Exploring Invariant Sets and Invariant Measures

Michael Dellnitz*, Andreas Hohmann†, Oliver Junge‡, Martin Rumpf§

September 5, 1997

Abstract

We propose a method to explore invariant measures of dynamical systems. The method is based on numerical tools which directly compute invariant sets using a subdivision technique, and invariant measures by a discretization of the *Frobenius-Perron operator*. Appropriate visualization tools help to analyze the numerical results and to understand important aspects of the underlying dynamics. This will be illustrated for examples provided by the Lorenz system.

*Mathematisches Institut, Universität Bayreuth, D-95440 Bayreuth
†Konrad-Zuse-Zentrum für Informationstechnik Berlin, D-10711 Berlin
‡Mathematisches Institut, Universität Bayreuth, D-95440 Bayreuth
§Institut für Angewandte Mathematik, Universität Freiburg, D-79104 Freiburg

1 Introduction

Let a certain dynamical system — in form of an ordinary or partial differential equation, say — be given. Suppose that this system has the potential to exhibit very complicated dynamical behavior, and that one wants to study this behavior numerically. Then, typically, one would choose some initial condition and integrate the system for a long period of time to extract the desired information out of the data provided by this single trajectory.

However, important aspects of long term dynamical behavior are characterized by two mathematical objects: an *invariant set* and a corresponding (natural) *invariant measure* supported on this set. From this point of view it seems more adequate to approximate these two objects directly rather than to compute single trajectories. Indeed, it is easy to construct situations in which the computation of a single trajectory may be misleading in the sense that the results do not reflect correctly the real dynamical behavior.

To simplify the statements we consider discrete dynamical systems,

$$x_{j+1} = f(x_j), \quad j = 0, 1, 2, \dots ,$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is a diffeomorphism. Recall that a set $A \subset \mathbb{R}^n$ is *invariant* if $f(A) = A$. The simplest example of an invariant set is a fixed point of f. An *invariant measure* μ is a probability measure on the Borel σ-algebra $\mathcal{B}(\mathbb{R}^n)$ such that

$$\mu(B) = \mu(f^{-1}(B)) \quad \text{for all } B \in \mathcal{B}(\mathbb{R}^n). \tag{1.1}$$

For instance, the Dirac measure supported on a fixed point of the diffeomorphism f is an invariant measure.

In this paper we approximate both invariant sets and invariant measures directly, that is, we do not compute single trajectories for a long period of time. Rather we use the following two methods:

1. A subdivision algorithm for the outer approximation of an invariant set A by a box covering as recently developed in [1]. The underlying idea for this method is to construct a sequence of box coverings $\mathcal{B}_k$ shrinking down to the invariant set for $k \to \infty$.

2. A discretization of the *Frobenius-Perron operator* for the approximation of the natural invariant measure on A. The idea is to view invariant measures as fixed points of this operator, and in the numerical treatment one has to solve a corresponding eigenvalue problem. The convergence of this method

is guaranteed for so-called *expanding maps* which possess absolutely continuous invariant measures (see [13]). More recently it has been shown that also *SBR-measures* can be approximated via certain random perturbations of the underlying system (see [11, 3]).

However, once the invariant measure has been computed (up to a certain accuracy), the question arises how to make use of the enormous amount of data. Perhaps the most appealing way is to visualize the information, and, in fact, the main purpose of this paper is to propose a method for the visualization of invariant measures based on the two numerical techniques mentioned above. In particular, we do not present the mathematical details which are needed to prove convergence of the two algorithms. But we will sketch some qualitative aspects and conjectures underlined by corresponding visual insights.

Three different types of graphical representations of the invariant sets and measures will be presented. These methods make use of the hierarchical structure of the data already produced in the numerical computation. This allows a flexible and – depending on the chosen level of interest from the hierarchy of data – interactive impression of the characteristics of the dynamical behavior on a graphic workstation.

A more detailed outline of the paper is as follows. In Sec. 2 we briefly summarize the subdivision algorithm as developed in [1]. The discretization of the Frobenius-Perron operator is discussed in Sec. 3, and Sec. 4 contains aspects concerning the visualization. Finally, in Sec. 5, we illustrate the visualization by several examples using the well known Lorenz system.

2 The Subdivision Algorithm

In order to make this paper self-contained we describe the numerical method that we use for the approximation of the invariant sets under consideration. The reader is refered to [1] for a more detailed description.

Description of the Algorithm

The central object which is approximated by the subdivision algorithm is the so-called relative global attractor:

DEFINITION 2.1 Let $Q \subset \mathbb{R}^n$ be a compact set. We define the *global attractor relative to* Q by

$$A_Q = \bigcap_{j \geq 0} f^j(Q). \tag{2.1}$$

Observe that $f^{-1}(A_Q) \subset A_Q$, but A_Q is not necessarily an invariant set. But, by construction, A_Q always contains all the attracting sets which are inside Q. In particular, if Q contains the global attractor A of f then $A_Q = A$.

The subdivision algorithm for the computation of relative global attractors generates a sequence $\mathcal{B}_0, \mathcal{B}_1, \mathcal{B}_2, \ldots$ of finite collections of compact subsets of $\mathbb{R}^n$ with the property that for all integers k the set

$$Q_k = \bigcup_{B \in \mathcal{B}_k} B$$

is a covering of the relative global attractor under consideration. Moreover the sequence of coverings is constructed in such a way that the diameter

$$\operatorname{diam}(\mathcal{B}_k) = \max_{B \in \mathcal{B}_k} \operatorname{diam}(B)$$

converges to zero for $k \to \infty$.

Let us be more precise. Given an initial collection $\mathcal{B}_0$, one inductively obtains $\mathcal{B}_k$ from $\mathcal{B}_{k-1}$ for $k = 1, 2, \ldots$ in two steps.

