

# A study of rigorous ODE integrators for multi-scale set-oriented computations

Tomoyuki Miyaji

*Meiji Institute for Advanced Study of Mathematical Sciences, Meiji University, 4-21-1  
Nakano, Nakano-ku, Tokyo 164-8525, Japan*

Paweł Pilarczyk\*

*Institute of Science and Technology Austria, Am Campus 1, 3400 Klosterneuburg, Austria*

Marcio Gameiro

*Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, Caixa  
Postal 668, 13560-970, São Carlos, SP, Brazil*

Hiroshi Kokubu

*Department of Mathematics, Kyoto University, Kyoto 606-8502, Japan*

Konstantin Mischaikow

*Department of Mathematics, Hill Center-Busch Campus, Rutgers, The State University of  
New Jersey, 110 Frelinghusen Rd, Piscataway, NJ 08854-8019, USA*

---

## Abstract

We study the usefulness of two most prominent publicly available rigorous ODE integrators: one provided by the CAPD group ([capd.ii.uj.edu.pl](http://capd.ii.uj.edu.pl)), the other based on the COSY Infinity project ([cosyinfinity.org](http://cosyinfinity.org)). Both integrators are capable of handling entire sets of initial conditions and provide tight rigorous outer enclosures of the images under a time- $T$  map. We conduct extensive benchmark computations using the well-known Lorenz system, and compare the computation time against the final accuracy achieved. We also discuss the effect of a few technical parameters, such as the order of the numerical integration method, the value of  $T$ , and the phase space resolution. We conclude that COSY may provide more precise results due to its ability of avoiding the variable dependency problem. However, the overall cost of computations conducted using CAPD is typically lower, especially when intervals of parameters are involved. Moreover, access to COSY is limited (registration required) and the rigorous ODE inte-

---

\*Corresponding author; see [www.pawelpilarczyk.com](http://www.pawelpilarczyk.com)

The final version of this preprint has been published in *Applied Numerical Mathematics* on April 13, 2016; DOI [10.1016/j.apnum.2016.04.005](https://doi.org/10.1016/j.apnum.2016.04.005)

grators are not publicly available, while CAPD is an open source free software project. Therefore, we recommend the latter integrator for this kind of computations. Nevertheless, proper choice of the various integration parameters turns out to be of even greater importance than the choice of the integrator itself.

*Keywords:* dynamical system; time-1 map; chain recurrent set; rectangular grid; algorithm; rigorous numerics

dynamical system, time-1 map, chain recurrent set, rectangular grid, algorithm, rigorous numerics

*2010 MSC:* 65-05, 34-04, 65L05, 65L70

---

## 1. Introduction

Set-oriented computations for dynamical systems focus on collective analysis of trajectories starting in a given set. They are crucial for implementing Conley's topological approach [6] to qualitative analysis of dynamics. This approach provides means for concise rigorous description of global dynamics. At the center of this approach is the notion of a Morse decomposition. It is a finite collection of isolated invariant sets that encompass all the chain recurrent dynamics, so that the dynamics in the remainder of the phase space is gradient-like.

Starting with a decomposition of the phase space into a finite grid, computation of rigorous outer bounds for the images of these sets allows one to represent the dynamics by means of a multivalued mapping on grid elements. Fast graph algorithms can then be used to determine a collection of isolating neighborhoods that enclose a Morse decomposition. These neighborhoods are built of grid elements, and thus we shall call them a *combinatorial Morse decomposition*. If the system depends on some parameters, and one provides intervals of parameters for the computations instead of their individual values, then the combinatorial Morse decomposition is valid for all the parameters in these intervals. Conducting this procedure for gradual refinements of an initially coarse grid, and restricting finer computations to a subset that contains all the chain recurrent dynamics found at the previous stage, gives rise to a multi-scale approach that is much more efficient for dissipative systems than working with a single *a priori* chosen decomposition of the space. Moreover, conducting computations for small intervals that combine into large ranges of parameters allows one to determine continuation of individual Morse sets or entire Morse decompositions, and thus identify sets of parameters with equivalent qualitative dynamics. As a consequence, a database of all the types of outer approximations of global dynamics encountered in the investigated system across the parameter ranges can be built. Computation of the Conley indices [6, 30] of the combinatorial Morse sets, which can be done using the multivalued map on grid elements directly, provides additional topological information on the actual Morse sets. These computations are easily parallelizable, e.g., using a dynamic scheme proposed in [26].

The approach described above provides an automatic method for the analysis of global dynamics in multi-parameter discrete-time dissipative systems induced

by continuous maps; see [1, 3, 4, 13, 14, 27] for the introduction of the method and applications to specific models in population dynamics, theoretical physics and epidemiology. We emphasize the wide applicability of this method, which is due to the mild assumptions on the system, as well as admitting the situations where the generator of the dynamical system is not known precisely; indeed, the the only essential input to the set-oriented construction and topological computations is the multivalued map on the grid. Moreover, application of this method to a discrete-time dynamical system is straightforward if a formula for the underlying map is given. In such a situation, one can choose a rectangular grid in  $\mathbb{R}^n$ , and then determine the desired multivalued representation by direct evaluation of the formula for the map in interval arithmetic [22] and covering the computed images with grid elements.

Many phenomena, however, are modeled by continuous-time dynamical systems (flows). One possible way of generalizing this method to ODEs is through the time- $T$  map; that is, by means of the analysis of the discrete-time dynamical system obtained from the flow by considering the translation by a fixed time  $T > 0$ . For this purpose, however, one must compute outer bounds for solutions of the ODE on sets (grid elements). Effective rigorous integration of ODEs in this context is not a trivial task, because various aspects must be taken into consideration, such as the wrapping effect, the exponential growth of overestimates, and the dependency between the variables; we discuss these problems further in Section 2.1. An additional difficulty is that the computations must be conducted for intervals of parameters, so that the resulting outer bound is valid for all the parameters in the provided intervals.

As far as we are aware, there are currently two publicly available software packages that provide methods optimized for this purpose: CAPD [5] and COSY [7]; we introduce them in Sections 2.2 and 2.3, respectively. Both are based on the Taylor series expansion of solutions of ODEs, which facilitates the estimate and control of numerical errors during the numerical integration of ODEs, the two ingredients necessary for the rigorous computation. However, both methods differ considerably by the approach to the problem, and a decision on which to choose for a particular application is not obvious. Although the effect of set-oriented integration is mainly a trade-off between precision and efficiency, a different approach might provide a substantially different balance. Additionally, it is not the computation for individual grid elements but the cumulative effect of a multi-scale approach that makes the difference.

Taking this into consideration, our work is aimed at answering the question of finding an optimal method and tweaking its parameters in the context of multi-scale rigorous set-oriented computations. For this purpose, we conduct an extensive benchmark of COSY and CAPD, in order to verify to what extent they are capable of accomplishing the goals in an effective and efficient way. Namely, we apply the method introduced in [1] to a time- $T$  map in the well known Lorenz system [16]. We consider a few different combinations of parameters of the Lorenz system (both single-valued parameters and intervals as well), a selection of different values for the order of the method, the depth of the subdivision of the phase space, the integration time  $T$ , and the number of steps into which this

integration time is subdivided by the integrator. This systematic comparison is expected to provide a reasonable foundation for the choice of the integration method and the associated specific settings. There are two quantities that we focus on while evaluating the methods: the CPU time, which corresponds to the cost of the computation, and the volume of the final Morse sets obtained, which is an indicator of the quality of the result: the smaller the volume, the tighter (and thus better) the outer estimates.

One may be tempted to compare the effectiveness of the two integrators by merely computing the sizes of the resulting images to compare their accuracy. However, integration parameters of both integrators may be tweaked so that one can outperform the other, often at the expense of speed. Moreover, there is risk that in various regions of the phase space, the integrators may perform differently; for example, high expansion far from the chain recurrent set may result in deteriorated accuracy and speed, but this is not relevant for the set-oriented computations that are eventually restricted to a neighborhood of the chain recurrent set and connecting orbits. Therefore, we decided to conduct the rather convoluted comprehensive test to see how both integrators actually perform in a practical application.

The results of our computations are shown and discussed in Section 3. These results suggest that COSY may provide more precise results due to its ability of avoiding the variable dependency problem, which is especially important if the integration time  $T$  is large. However, the overall cost of computations conducted using CAPD is typically lower, especially when intervals of parameters are involved, as opposed to single parameter values. Therefore, we recommend the latter integrator for this kind of computations.

