SET ARITHMETIC AND THE ENCLOSING PROBLEM IN DYNAMICS

MARIAN MROZEK AND PIOTR ZGLICZYŃSKI

Abstract. We study the enclosing problem for discrete and continuous dynamical systems in the context of computer assisted proofs. We review and compare the existing methods and emphasise the importance of developing a suitable set arithmetic for efficient algorithms solving the enclosing problem.

1. Introduction.

In the paper we discuss the following *enclosing problem*, which in general may be formulated as follows. Given a set $S \subset \mathbb{R}^n$, a natural number d and a function $f : \mathbb{R}^n \to \mathbb{R}^n$ provide an algorithm constructing a set $\tilde{S}$ such that

(1) it may be proved that $f^d(S) \subset \tilde{S}$
(2) the difference $\tilde{S} \setminus f^d(S)$ is possibly small
(3) the algorithm is possibly fast.

To ensure rigor of numerical computations needed to fulfill postulate 1 interval arithmetic (see Sec. 5) is used in computations. Note that in general it is difficult to fulfill postulates 2 and 3 simultaneously. Hence, depending on applications, the emphasis is either on postulate 2 or 3. Also note that the natural strategy in solving the problem for $d > 1$ is to iterate a well chosen algorithm for $d = 1$. Unfortunately, this combined with interval arithmetic leads to the so called wrapping effect (see Sec. 8), which causes that the algorithms are neither fast nor effective.

The enclosing problem is of interest in itself and several classical approaches to it may be found in literature (see [3],[6],[13]). In these

Date: July 30, 2000.

1991 *Mathematics Subject Classification.* 65G40,65L70.

Key words and phrases. dynamical systems, enclosing problem, interval arithmetic, rigorous numerical analysis.

Research supported by KBN, Grant 2 P03A 029 12.

Research supported by KBN, Grant 2 P03A 029 12 and NSF–NATO grant DGE–98–04459.

papers authors propose to fight the wrapping effect by replacing Cartesian products of intervals used to approximate sets by some finer sets. Lohner [6] uses parallelepipeds and Neumaier [13] uses ellipsoids. The right choice of a countable class of sets used to approximate arbitrary sets seems to be the key strategy in developing efficient solutions to the enclosing problem. In other words, a suitable set arithmetic on top of the elementary interval arithmetic must be developed.

The classical approaches considered the case when S was a very small ball or even a point and the emphasis was on minimizing $\tilde{S}$ and not on the speed of the algorithm. Recently the enclosing problem became of vital importance in computer assisted proofs in dynamics based on topological techniques (see [4], [7], [8], [9], [11], [12], [16], [17]). In such applications the map f is usually a generator of a discrete dynamical system, a Poincaré map of a flow or just a t-translation along the trajectories of an ordinary differential equation. In order to verify the assumptions of certain topological criteria for particular dynamical behaviour like the existence of periodic solutions or chaotic invariant sets it is necessary to verify some inclusions of the form $f(S) \subset T$, where S and T are given subsets of $\mathbb{R}^n$. However, unlike the classical case the set S is usually large and often irregular in shape. Thus, in order to verify the inclusion one covers the set S with a large number of small, regular in shape sets S_i and tries to verify that $f(S_i) \subset T$ for every i. This reduces the problem to the classical case with the difference that since the set T is given and large when compared to the sets S_i, the emphasis is more on the speed of the algorithm than on minimizing the sets $\tilde{S}_i$ enclosing the sets $f(S_i)$.

The aim of this paper is to review and compare various techniques used to solve the enclosing problem in the context of their applicability to computer assisted proofs in dynamics and provide a general framework for future investigations in this area. In particular we want to emphasise the importance of developing a suitable set arithmetic for efficient algorithms solving the enclosing problem.

Special emphasis is given to the method of Lohner [6]. We reformulate his ideas in a language which is more universal and hopefully simpler to understand for people who do not work in numerical analysis. As a simple by-product we obtain a generalization of the Lohner method to discrete dynamical systems.

2. Preliminaries.

Troughout the paper we make a general assumption, that f may be approximated by a rational map, i.e. there exists a rational map

$g : \mathbb{R}^n \to \mathbb{R}^n$ and a small continuous map $w : \mathbb{R}^n \to \mathbb{R}^n$ of known bound such that in a certain domain $D \subset \mathbb{R}^n$ containing S we have the following decomposition

$$f(x) = g(x) + w(x) \text{ for } x \in D. \tag{1}$$

In some applications f may be just rational and that case is also of our interest but in most applications f is not rational.

We want to emphasize that the case when f comes from a differential equation (t-translation, Poincaré map) may be treated just as a special case of our problem. Let

$$x' = V(x), \quad x \in \mathbb{R}^n \tag{2}$$

be an ordinary differential equation in $\mathbb{R}^n$ generating a flow $\varphi : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ and let

$$\varphi_t : \mathbb{R}^n \ni x \to \varphi(x, t) \in \mathbb{R}^n$$

denote the corresponding t-translation operator for $t > 0$. Taking a small $h > 0$, a natural number n satisfying $t/n < h$ and a numerical method $\Phi : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ such that

$$\varphi(x, s) = \Phi(x, s) + \rho(x, s), s \in [0, h]$$

for a small remainder function ρ we reduce the problem of enclosing $\varphi_t(S)$ to the problem of enclosing $f^n(S)$, where $f := \varphi_{t/n} = \Phi_{t/n} + \rho_{t/n}$ fulfills the requirement (1). The case of a Poincaré map is similar in spirit though slightly more complicated, because an extra care is needed to estimate the intersection with the Poincaré section.

