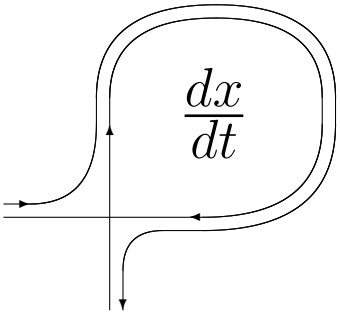


DIFFERENTIAL EQUATIONS
AND
CONTROL PROCESSES
N 3, 2005
Electronic Journal,
reg. N P23275 at 07.03.97

http://www.neva.ru/journal
e-mail: diff@osipenko.stu.neva.ru

Ordinary differential equations

Investigating Dynamics by Symbolic Analysis: Tunings for an Efficient Computation of the Symbolic Image

Danny Fundinger

Institute of Parallel and Distributed Systems, University of Stuttgart,
Universitätsstraße 38, 70569 Stuttgart,
Germany
danny.fundinger@destinydreaming.de

Abstract Numerical methods based on the conceptual framework of symbolic analysis are important tools for the study of non-linear dynamical systems. Implementations of these concepts have been applied successfully to many investigation tasks, like the localization of the chain recurrent set, attractors, and their domains of attraction as well as the computation of the Morse spectrum and verification of hyperbolicity. However, the field of application is still limited. Reason is that the construction of the symbolic image, which is the basic task of every computation, can require a large amount of memory resources. Hereby, the amount of required resources does not only depend on the dimension of the dynamical system and the performed investigation task but also on the specific characteristics of the system's dynamics. In this work, we propose methods for the tuning of the construction process. The main idea is to use higher function iterates in order to build the symbolic image. The application of this technique allows a significant reduction of required memory resources so that the investigation methods can be applied in a wider range of scenarios.

1 Introduction

The theoretical concepts of *symbolic analysis* provide a unified framework for the acquisition of information about the flow of a dynamical system without any restrictions concerning the stability of specific invariant sets. The mathematical theory was presented in a series of works by G. S. Osipenko [20, 21, 22, 26]. It can be considered close to *Cell-to-Cell mappings* [13, 14] and multivalued mapping [19]. It is also related with *symbolic dynamics* [2, 4, 16, 30] and *set-oriented methods* [6, 9]. The main idea is the construction of a directed graph which represents the structure of the state space for the investigated dynamical system. This graph is called the *symbolic image* of the focused system and can be seen as an approximation of the system flow. Each vertex of the graph is an area of the discretized phase space, and a successive refinement is achieved by multilevel subdivision. From the computational point of view, the usage of such a graph bears the big advantage that, once it is constructed, all investigations are matters of graph analysis, e.g. each strongly connected part represents a component of the chain recurrent set of the flow. More sophisticated computational analysis of the symbolic image graph allows, among others, the localization of the chain recurrent set [22], periodic orbits [21], invariant sets [15], attractors and their basins [11, 12, 24, 27] as well as the computation of the *Morse Spectrum* [23, 25, 28] or verification of hyperbolicity. A comprehensive overview can be found in [26].

Due to the fact that the construction of the symbolic image graph is the basic task for any investigation based on symbolic analysis, an efficient implementation of this construction process is crucial for the application of the method. Adequate algorithms and data structures to achieve this task are proposed in [10]. Also a software was developed which allows the practical application of the numerical techniques. This software is part of the larger non-commercial package AnT [3] and available for download, see [1].

The target of this work is to introduce tuning techniques which allow the application of our investigation methods in scenarios where the construction of a regular symbolic image graph is not or only to a limited extent possible. Such a case happens, for instance, if the memory resources are exceeded and the desired investigation can not be finished. The essence of our approaches for tuning is that some aspects important in theory are neglected for the sake of a successful practical application of the investigation method. The tunings proposed in this work are motivated by empirical studies of our computations.

Their application often leads to a more efficient and/or more precise calculation.

This work is structured as follows. In the following section we give a short introduction into the basic concepts of symbolic analysis. Then we point out the problems which occur in a practical application of the concepts. In Sec. 4 we propose the first possible tuning, the use of higher function iterates for dynamical systems discrete in time. Afterwards, in Sec. 5, we outline related tunings for systems continuous in time. In Sec. 6 we present a further technique, the reconstruction of fragmented solutions. This technique can be applied in combination with higher function iterates in order to improve the quality of the computed results. In Sec. 7 two numerical case studies are presented to show and verify the application of the proposed techniques in practice. In Sec. 8 we give a conclusion about the methods presented in this work.

2 Construction and Investigation of the Symbolic Image

We consider a dynamical system generated by a continuous mapping

$$\mathbf{f} : M \mapsto M$$

on a compact C^∞ manifold M. For the purpose of simplification we assume a dynamical system discrete in time. Anyhow, all definitions which base on this assumption and which are introduced here can also be applied on systems continuous in time if we introduce a shift operator along trajectories. A more detailed description follows in Sec. 5. Note furthermore that the theoretical results regarding symbolic analysis [26] are based on the assumption that the underlying mapping is a homeomorphism. However, the parts of the theory which are discussed here are also valid without any restrictions for continuous mappings.

Let $C = \{M(1), ..., M(n)\}$ be a finite covering of closed sets for the domain M. The sets $M(i)$ are named boxes of the covering. For each box $M(i)$ we consider its image

$$\mathbf{f}(M(i)) = \{\mathbf{y} \,|\, \mathbf{y} = \mathbf{f}(\mathbf{x}), \mathbf{x} \in M(i)\}\,. \tag{1}$$

Then we define the covering $C(i)$ consisting of boxes $M(j) \in C$ whose intersections with $\mathbf{f}(M(i))$ are not empty :

$$C(i) = \{M(j) : M(j) \cap \mathbf{f}(M(i)) \neq \emptyset\}. \tag{2}$$

Let us construct a directed graph G which matches to each box $M(i)$ the vertex c_i. The vertices c_i and c_j (also denoted cells in the following) are connected by the directed edge $c_i \to c_j$ if and only if $M(j)$ is an image box of $M(i)$, i.e. $M(j) \in C(i)$. See also Fig. 1.

