# Minimum Modulus Visualization of Algebraic Fractals*

**Severino F. Galán**
Artificial Intelligence Dept. at UNED
C/ Juan del Rosal 16, 28040 Madrid (Spain)
E-mail: seve@dia.uned.es

**Abstract**

Fractals are a family of shapes formed by irregular and fragmented patterns. They can be classified into two main groups: geometric and algebraic. Whereas the former are characterized by a fixed geometric replacement rule, the latter are defined by a recurrence function in the complex plane. The classical method for visualizing algebraic fractals considers the sequence of complex numbers originated from each point in the complex plane. Thus, each original point is colored depending on whether its generated sequence escapes to infinity. The present work introduces a novel visualization method for algebraic fractals. This method colors each original point by taking into account the complex number with minimum modulus within its generated sequence. The advantages of the novel method are twofold: on the one hand, it preserves the fractal view that the classical method offers of the escape set boundary and, on the other hand, it additionally provides interesting visual details of the prisoner set (the complement of the escape set). The novel method is comparatively evaluated with other classical and non-classical visualization methods of fractals, giving rise to aesthetic views of prisoner sets.

**Keywords:** Complex plane, iterated complex function, algebraic fractal, visualization, minimum modulus.

## 1 Introduction

Fractal geometry [21, 24, 10, 9, 12] studies shapes characterized by irregular and fragmented patterns exhibiting certain degree of self-similarity. The term "fractal" [19, 20] was coined by Benoit B. Mandelbrot in 1975, who formally defined it as *a set for which the Hausdorff-Besicovitch dimension strictly exceeds the topological dimension.*

Fractals can be divided into two main categories: geometric and algebraic. Geometric fractals are generated by means of a fixed replacement rule. In general, an object known as *initiator* is

replaced with another object called *generator*. Then, such a replacement is repeated independently for several parts of the resulting object. Iterating this process gives rise to a geometric fractal as illustrated in Section 2.1 for some classical examples. Algebraic fractals are obtained by a recurrence function in the complex plane. A complex function $f(z)$ is iteratively applied for each point $z_0 \in \mathbb{C}$, giving rise to a sequence of complex numbers $\{z_0, f(z_0), f(f(z_0)), \ldots\}$. Each point $z_0$ is colored depending on whether the modulus of the complex numbers in its associated sequence escapes to infinity.

The usual method of visualizing algebraic fractals is based on analyzing the sequence of complex numbers originated from iteratively applying $f(z)$ to each point $z_0 \in \mathbb{C}$. Typically, each original point $z_0$ is colored depending on whether or not its generated sequence diverges to infinity, that is, the modulus of the complex numbers in the sequence originated from $z_0$ becomes greater than a large enough threshold. As an alternative to binary colors, the time at which the sequence reaches the threshold is usually employed to scale the color. Other relevant methods for fractal visualization are based on different techniques like the following:

- At each time, coloring $z_0$ with the phase of the current complex number in its sequence [35].
- Scaling the color for $z_0$ according to the minimum distance to the real or imaginary axes of the complex numbers in its entire sequence [25, 26]. This method is known as "Pickover stalks" or "Pickover biomorphs" in the literature.
- Redefining the escape condition for the sequence of complex numbers as falling within a region in $\mathbb{C}$ called "orbit trap", which can be arbitrarily defined by the user [33].
- Selecting the initial points randomly, iterating only those initial points that escape, counting the pixels where they land on, and generating a grey-scale image from the hit counts. This method is called "The Buddhabrot Technique" [13].
- Iterating the "distance ratio" with two points, which generates generalized fractals [37].

This work presents a novel visualization method for algebraic fractals which is similar to that of C. A. Pickover. The method colors each original point in the complex plane by selecting the complex number with minimum modulus within its generated sequence. Thus, whereas C. A. Pickover chooses the complex number in the sequence with minimum distance to the axes of the complex plane, the novel method picks up the complex number in the sequence with minimum modulus. (Note that these two complex numbers can be quite different.) The novel method preserves the fractal view that the classical visualization method offers of the boundary for the escape set. Furthermore, when compared with the other visualization methods, it provides aesthetic visual details of the prisoner set that complements the escape set.

The rest of the paper is organized as follows. Section 2 reviews the most relevant types of fractals and how they are visualized. Section 3 describes the new visualization method for algebraic fractals that is based on the concept of minimum modulus. Section 4 comparatively evaluates the novel method from a visual perspective. Finally, Section 5 contains the main conclusions and discusses future research directions.

# 2 Related work

Fractals are ubiquitous in nature. Physical objects as diverse as coastlines, clouds, mountains, or trees (among many others) which are difficult to describe through Euclidean geometry, can be easily modeled by means of fractal geometry. Nonetheless, due to their inherent aesthetic

properties, the most famous fractals are those that were created by mathematicians and have no parallel in nature. From a general point of view, depending on the way they are generated, fractals can be classified into two groups: *geometric* and *algebraic*. In the rest of this section, the two groups are reviewed in turn.

## 2.1 Geometric fractals

Fractals that can be generated by means of a fixed geometric replacement rule are known as geometric. Specifically, an object called *initiator* is considered and replaced with another object called *generator*. Next, the mentioned replacement is repeated in an independent way for certain parts of the resulting object which are similar to the initiator. The final geometric fractal results from iterating this process over time.

Some relevant examples of geometric fractals, designed by reputed mathematicians over the past few centuries, are included in Figure 1. The initiator, the generator and the final fractal are depicted in each case. Since these fractals represented a challenge to the continuity and differentiability concepts of Euclidean geometry, they were known as "mathematical monsters". As described in [21, 24, 29, 12], geometric fractals can also be used to simulate natural objects exhibiting fractal characteristics like mountains, clouds, plants, etc.

## 2.2 Algebraic fractals

Algebraic fractals are those generated by the iterative application of a function defined in the complex plane. For every point $z_0$ within the studied domain, a complex function $f(z)$ is repeatedly applied with $z_0$ as initial value, originating a sequence of complex numbers $\{z_0, f(z_0), f(f(z_0)), \ldots\}$. This technique, called "Picard iteration", is the option used in the present work. Other types of iterations employed in fractal geometry can be found in [22, 15, 23, 16, 38, 31, 18, 28].

The visualization of algebraic fractals is performed by analyzing the sequence of complex numbers originated from iteratively applying $f(z)$ to each point $z_0$. Usually, each original point $z_0$ is colored depending on whether or not its generated sequence diverges to infinity, that is, the modulus of the complex numbers in the sequence created from $z_0$ becomes greater than a large enough threshold. The threshold conditions under which escape is guaranteed to take place for different complex functions have been extensively studied in the literature [5].

As an alternative to binary colors, given a fixed maximum number of iterations $i_{\max}$, the specific iteration at which the threshold is reached for a point $z_0$ is typically employed to scale the color at that point. Thus, if the first and last iterations at which the threshold is reached within the whole complex domain studied are denoted as $i_{\text{first}} \leq i_{\max}$ and $i_{\text{last}} \leq i_{\max}$ respectively, shades of color can be assigned to points. In this way, the shade of color assigned to $z_0$ is proportional to the iteration at which the threshold is reached by the sequence originated from $z_0$ such that "white" corresponds to $i_{\text{first}}$ and "black" corresponds to iterations after $i_{\text{last}}$. Consequently, prisoner points are colored in black.

Figure 2 contains several examples of classical visualization of fractals through binary (left column) and shaded (right column) colors. In the case of shaded colors, a logarithmic scale is used in order to emphasize the details and, like in the rest of figures in this work, a threshold equal to $10^6$ is employed. The examples shown in Figure 2 correspond to: (1) Julia fractals [17] for the functions $f(z) = z^2 + c$ and $f(z) = z^3 + c$, where $c \in \mathbb{C}$ is a constant complex number, and (2) the Mandelbrot fractal [21] for the function $f(z) = z^2 + z_0$ where $z_0 \in \mathbb{C}$ is the complex number to be colored.

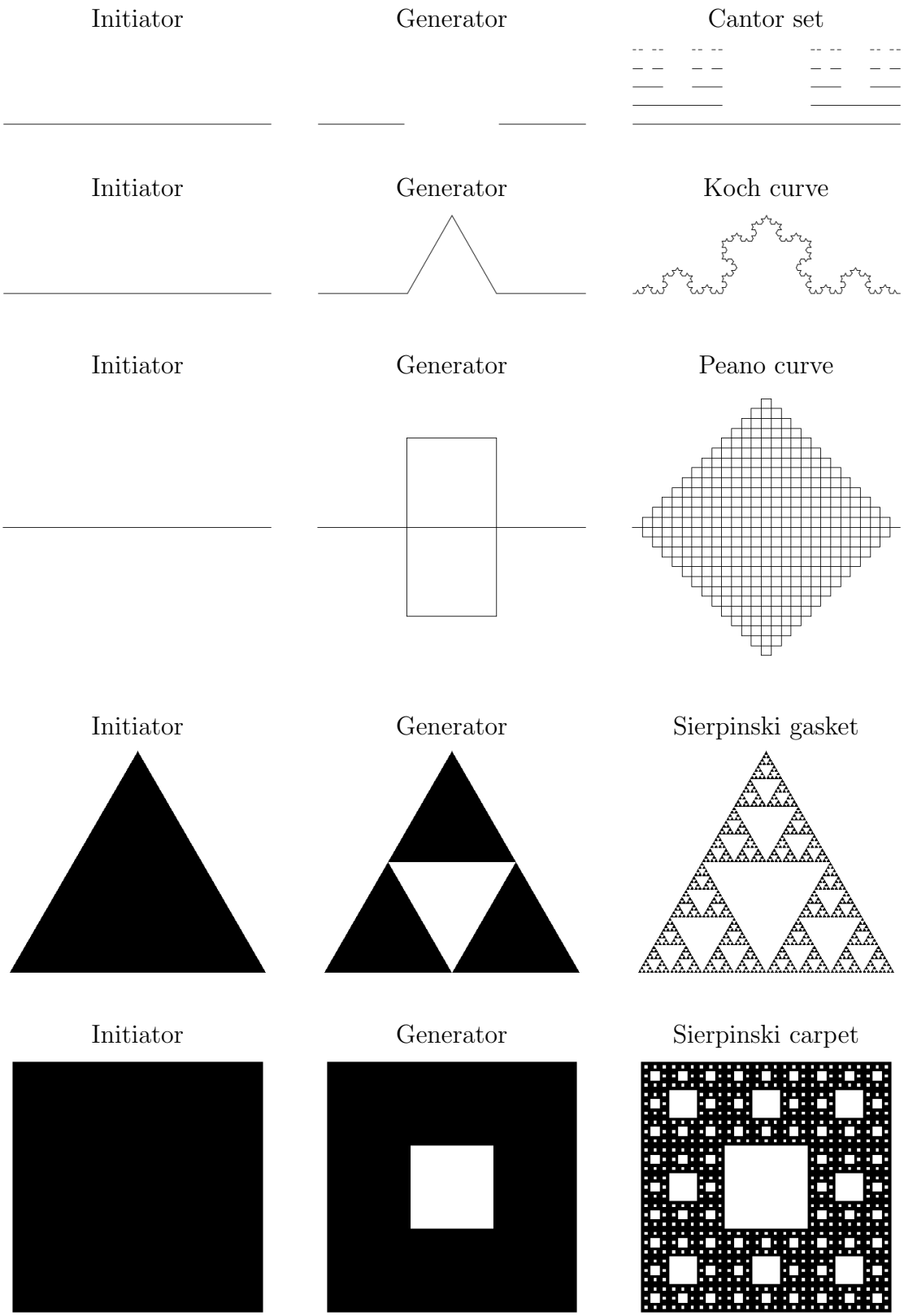



Figure 1: Examples of geometric fractals. The initiator, the generator and the final fractal are illustrated in each row. Note that the generator for the Peano curve consists of nine segments whose length is equal to one third of the initiator length.

