the Mandelbrot set's boundary the higher the refinement level is, where the set, computed with the Figueiredo algorithm will show interior and can be judged as connected. The easiest seeds in relative terms are the ones well inside a hyperbolic component. Therefore it was natural to start analyzing Mtiles in the vicinity of hyperbolic centers to detect new interior regions in the Mtile plane. Smaller hyperbolic components need in general higher levels as well.

The main cardioid and period-2 bulb are not analyzed, as their full region is well-known by analytical means [12, 13].

The algorithms works in several sequential steps:

Step 0. Hyperbolic Center List

This step is performed without rounding control, as its output is merely used as a directional tool and not as an answer to what complex values lie in the Mandelbrot set. Any other list of approximated center coordinates and size estimates could as well be used here.

A point-sampled Mandelbrot set is computed via standard escape-time approach. Non-overlapping squares are inscribed into the bulbs and cardioids. A square's geometric middle point serves as an approximation to the hyperbolic center, the side length of the square as an approximation to the size of the component. This step outputs a plain list of those coordinate and size values and is used in the subsequent reliable computation steps as a guide to where the cell-mapping analysis should take place.

Bad approximations or a component harbouring more than one inscribed square affect the reliable steps only in terms of increased computional effort, e.g. looking for interior in the wrong places, but they do not compromise the mathematical validity of the resulting lower bound.

Step 1. Activating Hyperbolic Centers

The algorithm goes through the entire list of hyperbolic center coordinates once at the start of a refinement. The Mtile which harbours a center's coordinates is calculated and its interval Julia set is analyzed to see whether the critical point's orbit is bounded. If the answer is yes at the current $M_m J_j$ refinement, the center is activated in this and higher refinement levels for further extension of the interior region.

Step 2. Analyzing Active Regions

All Mtiles that lie inside or overlap with the main cardioid or the period-2 bulb are marked as not-to-be-analyzed. This test is performed in a reliable manner with interval arithmetics and a number type providing sufficient precision.

A square region around the center's complex coordinates of side length 3 times the estimated component size from the initial list (empirical value) is defined as the active region in which promising Mtiles are analyzed.

If the active region lies entirely in the upper half of the Mandelbrot set, the center can be discarded due to x-axis symmetry. If the region straddles the x-axis it is kept and analyzed. The active region then deals with symmetry to reduce the number of Mtiles to analyze.

Then a point grid of size d Mtiles is imposed over the active region, the center pixel being a grid point. Mtiles at the grid points are analyzed via Jset critical orbit contruction in a specific order. One starts with the grid point on the horizontal line through the region's center at relative Mtile coordinates $(d, 0)$. If this can be judged as M-interior or has already been previously, the routine proceeds upwards to (d, d), $(d, 2d)$ until the first Mtile could no longer be determined to lie in the Mandelbrot set's interior at the current refinement. Then the algorithm proceeds to $(2d, 0)$ and the scanning is repeated. Similarly this is also done downwards, and for the left side of the hyperbolic center. Subsequently the whole process is started again from the vertical line through the center and going left or right. The aim of this approach is to roughly follow the shape of the hyperbolic component part that can be detected in the current $M_m J_j$.

After no more change occurs, if two interior judged Mtiles on grid points lie directly adjacent to one another in a horizontal line, their connecting Mtile line is fully analyzed by orbit construction, as is done for vertically adjacent ones. This aims at finding closed squares, whose border Mtiles are all interior to the Mandelbrot set. Then a flood-fill is conducted to fill those closed regions.

The gridding process is performed in decreasing grid sizes of $d = 64, 32, 16$ and down to 8 (empirical value).

This algorithmic step is visualized for the south period-3 bulb (fig. 1). After the center's Mtile was judged as interior in $M_{13}J_{11}$, gridding quickly resulted in quite a large portion of the bulb to be flood-fillable.