Our work constitutes an essential step towards the generalization of the method that was introduced in [1] for discrete-time semidynamical systems (induced by maps). However, since the computations for ODEs are much more demanding than for maps, simple application of the method introduced in [1] to a time- $T$  map turns out not to be efficient enough. Larger overestimates encountered in the computation of outer bounds for images of grid elements prompt the need for additional improvements before the method can be effectively applied in practice. Out of several possibilities, in Section 5 we briefly outline two ideas for such improvements, and comment on the difficulties that arise when they are applied.

The structure of the paper is as follows: In Section 2, we discuss selected problems that appear in the computation of possibly tight outer bounds for images of sets under the translation by time  $T$  in the flow induced by an ODE, and we introduce the two methods aimed at providing effective solutions, based on the Taylor series expansion and represented by the CAPD and COSY software. In Section 3, we conduct an extensive benchmark comparison of the two integrators applied to a practical problem of computing Morse decompositions in the Lorenz system [16] for a few selected values of the parameters. Section 4 contains the conclusion from our computations. The paper ends in Section 5 with remarks on further work that needs to be done in order to make the set-oriented approach introduced in [1] applicable to ODEs in practice.

## 2. Taylor-series methods for ODE integration

There are several challenges related to computing the images of sets by the time- $T$  map in a flow given by an ODE. We discuss these challenges below, and we describe two most prominent numerical methods for the computation of rigorous outer bounds for  $\varphi_T = \varphi(\cdot, T)$  on sets. Both use a sophisticated approach in order to achieve optimal results. One method was initially introduced by R.J. Lohner [15] and implemented by D. Wilczak and T. Kapela in the CAPD software [5]. The other method was developed by K. Makino and M. Berz [2, 18], and implemented using the COSY Infinity software package [7].

### 2.1. Introduction

#### 2.1.1. Preliminaries

Consider a vector field on  $\mathbb{R}^n$  that depends on  $m$  parameters:

$$f: \mathbb{R}^n \times \mathbb{R}^m \ni (x, \lambda) \mapsto f_\lambda(x) \in \mathbb{R}^n.$$

In order to simplify the notation, we shall omit the explicit specification of the parameter  $\lambda$  unless necessary. Let  $\varphi: \mathbb{R}^n \times \mathbb{R} \ni (x, t) \mapsto \varphi(x, t) \in \mathbb{R}^n$  denote the flow induced by the vector field  $f$ . Let  $T > 0$ . The *time- $T$  map* is the translation by the time  $T$  in the flow  $\varphi$ , that is, the map  $\varphi_T: \mathbb{R}^n \ni x \mapsto \varphi(x, T) \in \mathbb{R}^n$ .

Let  $\mathbb{I}\mathbb{R}$  denote the set of all the closed intervals with endpoints in  $\mathbb{R}$ . We shall denote the elements of  $\mathbb{I}\mathbb{R}^n$  as variables in brackets, e.g.,  $[x]$  or  $[y]$ . Note that these should be treated as separate symbols that are different from  $x$ ,  $y$  etc. In fact,  $x$  may occasionally denote a real number contained in  $[x]$ .

#### 2.1.2. Sources of overestimates

Since computing the time- $T$  map is a much more complicated process than simple evaluation of a map formula with interval arithmetic, the following three sources for overestimates that generally appear in this type of computation must be seriously taken into consideration: (1) errors introduced by the numerical method; (2) dependencies between variables in expressions; and (3) the wrapping effect. We discuss these problems below in more detail. There are also round-off errors due to the fact that the real numbers are represented in the computations by a finite set of rational numbers, and thus the result of each arithmetic operation must be approximated, unless it is already a representable number (which, unfortunately, happens very rarely). Although these errors tend to accumulate, they are typically by orders of magnitude smaller than the other three sources of inaccuracy. Moreover, increasing the precision of arithmetic in use may eradicate the problem, should the influence of these errors be significant. Therefore, although the round-off errors must be taken care of in rigorous numerical computation (e.g., by using interval arithmetic [22]), they do not need special attention and thus we do not discuss them beyond this point.

### 2.1.3. Errors introduced by the numerical method

Since there is no general direct formula for the evaluation of the time- $T$  map, it is necessary to use an approximate formula, for example, coming from a truncation of an infinite series to the first few terms, which corresponds to linearization or approximation of exact values by a polynomial expression; see Sections 2.2 and 2.3 for specific formulas based on Taylor series expansion. As a consequence, a method for the computation of the time- $T$  map inevitably introduces errors due to such truncations.

Since these errors grow considerably with the increase in the time step  $T$ , it is often very beneficial to split the time  $T$  into several steps:  $T = k\tau$ , where  $k \in \mathbb{N}$  (e.g.  $k = 20$ ), and to evaluate  $k$  times the map  $\varphi_\tau$ , each time taking the previously computed image as the input to the next step of the method.

### 2.1.4. Dependencies between variables in expressions

The coordinates of  $[x]$  appear several times in the formulas that bound  $\varphi_T([x])$ , and thus straightforward evaluation of these formulas using interval arithmetic may introduce considerable overestimates. To explain this phenomenon, consider  $p(z) = z \cdot z$ . It is obvious that the range of the possible values attained by  $p$  on  $z \in [-1, 1]$  is  $[0, 1]$ . However, when evaluating this expression directly as  $[-1, 1] \cdot [-1, 1]$ , one forgets about the dependency between the two variables. Note that an outer bound for all the values that can be obtained by taking the product of numbers in these intervals is  $[-1, 1]$ .

In general, this problem is difficult to tackle. The key to a satisfactory solution is to handle the evaluation of expressions properly. The cost of such a solution is considerable. A solution is provided in the COSY Infinity software. COSY Infinity stores only the monomials whose coefficients are non-zero and greater than a certain threshold value. If the coefficient drops below the threshold, then the monomial is swept into the remainder interval. See [29] for details. Since the dependency on the initial data is represented by the polynomial part of a Taylor model and is carried through the entire integration process, the overestimation due to the dependency problem is effectively suppressed. Note that CAPD has very limited capabilities of addressing this problem.

### 2.1.5. The wrapping effect

Given a grid element  $Q$ , due to the possible rotation in the flow, as well as non-uniform speed of the flow in the various regions of the phase space along which the initial grid element travels, the actual shape of  $\varphi_t(Q)$  may be very different from a product of intervals (a rectangular box with edges parallel to the coordinate axes). Therefore, enclosing this image in a product of intervals may introduce considerable volume of unnecessary addition with which the actual image is “wrapped” (thus the name of the effect).

This problem can be addressed by using an appropriate class of representable sets which must be more general than products of intervals. Parallelepipeds, also combined with additional sets for storing small errors (coming from rounding or possibly from other sources) is proposed in CAPD; see e.g. [24] for a thorough

discussion of this subject. Taylor models which are, roughly speaking, images of  $[-1, 1]^n$  by polynomial functions, are proposed in COSY.

It is worth to mention that while computing the time- $T$  map in several steps, enclosing the intermediate sets in intervals increases the wrapping effect. Therefore, suitable chosen class of representable sets helps decreasing the wrapping effect also at this stage, so this technique helps in both problems (1) and (2).

## 2.2. Using the CAPD software

The Computer Assisted Proofs in Dynamics (CAPD) group develops a comprehensive C++ software library [5] that provides routines aimed at rigorous numerics for ODEs. In particular, a method for effective and efficient integration of ODEs is implemented, which computes outer enclosures of images of bounded sets of certain types under a time- $T$  map, given the vector field. The software was initiated in the 1990s by M. Mrozek, then co-developed with P. Zgliczyński (see [24, 33]) on the basis of the ideas by R.J. Lohner [15] with further developments. The currently available numerical methods for ODEs in the CAPD software were developed and implemented by D. Wilczak and T. Kapela. Unfortunately, the CAPD group have published very few papers in which they describe the mathematical methods behind the CAPD software; the interested reader is referred to [12, 31, 32] for details and sample applications.

In what follows, we assume that the elements of the grid  $\mathcal{G}$  are rectangular sets. Consider  $Q = [a_1, b_1] \times \cdots \times [a_n, b_n] \in \mathcal{G}$ . Let  $x = (\frac{a_1+b_1}{2}, \dots, \frac{a_n+b_n}{2})$  be the central point of the box  $Q$ . A straightforward approach to the computation of a rigorous enclosure  $R(Q)$  of  $\varphi_t(Q)$  is based on the Taylor series expansion of the map  $t \mapsto \varphi(x, t)$  at  $t = 0$ . Note that the derivatives of  $\varphi(x, t)$  with respect to  $t$  can be computed using the vector field  $f$  (the right-hand side of the ODE). One would truncate the series to a few initial terms and use one of the typical formulas that bound the remainder in order to provide rigorous outer bounds for the result. Eventually, one would apply interval arithmetic to compute an outer bound for  $\varphi_t(Q)$  in terms of a product of intervals.