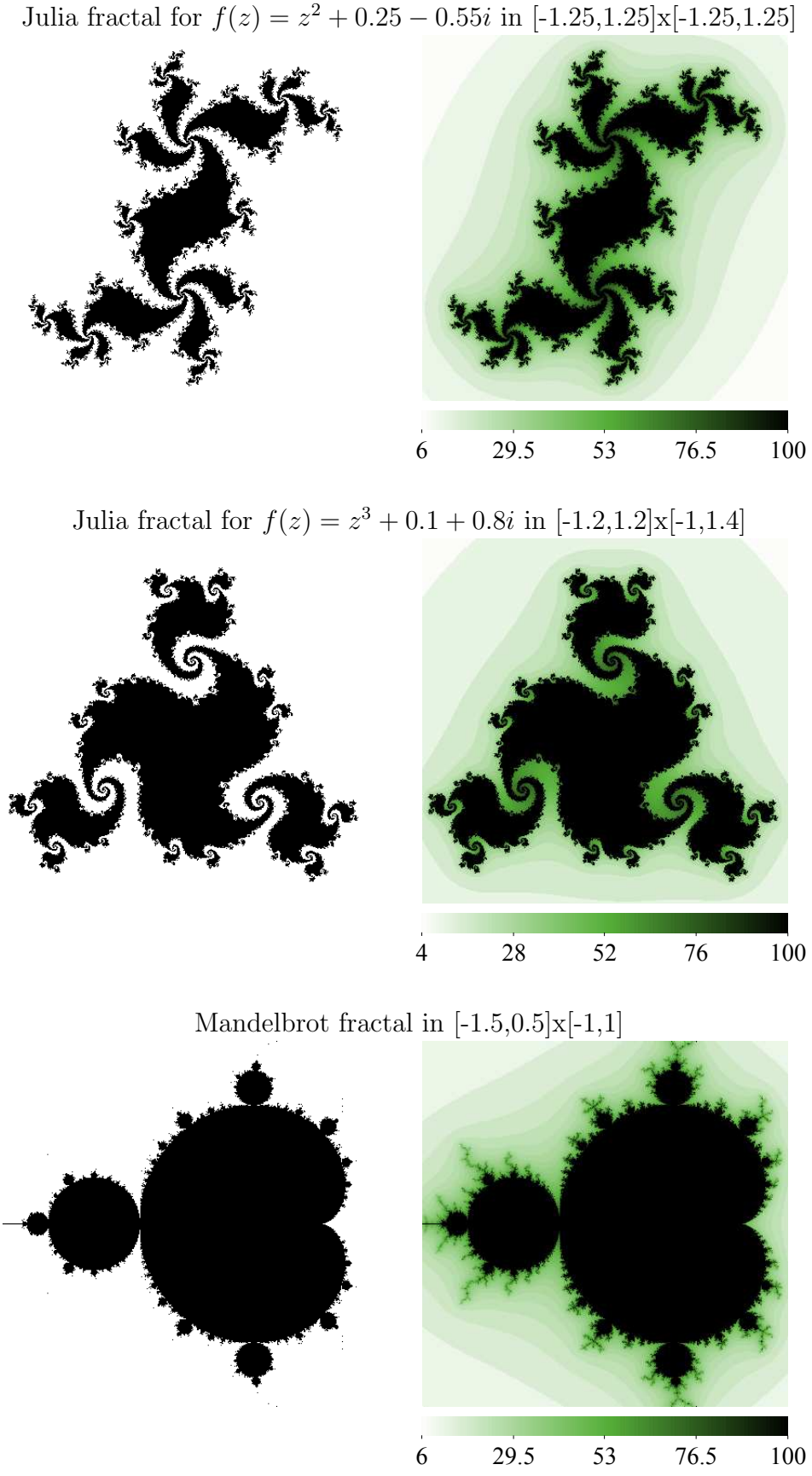



Figure 2: Examples of classical visualization of fractals. The left column corresponds to binary coloring, whereas the right column corresponds to shaded coloring. A 401x401 grid is used in discretized space and shown at iteration $i_{\max} = 100$.

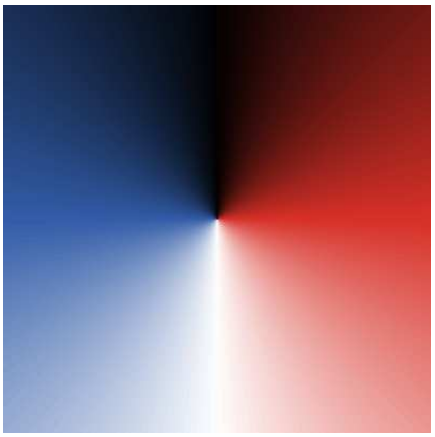


Figure 3: Correspondence between angles and colors used in this work for the *phase plot* visualization method.

Other relevant non-classical methods for visualizing fractals have been described in the literature. Among them, the methods closely related with the present work are the following:

- The phase of the corresponding complex number in the sequence (angle formed by its associated vector with the real axis) can be employed to color the original point $z_0$ at each iteration [35]. If the sequence escapes to infinity at iteration $t$, the phase of the complex number generated at $t-1$ is used to color $z_0$ from iteration $t$ onwards. Figure 3 shows the correspondence between angles and colors used in the present work for the *phase plot* method.

- C. A. Pickover [25, 26] determined the color associated to $z_0$ according to the minimum distance to the real or imaginary axes of the complex numbers in the entire sequence originated from $z_0$. This method is called *biomorphs* in the rest of the paper.

Figure 4 illustrates the examples in Figure 2 visualized through the phase plot method (left column) and through the biomorphs method (right column). In order to accentuate the details, a logarithmic scale is applied in the case of the biomorphs method.

# 3 Visualization by minimum modulus

This section presents a novel visualization method for algebraic fractals whose underlying idea is more similar to that of the biomorphs method than to that of the classical method. Under the new method, each original point in the complex plane $z_0$ is assigned the complex number with minimum modulus within its generated sequence $\{z_0, f(z_0), f(f(z_0)), \ldots\}$, and $z_0$ is colored with a shade proportional to the iteration at which such a minimum-modulus complex number was generated. If more than one complex number in the sequence have the minimum modulus, the first of the corresponding iterations is chosen.

Algorithm 1 contains the pseudocode for the novel visualization method. Given a fixed maximum number of iterations $i_{\max}$, the specific iteration at which the minimum modulus is reached for a point $z_0$ is employed to scale the color at that point. In this way, if the first and last iterations at which a minimum modulus is reached within the whole complex domain studied are denoted as $i_{\text{first}} \leq i_{\max}$ and $i_{\text{last}} \leq i_{\max}$ respectively, shades of color can be assigned to points such that "black" corresponds to $i_{\text{first}}$ and "white" corresponds to iterations after $i_{\text{last}}$ (see Step 20 in Algorithm 1).

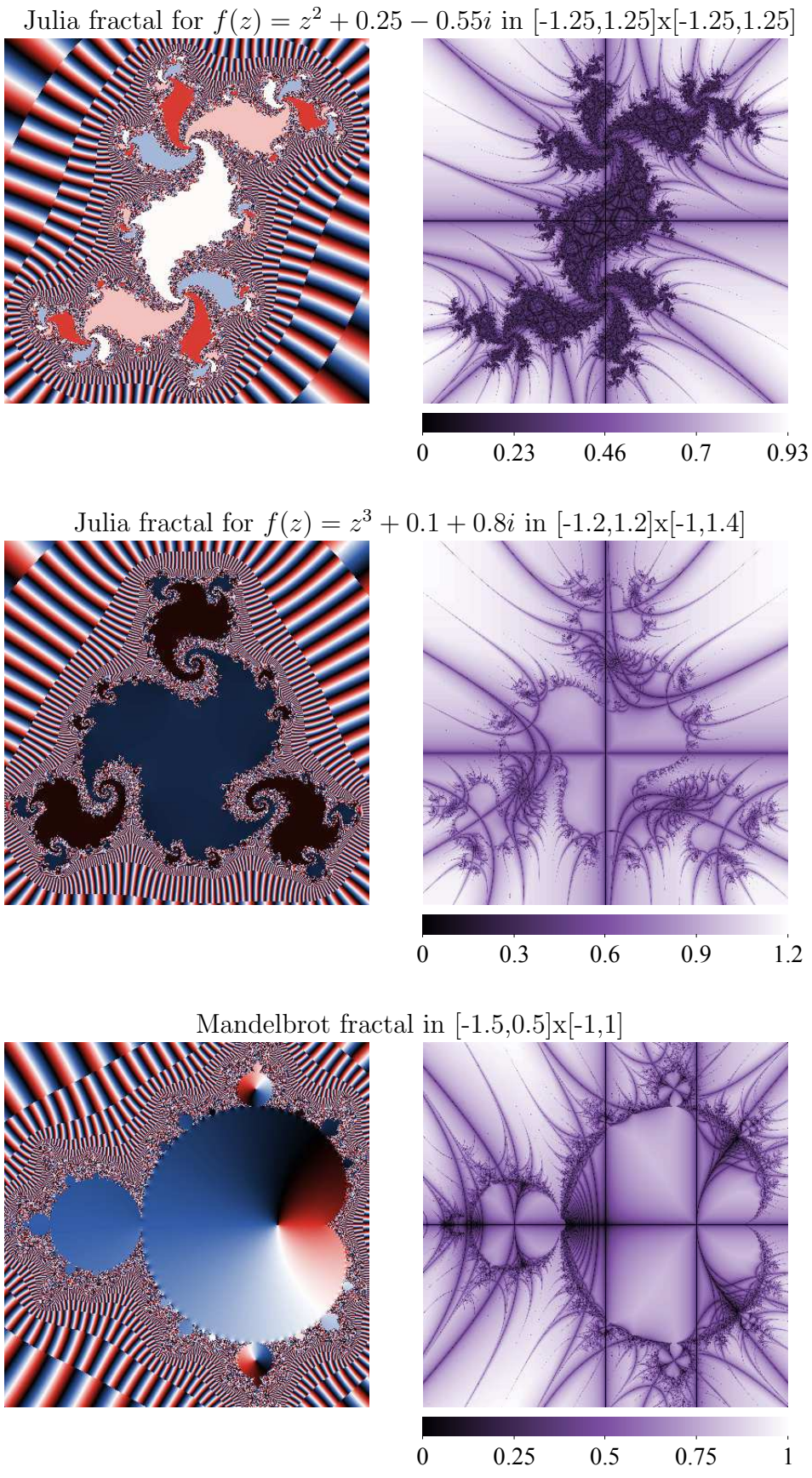



Figure 4: Examples of fractal visualization by means of a *phase plot* (left column) and a *biomorphs plot* (right column). A 401x401 grid is used in discretized space and shown at iteration $i_{\max} = 100$.