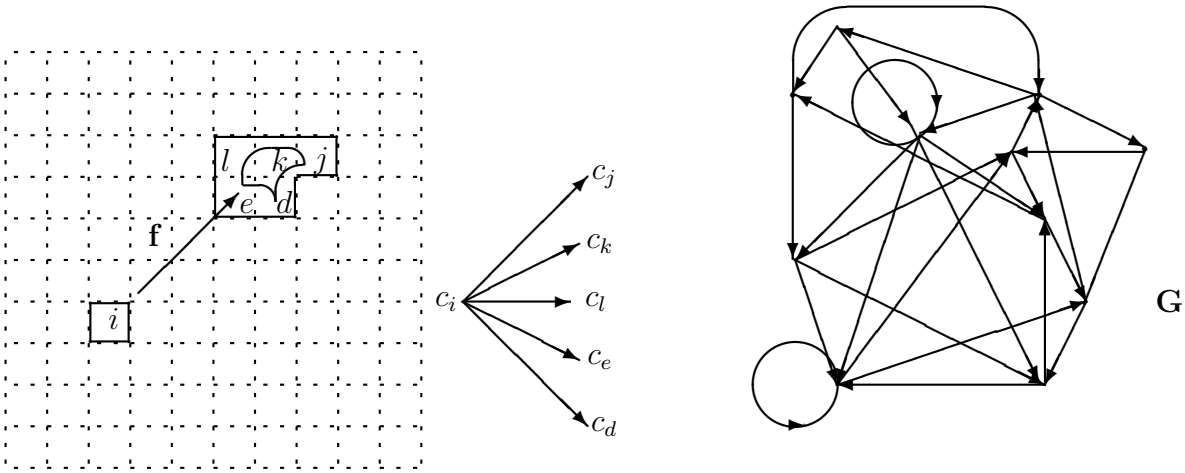


Figure 1: Construction of the symbolic image.

Definition 2.1. *The graph G constructed as described above is called the symbolic image of* $\mathbf{f}$ *with respect to the covering C.*

We can consider the symbolic image as a finite approximation of the system flow $\mathbf{f}$. The precision of the approximation depends on the covering C. By variation of C we can change the symbolic image of $\mathbf{f}$.

Definition 2.2. *An infinite in both directions (bi-infinite) sequence $\{c_{i_k}\}$ of cells in the graph G is called an* admissible path *(or simply a path) if for each k the graph G contains the edge $c_{i_k} \to c_{i_{k+1}}$. A path $\{c_{i_k}\}$ is said to be p-periodic if $c_{i_k} = c_{i_{k+p}}$ for each $k \in \mathbb{Z}$.*

There is a natural correspondence between the admissible paths on the symbolic image G and the periodic orbits of $\mathbf{f}$. Roughly speaking, an admissible path represents the trace of an orbit and vice versa.

Definition 2.3. *A cell of a symbolic image is called* recurrent *if there is a periodic path passing through it. Two recurrent cells c_i and c_j are called* equivalent *if there is a periodic path containing both, c_i and c_j.*

Denote the set of cells in G as $V(G)$ and the set of recurrent cells as $RV(G)$. Obviously, we have the relation $RV(G) \subseteq V(G)$. The set $RV(G)$ decomposes into classes of equivalent recurrent cells

$$H_i = \{c_j \mid \text{there is a periodic path containing } c_i \text{ and } c_j\}. \tag{3}$$

Hence, the sets H_k, each representing a disjunct equivalence class, form together the set

$$\zeta = \{H_k\},\ \forall H_i, H_j \in \zeta : H_i \cap H_j = \emptyset,\ RV(G) = \bigcup_{H_k \in \zeta} H_k. \tag{4}$$

In graph theory, a class H_k is called a set of *strongly connected components* of the graph G. The boxes $M(i)$ belonging to the cells $RV(G)$ are a neighborhood of the *chain recurrent set* [5] and the boxes belonging to a set H_k are a neighborhood of a component of the chain recurrent set, see [26]. The chain recurrent set contains all types of return trajectories, like periodic or recurrent orbits. Each of these return trajectories is a component of the chain recurrent set. For this reason, the detection of the recurrent cells $RV(G)$ and ζ are basic tasks on a symbolic image graph and required as a first step for almost all other investigation methods.

In order to get a better approximation of the vector field $\mathbf{f}$, a multilevel subdivision procedure will be applied, see also Dellnitz *et al.* [8, 7] for a similar approach. We consider a covering C^s, $s \geq 0$, and its symbolic image G^s. A selection $S(G^s)$ of cells which cover the areas of interest, for example $S(G^s) = RV(G^s)$, is chosen for subdivision. The boxes of the covering which correspond to the cells of $S(G^s)$ will be subdivided into smaller units. These units define a new, more precise covering C^{s+1} for which the next symbolic image G^{s+1} can be constructed. The former covering C^s and the graph G^s can then be deleted in order to free memory space. In theory, this procedure is applied for $s \to \infty$. In practice, however, subdivision can only be applied several times whereby the approximation of the symbolic image becomes more precise for each subdivision.

3 Problems of a Practical Computation

The construction of a symbolic image G^s of a subdivision step s, its subdivision and the localization of recurrent cell sets are the basic steps of every investigation based on symbolic analysis. Appropriate algorithms and data structures to achieve this task are proposed in [10]. Note hereby that the construction of a symbolic image based on numerical calculations is always only an estimation of the "real" symbolic image G^s. Besides the usual numerical errors which occur by the computation of a vector field $\mathbf{f}(\mathbf{x})$ for $\mathbf{x} \in M$, another reason for this is

the fact that the construction of the image

$$\begin{aligned} T(i) &= \mathbf{f}(M(i)) \\ &= \{\mathbf{y} \mid \mathbf{y} = \mathbf{f}(\mathbf{x}), \mathbf{x} \in M(i)\} \subset M \end{aligned} \tag{5}$$

for a box $M(i)$ would involve the calculation of $\mathbf{f}(\mathbf{x})$ for every $\mathbf{x} \in M(i)$. This is, of course, beyond the limits of every finite numerical computation.

In the approach proposed in [10], the image $T(i)$ will be approximated by a finite set of points. This technique was also used by [14, 7], and has proved to be a good approach in practice. From each box $M(i)$ a representative set of k points is selected, the so-called *scan points* $S(i) = \{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$. Then the approximation $\tilde{T}(i)$ of the region $T(I)$ in the state space is calculated by

$$\tilde{T}(i) = \mathbf{f}(S(i)) = \{\mathbf{y}_j \mid \mathbf{y}_j = \mathbf{f}(\mathbf{x}_j), \mathbf{x}_j \in S(i)\}\,. \tag{6}$$

Performance analysis has shown that the computation time of algorithms based on this approach are no major obstacle for the construction of the symbolic image. Assuming n_s is the number of cells belonging to a symbolic image G^s, i.e. $n_s = |V(G^s)|$, all computations can be performed in $O(n_s \cdot \log(n_s))$, see [10]. Instead, the crucial factor is the size of the input value n_s or, in other words, the memory resources required for a computation. Note that n_s could grow almost exponential for $s \to \infty$, i.e. during the subdivision process. Ideally, this growth rate should only depend on the investigation task or, more precisely, the dimension of those objects which are the subjects of investigation. However, due to the complexity of the underlying dynamics, this is often not the case. Instead, we observed that in many computations much more cells are selected for subdivision than necessary. The major problem we come across is that not only those cells are selected which correspond to boxes containing parts of the solution, but also several more cells which correspond to boxes in the neighborhood of the solution. Reason for this is a too coarse discretization of the phase space. In the following, we will refer to this phenomenon as *clustering*. Due to clustering, the growth rate of the number of cells in a symbolic image increases during the subdivision process, and the accuracy of the computation shrinks. Moreover, the analysis of computed data is more difficult.