Pre-gridding test. By algorithmic design, gridding would always analyze both the entire horizontal and the whole vertical line going through the active region's center in the smallest grid size. However, far from the center, these will most probably not be interior Mtiles that belong to the component of the region's center and are assumably part of an overlapping component that is being analyzed in its own active region.

Therefore a pre-test is routinely performed to roughly estimate the size of the detectable component in the current M_mJ_j levels. Starting from the center Mtile repeated jumps to the right in steps of 64 Mtiles are performed. The Mtile at landing position is orbit-analyzed. The jumps stop when the first Mtile cannot be judged interior. The same is done to the left and vertically up and down. The circumferencing rectangle at those stopping Mtiles is enlarged by 32 Mtiles in all directions and defines the space in which the gridding takes place. Enlarging is performed to allow for the region to grow in case a 64-jump happens to land just adjacent to a current outermost interior Mtile. This stopping process can lead to a considerable reduction of the Mtiles per active center to be analyzed and result in a speed-up.

Step 3. Lower Area Bound

Because of their square shape, active regions are not necessarily disjoint to one another and can harbour multiple hyperbolic components. Therefore it is necessary to account for multiplicity of interior Mtiles.

Main cardioid and period-2 bulb are excluded here, as their area is analytically known to be $7\pi/16$ [13] and externally added using WolframAlpha.

2.3 Hard- and Software

All computations were conducted on an AMD Ryzen 7 2700X 3.7 GHz processor with 48 GB DDR4 RAM and up to 8 parallel Mtiles analyzed at once using OpenMP on Windows 10 64-bit. Over night the software was the sole process, during the day it was run in the background. gcc 5.1.0 tdm64 was used as C++ compiler.

For orbit construction, 1 byte per Jtile is required, in total $1 \cdot 2^j \cdot 2^j \cdot$ *parallelicity* bytes. All Julia sets are kept in memory simultaneously.

Active regions also need 1 byte per Mtile, however it is not predictable which ones are active at a given refinement level and how large they are. Currently, out of convenience, all active regions are kept simultaneously in memory as well.

