

DIFFERENTIAL EQUATIONS
AND

CONTROL PROCESSES

N 3, 2004

Electronic Journal,

reg. N P23275 at 07.03.97

<http://www.neva.ru/journal>

<http://www.imop.csa.ru/diff>

e-mail: diff@osipenko.stu.neva.ru

Computer modeling in dynamical and control systems

Inwards to Chaos and Complex Mapper:
two graphical interfaces for simulations in complex analysis and dynamic

Alessandro Rosa

Freelance programmer and historian of mathematics

email: zandor_zz@yahoo.it

Abstract

This article presents the main features of two programs, *Inwards to Chaos* and *Complex Mapper*, coded by the author and aiming to support the investigations in complex analysis and dynamics in one complex variable via some techniques differently based upon colors, upon Geometry and Topology. Both programs are built on visual interfaces so that all integrated features can be reached through an easy and intuitive use of windows graphics. All has been customized for a user-friendly focus on the behavior of a given complex system. Their use wants to be flexible and both can be offered as valid tools for researchers and also for teaching courses in complex analysis and dynamics, due to the chance of displaying any user-defined map, because a complete parser handling arithmetic, trigonometry and hyperbolic functions is implemented. Moreover, the user can also count on the support of an on-line and step-by-step help. *Inwards to Chaos* mainly points to global dynamics, besides developing the chromatic aspects of fractal images in the complex plane, where colors play an enlightening role for the dynamics involved, beside the search for the aesthetic taste. *Complex Mapper* is more devoted to the investigations on local systems and it focuses on the geometry of the dynamics of maps. This article is simply descriptive and all information about the internals, the algorithms and about the ways they have been coded can be obtained by writing to author's e-mail below.

1 Introduction

My interests in complex analysis and dynamics, together with my experience as programmer, led me to produce two programs as a self-help to understand

many topics of such fields. During their coding, I developed techniques to display complex (possibly iterated) maps in one complex variable and to take this all to quaternions just for fun. In order of time, I first coded *Inwards to Chaos (IC)*, mostly devoted to complex and quaternion dynamics; some months later, another program - *Complex Mapper (CM)* - came up and its development recently got to the latest release 2.00¹. Since the first lines of code for *IC*, I found essential to count on a parser to input any user-defined formula, either for complex numbers and for quaternions: it is author's opinion that any robust computer program shall be based upon an high degree of freedom, with special regard to public domain software devoted to Mathematics. Then I developed methods to focus on aspects involving graphics, computation and also to play with fractal images which, in several occasions, look very pretty.

2 Inwards To Chaos

The first rows of code were written in spring 2000². The development of *IC* went side by side to my learning of complex dynamics so that first the early coding tested my notions which, conversely, I used after to refine by computing several different examples, thanks to the mentioned freedom of inputing. In that same period I saw my interests in complex dynamics rapidly increasing and something was urgently required to produce figures, after I also embarked upon the writing of a book³.

As the program opens a window welcoming (*See* Figure 1) the user to five possible sections is shown: 1) complex; 2) complex in 3D; 3) quaternions; 4) domain coloring; 5) reals.

2.1 COMPLEX

The goal of this section is to explain how *IC* investigates on the iterations of functions in one complex variable by playing with a number of parameters. It opens with the dialog window in Figure 3, very simple in structure, according to a deliberate feature the author chose for *Inwards*: no much stuff at a glance, so that any process can be easily understood and managed. The dialog shows

¹As this article is in print, new features are being developed and a recent version will be released on August.

²*IC* is included inside a package, also including another program, entitled '*IFS to Chaos*' and devoted to *Iterated Function Systems*, which will not be treated in this article.

³See [1]. Currently on final revision, it concerns the history of complex dynamics via the translations in English of the original works in French and German and a long historical investigation.



Figure 1: The starting window.



Figure 2: The palette applied to figures 3,5 and 6.

the Mandelbrot set in the large view and a Julia set⁴ in the little one on the top left corner; in addition region coordinates are displayed at the four sides.

As anticipation of section 2.2, a 3D view of the same set is displayed on the figure 3 on the right: given an upper bound of iterates for preventing the process to loop⁵, the height of the solid Mandelbrot set is computed according to the number of iterates which are trapped inside a 'critical' disk of given radius⁶.

At the bottom, two rectangles show the complex xy coordinates ($z = x + iy$) of the pointer while floating over the large window. On the top right corner, a green button displays the tips related to this window, while the red one minimizes the dimensions so that some screen space is saved when multiple dialog windows are opened all together. Two menu entries on the top, below the caption, collect a number of basic operations for drawing, storing and resizing the image. At a first glance, all might look like pour to manage robustly all processes and calculations here involved. Hence it is worth introducing the window in Figure 4 on the left.

The fundamental parameters and tools to play with iterates are offered by a management panel (*see* figure 4) : region coordinates, methods to trap the iterated orbits, a number of iterations methods for solving equations⁷, prede-

⁴When 'Mandelbrot' mode is set, the 'J' key shows the Julia set related to pointer position.

⁵If, for example, it should depend on analytic properties exclusively and it might not be satisfied in the investigated region.

⁶This method is commonly known as 'escape time' or 'Level Set'. *See* [2], page 190.

⁷Historical note: complex dynamics first stepped to understand the behavior of such methods all over the

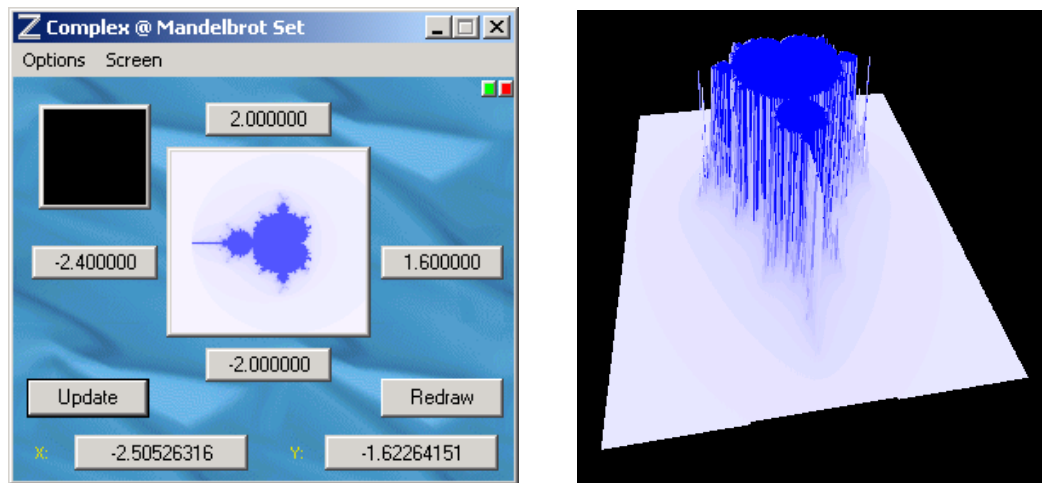


Figure 3: the window of the 'complex' section on the left and a perspective 3D view of the Mandelbrot set on the right.