**Algorithm 1** Visualization of Algebraic Fractals by Minimum Modulus

**Input:**
$f(z)$, complex function whose iterative application generates a fractal
$D \subset \mathbb{C}$, complex domain of $f(z)$ which is discretized into a series of grid points
$i_{\max}$, maximum number of iterations
**Output:**
A visualization of the fractal in $D$ by minimum modulus

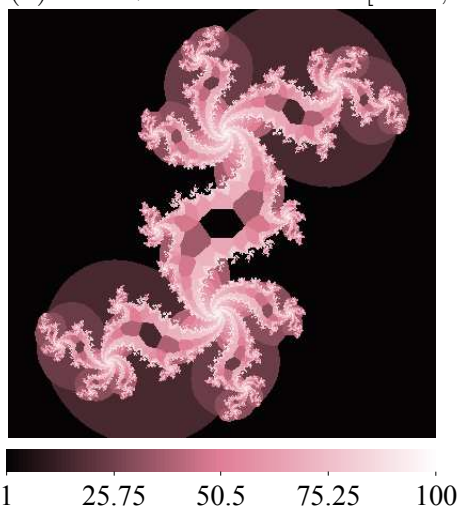
$i \leftarrow 0$; // $i$: current iteration
**for** $z \in D$
  $z.currentIteratedComplexNumber \leftarrow z$;
  $z.minimumModulusSoFar \leftarrow |z|$;
  $z.minimumModulusSoFarIteration \leftarrow i$;
**end-for**
**do**
  $i \leftarrow i + 1$;
  **for** $z \in D$
    $z.currentIteratedComplexNumber \leftarrow f(z.currentIteratedComplexNumber)$;
    **if** $|z.currentIteratedComplexNumber| < z.minimumModulusSoFar$
      $z.minimumModulusSoFar \leftarrow |z.currentIteratedComplexNumber|$;
      $z.minimumModulusSoFarIteration \leftarrow i$;
    **end-if**
  **end-for**
**until** $i = i_{\max}$
$i_{\text{first}} \leftarrow \min_{z \in D}\{z.minimumModulusSoFarIteration\}$;
$i_{\text{last}} \leftarrow \max_{z \in D}\{z.minimumModulusSoFarIteration\}$;
**for** $z \in D$
  Color $z$'s grid point by using a shade: $\frac{z.minimumModulusSoFarIteration - i_{\text{first}}}{i_{\text{last}} - i_{\text{first}}} \in [0, 1]$.
**end-for**

As a result, unlike in the classical method, prisoner points receive a wide range of color shades in the new method and give rise to interesting and diverse patterns. Figure 5, where a logarithmic scale is used to intensify the details, shows the examples in Figure 2 visualized through the new method.

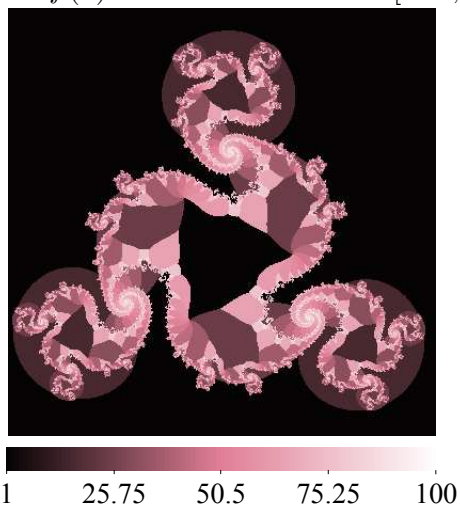
Whereas the biomorphs method selects the complex number in the sequence with minimum distance to the axes, the new method selects the complex number in the sequence with minimum modulus. Since these two complex numbers can be quite different from each other, the shade provided by the new method will be in general quite different from that provided by the biomorphs method. Note also that the classical method and the new method are opposite in nature: Whereas the classical method analyzes divergence of the complex numbers in the sequence (maximum modulus), the new method studies closeness to the origin (minimum modulus).

To end this section, an example of minimum modulus visualization is illustrated iteration by iteration. Thus, Figure 6 contains the first twelve iterations when the method is applied to the complex function $f(z) = z^2 + 0.25 - 0.55i$ in [-1.25,1.25]x[-1.25,1.25], whose final visualization at iteration 100 was illustrated at the top of Figure 5. From the twelve pictures in Figure 6, note that the minimum modulus method seems to operate like a geometric fractal, although it is difficult to define exactly what the initiator and generator sets really are in this case.

Julia fractal for $f(z) = z^2 + 0.25 - 0.55i$ in [-1.25,1.25]x[-1.25,1.25]




Julia fractal for $f(z) = z^3 + 0.1 + 0.8i$ in [-1.2,1.2]x[-1,1.4]




Mandelbrot fractal in [-1.5,0.5]x[-1,1]

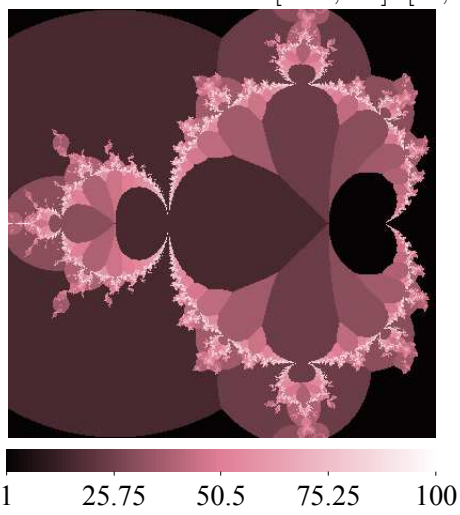



Figure 5: Examples of fractal visualization by means of the *minimum modulus* method. A 401x401 grid is used in discretized space and shown at iteration $i_{\max} = 100$.

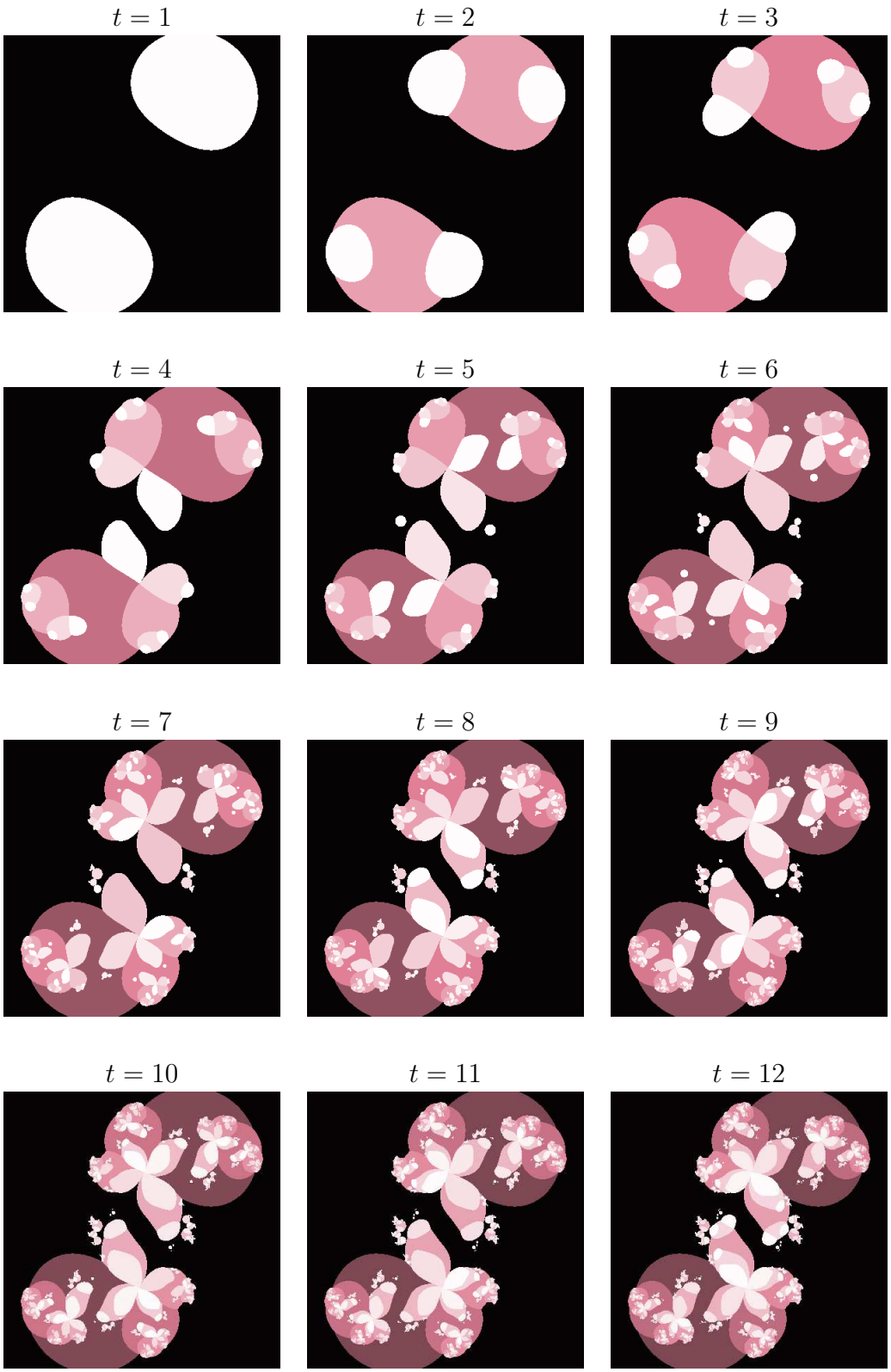



Figure 6: First twelve iterations for the minimum modulus visualization of the function $f(z) = z^2 + 0.25 - 0.55i$ in [-1.25,1.25]x[-1.25,1.25], whose visualization at iteration 100 is shown at the top of Figure 5. A 401x401 grid is used in discretized space.

# 4 Comparative evaluation

In the present section, the novel visualization method for algebraic fractals introduced in Section 3 is comparatively evaluated with the three relevant methods described in Section 2.2: (1) the classical method based on the iteration at which the sequence escapes to infinity, (2) the phase plot method, and (3) the biomorphs method by Pickover based on the minimum distance to the complex plane axes. The evaluated algorithms are implemented within NetLogo[1] [36], an agent-based programming environment well suited for modeling, inspecting and visualizing complex systems developing over time.

## 4.1 Visualization results

Figures 7 through 17 contain several visualization examples of fractals corresponding to different complex functions and domains. Each figure shows the visualization of the four considered methods (`classical`, `phase`, `biomorphs`, and `minimum-modulus`) for a specific complex function and domain in the complex plane.

