Robust Visualization of Strange Attractors using Affine Arithmetic

Afonso Paiva

Departamento de Matemática, PUC-Rio, Rio de Janeiro, Brazil

Luiz Henrique de Figueiredo

IMPA — Instituto de Matemática Pura e Aplicada, Rio de Janeiro, Brazil

Jorge Stolfi

Instituto de Computação, UNICAMP, Campinas, Brazil

Abstract

We propose the use of affine arithmetic in cell-mapping methods for the robust visualization of strange attractors and show that the resulting cellular approximations converge faster than those produced by cell-mapping methods based on classical interval arithmetic.

Key words: strange attractors, cell mapping, interval arithmetic, affine arithmetic, discrete dynamical systems

Email addresses: apneto@mat.puc-rio.br (Afonso Paiva), lhf@impa.br (Luiz Henrique de Figueiredo), stolfi@ic.unicamp.br (Jorge Stolfi).

1 Introduction

The goal in the study of discrete dynamical systems is to understand the long-term behavior of the iterates of a map $f\colon \mathbf{R}^n \to \mathbf{R}^n$. We are interested in what happens to the *orbit* $\mathcal{O}(p)$ of a point $p \in \mathbf{R}^n$:

$$\mathcal{O}(p) = \{p, f(p), f(f(p)), f(f(f(p))), \ldots\}.$$

Typically, such orbits either diverge to infinity or converge to a manifold in $\mathbf{R}^n$. (A *manifold* is a well-behaved set, such as a point, a set of isolated points, a curve, a surface, etc.) This limit set is called the *attractor* of the dynamical system. Not all attractors are well behaved and in many cases the orbits accumulate on sets that have complicated geometry and topology. Such limit sets are known as *strange attractors* and can exist even for the simplest non-linear maps f. A prime example of this phenomenon is given the famous Hénon map [1], which acts on the plane $\mathbf{R}^2$ as follows:

$$f(x, y) = (1 + y - ax^2, bx),$$

where a and b are parameters. The Hénon strange attractor is obtained by setting $a = 1.4$ and $b = 0.3$; it is shown in Figure 1. (Strictly speaking, it has not been mathematically proved that the Hénon attractor is actually a strange attractor in the technical sense. The mathematics of the Hénon map is very complicated and its dynamics is not yet fully understood [2].)

[Fig. 1 about here.]

Mathematicians usually start their study of a dynamical system by drawing a picture of its orbits. The simplest method for producing such a picture is the point-sampling method discussed in Section 2. However, as also discussed in Section 2, this method is not robust: it depends on trial and error, is subject to rounding errors, and may produce pictures that are not reliable. Other methods for approximating attractors reliably have been proposed, such as the cell-mapping method, which we discuss in Section 3. Although it can be based on point sampling, the cell-mapping method is made robust by using interval arithmetic [3], as described by Michelucci [4]. We propose here the use of affine arithmetic [5] instead of interval arithmetic in cell mapping. In Section 4 we show some examples of the performance of the cell-mapping method based on affine arithmetic for creating robust pictures of strange attractors.

2 Point sampling

The simplest way to "see" the dynamics of a discrete system is to draw a picture of its orbits using the following sampling method:

(1) Guess or somehow find a box Ω containing the attractor.
(2) Select a set of random starting points in Ω.
(3) For each starting point $p \in \Omega$, compute but do not plot the first n_0 points in the orbit $\mathcal{O}(p)$.
(4) Compute and plot the next n_1 points in $\mathcal{O}(p)$.

If the attractor is a manifold, then picture will show a fairly dense approximate sampling of the manifold. If the orbits diverge to infinity, the picture will show nothing (if we do not plot outside Ω). For strange attractors, the picture will show a cloud of dots that clearly has some structure, but this structure is elusive to describe. For instance, detailed pictures of the Hénon attractor suggest that it has Cantor-set cross-sections [1] and thus has a fractal nature.

The point-sampling method is very simple to understand and implement. It generates nice pictures. The main difficulty we face when using this method is how to choose n_0 and n_1. Choosing n_0 too small will include transient parts of the orbits, that is, points that are not yet near the attractor. Choosing n_1 too small will risk not covering the attractor sufficiently well. On the other hand, choosing n_0 or n_1 too large may be wasteful and inefficient. In practice, we just choose n_0 and n_1 by trial and error. However, there is no way to be sure that we have selected good values for n_0 or n_1.

Another difficulty with the point-sampling method is that it is implemented using floating-point arithmetic, which is subject to rounding errors [6]. For chaotic dynamical systems — the ones that have strange attractors — rounding errors are potentially serious, because orbits starting at nearby points can diverge from each other exponentially. Sometimes, this strong sensitivity to initial conditions does not affect the overall picture, because numerically computed orbits are "shadowed" by exact orbits that capture the typical behavior of the system. However, the truth is that rounding errors affect numerical simulations of dynamical systems in very complex ways [7]. Well-conditioned dynamical systems may display chaotic numerical behavior [8, 9]. Conversely, numerical methods can suppress chaos in some chaotic dynamical systems [9].

As a consequence of both difficulties, the pictures generated with the point-sampling method will probably not display the attractor reliably. This is specially serious when we have just started investigating a dynamical system and its attractor is not yet known.

3 Cell mapping

An alternative to point sampling is cell mapping [10, 11]. The main idea in this method is to decompose Ω into cells (typically using a uniform rectangular grid) and study the dynamics induced by f on this set of cells. Instead of asking where each point goes under f, we ask where each cell goes. More precisely, we consider the directed graph having the cells as vertices and having an edge from cell A to cell B if $f(A)$ intersects B. This means that A goes (partially) to B. This graph is called the *cell graph*. The key observation in the cell-mapping method is that the strongly connected components of the cell graph must cover the attractor of f. Cells having no edge into them cannot contain any part of the attractor because f never takes points into those cells. Cells in the same strongly connected component are (partially) mapped into each other by iterates of f. Thus, strongly connected components capture the *transitivity* of the attractor. Figure 2 shows a cell graph for the Hénon map.

[Fig. 2 about here.]

Given a sufficiently fine cell decomposition of Ω, we can find a good approximation of the attractor by finding the strongly connected components of the corresponding cell graph. A more efficient approximation for the attractor can be found by using recursive subdivision: start with a coarse cell decomposition of Ω; find the cell graph induced by f; find its strongly connected components; subdivide the cells in these components into smaller cells; rebuild the cell graph using the smaller cells; find its strongly connected components; and repeat until the cells are small enough. Efficiency comes from not having to start from a very fine cell decomposition; only the cells in the strongly connected components are refined. Figure 3 shows the convergence of this recursive subdivision to the Hénon attractor with classical parameters $a = 1.4$ and $b = 0.3$.

[Fig. 3 about here.]

Finding the strongly connected components of a graph can be done in time linear in the size of the graph, using an elegant algorithm by Tarjan [12]. This leaves as the main difficulty in the cell-mapping method how to find the edges in the cell graph, that is, how to decide which cells $f(A)$ intersects. We call this the *edge problem*. Because f is a non-linear map, there is no simple geometric description for $f(A)$ on which to base an exact intersection test.