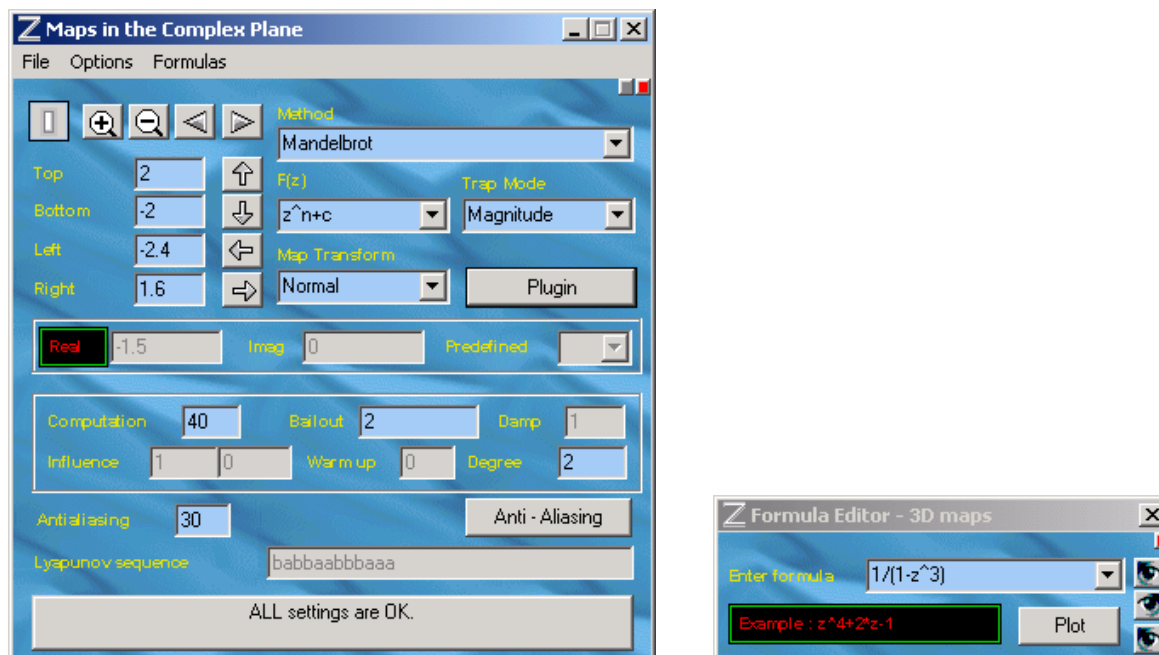


Figure 4: On the left the panel reporting the parameter values for the figures 3 and, on the right, the editor to input user-defined formulas (here showing the map displayed in figures 5).

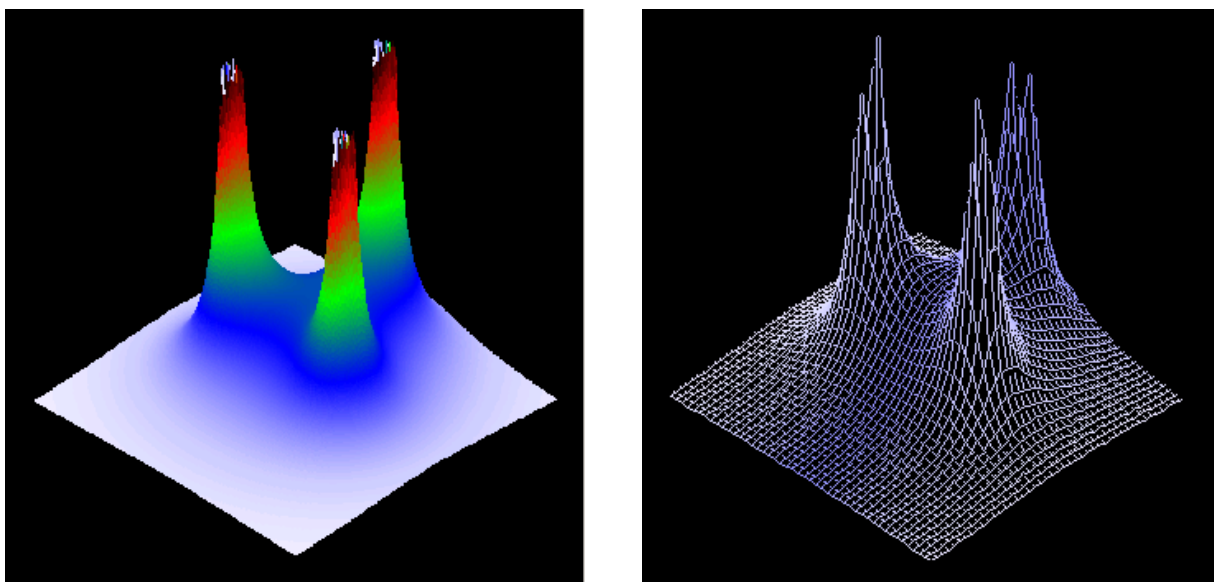


Figure 5: A 3D shaded coloring on the left and a wireframe version on the right.

finer formulas for any beginner and mostly an input editor on the right). This is one of the strongest features of *Inwards*: the parser includes direct and inverse trigonometric, hyperbolic and transcendental functions mainly⁸ and computes symbolically the first derivative or of higher orders by recursion. Some implemented iterative methods are: Julia (simply iteration), Mandelbrot (iteration in one-parameter-space) and, among those ones for solving equations, Newton-Raphson either for multiple and non-multiple roots, Koenigs, Schröder, Steffensen, Halley, Collatz, secant and more.

2.2 COMPLEX IN 3-D⁹

Basically it deals with 3D colored histograms whose upper vertexes for each column are linked (Figures 5, left and right) or not (Figure 6). The height is computed according to a list of rules. This method turns out to be very enlightening for understanding the map behavior about a singularity, such as poles in Figures 5 where $f(z) = 1/(z^3 - 1)$ is displayed with shaded colors (on the left) and as a wireframe (on the right). Notice that the asymptotic values on the unit circle are the three 'peaks'. Figure 6 displays the 3D model of the transformed map arising from Newton-Raphson method applied to the cubic equation $z^3 - 1 = 0$. Basically this is like an histogram: the height per each seed point is set by the minimum value of the iteration index, required to

Riemann sphere.

⁸... and more other options. See help file of *Inwards*.

⁹This option shall be managed together with the dialog in the COMPLEX section.

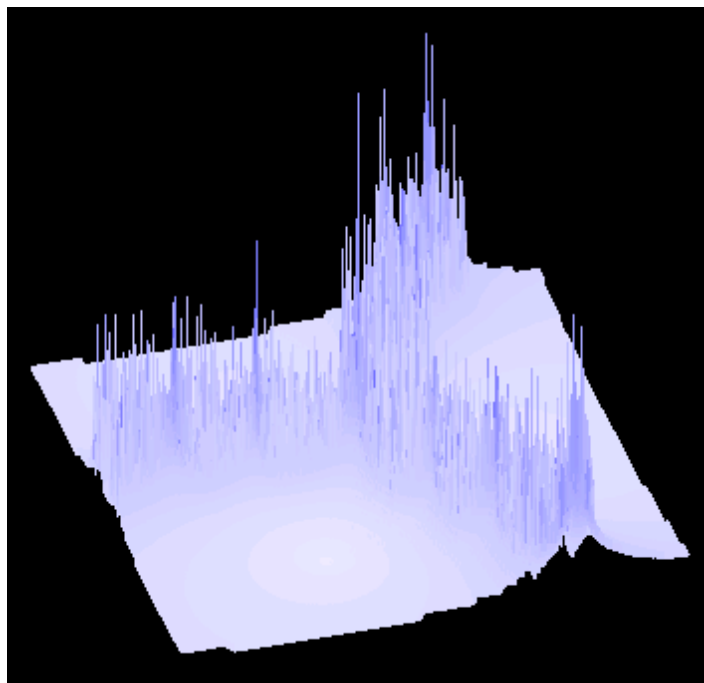


Figure 6: 3D Newton's Method for $z^3 - 1 = 0$.

reach a root at given condition¹⁰. Such index increases for closer and closer neighborhoods of the Julia set which splits the basins of convergence to the three roots, while it decreases as we get closer to the roots, as shown in the three 'valleys'. Any 3D plot can be rotated or viewed in perspective.