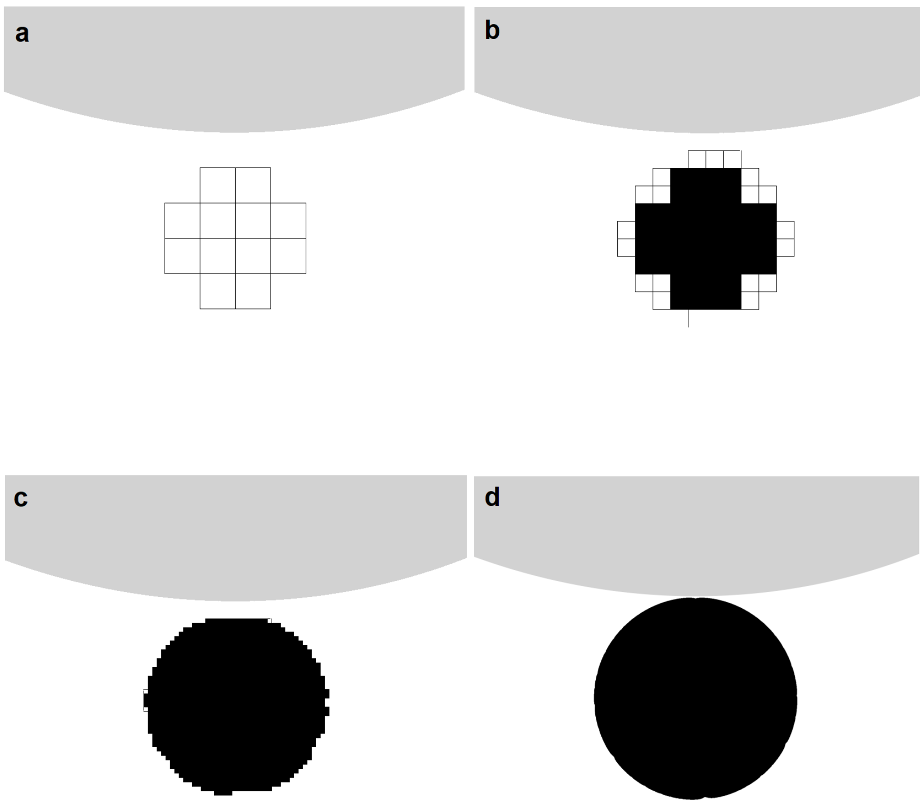

Figure 1: The south period-3 bulb in algorithmic steps. Part of the main cardioid is showing at the top. Those pale gray Mtiles were marked beforehand and not orbit-analyzed.

a, the bulb emerged at $M_{13}J_{11}$. The approximated hyperbolic center's Mtile (here the central black pixel) was orbit analyzed and judged interior. A grid of size 64 Mtiles was subsequenty orbit analyzed and resulted in closed squares.
b, the interior of the closed squares was flood-filled exploiting simple connectivity. A 2nd grid of 32 Mtiles was analyzed and resulted in many closed squares being flood-fillable.
c, after gridding and flood-filling the next grid values of 16 and 8, the final result of the bulb at the emerging $M_{13}J_{11}$ was reached. At the left side, two 8-grid squares remain open and are not fillable.
d, at the end of $M_{17}J_{15}$, the 3-bulb has grown close to the main cardioid and shows an indentation at its bottom for a smaller bulb to attach. The outermost shell is still composed of 9x9-Mtile squares but appears smooth only due to image downscale.

Table 1: Lower bounds on the area of the Mandelbrot set at various refinement levels

Level	Area[1]	Interior Mtiles[2]	Time[3]	Guide quality[4]	Fill advantage[5]	Active centers
$M_{13}J_{11}$	1.4305	235 450		90%	9.1	12
$M_{14}J_{12}$	1.4506	1 279 056		92%	5.9	25
$M_{15}J_{13}$	1.4668	6 201 516	11 min	93%	6.3	51
$M_{16}J_{14}$	1.4789	28 058 290	2 hrs	94%	6.9	86
$M_{17}J_{15}$	1.4875	121 432 004	15 hrs	95%	7.5	134
$M_{18}J_{16}$	1.4932	510 218 134	6.7 days	95%	8.2	182

[1]Sum of the area of the identified interior Mtiles[2] plus the analytically known area of the main cardioid and the period-2 bulb using WolframAlpha and truncating results after 4 decimals.
[2]Interior Mtiles were counted after computation terminated for the indicated refinement level, excluding main cardioid and period-2 bulb and factoring in multiplicity of detection and symmetry as would be seen in a final image of the Mandelbrot set.
[3]Cumulative duration for this and previous refinement levels.
[4]Ratio between the number of Mtiles judged as interior by orbit construction in this refinement and the number of Mtiles for which orbit construction was initiated. It serves as an indicator to how well the point-sampled data guided the IA/CM algorithm to promising regions in the complex plane.
[5]Ratio of the number of Mtiles judged interior by flood-fill in this refinement and the number of Mtiles judged as interior by orbit construction. It indicates how well gridding performed in reducing the number of orbit construction calls by exploiting simple connectivity of the Mandelbrot set.

3 Results and discussion

3.1 Area Bound

For all software runs, independent of the actual M_mJ_j refinements, the same list of 1 112 approximative hyperbolic centers lying in the whole complex 2-square was used, derived from a point-sampled Mandelbrot set at level M_{15}. Only a small fraction of those were actually detectable via IA/CM at our current highest refinement (table 1).

The best lower area bound achieved so far, is

$$510\ 218\ 134 \cdot \left(\frac{4}{2^{18}}\right)^2 + \frac{7\pi}{16} \geq 1.4932$$

obtained from a 6.7 days computation of $M_{18}J_{16}$. As a control, interval Julia sets from the component's center Mtiles were computed in their entirety in a reliable manner, showing cycles of lengths 3 to 27 (data not shown).