1. *Subdivision:* Construct a new collection $\hat{\mathcal{B}}_k$ such that

$$\bigcup_{B \in \hat{\mathcal{B}}_k} B = \bigcup_{B \in \mathcal{B}_{k-1}} B$$

and

$$\operatorname{diam}(\hat{\mathcal{B}}_k) \leq \theta \operatorname{diam}(\mathcal{B}_{k-1})$$

for some $0 < \theta < 1$.

2. *Selection:* Define the new collection $\mathcal{B}_k$ by

$$\mathcal{B}_k = \left\{ B \in \hat{\mathcal{B}}_k : f^{-1}(B) \cap \hat{B} \neq \emptyset \quad \text{for some} \quad \hat{B} \in \hat{\mathcal{B}}_k \right\}.$$

We now formulate the result which establishes the convergence of the algorithm to the relative global attractor (for a proof see [1]).

PROPOSITION 2.2 *Let A_Q be the global attractor relative to the compact set Q, and let $\mathcal{B}_0$ be a finite collection of closed subsets with $Q_0 = \bigcup_{B \in \mathcal{B}_0} B = Q$. Then*

$$\lim_{k \to \infty} h\left(A_Q, Q_k\right) = 0,$$

where we denote by $h(B, C)$ the usual Hausdorff distance between two compact subsets $B, C \subset \mathbb{R}^n$.

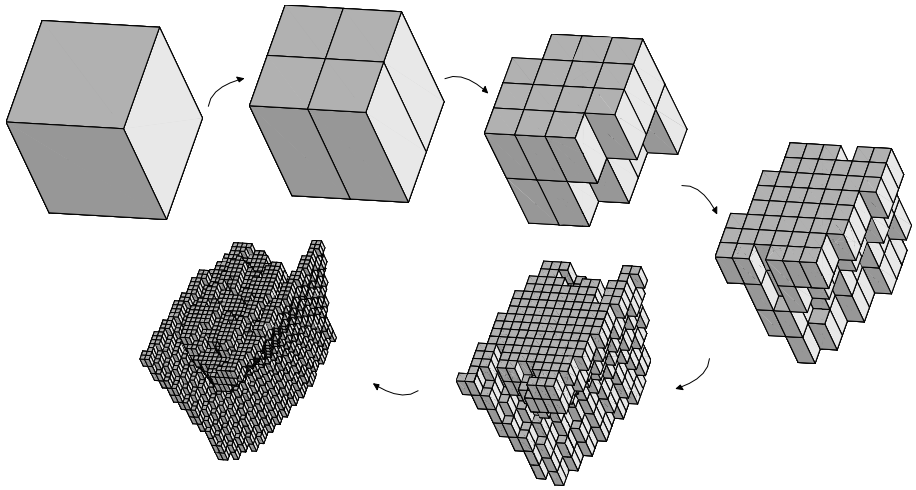


Figure 1: A typical sequence of coverings of a relative global attractor.

As a consequence of Proposition 2.2 we may in principle approximate any relative global attractor up to a prescribed accuracy. The speed of convergence depends on the rate of attractivity. See again [1] for error estimates based on the existence of a hyperbolic structure on A_Q. In the realizaton of the algorithm we have to work with a certain type of sets $B \in \mathcal{B}_k$ and a specific subdivision strategy. It turns out to be particularly useful to work with generalized rectangles R in $\mathbb{R}^n$, i.e. hexaeder in $\mathbb{R}^3$. A typical sequence of coverings obtained by the subdivision algorithm is shown in Figure 1.

Remark 2.3 Based on the subdivision algorithm a continuation method has been developed in [2] which allows to approximate directly invariant manifolds. We will use this method in Sec. 5 to obtain a covering of the closure of an unstable manifold in the Lorenz system.

Storage of the Boxes

The efficiency of the subdivision algorithm and the visualization techniques significantly depends on the specific storage scheme for the boxes $R \in \mathcal{B}_k$ used in our implementation. This scheme leads to sparse storage requirements in particular for very large box collections which give a suitable fine approximation of the considered invariant sets. Therefore let us now describe the corresponding data structure in more detail.

The closed subsets constituting the collections $\mathcal{B}_k$ are generalized rectangles of the form

$$R(c,r) = \{y \in \mathbb{R}^n : |y_i - c_i| \leq r_i \text{ for } i = 1, \ldots, n\},$$

where $c, r \in \mathbb{R}^n$, $r_i > 0$ for $i = 1, \ldots, n$, are the center and the radius respectively.

We start the subdivision algorithm with a single rectangle $\mathcal{B}_0 = \{R\}$. In the k-th subdivision step we subdivide each rectangle $R(c, r) \in \mathcal{B}_k$ of the current collection by bisection with respect to the j-th coordinate, where j is varied cyclically, that is, $j = ((k-1) \mod n) + 1$. This division leads to two rectangles

$$R_-(c^-, \hat{r}) \quad \text{and} \quad R_+(c^+, \hat{r}), \tag{2.2}$$

where

$$\hat{r}_i = \begin{cases} r_i & \text{for } i \neq j \\ r_i/2 & \text{for } i = j \end{cases}, \quad c_i^{\pm} = \begin{cases} c_i & \text{for } i \neq j \\ c_i \pm r_i/2 & \text{for } i = j \end{cases}.$$

The collections $\mathcal{B}_k$ can easily be stored in a binary tree. Figure 2 shows the representation of three subdivision steps in three-dimensional space ($n = 3$) together with the corresponding sets Q_k, $k = 0, 1, 2, 3$. Note that each $\mathcal{B}_k$ and the corresponding covering Q_k are completely determined by the tree structure, and the initial box $R(c, r)$. Using this scheme, the memory requirements grow only linear in the dimension n of the problem.