The simplest solution for the edge problem is to use point sampling: for each cell A and for each point p in a finite set of samples chosen in A, we identify the cell B that contains $f(p)$ and add the edge $A \to B$ to the cell graph. For best results, we can sample each cell A more finely on its boundary [13]. Again, this solution is very simple to implement and gives good results. However, it is not guaranteed to find the complete cell graph: we may have missed an

edge $A \to B$ simply because no sample point happened to be mapped into B.

A robust solution for the edge problem is to use enclosures: instead of sampling $f(A)$ (via a sampling of A), we compute an *enclosure* for $f(A)$, that is, a set $\bar{f}(A)$ that contains $f(A)$. We aim for an enclosure $\bar{f}(A)$ that is simple to represent and to test for intersection with cells. Computing the cell graph using such enclosures results in a graph that is larger than the exact cell graph, which we cannot compute in general. However, if enclosures shrink as cells shrink, then the refinement provided by cell subdivision will make this larger graph converge to the attractor. In other words, we get a sequence of covers that converge to the attractor.

Interval arithmetic [3] is the natural tool for computing enclosures. It provides box enclosures that shrink as needed. It also naturally handles rounding errors. More precisely, given a real function of n variables $g\colon \mathbf{R}^n \to \mathbf{R}$, interval arithmetic provides an automatic way for computing an interval $\bar{g}(X_1, \ldots, X_n) \subseteq \mathbf{R}$ that contains all the values taken by g in a box $X_1 \times \ldots \times X_n \subseteq \mathbf{R}^n$. Interval arithmetic works by extending all elementary operations and functions from numbers to intervals. For a vector-valued function $f\colon \mathbf{R}^n \to \mathbf{R}^m$, we can combine the intervals corresponding to each coordinate of f and get a box $\bar{f}(X_1, \ldots, X_n) \subseteq \mathbf{R}^n$ that contains all the values taken by f in a box $X_1 \times \ldots \times X_n \subseteq \mathbf{R}^n$. In other words,

$$\bar{f}(X_1, \ldots, X_n) \supseteq f(X_1, \ldots, X_n) = \{f(x_1, \ldots, x_n) : x_i \in X_i\}.$$

By carefully controlling rounding direction to ensure outward rounding, interval arithmetic provides enclosures that are robust even in the presence of rounding errors. Such enclosures are suitable to the cell-mapping method. (Rounding control is mandated by the IEEE 754 standard for floating-point arithmetic and is available in virtually all machines.)

Michelucci [4] described a cell-mapping method based on interval arithmetic and showed that it provides pictures of strange attractors that are closer to reality, because they can discard large transient parts that are hard to identify with point-sampling methods. Figure 4 (which also appears in [4]) shows pictures of the Hénon attractor for $a = 1.4$ and $b = 0.3$, 0.4, and 0.5 using the point-sampling method and the cell-mapping method based on interval arithmetic. While for $b = 0.3$ there is almost no difference between the two pictures, for $b = 0.4$ and $b = 0.5$ we see large gaps in the attractor that were not found by the point-sampling method. Michelucci showed that the difference is even more dramatic for larger values of b.

[Fig. 4 about here.]

Figure 5 shows a significant difference in the results of the two methods for the Holmes map given by

$$f(x, y) = (1.5x - x^3 + 0.95y, x).$$

The point-sampling picture is shown in red and the cell-mapping picture is shown in grey. Again, note the large gaps not found by the point-sampling method.

[Fig. 5 about here.]

Cell-mapping methods based on interval arithmetic can also find pieces of the attractor that are missed by point sampling. Figure 6 shows the attractor of the logistic map given by $f(x, y) = (y, ay(1 - x))$ for $a = 2.27$. The point-sampling method misses the fixed point at (t, t) with $t = 1 - 1/a$, but the cell-mapping method finds it.

[Fig. 6 about here.]

Finally, cell-mapping methods based on interval arithmetic can also be used to study periodic orbits [14].

The main problem with using interval arithmetic in cell mapping is that its enclosures $\bar{f}(A)$ tend to be much larger than the exact result $f(A)$. (This called the *overestimation problem.*) As a consequence, many false edges are created in the cell graph. Because interval arithmetic is limited to computing box enclosures that are aligned with the coordinate axes, it cannot adapt them to the shape and orientation of $f(A)$. (This called the *wrapping problem.*) In other words, even if $\bar{f}(A)$ was the smallest box containing $f(A)$, it would probably still induce many false edges in the cell graph. As a consequence, the strongly connected components of the cell graph are larger than necessary and the cellular approximation converges slowly to the attractor. This shows as "fat" grey regions in Figure 5.

Affine arithmetic [5] is another tool for computing enclosures. It was explicitly designed to reduce overestimation by taking into account first-order correlations in the evaluation of expressions. (Correlations of subexpressions is the main source of overestimation in interval arithmetic.) As a consequence, affine arithmetic often obtains smaller enclosures than interval arithmetic. Moreover, affine arithmetic provides zonotope enclosures that adapt themselves better to the shape and orientation of the underlying set, thus also reducing the wrapping effect. (A *zonotope* is a centrally symmetric convex polytope.) Figure 7 illustrates the difference between the box enclosures computed with interval arithmetic and the zonotope enclosures computed with affine arithmetic, and their influence on the solution of the edge problem. This example uses a quintic map (that is, a polynomial map of degree 5).

[Fig. 7 about here.]

4 Examples

We shall now show some examples of strange attractors in $\mathbf{R}^2$ and their approximation using point sampling, cell mapping with interval arithmetic, and cell mapping with affine arithmetic. We have chosen random polynomial maps of moderate degree to investigate how the correlations implied by long polynomial expressions affect the performance of cell mapping based on interval arithmetic.

We used a method by Sprott [15] for finding polynomial maps that have strange attractors. The region of interest Ω was found with point sampling. (Michelucci [4] avoided guessing Ω by using projective oriented geometry.) Following Sprott, we shall represent polynomial maps $(x', y') = f(x, y)$ with a letter code. For instance, a quadratic map in its most general form is given by

$$x' = a_1 + a_2 x + a_3 x^2 + a_4 xy + a_5 y + a_6 y^2$$
$$y' = a_7 + a_8 x + a_9 x^2 + a_{10} xy + a_{11} y + a_{12} y^2$$

and of course is determined by the values of its twelve coefficients $a_1, \ldots, a_{12}$. Sprott proposed to represent quadratic maps by a 12-letter name, coding these coefficients using the letters A through Y, each letter representing a number in the interval $[-1.2, 1.2]$, with an increment of 0.1. Thus, we have $A = -1.2$, $B = -1.1$, $\ldots$, and $Y = 1.2$. This coding method can be extended to higher-degree maps: a cubic map will have a 20-letter name and a quintic map will have a 42-letter name. Figure 7 used the quintic map named *QBKSKIXQMKEOVVMAHXLBOQQJXEYMBUMBOEFVDBAPWU*.

