Noname manuscript No.
(will be inserted by the editor)

On sample-based computations of invariant sets

Shen Zeng

Received: date / Accepted: date

Abstract In this paper, the classical problem of uncovering the maximal invariant set of a (discrete-time) dynamical system is illuminated from a novel perspective, which in particular leads to a novel sample-based computational procedure to compute the invariant set. The mathematical description of these new insights can be formulated in strikingly basic set-theoretic terms, and more importantly, be efficiently realized computationally in terms of different sample-based implementations. We illustrate the simplicity and efficiency of the computational method on three examples with a maximal invariant set that is unstable in both time directions, the classical Hénon map, a three-dimensional analogue of the Hénon map, and a Van der Pol oscillator.

Keywords Discrete-time dynamical systems · Invariant set

1 Introduction

Let (X, d) be a metric space and consider a one-to-one mapping $F : X \to X$, which defines a discrete-time dynamical system by means of the iteration

$$x_{k+1} = F(x_k), \qquad k = 0, 1, 2, \ldots.$$

One of the fundamental objects in the study of discrete-time dynamical systems are subsets of X that are invariant under the application of the mapping F. More specifically, in the context of discrete-time systems, one has the notion of forward invariance of a subset $I \subset X$ under F, described by $F(I) \subset I$, which would guarantee that a trajectory initialized at a point in I will stay in I, and backward invariance of a subset $I \subset X$ under F, described by $F^{-1}(I) \subset I$. Subsets $I \subset X$ that are both forward invariant and backward invariant under F, i.e. satisfy $F(I) = I$, are of particular interest and are simply referred to as invariant sets.

S. Zeng
Department of Electrical & Systems Engineering, Washington University in St. Louis
Tel.: +1-314-9359357
E-mail: s.zeng@wustl.edu

Invariant sets can be regarded as one of the most studied objects in the field of dynamical systems, as many different dynamic phenomena encountered in (discrete-time) dynamical systems are fundamentally linked to the structure of invariant sets. Many very comprehensive methods have been developed for computing representations of invariant sets, or, more generally, invariant measures of dynamical systems. The perhaps most prominent one is GAIO [6,7,5], which is a software package with set-oriented routines at its core. For the problem of reconstructing invariant sets, GAIO is able to provide an outer approximation of the maximal invariant set by means of a box covering, which is obtained through a recursive subdivision routine. In the terminology of [9], this is referred to as a direct method, which is in contrast to indirect methods such as straightforward simulations. The great advantage of direct methods over indirect methods is that direct methods are able to uncover invariant sets that are unstable in forward or backward time.

More recently, somewhat uncoupled from the study of invariant sets and measures of dynamical systems, the combination of the spectral analysis of Koopman operators and the computational tools provided by the method of Dynamic Mode Decomposition [11,13,4,14,2,12,3] to study different aspects of complex systems have sparked great interest in sample-based and data-driven methodologies. The fundamental premise in these works is the availability of two large sets of points X_1 and X_2, called data snapshots, which are related by $X_2 = F(X_1)$, i.e. the points contained in the snapshot X_2 are the images of the points in X_1 under the mapping F. It has been very successfully shown over the years that many important quantities of dynamical systems can be extracted from solely these data snapshots, such as the computation of first integrals of motion, isostables and isochrons, as well as other Koopman eigenfunctions which often can be of great relevance to the analysis and also the control of dynamical systems. Inspired by these recent breakthroughs in the development of operator-theoretic methods leveraging data snapshots, we adopt a sample-based treatment for the computation of maximal invariant sets.

While the starting point of adopting samples as a means to characterize the invariant set is the same as [9], our subsequent considerations lead to fundamentally new viewpoints and methodologies that are quite different from the aforementioned approach, as well as other direct methods, such as GAIO. It is interesting to note that our proposed method is actually the result of a very fundamental consideration that can in fact be directly traced back to the very roots of how the structures of invariant sets are being understood in the simplest specific cases, such as the Horseshoe mapping, or one-dimensional quadratic maps, see e.g. [10]. In this sense, the main contribution of this paper is turning the conceptually most basic considerations that are routinely being performed in text books into computationally tractable iterative procedures by leveraging the powerful and ubiquitous sample-based viewpoint. Another key feature is that, like in the Koopman operator theoretic methods, the proposed method in principle only requires data snapshots X_1 and X_2, and no further knowledge of the mapping F, be it in terms of an analytic expression or the ability to evaluate F at any given point.

In the next section, we first describe our new sample-based methodology for reconstructing on an illustrative example. We then discuss the theoretical aspects associated with it and also provide some further discussion on a more efficient implementation of the basic idea, as well as some further illustrative examples.

2 Illustration of the main idea

We consider the example of the well-studied Hénon map given by

$$F(x_1, x_2) = \begin{pmatrix} 1 - ax_1^2 + x_2 \\ bx_1 \end{pmatrix},$$

with the choice of parameters $a = 1.4$ and $b = 0.3$, as originally considered in [8]. On the left of Figure 1 we illustrate a set of points $x^{(i)}$ in blue and the corresponding function values $F(x^{(i)})$ in red.

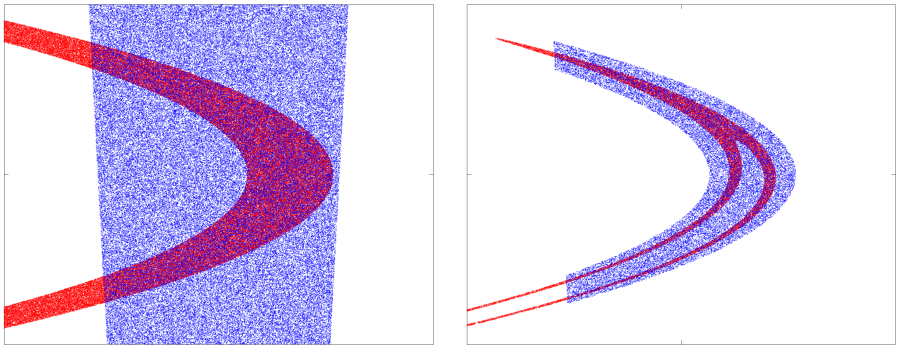

Fig. 1 Left: This figure illustrates a set of blue and red points, where each red point is the function value $F(x^{(i)})$ of a blue point $x^{(i)}$ in the considered box. Right: The blue points for which there are no red points in the vicinity, are trimmed off along with their red counterparts to yield the plot on the right-hand side. The blue set obtained in this way simply resembles the intersection of the blue and red sets on the left-hand side, and the corresponding new red set is starting to reveal a more detailed structure.