If X, Y, Z are sets then the function $\omega : X \times Y \to Z$ is called an operation. If $(x, y) \in \operatorname{dom} \omega$, then we say that $x\omega y$ is defined and we write $x\omega y := \omega(x, y)$. The three basic arithmetic operations $+, -, \cdot$ are examples of operations in $\mathbb{R}$.

Throughout the paper $\mathbb{R}$ and $\mathbb{Z}$ denote respectively the set of real numbers and the set of integers. We also need the extended set of real numbers

$$\bar{\mathbb{R}} := \mathbb{R} \cup \{ \infty, -\infty, \mathrm{NAN} \},$$

which consists of all real numbers, plus and minus infinity and a special symbol NAN. Here NAN stands for *not-a-number*. It is assumed that the result of any division by zero, any undefined operation involving infinity as well as any arithmetic operation involving NAN is NAN. This convention enables us to treat all four elementary arithmetic operations $+, -, \times, /$ as defined on $\bar{\mathbb{R}} \times \bar{\mathbb{R}}$.

We treat $\bar{\mathbb{R}}$ as a partially ordered set with $\mathbb{R} \cup \{ \infty, -\infty \}$ ordered in the standard way and NAN non-comparable with any other element.

3. Arithmetic operations on sets.

In the sequel we will frequently deal with sets enclosing numbers, vectors or matrices. We will use bold characters to denote such sets. We will extensively use the standard notation

$$f(\mathbf{a}) := \{f(a) \mid a \in \mathbf{a}\},$$

where $f : X \to Y$ is a map and $\mathbf{a} \subset \operatorname{dom} f \subset X$. In particular we will use the notation in case of operations. Let $\diamond : X \times Y \to Z$. For $\mathbf{a} \times \mathbf{b} \subset \operatorname{dom} \diamond \subset X \times Y$ we write

$$\mathbf{a} \diamond \mathbf{b} := \{a \diamond b \mid a \in \mathbf{a}, b \in \mathbf{b}\}$$

and call this operation *an extension of* $\diamond$ *to sets.*

The notation is convenient in formal calculations. In particular it is convenient to remember the following straightforward propositions.

Proposition 3.1. *If* $\diamond : X \times X \to X$ *is commutative, then its extension to sets is also commutative. If* $\diamond : X \times X \to X$ *is associative, then its extension to sets is also associative.*

Not every property of operations carries over automatically to its extension to sets as the following straightforward proposition shows.

Proposition 3.2. *Assume* $+ : X \times X \to X$ *and* $\cdot : Y \times X \to X$ *are operations such that for all* $y \in Y$ *and for all* $x_1, x_2 \in X$

$$y \cdot (x_1 + x_2) = y \cdot x_1 + y \cdot x_2,$$

whenever both sides make sense. Then for all $\mathbf{b} \subset Y$ *and* $\mathbf{a}_1, \mathbf{a}_2 \subset X$

$$\mathbf{b} \cdot (\mathbf{a}_1 + \mathbf{a}_2) \subset \mathbf{b} \cdot \mathbf{a}_1 + \mathbf{b} \cdot \mathbf{a}_2,$$

whenever both sides make sense. Moreover, the inclusion cannot be replaced by an equality in general.

4. Finite encoding.

We base our considerations concerning computations on the commonly accepted Turing model. There are three important assumptions in this model: computations are performed on a finite sequence of two symbols (0 and 1), there is only a finite number of rules which may be applied to transform the sequence and each rule can read or write only one symbol in the sequence at a time. Obviously, if we want to perform computations on sequences of more than two symbols but the number of symbols is still finite, then a suitable encoding of symbols with sequences of 0 and 1 reduces the computations to the Turing model. A similar relaxation is possible if one wants to let each rule read and write a finite number of symbols at a time.

To perform computations on a finite family of objects which formally are not finite sequences of symbols one has to bijectively encode the objects with such sequences, then perform the computations on the encoding and finally translate the result back to the original objects. For example if one wants to perform computations on the family of subsets of a finite collection of planes in $\mathbb{R}^3$, one can first enumerate the planes in some way and then represent a subset with a sequence which has 1 on the i-th place if the i-th plane is present in the subset and 0 otherwise.

A problem arises when we want to perform computations on a set of objects $\mathcal{O}$ which is uncountable, because then there is obviously no bijective encoding. In such a situation we usually select a countable subset $\hat{\mathcal{O}} \subset \mathcal{O}$ for which we establish a bijective encoding (we call the elements of $\hat{\mathcal{O}}$ representable objects) and we represent arbitrary objects in $\mathcal{O}$ via a surjective map $\hat{} : \mathcal{O} \to \hat{\mathcal{O}}$. A drawback of this approach is that we introduce approximations into the computations (an object $o \in \mathcal{O}$ is approximated by its representation $\hat{o} \in \hat{\mathcal{O}}$). Since the approximations may take place a large number of times in course of the computation, not only is the final result not exact but usually we do not know how bad the approximation is. Nevertheless we will see in the sequel that sometimes the approach provides exact bounds instead of exact solutions.