Figure 8 shows the higher convergence of cellular approximations based on affine arithmetic for that quintic map. Interval arithmetic enclosures shrink linearly whereas affine arithmetic enclosures shrink quadratically [5]. Figures 9, 10, and 11 show other examples for quadratic, cubic, and quintic maps. Tables 1, 2, 3, and 4 measure this convergence by showing how the sizes of the cell graphs vary with the size $L \times L$ of the starting grid. In these tables, V is the number of vertices in the graph, E is the number of edges in the graph, SCC is the number of vertices in the strongly connected components. For the affine arithmetic variant, we also show the percentages relative to those numbers of the interval arithmetic variant.

These results suggest that cell mapping with affine arithmetic can be a useful tool for the robust visualization of strange attractors.

Acknowledgments

The authors are partially supported by CNPq research grants. Afonso Paiva is a member of the Matmídia laboratory at PUC-Rio, which is sponsored by CNPq and Petrobras. L. H. de Figueiredo is a member of Visgraf, the computer graphics laboratory at IMPA, which is sponsored by CNPq, FAPERJ, FINEP, and IBM Brasil.

The authors thank the anonymous reviewer, whose valuable suggestions helped improve the presentation of this paper.

References

[1] M. Hénon. A two-dimensional map with a strange attractor. *Communications in Mathematical Physics*, 50(1):69–77, 1976.

[2] M. Benedicks and L. Carleson. The dynamics of the Hénon map. *Annals of Mathematics*, 133(1):73–169, 1991.

[3] R. E. Moore. *Interval analysis*. Prentice-Hall, 1966.

[4] D. Michelucci. Reliable representations of strange attractors. In W. Kraemer and J. W. von Gudenberg, editors, *Scientific Computing, Validated Numerics, Interval Methods*, pages 379–389, Kluwer, 2001.

[5] L. H. de Figueiredo and J. Stolfi. Affine arithmetic: concepts and applications. *Numerical Algorithms*, 37(1-4):147–158, 2004.

[6] D. Goldberg. What every computer scientist should know about floating-point arithmetic. *ACM Computing Surveys*, 23(1):5–48, 1991.

[7] R. M. Corless. What good are numerical simulations of chaotic dynamical systems? *Computers & Mathematics with Applications*, 28(10–12):107–121, 1994.

[8] E. Adams, W. F. Ames, W. Kühn, W. Rufeger, and H. Spreuer. Computational chaos may be due to a single local error. *Journal of Computational Physics*, 104(1):241–250, 1993.

[9] R. M. Corless, C. Essex, and M. A. H. Nerenberg. Numerical methods can suppress chaos. *Physics Letters A*, 157(1):27–36, 1991.

[10] C. S. Hsu. *Cell-to-Cell Mapping*. Springer-Verlag New York, 1987.

[11] C. S. Hsu. Global analysis by cell mapping. *International Journal of Bifurcations and Chaos*, 2(4):727–771, 1992.

[12] R. Tarjan. Depth-first search and linear graph algorithms. *SIAM Jounal on Computing*, 1(2):146–160, 1972.

[13] M. Dellnitz and A. Hohmann. A subdivision algorithm for the computation of unstable manifolds and global attractors. *Numerische Mathematik*, 75(3):293–317, 1997.

[14] Z. Galias. Interval methods for rigorous investigations of periodic orbits. *International Journal of Bifurcation and Chaos*, 11(9):2427–2450, 2001.

[15] J. C. Sprott. Automatic generation of strange attractors. *Computer & Graphics*, 17(3):325–332, 1993.

[Fig. 8 about here.]

[Fig. 9 about here.]

[Fig. 10 about here.]

[Fig. 11 about here.]

[Table 1 about here.]

[Table 2 about here.]

[Table 3 about here.]

[Table 4 about here.]

List of Figures

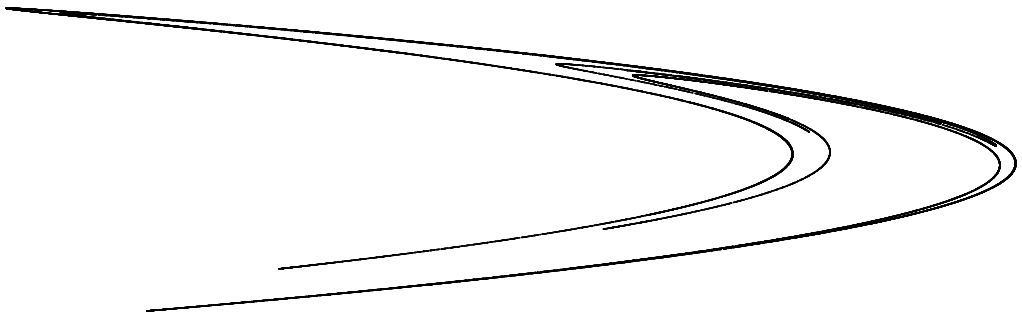


Fig. 1. The Hénon strange attractor.

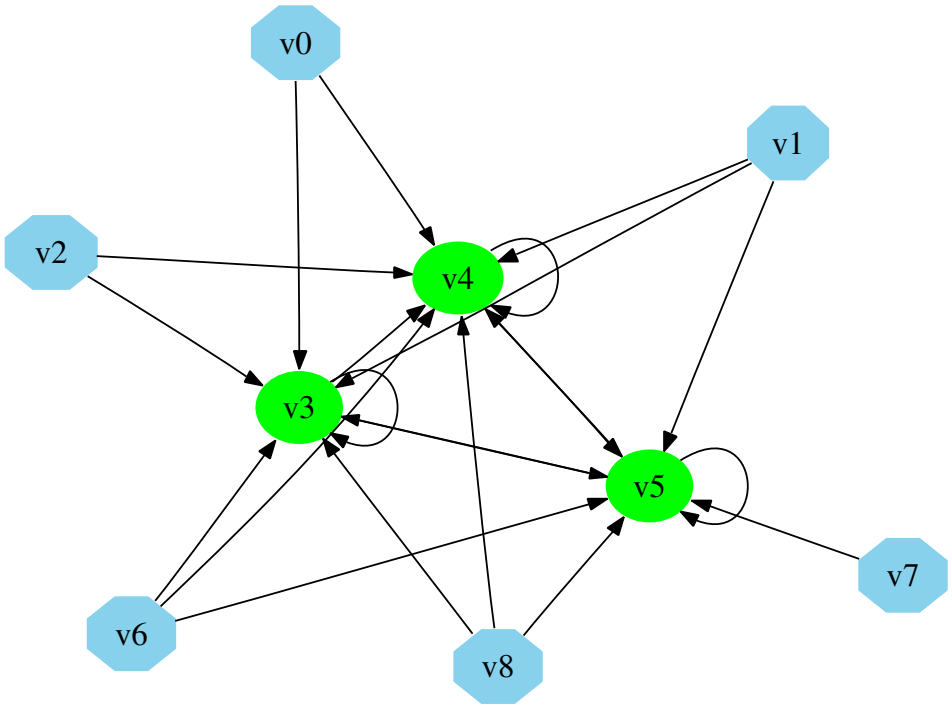


Fig. 2. Cell graph of the Hénon map based on a 3×3 rectangular subdivision. The strongly connected component is shown in green.