Guide quality at level M_mJ_j was calculated by dividing the number of Mtiles that were judged interior in tnat level combination by the number of all Mtiles in this level, that were orbit analyzed. The high value of 90% throughout all levels indicated that using point-sampled data was a valid means to mostly analyze regions where interior of the Mandelbrot set resides snd can be found at a given refinement. By algorithmic design of the gridding process, the expansion in any one direction necessarily ends with a non-interior judged Mtile.

Flood-filling provided a substantial advantage over orbit construction. About 7 Mtiles could on median be flood-filled per Mtile judged interior by CM/IA analysis. It was not investigated, which of the grid sizes provided the highest advantage. Therefore it seems possible to increase this value further by selecting different grid sizes or size decreasing steps other than simply dividing in half. As aging centers, that have been found some refinements ago, necessarily approach their true boundary, it can be argued that large grid sizes are not likely to result in closed curves and might be skipped altogether.

Area gain by M-refinement increase is slowing down as is to be expected. The software finds smaller and smaller new active regions while the Mtile outer shell for older ones approaches the actual boundary.

3.2 Current Limitations

Non-disjointness

As hyperbolic components of the Mandelbrot set are not square in shape, active regions might overlap with nearby active regions harbouring a different hyperbolic center. In that case, it can happen that an active region actually contains two growing interior shapes (fig. S1a), which would result in analyzing the same Mtile multiple times. This is currently not specifically handled, as absolute running times were deemed acceptable. An afterwards analysis showed, that this happened in less than a tenth of the active regions at level M_{18}. A second shape might not always be detected as for this to happen, the gridding process must find a grid point within that sub-region that will be orbit-analyzed. If it jumps over a very small region, this will not emerge.

Minimal Grid Size

Currently the minimal grid size imposed on an active region is 8. However, if the Mtiles to be analyzed approach the true boundary of the hyperbolic component, there might just not be enough space for a complete 9x9 grid to entirely lie within the interior. Decreasing the minimal grid size can remedy this, however at the expense of increased running time and lower fill advantage. Therefore lower grid sizes were not used on a regular basis and are reserved for the upcoming highest M-refinement level possible due to memory availability to then squeeze out as much interior Mtiles as possible. This room problem is naturally taken care of by increasing the M-refinement level, as then the interior Mtile outer shell retreats from the true boundary as Mtiles get smaller in width and allows for more Mtiles to be analyzed.

Pre-gridding Test

The test implicitly assumes that the largest extent of the true interior is adopted close to the horizontal and vertical lines through the center. While this is the case for roughly circular shapes of the bulbs if center approximations are exact enough, it can lead to an unwanted cut-off in other shapes.

If the horizontal line happens to pass through the cusp of a cardioid and the jump happens to land just outside the cusp between the lobes of the cardioid, those will be cut off (fig. S1a). Switching the pre-test off allows gridding to detect the lost parts (fig. S1b).

However, as this incident was only encountered once for a minibrot around $-1.758 + 0i$, the pre-test is routinely performed.

Choice of Parameters

The choice of parameters is based on preliminary, non-exhaustive analysis. Refinement levels used for the Mset and the Julia set are not deeply intertwined. If they however differ substantially, one will dominate the width of an IA analysis. The choice of a relative difference is nevertheless somewhat arbitrary. Other parameter values may exist that lead to either a higher or a faster reached lower area bound. A detailed analysis of the influence of each parameter was beyond the scope of this work.

The point-sampling step was done at refinement level M_{15} once before the reliable computation was started and used throughout all steps. Less than a third of the found approximations to hyperbolic centers in the lower half of the complex plane were activated so far.