2.3 QUATERNIONS

Around early 80's the field of complex dynamics, also by means of the spectacular images of Julia sets, resurrected and earned so much popularity (getting an audience also outside the mathematicians circuit in those years) that became a mainstream scientific topic. This all let enthusiasms and aesthetic tastes grow up, pushing (me too!) to extend analogous graphic investigations¹¹ from \mathbb{C} to a larger numerical field, namely to quaternions¹². Practically the visualization of a Julia set is more laborious for quaternions and, due to a very huge number of computations therein involved, it runs quite slower than in \mathbb{C} . The problem of the immersion of this 4D object into a 3D space is of minor importance

¹⁰Here an accuracy of $10e^5$.

¹¹see [7], [11] and [12].

¹²Invented by Sir William Rowan Hamilton in the middle of the XIXth century, they consist of four (*quater*, in Latin) vectors $h = a + ib + jc + kd$, with non-commutative multiplication: in fact the quaternion field is not abelian and often indicated as \mathbb{H} . In the recent years, some software was developed to display Julia sets in the octonions, sedenions and in other numerical fields too.

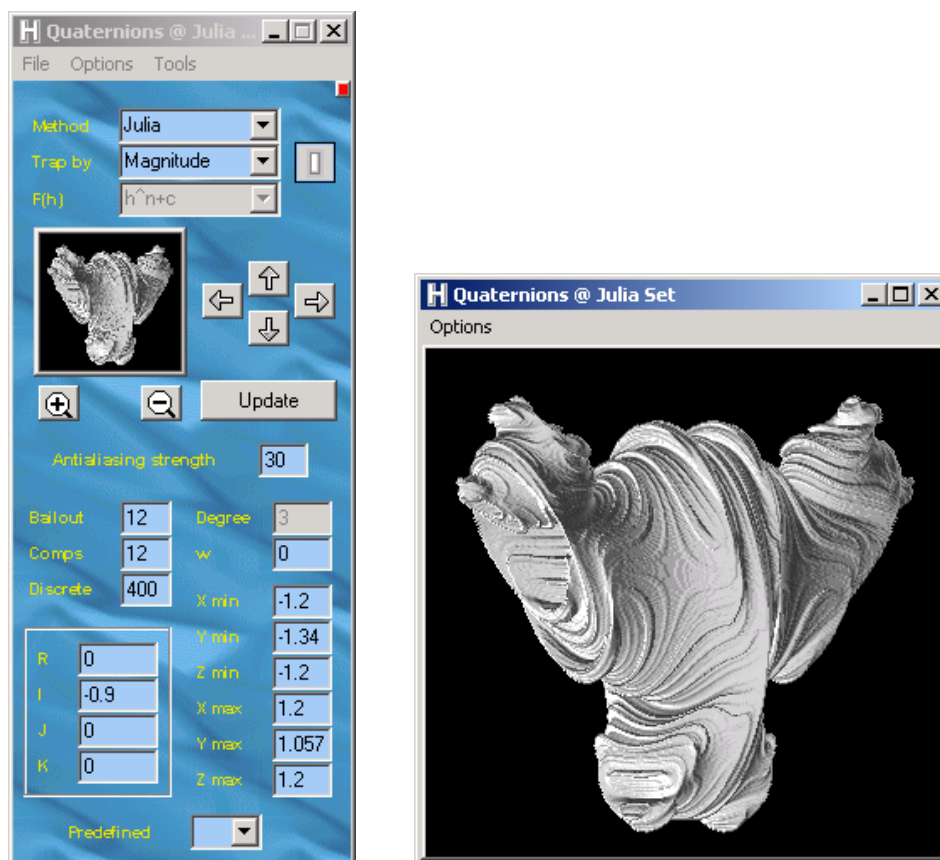


Figure 7: On the left, the starting panel for quaternions and a magnified view of a cubic Julia set on the right.

and its solution is easy and intuitive¹³. The main question here is how to accomplish a good rendering of such *solid* Julia sets because the images must evince the fractal nature of their boundary. Naive and rough computations might display 3D models with a flat looking so that bad¹⁴ images would come out. *IC* implements a solution relying on both a slightly modified version of Phong illumination model, see [6], p. 729, (assuring a very careful and realistic computation of the light rays intensity and reflection onto a surface) and an accurate scanning algorithm of the Julia set. *IC* opens to quaternion iterates by means of a parameters management panel (See Figure 7 on the left).

Since the goal of this article is just to show the features of *IC* and *Mapper*, the author will not get into technical details, referring the reader to [14] for a deeper discussion by pseudo-code in order to illustrate the general lines on how

¹³During the scanning of a whole 3D space - a cube for example - the three XYZ values of the ordered triplet in $X \times Y \times Z$ coordinates are set to a, b, c quaternion components respectively, which is tested to belong to the Julia set or not

¹⁴One important goal is to render the 'fractality' of the Julia set boundary. Since the details are too little, they could easily escape: a good solution was discovered to rely into an appropriate calculus of the light per each boundary point, given a source somewhere in the model.

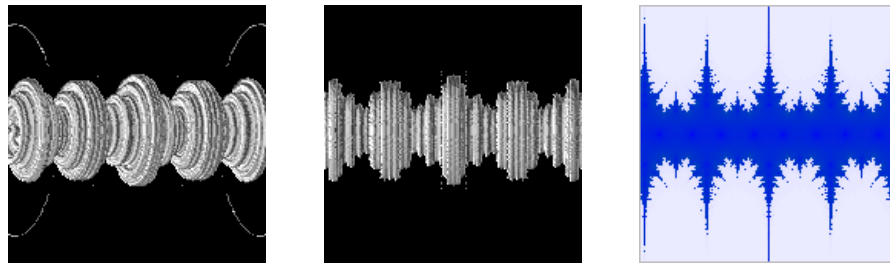


Figure 8: Two pictures of quaternion Julia sets for $\cos(h)$ and for $\cos(z)$ on the right. Note the analogous shape.