We stress that for each blue point $x^{(i)}$, there is a corresponding red point $F(x^{(i)})$ lying in the box $[-3, 3] \times [-0.6, 0.6]$. This fact also explains the particular shape of the blue set, which necessarily tapers towards the bottom so as to not give rise to points of which the corresponding red point would lie outside the box $[-3, 3] \times [-0.6, 0.6]$. In this way, the action of the mapping F is implicitly encoded in these sets of points and in fact not further needed for our algorithm.

Now we observe that there are areas in which there are only blue points and no red points at all, such as in the close vicinity of the origin, and also areas in which there are only red points with no blue points at all, such as towards the left. Now as far as the search for invariant sets in the box $[-3, 3] \times [-0.6, 0.6]$ is concerned, these areas are of no interest and can simply be discarded. In more detail, since the red area is by definition the image of the blue area under F, an area with only blue points and no red points will not be the target of any blue points $x^{(i)}$ after being mapped by F, and thus cannot be a part of a (forward) invariant set.

The right-hand side of Figure 1 shows the result of discarding the pairs of points $(x^{(i)}, F(x^{(i)}))$ for which the $x^{(i)}$ lies in an area with only blue points and no red points. Clearly, the resulting blue set can be recognized as the intersection of the blue and red sets on the left-hand plot of Figure 1. However, since it is *pairs* $(x^{(i)}, F(x^{(i)}))$ of corresponding blue and red points that are thrown away, the new

red set has significantly changed, exposing two branches towards the bottom. This first iteration already describes the main idea of the novel perspective put forth in this paper, namely the repeated trimming off the excess parts to gradually reveal the actual structure of the maximal invariant set. It is an interesting observation that the obtained blue and red sets in the above described process bear a close resemblance with the results obtained from an actual forward simulation. This highlights an apparent *duality* between the process of discarding the pairs $(x^{(i)}, F(x^{(i)}))$ that are in violation of the invariance condition with the process of propagating the set B_0 forward or backward in time, which has not been reported before. This novel duality between deletion of specific pairs $(x^{(i)}, F(x^{(i)}))$ and propagations forward or backward in time is key to our sample-based approach as it essentially allows us to compute forward/backward iterations of sets in a numerically stable way.

Having arrived at the situation on the right-hand side of Figure 1, we can continue to trim off the excess blue parts, or, in an analogous fashion, trim off the red parts. Figure 2 shows the result of applying an alternating trimming of blue excess and red excess parts in very visual terms. Again, as we remove red/blue points with corresponding blue/red counterparts, the iteration is beginning to expose finer and finer structures of the invariant set. We note that our particular practice of keeping track of both the red and blue sets at all steps is particularly helpful in (manually) performing the iteration and allows us to closely follow the procedure as the invariant set is steadily being "excavated". Indeed, seeing the more detailed structures of the invariant set gradually emerge with each iteration can clearly add to our understanding of the structure of the invariant set, much in the same way as in the classical considerations about the invariant set of the Horseshoe map.

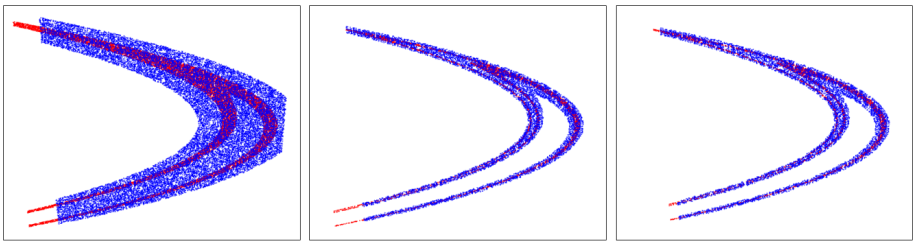

Fig. 2 This figure illustrates the second, third, and forth iteration step, respectively. The left plot results from trimming off the three protruding red ends in the right-hand plot of Figure 1. In the third iteration, the blue excess parts are trimmed again, yielding the result shown in the middle plot. The right-hand plot shows the result of trimming the two protruding red ends from the middle plot.

We find that with as few as four iterations, the obtained result already very clearly resembles the well-known structure of the invariant set of the Hénon map. This is in contrast to the usual dozen forward iterations of the points $\{x^{(i)}\}$ that are required to obtain the same result. Figure 3 shows a close-up view of the end result obtained from repeatedly applying these iterations. Again, an important distinction between the mechanism of the method proposed in this paper and existing approaches is that the invariant set can in principle be computed by leveraging only the initial configuration of samples $(x^{(i)}, F(x^{(i)}))$ and in particular would not require any further forward propagation of points $\{x^{(i)}\}$.