Taking the theoretical point of view, the selection of too many cells for subdivision does not matter. By successive application of the subdivision process the discretization of the phase space gets finer. Eventually, those boxes which do not contain a solution will be deleted and the solution is detected as precisely as one likes. However, taking the practical point of view, one has to deal

with limited resources. That means that the number of applicable subdivisions is limited by the memory space of the computation machine which only allows the storage of a symbolic image graph of a limited size. Therefore, it is our strong concern to avoid clustering, i.e. the selection of cells for subdivision which do not contain a solution. This aim can only be achieved by a change of paradigm. The target is not anymore the rigorous construction of the symbolic image graph for a phase space discretization. We are not interested in providing all existing edges between the cells as requested in the theoretical approach, but rather more only those edges which are necessary for the detection of the solution. By doing so, we are also aware of the fact that some important information might get lost. However, empirical studies have shown that computational investigations are mostly limited by performance resources instead of an insufficient approximation of the symbolic image. A significant reason for this is that the method is typically quite robust.

4 Use of Higher Iterated Functions

When dealing with dynamical systems discrete in time $\mathbf{x}_{n+1} = \mathbf{f}(\mathbf{x}_n)$, the points $\mathbf{y} \in \tilde{T}(I)$, see Eq. 6, which represent the images of $\mathbf{x}$, are calculated as direct successors of the scan points: $\mathbf{y} = \mathbf{f}(\mathbf{x})$. We propose now that in some cases it is more suitable to use an iterated function of $\mathbf{f}$ and calculate the image points by

$$\mathbf{y} = \mathbf{f}^{[n]}(\mathbf{x}), \quad n > 1 \tag{7}$$

In other words, the symbolic image is not constructed for the function $\mathbf{f}$ but for the n-th iterated function $\mathbf{f}^{[n]}$. In the following, we will denote a symbolic image constructed for $\mathbf{f}$ by G_f and for $\mathbf{f}^{[n]}$ by $G_{f^{[n]}}$.

Obviously, the symbolic image graph $G_{f^{[n]}}$ with $n > 1$ differs from G_f. More precisely, $G_{f^{[n]}}$ might have less edges than G_f. However, $G_{f^{[n]}}$ is still useful for investigations. In order to clarify this, we introduce some theorems about the relations of $\mathbf{f}^{[n]}$ and $\mathbf{f}$ with regard to invariant sets.

Definition 4.1. *A set $Q \subset M$ is said to be invariant if $\mathbf{f}(Q) = Q$.*

Proposition 4.1. *If $Q \subset M$ is an invariant set for $\mathbf{f}$, then also for any $\mathbf{f}^{[n]}$, $n \in \mathbb{N}$, i. e.*

$$\mathbf{f}(Q) = Q \Rightarrow \mathbf{f}^{[n]}(Q) = Q.$$

Proof. Obviously, if $\mathbf{f}(Q) = Q$ then $\mathbf{f}(\mathbf{f}(Q)) = \mathbf{f}(Q) = Q$. We can conclude that then also $\mathbf{f}^{[n]}(Q) = \mathbf{f}^{[n-1]}(\mathbf{f}(Q)) = \mathbf{f}^{[n-1]}(Q) = \cdots = Q$. □

Considering this result, we can conclude that all invariant sets of a dynamical system generated by $\mathbf{f}$ can also be found in a dynamical system generated by $\mathbf{f}^{[n]}$.

Proposition 4.2. *If $Q' \subset M$ is an invariant set for $\mathbf{f}^{[n]}$, $n \in \mathbb{N}$ then $Q = \bigcup_{0 \leq k < n} \mathbf{f}^{[k]}(Q')$ is an invariant set for $\mathbf{f}$, i. e.*

$$\mathbf{f}^{[n]}(Q') = Q' \Rightarrow \bigcup_{0 \leq k < n} \mathbf{f}^{[k]}(Q') = Q = \mathbf{f}(Q).$$

Proof. Note that by definition $\mathbf{f}^{[0]}(\mathbf{x}) = \mathbf{x}$ and, hence, $\mathbf{f}^{[0]}(Q') = Q'$. We split the proof in two parts:

1. $\mathbf{f}(Q) \subseteq Q$: If $\mathbf{x} \in Q \Rightarrow \mathbf{x} = \mathbf{f}^{[k]}(\mathbf{x}')$ for some $\mathbf{x}' \in Q'$ and some $k < n$ $\Rightarrow \mathbf{f}(\mathbf{x}) = \mathbf{f}^{[k+1]}(\mathbf{x}')$. Obviously, $\mathbf{f}^{[k+1]}(\mathbf{x}') \in Q$ because if $k + 1 < n$ then $\mathbf{f}^{[k+1]}(\mathbf{x}') \in Q$, and if $k + 1 = n$ then $\mathbf{f}^{[k+1]}(\mathbf{x}') = \mathbf{f}^{[n]}(\mathbf{x}') \in Q' \subset Q$.

2. $Q \subseteq \mathbf{f}(Q)$: We first state that $\mathbf{f}^{[k]}(Q') \subseteq \mathbf{f}(Q)$ for each $k < n$ because if $k = 0$ then $\mathbf{f}^{[0]}(Q') = Q' = \mathbf{f}^{[n]}(Q') = \mathbf{f}(\mathbf{f}^{[n-1]}(Q')) \subseteq \mathbf{f}(Q)$, and if $k > 0$ then $\mathbf{f}^{[k]}(Q') = \mathbf{f}(\mathbf{f}^{[k-1]}(Q')) \subseteq \mathbf{f}(Q)$. Next we state that if $\mathbf{x} \in Q \Rightarrow \mathbf{x} \in \mathbf{f}^{[k]}(Q')$ for some k ¡ n. It follows immediately that $\mathbf{x} \in \mathbf{f}^{[k]}(Q') \subseteq \mathbf{f}(Q)$.

□

Proposition 4.3. *If $Q' \subset M$ is an invariant set for $\mathbf{f}^{[n]}$, $n \in \mathbb{N}$ then there is an invariant set Q for $\mathbf{f}$ with $Q' \subseteq Q$, i. e.*

$$\mathbf{f}^{[n]}(Q') = Q' \Rightarrow \exists Q \supseteq Q' : \mathbf{f}(Q) = Q.$$

Proof. The theorem is an immediate conclusion of Proposition 4.2. □

An important conclusion of Proposition 4.3 is that a dynamical system generated by $\mathbf{f}^{[n]}$ still consists of the same invariant sets than the one generated by $\mathbf{f}$. Each invariant set Q' found for $\mathbf{f}^{[n]}$ belongs to an invariant set Q of $\mathbf{f}$. Furthermore, according to Proposition 4.1, all sets Q of $\mathbf{f}$ can be found in the dynamical system of $\mathbf{f}^{[n]}$.