As an example consider the common solution used in hardware to perform computations on real numbers. One first chooses a finite subset $\hat{\mathbb{R}} \subset \bar{\mathbb{R}}$ such that

$$0, -\infty, +\infty, \mathrm{NAN} \in \hat{\mathbb{R}}, \tag{3}$$

$$x \in \hat{\mathbb{R}} \ \Rightarrow\ -x \in \hat{\mathbb{R}} \tag{4}$$

and treats it as the set of symbols. Usually $\hat{\mathbb{R}}$ is taken as a set of certain binary fractions to facilitate further encoding into sequences of 0 an 1. Then one takes a map $\langle\cdot\rangle : \bar{\mathbb{R}} \to \hat{\mathbb{R}}$ such that

$$x \in \hat{\mathbb{R}} \ \Rightarrow\ x = \langle x\rangle, \tag{5}$$

$$x \leq y \ \Rightarrow\ \langle x\rangle \leq \langle y\rangle. \tag{6}$$

Such a map is obviously a surjection. Any real number x which is not in $\hat{\mathbb{R}}$ is replaced in the course of computations by $\langle x\rangle$. The process is known as *rounding*. We call the number $\langle x\rangle$ the *representation* of x and the pair $(\hat{\mathbb{R}}, \langle\cdot\rangle)$ the *representation* of $\bar{\mathbb{R}}$. One can easily verify that for any $x \in \bar{\mathbb{R}} \setminus \{\mathrm{NAN}\}$

$$\downarrow(x) \leq \langle x\rangle \leq \uparrow(x),$$

where

$$\downarrow\!(x) := \max \{\, y \in \hat{\mathbb{R}} \mid y \leq x \,\}$$
$$\uparrow\!(x) := \min \{\, y \in \hat{\mathbb{R}} \mid y \geq x \,\}$$

are the *lower* and *upper representations* of x.

As we already mentioned, the price one pays for such an approach is that the computations are approximate and there is no way (other than statistical analysis) to tell how bad the approximations is.

Even if the input of the computation consists of representable numbers, the result of an arithmetic operation on representable numbers may not be a representable number and has to be replaced by its representation. This introduces deviations from the exact results which keep propagating in course of computations. Though usually there is no way to obtain exact results, it is possible to get exact lower and upper bounds. This is achieved by means of the *interval arithmetic*.

5. Interval arithmetic.

The concept of interval arithmetic goes back to 50's and has several fathers. Probably the earliest paper on this subject is the work of M. Warmus [14]. Several other early papers are listed in the monograph by R. Moore [10].

The idea of the interval arithmetic consists in replacing every representation $\langle x \rangle$ on input by the interval $[\downarrow\!(x), \uparrow\!(x)]$ and perform calculations on intervals in such a way that at every stage of calculations the interval contains the exact result.

To be more precise we define the set of *intervals over $A \subset \bar{\mathbb{R}}$* by

$$\mathbb{I}(A) := \{[a^-, a^+] \mid a^-, a^+ \in A,\ a^- \leq a^+\}.$$

In particular we will write $\mathbb{I}$, $\bar{\mathbb{I}}$, $\hat{\mathbb{I}}$ respectively for $\mathbb{I}(\mathbb{R})$, $\mathbb{I}(\bar{\mathbb{R}})$ and $\mathbb{I}(\hat{\mathbb{R}})$. The elements of $\hat{\mathbb{I}}$ will be called *representable intervals*. Note that the only interval in $\bar{\mathbb{I}}$ and $\hat{\mathbb{I}}$ containing NAN is $[\mathrm{NAN}, \mathrm{NAN}]$.

The set of real numbers may be viewed as a subset of $\mathbb{I}(\bar{\mathbb{R}})$ via the embedding

$$\bar{\mathbb{R}} \ni x \to [x, x] \in \mathbb{I}(\bar{\mathbb{R}}).$$

The *size* of an interval $\mathbf{a} = [a^-, a^+] \in \mathbb{I}$ is defined as the difference $a^+ - a^-$. We also define the *mid point* of $\mathbf{a}$ by

$$\mathrm{mid}(\mathbf{a}) := \langle \tfrac{a^- + a^+}{2} \rangle.$$

The reader should notice the by the above definition the mid point of an interval is always a representable number.

To perform calculations on intervals we use representable intervals. To encode representable intervals we just store two endpoints of the interval. If $\mathbf{a} \in \mathbb{I}$ is an arbitrary interval then we represent $\mathbf{a}$ as the smallest representable interval containing $\mathbf{a}$. Such an interval will be called the representation of $\mathbf{a}$ and denoted by $\updownarrow(\mathbf{a})$. One can easily verify that for any $a^-, a^+ \in \mathbb{R}$, $a^- \leq a^+$

$$\updownarrow([a^-, a^+]) = [\downarrow(a^-), \uparrow(a^+)].$$

Obviously such an encoding is only surjective.

As in the case of the arithmetic of representable numbers, arithmetic operations on intervals cannot be performed exactly. However in the case of intervals there is an easy way to guarantee that the computed result encloses the exact result. To achieve this, given an arithmetic operation $\diamond$, one defines the operation $\hat{\diamond}$ by

$$\mathbf{a}\hat{\diamond}\mathbf{b} := \begin{cases} \updownarrow(\mathbf{a} \diamond \mathbf{b}) & \text{if NAN} \notin \mathbf{a} \diamond \mathbf{b} \\ [\text{NAN}, \text{NAN}] & \text{otherwise} . \end{cases}$$

for $\mathbf{a}, \mathbf{b} \in \bar{\mathbb{I}}$. An important remark is that the right hand side of the above formula is useless from the computational point of view, since we cannot compute $\mathbf{a} \diamond \mathbf{b}$ (comp. last paragraph of Section 4.). Fortunately, for each concrete arithmetic operation $\diamond \in \{+, -, \times, /\}$ one can easily find a simple algorithm transforming the encoding (endpoints) of $\mathbf{a}, \mathbf{b}$ to the encoding of the result, which makes the computation of $\mathbf{a}\hat{\diamond}\mathbf{b}$ possible.