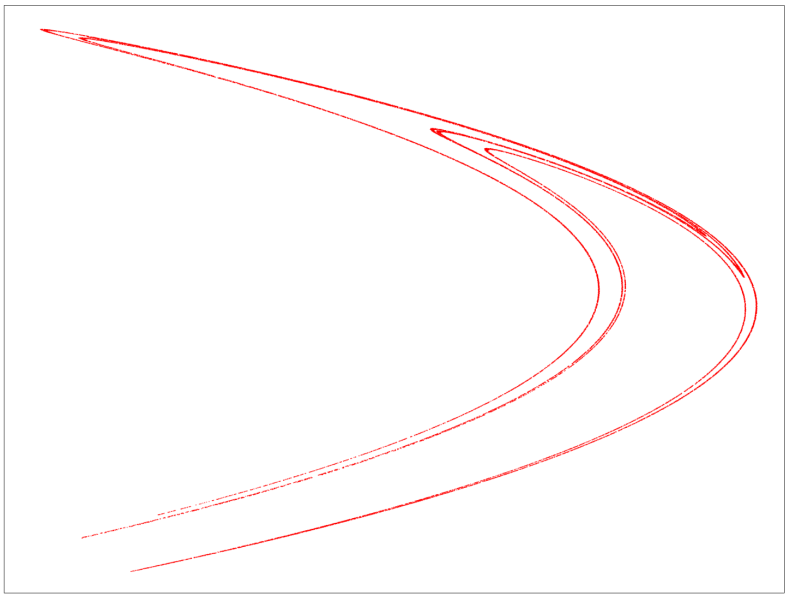

Fig. 3 This figure illustrates the result obtained from the iteration, which is a configuration of points in the shape of the invariant set of the Hénon map, which is slightly larger than the attractor of the Hénon map, which can be obtained from repeated forward propagations.

It turns out that formalizing the above described constructive method to compute the maximal invariant set of the above discrete-time dynamical system within a given set $B_0 \subset X$ (in the case that F is one-to-one) in terms of a set-valued iteration is not as straightforward as one would initially imagine. Perhaps one of the first pitfalls here is the temptation to formulate the iterative procedure as an iteration on the product space $X \times X$, being initialized with the *graph* $\{(x, F(x)) : x \in B_0\}$. While for the numerical implementation the idea to keep track of both the blue and the red set in all steps has proved to be very useful and insightful, for the theoretical study, both the formulation and its proof turn out to be quite cumbersome in that framework. The following theorem provides a resolution, revealing that the most effective description of our novel practical method described in the beginning turns out to be a strikingly simple one.

Theorem 1 *Let X be a set and consider a one-to-one mapping $F : X \to X$. Starting with some initial set $B_0 \subset X$, we use F to generate a sequence $(B_k)_{k\in\mathbb{N}_0}$ of subsets of X by means of the iteration*

$$B_{k+1} := F^{(-1)^k}(B_k) \cap B_k.$$

Then the generated sequence $(B_k)_{k\in\mathbb{N}_0}$ has the following properties:

(i) for any invariant set $I \subset B_0$, it holds that $I \subset B_k$ for all $k \in \mathbb{N}_0$,
(ii) $B_0 \supset B_1 \supset B_2 \supset B_3 \supset \ldots$,
(iii) if there exists $k^\star \in \mathbb{N}$ such that $B_{k+1} = B_k$ for all $k \geq k^\star$, then $F(B_{k^\star}) = B_{k^\star}$.

Properties (ii) and (iii) certify that the sets in the generated sequence are in general shrinking and, if no shrinking takes place, have fully converged to an invariant set of F that, by property (i), is maximal in the sense that it includes all invariant sets I contained in the initial set B_0.

One way one could prove this theorem is through observing that this iteration essentially provides outer approximations of $\mathrm{Inv}(B_0) = \bigcap_{k\in\mathbb{Z}} F^k(B_0)$ by means of intersections over a finite subset of integer indices k, from which all the properties listed in the theorem naturally follow. In more detail, if we let $(c_\ell)_{\ell\in\mathbb{N}}$ denote the sequence of integers $0, -1, 1, -2, 2, -3, 3, \ldots$, or, more explicitly, $c_\ell = \sum_{m=1}^{\ell}(-1)^m m$, then it can be readily verified that the sets B_k obtained by the forward-backward iteration described in Theorem 1 can be more explicitly written as

$$B_k = \bigcap_{\{c_\ell\}_{\ell\leq k}} F^{c_\ell}(B_0).$$

The sets B_k thus clearly constitute outer approximations of the invariant set, which will converge to the invariant set in the Hausdroff distance as $k \to \infty$. In this sense, the key idea and main contribution of this paper may be described as providing a computationally efficient way to compute the "instantaneous" intersection

$$\mathrm{Inv}(B_0) = \bigcap_{k\in\mathbb{Z}} F^k(B_0) = \ldots \cap F^{-2}(B_0) \cap F^{-1}(B_0) \cap B_0 \cap F(B_0) \cap F^2(B_0) \cap \ldots,$$

which is the very definition of an invariant set but clearly cannot be efficiently carried out on a computer. We would like to stress that while it is indeed straightforward to verify that the set-valued iteration $B_{k+1} = F^{(-1)^k}(B_k) \cap B_k$ proposed in this paper in fact just provides an approximation of $\mathrm{Inv}(B_0)$, the converse direction, i.e. the synthesis of the iteration from the definition of $\mathrm{Inv}(B_0)$ is in fact not too natural. Indeed, it is only through our specific viewpoint associated with the idea of deleting pairs $(x^{(i)}, F(x^{(i)}))$ in violation of the invariance condition that led us to consider this particular scheme.

In the following, we present a proof that operates on a very elementary level.

Proof We first show property (i), i.e. that a subset $I \subset B_0$ that is invariant under F will be a subset of any iterate B_k, for any $k \in \mathbb{N}_0$. The proof is by induction. Clearly, we have $I \subset B_0$ by assumption. Now suppose we have $I \subset B_k$ for some $k \in \mathbb{N}_0$. If k is odd, then by the definition of the set iteration we have the iteration $B_{k+1} = F^{-1}(B_k) \cap B_k$. To show that $I \subset B_{k+1}$, it suffices to show that $I \subset F^{-1}(B_k)$, as $I \subset B_k$ holds by the induction hypothesis. Because I is (forward) invariant under F, we have $F(I) \subset I \subset B_k$, which is equivalent to $I \subset F^{-1}(B_k)$. In the case that k is even, we have the iteration $B_{k+1} = F(B_k) \cap B_k$. Analogously to the odd case, we just have to show that $I \subset F(B_k)$. Let $x \in I$. Because I is (backward) invariant under F, we have $F^{-1}(x) \in I$. Using again the induction hypothesis $I \subset B_k$, we conclude that $F^{-1}(x) \in B_k$, or, equivalently, $x \in F(B_k)$. This shows that $I \subset F(B_k)$, and ultimately that $I \subset B_{k+1}$. Property (ii) holds by construction. To show property (iii), assume without loss of generality that $k \geq k^\star$ is odd, in which case we have $B_{k+1} = F(B_k) \cap B_k = B_k$, and, since, in particular, $B_{k+1} = B_k$, $B_{k+2} = F^{-1}(B_k) \cap B_k = B_{k+1} = B_k$. From $F(B_k) \cap B_k = B_k$ it can be inferred that $B_k \subset F(B_k)$. Similarly, from $F^{-1}(B_k) \cap B_k = B_k$, it can be inferred that $B_k \subset F^{-1}(B_k)$, i.e. $F(B_k) \subset B_k$. Thus $F(B_k) = B_k$. □