We note now that most investigations based on symbolic analysis aim to detect specific types of invariant sets. For instance, the chain recurrent set is

invariant as well as each component of it. If all invariant sets of $\mathbf{f}$ are preserved in the dynamical system of $\mathbf{f}^{[n]}$ then, obviously, they can also be detected in $G_{f^{[n]}}$. However, note that the characteristics of the sets might change. Let us look, for instance, on the periodic points of $\mathbf{f}$. We denote by $P(p)$ the set of all points in M which are p-periodic. Obviously, each $P(p)$ for $p \geq 1$ is an invariant set. Consider now that the set $P(6)$ of $\mathbf{f}$ is equivalent to the invariant set $P(2)$ of $\mathbf{f}^{[3]}$ but the points belonging to these sets have a different periodicity with respect to $\mathbf{f}$ and $\mathbf{f}^{[3]}$. Hence, one has to be careful when analyzing the results of $G_{f^{[n]}}$. However, although the periodicity might change, every periodic point of $\mathbf{f}$ is also periodic for $\mathbf{f}^{[n]}$, and no other periodic points than for $\mathbf{f}$ are found for $\mathbf{f}^{[n]}$. The same is true, for instance, for points belonging to quasiperiodic trajectories.

Each edge in the graph $G_{f^{[n]}}$ represents a longer part of a trajectory than in G_f. In terms of tuning this is of interest because transient dynamics can then be better distinguished from asymptotic ones. In a coarse discretization of the phase space, less cells corresponding to boxes which do not contain a solution are recurrent and selected for subdivision. Consequently, clustering can be reduced and the growth rate of cells during the subdivision process is lower. However, the tuning has also some drawbacks. First of all, the computation time for the construction of $G_{f^{[n]}}$ increases by factor n in comparison to G_f. Furthermore, it is more likely that unstable parts of the solution, e.g. unstable periodic or quasiperiodic points, might not be detected because the forward iterates $\mathbf{y} = \mathbf{f}^{[n]}(\mathbf{x})$ diverge stronger from these objects than $\mathbf{y} = \mathbf{f}(\mathbf{x})$. Last but not least, taking the analytical point of view, one must be aware about the change of characteristics regarding the invariant sets of $G_{f^{[n]}}$ in G_f.

5 Discretization Time for Systems Continuous in Time

Only dynamical systems discrete in time have been discussed so far. If we are dealing with systems continuous in time given by an ODE, i.e. $\dot{\mathbf{x}} = \mathbf{F}(t, \mathbf{x})$, $t \in \mathbb{R}$, some kind of mapping is required which transforms an orbit continuous in time into one discrete in time. A *shift operator* along trajectories is needed. In the implementation proposed in [10], a stroboscopic mapping is used as shift operator. Such a mapping has the form $\mathbf{f}(\mathbf{x}) = \phi(t, \mathbf{x})$ with $\phi(0, \mathbf{x}) = \mathbf{x}$ and is applicable to differential equations which are autonomous. It can be calculated by solving the equation

$$\dot{\mathbf{x}}(t) = \mathbf{F}(\mathbf{x}(t)) \tag{8}$$

for the time t and initial conditions $\mathbf{x}(0) = \mathbf{x}$. We assume a fixed $t > 0$. In that case, $\phi(t, \mathbf{x})$ is also called a *time-t map*. Such a time-t map is a restriction of ϕ to $M \times t\mathbb{Z}$ and, hence, a discretization of the dynamical system continuous in time.

Consider now that in the context of an implementation it is suitable to use a small integration step size Δt for the applied integration method in order to minimize numerical errors. Hence, we do not calculate $\phi(t, \mathbf{x})$ explicitly. Instead, we use an integration step size $\Delta t = t/n$ with $n \in \mathbb{N}$ and iterate ϕ for n times so that

$$\phi(t, \mathbf{x}) = \phi(\Delta t \cdot n, \mathbf{x}) = \phi^n(\Delta t, \mathbf{x}).$$

This approach allows the numerical computation of time-t maps for any precision, independently of the chosen discretization time t. In the following, we use the notation $\mathbf{f}(\mathbf{x}) = \phi(\Delta t, \mathbf{x})$ so that the time-t map for a $t = \Delta t \cdot n$ is given by $\mathbf{f}^{[n]}(\mathbf{x})$. Hence, the symbolic image is constructed assuming $\mathbf{f}^{[n]}(\mathbf{x})$ as the system function instead of $\mathbf{f}(\mathbf{x})$.

So we see that for dynamical systems continuous in time the iteration of the function must be considered in a different context than the iteration of discrete systems. The number of integration steps n, which determines for a fixed integration step size Δt the discretization time $t = \Delta t \cdot n$, is an essential part of the parameter setting. By variation of t one changes the discretization of the continuous trajectory starting from a point $\mathbf{x} \in M(i)$. Taking the theoretical point of view, the ideal approach is to vary the time t for each scan point $\mathbf{x} \in M(i)$ in such a way that the image $\mathbf{f}^{[n]}(\mathbf{x}) = \phi^{[n]}(\mathbf{x}, \Delta t) = \mathbf{y}$ lies in the next neighboring box $M(i')$ to which the continuous trajectory started at $\mathbf{x}$ moves to. By doing so, the complete dynamics of the underlying system can be preserved by the symbolic image graph. Of course, in that context one must also consider and properly treat the case that a trajectory $\mathbf{x}(t)$ might never leave its initial box. This happens if the box contains an invariant set, whereby it can be assumed for the box size shrinking to zero, that this invariant set is a fixed point.

Unfortunately, empirical experience showed that this approach fails for many practical applications. We observed that even for simple dynamical systems the symbolic image grows too much in each subdivision step because too many cells which do not contain a solution are selected for subdivision. A way to avoid this is to fix a reasonably large discretization time t for all computations. Due to the fact, that longer forward iterates are computed, the same effect happens as for the iteration of the system function, described in the last

section. Transient dynamics can be better distinguished from asymptotic ones, and less cells are selected for subdivision which do not contain a solution. This is in most cases a necessity for a reasonable numerical simulation. Note that the detection of stable parts of a solution is not affected by a high setting of t. However, unstable parts may not be detected because the trajectories started close to them diverge. So it is after all still essential that the symbolic image is constructed by the combination of many short forward iterates instead of a few long ones. Otherwise, the distinctive features of this investigation method can not be used. This means that t must be set to a value so that the symbolic image does not grow too big but that also the information about the whole solution will persist. It is not guaranteed that such a setting exists for every system in focus, and if it exists, there is not yet a general rule how to derive it. Only user experience and heuristic testing can lead to the most proper setting of t.

