RELIABLE REPRESENTATIONS OF STRANGE ATTRACTORS

Dominique Michelucci
École Nationale Supérieure des Mines de Saint-Étienne,
158 cours Fauriel, F 42023 Saint-Étienne cedex 2
Dominique.Michelucci@emse.fr

Keywords: Hénon map, strange attractors, interval arithmetic, strongly connected components, directed projective interval arithmetic.

Abstract The method described here relies on interval arithmetic and graph theory to compute guaranteed coverings of strange attractors like Hénon attractor. It copes with infinite intervals, using either a geometric method or a new directed projective interval arithmetic.

1. INTRODUCTION

This paper shows that inaccuracy of floating-point arithmetic generates severe mistakes, when one draws sets like Hénon attractors with the classical "orbit method". It presents as well a reliable method, which computes a safe covering of the strange attractor. This method uses interval arithmetic and graph theory. Ensuring reliability requires the subdivision of the entire plane, and thus the computation with infinite intervals. Two solutions are implemented: the first is a geometric one, using the classical mapping between the oriented projective plane and a half-sphere; the second introduces an arithmetic on directed projective intervals, and seems new.

Previous related works. Interval-based methods are used to compute rigorous enclosures of transversal homoclinic points in discrete dynamic systems [NR93], and in Poincaré map of the Thiele-Wilson system [RNS94]: the existence of such transversal homoclinic points proves the chaoticity of these systems. Interval-based methods are also used to compute periodic points of dynamic systems [Gal97], or prove their non-existence, for the classical Hénon map. Impact of floating-point arithmetic inaccuracy on simulations is studied by M. Pichat and J. Vignes [PV95]; they compute pictures of the invariant set of Hénon rotations

(area preserving Hénon maps [Hén69]) with the classical orbit method: resulting pictures depend on the used rounding mode for floating-point arithmetic. They use a stochastic arithmetic in order to eliminate, or at least reduce, the numerical noise due to floating-point arithmetic. J-F. Colonna [Col96] runs the same N-body simulation program on several computers (all respecting the IEEE norm of floating-point arithmetic) with the same initial state, and obtains different results, because of *a priori* immaterial differences between compilers or between processors; for instance a compiler compiles $a \times b \times c$ as $a \times (b \times c)$, and another compiles it as $(a \times b) \times c$; these two expressions differ slightly with floating-point arithmetic, and a non-linear behavior can exponentially expand this initially negligible difference.

Implementation. The language Ocaml from INRIA, and its exact rational arithmetic library: Num are used to program a straightforward interval arithmetic with $+, -, \times, \div$ operations. Bounds of intervals are rational numbers, so there is no need of outwards rounding, as opposed to floating-point based interval arithmetics. I initially chose this solution for its simplicity in programming and its reliability but, ironically, this work revealed several bugs in first releases of Num ...

Plane. Section 2 defines Hénon maps and strange attractors. Section 3 presents the orbit method, section 4 the interval-based method, section 5 two methods to cope with infinite intervals. Section 6 concludes.

2. HÉNON MAP AND ATTRACTOR

The Hénon map and the corresponding strange attractor (or invariant set) are used throughout this paper as a typical example. The Hénon map transforms point $(x, y) \in \mathbb{R}^2$ to point

$$H(x, y) = (y + 1 - ax^2, bx) \in \mathbb{R}^2 \tag{1.1}$$

classically with parameters value: $a = 1.4$ and $b = 0.3$. The Jacobian is

$$H'(x, y) = \begin{pmatrix} -2ax & b \\ 1 & 0 \end{pmatrix} \tag{1.2}$$

It has determinant $|H'(x, y)| = -b$: when $|b| < 1$, H is contracting (i.e. H contracts areas). In the classical case $a = 1.4$ and $b = 0.3$, Hénon exhibited a trapped quadrilateral R, whose image $H(R)$ is contained in R; the so-called Hénon attractor can be defined as $\lim H^{(k)}(R)$ when $k \to \infty$ [Hén76] (this paper does not use this definition, but another one based on recurrent points, see below). This set is today one of the icons of chaos [Ott93, Rob95], as Lorenz attractor, Rössler attractor, Julia sets or Mandelbrot set.