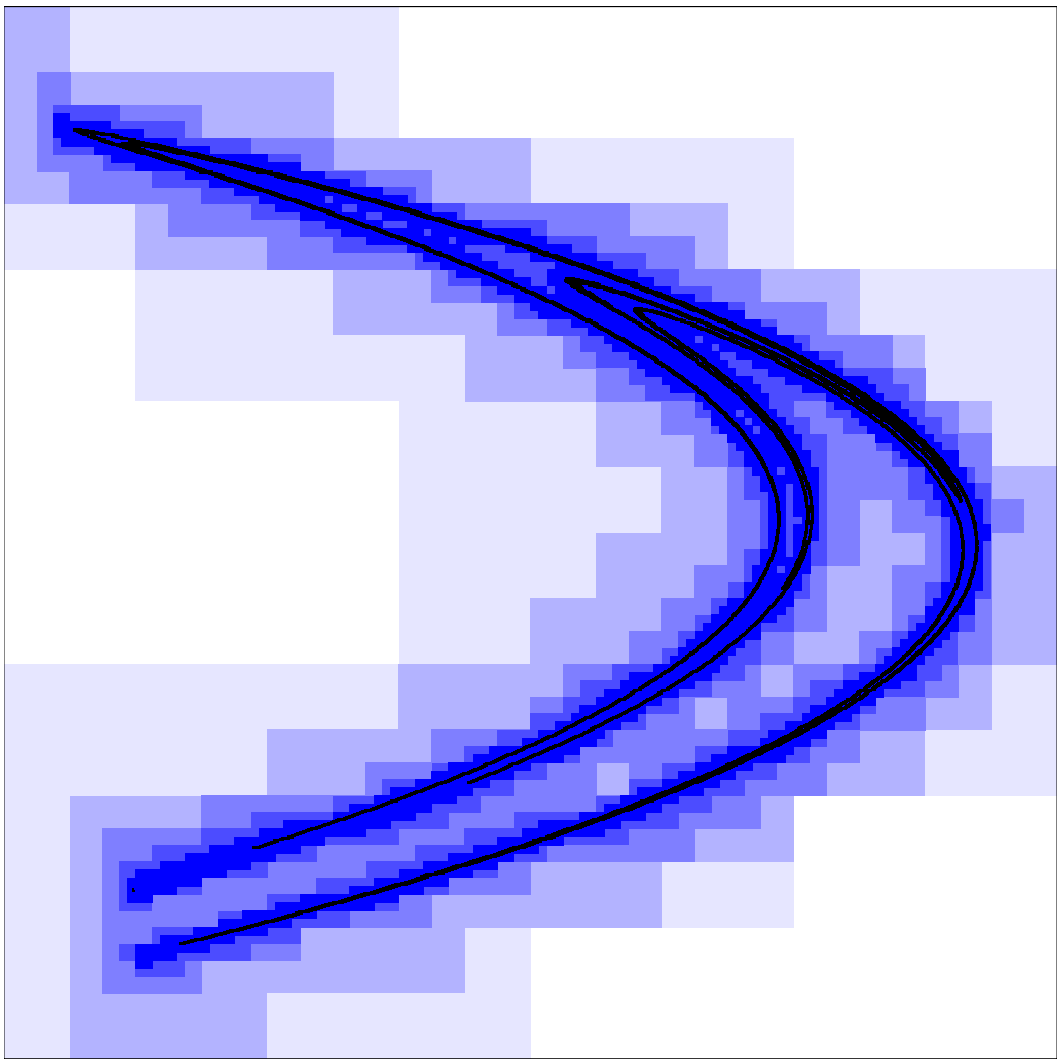


Fig. 3. Cellular approximation for the Hénon attractor.

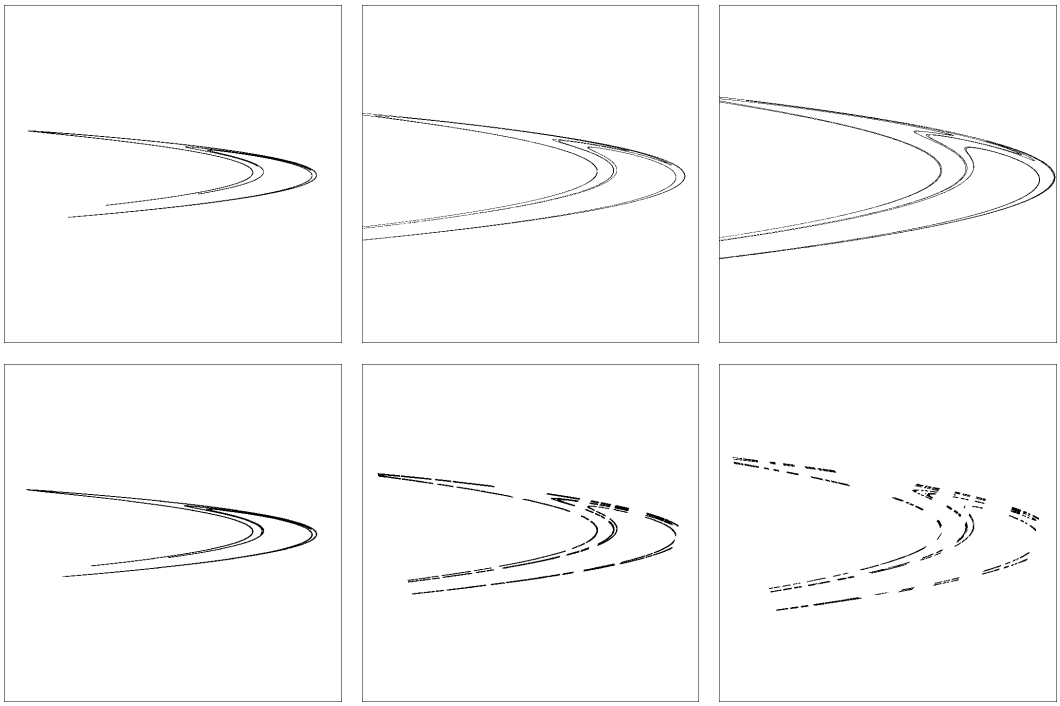


Fig. 4. Hénon strange attractor for $a = 1.4$ and $b = 0.3$ (left), 0.4 (center), and 0.5 (right). Top row: point-sampling method. Bottom row: cell-mapping method using interval arithmetic.