Sometimes it is convenient to decompose an interval as a sum of a representable number and a representable interval containing zero. For this end we introduce the notation

$$\hat{\mathbb{I}}_0 := \{\, \mathbf{a} \in \hat{\mathbb{I}} \mid 0 \in \mathbf{a} \,\}.$$

and define the *decomposition* of an interval $\mathbf{a} \in \mathbb{I}$ as the pair

$$(\text{mid}(\mathbf{a}), \mathbf{a} \hat{-} \text{mid}(\mathbf{a})) \in \hat{\mathbb{R}} \times \hat{\mathbb{I}}_0.$$

Note the following straightforward proposition.

Proposition 5.1. *If $(c, \mathbf{c}_0) \in \hat{\mathbb{R}} \times \hat{\mathbb{I}}_0$ is the decomposition of $\mathbf{c}$ then $\mathbf{c} \subset c + \mathbf{c}_0$.*

□

The elements of $\mathbb{I}^n$ will be called *n-dimensional interval vectors* and elements of $\mathbb{I}^{n \times n}$ - *n-dimensional interval matrices.*

The encoding of interval vectors and matrices is componentwise. We just represent interval vectors and interval matrices as sequences of representations of their entries.

We define the *size* of an interval vector or an interval matrix as the maximum of sizes of the components, the mid point of an interval vector or an interval matrix as the interval vector or matrix consisting of the mid points of the components. Similarly we define the decomposition of an interval vector or matrix.

Apart from interval vectors and matrices we also define interval sets. An *interval set* is a Cartesian product of intervals. An *interval representable set* is a Cartesian product of representable intervals. There is a one-to-one correspondence between interval vectors and interval sets given by

$$(\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n) \leftrightarrow \mathbf{a}_1 \times \mathbf{a}_2 \times \ldots \times \mathbf{a}_n$$

for $\mathbf{a}_1, \mathbf{a}_2, \ldots \mathbf{a}_n \in \mathbb{I}$. Since we can treat matrices as $n \times m$-vectors, a similar correspondence applies to matrices. In the sequel we will freely pass from one interpretation to the other.

If $\mathbf{s} \subset \mathbb{R}^n$ is a set, then $[\mathbf{s}]$ will stand for the minimal representable interval set which contains $\mathbf{a}$.

We define the sum of interval vectors or interval matrices, the product of an interval and an interval vector or an interval matrix, the product of interval matrices, the scalar product of interval vectors and the product of an interval matrix by an interval vector exactly as in the case of real vectors and matrices.

6. Rational expressions and rational functions.

Practically speaking the numeric evaluation of a function at a given argument is somehow reduced to calculations with the four elementary arithmetic operations $+, -, \cdot, /$. Therefore what is actually computed is a rational function. Since we will evaluate rational functions with interval arguments where the arithmetic operations are neither associative nor distributive, we need a slightly modified definition of a rational function.

We first recall the formalism of arithmetic expressions. For this purpose we fix a countably infinite set of symbols $\mathcal{V} = \{ X_1, X_2, \ldots \}$, called *variables*, and define a *word* to be an arbitrary finite sequence of symbols in $\mathcal{V} \cup \hat{\mathbb{R}} \cup \{ \ (\ , \ ) \ , +, -, \cdot, / \}$. If ω is a word then $|\omega|$ will stand for the length of ω, i.e. the number of elements in ω. The set of *(arithmetic) expressions* $\mathcal{W}$ is defined as the smallest set of words satisfying the following two properties

$$x \in \mathcal{V} \cup \hat{\mathbb{R}} \ \Rightarrow \ x \in \mathcal{W}$$

$$\omega_1, \omega_2 \in \mathcal{W} \ \Rightarrow \ (\omega_1 + \omega_2), (\omega_1 - \omega_2), (\omega_1 \cdot \omega_2), (\omega_1 / \omega_2) \in \mathcal{W}$$

The following proposition is an easy exercise.

Proposition 6.1. *If $\omega \in \mathcal{W}$ then exactly one of the following possibilities holds*

(i) $\omega = X_i$ *for some* $X_i \in \mathcal{V}$

(ii) $\omega = s$ *for some* $s \in \hat{\mathbb{R}}$

(iii) *there exist* $\omega_1, \omega_2 \in \mathcal{W}$ *such that* $\omega = (\omega_1 + \omega_2)$ *or* $\omega = (\omega_1 - \omega_2)$ *or* $\omega = (\omega_1 \cdot \omega_2)$ *or* $\omega = (\omega_1 / \omega_2)$.

For instance $((X_1 + X_2) + X_3)$ is an arithmetic expression but $+++$, $+(X_1 X_2)$, $X_1 + X_2 + X_3$, $X_1 + (X_2 + X_3)$ are not.