3 Implementations and Examples

As already suggested in the introductory example, the procedure described in Theorem 1 in terms of a set-valued iteration can in fact be readily put to use by representing the sets in the iteration in terms of samples. In the following, we provide a discussion of different ways to implement this procedure, starting from a naive one to more sophisticated ones.

3.1 Straightforward implementation of Theorem 1

In the most straightforward implementation, the starting point is to draw sufficiently many samples $x^{(i)}$ from the set B_0 to guarantee a uniform covering. We then map each point $x^{(i)}$ forward using F to obtain points $F(x^{(i)})$ while keeping track of the associations between points from B_0 and their images in $F(B_0)$. The subsequent implementation of the set-valued iteration is then as follows:

1. The update step $B_{k+1} = F^{-1}(B_k) \cap B_k$ is computationally realized by going through all the points $x^{(i)}$ and checking whether there are any points $F(x^{(j)})$ within some δ-ball centered at $x^{(i)}$, where δ is chosen appropriately (depending on the denseness of the samples). If for a given $x^{(i)}$, there is no point $F(x^{(j)})$ in the neighborhood of $x^{(i)}$, both $x^{(i)}$ and $F(x^{(i)})$ are discarded.
2. Similarly, the update step $B_{k+1} = F(B_k) \cap B_k$ is computationally realized by going through all the points $F(x^{(i)})$ and checking whether there are any points $x^{(j)}$ in its vicinity. If for a given $F(x^{(i)})$, there is no point $x^{(j)}$ in its neighborhood, both $F(x^{(i)})$ and $x^{(i)}$ are discarded.

The choice of the parameter δ is particularly important in view of the fact that we are not directly working with the sets B_k but only a finite representation of it by means of samples. For instance, one can think of cases where the mapping F is strongly expanding in a specific direction, in which case we can treat the parameter δ as a tolerance that prevents the wrongful deletion of points in areas where the image points $F(x^{(i)})$ are scarce, i.e. widely spread apart. To convey as clearly as possible to the reader the ability of our proposed method to reconstruct the full maximal invariant set rather than only some attracting subset, we first consider the example of a Van der Pol oscillator. This is a case in which a direct forward/backward simulation is clearly not able to yield the maximal invariant set but only the limit cycle or the origin as the equilibrium point. In the first plot of Figure 4, we start with a dense sampling of the square $B_0 = [-3, 3] \times [-3, 3]$ by means of a regular grid with a grid size of 2×10^{-5}, which results in roughly 9 million sample points $x^{(i)}$ for B_0.

Each of these points is then propagated forward with the flow Φ_t of the Van der Pol system with $t = 0.025$, yielding the corresponding red set in the first plot of Figure 4. As one can clearly see, there are nonoverlapping areas of the blue and red sets at the boundary. When trimming these portions off using the above described basic procedure, we witness a gradual transformation of the two sets. The bottom row of Figure 4 shows results after 30, 60, and 90 iterations, respectively. We can recognize that the change in the sets is somewhat reminiscent of the result of a straightforward simulation of the original blue set. Nevertheless, we stress that this trimming procedure is fundamentally different from an actual forward simulation.

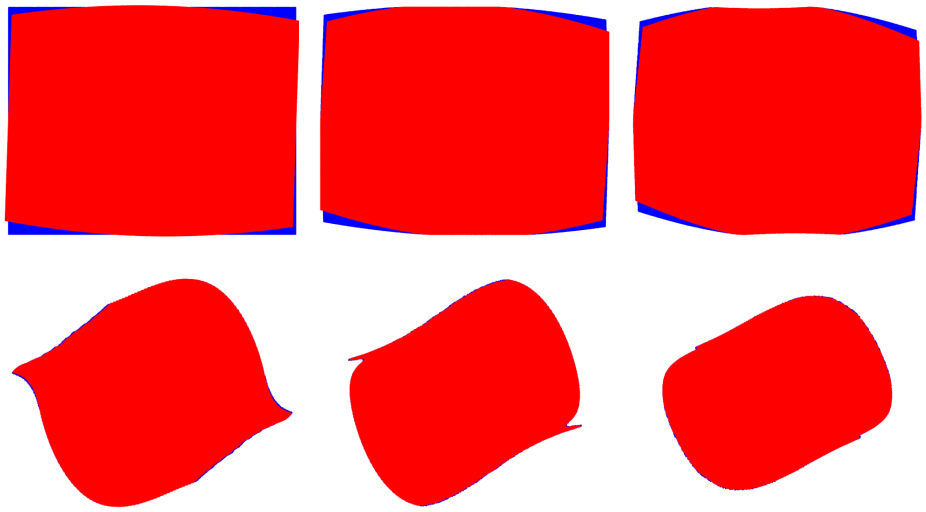

Fig. 4 Top row: The initial configuration and results at the two subsequent stages in the computation of the invariant set for a Van der Pol system. Bottom row: Three further results at different stages in the iterative trimming process. By considering these results of the iterations, we see that the trimming process is in fact reminiscent of a forward propagation of the original set B_0. A high-resolution animation can be found in [1].