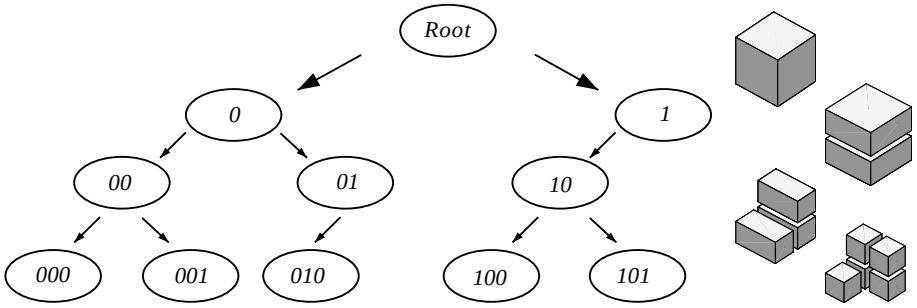


Figure 2: Storage scheme for the collections $\mathcal{B}_k$, $k = 0, 1, 2, 3$.

Furthermore, the tree structure makes it easy to retrieve adjacency information needed in the visualization. For $R \in \mathcal{B}_k$ let $F_{i,\pm}(R)$ denote the faces

$$F_{i,\pm}(R) = \{y \in R \ : \ y_i = c_i \pm r_i\}$$

(cf. Figure 3) and let $N_{i,\sigma}(R)$ be the adjacent box of R in $\mathcal{B}_k$ at face $F_{i,\sigma}(R)$ for $\sigma \in \{+, -\}$ and $i = 1, \cdots, n$. For a box $R \in \mathcal{B}_k$ we are then able to express adjacent boxes of its children R_+, R_- in terms of children of adjacent boxes of R as follows ($\sigma, \tau \in \{+, -\}$):

$$N_{i,\sigma}(R_\tau) = \begin{cases} R_{-\tau} & \text{if} \quad i = ((k-1) \mod n) + 1 \quad \text{and} \quad \sigma \neq \tau \\ (N_{i,\sigma}(R))_{-\tau} & \text{if} \quad i = ((k-1) \mod n) + 1 \quad \text{and} \quad \sigma = \tau \\ (N_{i,\sigma}(R))_{\tau} & \text{if} \quad i \neq ((k-1) \mod n) + 1 \end{cases}$$

(see also (2.2)). This allows us to track the adjacency information procedurally in the tree structure during runtime and hence we do not need to store it.

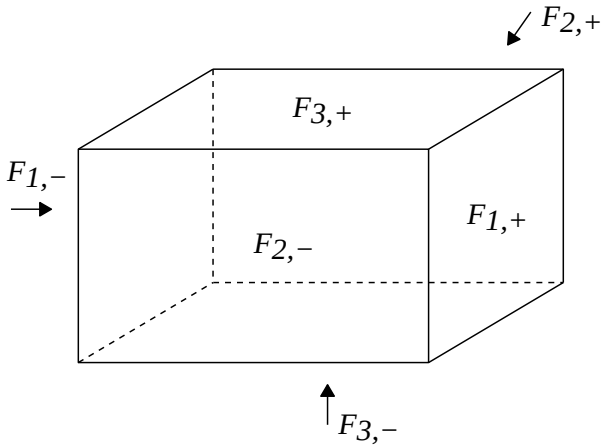


Figure 3: A single box R for $n = 3$ and its faces $F_{i,\sigma}$, $i = 1, 2, 3, \sigma \in \{+, -\}$.

3 Approximation of the invariant measure

Now we briefly describe the numerical method that we have used for the approximation of invariant measures.

Frobenius-Perron Operator

In the following we denote by $\mathcal{B}(\mathbb{R}^n)$ the Borel σ-algebra of $\mathbb{R}^n$ and by $\mathcal{M}$ the space of probability measures on $\mathbb{R}^n$.

DEFINITION 3.1 The operator $P : \mathcal{M} \to \mathcal{M}$ defined by

$$(P\mu)(B) = \mu(f^{-1}(B)) \quad \text{for all } B \in \mathcal{B}(\mathbb{R}^n) \tag{3.1}$$

is called the *Frobenius-Perron operator*.

Observe that it immediately follows from the definitions that an invariant measure μ is a fixed point of the Frobenius-Perron operator P, that is,

$$P\mu = \mu \quad \text{for all invariant measures } \mu \in \mathcal{M}. \tag{3.2}$$

More precisely the calculation of invariant measures is equivalent to the calculation of fixed points of P.

Numerical Approximation of P

To discretize the Frobenius-Perron operator $P : \mathcal{M} \to \mathcal{M}$ we have to project $\mathcal{M}$ onto some set $\mathcal{M}_h$, which can be dealt with numerically. The "stepsize" h should be

interpreted as a measure for the approximation quality which is supposed to increase for decreasing h.

Let $B_i \in \mathcal{B}_k$, $i = 1, \ldots, N$, denote the boxes in the covering obtained after k steps in the subdivision algorithm and let h be the diameter of $\mathcal{B}_k$. We choose $\mathcal{M}_h$ to be the set of density functions which are constant on each B_i, $i = 1, \ldots, N$, that is,

$$\mathcal{M}_h = \left\{ u \in \mathbb{R}^N : u_i \geq 0 \quad \text{and} \quad \sum_{i=1}^{N} u_i = 1 \right\}.$$

Then the discretized Frobenius-Perron operator $P_h : \mathcal{M}_h \to \mathcal{M}_h$ is given by

$$v = P_h u, \quad v_i = \sum_{j=1}^{N} \frac{m(f^{-1}(B_i) \cap B_j)}{m(B_j)} u_j, \quad i = 1, \ldots, N,$$

where m denotes the Lebesgue measure. Correspondingly we have the matrix representation

$$P_h = (p_{ij}), \quad \text{where } p_{ij} = \frac{m(f^{-1}(B_i) \cap B_j)}{m(B_j)} \text{ for } 1 \leq i, j \leq N. \tag{3.3}$$

By (3.2), an approximation μ_h of the density of an invariant measure μ is represented by a normalized eigenvector $u_h \in \mathcal{M}_h$ of P_h associated with the eigenvalue one.