We define recursively $\mathrm{Var}(\omega)$, the set of variables in ω by

$$\mathrm{Var}(\omega) := \begin{cases} \emptyset & \text{if } \omega = s, \text{ where } s \in \hat{\mathbb{R}} \\ \{X_i\} & \text{if } \omega = X_i, \text{ where } X_i \in \mathcal{V} \\ \mathrm{Var}(\omega_1) \cup \mathrm{Var}(\omega_2) & \text{if } \omega = (\omega_1 \diamond \omega_2) \end{cases} \tag{7}$$

where $\diamond \in \{ +, -, \cdot, / \}$.

Let ω be an expression. Let $n \in \mathbb{N}$ be large enough so that $\mathrm{Var}(\omega) \subset \{X_1, \dots X_n\}$. Let $x_1, \dots, x_n$ be real numbers. We define recursively $\omega(x_1, \dots x_n)$, the evaluation of ω at $(x_1, \dots, x_n) \in \mathbb{R}^n$ by

$$\omega(x_1, \dots x_n) := \begin{cases} s & \text{if } \omega = s, \text{ where } s \in \hat{\mathbb{R}} \\ x_i & \text{if } \omega = X_i, \text{ where } X_i \in \mathcal{V} \\ \omega_1(x_1, \dots x_n) \diamond \omega_2(x_1, \dots x_n) & \text{if } \omega = (\omega_1 \diamond \omega_2) \end{cases} \tag{8}$$

where $\diamond \in \{ +, -, \cdot, / \}$.

Similarly we define recursively $\langle \omega \rangle(\mathbf{x}_1, \dots, \mathbf{x}_n)$, the interval evaluation of ω at $(\mathbf{x}_1, \dots, \mathbf{x}_n) \in \hat{\mathbb{I}}^n$ by

$$\langle \omega \rangle(\mathbf{x}_1, \dots \mathbf{x}_n) := \begin{cases} s & \text{if } \omega = s, \text{ where } s \in \hat{\mathbb{R}} \\ \mathbf{x}_i & \text{if } \omega = X_i, \text{ where } X_i \in \mathcal{V} \\ \langle \omega_1 \rangle(\mathbf{x}_1, \dots \mathbf{x}_n) \langle \diamond \rangle \langle \omega_2 \rangle(\mathbf{x}_1, \dots \mathbf{x}_n) & \text{if } \omega = (\omega_1 \diamond \omega_2). \end{cases} \tag{9}$$

Let $\omega_1 := ((X_1 + X_2) + X_3)$ and $\omega_2 := (X_1 + (X_2 + X_3))$ be arithmetic expressions. Obviously $\omega_1 \neq \omega_2$ (the sequences of symbols are different) but for any $(x_1, x_2, x_3) \in \mathbb{R}^3$

$$\omega_1(x_1, x_2, x_3) = \omega_2(x_1, x_2, x_3),$$

because of the associativity of the addition operation in $\mathbb{R}$. However, the interval evaluations $\langle \omega_1 \rangle(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$ and $\langle \omega_2 \rangle(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$ need not be the same, because $\langle + \rangle$ is not associative in general. This explains why we set up the definition of the arithmetic expression in such a way that there is no ambiguity in the order of evaluation.

Assume $\omega = (\omega_i)_{i=1,\dots,m}$ is a vector of arithmetic expressions such that $\mathrm{Var}(\omega_i) \subset \{X_1, \dots X_n\}$. There is then an associated function $f^\omega = (f_1^\omega, \dots f_m^\omega) : \mathbb{R}^n \to \mathbb{R}^m$ given by

$$f_i^\omega(x_1, \dots x_n) = \omega_i(x_1, \dots x_n) \text{ for } i = 1, 2, \dots m$$

and an associated interval function $\langle f\rangle^\omega = (\langle f\rangle_1^\omega, \dots \langle f\rangle_m^\omega) : \hat{\mathbb{I}}^n \to \hat{\mathbb{I}}^m$ given by

$$\langle f\rangle_i^\omega(\mathbf{x}_1, \dots \mathbf{x}_n) = \langle\omega_i\rangle(\mathbf{x}_1, \dots \mathbf{x}_n) \text{ for } i = 1, 2, \dots m.$$

We say that the function $g : \mathbb{R}^n \to \mathbb{R}^m$ is a rational function if $g = f^\omega$ for a vector ω of arithmetic expressions. The function $\langle f\rangle^\omega$ is then called an interval extension of g. Note that for reasons explained earlier there may be two different arithmetic expressions ω_1 and ω_2 such that $f^{\omega_1} = f^{\omega_2}$. Consequently the interval extension of g need not be unique. Nevertheless we will write $\langle g\rangle$ meaning any possible interval extension of g.

The following theorem, which may be proved easily by induction, is crucial for interval evaluations of rational functions.

Theorem 6.2. *Assume $f : \mathbb{R}^n \to \mathbb{R}^m$ is a rational function. Then for any $\mathbf{x}_1, \dots \mathbf{x}_n \in \hat{\mathbb{I}}^n$ we have*

$$f(\mathbf{x}_1, \dots \mathbf{x}_n) \subset \langle f\rangle(\mathbf{x}_1, \dots \mathbf{x}_n). \tag{10}$$

□

Here there is one important comment in order. The left-hand side of (10) more formally should be written as $f(\mathbf{x}_1 \times \cdots \times \mathbf{x}_n)$ and denotes the image of the set $\mathbf{x}_1 \times \cdots \times \mathbf{x}_n$ by f. Therefore it is a pure notation and does not contribute towards a method of its computation. The right-hand side may be computed easily from its definition thus providing a method of enclosing the left-hand side.