The *orbit* of a point p is the set of its iterates:

$$O(p) = \{H(p), H^{(2)}(p), \ldots\} \tag{1.3}$$

The point p is said to be *recurrent* iff its orbit contains points which come back arbitrarily close to p. Non-recurrent points are said to be *transient*. Periodic points are obviously recurrent and, as is known, are dense in the set: $Rec(H)$ of recurrent points. The latter set: $Rec(H)$ is called the strange attractor of the map, sometimes its invariant set. Hénon attractor is the set of recurrent points of Hénon map.

3. THE ORBIT METHOD

To display these sets, the classical method selects a random point p in their basin of attraction, computes in floating-point arithmetic the orbit $O(p) = \{H(p), H^{(2)}(p), \ldots\}$, gets rid of the first (e.g. thirty) iterates, and plots the next thousand points or so. This method is used in Computer Graphics, but also in the numerical study of dynamic systems to compute Lyapunov exponents, fractal dimensions or Kolmogorov entropies.

Sometimes, no initial point in the attraction basin is known, for instance when no trapped region is known : then random points are picked, and their orbit is followed in the same way, while they are close enough to origin : after, it is assumed they are attracted by infinity.

Due to inaccuracy, the computed orbit $O(p)^* = \{H(p)^*, H^{(2)}(p)^*, \ldots\}$ diverges from the exact one, with an exponential rate (measured by Lyapunov exponents), until their difference is as large as the strange attractor size. But it is generally assumed that the global picture is correct, the main argument being the shadow lemma. The latter claims that, under some assumptions, there is another point (actually an infinity of points) q such that the difference between the computed orbit $O(p)^*$ and the exact orbit $O(q)$ stays arbitrarily small. Thus, though $O(p)^*$ becomes quickly completely wrong, it follows like its shadow the exact orbit $O(q)$. For Hénon parameters : $a = 1.4, b = 0.3$, this is confirmed by Fig. 1, which shows there is almost no difference between a picture obtained with the classical method, and the one obtained with the reliable interval-based method; the isolated periodic point $\omega = H(\omega)$ is left away by the orbit method, not by the interval-based one (the corresponding isolated pixel is likely not visible on the paper; for printing reasons, this paper does not show the more accurate pictures, the thin features of which disappear after printing). Also, the most accurate pictures (unfortunately not printable) computed with the interval-based method are of greater sharpness because, with the orbit method, some points may exit a bit out of the strange attractor which is locally repulsive, before being attracted again.