More specifically, given some  $r \in \mathbb{N}$ , it is known that for each  $i \in \{1, \dots, n\}$ , there exists  $c_i \in (0, T)$  such that

$$\varphi_i(x, T) = x_i + \sum_{k=1}^r \frac{t^k}{k!} \frac{d^k}{dt^k} \varphi_i(x, t)|_{t=0} + \frac{t^{r+1}}{(r+1)!} \frac{d^{r+1}}{dt^{r+1}} \varphi_i(x, t)|_{t=c_i}. \quad (1)$$

The number  $r$  is called the order of the Taylor polynomial. Due to the equality  $\varphi_i(x, c_i + t) = \varphi_i(\varphi(x, c), t)$ , in this formula one can replace the  $(r+1)$ -st derivative of  $\varphi_i(x, t)$  at  $t = c_i$  with the derivative of  $\varphi_i(\varphi(x, c), t)$  at  $t = 0$ . Therefore, (1) provides a rigorous bound for  $\varphi(x, t)$  if one evaluates this expression in interval arithmetic, where  $x$  is replaced with the interval vector  $[x]$ , and the term  $\varphi(x, t)|_{t=c}$  with  $\varphi([v], t)|_{t=0}$ , where the interval vector  $[v]$  is an enclosure for the trajectories starting at any point in  $[x]$  and running up to the time  $T$ , that is,  $\varphi([x], [0, T]) \subset [v]$ . The way of computation of a vector  $[v]$  proposed in [33] is relatively simple and boils down to an iterative procedure for finding a possibly

small vector  $[y]$  such that

$$[x] + [0, T] \cdot f([y]) \subset \text{int}[y]. \quad (2)$$

Then  $[v]$  can be taken as a (possibly small) representable interval vector containing the left-hand side of this inclusion. Note that the condition (2) was originally proposed in [22]. At this point we would like to make a remark that, instead of the approach based on (2), which can be called a first order rough enclosure, one can use a high order rough enclosure method proposed in [25]. This is indeed included in the default ODE solver provided by the CAPD software. We did some test computations using both possibilities, and it turned out that the difference is negligible in our case, most likely due to the low order of the method that we use and the large grid size (see Section 3).

The remainder in (1) includes the derivative of  $\varphi_i(\varphi(x, c_i), t)$  at  $t = 0$ , where  $c_i \in (0, T)$ . If the interval  $(0, T)$  is large then the trajectory starting at  $x$  may travel through considerable amount of the phase space, and thus the estimate for  $\varphi(x, c_i)$  for all  $c_i \in (0, T)$  may be large, especially if the range for all possible  $x \in [x]$  to consider is also significant (e.g. if one tries to integrate large grid elements).

Out of the several types of representable sets that can be used for computing the images of grid elements and are implemented in the CAPD software, we are going to use the sets of the form  $x + C \cdot [r_0] + B \cdot [r]$ , where  $x \in \mathbb{R}^n$  corresponds to the center of the set,  $B$  and  $C$  are  $n \times n$  matrices, and  $[r_0]$  and  $[r]$  are interval vectors, with the parallelepiped  $C \cdot [r_0]$  corresponding to the basic part of the set, and  $B \cdot [r]$  storing the QR-decomposition of the remaining errors. These sets are called *doubletons* in [24, §8.4], with the second component in the form of a *cuboid* [24, §8.3], and are implemented as the class `CORect2Set` in the CAPD software library [5]. We refer the interested readers to these sources for more detailed information on these sets. Our choice is motivated by the smallest outer enclosures typically obtained using these sets in comparison to the other classes available in the CAPD library.

Let  $\mathcal{G}$  be the collection of grid elements at a certain subdivision depth. Suppose that we would like to compute a combinatorial Morse decomposition of  $\mathcal{G}$  with respect to  $\varphi_T$  for some  $T > 0$ . Let  $s > 0$  denote the number of steps one would like to make, and  $\tau > 0$  denote the integration step size. Let  $\psi_\tau$  denote the numerical procedure for computing a representable set that is an enclosure of a given representable set by  $\varphi_\tau$ . For all  $Q \in \mathcal{G}$ , we compute  $\mathcal{F}(Q) \cap \mathcal{G}$  as follows:

1. Let  $R_0$  be a (possibly small) representable set that contains  $Q$ . In practice, we take a `CORect2Set` in which  $x$  is a representation of the central point of  $Q$ ,  $B$  and  $C$  are the identity matrices,  $[r_0]$  is an interval vector such that  $Q \subset x + C \cdot [r_0]$ , and  $[r] = 0$ .
2. For  $i = 1, \dots, s$ , compute  $R_i := \psi_\tau(R_{i-1})$ .
3. Compute a possibly small interval vector  $\widehat{R}(Q)$  containing  $R_s$ . In practice, we take  $\widehat{R}(Q) := x + C \cdot [r_0] + B \cdot [r]$ , where  $(x, C, [r_0], B, [r])$  are the

components of  $R_s$ , and the formula is evaluated using interval arithmetic using representable numbers.

4. Cover  $\widehat{R}(Q)$  with grid elements to compute  $\mathcal{R}(Q) := \{Q_0 \in \mathcal{G} \mid Q_0 \cap \widehat{R}(Q) \neq \emptyset\}$ .

In this way, a combinatorial representation of  $\varphi_T$  can be effectively computed.

We would like to mention that in addition to the  $C^0$  computations described above, the CAPD software is also capable of computing higher order jets. Namely, the  $C^n$ -Lohner method [32] makes it possible to compute derivatives (up to the order  $n$ ) of solutions with respect to the initial conditions. In contrast to this, to the best of our knowledge, COSY software is not capable of conducting  $C^n$  computations.

### 2.3. Using the COSY software

COSY Infinity [7] is an interpreter of a programming language with its own style. In order to download the software, one has to request a user license (free of charge) through a registration form at their Web site [7]. COSY provides a data structure called the *Taylor model* (explained below), as well as operations on Taylor models such as addition, subtraction, multiplication, division, and integration. The numerical operations can be conducted using interval arithmetic [22], and thus provide mathematically rigorous results. COSY Infinity is written in Fortran, but a C++ interface is also provided, which allows one to handle Taylor models in a C++ program directly. While CAPD provides a rigorous solver of ODEs, COSY does not currently provide an integrator that uses interval arithmetic. As the authors of COSY explained [personal communication, February 2012], this situation was due to certain software patent issues, and ensured that they were working on a patent-free solution. Unfortunately, it seems that the issue is more serious than initially expected, and the solution was not yet available when we were preparing this paper. Therefore, we implemented our own integrator, as described below, in the hope that the Taylor model concept is so powerful that even a less optimized integrator based on COSY might compete with the CAPD software.

The theory of Taylor models has been established by Berz, Makino, and their collaborators. In this paper, we follow their theory with slightly different notation; this change is made for the sake of readability and consistency of the paper. As for the details of the theory and of implementation of COSY, see [2, 8, 11, 17, 18, 19, 29] and references therein.

A Taylor model is a data structure consisting of a polynomial map with floating-point coefficients and an interval vector. The polynomial map is regarded as an approximation of a map, and the interval vector is a rigorous error bound of the approximation. The domain of a polynomial map is normalized to  $[-1, 1]^n$  in practice. Let  $[\xi] = [-1, 1]^n \in \mathbb{IR}^n$ . Let  $Q = [a_1, b_1] \times \cdots \times [a_n, b_n] \in \mathcal{G}$ . Remark that  $Q$  is the image of  $[\xi]$  under the affine map  $g$  defined by

$$g(\xi) = \text{mid}(Q) + \text{diag}(\text{rad}([a_1, b_1]), \dots, \text{rad}([a_n, b_n]))\xi, \quad \xi \in [\xi], \quad (3)$$

where  $\text{mid}(Q)$  denotes the middle point of  $Q$ , and  $\text{rad}([a, b]) := (b-a)/2$  denotes the radius of the interval  $[a, b]$ .

By the Taylor model method, we obtain a pair  $(p, [i])$  such that  $\varphi_T(g(\xi)) - p(\xi) \in [i]$  holds for any  $\xi \in [\xi]$ , where  $p: \mathbb{R}^n \rightarrow \mathbb{R}^n$  is a polynomial map and  $[i] \in \mathbb{IR}^n$ . Then  $\varphi_T(Q) = \varphi_T(g([\xi]))$  is contained in the set  $p([\xi]) + [i]$ . Since  $p$  is a polynomial map, an interval enclosure of  $p([\xi]) + [i]$ , and hence of  $\mathcal{F}(Q) \cap \mathcal{G}$ , is computable by machine interval arithmetic.