We will say that an operation $\diamond : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^p$ is rational if it is a rational function. Note that matrix product provides a simple example of a rational operation.

We can rewrite the last theorem in terms of an operation as in the following corollary.

Corollary 6.3. *If $\diamond : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^p$ is a rational operation and $\mathbf{v} \in \mathbb{R}^m, \mathbf{w} \in \mathbb{R}^n$ then*

$$\mathbf{v} \diamond \mathbf{w} \subset \mathbf{v}\langle\diamond\rangle\mathbf{w}.$$

□

7. Enclosures via simple representations of sets.

The first question we have to ask when discussing the problem of estimation of $f^d(S)$ is how we should represent sets. Actually the important question is what the family of representable sets should be, because once the family $\hat{\mathcal{S}}$ of representable sets is fixed, it is very natural to define $\hat{S}$, the representation of S as the smallest representable set containing S. Since then $f^d(S) \subset f^d(\hat{S})$, the next question we have to address is how to construct an estimate of $f^d(S)$ when S is a representable set. Obviously the two questions are strongly interconnected.

The choice of the class of representable sets depends on several factors. The first factor is the dimension of the space. In dimension one intervals serve well the purpose of representing compact, connected sets. However in higher dimensions we should also consider what our sets look like. If they are as simple as cuboids, balls or parallelepipeds the strategy will be simpler then in the case when they are relatively big and irregular in shape.

There are a few classes of simple sets which can be easily parametrized by a few numbers or vectors, which enables easy encoding of such sets, when the parameters are representable.

7.1. **Bounds via balls.** Assume a norm $||.||$ in $\mathbb{R}^n$ is given. Recall that the ball centered at $x \in \mathbb{R}^n$ of radius $r > 0$ is the set $B(x,r) := \{y \in \mathbb{R}^n \mid ||x-y|| \leq r\}$. We say that the ball $B(x,r)$ is representable if both x and r are representable. We encode such a ball as the pair $(x,r) \in \hat{\mathbb{R}}^2$.

The simplest strategy for finding an enclosure for $f(S)$ requires that f is Lipschitz with Lipschitz constant $L \in \hat{\mathbb{R}}$, $S = B(x,r)$ for some representable vector $x \in \mathbb{R}^n$ and some representable number $r \in \mathbb{R}^+$. It also requires that an algorithm is known which for a given representable $\epsilon > 0$ constructs a vector $y \in B(f(x),\epsilon)$. One then easily verifies the following proposition.

Proposition 7.1.

$$f(B(x,r)) \subset B(y, L \hat{\cdot} r \hat{+} \epsilon).$$

□

7.2. **Logarithmic norms.** The applicability of the last proposition depends crucially on the knowledge of a possibly small Lipschitz constant. In case the map $f = \varphi_t$ is the t-translation of the flow generated by the diffrential equation (2), a Lipschitz constant for f may be obtained from the classical differential inequalities theory as e^{Kt}, where

$$K := \sup \{ \, ||d_x V|| \mid x \in \varphi(S,[0,t]) \, \}$$

and $||.||$ denotes a norm in $\mathbb{R}^n$. Unfortunately this is usually a very bad Lipschitz constant. In particular it is always greater than one, even if the flow is contracting.

Much better Lipschitz constants for f may be obtained by means of the so called *logarithmic norm.* If Q is a square matrix, then its logarithmic norm $\mu(Q)$ is defined by

$$\mu(Q) := \lim_{h\to 0, h>0} \frac{||I + hQ|| - 1}{h}.$$

It turns out that the differential inequalities theory has a counterpart in which the norm is replaced by the function μ (for a very detailed case concerning non-autonomous differential equations see Theorem 10.6 in [5]). For our purposes a very special case of that theorem may be formulated as follows.

Theorem 7.2. *If*

$$K_\mu := \sup \{ \mu(d_x V) \mid x \in \varphi(S, [0, t]) \}$$

then $e^{K_\mu t}$ *is a Lipschitz constant for* $f = \varphi_t$.

7.3. **Bounds via interval sets.** If f is a rational function and S is a representable interval set, one can use the interval evaluation of f as an enclosure of $f(S)$:

Proposition 7.3. *Assume* f *is a rational function and* $S = \mathbf{x} \in \hat{\mathbb{I}}^n$ *is an interval set. Then* $f(\mathbf{x}) \subset \langle f\rangle(\mathbf{x})$.

□

If f is not rational, we still may be able to approximate f by a rational function g and a small function w from the decomposition (1). We assume that we know an interval set $\mathbf{w} \in \hat{\mathbb{I}}^n$ such that $w(x) \in \mathbf{w}$ for $x \in D$. We do not assume how big the bound $\mathbf{w}$ is, but the bound for $f(S)$ we are looking for will crucially depend on the size of $\mathbf{w}$.

Proposition 7.4.

$$f(\mathbf{x}) \subset \langle g\rangle(\mathbf{x}) \hat{+} \mathbf{w}.$$

□

7.4. **Set valued mean value theorem.** If f is C^1, then we define the *Jacobian matrix* of f at $\mathbf{x}$ by

$$J_{\mathbf{x}} f := ([\frac{\partial f_i}{\partial x_j}(\mathbf{x})])_{i,j=1,n}$$

We will first prove the set valued mean value theorem.