There are two ways to manipulate t. One is to increase or decrease the number n of iteration steps. Hereby, the precision of the computed parts of the trajectories does not change. Therefore it is the preferred way to control the size of t. However, the performance time for the construction of the symbolic image depends on n, and a high setting can significantly slow down the computation. Therefore, one can also consider to change the size of the integration step Δt instead of n. This can improve the performance of the calculation. Note that the precision of the computed parts of the trajectories depends on Δt. But as a consequence of the fact that the symbolic image graph is build from small forward iterates, the numerical error which arises from an increase of Δt is by far not as crucial as if long forward iterates would be computed.

6 Reconstruction of Fragmented Solutions

We have discussed the usage of higher iterated functions and large discretization times for symbolic analysis. In many calculations, these options turned out to be an adequate technique to tune investigation methods. However, it was also mentioned that unstable parts of a solution might not be detected if the number of iterations n or the time t is chosen too large. In practice, we observed, that for crucial settings of the parameters, some unstable invariant sets do not completely disappear at once, but rather more fall apart. Some parts of them are still recognized while others vanish, as it can be seen, for instance, in Figs. 3.(a) – (d) for the Lorenz system.

Such a phenomenon is a result of taking only a limited number of scan points per box in combination with following a relatively long run of trajectories in order to construct the edges of the symbolic image graph. This leads to a loss of information about the structure of unstable invariant sets. It is not our intention to give here a detailed analysis of this problem, but rather more a solution for the reconstruction of such unstable objects. Nevertheless, one should keep in mind that not every structure that looks like a disappearing unstable invariant set is necessarily a fragment of the solution. In some cases, it turned out that objects which seemed to be parts of unstable limit cycles belonged to non-cyclic orbits. So after the application of this method of reconstruction, further tests have to be applied to approve the correctness of obtained results.

The method, as introduced here, aims only on the reconstruction of the chain recurrent set. For other investigations, slight changes might be necessary. The reconstruction can be done by application of an extension to the symbolic image construction algorithm. The basic idea here is to add and/or select all cells belonging to boxes $M(i)$ of the symbolic image $G_{f^{[n]}}$ which will be passed by the forward iterates $\mathbf{f}^{[1]}(\mathbf{x}), \ldots, \mathbf{f}^{[n-1]}(\mathbf{x})$ on its way from $\mathbf{x}$ to its image $\mathbf{y} = \mathbf{f}^{[n]}$. Therefore, first the symbolic image $G_{f^{[n]}}$ will be constructed according to the standard approach. Then the investigation method is applied in order to get the set of recurrent cells $RV(G_{f^{[n]}})$. Afterwards, the following extension must be applied before the next subdivision. For every recurrent cell $c_i \in RV(G_{f^{[n]}})$ its corresponding box $M(i)$ is detected. Then, for every scan point $\mathbf{x} \in S(i)$ it has to be checked, whether its target point $\mathbf{f}^{[n]}(\mathbf{x}) = \mathbf{y} \in \tilde{T}(i)$ lies in a selected cell which is equivalent, i.e. belongs to the same set $H_k \in \zeta$ of equivalent recurrent cells. If so, we locate for each value $\mathbf{f}^{[k]}(\mathbf{x})$, $k = 1, \ldots, (n-1)$, the box $M(i')$ with $\mathbf{f}^{[k]} \in M(i')$. If the box $M(i')$ and its corresponding cell $c_{i'}$ do not exist for a visited area, they will be added to the symbolic image. Furthermore, the cell $c_{i'}$ will be marked as recurrent, no matter if it already existed or was just added.

If this extension is applied, the course of a trajectory, which connects recurrent cells, will be reconstructed. Note that the symbolic image can only become more precise by this extension. If a source cell c_i and its target cell $c_{i'}$ are recurrent and equivalent, then, consequently, all the cells corresponding to boxes which are passed by the connecting trajectory are also recurrent. In fact, if all numerically computed symbolic images of an investigation would have been exact and no approximation, reconstruction would not change them. As already mentioned, this operation might add new cells to the symbolic image.

So it can still be applied in a stage of subdivision when the fragmented invariant set has already fallen apart to a large extent. This can be seen in Fig. 3, where unstable limit cycles of the Lorenz system will be reconstructed in the 10-th subdivision step.

7 Numerical Case Studies

We present two numerical case studies in order to demonstrate the application of the above mentioned tunings. As examples, we chose a system continuous in time and one which is discrete in time. Both of them are 3-dimensional and they require high computational resources. For this reason, it is necessary to apply tuning methods. The used reference machine for all calculations was an `Asus L3000D` laptop with an `AMD Athlon XP-M 1400+` processor and `512MB SDRAM`.

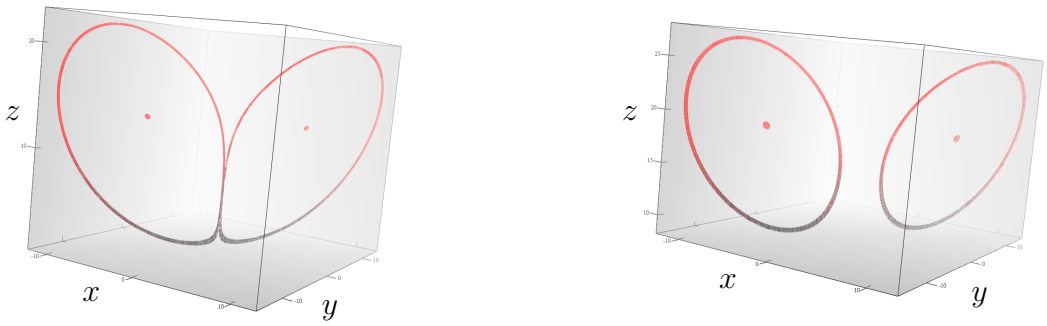
7.1 Lorenz System

We consider the well-known dynamical system continuous in time introduced by Lorenz in [18] and defined by

$$\begin{aligned} &\dot{\mathbf{x}}(t) = \mathbf{F}_L(\mathbf{x}(t)), \quad \mathbf{F}_L : \mathbb{R}^3 \to \mathbb{R}^3, \quad \mathbf{x} = (x, y, z)^T \\ &\mathbf{F}_L(\mathbf{x}) = \begin{pmatrix} \sigma(y - x), \\ x(r - z) - y, \\ xy - bz. \end{pmatrix} \end{aligned} \tag{9}$$

We use the standard parameter values $\sigma = 10$, $b = 8/3$ and investigate the Lorenz system at two values of the parameter r, namely $r_1 \approx 14.6$ and $r_2 \approx 20$. As shown in [29], for these settings exist an unstable fixed point $P = (0, 0, 0)^T$ and two stable ones C_1 and C_2, each of them accompanied by an unstable limit cycle. The value r_1 is chosen close to the so-called homoclinic explosion which occurs at $r \approx 13.926$, where the unstable manifolds of P return to the origin. Furthermore, at parameter value r_2, the both unstable limit cycles around C_1 and C_2 are situated close to each other and to C_1 and C_2.