For all  $Q \in \mathcal{G}$ , we compute  $\mathcal{F}(Q) \cap \mathcal{G}$  as follows:

1. Let  $g$  be the affine map defined by (3) above. Let  $p_0: \mathbb{R}^n \rightarrow \mathbb{R}^n$  be a polynomial and let  $[i_0] \in \mathbb{IR}^n$  satisfying  $g(\xi) - p_0(\xi) \in [i_0]$  for any  $\xi \in [\xi]$ . In practice, the coefficients of  $p_0$  are floating-point approximations of coefficients of  $g$ , and  $[i_0]$  is an interval vector containing the rounding errors.
2. For  $k = 1, \dots, s$ , compute a polynomial map  $p_k$  and an interval vector  $[i_k] \in \mathbb{IR}^n$  such that  $\varphi_{t_k}(g(\xi)) - p_k(\xi) \in [i_k]$  for any  $\xi \in [\xi]$ .
3. Compute an interval vector  $\hat{R}(Q)$  containing  $p_s([\xi]) + [i_s]$ . Note that  $\varphi_T(Q) = \varphi_T(g([\xi])) \subseteq p_s([\xi]) + [i_s] \subseteq \hat{R}(Q)$ .
4. Cover  $\hat{R}(Q)$  with grid elements to compute  $\mathcal{F}(Q) := \{Q_0 \in \mathcal{G} \mid Q_0 \cap \hat{R}(Q) \neq \emptyset\}$ .

Additional explanation is necessary on the second and third step above. At the third step, we compute  $\hat{R}(Q)$  containing  $p_s([\xi]) + [i_s]$ . COSY provides two functions to do it: `IN` and `LDB`. While `IN` simply computes  $p_s([\xi]) + [i_s]$  using interval arithmetic, `LDB` solves an optimization problem to bound  $p_s([\xi]) + [i_s]$ . The latter is more accurate but slower than the former. We use `IN` because we have to call this function very many times. As we will see in the next section, the accuracy is sufficient for our purpose.

At the second step, we solve the initial value problem of ODEs by the Taylor model method, which was originally proposed by Berz and Makino [2] and developed by Berz, Makino, and their collaborators. COSY provides only data structures for floating point numbers, intervals, vectors with floating point components, interval vectors, strings, booleans, polynomials, and Taylor models, and their fundamental operations and elementary functions. We implemented the Taylor model integrator on our own using these data structures.

Our implementation of the COSY integrator is based on putting together the various bits and pieces contained in [2], [8, Algorithm 2.30], and [19]. We describe the procedure carefully, because we would like to put emphasis on the implementation details.

1. (Set the initial condition. See [19, Section 2].) Let  $g$  be the affine map defined by the equation (3) above. Let  $p_L: \mathbb{R}^n \rightarrow \mathbb{R}^n$  be a polynomial and let  $[i_L] \in \mathbb{IR}^n$  be an interval vector satisfying  $g(\xi) - p_L(\xi) \in [i_L]$  for any  $\xi \in [\xi]$ . Let  $p_R: \mathbb{R}^n \rightarrow \mathbb{R}^n$  be the identity map and let  $[i_R] = [0, 0]^n \in \mathbb{IR}^n$ . Let  $\mathcal{U}_L = (p_L, [i_L])$ , and  $\mathcal{U}_R = (p_R, [i_R])$ . Let  $t = 0$ .

2. (Find an approximate solution. See [2, Section 3.5] and [8, Algorithm 2.30 1(a)].) Compute the following sequence of polynomials by Picard iteration

$$q_0(\tau, \xi) = p_L(\xi), \quad q_{n+1}(\tau, \xi) = p_L(\xi) + \int_0^\tau f(q_n(s, \xi)) ds$$

for  $n = 1, \dots, N$ , where all the polynomials are truncated up to  $N$ -th order terms. Let  $q = q_N$ , which is an approximate polynomial of the solution on the current time interval.

3. (Prove the existence of a solution. See [2, Section 3.5] and [8, Algorithm 2.30 1(b)].) Apply an inflation technique to obtain an interval vector  $[j] \in \mathbb{IR}^n$  satisfying  $\mathcal{P}(\mathcal{V}) \subset \mathcal{V}$ , where  $\mathcal{V} = (q, [j])$  and

$$\mathcal{P}(\mathcal{V}) = \mathcal{U}_L + \partial_\tau^{-1} f(\mathcal{V}).$$

At this step, the existence of a solution on the current time interval is proved. See [2] for the details of the proof.

4. (Compute a Taylor model for the image of the time- $t_k$  map. See [8, Algorithm 2.30 1(d)].) Substitute  $\tau = h$  into  $\mathcal{V} = (q, [j])$ , and let  $\tilde{\mathcal{U}}$  be the resulting Taylor model.  $\tilde{\mathcal{U}} \circ \mathcal{U}_R$  contains all the solutions at the next time grid.
5. (Perform the preconditioning. See [19].) In order to reduce wrapping effect, we apply the QR preconditioning method proposed by Makino and Berz [19, Def. 15 and Prop. 16].) Let  $\tilde{\mathcal{U}}_R := \mathcal{U}_R$ . Compute new  $\mathcal{U}_L$  and  $\mathcal{U}_R$  such that  $\mathcal{U}_L \circ \mathcal{U}_R \supseteq \tilde{\mathcal{U}} \circ \tilde{\mathcal{U}}_R$  and  $\mathcal{U}_L$  has an affine map as the polynomial part and has the zero interval as the remainder interval. Increment  $t$  by  $h$ .
6. Repeat the steps 2–5 until  $t = T$ .

We would like to remark that the calculations described in Step 2 above are conducted using the DA data structure in COSY, which allows one to compute elementary operations on polynomials and the integral while the calculations listed in Steps 3–5 are programmed using the TM data structure in COSY.

Let us briefly explain the idea of QR preconditioning; we refer to [19] for details. At each time step  $t = t_k$ , we have two Taylor models  $\mathcal{U}_L$  and  $\mathcal{U}_R$ . The image of the initial set by the time- $t_k$  map is represented by the composition of them.  $\mathcal{U}_L$  is chosen so that its polynomial part is affine and its remainder interval is zero.  $\mathcal{U}_R$  is rescaled so that its range is contained in the domain of  $\mathcal{U}_L$ . At steps 2 and 3, the Picard operator affects only  $\mathcal{U}_L$ , and  $\mathcal{U}_R$  does not change; see [19, Proposition 4]. Since the remainder interval of  $\mathcal{U}_L$  is very narrow, we can expect that the remainder interval of  $\mathcal{V}$ , which is one of the sources of wrapping effects, becomes narrow, too. Since the polynomial part of  $\mathcal{U}_L$  represents the coordinate system, choosing it appropriately helps us to reduce wrapping effects. In this sense, the QR preconditioning corresponds to QR factorization of the Lohner method. The preconditioning is not wrapping but factorization of a Taylor model, and the overestimation caused by preconditioning does not become a problem.

A question arises as to how often we should perform the QR preconditioning. When the integration time is short and the size of initial set is small, the naive Taylor model method without preconditioning is faster and gives the error bound comparable to the one that involves QR preconditioning. However, in our computations in the next section, the initial sets are relatively large. In those cases, the overestimates may grow rapidly. Therefore, we chose to perform the preconditioning at every time step in order to stabilize the integration process. As far as we have tested, this strategy is superior to the computation with preconditioning conducted at every other step (or every fifth step, or so), in both accuracy and computation time. It would be optimal to perform the preconditioning only when the size of the remainder interval exceeds some threshold. However, to the best of our knowledge, a general strategy for the choice of the optimal threshold is unknown; this topic requires further study.

Another method for reducing wrapping effects is called shrink wrapping [20]. It produces a new Taylor model which encloses the image of the initial set. However, the polynomial part of the resulting Taylor model is no longer a function of the initial values. As we need a Taylor model of time- $T$  map, we do not use shrink wrapping.

### 3. Benchmark comparison of CAPD and COSY

In order to compare the effectiveness and efficiency of the Lohner method (CAPD) and the Taylor model method (COSY), we conducted a series of computations of Morse decompositions, following the multi-scale set-oriented approach introduced in [1]. We used the Lohner method implemented in the CAPD software library, and the Taylor model integrator implemented using the C++ interface to COSY Infinity.