First, we only use the forward mapping Φ_t once in the initial step. Secondly, with our trimming procedure, points within the limit cycle will be left untouched and, in particular, will not thin out as they would in a forward simulation. Indeed, for an actual repeated forward propagation (i.e. a repeated application of Φ_t to the set B_0), the points in the limit cycle will all eventually converge to the limit cycle, leading to the area inside the limit cycle to thin out, regardless of how dense the grid for B_0 is chosen.

This again highlights the advantage of the dual, but numerically more favorable, stable procedure of deleting pairs of points as opposed to actual forward or backward simulations of the dynamics. Moreover, through the sample-based methodology, we are able to clearly see the intermediate steps in the forward-backward iteration, which yields additional valuable insights into the dynamics of the system, as well as the structure of invariant sets.

From a computational point of view, however, this uniform sampling strategy in the basic implementation may be regarded as inefficient, particularly for specific cases where the invariant set has a very fine structure, i.e. occupies only a small portion within the set B_0, such as in the case of the Hénon map. There, the number of sample points in $B_0 \setminus I$ would be quite large relative to the total number of sample points for B_0, causing an in fact unnecessary computational overhead. A more intelligent approach to the reconstruction of the invariant sets with a more delicate structure would be an "adaptive resampling strategy" where the rough "location" of the invariant set is first probed with a coarse sampling which is then gradually refined only in the areas that show signs of actually containing the invariant set. This strategy is presented in the next subsection.

3.2 A computationally more efficient variant

To illustrate the need for a more economic implementation of the general sample-based idea, we consider the Hénon map with parameters $a = 2.0$ and $b = 0.3$. On the left of Figure 5, we employed a coarse initial sampling of B_0, as well as the corresponding image points. Even though the sampling is very coarse, it is not hard to identify the overall situation. In particular, we can rather easily identify large areas with only blue points and no red points. Any sample point that we would have chosen in those areas would in fact be better invested in the areas with blue and red points, which would potentially increase the achievable resolution for the invariant set. This is the basic idea for the following adaptive resampling strategy:

1. Probe the whole set B_0 with a rather coarse (computationally inexpensive) sampling, by which the often much concentrated relevant areas are revealed.
2. These areas are then further probed with a slightly finer grid, i.e. by sampling with a higher resolution points $x^{(i)}$ (and computing the counterparts $F(x^{(i)})$) in the areas that contain both blue and red points (and only those areas).
3. The second step is then repeated.

Practically, this is implemented along the lines of the previous approach: We go through all red points and see whether there are any blue points within a ball of radius δ centered about the red points. Again δ is a parameter that can be adjusted based on the degree of compression/expansion of the map F. For the Hénon map considered here, it is well-known that there is a strong compression in the horizontal direction, as can also be seen in the first plot of Figure 5. The second plot of Figure 5 shows the result of "repopulating" the area containing both blue and red points with sample points of a grid with a finer resolution. Again, we also keep track of the corresponding red image points. By repeating this process, we obtain a very attractive adaptive and stable way for carrying out the iteration $B_{k+1} = F(B_k) \cap B_k$, cf. the other steps in Figure 5.

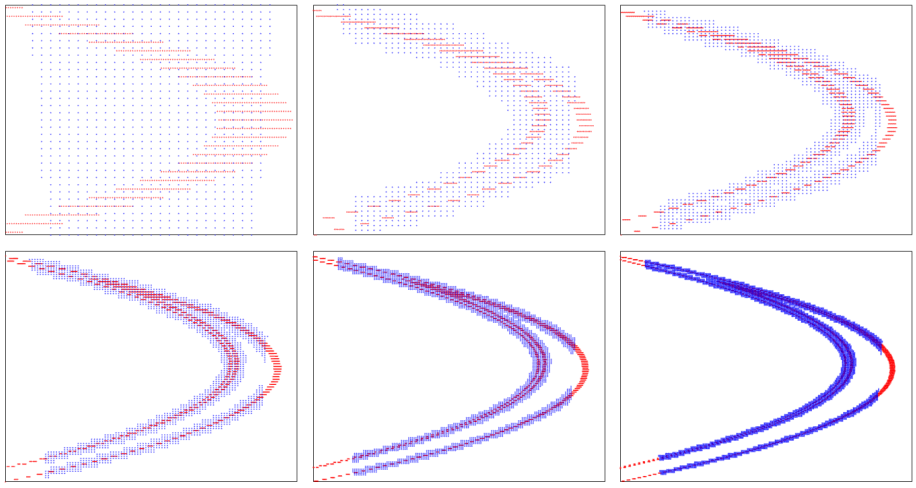

Fig. 5 This figure illustrates the gradual regrowing of areas containing both blue and red points. Starting from a coarse sampling of B_0 and its image $F(B_0)$, the grid size for the sample points in a subsequent step increases.

It is interesting to note that even though we only repopulate the blue points $x^{(i)}$ in areas with blue and red points, we keep ending up with red areas that are conspicuously protruding out from the blue areas, which is a characteristic feature of the particular Hénon map considered in this subsection. A counter measure for this situation is our familiar deletion process. By "manually" discarding those red points protruding from the blue support structure (along with their blue counterparts), we can stop these redundant red points from reoccurring. Figure 6 shows the trimming process applied to the sample configuration of the last plot of Figure 5.