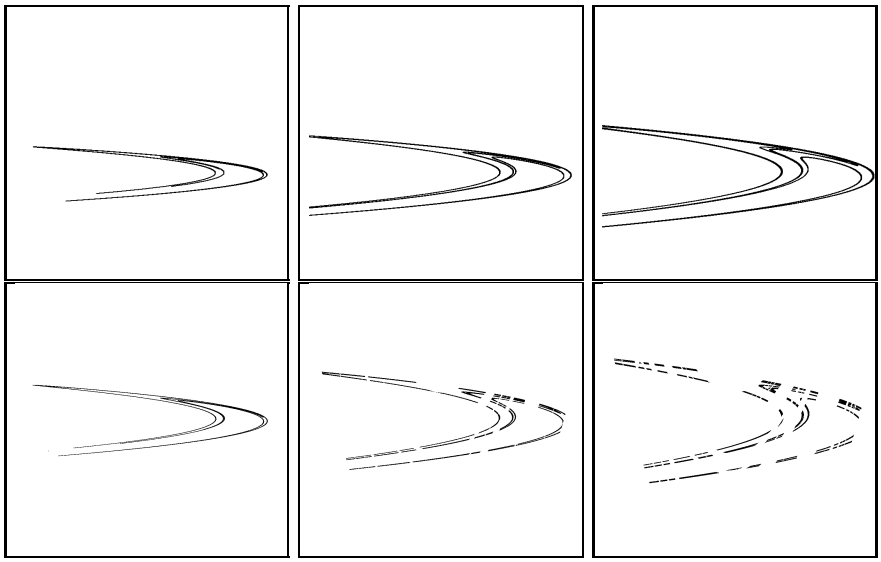

Figure 1 Strange attractors in square windows $[-1.5, 1.5]^2$. From left to right: $a = 1.4; b = 0.3, 0.4, 0.5$. Above: orbit method. Below: interval-based method (with a huge initial box).

For other values of b, namely $b = 0.4$ to 0.95 (the corresponding map H is still contracting, but the strange attractor is less attracting), pictures obtained by the two methods differ dramatically : see Fig. 1, 2, 3. It is due to the fact that some points, which exit a bit out of the strange attractor, are not attracted again, but escape.

The sharpness of pictures computed with the interval-based method is astonishing, when one is aware of the wrapping effect [Neu93] in interval arithmetics.

4. THE INTERVAL-BASED METHOD

The interval-based method presented in this section provides a safe covering of the strange attractor. For the sake of simplicity, let us first assume that a bounding box of the strange attractor is known: the next section will relieve of this constraint. The interval-based method subdivides the bounding box into regular square or rectangular grid cells, and computes with interval arithmetic a range of the image of each grid cell: it induces a directed graph.

Let c be a grid cell, and $H(c)$ be the range of $H(x, y)$ where $(x, y) \in c$: $H(c)$ is obtained with an interval arithmetic. Generally, i.e. when $H(c)$ does not fall outside the bounding box, $H(c)$ cuts grid cells $c_1 \ldots c_k$. In

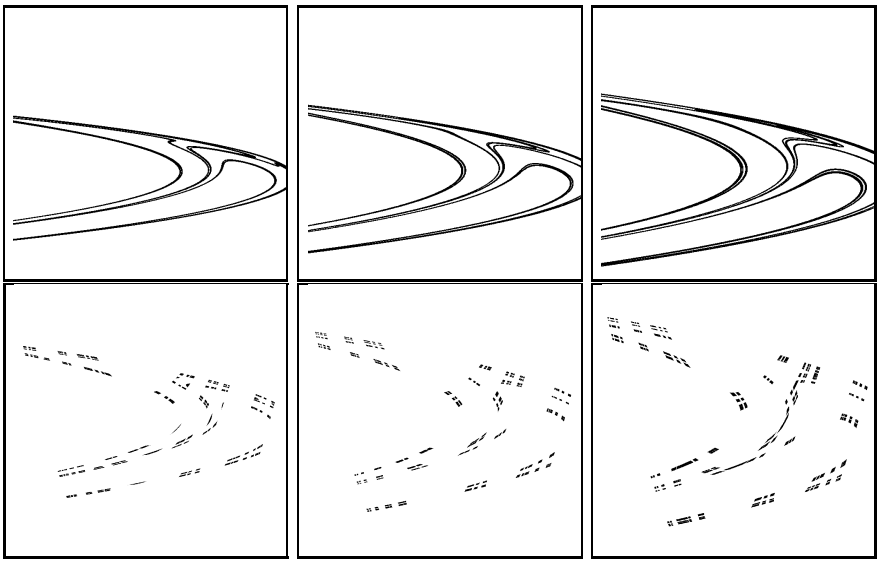

Figure 2 Strange attractors in square windows $[-1.5, 1.5]^2$. From left to right: $a = 1.4; b = 0.6, 0.7, 0.8$. Above: orbit method. Below: interval-based method (with a huge initial box).