Theorem 7.5. *For any* $\mathbf{h} \in \mathbb{I}_0^n$ *we have*

$$f(x+\mathbf{h}) \subset f(x) + J_{x+\mathbf{h}} f \cdot \mathbf{h}.$$

Proof: Take $h \in \mathbf{h}$. Using the standard mean value theorem for the components f_i of f we get for some $\theta_i \in [0,1]$ and $i = 1, 2, \dots n$

$$f_i(x+h) = f_i(x) + \sum_{j=1}^{n} \frac{\partial f_i}{\partial x_j}(x + \theta_i h) h_j.$$

Thus it follows that

$$f(x+h) \in f(x) + \left(\sum_{j=1}^{n} [\frac{\partial f_i}{\partial x_j}(x+\mathbf{h})] h_j \right)_{i=1,n} \subset$$
$$f(x) + J_{x+\mathbf{h}} \cdot h \subset f(x) + J_{x+\mathbf{h}} \cdot \mathbf{h}.$$

□

7.5. **Improved bounds via interval sets.** We again assume that S is an interval set, i.e. $S = \mathbf{x}$ for a certain interval vector $\mathbf{x}$. We rewrite $\mathbf{x}$ as the sum $x + \mathbf{r}$, where $(x, \mathbf{r})$ is the decomposition of $\mathbf{x}$ and take the pair $(x, \mathbf{r})$ as the representation of S.

Lemma 7.6. *Assume* $(x_1, \mathbf{x}_1^0)$ *is a decomposition of* $\langle g \rangle(x)$ *and put*

$$\mathbf{r}_1 := J_{\mathbf{x}} g \hat{\cdot} \mathbf{r} \hat{+} \mathbf{w} \hat{+} \mathbf{x}_1^0,$$

then $f(x+\mathbf{r}) \subset x_1 + \mathbf{r}_1$.

Proof: Let $u \in x + \mathbf{r}$. Then $u = x + r$ for some $r \in \mathbf{r}$. Thus we have

$$f(u) = g(u) + w(u) \in g(x) + J_{\mathbf{x}} g \cdot r + w(u) \subset$$
$$\langle g \rangle(x) + J_{\mathbf{x}} g \cdot r + \mathbf{w} \subset x_1 + J_{\mathbf{x}} g \cdot r + \mathbf{w} + \mathbf{x}_1^0 \subset$$
$$x_1 + J_{\mathbf{x}} g \hat{\cdot} \mathbf{r} \hat{+} \mathbf{w} \hat{+} \mathbf{x}_1^0 = x_1 + \mathbf{r}_1.$$

□

The classes of representable sets we considered in this section as well as classes we will consider in the next section are gathered in the following table. The table contains the set, their geometric form and their encoding.

Basic sets			
Description	Geometric Form	Encoding	Comments
a ball	$B(c, r)$	(c, r)	$c \in \hat{\mathbb{R}}^n, r \in \hat{\mathbb{R}}$, $r > 0$ the geometric form depends on norm used
an interval set	$c + \mathbf{r}$	$(c, \mathbf{r})$	$c \in \hat{\mathbb{R}}^n$, $\mathbf{r} \in \hat{\mathbb{I}}_0^n$
a parallelepiped	$c + \mathbf{B} \cdot \mathbf{r}$	$(c, \mathbf{B}, \mathbf{r})$	$c \in \hat{\mathbb{R}}^n, \mathbf{B} \in \hat{\mathbb{I}}^{n \times n}$, $\mathbf{r} \in \hat{\mathbb{I}}_0^n$
a cuboid	$c + \mathbf{Q} \cdot \mathbf{r}$	$(c, \mathbf{Q}, \mathbf{r})$	$c \in \hat{\mathbb{R}}^n, \mathbf{Q} \in \hat{\mathbb{I}}^{n \times n}$, $\mathbf{r} \in \hat{\mathbb{I}}_0^n$ and $\mathbf{Q}$ contains an orthogonal matrix
a doubleton	$c + C \cdot \mathbf{r}_0 + \mathbf{s}$	$(c, C, \mathbf{r}_0)$ followed by the encoding of $\mathbf{s}$	$c \in \hat{\mathbb{R}}^n, C \in \hat{\mathbb{R}}^{n \times n}$, $\mathbf{r}_0 \in \hat{\mathbb{I}}_0^n$ $\mathbf{s}$ - another basic set
an elliptic set	$c + L \cdot B(0, r)$	(c, L, r)	$c \in \hat{\mathbb{R}}^n, L \in \hat{\mathbb{R}}^{n \times n}$ $r \in \hat{\mathbb{R}}$

8. Shape sensitive enclosures.

At first it may seem that the enclosures presented in the previous chapter are perfect if our set S is an interval set or a ball. Actually the choices are usually satisfactory if we need to estimate only the first iterate of f. However if we try to iterate this process, we often encounter problems. The reason is that the image of an interval or a ball need not resemble the shape of the argument. Thus enclosing the image in an interval or a ball we may introduce a strong overestimation which will keep propagating under iteration in an exponential way. This process is known as the *wrapping effect*.

Lohner (see [6]) proposed a solution to this problem in the case f is a translation map along trajectories of a differential equation. We present a version of his method adapted to the more general discrete case in the Lemmas 8.3, 8.4 and 8.5.