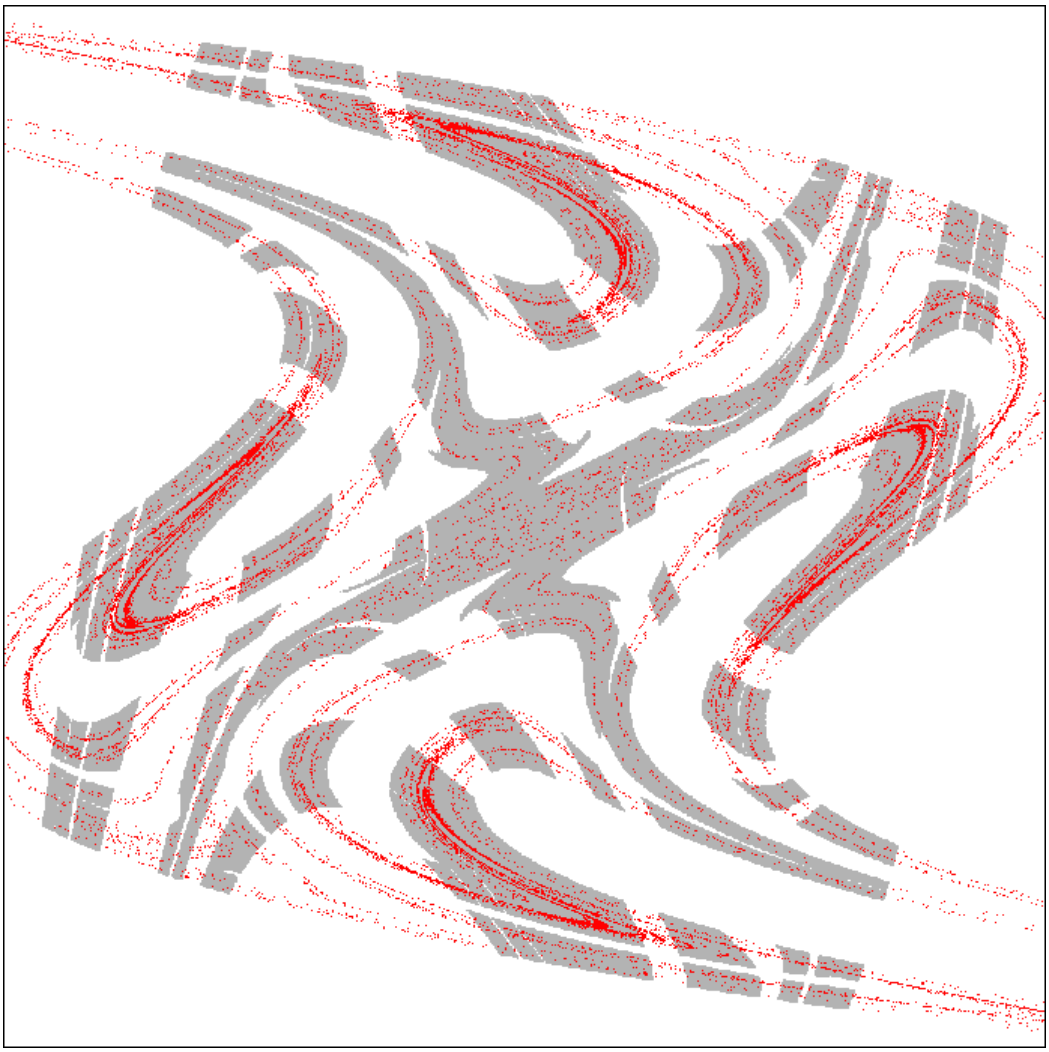
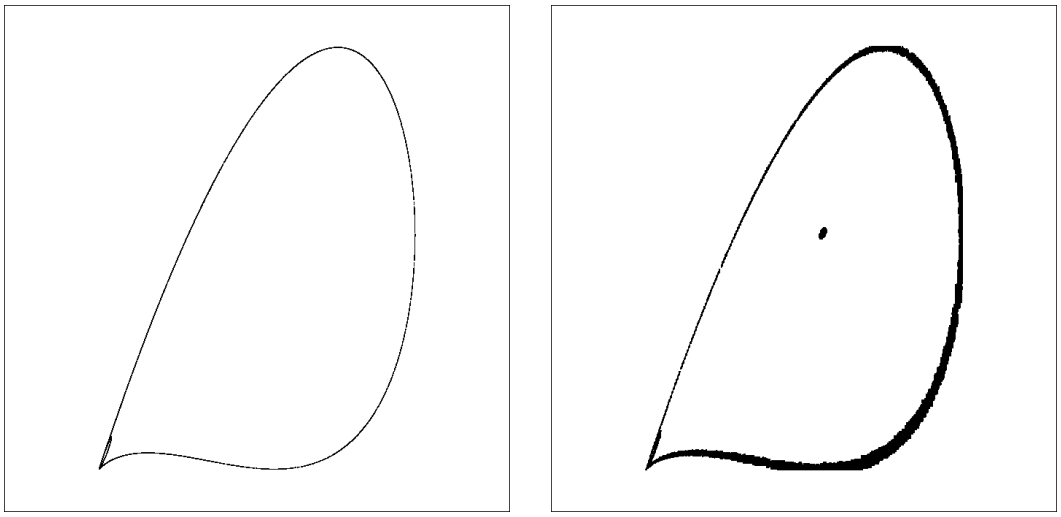


Fig. 5. Attractor of the Holmes map using sampling method (red) and cell mapping using interval arithmetic (grey).



(a) (b)

Fig. 6. Attractor of the logistic map computed with the sampling method (a) and with the cell-mapping method using IA (b).