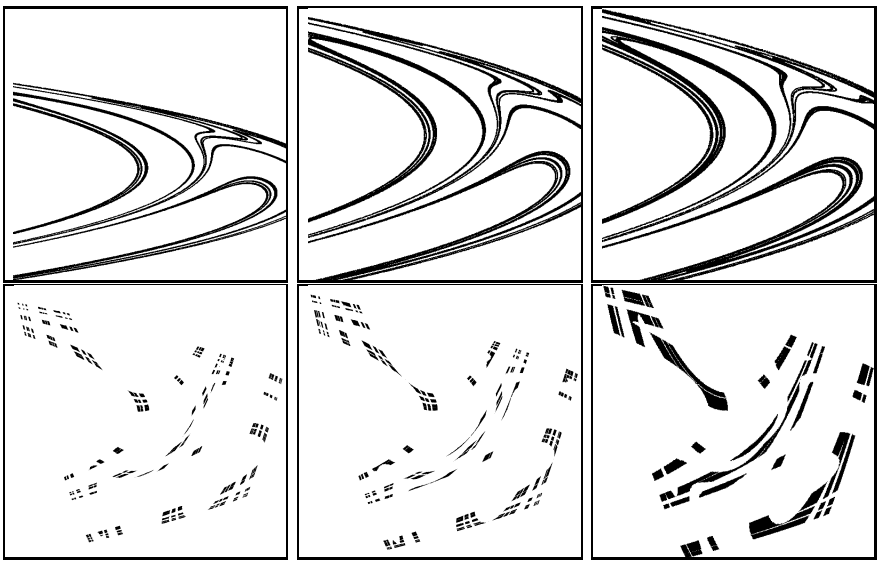

Figure 3 Strange attractors in square windows $[-1.5, 1.5]^2$. From left to right: $a = 1.4; b = 0.9, 0.95, 1.0$. Above: orbit method. Below: interval-based method (with a huge initial box).

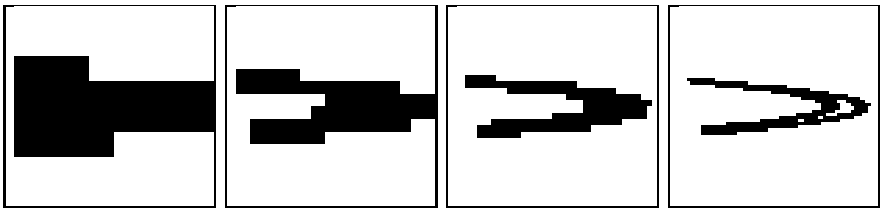

Figure 4 $a = 1.4, b = 0.3$. Square windows are $[-1.5, 1.5]^2$. From left to right, pictures have increasing resolution: $8^2, 16^2, 32^2, 64^2$ cells. Except the first one, each picture is obtained from the previous one by subdividing its recurrent cells.

the simple but rough implementation, arcs $c \to c_1, \ldots c \to c_k$ are added to the graph. To get a tighter result, a more advanced approach adds each arc $c \to c_i$ to the graph only iff there is some point $p \in c$ such that $H(p) \in c_i$. An alternate method subdivides each grid cell c into (say) 4×4 subcells, computes with interval arithmetic the range of the image of each subcell, then finds the cut grid cells. Note that, for the Hénon map, this last method is not interesting: since variables x and y occur only once in the computation of $H(x, y)$, there is no wrapping effect, and the resulting box obtained for $H(c)$ is the optimal one —up to outwards rounding of interval arithmetic, but anyway our implementation, which uses rational arithmetic, does not round outwards.

Anyway, it defines a directed graph between grid cells. If this graph contains no loop : $c \to c$ or $c \to c_1 \ldots \to c$, then the grid cell c cannot contain points of $Rec(H)$. Conversely, if there is at least one loop : $c \ldots \to c$ in the graph, then the grid cell c may contain some points of $Rec(H)$.