Some of the examples included in Figures 7-17 correspond to fractals generated from higher-order polynomials, since this kind of relevant fractals have been extensively studied in the literature. For instance, cubic polynomials are addressed in [2, 3, 11], quartic polynomials are analyzed in [11], and more general polynomial functions for fractal generation are studied in [27, 8, 14, 32, 6, 4, 34, 30, 1].

The following important consequences can be derived from Figures 7-17:

1. The novel visualization method preserves the fractal view that the classical method offers of the escape set boundary in all the examples. Therefore, since this boundary is the most relevant region under the classical method, the visualization quality offered by the classical method is not reduced by the novel method.

2. Remarkable aesthetic visual details of the prisoner set (the complement of the escape set) are provided by the novel visualization method in all the examples. Obviously, these details are missing under the classical visualization method. In the examples where the prisoner set occupies most of the complex domain studied and, consequently, the classical method generates almost entirely black visualizations (see Figures 13-16), the new method produces instead interesting details of this set.

3. In general, the biomorphs method offers interesting details of the prisoner set in all the figures. However, in Figures 14 and 15 the results are very poor. Even for these cases, the novel method produces images with rich details.

4. With respect to the phase plot method, the novel method offers in general much more elaborate and aesthetic details of the prisoner sets.

5. Finally, the novel method generates attractive visualizations which are captivating by its aesthetic richness.

[1]The NetLogo model used in this work can be downloaded from `http://www.ia.uned.es/~seve/PDFs/minimum-modulus.zip`

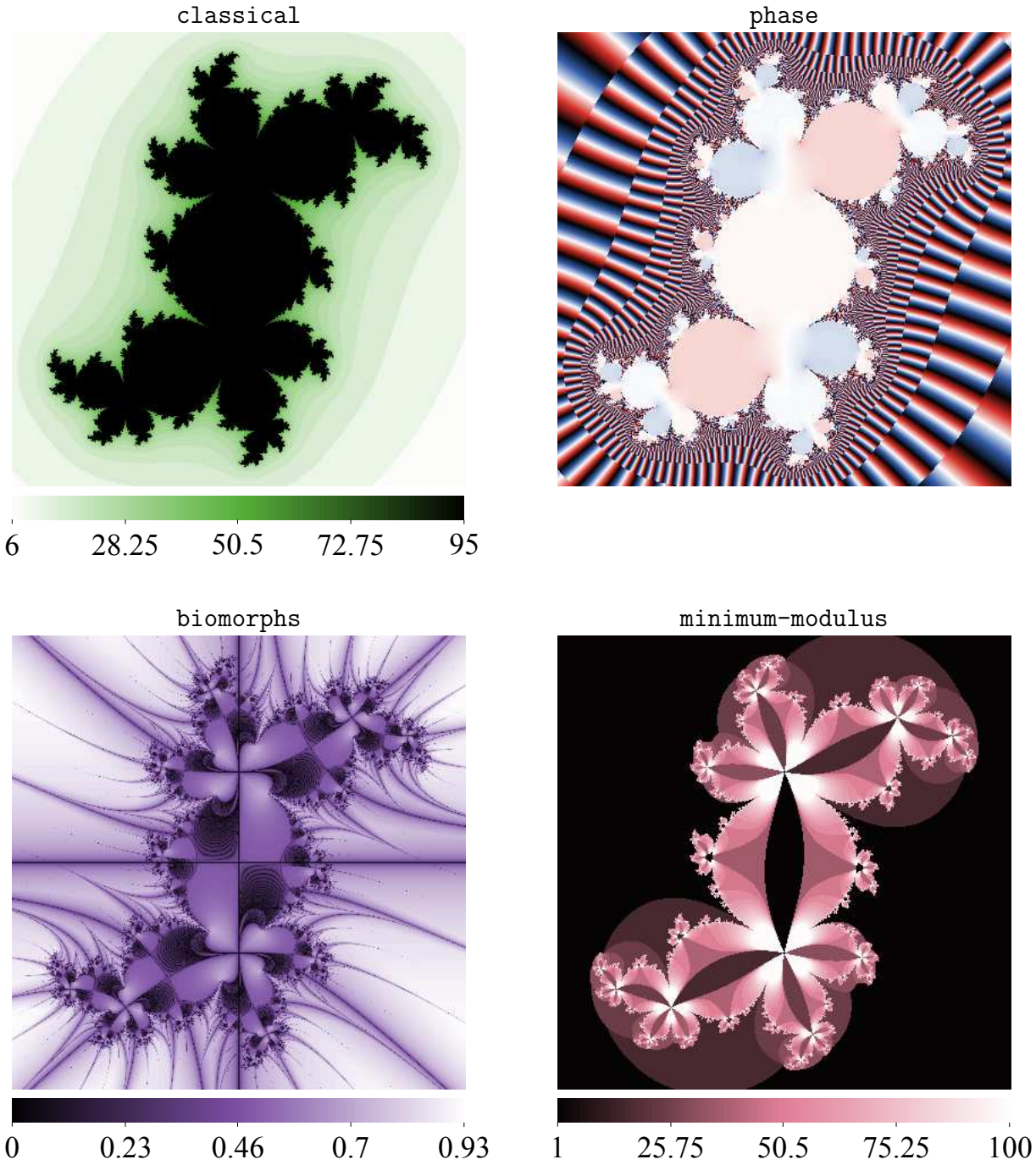



Figure 7: Example of fractal visualization through four different methods. This algebraic fractal corresponds to the complex function $f(z) = z^2 + 0.25 - 0.5i$ over the domain [-1.25,1.25]x[-1.25,1.25]. A 401x401 grid is used in discretized space and shown at iteration $i_{\max} = 100$.

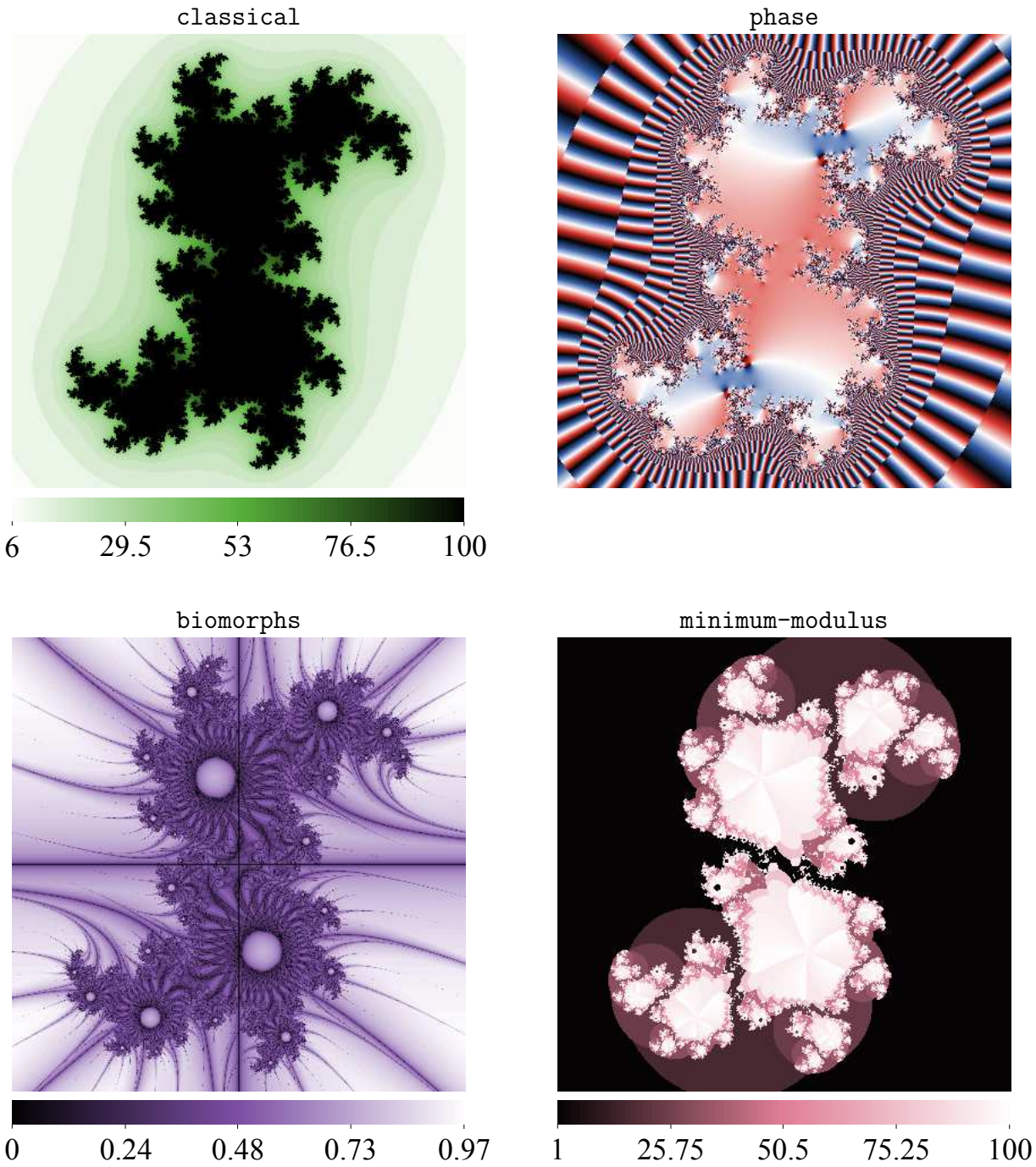



Figure 8: Example of fractal visualization through four different methods. This algebraic fractal corresponds to the complex function $f(z) = z^2 + 0.35 - 0.35i$ over the domain [-1.25,1.25]x[-1.25,1.25]. A 401x401 grid is used in discretized space and shown at iteration $i_{\max} = 100$.