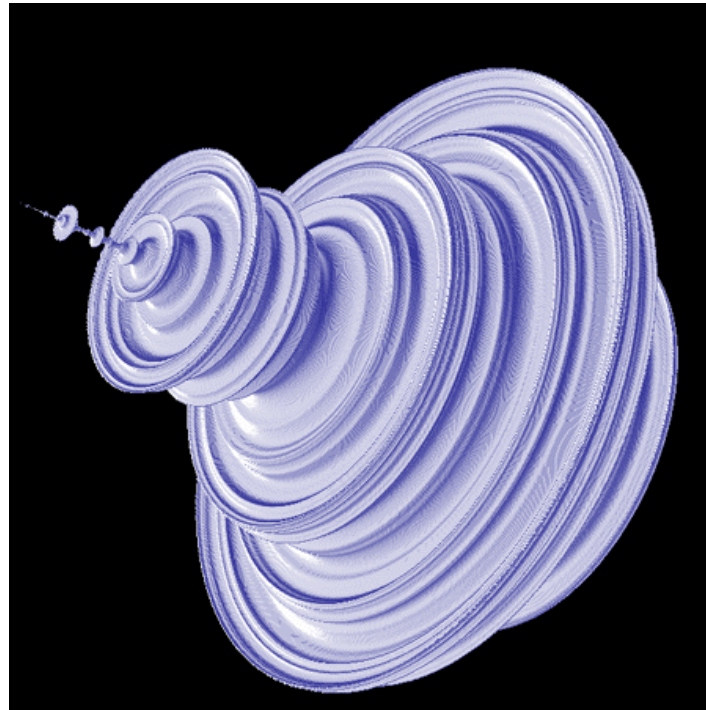


Figure 9: The Mandelbrot set in \mathbb{H} for $f(h) : h^2 + c$, where $h, c \in \mathbb{H}$.

to display and render these images. For an easier control, the author coded a C++ quaternion class for arithmetic and trigonometric, transcendental and hyperbolic operations *currently* solved via MacLaurin's series expansion¹⁵. It can be mathematically proven that quaternion Julia sets look like analogous sets in \mathbb{C} rotated along the vertical axis, say Y , as in Figures 8, where the Julia set associated to the iteration of the *cosine map* in the \mathbb{H} (left and middle) and in \mathbb{C} (right).

Finally the author argued that this method is inspired to an analogous one applied in \mathbb{C} : but it lacks in flexibility and, for quaternions, works with filled-in

¹⁵The author recently knew that direct formulas have been developed to compute such operations: *see* <http://world.std.com/~sweetser/index.html>. Their implementation will be pursued in next versions of *IC*.

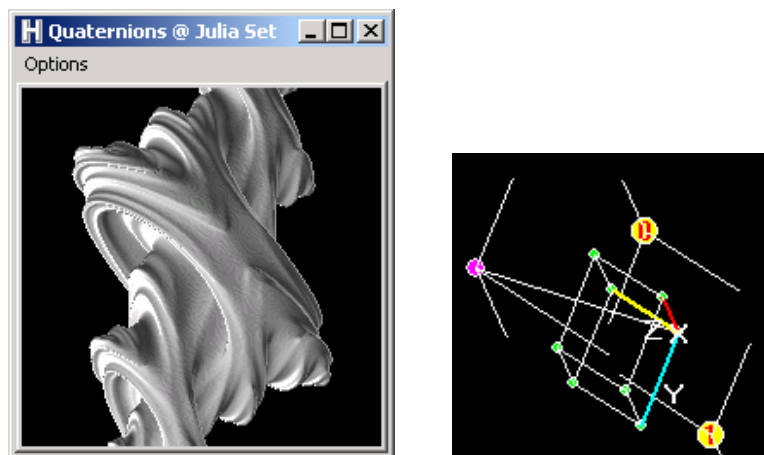


Figure 10: The quaternion Julia set for $z^2 + h$, where $h = 0.5555r + 0.5555i$, $h \in \mathbb{H}$. On the right, a 3D cube referencing the space position of the displayed 3D fractal.

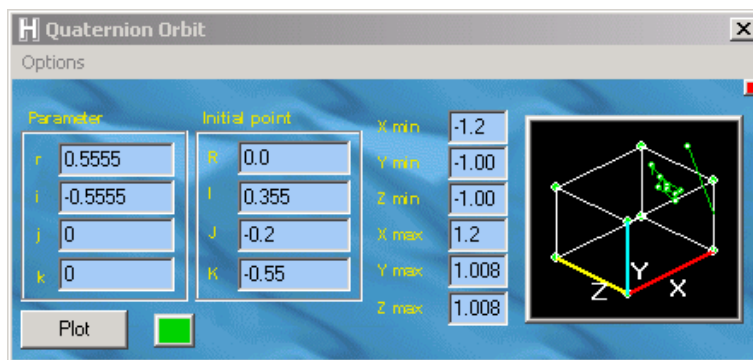


Figure 11: The analysis window tracking a single orbit in \mathbb{H} .

Julia sets¹⁶ only. Plotting other classes of Julia sets is an open problem for the author: for example, how to plot the quaternion Julia set coming from the Newton-Raphson method applied to $h^3 - 1 = 0$? *Inwards* allows a minimal analysis of such Julia sets by means of a window showing the orbit of a seed point, given the iterated map (see Figure 11).

2.4 TOWARDS DOMAIN COLORING ...

It will be discussed with more details inside the next section because this method is more consistent with the goals of *CM*.

¹⁶A 'filled-in Julia set' relates to iteration of polynomials and it is defined as the union of the Julia set of the polynomial itself and the bounded component of F . See [5], p. 65.

2.5 REALS

As long as *IC* was developed, it turned into a studio platform for reals too and then it came natural to implement some graphics for real maps, for sake of completeness and also to give the chance of studying a given case from different viewpoints, because it may be often required in mathematics.

3 Complex Mapper

As it also happened for *IC*, the birth of *CM* was exclusively dictated by practical needs and by new ideas coming up into my mind while working for the book [1]; so I started to code another program, with some features inherited from but lighter than in *IC*, relying on different software architecture and focusing on new graphical aspects. At the code level, the main difference between *IC* and *CM* is that the latter relies on a Multi-Document Interface (MDI), giving the chance of handling more experiments simultaneously. The main parent window is shown first, managing a set of child views, each one keeping the input formula to be examined. At a mathematical level, the basic feature (starting this all up) was the need of envisioning the *local* dynamical aspects of any (possibly iterated) map in one complex variable¹⁷. I mean that the graphical features already implemented into *IC*, more systematically based upon colors, put in evidence the final state of a dynamical system or, regarding the *Domain Coloring* method, the values distribution. I deliberately made *CM* evolve along the analytic path. The next paragraphs will enlighten the major aspects of *CM*. *CM* was essentially developed for everything to be user-friendly and easy to understand so that, in the latter direction, a (optionally removable) report window was implemented to let the user follow each step of any process.

3.1 LAYERS

All main processes have been organized as layers piled on each other: the working of *CM* is based upon a fixed sequence of sub-process. Levels are managed inside the window shown at Figure 12. The left column lists the levels, the right one indicates whether they are currently operative or not. The most relevant layers will be discussed further in details.

¹⁷The most important goals for this version of *CM* are essentially to display the local behavior about a singularity (critical, pole-like, essential) and the dynamics of a given iterated orbit.