To speed up the existence test for loops, strongly connected components of the graph are computed once and for all with the Tarjan algorithm [CLR90], which is linear in the graph complexity (number of vertices and arcs). As a result, each vertex of the graph, i.e. each grid cell c, is numbered with a strongly connected component index $\mathrm{scc}(c)$. Two vertices a, b have equal index $\mathrm{scc}(a) = \mathrm{scc}(b)$, i.e. belong to the same strongly connected components, iff the graph contains a directed path $a \to \ldots \to b$ and a directed path $b \to \ldots \to a$.

A strongly connected component is transient iff it contains only one vertex (i.e. a grid cell) c and if there is no loop $c \to c$ in the graph; in other words, iff none of the successors s of c in the graph has the same scc number as c. A transient cell contains strictly no recurrent point.

Otherwise, the graph permits, starting from c, to come back to c: the cell c is said to be recurrent, since c may contain some recurrent points of H.

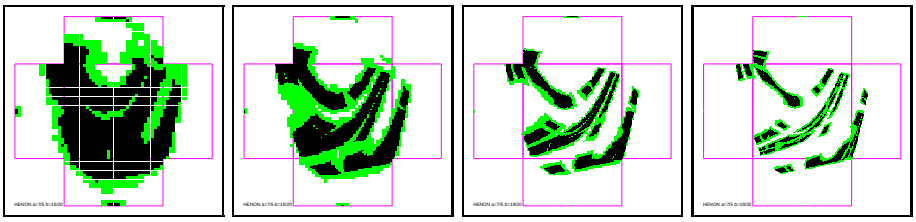

Figure 5 Unfolded hemicubes, with $a = 1.4, b = 0.95$, and increasing resolution. Resolution (i.e. number of cells) of the central face is: 64^2, 128^2, 256^2, 512^2.

The set of recurrent grid cells provides a safe covering of the strange attractor: indeed a transient grid cell contains no point of the strange attractor. Conversely, this method gives no guarantee that a recurrent grid cell really contains points of the strange attractor. A partial and only asymptotic answer is to subdivide recurrent cells, and restart the method : see Fig. 4; in practice, these successive zooms give a fast method to get an arbitrarily accurate picture of the strange attractor, up to the available memory space of course, and indeed even the most accurate pictures are computed in less than ten minutes on a standard personal computer, without any concern for optimization. However this method never proves a cell really cuts the strange attractor. Another possible method (not implemented in this work) is to search for periodic points in recurrent cells: since periodic points are dense in the strange attractor, their abscence proves that the grid cell contains no recurrent points. Each loop in the graph gives a candidate for a cycle. Some interval Newton method can then prove the existence, or the non-existence, of the cycle [Gal97].

Finally, it is worth noting this interval-based method also applies to non-contracting maps, for instance the inverse of contracting Hénon map ($|b| < 1$) defined by:

$$H^{-1}(x, y) = (y/b, x - 1 + ay^2/b^2) \tag{1.4}$$

and gives the same pictures of the strange attractor, just a bit more slowly. The orbit method is not relevant as far as expanding maps are concerned.

5. SUBDIVIDING THE ENTIRE PLANE

A bounding box of the strange attractor is not always available, for instance for non-classical values of parameters a, b of the Hénon map. A partial solution is to use a huge initial bounding box, like $[-2^{100}, 2^{100}]$, with a low grid resolution, like 16×16, then to iterately zoom, i.e.

subdivide in 2×2 or 4×4 recurrent cells; with this approach, the use of a huge initial bounding box does not penalize. However, there is always a doubt: is the initial bounding box really big enough? How can we be sure that the differences between the results of the two methods is not due to a wrong (for instance too small) initial box? A secure solution is to subdivide the entire plane $\mathbb{R}^2$. Since we will use a finite number of cells for computational reasons, some of them will be infinite. So we need some method to handle infinite cells, and underlying infinite intervals. Two methods are now presented.