#### 3.1. The test problem

As a test problem, we chose the well known Lorenz system [16]

$$\begin{cases} \dot{x} = \sigma(y - x) \\ \dot{y} = x(\rho - z) - y \\ \dot{z} = xy - \beta z \end{cases}$$

We considered the following six cases of parameters:

1.  $(\sigma, \rho, \beta) = (10, 13, 8/3)$ ,
2.  $(\sigma, \rho, \beta) = (10, 18, 8/3)$ ,
3.  $(\sigma, \rho, \beta) = (10, 28, 8/3)$ ,
4.  $(\sigma, \rho, \beta) = (10, 13, 8/3) + 10^{-2}([-1, 1], [-1, 1], [-1, 1])$ ,
5.  $(\sigma, \rho, \beta) = (10, 18, 8/3) + 10^{-2}([-1, 1], [-1, 1], [-1, 1])$ ,
6.  $(\sigma, \rho, \beta) = (10, 28, 8/3) + 10^{-2}([-1, 1], [-1, 1], [-1, 1])$ ,

where in the latter three cases the parameters are given as intervals. In cases 1 and 4, there are three equilibria and connecting orbits from the trivial one to nontrivial ones. In cases 2 and 5, in addition to the previous cases, there are two unstable limit cycles and connecting orbits between equilibria and cycles. In cases 3 and 6, the system exhibits chaotic behavior.

We chose to conduct the computations restricted to the following bounding boxes in phase space:

- (i)  $[-16, 16] \times [-20, 20] \times [-5, 35]$  for cases 1 and 4,
- (ii)  $[-20, 20] \times [-25, 25] \times [-5, 45]$  for cases 2 and 5,
- (iii)  $[-25, 25] \times [-35, 35] \times [-5, 65]$  for cases 3 and 6.

These rectangular areas, further denoted by  $B$ , are large enough to capture the combinatorial dynamics exhibited by the Lorenz system, and leave some margins for overestimates.

### 3.2. The computational procedure

When computing an outer bound for a Morse decomposition, there are a few internal technical parameters that need to be set up; they are listed in Table 1. A Morse decomposition was computed separately for every valid combination of these parameters, except for `InitialDepth`, where a successful computation completed for some value implies that the computations for higher values were not conducted, because they would yield precisely the same result, except with a longer computing time.

	CAPD	COSY
<code>InitialDepth</code>	5, 6, 7	5, 6, 7
<code>FinalDepth</code>	5–12	5–12
<code>Order</code>	2, 3, 4, 6, 9	5, 7, 8, 10
<code>Time</code>	0.05, 0.1, 0.2	0.05, 0.1, 0.2
<code>Steps</code>	10, 30, 100	10, 30, 100
<code>CosyDepth</code>	N/A	2, 3, 4, 5

Table 1: Method-specific parameters used for the computations with CAPD and COSY: initial and final subdivision depth of the bounding box in the phase space, the order of the approximation polynomial, the integration time  $T$ , the number of integration steps into which  $T$  was subdivided, and the subdivision depth for the computation of the Taylor models.

In the case of COSY, each computation of a Morse decomposition was preceded by computing and storing Taylor models for outer bounds of the image by  $\varphi_T$  of each element of the  $2^{d_0} \times 2^{d_0} \times 2^{d_0}$  uniform cubical grid into which the bounding box  $B$  in the phase space was subdivided, where  $d_0$  equals the value of the parameter called `CosyDepth`. The procedure described in Section 2.3 was applied for this purpose. It costs much time to solve ODEs on a fine grid by COSY. However, once we have obtained a Taylor model on an interval, we can use it directly to compute the image of a subinterval. In particular, we do not need to solve ODEs again for this purpose. We take advantage of this property.

This preliminary step was not conducted in the case of using CAPD, because the method applied does not have this feature.

As a starting point for the combinatorial computations, the bounding box  $B$  in the phase space was subdivided into  $2^{d_1} \times 2^{d_1} \times 2^{d_1}$  boxes, with the value of  $d_1$  set to `InitialDepth`. An outer enclosure of the image of each of the boxes by  $\varphi_T$  was computed in terms of an interval vector, which was covered by boxes afterwards. For that purpose, the procedure described in Section 2.2 was applied (in the case of CAPD), or the precomputed Taylor polynomials were used directly (in the case of COSY). In this way, a combinatorial cubical multivalued map was computed, as a representation of an outer estimate for  $\varphi_T$  on grid elements.

A combinatorial procedure for computing an outer bound for the invariant part of the area of interest was then applied, and the collection of boxes was restricted to the computed subset. Then the boxes were subdivided, and the computation of outer bounds of the images of the smaller boxes was conducted, giving rise to a new combinatorial cubical multivalued map. This step was repeated until the desired target subdivision depth had been reached, indicated as `FinalDepth`. Then an outer bound of a Morse decomposition was computed using the combinatorial map at the final subdivision depth. This multi-scale approach was introduced and explained in more detail in [1, §4.2].

The degree of the Taylor polynomial used in the numerical integration of grid elements is called the `Order` of the method. The integration time  $T$  in the flow is called `Time` in Table 1. The image of each grid element was computed in an *a priori* defined number of steps; this number of steps is called `Steps` in Table 1, and corresponds to the number of subintervals into which the interval  $[0, T]$  was subdivided.

In the COSY computations, we used Taylor models with seven variables, namely for the time, the three-dimensional phase space, and the three parameters. In Cases 4–6, the parameters given as intervals had to be treated as separate variables in order to avoid the dependency problem. In order to avoid technical complications arising from the fact that `RE` (floating point number) and `TM` (Taylor model) are incompatible data types in COSY, we decided to simplify the code by treating at least one parameter (we chose  $\rho$  for that purpose) as a variable also in Cases 1–3. However, in these cases, the parameter  $\rho$  was fixed, so this implementation did not affect the computational cost.

Some constraints were put on the computations, and thus some combinations of the parameters were considered to lead to a failure even if the actual computations might have provided legitimate results. In particular, it was requested that none of the computations exceeded 8 GB of memory, and that the constructed combinatorial Morse decomposition was contained in the interior of the prescribed bounding box  $B$  in the phase space, so that the constructed sets were indeed isolating neighborhoods.

Apart from these constraints, the only possible reason for a computation to fail was the occurrence of uncontrollable increase in overestimates (a “blow-up” of the solution), which made the numerical integrator fail in computing a reasonable outer bound for the image of a grid element under  $\varphi_T$ . This typically

happened, for example, in the case of CAPD at low initial subdivision depths (such as 5), because of taking a grid that was too coarse (too large boxes to integrate), and also when trying to use the integration time that was too long, such as  $T = 0.2$  in some cases.

### 3.3. Results and discussion

A sample combinatorial Morse decomposition is shown in Figure 1. Although it was not possible to separate different periodic or stationary solutions in many cases of the computations, in general, the achieved accuracy in approximating the chain recurrent structures in most cases was very good.

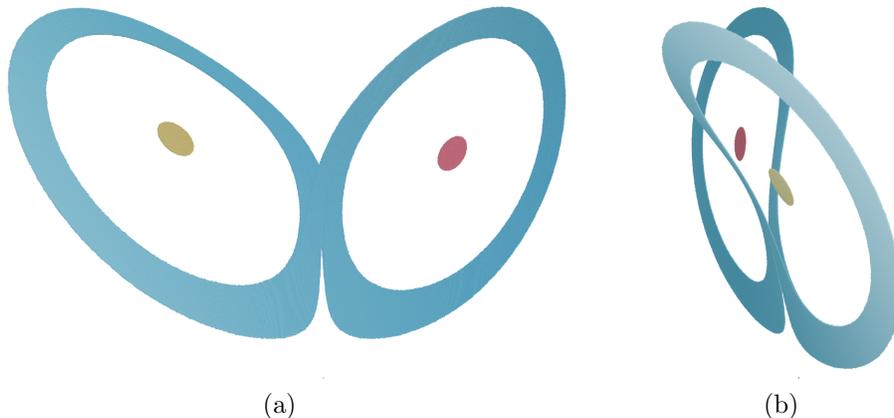


Figure 1: Two different projections (a) and (b) of a combinatorial Morse decomposition computed for Case 2 ( $r = 18$ ) using CAPD with the method order 3, the final subdivision depth of 11, and  $T = 0.1$  divided into 30 steps. The large combinatorial Morse set is an isolating neighborhood of two unstable limit cycles. A picture that looks the same was obtained using COSY with method order 5, `CosyDepth` 5, and the same other parameters.