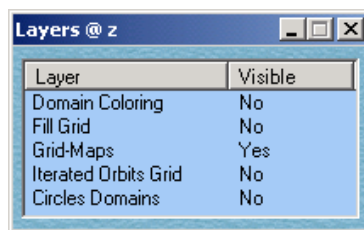
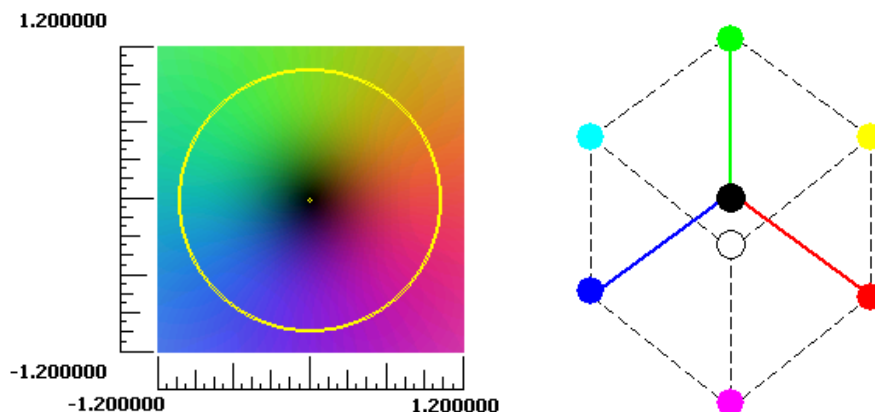


Figure 12: The levels management window.

Figure 13: On the left, coloring \mathbb{C} through the identity map and the model of the RGB-color cube on the right.

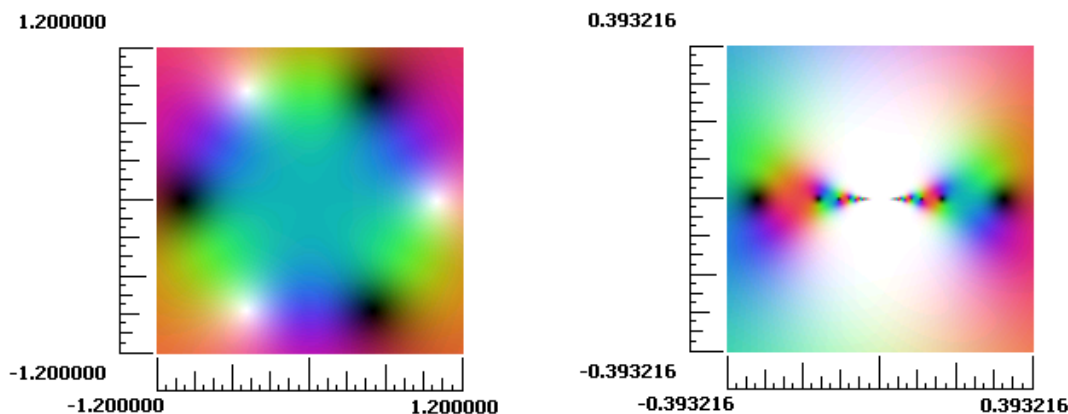
3.2 DOMAIN COLORING

This is a method applied recently in complex analysis and, as far as the author knows, spread inside some web-sites devoted to this field¹⁸.

A trouble with complex numbers is that they cannot be sorted. Either if assumed in the standard formula $z = x + iy$ or in the Euler's form $re^{i\theta}$, because they consist of two components: so it is not *a priori* decidable which component shall be preferred for the sort to take place. But classifications are required somehow since they are often useful to evince analytical properties. The '*Domain Coloring*' (DC) method overcomes this trouble and, getting rid of any standard approach, it invents a new way which sets a one-to-one map from \mathbb{C} (see Figure 13, left plot) onto the RGB¹⁹ cube (right plot) so that each complex point is associated to one and only one color.

¹⁸See Farris at <http://rici.scu.edu/~farris>, Godfrey at <http://winnie.fit.edu/~gabdo>, Lundmark at <http://www.mai.liu.se/~halun>.

¹⁹It is the acronym of Red-Green-Blue, one of the so-called 'additive systems', applied by many standards in video-diffusion. Each color is detected by a vector of 3 *primary* (that is, assumed to be undecomposable in this system itself) colors. If the range is the same for all 3 components, a cube can be ideally modeled (Figure 13 on the right). See [6], p. 584.

Figure 14: *Domain coloring in action.*

In particular, the left plot of Figure 13 displays how \mathbb{C} is colored²⁰ under the action of the identity map $f(z) : z$ and *DC*, the simplest way to show how *DC* colors the complex plane. The unit circle has been plot in yellow in order to gain a start point for understanding the method itself. Figure 14 (on the left) shows *DC* in action with $f(z) = (z^3 + 1)/(z^3 - 1)$. The black and white regions are located, as known, on the unit circle and they detect the zeros and poles respectively. The right plot shows the wild behavior about the essential singularity at the origin for $f(z) = \sin(1/z)$.

Anyway, up to now, we gave nothing new about *DC*: these first lines were only a very short survey on its use in complex analysis. My new simple idea consists in applying *DC* to complex dynamics too, in order to study the final values distribution after a finite number of iterates: if *DC* is applied to the image values returned by $f(z)$ as usual, why cannot we do the same with $f^{\circ n}(z)$, again a complex map where n is just the index of the n^{th} -fold application of $f(z)$? Test showed that, if properly used in complex dynamics, *DC* can reveal the Julia sets structure, for example, if consisting of pure repelling fixed points or poles. Let us take a couple of examples. Figures 15 display the Julia set of $f(z) = z^2 - 0.2 + 0.7i$. We have two basins of attraction: the white²¹ one to the point at infinity and the (bounded) basin to an attracting cycle of period 3²². Figures 16 show the known example of the Newton-Raphson method applied

²⁰I slightly modified the original (*DC*) algorithm so that regions neighboring the zeros and poles are painted black and white respectively. The original version does the converse.

²¹In our color convention.

²²We need to precise, in order to fix the power scope of *DC*, that this feature was *evinced*, not found, by this method, that is, it came up after some tests. See conclusions inside section 4 for more details.

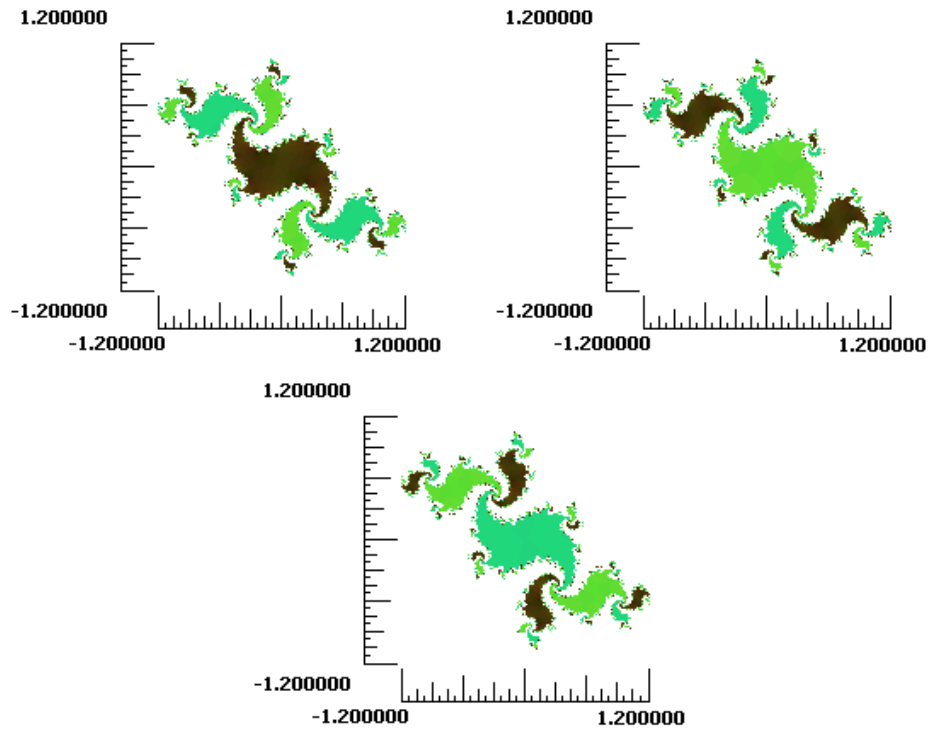


Figure 15: Displaying a cycle of period 3 via *domain coloring*.