The analysis of active centers is done in decreasing grid sizes of 64, 32, 16 and finally 8 Mtiles. This choice was mostly based on the goal, that a substantial part of the interior of newly found active centers should be detected by flood-fill rather than CM/IA-based orbit construction.

$M_m J_{m-2}$ was then deemed a suitable heuristic choice as it quickly allowed for checking more hyperbolic centers, therefore increasing the number of found interior regions while not spending too much time on getting them as big as possible in a specific M-refinement.

To obtain the largest possible lower area bound, especially when approaching the memory limit, it is recommended to perform an additional analysis of the data using a higher Julia set level, e.g. $M_m J_{m-1}$. Furthermore, the minimal grid size should be lowered to detect as much interior Mtiles a possible, snd pre-testing should be switched off.

4 Error-free Computation

Parameter and dynamical plane, both being the 2-square, are tiled into small squares of identical size with gridding lines parallel to the axis, and axis being grid lines themselves. Neither a Jtile nor an Mtile straddle any axis. Real and imaginary part of the coordinates of any corner point of a tile are dyadic fractions of the form n·4/$2^{refinement}$. As calculations only use addition, subtraction and multiplication, but no division, and any constant used is a dyadic fraction as well, all intermediate and final results are dyadic fractions and accurately representable by a suitable floating point number type. The use of a specialized IA library with rounding control is not necessary if, conservatively analyzed, the used number type provides enough mantissa bits.

In summary, for Mandelbrot refinements $m \leq 25$ and Julia set refinement levels $j \leq 25$, the presented code provides sufficient precision using C++ floating number types double and __float128 at the appropriate locations to allow for error-free calculations.

4.1 Precision Requirements for the Mandelbrot Set

Mtile's Corner Coordinates

Up to refinement level $m \leq 26$, C++ double provides sufficient precision to store coordinates.

The complex 2-square containing the Mset is partitioned into $2^m \times 2^m$ square-shaped Mtiles for refinement level m. Corner coordinates have an integer part of -2 to +2, needing at most 2 bits plus 1 sign. The fractional part is an integer multiple of the tile width of $\frac{4}{2^m} = \frac{1}{2^{m-2}}$ and needs at most $m-2$ fractional bits. This demands to $m+1$ bits in total, storable in C++ double number type with 52+1 mantissa+sign bits up to level $m \leq 50$.

Conversion of a coordinate real value w into a virtual Mtile pixel coordinate to access memory is done by

$$floor((w-(-2))/tilewidth) = floor((w+2) \cdot 2^{m-2})$$

This operation needs the bits of a coordinate plus 1 additional bit (as the integer part is maximally 4 instead of 2), in sum $m+2$ bits. The factor 2^{m-2} needs $m-2$ integer bits. As a fixed-point number, in total $2m$ bits are required, covered by C++ double up to $m \leq 26$.

Using the floor function, an Mtile or Jtile is identified with its lower left corner coordinate.

Test for Main Cardioid

For refinement levels $m \leq 25$, C++ __float128 (quad double) provides enough bits to store any (intermediate) result. A complex number (x,y) is in the main cardioid if the following holds [12]:

$$256x^4 + 512x^2 \cdot y^2 - 96x^2 + 32x + 256y^4 - 96y^2 - 3 < 0$$

For an Mtile as a complex interval to lie completely in the main cardioid, a straightforward interval extension is used leading to three outcomes, depending on whether the real-valued resulting interval for the left side of the inequality straddles or touches zero or lies completely on any one side.

Bit requirement analysis conservatively treats every term as an absolute value and every subtraction as addition, accounting for possible intermediate results being larger than the final value.

$$|256x^4| + |512x^2 \cdot y^2| + |96x^2| + |32x| + |256y^4| + |96y^2| + |3|$$

The maximum integer value with x, y in the 2-square is (indepedendent of refinement level) 17 219, needing 15 bits. As x and y as real numbers are both within the same range and need $m - 2$ fractional bits, the highest number of fractional bits for the absolute sum is required for the largest summed exponent being 4, so $4 \cdot m - 8$ fractional bits. At $m \leq 25$ that demands 15 integer bits, 92 fractional bits plus 1 sign, in total 108 bits, covered by __float128 with 112+1 mantissa+sign bits.