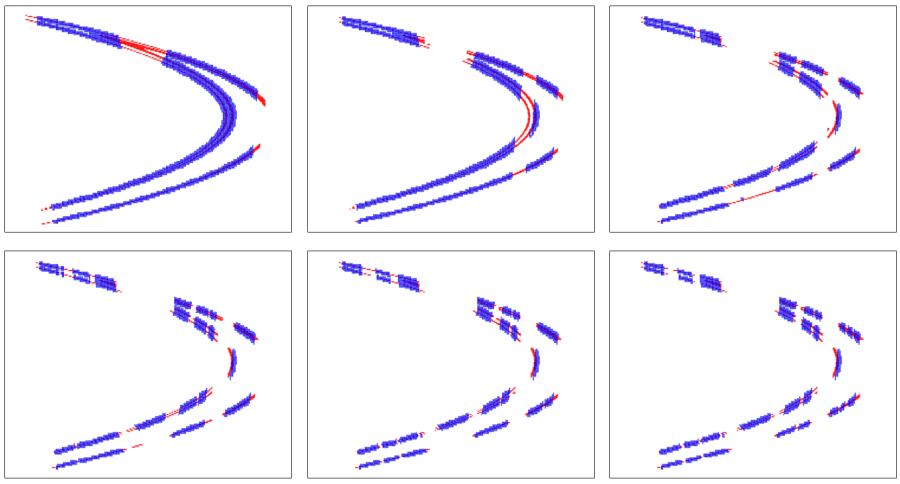

Fig. 6 The first plot illustrates the result of deleting the protruding pieces of the red arch witnessed in the plots of Figure 5 along with their corresponding blue points, which were evidently located in the upper middle part. The deletion of the blue points in the upper middle part exposes red points that can be further trimmed, and so on.

After the trimming process, we can continue to apply an arbitrary combination of the regrowing process, to obtain a finer resolution of the remaining structures, and the trimming process, to remove redundant parts. This procedure is summarized slightly more formally in the appendix. For the actual implementation, as well as a high-resolution animation of an alternating application of the regrowing and trimming process, we refer to [1].

By repeatedly applying this computationally efficient procedure, we are in fact able to produce reconstructions of invariant sets with unprecedented resolutions. This fact is highlighted in Figure 7, where we show the result of repeatedly applying the regrowing and trimming iteration, resulting in the reconstruction of the invariant set of the Hénon map with $a = 2.0$ and $b = 0.3$ by means of about 45 million sample points. The bottom plot of Figure 7 shows a zoom on the left part of the two pairs close to the red arrow, revealing the fractal structure of the invariant set of the considered Hénon map very clearly.

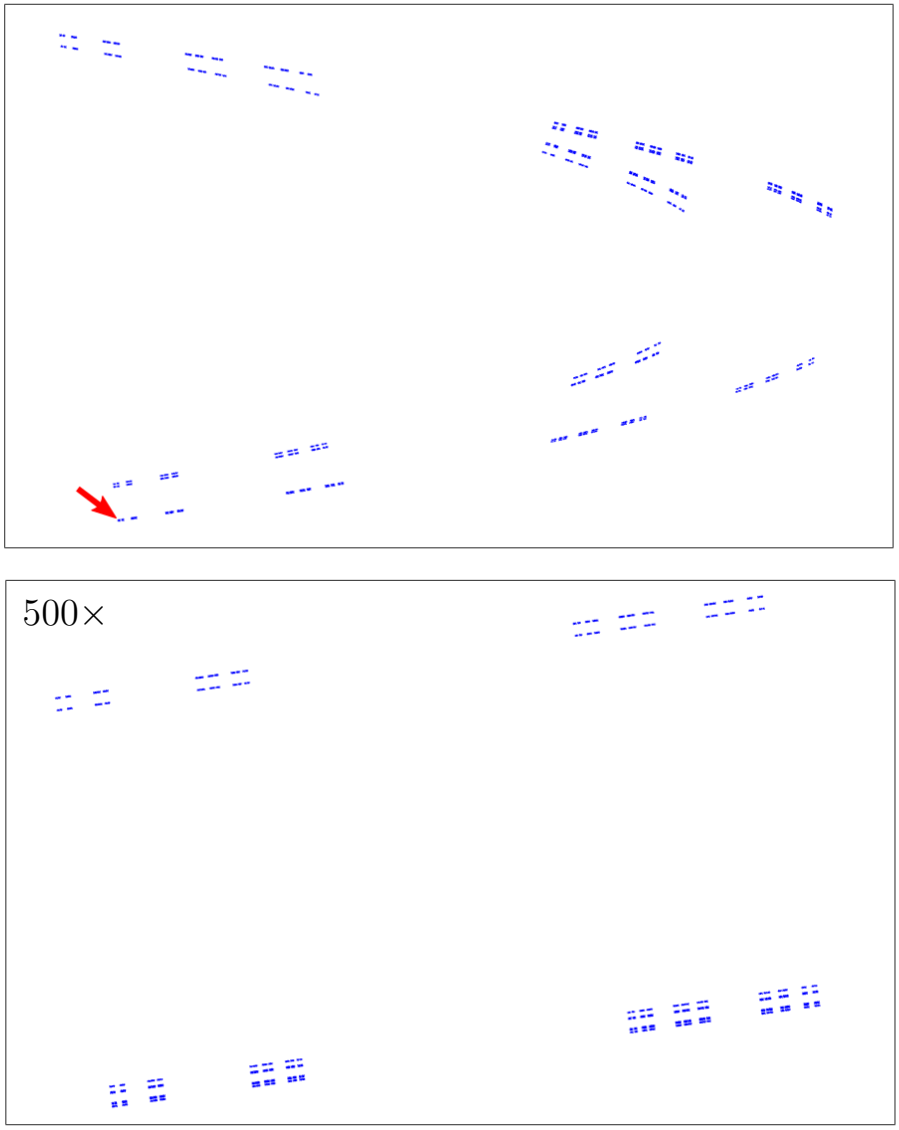

Fig. 7 Top: Maximal invariant set of the Hénon map with $a = 2.0$ and $b = 0.3$ in $B_0 = [-1.2, 1.2] \times [-0.35, 0.35]$. The red arrow indicates the zoomed area shown in the bottom plot.

The procedures presented in this section are all readily applicable to higher dimensions. To illustrate a reconstruction via the adaptive resampling scheme for a three-dimensional example, we consider the mapping

$$F(x_1, x_2, x_3) = \begin{pmatrix} x_2 \\ x_3 \\ 1.4 + 0.1x_1 + 0.3x_2 - x_3^2 \end{pmatrix}.$$

This is another example in which it is not possible to obtain a very clear view of the (forward) invariant set, which is a chaotic saddle, through a simple repeated application of forward iterations. Figure 8 illustrates the gradual growing procedure at six subsequent steps.