The crucial point in the computation of the discretized Frobenius-Perron operator is the efficient computation of the transition probabilities p_{ij}. The denominator poses no problem since we have chosen the B_j's to be generalized rectangles. Hence we just have to approximate the Lebesgue measure of $f^{-1}(B_i) \cap B_j$, that is, the measure of the subset of B_j that is mapped into B_i. In the computations we have used the Monte Carlo approach as described in [10]. We view $m(f^{-1}(B_i) \cap B_j)$ as the integral

$$m(f^{-1}(B_i) \cap B_j) = \int_{B_j} \chi_{f^{-1}(B_i)} \, dm,$$

and approximate it by a finite sum

$$\frac{1}{m(B_j)} \int_{B_j} \chi_{f^{-1}(B_i)} \, dm \approx \frac{1}{K} \sum_{k=1}^{K} \chi_{f^{-1}(B_i)}(x_k) = \frac{1}{K} \sum_{k=1}^{K} \chi_{B_i}(f(x_k)),$$

where the x_k's are selected at random in B_j from a uniform distribution. Evaluation of $\chi_{B_i}(f(x_k))$ only means that we have to check whether or not the point $f(x_k)$ is contained in B_i.

REMARK 3.2 (a) Results on the convergence properties of the discretization technique described above can be found in e.g. [13, 5, 10] for the case where the underlying measure is absolutely continuous with respect to Lebesgue measure.

But also in the situation where the underlying dynamical system has a hyperbolic invariant set supporting an SBR-measure one can obtain convergence results: the idea is to use the concept of *small random perturbations* of discrete dynamical systems which allows to work within the simpler framework of stochastic dynamical systems (see [3, 11]).

(b) We remark that there is also a deterministic exhaustion technique to approximate P_h. This method has been developed in [9].

The Complete Algorithm

The strategy for the approximation of an invariant measure supported on a relative global attractor can now be formulated as follows:

1. Approximate a relative global attractor A_Q by the subdivision algorithm to obtain a covering by the set of boxes $\{B_1, B_2, \ldots, B_N\}$.

2. Use this collection of boxes to compute the discretized Frobenius-Perron operator P_h by (3.3).

3. Compute the eigenvector u_h corresponding to the eigenvalue 1 of P_h to obtain an approximation μ_h of the natural invariant measure μ on A_Q (see (3.2)). We remark that P_h is represented by a stochastic matrix and therefore always possesses the eigenvalue one. Moreover, the corresponding eigenspace is one-dimensional if P_h is irreducible.

REMARKS 3.3 (a) The methods used so far for the computation of invariant measures by means of the Frobenius-Perron operator work with a covering of the whole set $Q \subset \mathbb{R}^n$ (see e.g. [9, 10]). Obviously, the number of boxes N in our covering $\mathcal{B}_k$ of the attractor is in general at least an order of magnitude smaller in comparison to a covering of Q of the same fineness. Thus, we end up with a much smaller eigenvalue problem for the approximate invariant measure. In fact, the size of the eigenvalue problem depends on the size of the attractor, that is, on the complexity of the dynamics, and not on the size of Q or the dimension of phase space. This allows us to compute invariant measures for complicated attractors with higher accuracy.

(b) The algorithm described above is integrated into the C++ code GAIO (**G**lobal **A**nalysis of **I**nvariant **O**bjects). A link to a detailed description of GAIO can be found on the homepages of part of the authors:

`http://www.uni-bayreuth.de/departments/math/~mdellnitz`
`http://www.uni-bayreuth.de/departments/math/~ojunge`

4 Visualization Techniques

Tree–type representations of large volume data sets have been discussed in a different visualization context by several authors [4, 6, 16]. They particularly focus on the storage benefits of tree encoded objects and on the hierarchical searching for intersections e. g. with isosurfaces. In addition, we here discuss further advantages which significantly help to speed up image generation. The hierarchical structures used for storing $\mathcal{B}_k$, which are introduced in Sec. 2 are also well suited to set up hierarchical visualization procedures for different types of efficient and flexible post processing. To enlighten geometrical aspects of the invariant set and the corresponding discrete measure under consideration three major display methods have turned out to be very useful.

(i) With the first method the geometry of the approximate invariant sets is visualized. Here we simply pick up the hull of $\mathcal{B}_k$.

(ii) The second technique allows to extract the intersection of $\mathcal{B}_k$ with an arbitrary plane. Additionally we map the measure onto the intersection surface using some color shading.

(iii) Finally a transparent volume rendering gives a global impression of the collection $\mathcal{B}_k$ and, at the same time, of the approximate invariant measure μ_h, where the color and transparency of every box $R \in \mathcal{B}_k$ depends on the value of μ_h on R.

All these methods can be implemented in terms of a recursive traversal of the sparse tree representation of $\mathcal{B}_k$. Traversing the tree structure and implementing these techniques several algorithmical aspects are of particular interest.

Hierarchical Searching

First of all it is obvious that the search for an intersection plane can easily be guided by an appropriate tree traversal, where R_+ and R_- are only visited for boxes R which are intersected by the currently selected plane. The searching algorithm is similar to the binary search necessary to identify a box containing preimages $f^{-1}(x)$ for x in some box $B \in \mathcal{B}_k$, which is already used in the selection step of the subdivision algorithm (cf. Sec. 2).