Test for Period-2

For refinement levels $m \leq 25$, C++ double number type provides sufficient precision. An Mtile lies within the period-2 circle if all four corner points are within.

A point (x,y) lies inside if the following inequality holds:

$$x^2 + 2x + y^2 + \frac{15}{16} < 0$$

Bit analysis is done on the absolute form: $|x^2| + |2x| + |y^2| + |\frac{15}{16}|$

The integer part is at most 13 independent of the M-level, needing 4 integer bits plus 1 sign, and the highest summed exponent is 2, demanding $2 \cdot m - 4$ fractional bits, the constant 15/16 needs 4 fractional bits.

For $m \leq 25$, this amounts to at most 4 integer, 46 fractional and 1 sign bit, in total 51, covered by C++ double.

4.2 Number Type Requirement for the Julia Set

In summary, for refinement levels $j \leq 25$, C++ double provides enough prcci-sion.

Coordinate representation and complex number to pixel conversion is performed as in the Mandelbrot case. As the refinement level for Julia sets in our setting is always smaller than M-refinement, the number type double is also feasible here.

Orbit Construction

For refinement levels $j \leq 25$, C++ double provides enough precision.

As described in [4], only one iteration is done per Jtile to obtain the bounding box via evaluating an interval extension of the component functions of the iteration polynomial, then identifying the Jtiles that box intersects. The subsequent step is carried out again on a single Jtile and its corner coordinates.

The component functions of $z^2 + c = (x + i \cdot y)^2 + (d + e \cdot i)$ with d, e being Mandelbrot set and x, y being Julia set coordinates.

$$x^2 - y^2 + d$$

$$2xy + e$$

are also analyzed in their absolute form.

Analyzing $|x^2| + |y^2] + |d|$: The largest integer value is 10 for d,e,x,y in the 2-interval, needing 4 bits, the highest summed exponent in x,y being 2, demanding $2 \cdot j - 4$ fractional bits, plus 1 sign, d needs $m - 2$ fractional bits. The resulting integer part needs 5 bits, the fractional part $\max\{2 \cdot j - 4, m - 2\}$ bits. For $m \leq 25$ and $j \leq 25$ this is handled with C++ double.

As in $|2xy| + |e|$ the integer part is also at most 10 and the summed exponent being 2, it needs at most the number of bits the previous component function demanded.

5 Future Work

Currently, point-sampled data is only used in guiding the CM/IA orbit construction to promising Mtiles. It is planned to use such data for Jtile orbit construction itself. The initial Jtile in the queue would then harbour a periodic point rather than the critical point. This will decrease the number of Jtiles to be analyzed in CM/IA orbit construction. In addition, point-sampled data will be used as a guide to compute a reliable exterior of the Mandelbrot set and arrive at a guaranteed upper area bound.

Extending the current code to obtain lower bounds on the area of Multibrot sets, requires only slight modifications, mainly removing the main cardioid and period-2 bulb special treatment and using the correct symmetry to reduce computational effort.

Acknowledgement

I wish to thank the people at fractalforums.org for valuable discussions over the last year.

References

[1] A. Douady, J. Hubbard, On the dynamics of polynomial-like mappings, *Annales scientifiques de l'E.N.S. 4e serie 2* (1985) 287–343.

[2] N. Lesmoir-Gordon (editor), The colours of infinity. The beauty and power of fractals. Springer, `doi:10.1007/978-1-84996-486-9`.

[3] J. Ewing, Can we see the mandelbrot set?, *College Math. J.* 1190, 26(2) 90-99.