#### 3.3.1. COSY vs. CAPD

The comparison between the results achieved using COSY and those obtained using CAPD is summarized in Figure 2. Each graph is a scatter plot of the computation time against the volume of the union of the Morse sets computed for each combination of the parameters listed in Table 1, with the exceptions discussed in Section 3.2. Results for Cases 1–3 are shown in the left-hand column, and those for Cases 4–6 are in the right-hand column. Red circles indicate the results of the Lohner method (CAPD), and blue squares correspond to those of the Taylor model method (COSY). Computations that caused a failure are not shown in these charts.

The analysis of the results shown in Figure 2 leads to the conclusion that in our tests both methods performed very well, although with some key differences.

For the case of single parameter values, COSY was slower than CAPD when computing with coarser resolutions (Morse sets with larger volumes) and it was faster than CAPD at finer resolutions (Morse sets with smaller volumes). On the

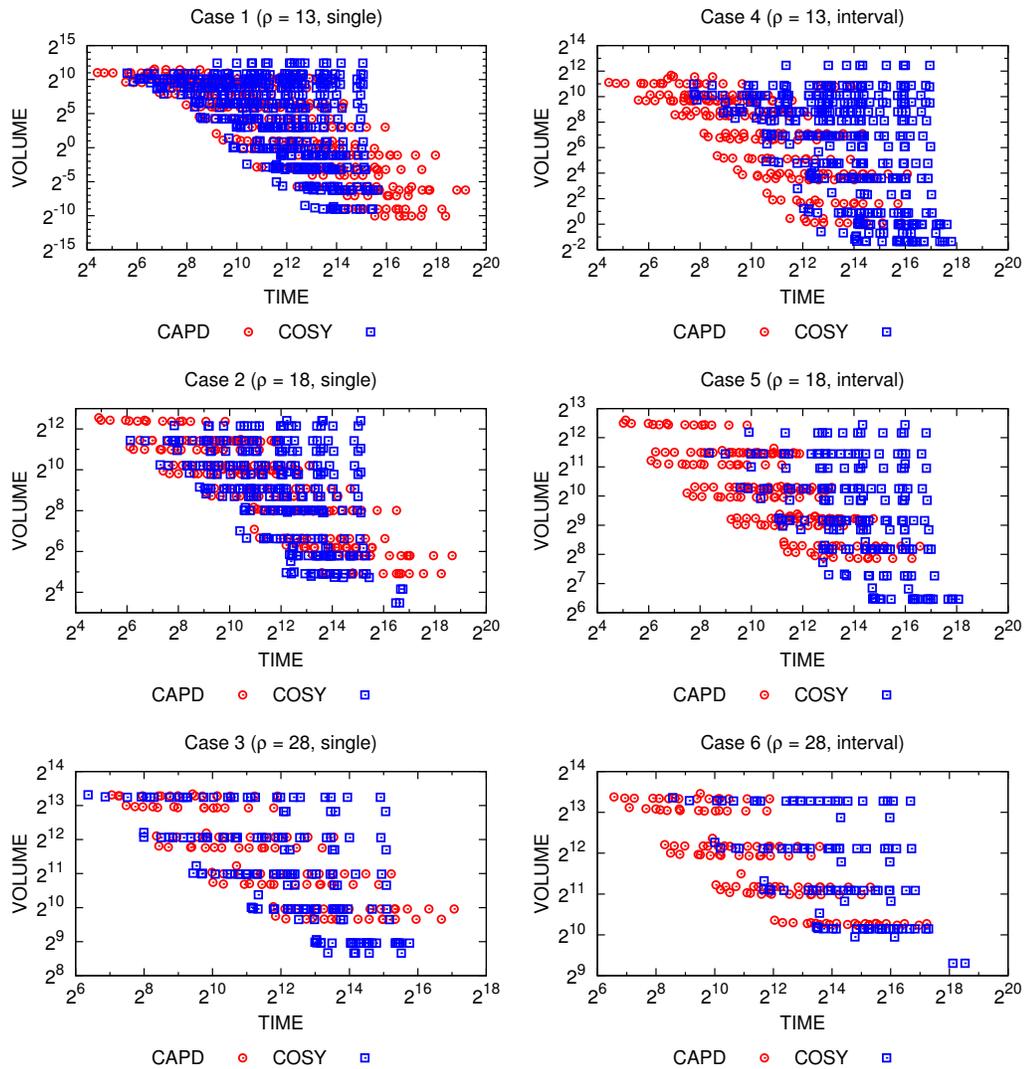


Figure 2: Scatter plots of the computation time (in seconds) against the total volume of the Morse sets for each combination of the parameters listed in Table 1 for which the computation was successful.

other hand, for the case of interval parameters, CAPD was always faster than COSY. In Cases 1–3, several CAPD computations were slower than those done by COSY. These cases were encountered for high values of `FinalDepth`. This is due to the fact that COSY spends most of the computational time constructing the Taylor models at a coarse resolution, and hence it does not need to solve the ODEs again when progressing to finer subdivision depths. The advantage of using this feature increases with the increase in the final subdivision depth. On the other hand, in Cases 4–6, COSY was much slower than CAPD. The main reason for this was the fact that COSY used Taylor models with seven variables, as explained above, and then the number of coefficients in the Taylor polynomials was very large; as a consequence, it turned out that the cumulative cost of the computation was higher than solving the initial value problems at each subdivision depth separately using CAPD.

In terms of accuracy, COSY seems to perform better than CAPD, for both the single parameter and the interval parameter cases. Indeed, the results of the computations performed by COSY are slightly more accurate than those produced by CAPD, specially at finer resolutions, since COSY produces Morse sets with smaller volumes than CAPD. A very likely reason for this may be the excellent way of tracking the dependencies between variables in the expressions.

### 3.3.2. Integration parameters

Tweaking the various parameters of the numerical computations may have considerable effect on the computational cost and the achieved accuracy. In the discussion below, the gain or loss from changing the parameters is expressed by means of the average  $\pm$  standard deviation of the results.

**FinalDepth.** In Table 2, evidence is shown that taking a higher final subdivision depth yields a much finer approximation of the constructed outer bound for a Morse decomposition. However, the computational cost if this improvement is considerable, both in terms of computation time and memory used (the latter not shown). Note that, in general, the actual proportions in this kind of results strongly depend on a particular system, and also on the proximity and hyperbolicity of the Morse sets.

**Order.** Increasing the order of the method results in the increase in the cost of computation of the image of each individual grid element by  $\varphi_T$ . On the other hand, since the accuracy of the result of such a computation is better, the constructed combinatorial map is smaller, and so are the computed outer bounds for the Morse sets. Therefore, increasing the order of the method may be beneficial up to the point where very little improvement can be achieved, mainly due to the coarseness of the boxes in comparison to the accuracy of the computation. Results of the comparison are gathered in Table 3. It is clear that increasing the order of the method from the lowest one taken into consideration to the next one is indeed beneficial, but the cost (in terms of the computation time) of further increases in the order is clearly not worth the gain. This seemingly contradicts the findings of other users of CAPD and COSY who found out that using higher orders was always beneficial. However, please note that, in the combinatorial method, the image of each cube computed by the

final depth increase	time increase [%]	volume decrease [%]
CAPD		
6 $\rightarrow$ 7	126 $\pm$ 34	63.0 $\pm$ 3.1
7 $\rightarrow$ 8	146 $\pm$ 51	62.9 $\pm$ 12.6
8 $\rightarrow$ 9	184 $\pm$ 48	66.8 $\pm$ 18.2
9 $\rightarrow$ 10	244 $\pm$ 39	70.5 $\pm$ 20.2
10 $\rightarrow$ 11	278 $\pm$ 87	83.9 $\pm$ 8.2
11 $\rightarrow$ 12	168 $\pm$ 65	88.5 $\pm$ 1.0
COSY		
5 $\rightarrow$ 6	104 $\pm$ 16	68.9 $\pm$ 0.2
6 $\rightarrow$ 7	131 $\pm$ 21	59.5 $\pm$ 1.8
7 $\rightarrow$ 8	182 $\pm$ 30	61.1 $\pm$ 9.8
8 $\rightarrow$ 9	208 $\pm$ 24	68.8 $\pm$ 16.9
9 $\rightarrow$ 10	244 $\pm$ 23	73.7 $\pm$ 19.5
10 $\rightarrow$ 11	258 $\pm$ 30	78.6 $\pm$ 14.6
11 $\rightarrow$ 12	169 $\pm$ 42	86.0 $\pm$ 6.6

Table 2: Additional cost (in terms of the computation time) of increasing the final subdivision depth, and the gain in terms of the decrease in the volume of the constructed Morse decomposition (average relative difference  $\pm$  standard deviation).

rigorous integration method is covered by cubes with respect to the grid that is much coarser than the gain in accuracy achieved by a higher order method, except for the lowest orders, where the accuracy is indeed very low.