5.1. THE ORIENTED PROJECTIVE PLANE

Not surprisingly, a method is to use homogeneous coordinates and the projective plane. Actually, there are two projective planes: the one-sided (or non-oriented, or unsigned) projective plane in which direction (a, b) (i.e. the point at infinity, in the direction (a, b)) and $(-a, -b)$ do not differ, and the two-sided (or oriented, or signed) projective plane in which they are opposite [Sto91]. In the non-oriented projective plane or space, it is impossible to define *the* line segment between two points, i.e. to distinguish the two simple arcs they cut the line into: it is impossible to define intervals or cells. Thus the method uses the two-sided projective plane.

In practice, each finite point (X, Y) of the plane $\mathbb{R}^2$ is represented by any point $(h, x, y) = (h, hX, hY)$ with $h > 0$ in the oriented projective plane – In Computer Graphics, people often use $h = 1$. A point at infinity in the direction (a, b) is represented by any point $(0, ha, hb)$ where $h > 0$, so both directions (a, b) and $(-a, -b)$ are indeed opposite – In Computer Graphics, people often use h so that $(ha)^2 + (hb)^2 = 1$ (up to round-off errors).

With this representation, points (X, Y) of the plane $\mathbb{R}^2$ can be mapped to the upper half-sphere: $(h \geq 0, x, y)$ with $h^2 + x^2 + y^2 = 1$ and $x = hX$ and $y = hY$: points $(0, 0, 0)$, (h, x, y) and $(1, X, Y)$ are aligned. This half-sphere has finite area, and it is subdivided into a finite number of finite area cells, instead of the plane.

In fact, for a computer scientist, the upper half-cube: $h \in [0, 1], x \in [-1, 1], y \in [-1, 1]$ happens to be more convenient than the upper half-sphere: the square root operation is no more necessary, it is trivial to subdivide the 5 rectangular faces of the upper half-cube into rectangular or square cells, and finally these grid cells (h, x, y) have zero volume (it would be impossible with grid cells on a sphere), achieving a better accuracy.

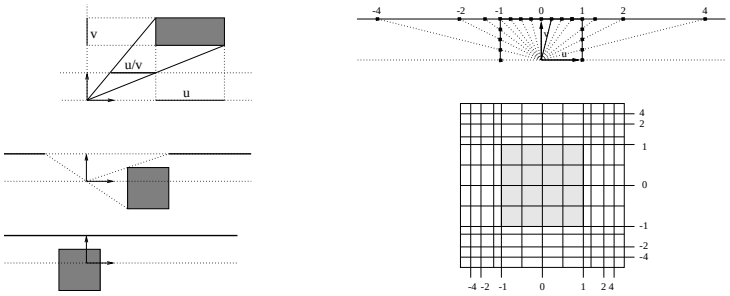

Figure 6 Left: projecting an OPI on the real line can give: a standard finite interval (above), two half infinite intervals (middle), $\mathbb{R}$ (below). Right: the partition of $(-\infty, +\infty)$ with OPI (above); how the entire plane is mapped to a finite square (below); the deformation is null inside the gray square and increases outwards.

The Hénon map has to be reformulated in terms of homogeneous coordinates. It maps now the point (h, x, y) into $(h^2, yh + h^2 - ax^2, bxh)$. A range for the image of a cell is straightforwardly computed with an interval arithmetic; this box generally lies no more on the hemicube: some projection is necessary, and easily done. Iterative zooms on recurrent cells are still possible, and still speed up the method. Fig. 5 shows the unfolded hemicube, with increasing resolutions. For all values of parameters a, b, the projective method has given results similar to the previous method, which starts with a huge initial bounding box (except of course for recurrent points at infinity).

This projective variant of the interval-based method is slower than its affine counterpart, and more space consuming for a given resolution. But, once the projective interval-based method has provided a bounding box for the strange attractor, it is of course possible to use the affine variant, this time in a reliable way.