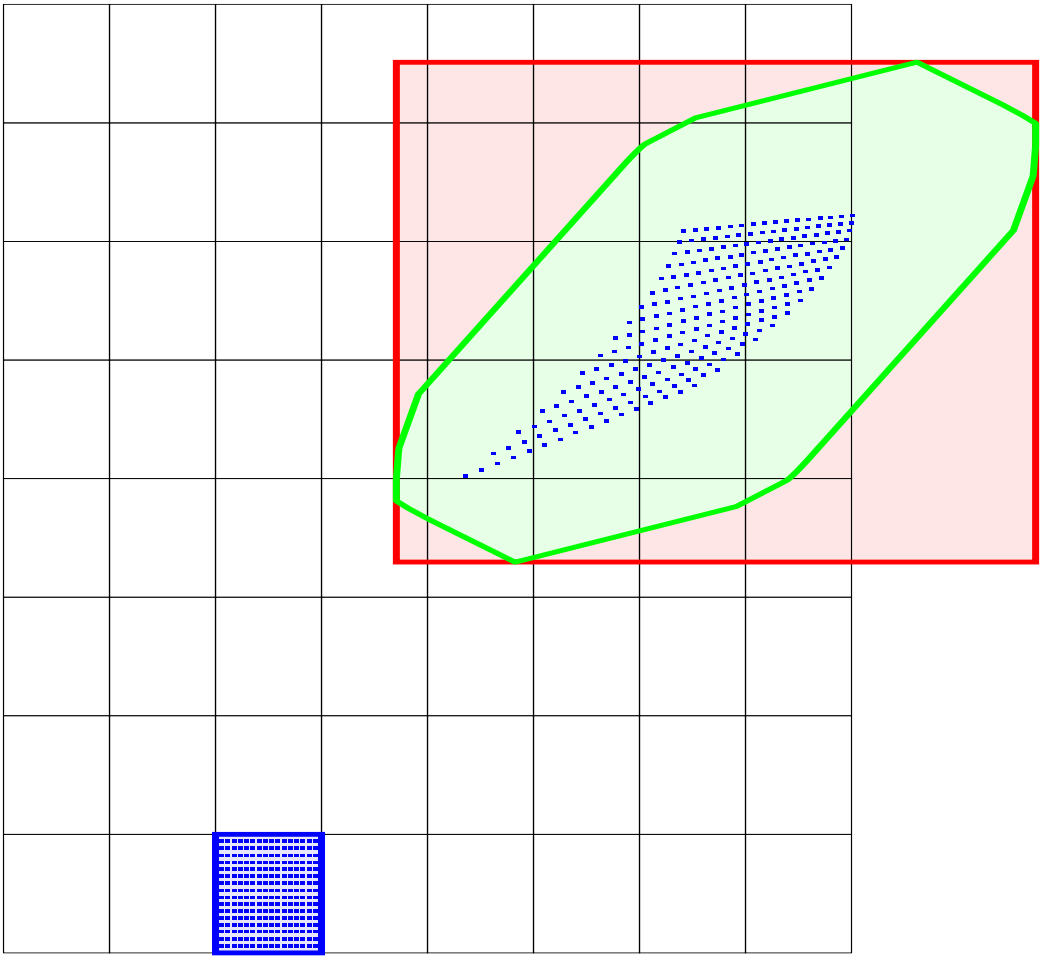


Fig. 7. Approximating $f(A)$ with the sampling method (blue), with cell mapping using IA (red) and cell mapping using AA (green).

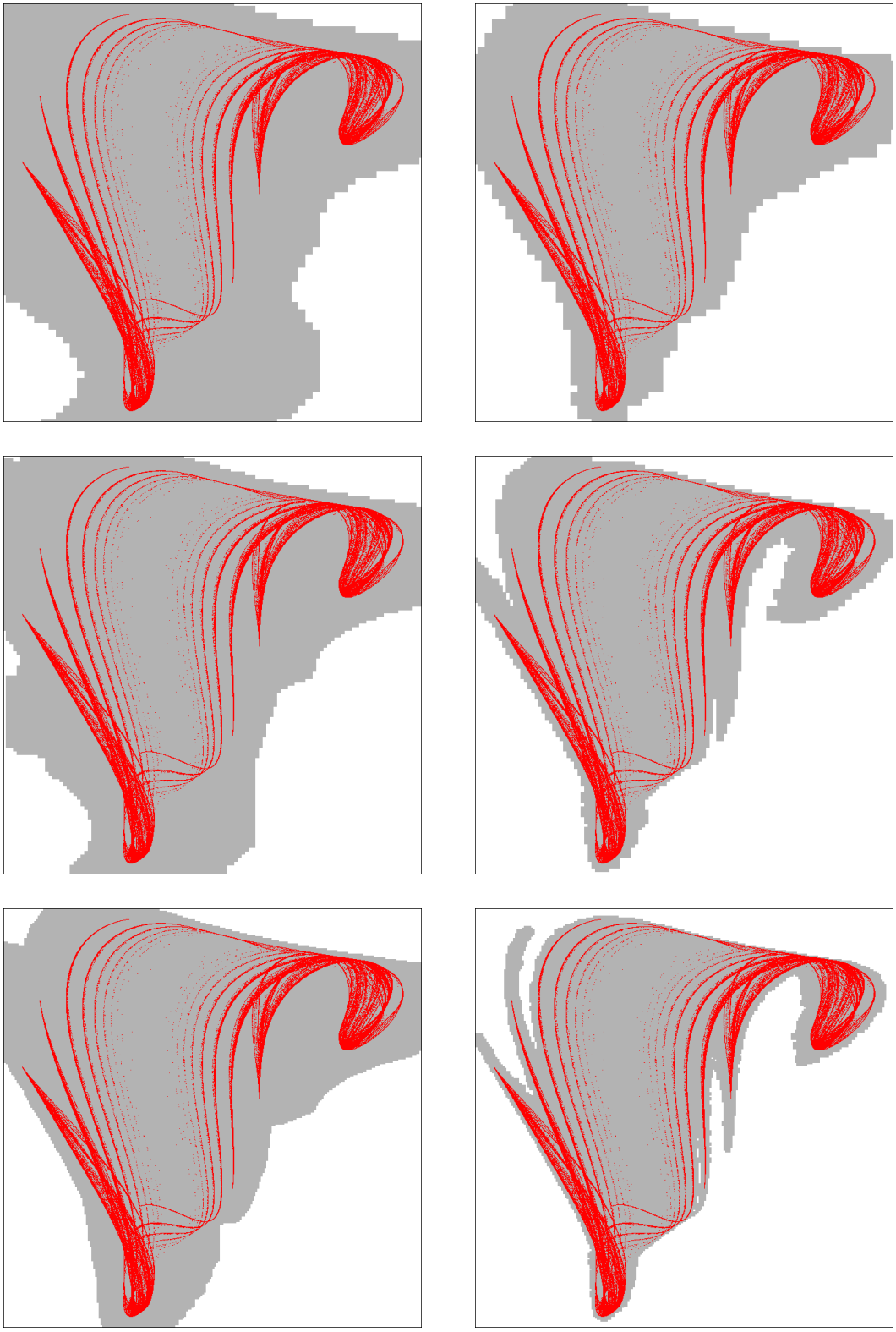


Fig. 8. Strange attractor of the quintic map *QBKSKIXQMKEOVVMAHXLBOQQJXEYMBUMBOEFVDBAPWU* in $[-0.260287, 0.952145] \times [-0.994131, 0.968682]$. Point-sampling method show in red. Cell-mapping method shown in grey using IA (left) and AA (right). From the top to bottom, pictures have resolution 64^2, 128^2, and 256^2 cells.