[4] L.H. Figueiredo D. Nehab, J. Stolfi and J. Oliveira, Images of julia sets that you can trust. 2013.

[5] L.H. Figueiredo, D. Nehab, J. Stolfi and J. Oliveira, Rigorous bounds for polynomial julia sets, *Journal of Computational Dynamics* 3 (2016) 113. `doi:10.3934/jcd.2016006`.

[6] T. Förstemann, Numerical estimation of the area of the mandelbrot set. 2012

[7] T. Förstemann, Quad-tree tesselation and Koebe theorem, 2017, http://www.foerstemann.name/labor/area/Mset_area_TE_2017.pdf

[8] A. Fowler, M. McGuinness, The size of mandelbrot bulbs, *Chaos, Solitons & Fractals: X* 3 (2019) 100019. `doi:10.1016/j.csfx.2019.100019`.

[9] D. Bittner, W. Cheong, D. Gates and H. Nguyen, New approximations for the area of the mandelbrot set, *Involve, a Journal of Mathematics* 10(4) (2017) 555–572. `doi:10.2140/involve.2017.10.555`.

[10] J. Ewing, G. Schober, The area of the mandelbrot set, *Numerische Mathematik* 61 (1992) 59–72. `doi:10.1007/BF01385497`.

[11] Y. Fisher and J. Hill, Bounding the Area of the Mandelbrot Set, *submitted to Numerische Matehmatik*, 1997

[12] https://en.wikibooks.org/wiki/Fractals/Iterations_in_the_complex_plane/Mandelbrot_set/mandelbrot#Cardioid_and_period2_checking.

[13] http://www.iquilezles.org/www/articles/mset_area/msetarea.htm.

Supplementary Data

Software

- C++ source code
- list of approximated hyperbolic centers
- compressed raw data after $M_{18}J_{16}$ and initial data for M_{19}
- compressed 2-color bitmap of the 2-square containing all identified active regions
- README for command-line parameters of the source code
- batch file used to calculate the current area bound

Active Regions with Multiple Growing Shapes

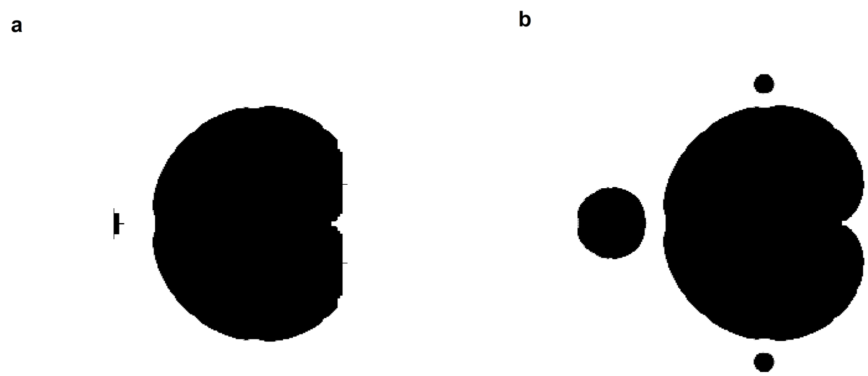

Figure S1: An active region harbouring multiple Mandelbrot set hyperbolic components.

a, the active center at $-1.758 + 0 \cdot i$ lies in the larger cardioid. The gridding process detected a 2nd, smaller M-interior region around $-1.773 + 0 \cdot i$, which itself is a center in a different active region. Using the 64-jump pre-test (see Algorithmic design), the smaller region will not grow to its full size here. The lobes above and below the cusp of the larger cardioid are cut-off due to the pre-test.
b, switching off pre-testing allows for the detection of the lobes with the described drawback having to analyze Mtiles far away from the center. The smaller component is growing as well, demanding analysis of identical Mtiles in multiple active regions. The same is true for the lower small bulb near the cardioid. Due to symmetry, the upper component is simply mirrored and not calculated.