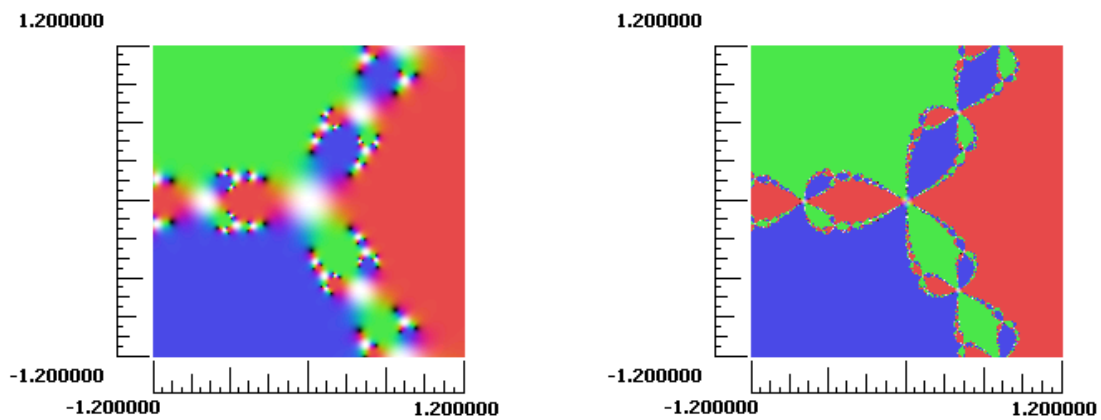


Figure 16: Julia set generation for $z^3 - 1 = 0$ under Newton-Raphson's method.

to the equation $f(z): z^3 - 1 = 0$, yielding the transformed map $g(z)$:

$$\frac{2z^3 + 1}{3z^2}$$

The left plot shows the state of the system after 5 iterations; notice the generation of the Julia set at the poles (white regions). On the right, the same system after an higher number of iterations: now there is no *sensitive dependence* on the value of the seed point²³. Only three colors are shown and this proves empirically that all orbits (except at the Julia set) converge to the attracting fixed points of $g(z)$, that is, to the zeros of $f(z)$.

3.3 GRID-MAPS

The concept behind the method of ‘*grid-maps*’ (*GD*), as I named²⁴, is very simple. Grid-maps can be displayed in three steps: (1) assume a grid²⁵ of complex points, (2) map (or iterate) such points, (3) link the resulting image points by the same rule as at 1. *CM* implements the *orthogonal* and the *radial* grid-maps, displayed in Figures 17. Early α -versions of *CM* implemented only the orthogonal type but their running suggested to give the user the chance of choosing the radial shape too, because there would be cases whose understanding requires different grid-maps. *GD* are very useful to look at the local dynamics about a singularity. For example, we will illustrate (locally) the final state of two local dynamical systems: the first one with a rationally indifferent fixed point and the other with an irrationally indifferent fixed point²⁶.

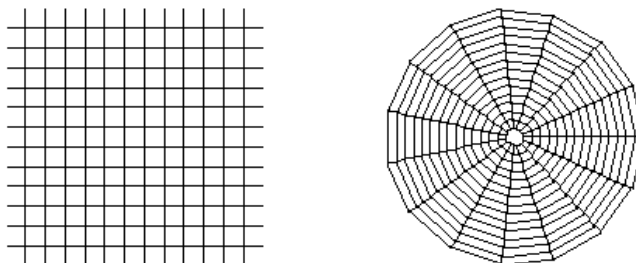


Figure 17: The *source* orthogonal and radial grids.

²³With regard to the famous expression, this is the *initial condition*.

²⁴I apologize myself for my lack of knowledge in the literature of methods in complex analysis: maybe this same method keeps already a name but I do not know any standard nomenclature yet. As far as I know, I found it in [10], but without any given standard nomenclature.

²⁵We do not intend a grid as only a geometrical shape but, more abstractly, as the rules linking all points.

²⁶The author assumed that the reader already keeps a basic knowledge of complex dynamics.

The first example is shown by iterating $f(z) = z + z^5$: this map has a rationally indifferent point at the origin where four attracting and four repelling directions branch out. Here the holomorphic germ of the Taylor expansion about the origin can be represented in the form $z + e^{i\theta}z^n$, where $n - 1$ is the number of attracting and repelling directions and θ is a rational number.

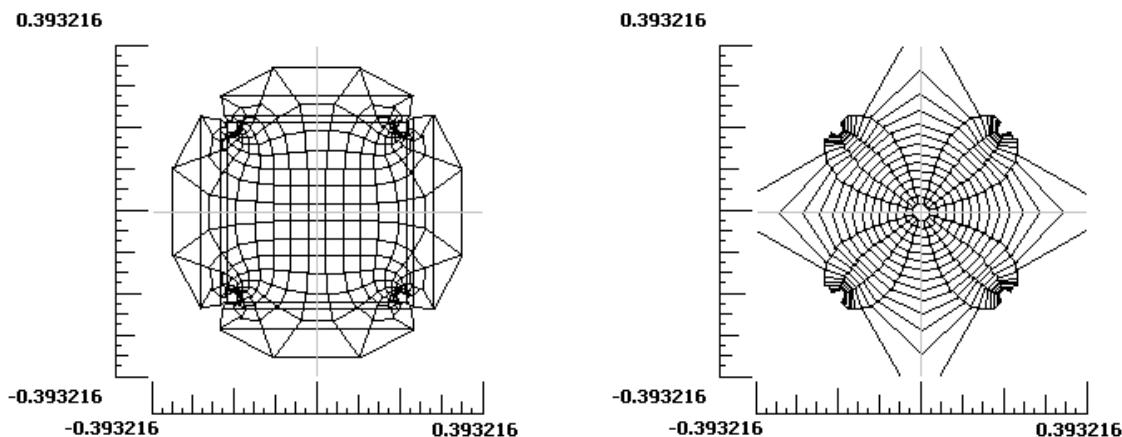


Figure 18: Applying the two grids to the dynamics about a rationally indifferent point.

Iterates give rise to the known vector field, namely the Fatou-Leau flower with 4 petals. Applying the orthogonal grid does not draw any clear conclusion: we are just able to deduce a slower and a slower convergence rate as nearer as we get to the origin: the graphic result is that a new grid, quasi-orthogonal as the source one, is displayed (look at the left image of Figures 18): the image grid looks like the source orthogonal one about the origin, whereas it gets more and more deformed as far as we move away. On the contrary the radial grid works better, as we can evince from the figure 17 on the right where the characteristic shape of the Fatou-Leau flower comes out, having four repelling directions localized at the four points $\pm 1, \pm i$, as predicted by the theory. So this question comes out naturally: is there a way to determine which is the most suitable grid model for a given case?