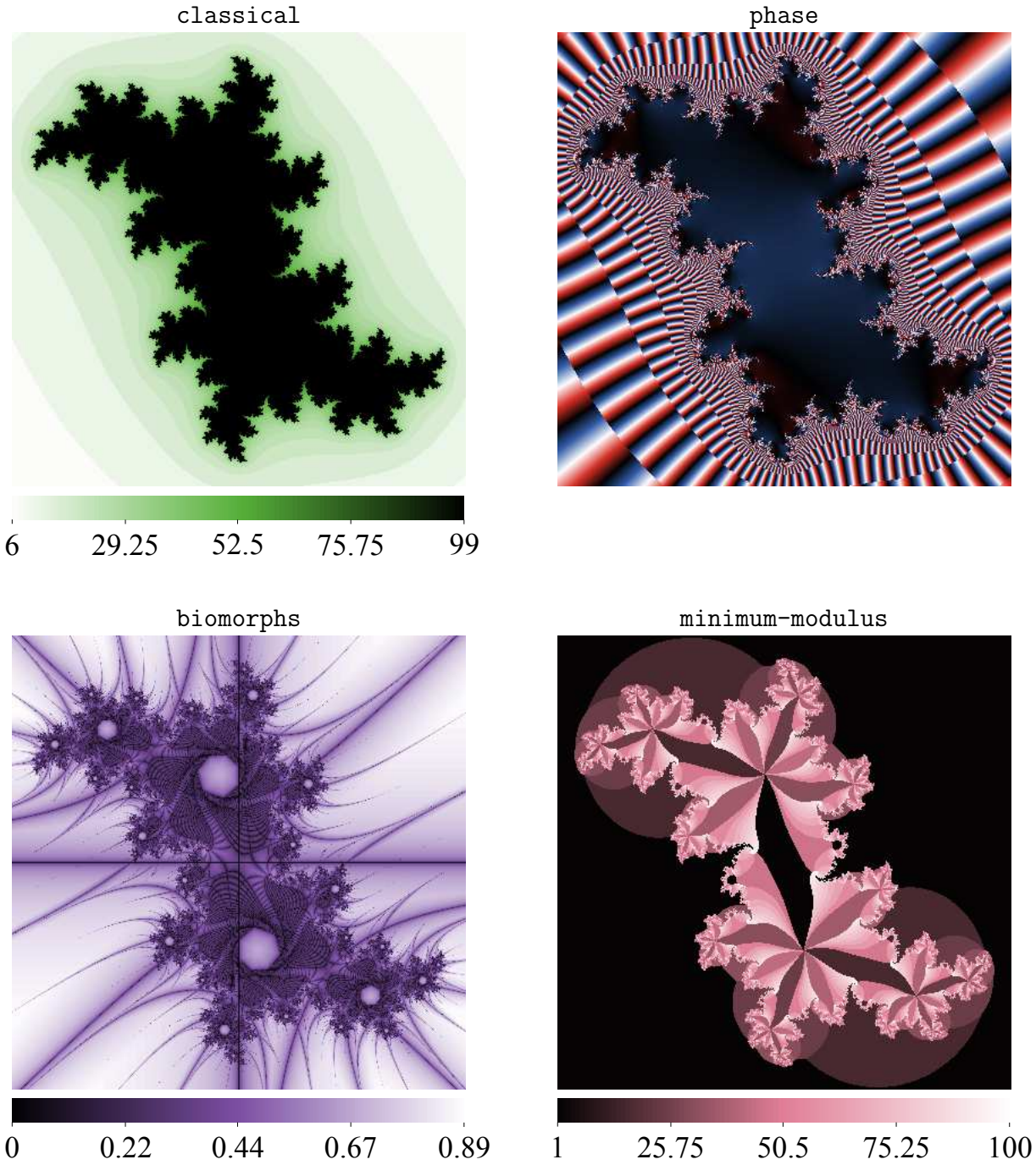



Figure 9: Example of fractal visualization through four different methods. This algebraic fractal corresponds to the complex function $f(z) = z^2 + 0.12 + 0.59i$ over the domain [-1.25,1.25]x[-1.25,1.25]. A 401x401 grid is used in discretized space and shown at iteration $i_{\max} = 100$.

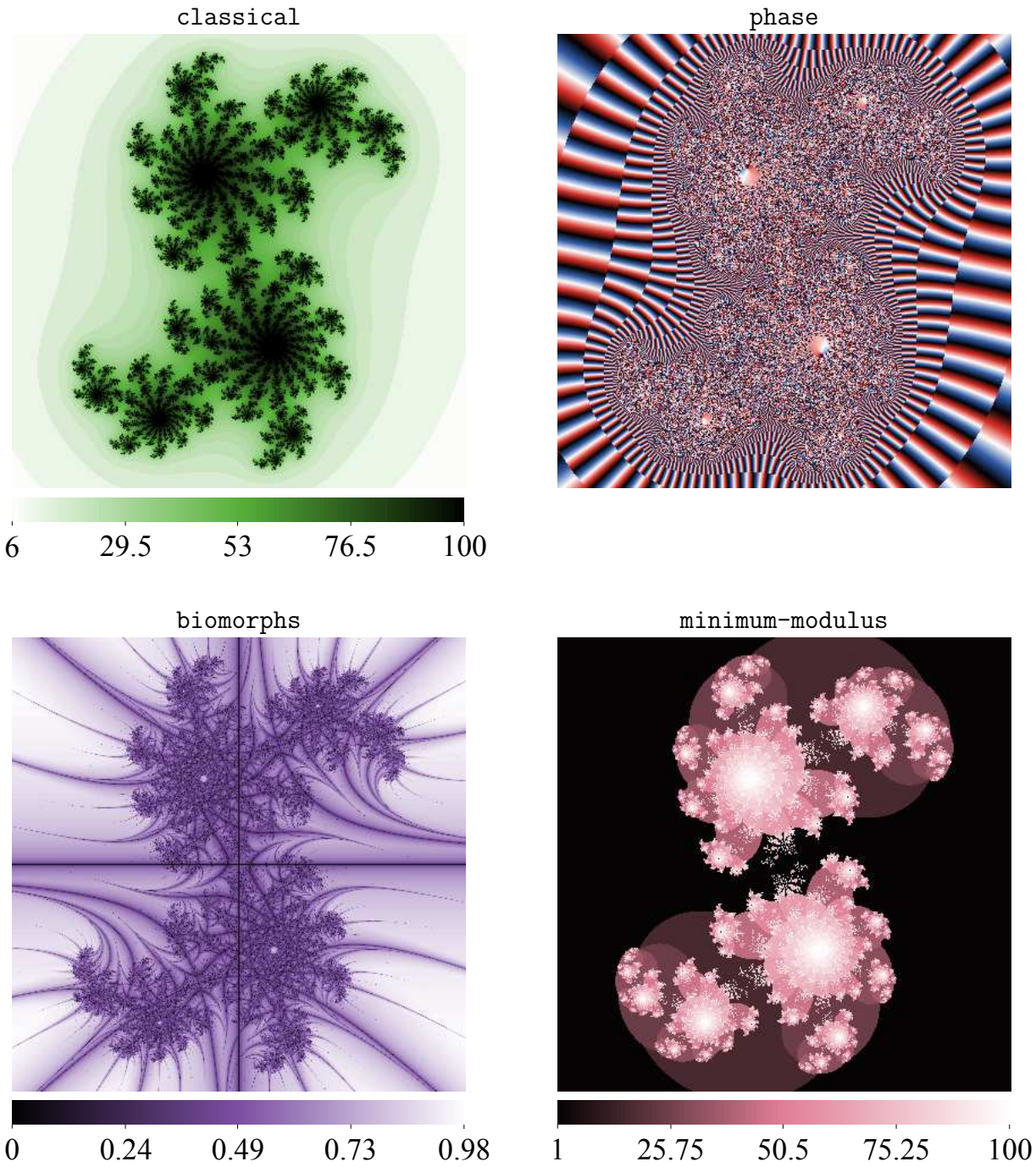



Figure 10: Example of fractal visualization through four different methods. This algebraic fractal corresponds to the complex function $f(z) = z^2 + 0.38 - 0.29i$ over the domain [-1.25,1.25]x[-1.25,1.25]. A 401x401 grid is used in discretized space and shown at iteration $i_{\max} = 100$.

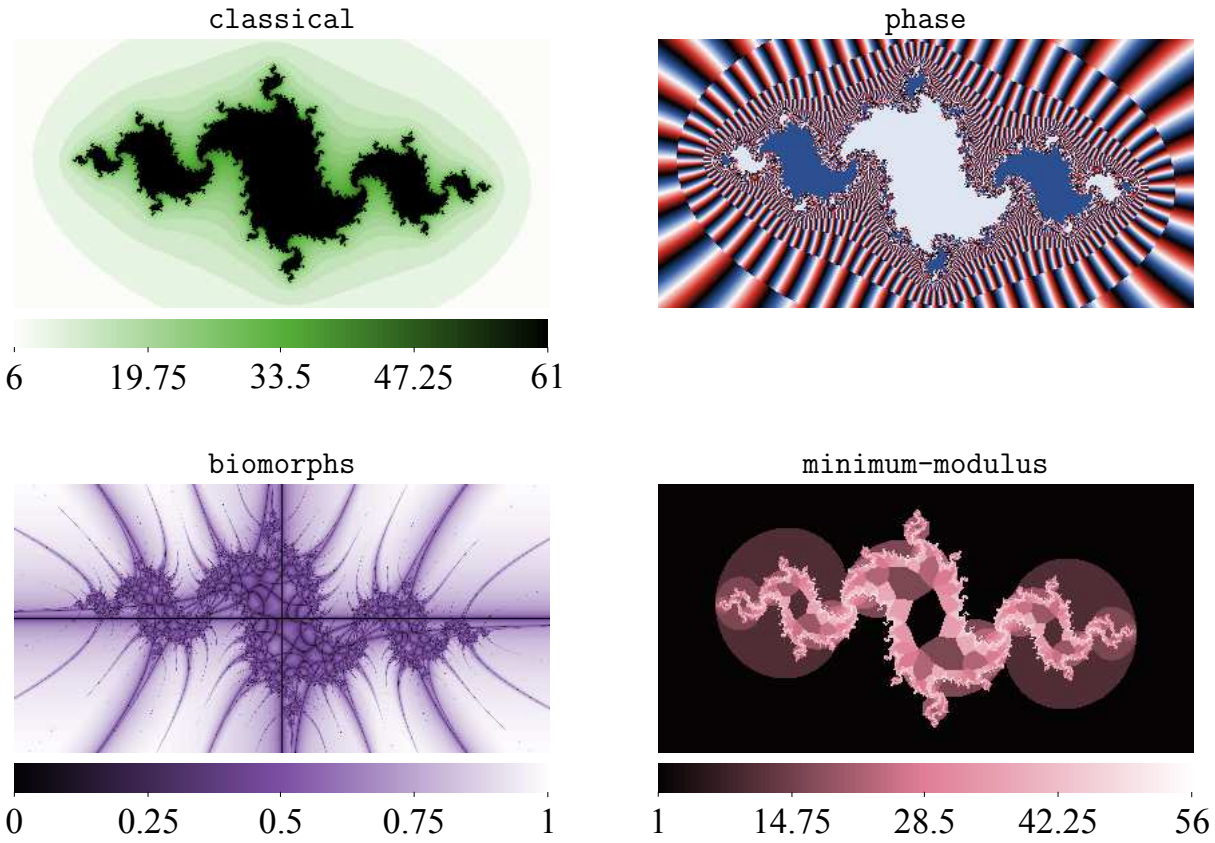



Figure 11: Example of fractal visualization through four different methods. This algebraic fractal corresponds to the complex function $f(z) = z^2 - 0.9 + 0.2i$ over the domain [-2,2]x[-1,1]. A 401x201 grid is used in discretized space and shown at iteration $i_{\max} = 100$.