Depth Sorting of the Boxes

For the volume rendering it is essential to order the rendered boxes depending on their distance from the user in the viewing direction [12]. A partial ordering will

be sufficient, which guarantees that successors in the traversal will not be hidden by previously traversed boxes. We immediately see that an ordering of the child boxes solely depends on the viewing direction and the type of refinement, i.e. the coordinate direction in which the current box gets splitted up. Let us introduce $\sigma(j) \in \{+,-\}$ such that $R_{\sigma(j)} > R_{-\sigma(j)}$ where ">" denotes the ordering relation in the viewing direction and $j = ((k-1) \mod n)+1$ is the direction of bisection in the k-th subdivision step. That is $R_{\sigma(j)}$ is closer to the viewer than $R_{-\sigma(j)}$. Since the $\sigma(j)$ do not depend on a specific box one can compute them in advance. A recursive traversal of the tree structure for display purposes with respect to this ordering is sketched by the following C–type pseudo code:

```
Display(R, l) {
    j = ((l − 1) mod n) + 1;
    if ( l < k ) {
        if R_{−σ(j)} ∈ B_{l+1}
            Display(R_{−σ(j)}, l + 1);
        if R_{σ(j)} ∈ B_{l+1}
            Display(R_{σ(j)}, l + 1);
    }
    else RenderBox(R, l);
}
```

The traversal is started on the single box $R \in \mathcal{B}_0$ on level $l = 0$ calling the recursive procedure Display(R, l). On some level of interest k — or at least on leaf nodes — we finally call a rendering procedure RenderBox(). This could then be the display of the polygonal hull of $\mathcal{B}_k$ or a volume rendering of the invariant measure on $\mathcal{B}_k$ (cf. Fig. 4). Displaying the hull consists of drawing illuminated patches for the faces of the considered box. Out of the six bounding faces $\{F_{j,\sigma}\}$, $j = 1, 2, 3$, $\sigma \in \{+,-\}$ of R (cf. Fig. 3) only those faces $F_{j,\sigma(j)}$ are actually rendered for which no neighboring box $N_{j,\sigma(j)}(R)$ exists. All the others are definitely invisible. Therefore tracking the adjacency information and simultaneously excluding the interior faces of the set collection $\mathcal{B}_k$ considerably reduces the remaining rendering cost. As already sketched above the adjacency relation $N_{j,\sigma}(R)$ can procedurally be tracked during the tree traversal. Furthermore, $\sigma(j)$ already used for a straightforward and inexpensive ordering supports easy backface culling, that is, not to draw patches for the faces $F_{i,-\sigma(j)}$ which are pointing away from the viewer. This again reduces the amount of patches to be considered.

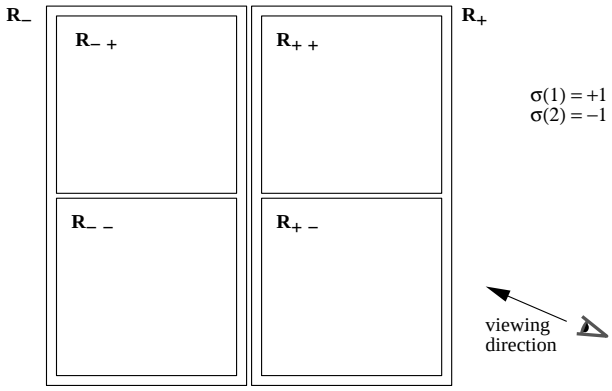


Figure 4: A three level box hierarchy in 2D, which is traversed depending on the viewing direction. The procedure is called on the different boxes in the order: R_-, R_{-+}, R_{--}, R_+, R_{++}, R_{+-}.

Color and Transparency Representing Measure Density

For transparent volume rendering purposes we draw colored and transparent splats, which are collections of several patches for each box $R \in \mathcal{B}_k$. With this we also get insight into interior parts of the box collection, where thick areas appear less transparent than thin ones. In particular the whole collection $\mathcal{B}_k$ is visible. Here we suppose that the addressed graphics library correctly handles transparent patches, if they are processed depth first corresponding to the viewing direction. But this has actually been done by our type of traversal. For each box $R \in \mathcal{B}_k$ we compute

$$\lambda(R) = \frac{\log \mu(R) - \log \min_{\widetilde{R} \in \mathcal{B}_k} \mu(\widetilde{R})}{\log \max_{\widetilde{R} \in \mathcal{B}_k} \mu(\widetilde{R}) - \log \min_{\widetilde{R} \in \mathcal{B}_k} \mu(\widetilde{R})}.$$

Then $\max\{\lambda(R)\,\mathrm{diam}(R), \epsilon\}$ for a small ϵ is a suitable choice for an opacity factor. This ensures that regions with low measure will be more transparent than those with high measure. The ϵ guarantees that the whole set collection $\mathcal{B}_k$ is visible. In addition we assume the color of the transparent patches also to depend on the values $\lambda(R)$ on R. Let us denote by v_α for $\alpha = [\alpha^1, \alpha^2, \alpha^3]$ the common edge of the faces $F_{i,\alpha^i\sigma(i)}$. Then for $+,-$ indicating solely positive or negative index components, v_+ and v_- are a possibly non-unique foremost vertex, and an invisible vertex, respectively. We obtain a reasonable approximation for a single transparent box drawing the following six transparent triangles Δ indexed by their vertices:

$$\Delta_{v_+, v_\alpha, v_\beta}$$

where the α and β have at least one positive and one negative entry and differ only in one component (cf. Fig. 5).

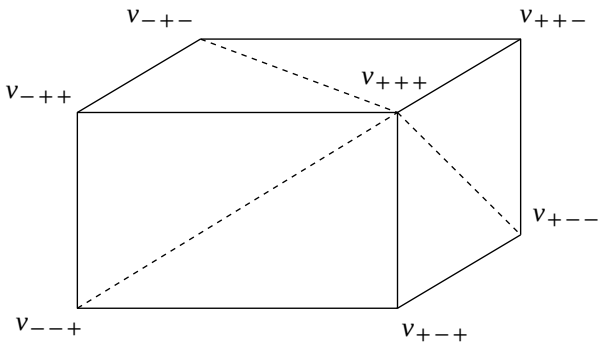


Figure 5: A box R and its vertices indexed according to the current viewing direction.

Graphic Environment: The visualization is embedded in the concept of arbitrary meshes [14] implemented on the object oriented software platform GRAPE [7, 15], which allows an interactive exploration of such hierarchical data. In particular, the calculation of the intersection plane is immediately available on our type of data by inheritance. The volume rendering is adapted to the specific tree structure discussed here. All parameters, such as the level of interest in the hierarchy, the intersection plane and color shading parameters, the viewing direction etc. can be controlled interactively.