**Time.** Taking a larger translation time  $T$  in the flow may contribute to the increase in the overestimates on the one hand, but may also, on the other hand, allow the contraction and expansion present in the system to come into effect and make the constructed Morse sets considerably smaller. It is not easy to find a proper balance between these two. As shown in Table 4, increasing  $T$  from 0.05 to 0.1 in our experiments reduced the overall computation cost and increased the accuracy of the resulting Morse decomposition. Increasing  $T$  further, however, resulted in an increase in the computation cost, and further improvement in the final accuracy of the computed Morse sets was observed.

**Steps.** The number of steps generally affects the computational cost directly (see Table 5), because the cost of computing  $\varphi_\tau$  is essentially independent of the value of  $\tau$ . Although increasing the number of steps helps decrease the overestimates and thus constructing smaller bounds on the chain recurrent sets, the gain is not as significant as the increase in the cost. Taking this into account, it turns out that with a fixed integration time  $T$ , the best strategy is to take as few steps as possible, while still avoiding the solution “blow-up” effect. Indeed, the gain resulting from further increases in the number of steps is usually marginal.

We would like to point out the fact that, in general, using a variable step size might be more beneficial than keeping the step size fixed. However, choosing the optimal step size is a highly nontrivial task, because one must balance the overestimates which increase with the increase in the step size, as well as their

order increase	time increase [%]	volume decrease [%]
CAPD		
2 → 3	27.8 ± 7.8	2.011 ± 4.237
3 → 4	27.6 ± 7.3	0.450 ± 2.069
4 → 6	54.6 ± 8.2	0.057 ± 0.391
6 → 9	67.3 ± 7.3	0.001 ± 0.008
COSY		
5 → 7	230.1 ± 155.2	4.764 ± 11.845
7 → 8	59.3 ± 20.5	0.329 ± 1.854
8 → 10	143.9 ± 73.2	0.176 ± 1.434

Table 3: Additional cost (in terms of the computation time) of increasing the order of the method, and the gain in terms of the decrease in the volume of the constructed Morse decomposition (average relative difference ± standard deviation).

increase in the integration time $T$	time increase [%]	volume decrease [%]
CAPD		
0.05 → 0.1	-23.0 ± 10.6	35.4 ± 24.8
0.1 → 0.2	53.1 ± 74.7	38.8 ± 45.9
COSY		
0.05 → 0.1	-0.03 ± 10.7	48.8 ± 24.9
0.1 → 0.2	7.3 ± 10.9	47.8 ± 38.1

Table 4: Additional cost (in terms of the computation time) of increasing the integration time  $T$  in the flow and the gain in terms of the decrease in the volume of the constructed Morse decomposition (average relative difference ± standard deviation).

increase in the number of steps	time increase [%]	volume decrease [%]
CAPD		
10 → 30	174.9 ± 30.9	2.73 ± 4.75
30 → 100	218.9 ± 19.7	1.27 ± 2.35
COSY		
10 → 30	72.1 ± 62.0	2.86 ± 9.94
30 → 100	120.8 ± 75.3	0.71 ± 3.06

Table 5: Additional cost (in terms of the computation time) of increasing the number of integration steps, and the gain in terms of the decrease in the volume of the constructed Morse decomposition (average relative difference ± standard deviation).

increase in CosyDepth	time increase [%]	volume decrease [%]
2 $\rightarrow$ 3	70.5 $\pm$ 105.0	4.80 $\pm$ 12.91
3 $\rightarrow$ 4	218.3 $\pm$ 202.6	2.34 $\pm$ 8.90
4 $\rightarrow$ 5	418.3 $\pm$ 221.7	1.83 $\pm$ 5.41

Table 6: Additional cost (in terms of the computation time) of increasing the subdivision depth for the computation of the Taylor Models when using the COSY integrator, and the gain in terms of the decrease in the volume of the constructed Morse decomposition (average relative difference  $\pm$  standard deviation).

accumulation effect, against the computation time which decreases if the steps are larger (because the number of steps that must be conducted to reach the prescribed value of  $T$  becomes smaller). In the ODE integration software, automatic determination of the time step is usually done on the basis of heuristics based on some rough estimates. For example, CAPD software offers a method for computing an outer bound for  $\varphi_T$  in which the step size is determined on the basis of requested error tolerance at each step. However, smaller errors that propagate through many steps may have a worse cumulative effect than large errors committed in a smaller number of steps. We conducted a small benchmark with a few dozens sample computations, trying a selection of different levels of error tolerance. The tests showed that using a low order method typically results in much longer computation times (even more than 10 times longer in our test) with virtually no difference in the final Morse decomposition size, independent of the error tolerance. On the other hand, using a high order method may result in decreased computation time (up to the factor of 20), but the resulting Morse decomposition is typically very large, unless a much lower error tolerance is chosen, which in turn results in much less significant gain in computation time. To sum up, our experiment showed that clever algorithms for automatic step determination are necessary in order to consistently outperform the approach based on the fixed step size in all cases.

**CosyDepth.** Finally, increasing the depth of the subdivision of the phase space at which the Taylor Models were computed when using the COSY integrator, resulted in considerable increase in the computation cost (see Table 6). Unfortunately, the quality of the final result, measured as the volume of the constructed Morse decomposition, was not improved significantly. This feature shows the usefulness of the concept of a Taylor Model for computing bounds for images of subsets of the domain on which the Taylor Model has already been computed.

#### 4. Conclusion

The results of our comparison indicate that if one prizes accuracy in the integration of a set of initial conditions, COSY is the better of the two options. On the other hand, if the emphasis is on better computational time, and very high

accuracy is not the crucial feature, then CAPD is the better option, especially in the case of interval parameters.

Taking all the above into consideration, we conclude that the CAPD software package provides a more suitable ODE integrator for the purpose of multi-scale set-oriented computations (like those tested in our experiments) than the COSY Infinity software. However, proper choice of the integration time  $T$  and the phase space subdivision depth, as well as adjustment of the various integration parameters (such as the order of the method and the number of integration steps) is of even greater importance for achieving valuable results at reasonable cost than the choice of the integrator itself.

## 5. Future work

In order to conduct the computations in the spirit of [1] for flows, one can apply the time- $T$  map approach as follows: One can compute the Morse decompositions as well as the Conley indices of the constructed isolating neighborhoods of the Morse sets, with respect to the discrete-time dynamical system obtained as a time- $T$  translation in the flow, for some  $T > 0$ . It is known that the Morse decompositions constructed in this way are valid for the flow, too, and so are the Conley indices, and the information about connecting orbits as well; see [23] for the appropriate theorems, and [13, §5.2] for a more detailed explanation.

Unfortunately, there are several obstacles that make it difficult to apply the time- $T$  map approach to the multi-scale set-oriented method [1] directly. This is mainly because of considerably larger overestimates in the computation of outer approximations of images of grid elements by the time- $T$  translation in the flow than those that are normally encountered if the map is given directly by a simple formula. Below we discuss two possible improvements and comment on the difficulties that arise when one attempts to apply them.

The first improvement concerns the way in which the results of integration are covered by grid elements. The method and software [10, 21, 28] used for the computation of the homological Conley index in [1] requires that the constructed combinatorial map is acyclic, which can be achieved in the most reliable and straightforward way by using convex enclosures. In the case of a rectangular grid, this implies that the computed bounds for images of grid elements must be enclosed in products of intervals, that is, in boxes with edges parallel to the axes. Unfortunately, this may result in considerable overestimates. In many cases it is possible to determine a collection of grid elements which is satisfactory to cover the computed bound but does not satisfy the convexity assumption, especially if working with parallelepipeds or similar sets. Using this kind of tighter enclosures, however, introduces the risk that the constructed map does not satisfy the acyclicity assumption, and thus prompts the need to improve the theoretical apparatus used to compute the homological Conley index. A first approach in this direction has been made [9] by relaxing the assumption of acyclicity in the algorithm for the computation of the homomorphisms induced in homology by combinatorial maps.