In order to reproduce these results with methods of symbolic analysis, we compute the chain recurrent set. We define for r_1 and r_2 the domain spaces $M_1 = [-35.0; 35.0] \times [-35.0; 35.0] \times [0.0; 30.0]$ and $M_2 = [-20.0; 20.0] \times$



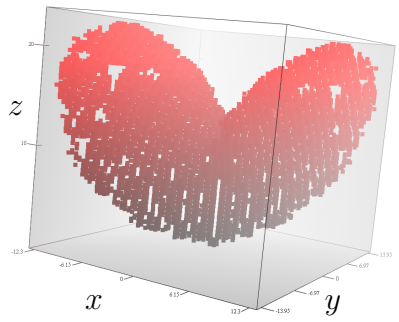
(a) $r_1 = 14.6$ (b) $r_2 = 20$

Figure 2: *Lorenz system: Computation of an outer covering of the chain recurrent set at positions* $r_1 = 14.6$ *and* $r_2 = 20$.

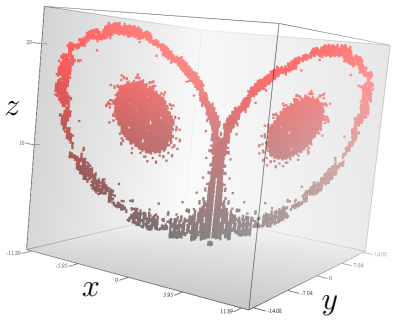
$[-20.0; 20.0] \times [0.0; 30.0]$ as the area of investigation. The division of these spaces is initially set to $4 \times 4 \times 2$ and $2 \times 2 \times 2$ boxes. In the following subdivision stages each box is divided into $2 \times 2 \times 2$ smaller boxes. The integration step Δt is set to 0.001, and the number of iteration steps to $n_1 = 100$, $n_2 = 200$. In order to compute the integration step $\phi(\Delta t, \mathbf{x})$, the Runge-Kutta method was applied.

Figs. 2(a) and 2(b) show the results of the calculations for the parameters r_1 and r_2. Remarkably, one can see that the limit cycles for r_1 still touch each other, which is due to some numerical inaccuracy, while for r_2 the cycles shrinked closer around C_1 and C_2. The computations took 30 minutes for r_1 and 2 hours for r_2. Ten subdivision steps were computed, and the symbolic images contained up to 1 400 000 cells. Hereby, the high computation time is mainly due to the relative high setting of the iteration time t. Furthermore, the unstable fixed point P can not be computed by this setting. However, if t would be set to a lower value, the limit cycles could not be detected at all because too many cells would be selected for subdivision and the memory resources would be exceeded after a few subdivisions.

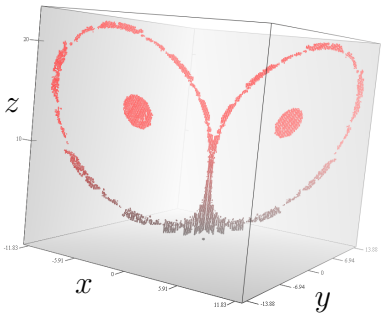
Several subdivision steps for the parameter setting $r_1 = 14.6$ are illustrated by the Figs. 3(a–f). We see that the principal shape of the cycles becomes visible in the fifth subdivision step, while the distinction into fixed points and cycles is visible in the sixth subdivision. Note that if t would be set to a smaller value, this distinction could not be computed by our methods. Too many cells would be considered recurrent and the size of the symbolic image would be



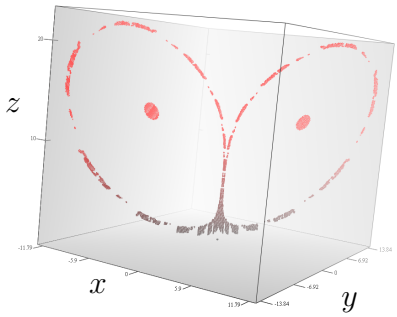
(a) fifth subdivision step



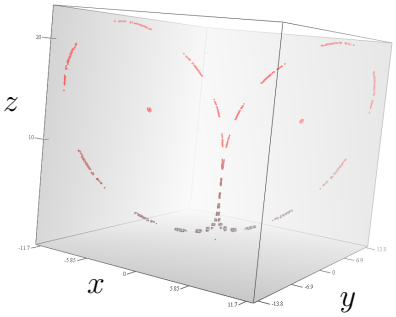
(b) sixth subdivision step



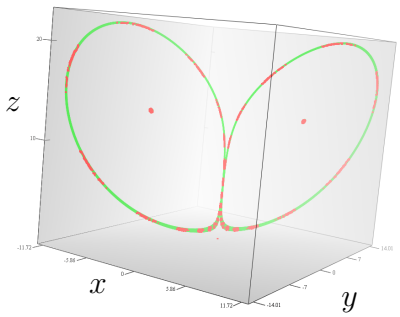
(c) seventh subdivision step



(d) eighth subdivision step



(e) tenth subdivision step



(f) completion of the fragmented limit cycle

Figure 3: *Lorenz system: Reconstruction of unstable limit cycles at parameter* $r_1 = 14.6$ *with a large discretization time. The limit cycles fall apart and vanish by time (red), but will be completed (green).*

too big for further calculations after the sixth or seventh subdivision step. In Figs. 3(c–e) we see the computations for the next subdivision steps. Although the high setting of t allows the computation of the distinct invariant sets, a side effect is that parts of the unstable limit cycles get lost. For this reason, the method for reconstruction of the fragmented solutions must be applied. The results are shown in Fig. 3(f). We see that the final computation produces a precise outer covering of the unstable limit cycles.

7.2 Food Chain Model

Next we analyze a discrete system of mathematical biology. The 3-dimensional dynamical model describes a discrete food chain model, studied by Lindström in [17]. The system is defined by

$$\begin{aligned}
&\mathbf{x}(n+1) = \mathbf{f}_{dfc}(\mathbf{x}(n)),\\
&\mathbf{f}_{dfc} : \mathbb{R}^3 \to \mathbb{R}^3, \quad \mathbf{x} = (x, y, z)^T \qquad (10)\\
&\mathbf{f}_{dfc}(\mathbf{x}) = \begin{pmatrix} \dfrac{\mu_0\, x\, e^{-y}}{1 \;+\; x\; \max\left(e^{-y}, g(z)g(y)\right)} \\ x\mu_1\, x\, y\, e^{-z} g(y) g(\mu_2\, y\, z) \\ \mu_2\, y\, z \end{pmatrix},\\
&\text{with} \quad g(s) = \begin{cases} \dfrac{1 - e^{-s}}{s}, & \text{if} \quad s \neq 0, \\ 1, & \text{if} \quad s = 0. \end{cases}
\end{aligned}$$

We only focus on the following parameter setting: $\mu_0 = 3.4001$, $\mu_1 = 1$ and $\mu_2 = 4$.