5 Example

As an example we visualize invariant sets and invariant measures for the Lorenz system

$$\begin{aligned} \dot{x} &= \sigma(y-x) \\ \dot{y} &= \rho x - y - xz \\ \dot{z} &= -\beta z + xy. \end{aligned}$$

We set

$$\sigma = 10 \quad \text{and} \quad \rho = 28$$

and approximate the closure of the twodimensional unstable manifold of the steady state solution

$$q = \left(\sqrt{\beta(\rho-1)}, \sqrt{\beta(\rho-1)}, \rho - 1\right)$$

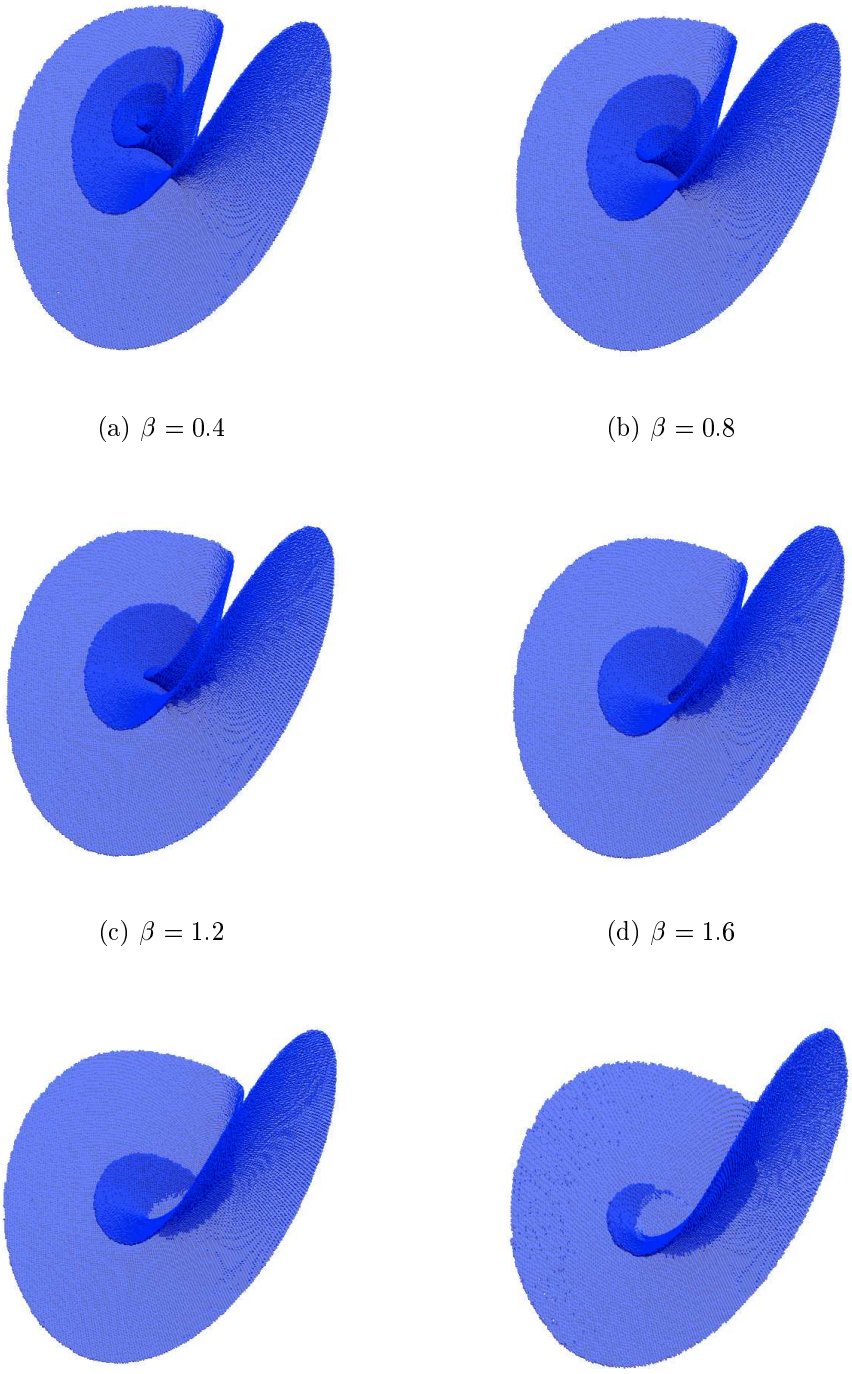
for the six parameter values $\beta \in \{0.4, 0.8, 1.2, 1.6, 2.0, 8/3\}$ by a continuation technique based on the subdivision algorithm (see [2]). The results are visualized in Figures 6-8 using the three different visualization techniques introduced in Sec. 4.

In Figure 6 the shape of the closure of the unstable manifold is visualized. The six results illustrate that essentially only the "spiraling behavior" of the relative global attractor around the z-axis changes while β is varied: for $\beta = 0.4$ we can see that part of the unstable manifold is winding around the z-axis whereas for $\beta = 8/3$ this behavior has disappeared.

In Figure 7 this unwinding behavior becomes even more transparent. Moreover, the color shading in the intersection planes shows that for small values of β the dynamical behavior stays close to the z-axis for relatively long periods of time which is not the case for larger values of β.

Finally, in Figure 8 we visualize the invariant measures on the relative global attractors. These results indicate that, in contrast to the geometric shape, the invariant measures change drastically while β is varied. First we observe that for small values of β there are big regions of low density on the invariant set. In fact, for $\beta = 0.4$ a direct numerical simulation indicates that there exists a stable periodic solution inside the yellow area in subfigure (a). Moreover, the steady state solution q itself is apparently not contained in the support of the invariant measure on A_Q for $\beta \in [0.4, 2.0]$. (Observe that the nonzero steady states of the Lorenz system lie inside the blue "islands" close to the center of the objects in Figure 8.) Finally, it appears that for $\beta = 8/3$ the Lorenz attractor A_Q is equal to the support of the measure which is computed.