5.2. ORIENTED PROJECTIVE INTERVALS

The previous solution uses only one homogeneous coordinate h per point. An alternative solution, also implemented and discussed now, uses one homogeneous coordinate for each coordinate x and y. Thus, contrarily to the first solution, the x value can be infinite while the y value remains finite, or vice versa.

The main advantage of this second approach is that it suggests the definition of *oriented projective intervals*, and the corresponding oriented projective interval arithmetic. The problem of managing infinity is thus solved at the arithmetic level, and no more at the geometric and algo-

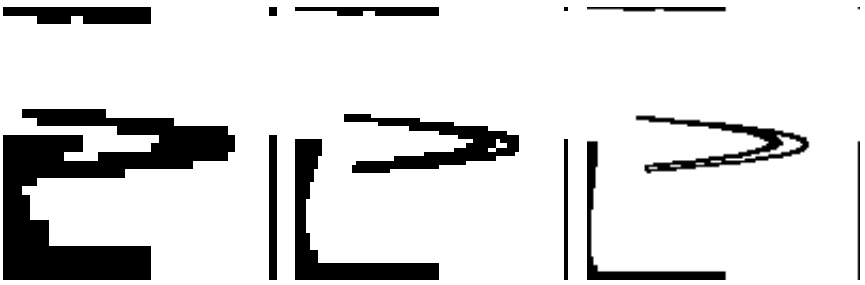

Figure 7 $a = 1.4, b = 0.3$. Projective interval method with increasing resolution: $[0,1]$ is cut into 8, 16, 32 intervals (idem for $[-1,0]$ and $(-\infty,-1]$ and $[1,\infty)$).

rithmic level as opposed to the previous solution: thus this arithmetic simplifies the algorithm.

An oriented projective interval (OPI for short) is defined by a couple (u,v) of two standard intervals, and is equal to their ratio: u/v. When $0 \notin v$, the OPI is equal to some usual finite interval: see Fig. 6. When $0 \in v$ and $0 \notin u$, the OPI is the union of two semi infinite intervals: see Fig. 6; for instance $[1,3]/[-1/2,1/2]$ is equal to $(-\infty,-2] \cup [2,+\infty)$. More precisely, the OPI u/v represents all real numbers U/V such that $U \in u$ and $V \in v - \{0\}$ (that is: the interval v except 0). The basic operations are defined by the following rules:

$$\frac{u}{v} + \frac{u'}{v'} = \frac{uv' + u'v}{vv'} \qquad \frac{u}{v} \times \frac{u'}{v'} = \frac{u \times u'}{v \times v'} \qquad \frac{1}{\frac{u}{v}} = \frac{v}{u} \tag{1.5}$$

Note there is no special case for the division. All computations of arithmetic expressions are performed with couples (u,v) and it is only at the end (for computing which cells are cut by the range of the image of a cell) that it is needed to convert the resulting OPI (u,v) to a (possibly disconnected) set of real numbers, in fact to project the OPI (u,v) to the real line: see Fig. 6. It is only during this projection that an error can occur: if $v = [0,0]$, the exception: "Division by an empty interval" is raised.

This method gives results similar to the two previous ones. Fig. 6 shows how the interval $(-\infty,+\infty)$ is subdivided, in x and y, and how the corresponding square is deformed and mapped to the plane. Fig. 7 shows this method in action, with increasing resolutions. Fig. 8 shows the results for various values of parameters.

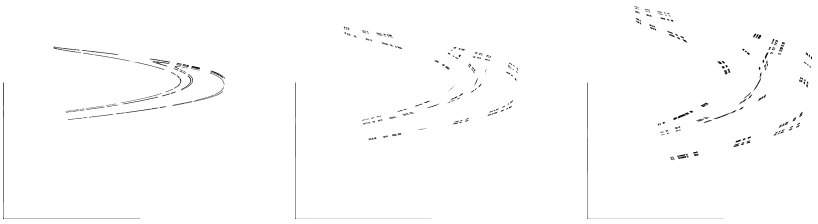

Figure 8 Results with the projective interval method. $a = 1, 4$, and from left to right, $b = 0.4, 0.6.0.8$. Resolution is 512: intervals $(-\infty, -1], [-1, 0], [0, 1], [1, +\infty)$ are regularly subdivided into 512 cells. Note the recurrent points near or at infinity.