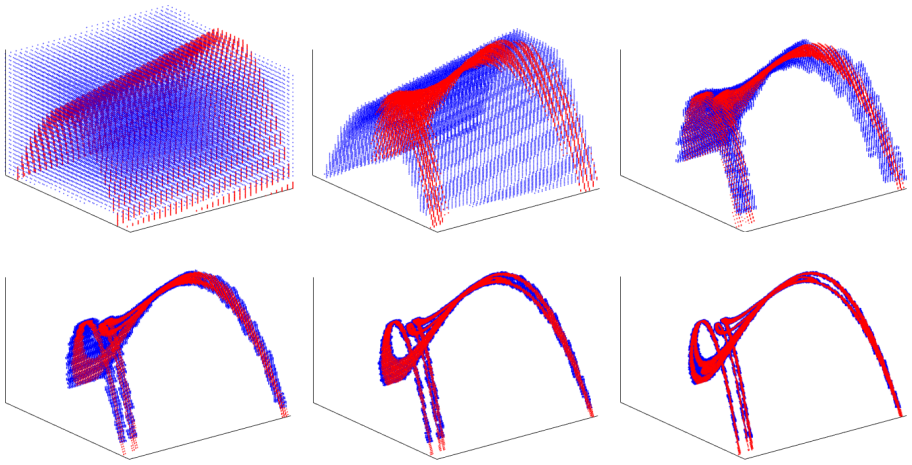

Fig. 8 Six steps in the growing process for the three-dimensional system.

The end result is shown in Figure 9, revealing the forward invariant set in an again unprecedented resolution.

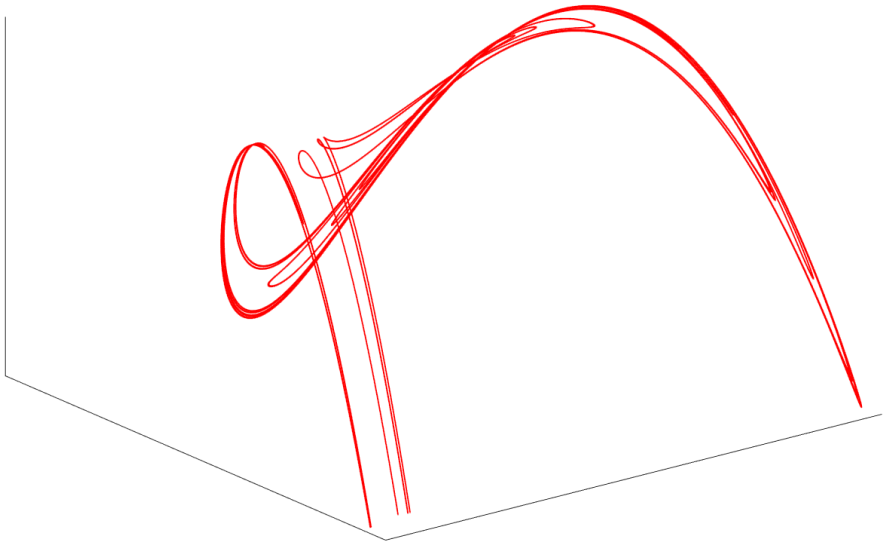

Fig. 9 The forward invariant set of the three-dimensional system obtained from repeatedly applying the growing procedure.

3.3 Discussion of the sample-based methodology

The quite straightforward implementation as well as the different types of results and insights obtained in the three considered examples have clearly highlighted key strengths of the proposed sample-based methodology. In particular, the idea to keep track of the pairs $(x^{(i)}, F(x^{(i)}))$ rather than outer approximations in terms of somewhat less transparent boxes in related approaches, such as GAIO, provides us with a deeper understanding of the overall situation as it fully illuminates the dynamic processes taking place "inside the boxes". For instance, the novel insight about the very useful equivalence of the deletion process for the samples with forward/backward propagation is purely the result of those considerations. Moreover, by being able to follow the iterative process of how the invariant set is gradually being revealed as the excess parts are being shaved off indirectly also provides us with a better understanding of the overall composition of the maximal invariant set, as well as the dynamics of the considered system. The example of the Van der Pol oscillator, in particular, illustrated the dynamics in the process of removing pairs $(x^{(i)}, F(x^{(i)}))$ that violate the invariance condition very clearly.

From a computational perspective, the illustrated duality between deletion of points and forward/backward propagations of sets is also very useful for two particular reasons. First, as we saw in example of the Van der Pol system, the deletion process provides a stable way to propagate a set forward/backward in time, the evolution of which can be clearly followed in each iteration step. Secondly, by deleting pairs of points in iterations, we are reducing the number of particles that we have to work with, which results in a reduced computation time. Indeed, a direct comparison with GAIO revealed that the proposed sample-based implementation yields a more accurate result of the invariant set in a shorter time.

More specifically, for the example of the Hénon map illustrated in Figure 6, an initial coarse 25×25-grid is chosen, resulting in approximately 500 sample points for describing B_0. The application of 16 combined regrowing and trimming steps with an increasing grid resolution of a factor of 1.5, resulting in a final resolution of approximately 16000×16000, took 29 seconds using a desktop computer with an Intel Core i7-7700 CPU (3.60GHz) and 16 GB RAM. For comparison, an application of GAIO to obtain a reconstruction of the invariant set with the same resolution took 40 seconds. In addition to the complete insight into the structure of the invariant set and the computational advantages of the proposed sample-based methodology, we stress the added degree of flexibility, such as the ability to increase the resolution in each step by an arbitrary factor, as opposed to the somewhat fixed factor of 2 in GAIO, as well as the choice regarding the applications of the regrowing/trimming procedures in each iteration step.