Acknowledgments. Research of MD and OJ is partly supported by the Deutsche Forschungsgemeinschaft under Grant De 448/5-1. MD additionally acknowledges support by the Konrad-Zuse-Zentrum für Informationstechnik Berlin. The visualization platform GRAPE has been developed at the SFB 256 at the University of Bonn and at the Institute for Applied Mathematics at the University of Freiburg.

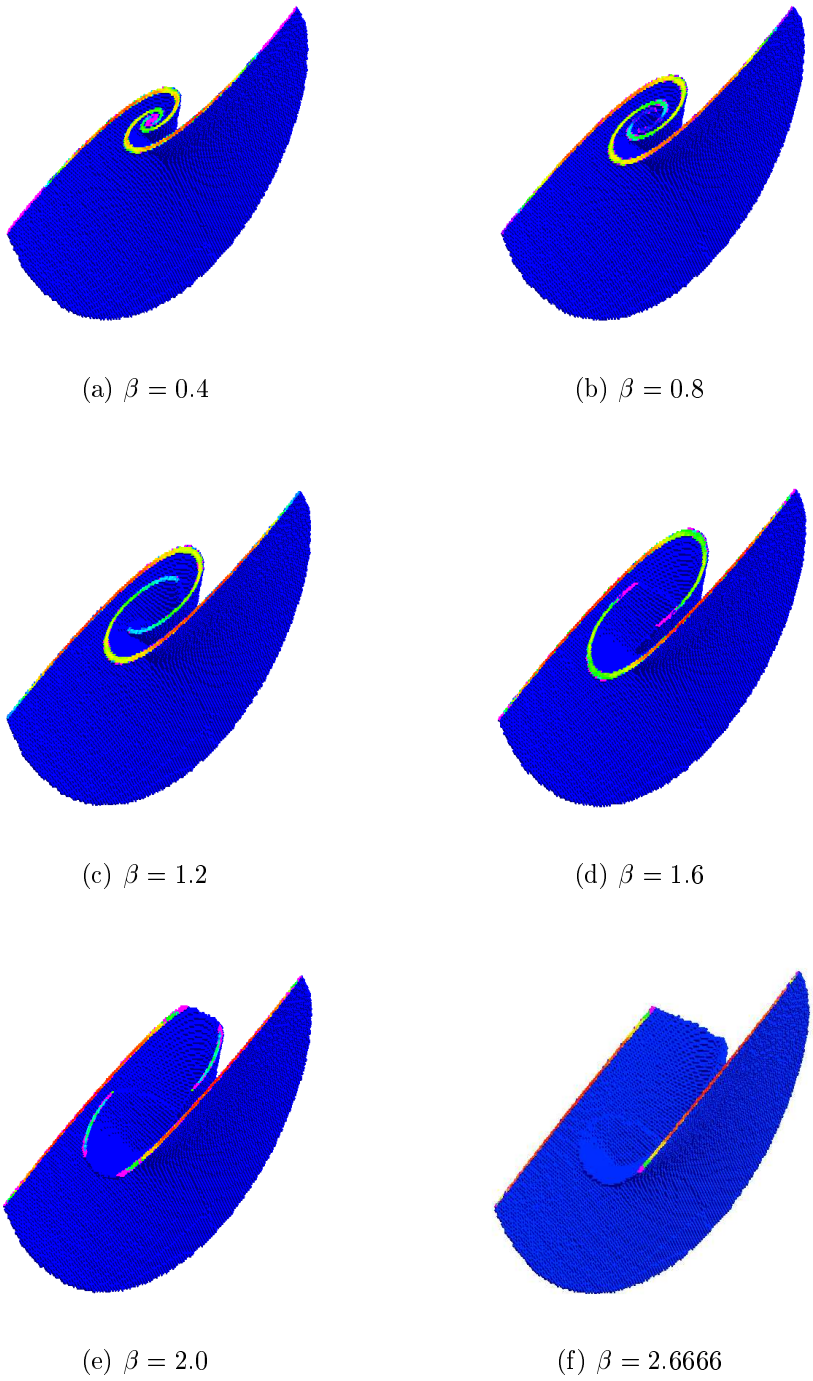


(a) $\beta = 0.4$ (b) $\beta = 0.8$

(c) $\beta = 1.2$ (d) $\beta = 1.6$

(e) $\beta = 2.0$ (f) $\beta = 2.6666$

Figure 6: Six coverings of the unstable manifold of the steady state q using 24 steps in the subdivision algorithm.