The analytic results of Lindström showed, that Eq. 10 possesses at most four fixed points. However, our main intention is not the localization of some fixed points but rather more the computation of the complete chain recurrent set within the area $M = [-1.0;\, 4.0] \times [-1.0;\, 4.0] \times [0.0;\, 1.6]$. It is not possible to get an appropriate approximation of the chain recurrent set by means of usual symbolic image construction. The tuning techniques must be applied to get satisfiable results. By doing so, the equilibrium points, and maybe some other information, get lost in the symbolic image after several subdivision steps. On the other hand, two invariant manifolds can be detected which belong to different components of the chain recurrent set, see Fig. 4. By application of

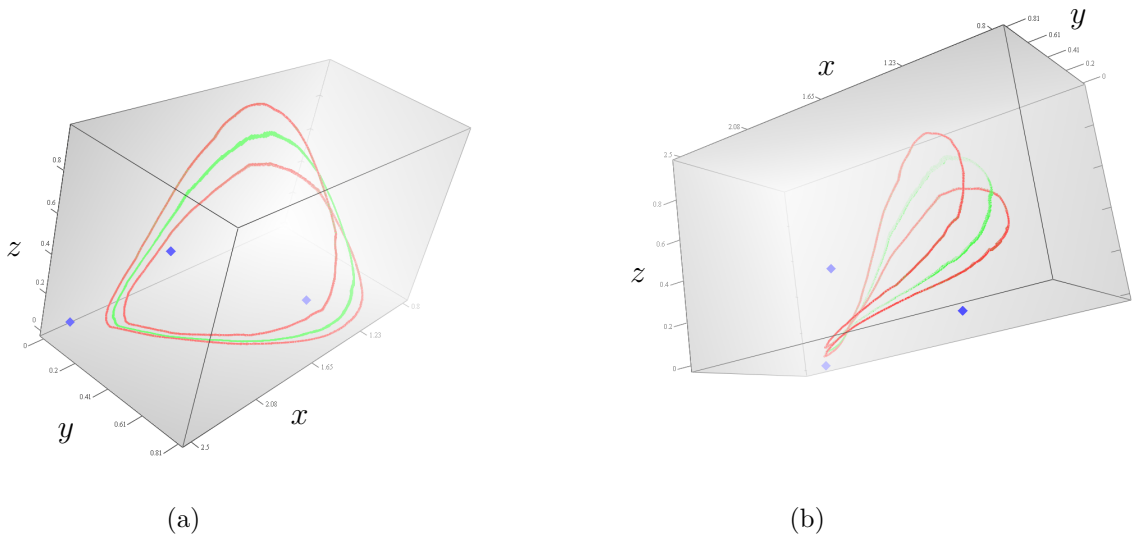
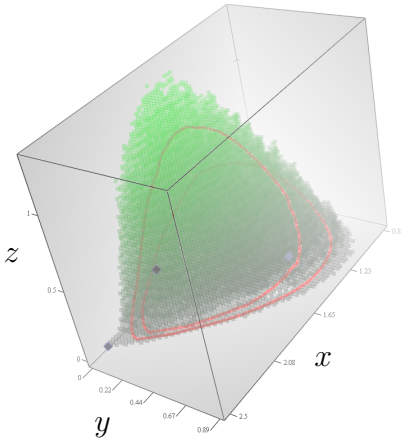


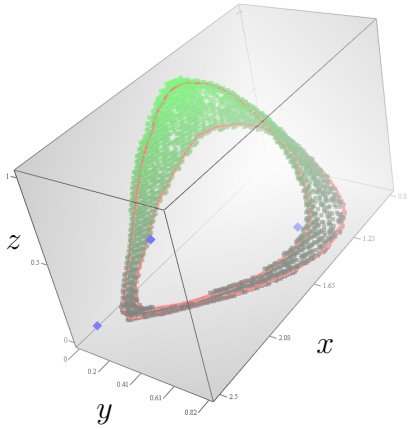
Figure 4: *Discrete food chain model: Two different views of the outer covering of the chain recurrent set. The attractor (red), an unstable quasiperiodic cycle (green) and the unstable fixed points (blue) are shown. The fourth fixed point at* $(0, 0)$ *can not be seen.*

forward iteration, it can be verified that both of them consist of quasiperiodic trajectories, and that one is a stable invariant set, namely an attractor (red), while the other is an unstable invariant set (green). Hereby, the unstable entity is not a repeller but of saddle type. Such a calculation takes around one hour and the symbolic image grows up to $\approx 1\,100\,000$ cells. The long calculation time is mainly caused by the application of the tuning-techniques. Note that the localization of the unstable quasiperiodic manifold is, from the computational point of view, a nontrivial task. To the authors' knowledge, no other numerical computation method is able to detect this entity. This is, among others, due to the fact, that the system described by Eq. 10 is only piecewise-smooth, has no explicit inverse, and possesses dynamics which are, in general, difficult to handle. Only by application of the tuning methods, the results can be computed.

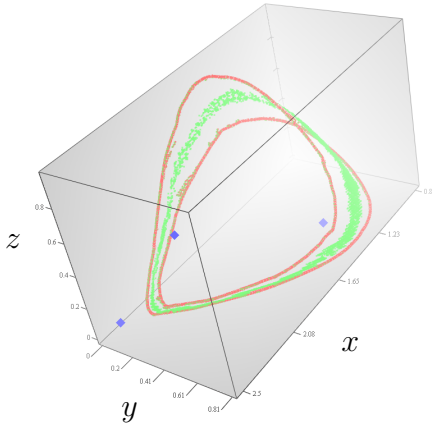
In order to get a better impression how the construction process works, Fig. 5 shows the results of several subdivision steps. Hereby, 17 scan points per box are taken. The rough position of the attractor can be located after the second subdivision of the domain space M into $200 \times 200 \times 32$ regions, see Fig. 5(a), then, in the third subdivision, see Fig. 5(b), the principal shape of the attractor becomes visible. But only after the fourth subdivision into $1\,200 \times 1\,200 \times 192$ regions, see Fig. 5(c), the symbolic image splits into two different sets of equivalent recurrent cells, which correspond to the stable and



(a) second subdivision step



(b) third subdivision step



(c) fourth subdivision step

Figure 5: *Discrete food chain model: Numerically calculated fixed points and three subdivision steps of the symbolic image construction. The outer covering of the chain recurrent set (green) as well as the attractor (red) are shown. Note that in this example the attractor was computed by forward iterates.*

unstable invariant manifolds. In order to achieve these results, it is necessary to compute the symbolic image graph for the iterated function $\mathbf{f}^{[40]}$ in the third subdivision and for $\mathbf{f}^{[80]}$ in the fourth subdivision step. Otherwise, the principal shape of the cone, see Fig. 5(a), would persist during further subdivisions. Additionally, reconstruction of the fragmented parts must be applied in order to avoid that the cycles vanish. The final result, see Fig. 4, is computed after the sixth subdivision. Note that in the subdivisions 5 and 6, also the function $\mathbf{f}^{[40]}$ is used and reconstruction of the cycles applied.