Actually no definitive response can be given and it is opined that only the heuristic approach rules here, therefore no straight rules can be defined because all depends on user needs. The radial grid also helps here to understand that a sufficiently small neighborhood of the rationally indifferent point can be conjugated to an open disk centered at the origin and the radial grid appeared as the obvious solution to evince these dynamics in the best way. The second example shows the dynamics about an irrationally indifferent fixed point, namely when the holomorphic germ of the Taylor expansion about the origin is in the form $z + e^{i\theta}z^n$ where θ is irrational and satisfying a diophantine condition. In

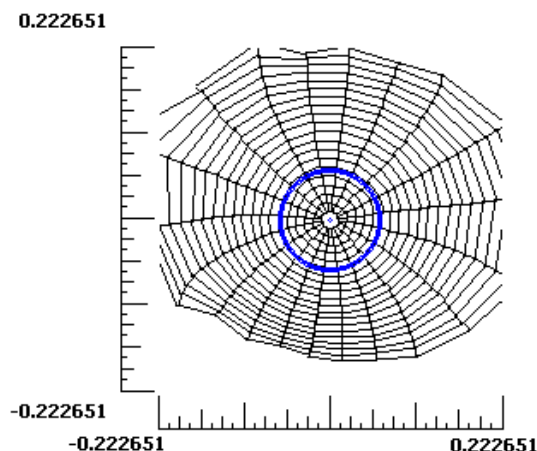


Figure 19: A blow-up of a neighborhood enclosing a Siegel disk. The blue circle tracks approximately that maximal rotation simply connected domain.

1942 C.L. Siegel proved that there exists a subset of irrational numbers enjoying this property and allowing the linearization about the fixed point so that a bounded neighborhood of the origin is conformally equivalent to a rotational disk (so called 'Siegel disk'). Irrationals which are not diophantine are defined 'Liouville numbers', first studied by Cremer ([3] and [4]) between 20's and 30's. Irrationally indifferent points split in *Siegel* and *Cremer* points, whether the linearization applies or not²⁷. The radial grid is the best to enlighten the experiment here again. A given neighborhood, deliberately exceeding the linearization domain, was chosen to look at the deformation outside the maximal rotation domain (here conformal to a disk), as clearly shown in Figure 19.

3.4 CIRCLE / POLYGONAL DOMAINS

Sometimes a grid-map, in both forms, could produce too messy graphics and thus it is useful to have part of the source grid-map, say a circle or an n -side polygonal domain, whose generation is achieved by means of parameters management window including, in particular, a combo-box listing three options: 'Draw', 'Map' and 'Contour'.

The first option simply draws the circle/polygon, the second tracks the orbit exclusively and the last one plots the contours passing through the iterated polygon points, step by step. We will show here four different cases where

²⁷The set of such numbers has zero Lebesgue measure. Ricardo Pérez-Marco posed me [13] the open question on how to display, through digital graphics, the shape of the univalent neighborhood arising from the iterates of a given $f(z)$ about a Cremer point: the finite digits computation breaks the irrationality, approximating and turning the same number into an algebraic one, always diophantine so that the linearization applies and the experiment does not make sense anymore.

these options show to be much useful.

Figures 20 show the Fatou-Leau flower again but via polygonal domains: on the left, the boundary points of four circles mutually tangent at the origin and with very small radius are iterated²⁸; the right picture refers the same example but two concentric circles are assumed. Figure 21 displays the simple case $f(z) = z^2$: three (differently colored) source circles outside ($r = 1.1$), inside ($r = 0.8$) and on the unit disk ($r = 1$) have been iterated²⁹ and plot respectively. Finally Figure 22 shows the rotational character of the dynamics in the neighborhood of an Herman ring generated by iterating the complex map (see [9], page 150) $f(z) : e^{2\pi i\theta} z^2 \frac{(z-4)}{(1-4z)}$, where $\theta = 0.6151732\dots$.

3.5 ADDITIONAL SUBSTITUTIONS

Often the study of a the behavior of a map (that a given $f(z)$ iterated once) or of a dynamical system (iterated more than once) requires to deal with additional formulas in order to evince graphically some properties. *CM* achieves it by means of the *additional substitutions* feature: let the input map $f(z)$ be the main one. Two management modes have been implemented: ‘*Simple*’ and ‘*Attach*’. The first one refers to a map applied to the image value of (possibly iterated) $f(z)$. For example let $f(z) : 1/z$ and $g(z) : \cos(z)$ be the main and the additional map respectively. Then *CM* displays the point returned by $g(f(z)) : \cos(1/z)$. The ‘*Attach*’ mode simply appends $g(z)$ to $f(z)$. For example, assume that the local (i.e. inside the given region) fixed points are to be displayed after a given number of iterations: so let $g(z) = -z$ be the additional substitution helping to detect the fixed points. We can attach it to $f^{on}(z)$ and then have $f^{on}(z) - z$, whose zeros are the required fixed points. The following example could be more inspiring. In figures 18, we are studying the dynamical systems generated by the iteration of the quadratic map $f(z) = z^2$. As known, the Julia set is the unit circle, which splits \mathbb{C} into two basins: the interior ($|z| < 1$) and the exterior ($|z| > 1$) of the unit circle.

The left image of Figures 23 came out as we iterated $f(z) = z^2$ five times and attached it to $g(z) = -z$; thus it shows the localization of ($2^5 = 32$) fixed points, i.e. all zeros of $f^{o5}(z) - z = 0$: $z^{32} - z = 0$. What we see is one attracting point at the origin and 31 repelling ones, belonging to the Julia set, on the unit circle. The figure on the right shows the values distribution of —

²⁸Figure 20 is a computer display of the geometrical evidence of such flowers: its generation is inspired from Julia’s related proof of flowers. See [8], fourth part.

²⁹Circles with a same color are the forward images.

$f'(z) = 2z$ for $f^{\circ 5}(z) : z^{32}$. As expected, while the values inside the unit disk are black (i.e. the values of the first derivative is very close to the origin, which is therefore an attracting fixed point), the fixed points on the unit circle turn into red (the value is 2 - compare to the identity map shown in figure 13), showing their repelling character because of $|f'(z)| > 1$.

4 CONCLUSIONS

I am very indebted to Daniel S. Alexander (Drake University, Iowa) and to Felice Iavernaro (University of Bari, Italy) for revising this article and giving me many tips to improve it. The author recalls that these methods presented are not to be assumed as a true way to do Mathematics: their goal is to support the understanding of results exclusively, which have to be accomplished via analytic approach, based upon rigor, logic and evidence. *IC* and *CM* can be freely downloaded, for researching and teaching purposes, at <http://www.malilla.supereva.it>. The development of *IC* and *CM* is currently in progress, suggestions are very well appreciated. The reader is invited to get in touch with the author if further details, on ideas and algorithms computing all images throughout this article, will be required.

References

- [1] Alexander D.S., Iavernaro F., Rosa A., *Early days in complex dynamics*, (in preparation).
- [2] Barnsley M.F., Devaney R.L., Mandelbrot B.B., Peitgen H.-O., Saupe D., Voss R.F., *The science of fractal images*, Springer-Verlag, 1988.
- [3] Cremer H., *Überkonvergenz und Zentrumproblem*, Math. Annalen, 110, 1935, 739-744.
- [4] Cremer H., *Über die Häufigkeit der Nichtzentren*, Math. Annalen, 115, 1937, 573-580.
- [5] Carleson L., Gamelin T.W., *Complex dynamics*, Springer-Verlag, 1993.
- [6] Foley J.D., van Dam A., Feiner S.K., Hughes J.F., *Computer Graphics, principles and practice*, 2nd edition in C, Addison-Wesley 1997.