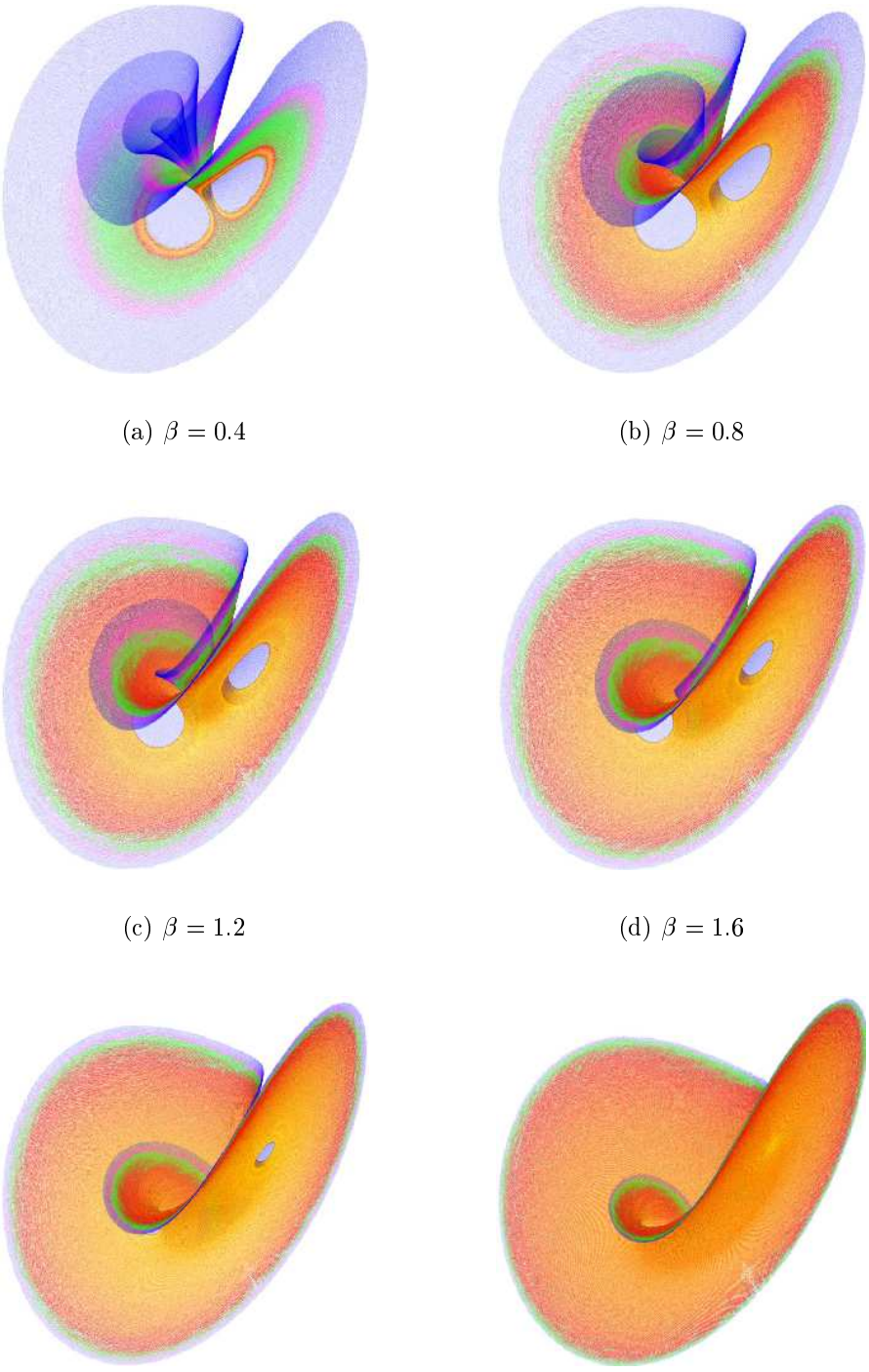


(a) $\beta = 0.4$ (b) $\beta = 0.8$

(c) $\beta = 1.2$ (d) $\beta = 1.6$

(e) $\beta = 2.0$ (f) $\beta = 2.6666$

Figure 7: The intersection planes for different values of β. The color shading indicates the different levels of the density on the boxes. (The density ranges from pink (low density) $\rightarrow$ blue $\rightarrow$ green $\rightarrow$ yellow $\rightarrow$ orange (high density).)



(a) $\beta = 0.4$ (b) $\beta = 0.8$

(c) $\beta = 1.2$ (d) $\beta = 1.6$

(e) $\beta = 2.0$ (f) $\beta = 2.6666$

Figure 8: Approximations of the invariant measure on $\mathcal{B}_{28}$. (The density ranges from blue (low density) $\to$ pink $\to$ green $\to$ red $\to$ yellow (high density).)

References

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

[2] M. Dellnitz and A. Hohmann. The computation of unstable manifolds using subdivision and continuation, in *Nonlinear Dynamical Systems and Chaos* (H.W. Broer, S.A. van Gils, I. Hoveijn und F. Takens eds.), *PNLDE* **19** (Birkhäuser, 1996), 449-459.

[3] M. Dellnitz and O. Junge. On the approximation of complicated dynamical behavior. Submitted, 1996.

[4] L. J. Doctor and J. G. Torborg. Display techniques for octree–encoded objects. *IEEE Comput. Graph. Appl.* **1** (3) (1981), 29–38.

[5] J. Ding, Q. Du and T.Y. Li. High order approximation of the Frobenius-Perron operator, *Appl. Math. Comp.* **53** (1993), 151-171.

[6] I. Gargantini. Linear octrees for fast processing of three–dimensional objects, *Comput. Graph. Image Process.* **20** (4) (1982), 365–374.

[7] M. Geiben and M. Rumpf. Visualization of finite elements and tools for numerical analysis, *Advances in Scientific Visualization,* F. Post and A. H. Hin eds., (Springer, 1993).

[8] J. Guckenheimer and P. Holmes. *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*, (Springer, 1986).

[9] R. Guder, M. Dellnitz and E. Kreuzer. An adaptive method for the approximation of the generalized cell mapping. *Chaos, Solitons and Fractals*, **8**(4):525-534, 1997.

[10] F.Y. Hunt. A Monte Carlo approach to the approximation of invariant measures, National Insitute of Standards and Technology, *NISTIR* **4980**, (1993).

[11] Yu. Kifer. Computations in dynamical systems via random perturbations, Preprint (1996).

[12] M. Levoy. Efficient ray tracing of volume data, *ACM Trans. Graph.* Vol. 9, 3 (1990), 245–261.

[13] Li, T. Y.: Finite approximation of the Frobenius-Perron operator, a solution to Ulam's conjecture, *J. Appr. Theory* **17** (1976), 177–186.

[14] M. Rumpf, A. Schmidt and K. G. Siebert. Functions defining arbitrary meshes, a flexible interface between numerical data and visualization routines, Report 14, SFB 256, Bonn, (1994), to appear in *Computer Graphics Forum*.

[15] M. Rumpf and A. Wierse. GRAPE, eine objektorientierte Visualisierungs– und Numerikplattform. *Informatik, Forschung und Entwicklung* **7** (1992), 145–151.

[16] J. Wilhelms and A. Van Gelder. Octrees for faster isosurface generation, *ACM Trans. Graph.* Vol. 11, 3 (1992), 201–227.