8 Conclusion

Several tunings for the practical application of symbolic analysis were proposed. We introduced the application of higher function iterates and the reconstruction of fragmented solutions in order to improve the numerical computations. Furthermore, the usage of these methods for dynamical systems continuous in time was discussed. Hereby, we stated that a rough approximation of the system flow is often sufficient for the investigation task. Indeed, a too precise simulation of the system flow can often hardly be computed and does not provide better results. The techniques we proposed are not in accordance with the original concepts of symbolic analysis. Nevertheless, they proved to be helpful in case the application of the original method fails or provides a solution of low precision due to clustering. Depending on the dynamics and dimension of the underlying system, the application of the tuning techniques can be a necessity in order to achieve appropriate results. Two numerical case studies were given in order to illustrate the application of the tunings.

The tunings we proposed show also that the capabilities of the investigation methods based on symbolic analysis depend to a large part on the solution of those problems which occur in the implementation and practical application of the theoretical concepts. This field of research is still wide open, and we believe that there is a large, yet undiscovered, potential for the development and application of further tuning methods.

Acknowledgements

This work was performed as a joint research project of the University of Stuttgart and the St. Petersburg State Polytechnic University, Russia. The author thanks

V. Avrutin, G. Osipenko and M. Schanz for discussions and the help provided for the realization of this project.

References

[1] Home page of the AnT 4.669 project, 2005. Available at `http://www.AnT4669.de`.

[2] V. M. Alekseev. *Symbolic Dynamics*. 11th Mathematical School, Kiev, 1976. In Russian.

[3] V. Avrutin, R. Lammert, M. Schanz, G. Wackenhut, and G. S. Osipenko. On the software package AnT 4.669 for the investigation of dynamical systems. In G. S. Osipenko, editor, *Fourth International Conference on Tools for Mathematical Modelling*, volume 9, pages 24 – 35. St. Petersburg State Polytechnic University, Russia, June 2003.

[4] R. Bowen. Symbolic dynamics. *Ann. Math. Soc.*, 8, 1982.

[5] C. Conley. Isolated invariant set and the morse index. *CBMS Regional Conference Series*, 38, 1978.

[6] M. Dellnitz and A. Hohmann. The computation of unstable manifolds using subdivision and continuation. In H. W. Broer, S. A. van Gils, I. Hoveijn, and F. Takens, editors, *Nonlinear Dynamical Systems and Chaos*, volume 19, pages 449 – 459. Birkhäuser, 1996.

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

[8] M. Dellnitz and O. Junge. An adaptive subdivision technique for the approximation of attractors and invariant measures. *Comput. Vis. Sci.*, 1:63 – 68, 1998.

[9] M. Dellnitz and O. Junge. Set oriented numerical methods for dynamical systems. In G. Iooss B. Fiedler and N. Kopell, editors, *Handbook of Dynamical Systems II: Towards Applications, World Scientific*, pages 221 – 264, 2002.

[10] D. Fundinger. Analysis of symbolic images and integration into the AnT-project. Diploma Thesis, University of Stuttgart, 2003.

[11] D. Fundinger. On the investigation of attractors and repellers: Algorithms for the localization of basins and construction of filtrations. Submitted to Discrete and Continuous Dynamical Systems – Series B, 2004.

[12] D. Fundinger and G. S. Osipenko. Computation of attractors and their basins. In G. Osipenko, editor, *Fourth International Conference on Tools for Mathematical Modelling*, volume 9, pages 193 – 207. St. Petersburg State Polytechnic University, Russia, June 2003.

[13] C. S. Hsu. A theory of Cell-to-Cell mapping dynamical systems. *Journal of Applied Mechanics*, 47:931 – 939, 1980.

[14] C. S. Hsu. *Cell-to-Cell Mappings*. Springer, N.Y., 1987.

[15] S. Y. Kobjakov, D. Y. Matiassevitch, and G. S. Osipenko. Location of the invariant set. In G. Osipenko, editor, *Fourth International Conference on Tools for Mathematical Modelling*, volume 9, pages 300 – 307. St. Petersburg State Polytechnic University, Russia, June 2003.

[16] D. Lind and B. Marcus. *An introduction to symbolic dynamics and coding*. New York, 1995.

[17] T. Lindström. On the dynamics of discrete food chains: Low- and high-frequency behavior and optimality of chaos. *Journal of Mathematical Biology*, 45:396 – 418, 2002.

[18] E. N. Lorenz. Deterministic nonperiodic flow. *J. Atmos. Sci.*, 20:130–141, 1963.

[19] K. Mischaikow. Topological techniques for efficient rigorous computations in dynamics. *Acta Numerica*, 2002.

[20] G. S. Osipenko. On a symbolic image of dynamical system. *Interuniv. Collect. sci. Works*, pages 101 – 105, 1983. In Russian.

[21] G. S. Osipenko. The periodic points and symbolic dynamics. *Prog. Nonlinear Differ. Equ. Appl.*, 12:261 – 267, 1993.

[22] G. S. Osipenko. Localization of the chain recurrent set by symbolic dynamics methods. In *Proceedings of Dynamics Systems and Applications*, volume 1, pages 227 – 282. Dynamic Publishers Inc., 1994.

[23] G. S. Osipenko. Morse spectrum of dynamical systems and symbolic dynamics. *Proceedings of the 15th IMACS World Congress*, 1:25 – 30, 1997.

[24] G. S. Osipenko. Construction of attractors and filtrations. *Banach center publication*, 47:173 – 192, 1999.

[25] G. S. Osipenko. Spectrum of a dynamical system and applied symbolic dynamics. *Journal of Mathematical Analysis and Applications*, 252(2):587 – 616, 2000.

[26] G. S. Osipenko. *Lectures on symbolic analysis of dynamical systems.* St. Petersburg State Polytechnic University, 2004.

[27] G. S. Osipenko and S. Campbell. Applied symbolic dynamics: Attractors and filtrations. *Discrete and Continous Dynamical Systems*, 5(1, 2):43 – 60, 1999.

[28] G. S. Osipenko, J. V. Romanovsky, N. B. Ampilova, and E. I. Petrenko. Computation of the morse spectrum. *Journal of Mathematical Sciences*, 120(2):1155 – 1166, 2004.

[29] C. Sparrow. *The Lorenz Equations: Bifurcations, Chaos, and Strange Attractors.* Springer, N.Y., 1973.

[30] P. Walters, editor. *Symbolic dynamics and its applications.* American Mathematical Society, July 1991.