Another idea for improving the accuracy of the computations is based on the observation that in the multi-scale approach [1, §4.2] one can often choose a larger integration time  $T$  at a finer subdivision depth, and thus benefit from the dissipativity present in the system. Indeed, smaller grid elements taken as initial conditions result in smaller overestimates in the integration. Moreover, problematic initial conditions whose trajectories quickly leave the bounding box in the phase space and are subject to considerable expansion, have mostly been eliminated at the previous subdivision depth, as they did not become part of the combinatorial Morse sets. Therefore, it is natural to expect that longer integration times are feasible. However, it turns out that an isolating neighborhood constructed at a coarser subdivision depth using the coarser combinatorial map need not be an isolating neighborhood for the new combinatorial map at a finer subdivision depth, especially if the integration time is increased. This is due to the fact that the numerical method does not have the “monotonicity” property on sets: Even if  $A \subset B$ , the image of  $A$  may actually “stick out” of the image of  $B$ . Moreover, the fact that the integration time  $T$  is increased implies that talking about the “monotonicity” of the multivalued map itself makes no sense; instead, one should talk about the “monotonicity” of isolating neighborhoods. Namely, if  $N$  is an isolating neighborhood with respect to a coarse grid then one would also like it to be an isolating neighborhood with respect to the combinatorial map computed on a finer grid. Unfortunately, this need not be the case, because longer integration time may imply more deformations and larger images of the finer grid elements, which in turn may spoil the isolation at the combinatorial level. Fixing this problem requires making corrections to the isolating neighborhoods in the constructed combinatorial Morse decomposition, or taking these subtleties into consideration in another way while computing the Conley indices.

Some additional experiments show that implementing both improvements may result in considerably better performance of the method. Indeed, it was possible to deal with some ODEs in an effective way; for example, when the hyperbolicity was strong (like in the Vanderpol equations) and the dynamics was simple (like in [13], where all the chain recurrent dynamics consisted of a small number of equilibria). However, dealing with more subtle dynamical systems seems to be still beyond the computing possibilities, and thus prompts further improvements and development of more effective methods.

## Acknowledgements

MG was partially supported by FAPESP grants 2013/07460-7 and 2010/00875-9, and by CNPq grants 305860/2013-5 and 306453/2009-6, Brazil.

The work of HK was partially supported by Grant-in-Aid for Scientific Research (No. 24654022, No. 25287029), Ministry of Education, Science, Technology, Culture and Sports, Japan.

KM was supported by NSF grants NSF-DMS-0835621, 0915019, 1125174, 1248071, and contracts from AFOSR and DARPA.

TM was supported by Grant-in-Aid for JSPS Fellows No. 24-5312.

A part of the research of TM and HK was also supported by JST, CREST.

Research conducted by PP has received funding from Fundo Europeu de Desenvolvimento Regional (FEDER) through COMPETE – Programa Operacional Factores de Competitividade (POFC) and from the Portuguese national funds through Fundação para a Ciência e a Tecnologia (FCT) in the framework of the research project FCOMP-01-0124-FEDER-010645 (ref. FCT PTDC/MAT/098871/2008); from the People Programme (Marie Curie Actions) of the European Union’s Seventh Framework Programme (FP7/2007-2013) under REA grant agreement no. 622033; and from the same sources as HK.

The authors express their gratitude to the Department of Mathematics of Kyoto University for making their server available for conducting the computations described in the paper, and to the reviewers for helpful comments that contributed towards increasing the quality of the paper.

## References

- [1] Z. Arai, W. Kalies, H. Kokubu, K. Mischaikow, H. Oka, P. Pilarczyk, A database schema for the analysis of global dynamics of multiparameter systems, *SIAM J. Appl. Dyn. Syst.* 8 (2009) 757–789.
- [2] M. Berz, K. Makino, Verified Integration of ODEs and Flows Using Differential Algebraic Methods on High-Order Taylor Models, *Reliable Computing* 4 (1998) 361–369.
- [3] J. Bush, M. Gameiro, S. Harker, H. Kokubu, K. Mischaikow, I. Obayashi, P. Pilarczyk, Combinatorial-topological framework for the analysis of global dynamics, *Chaos* 22 (2012) 047508.
- [4] J. Bush, K. Mischaikow, Coarse dynamics for coarse modeling: An example from population biology, *Entropy* 16 (2014) 3379–3400.
- [5] Computer Assisted Proofs in Dynamics, Jagiellonian University, <http://capd.ii.uj.edu.pl/> (accessed on March 2, 2016).
- [6] C. Conley, *Isolated invariant sets and the Morse index*, American Mathematical Society, Providence, R.I., 1978.
- [7] COSY Infinity, Michigan State University, <http://www.cosyinfinity.com/> (accessed on March 2, 2016).
- [8] J. Grote, *High-Order Computer-Assisted Estimates of Topological Entropy*, PhD thesis, Michigan State University, 2001.
- [9] S. Harker, H. Kokubu, K. Mischaikow, P. Pilarczyk, Inducing a map on homology from a correspondence, *Proc. Amer. Math. Soc.*, 144 (2016) 1787–1801.

- [10] S. Harker, K. Mischaikow, M. Mrozek, V. Nanda, Discrete Morse Theoretic Algorithms for Computing Homology of Complexes and Maps, *Found. Comput. Math.* 14 (2014) 151–184.
- [11] J. Hoefkens, Rigorous Numerical Analysis with High-Order Taylor Models, PhD thesis, Michigan State University, 2001.
- [12] T. Kapela, P. Zgliczynski, A Lohner-type algorithm for control systems and ordinary differential inclusions, *Discrete and Continuous Dynamical Systems B* 11 (2009) 365–385.
- [13] D.H. Knipl, P. Pilarczyk, G. Röst, Rich bifurcation structure in a two-patch vaccination model, *SIAM J. Appl. Dyn. Syst.* 14 (2015) 980–1017.
- [14] E. Liz, P. Pilarczyk, Global dynamics in a stage-structured discrete-time population model with harvesting, *J. Theoret. Biol.* 297 (2012) 148–165.
- [15] R.J. Lohner, Computation of guaranteed enclosures for the solutions of ordinary initial and boundary value problems, in: J.R. Cash, I. Gladwell (Eds.), *Computational Ordinary Differential Equations*, Clarendon Press, Oxford, 1992, pp. 425–436.
- [16] E.N. Lorenz, Deterministic non-periodic flows, *J. Atmos. Sci.* 20 (1963) 130–141.
- [17] K. Makino, Rigorous Analysis of Nonlinear Motion in Particle Accelerators, PhD thesis, Michigan State University, 1998.
- [18] K. Makino, M. Berz, Taylor models and other validated functional inclusion methods, *International Journal of Pure and Applied Mathematics* 4 (2003) 379–456.
- [19] K. Makino, M. Berz, Suppression of the wrapping effect by Taylor model-based verified integrators: Long-term stabilization by preconditioning, *International Journal of Differential Equations and Applications* 10 (2005) 353–384.
- [20] K. Makino, M. Berz, Suppression of the Wrapping Effect by Taylor Model-based Verified Integrators: Long-term Stabilization by Shrink Wrapping, *International Journal of Differential Equations and Applications* 10 (2005) 385–403.
- [21] K. Mischaikow, M. Mrozek, P. Pilarczyk, Graph approach to the computation of the homology of continuous maps, *Found. Comput. Math.* 5 (2005) 199–229.
- [22] R.E. Moore, *Interval Analysis*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1966.
- [23] M. Mrozek, The Conley index on compact ANR's is of finite type, *Results in Mathematics* 18 (1990) 306–313.

- [24] M. Mrozek, P. Zgliczyński, Set arithmetic and the enclosing problem in dynamics, *Ann. Polon. Math.* 74 (2000) 237–259.
- [25] N.S. Nedialkov, K.R. Jackson, J.D. Pryce, An effective high-order interval method for validating existence and uniqueness of the solution of an IVP for an ODE, *Reliable Computing* 7 (2001) 449–465.
- [26] P. Pilarczyk, Parallelization method for a continuous property, *Found. Comput. Math.* 10 (2010) 93–114.
- [27] P. Pilarczyk, L. García, B. A. Carreras, I. Llerena, A dynamical model for plasma confinement transitions, *J. Phys. A: Math. Theor.* 45 (2012) 125502.
- [28] P. Pilarczyk, K. Stolot, Excision-preserving cubical approach to the algorithmic computation of the discrete Conley index, *Topology Appl.* 155 (2008) 1149–1162.
- [29] N. Revol, K. Makino, M. Berz, Taylor models and floating-point arithmetic: proof that arithmetic operations are validated in COSY, *J. Log. Algebr. Program* 64 (2005) 135–154.
- [30] A. Szymczak, The Conley index for discrete semidynamical systems, *Topology Appl.* 66 (1995) 215–240.
- [31] I. Walawska, D. Wilczak, An implicit algorithm for validated enclosures of the solutions to variational equations for ODEs, in review.
- [32] D. Wilczak, P. Zgliczyński,  $C^r$ -Lohner algorithm, *Schedae Informaticae* 20 (2011) 9–46.
- [33] P. Zgliczyński,  $C^1$ -Lohner algorithm, *Found. Comput. Math.* 2 (2002) 429–465.