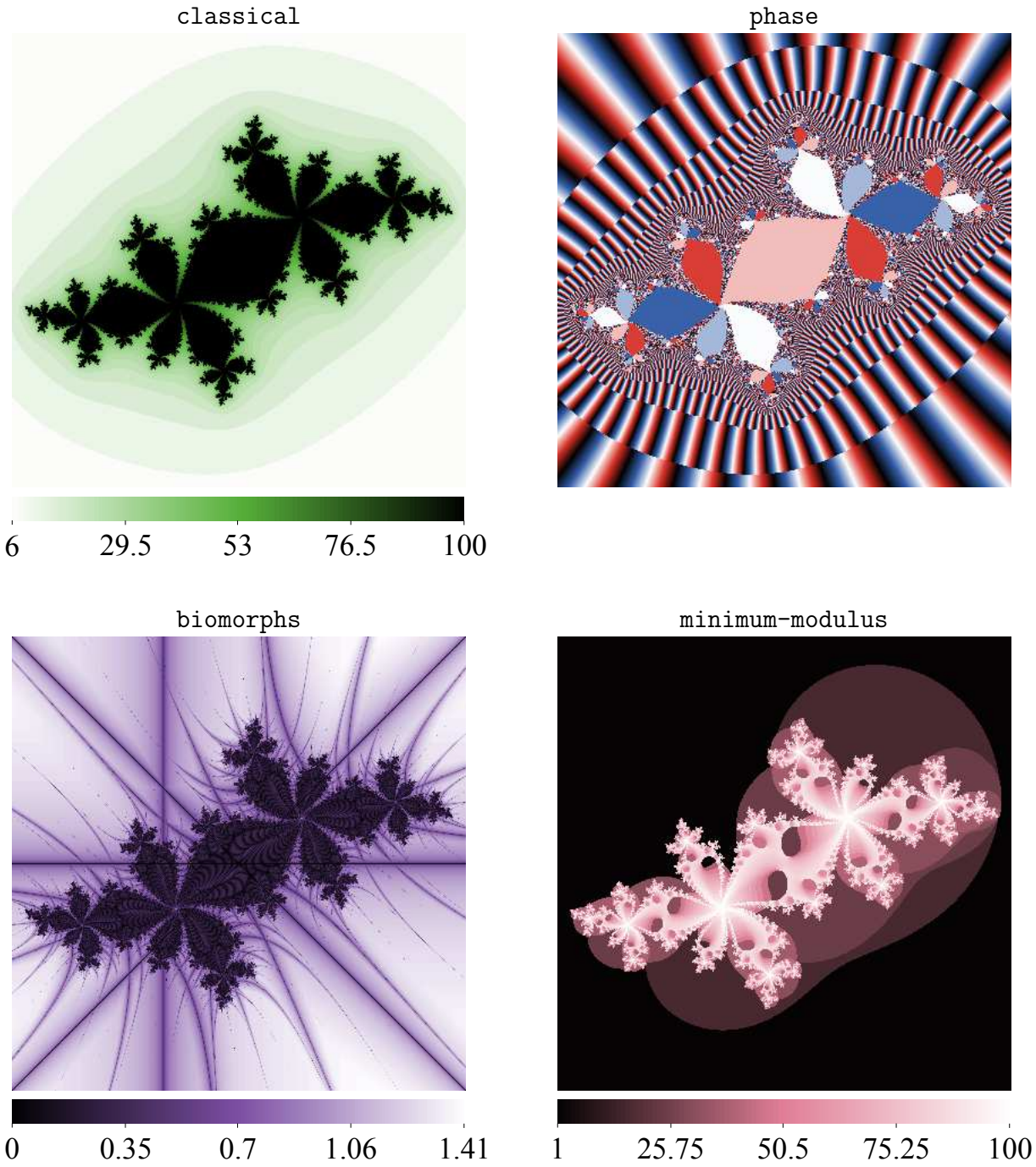



Figure 12: Example of fractal visualization through four different methods. This algebraic fractal corresponds to the complex function $f(z) = z^2 - z + 0.25 - 0.55i$ over the domain [-1,2]x[-1.5,1.5]. A 401x401 grid is used in discretized space and shown at iteration $i_{\max} = 100$.

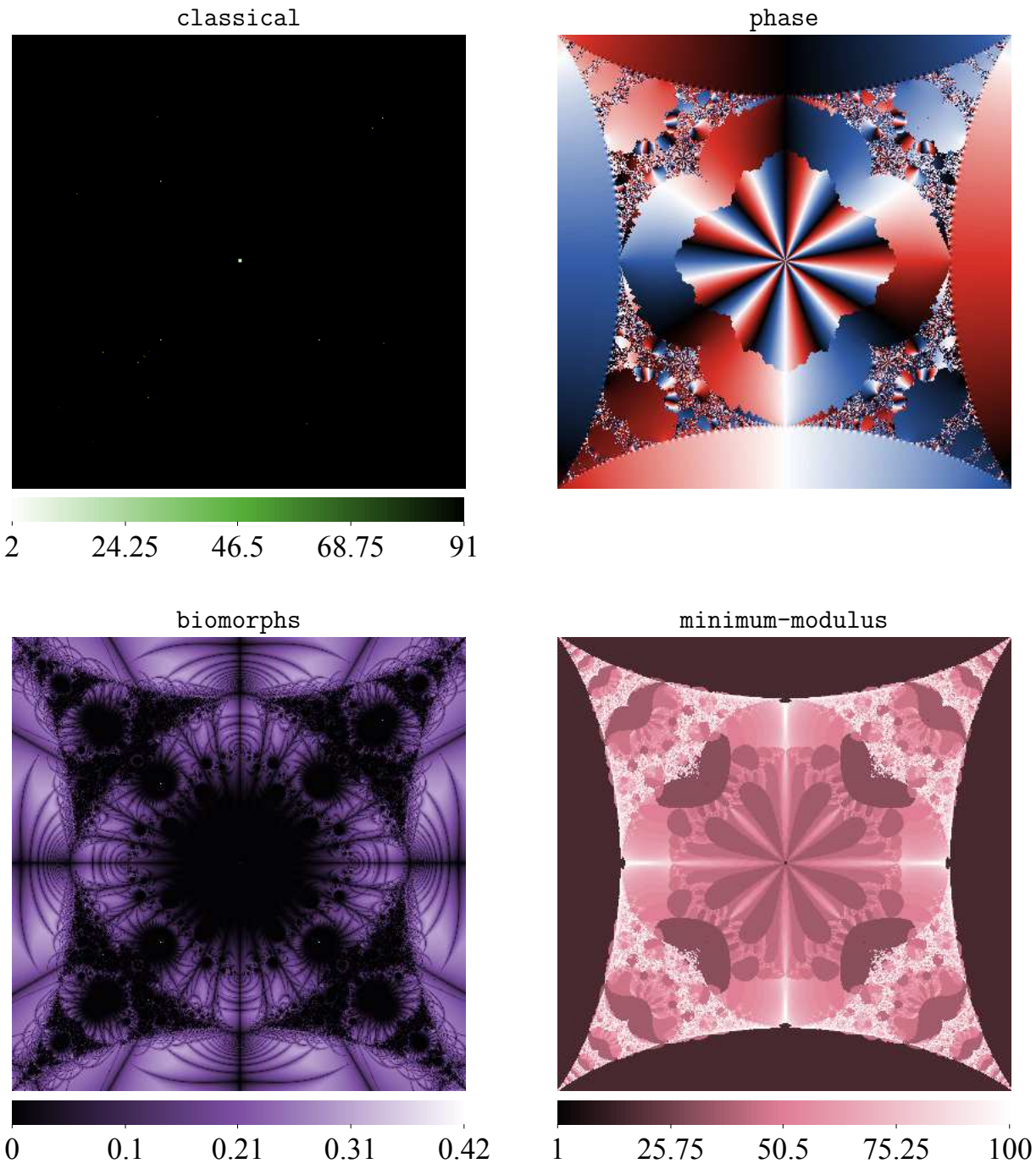



Figure 13: Example of fractal visualization through four different methods. This algebraic fractal corresponds to the complex function $f(z) = \frac{1}{(z+z_0)^3}$ over the domain [-1.21,1.2]x[-1.21,1.2], where $z_0 \in \mathbb{C}$ is the complex number to be colored. A 401x401 grid is used in discretized space and shown at iteration $i_{\max} = 100$.

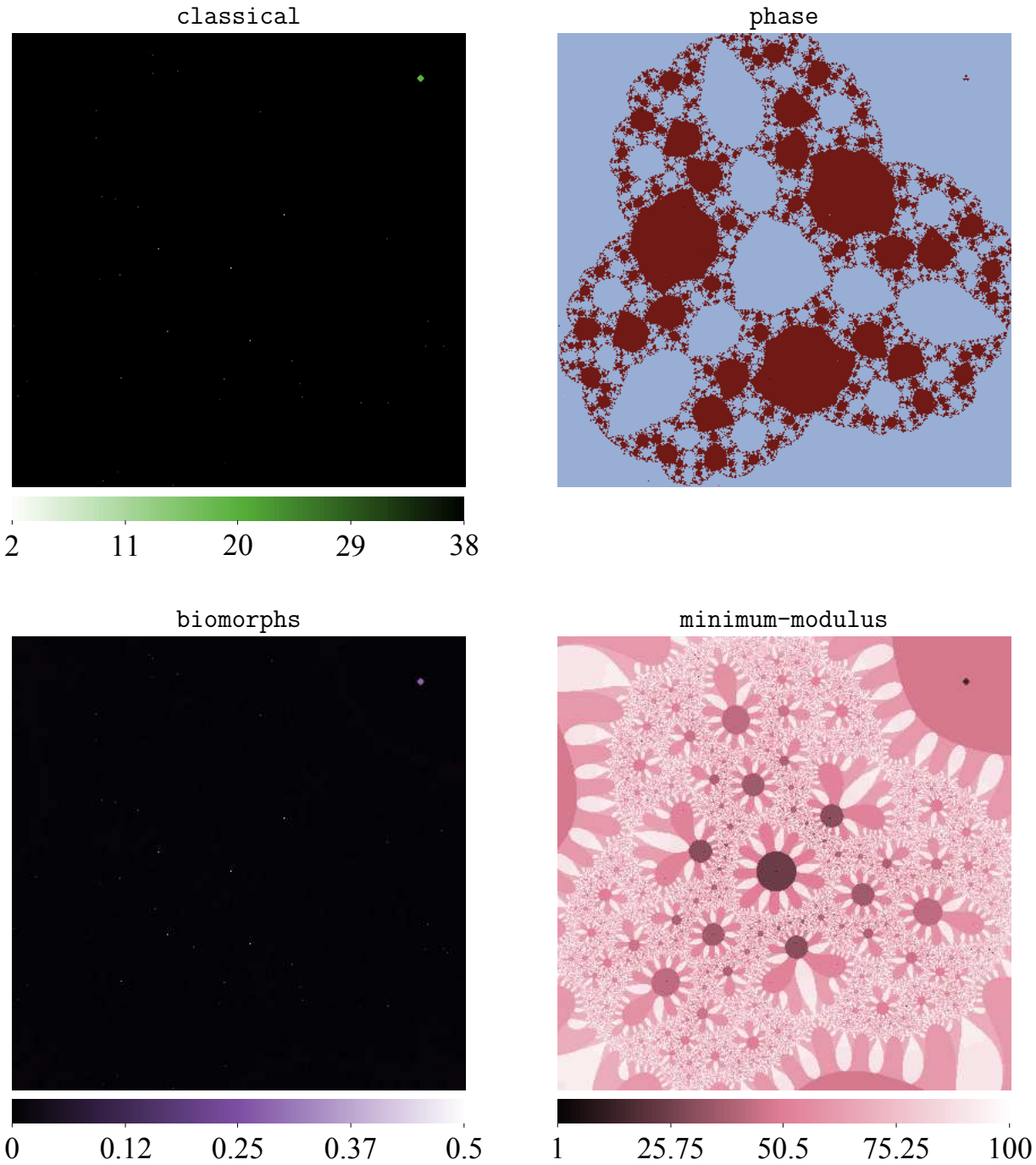



Figure 14: Example of fractal visualization through four different methods. This algebraic fractal corresponds to the complex function $f(z) = \frac{1}{(z+0.5+0.5i)^3}$ over the domain [-3.76,3]x[-3.76,3]. A 401x401 grid is used in discretized space and shown at iteration $i_{\max} = 100$.