4 Summary and discussion

In this paper, we have introduced a new perspective for the reconstruction of the maximal invariant set of discrete-time dynamical systems, which originates from the basic idea of viewing/plotting the sets B_k *along with the images* $F(B_k)$ alongside each other in one picture. While this idea may appear to be quite elementary on first sight, it turns that it is indeed a novel viewpoint that provides completely new insights, as well as new avenues for tackling the practical problem of computing invariant sets. In particular, the insights obtained from the process of deleting *pairs* of sample points $(x^{(i)}, F(x^{(i)}))$ in violation of the invariance condition, i.e. pairs of points affiliated with the sets $B_k \setminus (B_k \cap F(B_k))$ and $F(B_k) \setminus (B_k \cap F(B_k))$, respectively, are completely new.

We discussed the practical implementation of this basic method, as well as a more efficient variant thereof, and illustrated those on three well-studied problems. This in particular highlighted the efficiency, and simplicity of the implementations. It is perhaps somewhat surprising that this very efficient computational method that we found can in fact be traced back to the most basic considerations of the study of invariant sets that are described in countless introductory text books on the theory of dynamical systems, though only for very specific dynamical systems, such as the Horseshoe map. A main contribution of this paper is having highlighted how the basic conceptual idea, which is at the core of the very definition of an invariant set, can in fact be effectively put to use to arbitrary discrete-time dynamical systems.

The key step towards this direction, which is the sample-based approach, is inspired by recent developments in the study of Koopman operator theoretic approaches to the analysis of dynamical systems, which has been parallelly taking place to the developments of Frobenius-Perron operator theoretic approaches for uncovering invariant measures of dynamical systems. One of the main strengths of the proposed methodology is the simplicity on both a conceptual and computational level, which was illustrated in several numerical examples. With the recent interest in operator-theoretic methods leveraging data snapshots in mind, we conclude that the methodology presented in this paper may serve as one of many examples of a fruitful application of sample-based methods that may be further combined with other recent developments such as [9] to the study of complex dynamical systems.

5 Appendix – Pseudocode for the computationally efficient variant

Given a set of samples $X = \{x^{(1)}, \dots, x^{(N)}\}$, we denote $\mathrm{cover}(X, \delta) := \bigcup_{i=1}^{N} B_\delta(x^{(i)})$, where $B_\delta(x) := \{y \in \mathbb{R}^n : d(x, y) < \delta\}$. The more efficient computational procedure described in Section 3.2 can then be formulated as follows.

Algorithm 1 Efficient sample-based computation of invariant sets

1: Initialization: Probing the set B_0 with a coarse sampling $\delta_0 > 0$ and a set of samples $X_0 := \{x_0^{(1)}, \dots, x_0^{(N_0)}\}$ such that

$$\mathrm{cover}(X_0, \delta_0) \supset B_0 \text{ and } \mathrm{cover}(F(X_0), \delta_0) \supset F(B_0).$$

2: Repeat

Regrowing step: Given $\delta_k > 0$ and $X_k = \{x_k^{(1)}, \dots, x_k^{(N_k)}\}$, find $\delta_{k+1} < \delta_k$ and a set of samples $X_{k+1} = \{x_{k+1}^{(1)}, \dots, x_{k+1}^{(N_{k+1})}\}$ such that

$$\begin{aligned} &\mathrm{cover}(X_{k+1}, \delta_{k+1}) \supset \mathrm{cover}(X_k, \delta_k) \cap \mathrm{cover}(F(X_k), \delta_k) \\ \wedge\ &\mathrm{cover}(F(X_{k+1}), \delta_{k+1}) \supset F(\mathrm{cover}(X_k, \delta_k)). \end{aligned}$$

Deletion step: Save all $i \in \{1, \dots, N_{k+1}\}$ for which

$$\begin{aligned} &x^{(i)} \notin \mathrm{cover}(X_{k+1}, \delta_{k+1}) \cap \mathrm{cover}(F(X_{k+1}), \delta_{k+1}) \\ \vee\ &F(x^{(i)}) \notin \mathrm{cover}(X_{k+1}, \delta_{k+1}) \cap \mathrm{cover}(F(X_{k+1}), \delta_{k+1}), \end{aligned}$$

and delete the corresponding $x_{k+1}^{(i)}$ from X_{k+1}.

3: Until desired resolution is reached.

The author declares no conflict of interest.

References

1. `https://systemstheorylab.wustl.edu/invariant-sets/`.
2. Hassan Arbabi and Igor Mezic. Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the Koopman operator. *SIAM Journal on Applied Dynamical Systems*, 16(4):2096–2126, 2017.
3. Steven L Brunton, Bingni W Brunton, Joshua L Proctor, Eurika Kaiser, and J Nathan Kutz. Chaos as an intermittently forced linear system. *Nature communications*, 8(1):19, 2017.
4. Marko Budišić, Ryan Mohr, and Igor Mezić. Applied Koopmanism. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(4):047510, 2012.
5. Michael Dellnitz, Gary Froyland, and Oliver Junge. The algorithms behind GAIO – set oriented numerical methods for dynamical systems. In *Ergodic theory, analysis, and efficient simulation of dynamical systems*, pages 145–174. Springer, 2001.
6. Michael Dellnitz and Andreas Hohmann. A subdivision algorithm for the computation of unstable manifolds and global attractors. *Numerische Mathematik*, 75(3):293–317, 1997.
7. Michael Dellnitz and Oliver Junge. On the approximation of complicated dynamical behavior. In *The Theory of Chaotic Attractors*, pages 400–424. Springer, 1999.

8. Michel Hénon. A two-dimensional mapping with a strange attractor. In *The Theory of Chaotic Attractors*, pages 94–102. Springer, 1976.
9. Oliver Junge and Ioannis G Kevrekidis. On the sighting of unicorns: A variational approach to computing invariant sets in dynamical systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(6):063102, 2017.
10. Anatole Katok and Boris Hasselblatt. *Introduction to the modern theory of dynamical systems*, volume 54. Cambridge university press, 1997.
11. Igor Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41(1-3):309–325, 2005.
12. Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3(4):e1602614, 2017.
13. Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.
14. Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. A data–driven approximation of the Koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015.