- [7] Hart J.C., Sandin D.J., Kauffman L.H., *Ray Tracing Deterministic 3-D Fractals*, Computer Graphics, 23, 3, July 1989, 289-296.
- [8] Julia G., *Mémoire sur les itérations des fonctions rationnelles*, Journ. Math. Pure App., 8, (1918), 47-245.
- [9] Milnor J., *Dynamics in one complex variable*, Vieweg, 2000.
- [10] Needham T., *Visual complex analysis*, Oxford Univ. Press, 2000.
- [11] Norton V.A., *Generation and rendering of geometric fractals in 3-D*, Computer Graphics, 16, 3, 1982, 61-67.
- [12] Norton V.A., Melton E., *A close encounter in the fourth dimension*, SIGGRAPH Video Review, 39, 1988, 30.
- [13] Pérez-Marco R., *an informal discussion*, 2002.
- [14] Rosa A., *How to plot a Quaternion Julia Set?*, draft, downloadable at <http://malilla.supereva.it/Mirror/Pages/papers.html>

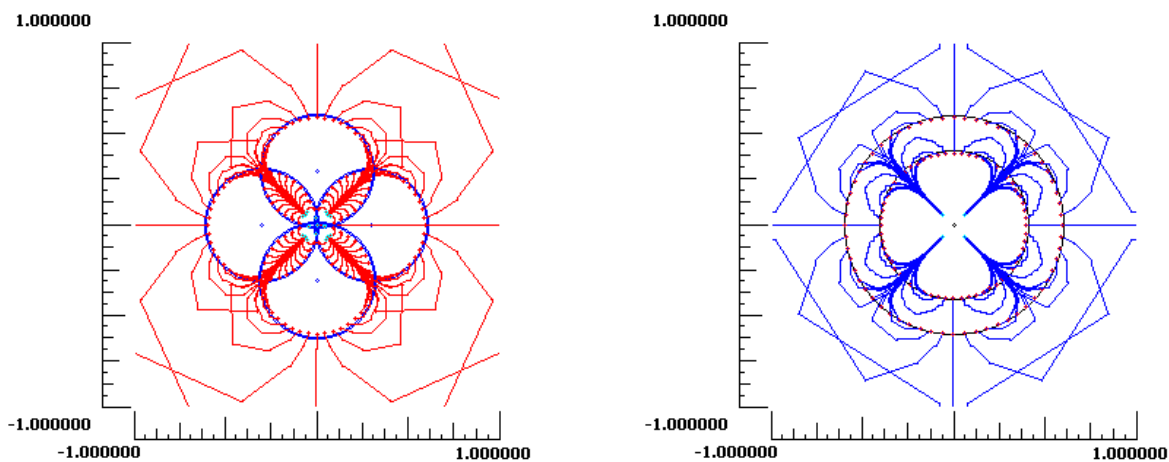


Figure 20: Polygonal domains iterated by $z + z^5 = 0$ about the origin.

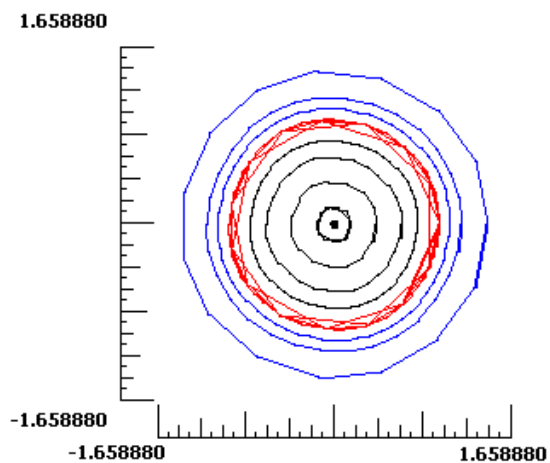


Figure 21: Dynamics of z^2 about the origin.

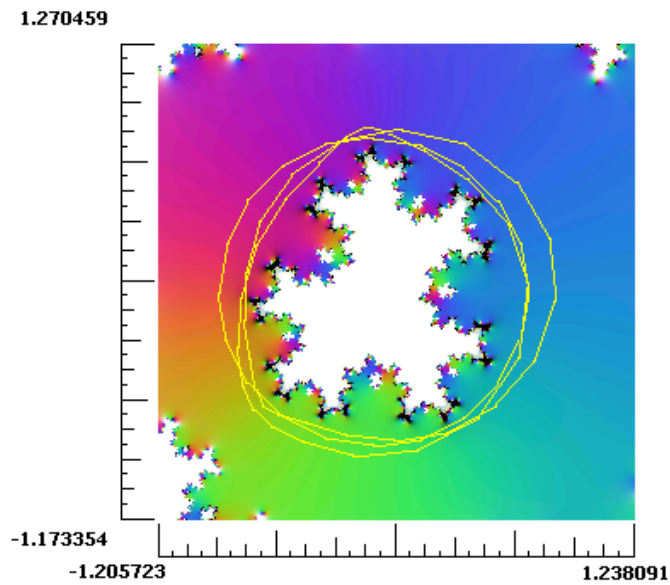


Figure 22: Dynamics inside a region around a Herman ring, viewed by *domain coloring* and *polygonal domains*.

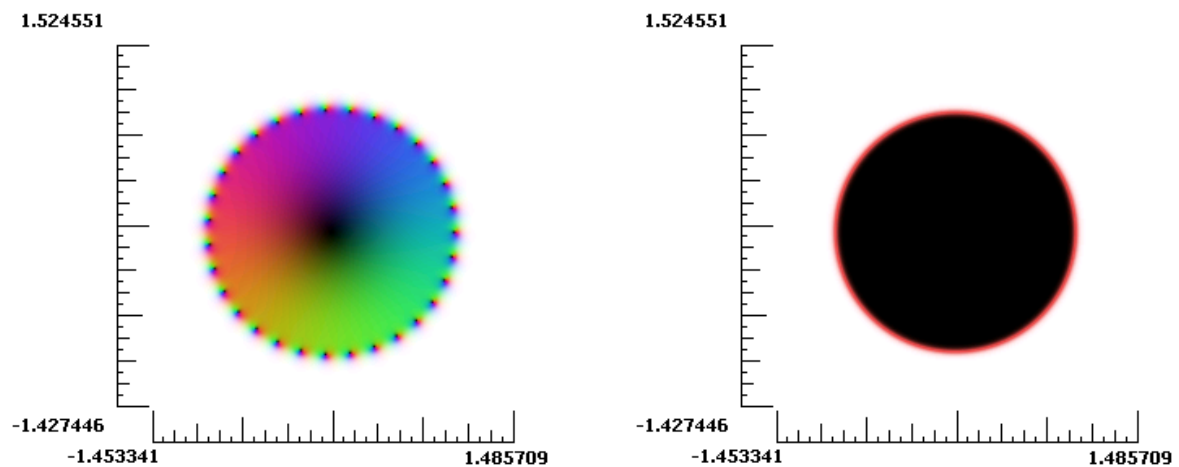


Figure 23: Illustrating *Additional substitutions* via *Domain Coloring*.