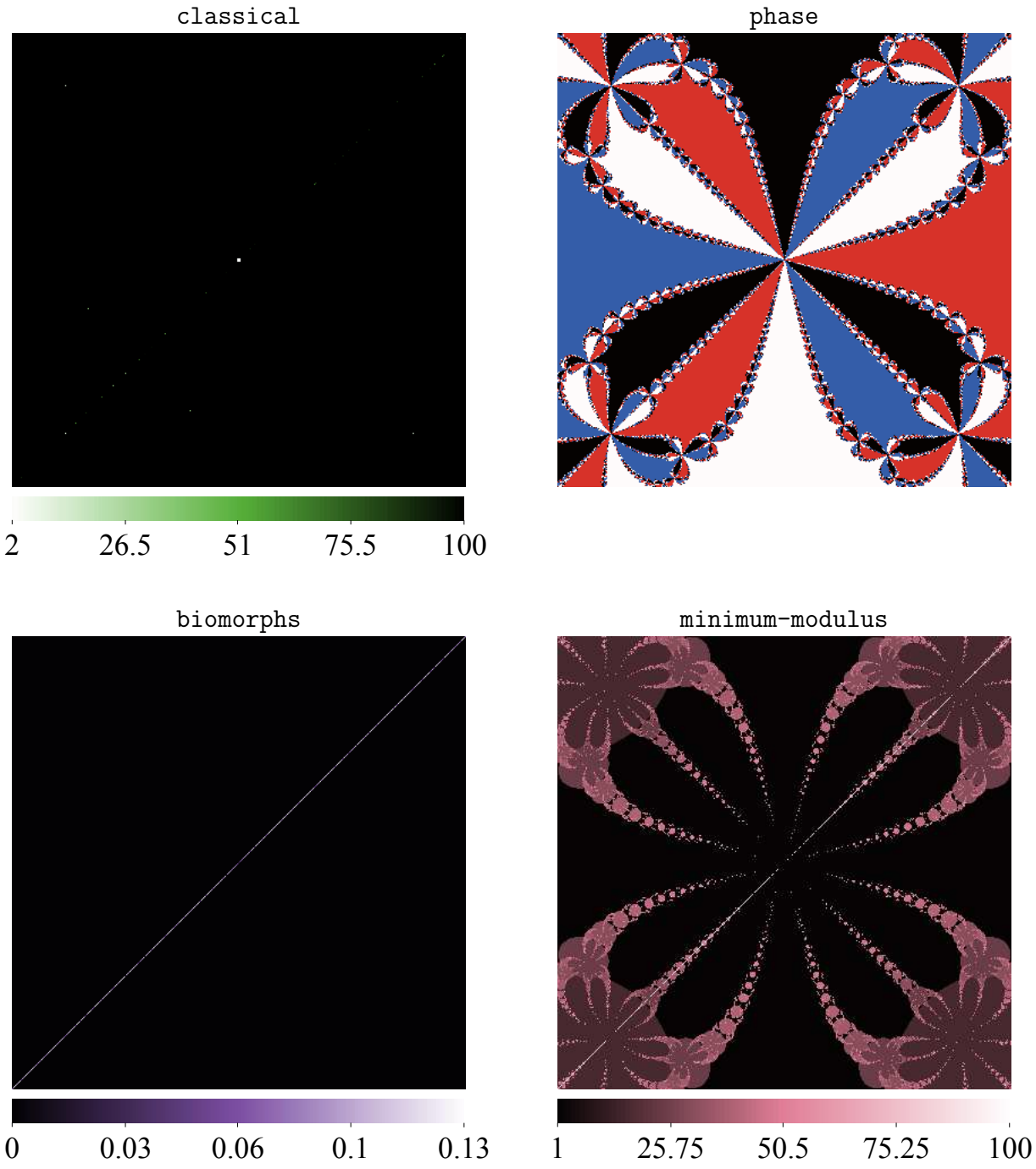



Figure 15: Example of fractal visualization through four different methods. This Newtonian algebraic fractal [7] corresponds to the complex function $f(z) = z - \frac{z^4-1}{4z^3}$ over the domain [-0.701,0.7]x[-0.701,0.7]. A 401x401 grid is used in discretized space and shown at iteration $i_{\max} = 100$.

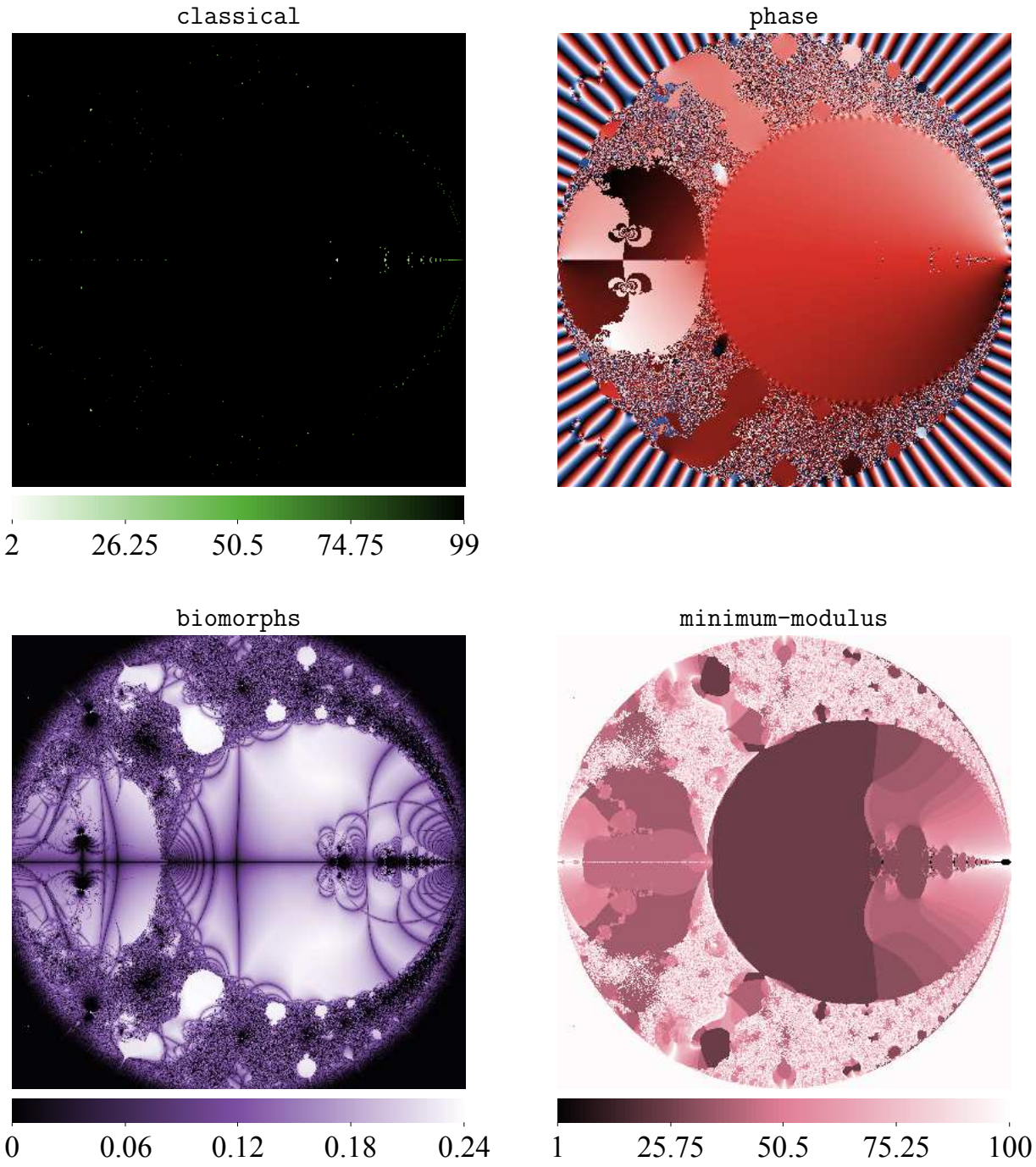



Figure 16: Example of fractal visualization through four different methods. This algebraic fractal corresponds to the complex function $f(z) = \frac{z}{e^z + z_0}$ over the domain [-2,0]x[-1,1], where $z_0 \in \mathbb{C}$ is the complex number to be colored. A 401x401 grid is used in discretized space and shown at iteration $i_{\max} = 100$.