6. CONCLUSION

This paper shows that the classical orbit method does not always give a reliable picture of strange attractors. It is an important fact, because the orbit method is not used only for drawing pictures, but also as a basic subroutine for many numerical studies on dynamic systems. A reliable interval-based method has been described, with two possible ways to reliably compute with infinite intervals. This method computes a reliable covering of the strange attractor.

We conclude with some questions raised by this work.

Greater zooms of the classical Hénon attractor, with the classical orbit method, make apparent its fractal nature. Unfortunately, the interval-based method is more space consuming than the classical orbit method, and it is unable to compute such greater zooms. In the more accurate pictures, each cell has width 1/2048 (sometimes 1/4096, for some parameters a, b with "small" strange attractors). Is it possible to overcome these limitations, resorting to some Computer Graphics techniques like quadtrees? One may also think to bound the Hénon map inside each cell with an interval affine map, and study the related IFS (iterated function system [Bar88]).

In principle, the method can give pictures of attraction basins (stable- and unstable-manifolds, in the dynamic system terminology): just compute which strongly connected components can be reached from each vertex (or cell) c. But since we can no more zoom only recurrent cells, we may again face problems due to memory space for big resolutions.

Finally, the method has been tested with conservative maps ($|b| = 1$) and various values for a. It works as well, though weaker resolutions are reached since more memory space is necessary, see Fig. 3 below right. In the conservative case, plotting only the strange attractor is frustrating:

when it is not empty, it has typically a non-zero area and a very complex internal structure, for instance KAM curves [Ott93, Rob95]. So the question arises is it possible from the graph to extract and visualize relevant features of the dynamic system, for instance using symbolic dynamic?

Acknowledgments

I gratefully acknowledge anonymous referees.

References

[Bar88] M. Barnsley. *Fractals everywhere*. Academic Press, 1988.

[CLR90] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. MIT Press, Cambridge, Massachusets, 1990.

[Col96] J-F. Colonna. Kepler, von Neumann and God. *The Visual Computer*, 12:346–349, 1996.

[Gal97] Z. Galias. Rigorous numerical studies of the existence of periodic orbits for the Hénon map. *J. Universal Computer Science*, 4(2):114–124, 1997.

[Hén69] M. Hénon. Numerical study of quadratic area-preserving mappings. *Quart. Applied Math.*, (27):291–312, 1969.

[Hén76] M. Hénon. A two dimensional map with a strange attractor. *Commun. Math. Phys.*, 50:69–77, 1976.

[Neu93] A. Neumaier. The wrapping effect, ellipsoid arithmetic, stability and confidence regions. *Comput. Supplementum*, 9:175–190, 1993.

[NR93] A. Neumaier and T. Rage. Rigorous chaos verification in discrete dynamical systems. *Physica D*, 67:327–346, 1993.

[Ott93] E. Ott. *Chaos in dynamical systems*. Cambridge University Press, 1993.

[PV95] M. Pichat and J. Vignes. CADNA: a tool for studying chaotic behavior in dynamical systems. *Proc. Real Numbers and Computers*, Ecole des Mines, St-Etienne, France, April 4–6, 1995.

[RNS94] T. Rage, A. Neumaier and C. Schlier. Rigorous verification of chaos in a molecular model. *Phys. Rev. E.*, 50, 2682–2688, 1994.

[Rob95] C. Robinson. *Dynamical systems: stability, symbolic dynamics, and chaos*. CRC Press, 1995.

[Sto91] J. Stolfi. *Oriented Projective Geometry*. Academic Press, 1991.