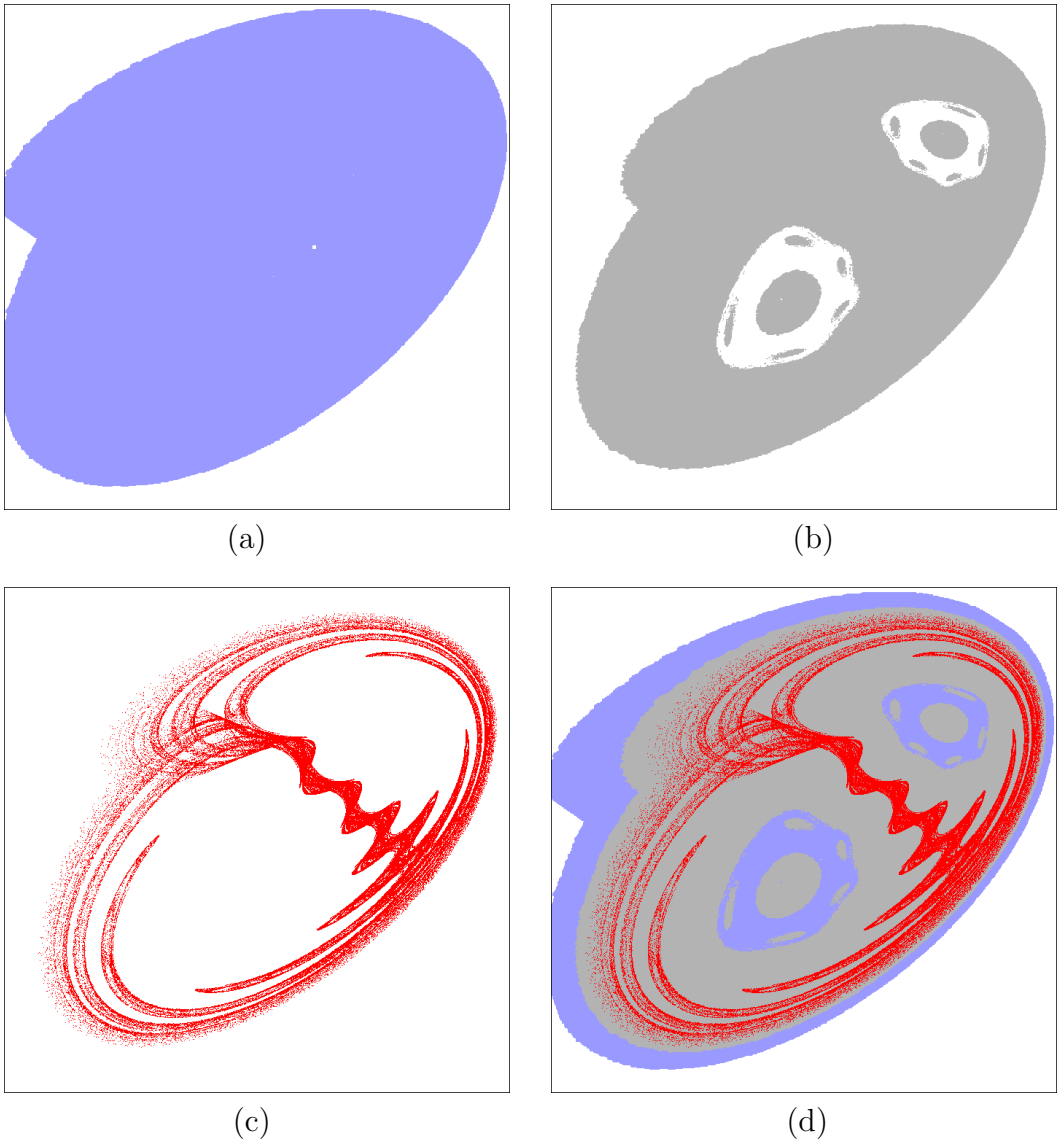


Fig. 9. Strange attractor of the quadratic map *LUFBBFISGJYS* in $[0.0, 2.0] \times [-1.8, -0.2]$. (a) Cell mapping based on IA (blue). (b) Cell mapping based on AA (grey). (c) Sampling method (red). (d) All together. Cell mapping done for a 1024×1024 cell grid.

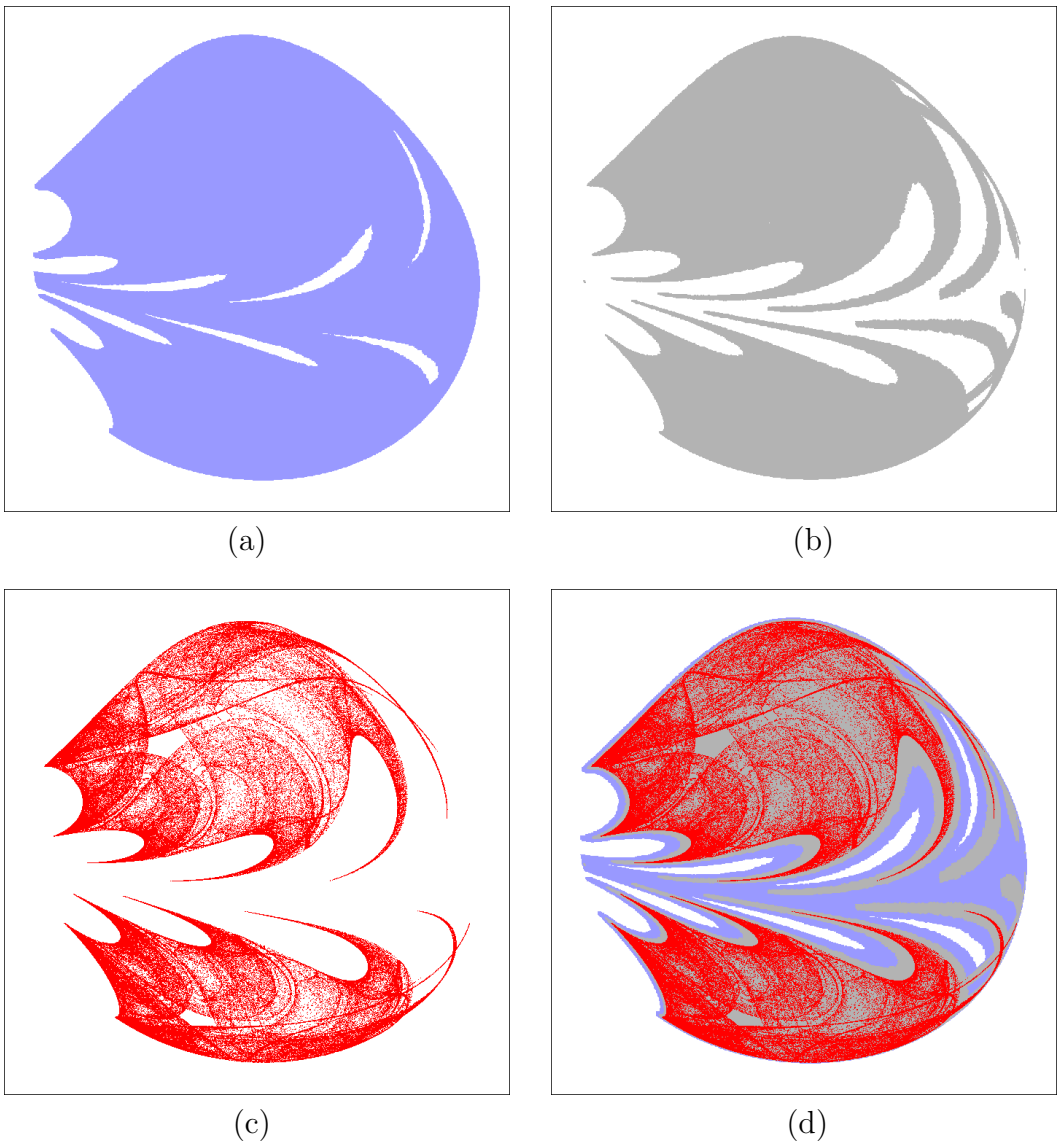


Fig. 10. Strange attractor of the cubic map *OVFKWKEIBPGNYPVKWCYU* in $[-0.707942, 0.537318] \times [-0.874148, 0.680792]$. (a) Cell mapping based on IA (blue). (b) Cell mapping based on AA (grey). (c) Sampling method (red). (d) All together. Cell mapping done for a 1024×1024 cell grid.