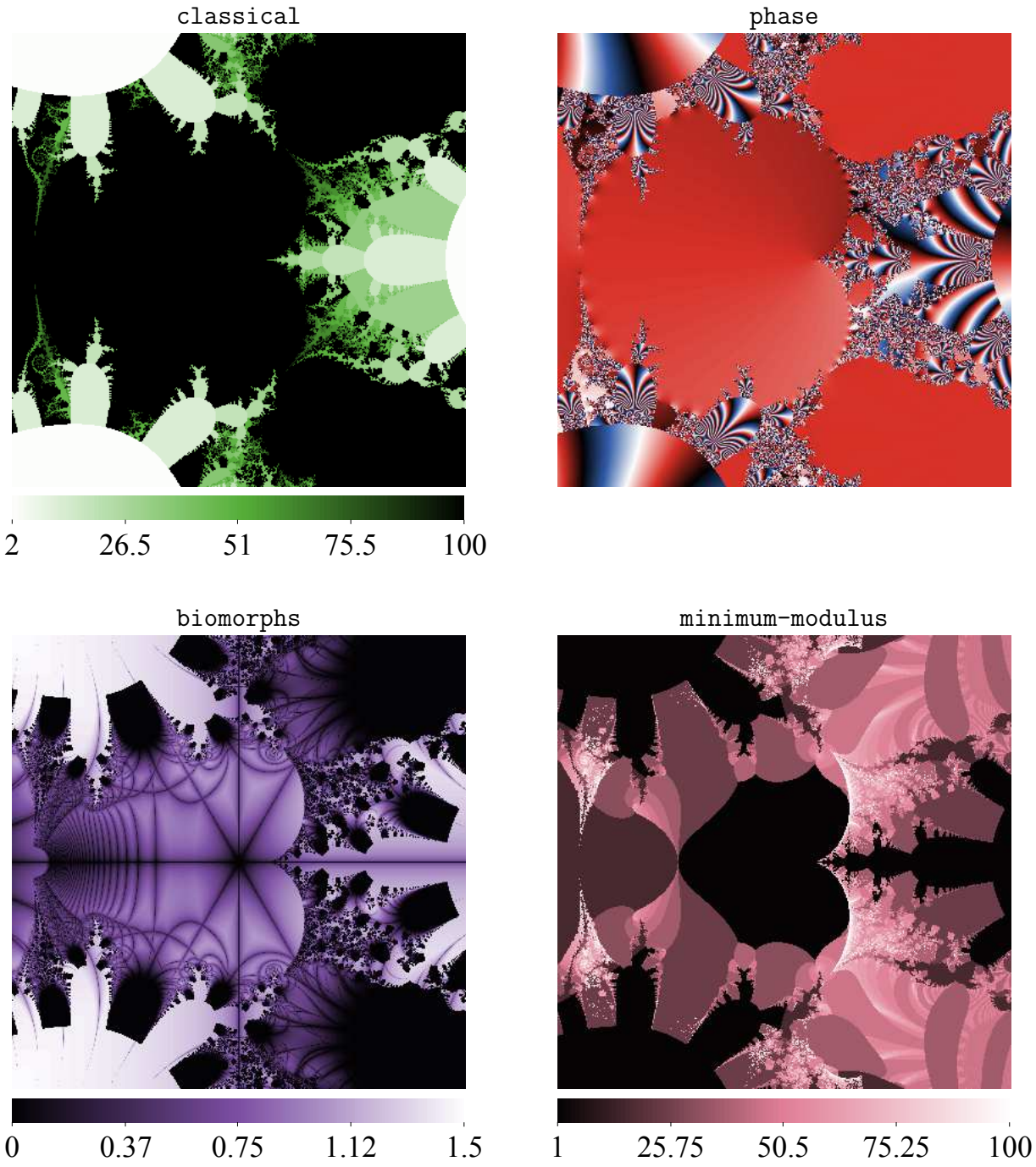



Figure 17: Example of fractal visualization through four different methods. This algebraic fractal corresponds to the complex function $f(z) = e^{z_0 \cdot z^2}$ over the domain [-1.5,1.5]x[-1.5,1.5], where $z_0 \in \mathbb{C}$ is the complex number to be colored. A 401x401 grid is used in discretized space and shown at iteration $i_{\max} = 100$.

### 4.2 Computational complexity

The time complexity of the four analyzed visualization methods is dominated by two variables (see Algorithm 1): $i_{\max}$, the maximum number of iterations that the function $f(z)$ is applied within each sequence, and $\text{card}(D)$, the cardinality or number of elements of the discretized complex domain $D$ where $f(z)$ is defined. Since each of the four visualization methods only needs to perform a few elementary operations in order to color $z_0 \in D$ each time a new complex number is generated within its sequence, the time complexity of the four methods is $\mathcal{O}(i_{\max} \cdot \text{card}(D))$. Specifically, the mentioned elementary operations have to do with calculating the modulus of a complex number and comparing it with the threshold (in the classical method), calculating the phase of a complex number (in the phase plot method), calculating the minimum distance of a complex number to the real or imaginary axes and comparing it with the minimum distance found so far in the sequence (in the biomorphs method), or calculating the modulus of a complex number and comparing it with the minimum modulus found so far in the sequence (in the minimum-modulus method).

Likewise, the space complexity of the four studied visualization methods is $\mathcal{O}(\text{card}(D))$, since a new complex number in the sequence needs to be stored for every $z \in D$ at each iteration.

In summary, the computational complexity is the same for all the analyzed visualization methods. Even if different elementary operations are executed depending on the method every time a grid point needs to be colored, these elementary operations are irrelevant regarding complexity calculation.

## 5 Conclusion and future research

Fractal geometry constitutes a field that has acquired great relevance in the last few decades. Fractals are ubiquitous in nature and can also be generated from iterated complex functions.

This work introduces a new visualization method for algebraic fractals generated in the complex plane. Unlike the classical visualization method, which considers the escape set to infinity, the new method deals with the minimum-modulus complex number in the sequence originated from each point in the complex plane. This allows the prisoner set to be colored in aesthetic ways, giving rise to interesting new shapes. Thus, one of the strengths of the minimum-modulus method is that it offers a unique and useful alternative to how existing methods tackle the fractal visualization problem.

As a future research direction, new complex functions could be studied in order to discover additional shapes generated by the novel visualization method. Another idea worth analyzing is the application of alternative iteration techniques to the Picard iteration used in this work (see Section 2.2). Finally, it could also be interesting to consider the second lowest value of the modulus in the sequence (instead of the minimum value) in order to determine the iteration for coloring.

## References


[1] C. F. Babbs. New fractals for computer generated art created by iteration of polynomial functions of a complex variable. Faculty Working Paper 16, Weldon School of Biomedical Engineering, Purdue University, 2017.

[2] B. Branner and J. H. Hubbard. The iteration of cubic polynomials. Part I: The global topology of parameter space. *Acta Mathematica*, 160:143–206, 1988.

[3] B. Branner and J. H. Hubbard. The iteration of cubic polynomials. Part II: Patterns and parapatterns. *Acta Mathematica*, 169:229–325, 1992.

[4] J. Cheng and J.-R. Tan. Generalization of 3D Mandelbrot and Julia sets. *Journal of Zhejiang University SCIENCE A*, 8(1):134–141, 2007.

[5] R. L. Devaney. *A First Course in Chaotic Dynamical Systems: Theory and Experiment*. CRC Press, 2020. 2nd Edition.

[6] R. L. Devaney and D. M. Look. Buried Sierpinski curve Julia sets. *Discrete and Continuous Dinamical Systems*, 13(4):1035–1046, 2005.

[7] M. Drexler, I. J. Sobey, and C. Bracher. On the fractal characteristics of a stabilised Newton method. Technical Report No. 95/26, Numerical Analysis Group, Oxford University, 1995.

[8] I. D. Entwistle. Julia set art and fractals in the complex plane. *Computers & Graphics*, 13(3):389–392, 1989.

[9] K. Falconer. *Fractal Geometry: Mathematical Foundations and Applications*. Wiley, 2014. Third Edition.

[10] D. P. Feldman. *Chaos and Fractals: An Elementary Introduction*. Oxford University Press, 2012.

[11] M. Frame and J. Robertson. A generalized Mandelbrot set and the role of critical points. *Computers & Graphics*, 16(1):35–40, 1992.

[12] M. Frame and A. Urry. *Fractal Worlds: Grown, Built, and Imagined*. Yale University Press, 2016.

[13] M. Green. The Buddhabrot Technique. `https://superliminal.com/fractals/bbrot/`, 1993.

[14] U. G. Gujar and V. C. Bhavsar. Fractals from $z \leftarrow z^{\alpha} + c$ in the complex $c$-plane. *Computers & Graphics*, 15(3):441–449, 1991.

[15] S. Ishikawa. Fixed points by a new iteration method. *Proceedings of the American Mathematical Society*, 44(1):147–150, 1974.

[16] L. O. Jolaoso and S. H. Khan. Some escape time results for general complex polynomials and biomorphs generation by a new iteration process. *Mathematics*, 8(12):2172, 2020.

[17] G. Julia. Mémoire sur l'iteration des fonctions rationnelles. *Journal de Mathématiques Pures et Apliquées*, 8(1):47–246, 1918.

[18] S. Kumari, K. Gdawiec, A. Nandal, M. Postolache, and R. Chugh. A novel approach to generate Mandelbrot sets, Julia sets and biomorphs via viscosity approximation method. *Chaos, Solitons & Fractals*, 163:1112540, 2022.

[19] B. B. Mandelbrot. *Les Objets Fractals: Forme, Hasard et Dimension*. Flammarion, 1975.

[20] B. B. Mandelbrot. *Fractals: Form, Chance and Dimension*. W. H. Freeman and Company, 1977.

[21] B. B. Mandelbrot. *The Fractal Geometry of Nature*. W. H. Freeman and Company, 1983.

[22] W. R. Mann. Mean value methods in iteration. *Proceedings of the American Mathematical Society*, 4(3):506–510, 1953.

[23] M. A. Noor. New approximation schemes for general variational inequalities. *Journal of Mathematical Analysis and Applications*, 251(1):217–229, 2000.

[24] H.-O. Peitgen, H Jürgens, and D. Saupe. *Chaos and Fractals: New Frontiers of Science*. Springer, 2004. Second Edition.

[25] C. A. Pickover. Biomorphs: computer displays of biological forms generated from mathematical feedback loops. *Computer Graphics Forum*, 5(4):313–316, 1986.

[26] C. A. Pickover. Accident, evolution, and art. *YLEN NEWSLETTER*, 12(19):3–5, 1999.

[27] C. A. Pickover and E. K. Horasani. Computer graphics generated from the iteration of algebraic transformations in the complex plane. *Computers & Graphics*, 9(2):147–151, 1985.

[28] D. J. Prajapati, S. Rawat, A. Tomar, M. Sajid, and R. C. Dimri. A brief study on Julia sets in the dynamics of entire transcendental function using Mann iterative scheme. *Fractal and Fractional*, 6(7):397, 2022.

[29] P. Prusinkiewicz and A. Lindenmayer. *The Algorithmic Beauty of Plants*. Springer, 2004.

[30] M. Rani and M. Kumar. Circular saw Mandelbrot set. In *Proceedings of the 14th WSEAS International Conference on Applied Mathematics (MATH'09)*, pages 131–136, 2009.

[31] A. A. Shadid, W. Nazeer, and K. Gdawiec. The Picard-Mann iteration with $s$-convexity in the generation of Mandelbrot and Julia sets. *Monatshefte für Mathematik*, 195:565–584, 2021.

[32] K. W. Shirriff. An investigation of fractals generated by $z \to \frac{1}{z^n} + c$. *Computers & Graphics*, 17(5):603–607, 1993.

[33] P. D. Sisson. Fractal art using variations on escape time algorithms in the complex plane. *Journal of Mathematics and the Arts*, 1(1):41–45, 2007.

[34] X.-Y. Wang, P.-J. Chang, and N.-N. Gu. Additive perturbed generalized Mandelbrot-Julia sets. *Applied Mathematics and Computation*, 189(1):754–765, 2007.

[35] E. Wegert and G. Semmler. Phase plots of complex functions: a journey in illustration. *Notices of the American Mathematical Society*, 58(6):768–780, 2011.

[36] U. Wilensky. NetLogo. `http://ccl.northwestern.edu/netlogo/`, 1999. Center for Connected Learning and Computer Science, Northwestern University, Evanston, IL.

[37] X. Zhang, T. Lv, and Z. Wang. A generalized Mandelbrot set based on distance ratio. In *Proceedings of the 14th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG'2006)*, pages 179–184, 2006.

[38] C. Zou, A. A. Shadid, A. Tassaddiq, A. Khan, and M. Ahmad. Mandelbrot sets and Julia sets in Picard-Mann orbit. *IEEE Access*, 8:64411–64421, 2020.