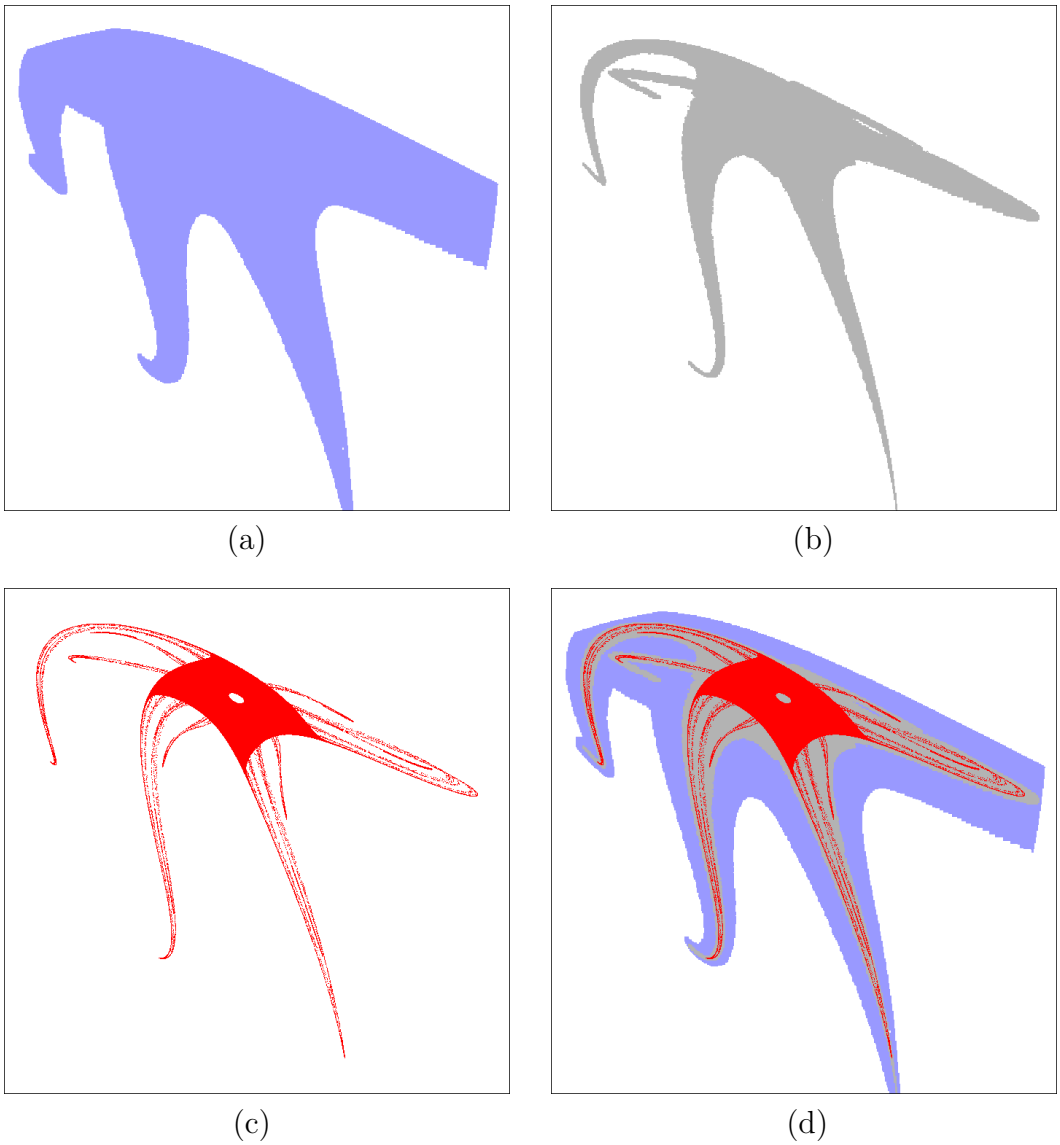


Fig. 11. Strange attractor of the quintic map *HVOIEGIDJCSFUFJCQGRUGMCLHEPWKRCCYFIRQPYAPH* in $[-1.14556, 0.411485] \times [-0.294105, 1.10915]$. (a) Cell mapping based on IA (blue). (b) Cell mapping based on AA (grey). (c) Sampling method (red). (d) All together. Cell mapping done for a 512×512 cell grid.

List of Tables

Table 1
Cell graph data for the quadratic map of Figure 9.

	IA			AA		
L	V	E	SCC	V	E	SCC
4	16	245	16	16 (100%)	99 (40%)	16 (100%)
8	64	1872	64	64 (100%)	383 (20%)	62 (97%)
16	256	9136	256	248 (97%)	1467 (16%)	234 (91%)
32	1024	38344	997	936 (91%)	5628 (15%)	876 (88%)
64	3988	152900	3728	3504 (88%)	21219 (14%)	3278 (88%)
128	14912	605570	13964	13112 (88%)	79355 (13%)	12508 (90%)
256	55856	2345067	52490	50032 (90%)	302750 (13%)	44363 (85%)
512	209960	8904283	200917	177452 (85%)	1043879 (12%)	160070 (80%)
1024	803668	34194297	734565	640280 (80%)	3669033 (11%)	542783 (74%)

Table 2
Cell graph data for the cubic map of Figure 10.

	IA			AA		
L	V	E	SCC	V	E	SCC
4	16	199	16	16 (100%)	123 (62%)	16 (100%)
8	64	1190	64	64 (100%)	484 (41%)	62 (97%)
16	256	5337	243	248 (97%)	1808 (34%)	221 (91%)
32	972	21429	892	884 (91%)	6132 (29%)	745 (84%)
64	3568	80832	3127	2980 (84%)	19907 (25%)	2733 (87%)
12	12508	268303	11090	10932 (87%)	72016 (27%)	10108 (91%)
25	44360	889579	40888	40432 (91%)	263257 (30%)	38215 (93%)
51	163552	3170359	154752	152860 (93%)	988495 (31%)	136247 (88%)
10	619008	11707914	568272	544988 (88%)	3518359 (30%)	442857 (78%)

Table 3
Cell graph data for the quintic map of Figure 8.

	IA			AA		
L	V	E	SCC	V	E	SCC
4	16	241	16	16 (100%)	178 (74%)	16 (100%)
8	64	2348	64	64 (100%)	763 (32%)	57 (89%)
16	256	16196	241	228 (89%)	2295 (14%)	187 (78%)
32	964	80095	853	748 (78%)	7094 (9%)	637 (75%)
64	3412	325048	3057	2548 (75%)	22426 (7%)	2162 (71%)
128	12228	1193467	10559	8648 (71%)	68490 (6%)	7161 (68%)
256	42236	3936173	35512	28644 (68%)	204113 (5%)	24170 (68%)

Table 4
Cell graph for the quintic map of Figure 11.

	IA			AA		
L	V	E	SCC	V	E	SCC
4	16	229	16	16 (100%)	187 (82%)	16 (100%)
8	64	2648	64	64 (100%)	825 (31%)	64 (100%)
16	256	22654	255	256 (100%)	2703 (12%)	217 (85%)
32	1020	144250	1004	868 (85%)	7541 (5%)	534 (53%)
64	4016	706540	3950	2136 (53%)	17788 (3%)	1337 (34%)
128	15800	3023995	12358	5348 (34%)	46066 (2%)	3722 (30%)
256	49432	10546388	31385	14888 (30%)	131297 (1%)	11267 (36%)
512	125540	32219415	82162	45068 (36%)	400870 (1%)	35838 (44%)