A good way to understand the key idea of Lohner approach is to consider first the case when f is a linear map. Even in this simple case the wrapping effect will take place, because linear maps neither transform interval sets to interval sets nor balls to balls. To avoid wrapping in this case we need a class of representable sets which is closed under linear maps. A possible choice are parallelepipeds in $\mathbb{R}^n$. A parallelepiped in $\mathbb{R}^n$ may be written as $\mathbf{p} := c + B \cdot \mathbf{r}$, where $c \in \mathbb{R}^n$, $B \in \mathbb{R}^{n\times n}$ and $\mathbf{r} \in \mathbb{I}^n$. We will say that $\mathbf{p}$ is representable if c, B and $\mathbf{r}$ are representable. The encoding of a representable parallelepiped is obviously $(c, B, \mathbf{r})$.

Using parallelepipeds with linear maps we would compute the images exactly if not the rounding errors. But we can expect a very good upper bound.

8.1. **Rearrangement.** One of the important problems concerning computations with a given class of representable sets is enclosing sets which appear in course of computations by a representable set. We would like to consider here a relatively simple problem of enclosing with a parallelepiped the sum of a parallelepiped and a small interval set. Thus a parallelepiped $c + P \cdot \mathbf{r}$ and an interval $\mathbf{v}$ are given and we want to find a possibly small parallelepiped $c + P_1 \cdot \mathbf{r}_1$ such that

$$c + P \cdot \mathbf{r} + \mathbf{v} \subset c + P_1 \cdot \mathbf{r}_1.$$

We will consider a slightly more general problem, where P and P_1 are replaced by interval matrices $\mathbf{P}$ and $\mathbf{P}_1$. Obviously it is sufficient to find a matrix $\mathbf{P}_1$ and an interval vector $\mathbf{r}_1$ such that

$$\mathbf{P} \cdot \mathbf{r} + \mathbf{v} \subset \mathbf{P}_1 \cdot \mathbf{r}_1.$$

The simplest choice is to fix arbitrarily $\mathbf{P}_1$ and search only for $\mathbf{r}_1$. To do so we introduce the concept of a weak inverse of an interval matrix. We say that an interval matrix $\mathbf{P}^* \in \mathbb{I}^{n\times n}$ is a *weak inverse* of an interval matrix $\mathbf{P} \in \mathbb{I}^{n\times n}$ if there exists a non-singular matrix $P \in \mathbf{P}$ such that $P^{-1} \in \mathbf{P}^*$. Obviously the weak inverse is not unique. Note that to find a weak inverse of a given interval matrix $\mathbf{P}$ it is enough to find an enclosure of the inverse matrix of a non-singular selector of $\mathbf{P}$. In particular, if $\mathbf{P}$ contains an orthogonal matrix then $\mathbf{P}^T$ is its weak inverse.

For $\mathbf{P}, \mathbf{P}_1 \in \mathbb{I}^{n\times n}$ and $\mathbf{r}, \mathbf{v} \in \mathbb{I}^n$ put

$$\operatorname{Rearr}_1(\mathbf{P}, \mathbf{P}_1, \mathbf{r}, \mathbf{v}) := (\mathbf{P}_1^* \mathbin{\hat{\cdot}} \mathbf{P}) \mathbin{\hat{\cdot}} \mathbf{r} \mathbin{\hat{+}} \mathbf{P}_1^* \mathbin{\hat{\cdot}} \mathbf{v} \tag{11}$$

We have the following lemma.

Lemma 8.1. *Assume $\mathbf{P}_1$ is an interval matrix and $\mathbf{P}_1^*$ is its week inverse. Then for $\mathbf{r}_1 := \operatorname{Rearr}_1(\mathbf{P}, \mathbf{P}_1, \mathbf{r}, \mathbf{v})$ we have the inclusion*

$$\mathbf{P} \cdot \mathbf{r} + \mathbf{v} \subset \mathbf{P}_1 \cdot \mathbf{r}_1.$$

Proof: We have

$$\mathbf{P}\cdot\mathbf{r}+\mathbf{v}\subset\mathbf{P}_1\cdot\mathbf{P}_1^*\cdot(\mathbf{P}\cdot\mathbf{r}+\mathbf{v})\subset$$
$$\mathbf{P}_1\cdot(\mathbf{P}_1^*\cdot\mathbf{P}\cdot\mathbf{r}+\mathbf{P}_1^*\cdot\mathbf{v})\subset\mathbf{P}_1\cdot(\mathbf{P}_1^*\hat{\cdot}\mathbf{P}\hat{\cdot}\mathbf{r}\hat{+}\mathbf{P}_1^*\hat{\cdot}\mathbf{v})\subset$$
$$\mathbf{P}_1\cdot\mathbf{r}_1.$$

□

Obviously in practice we try to choose $\mathbf{P}_1$ in such a way that $\mathbf{P}_1\cdot\mathbf{r}_1$ is as small as possible. Note that we put explicit parentheses around the matrix $\mathbf{P}_1^*\hat{\cdot}\mathbf{P}$ in the formula (11) requiring that the computation of the product precedes the multiplication by $\mathbf{r}$. The reason is as follows. Matrices in the product $\mathbf{P}_1^*\hat{\cdot}\mathbf{P}$ in general are of relatively small size compared to the size of $\mathbf{r}$, therefore the potential wrapping effect in this computation is also relatively small. The biggest wrapping will be introduced in the multiplication by $\mathbf{r}$. Hence it is worth to put maximum effort to minimize this wrapping. There is no wrapping in the multiplication by a matrix, when the off diagonal elements of the matrix are non-negative (see Anguelov, [1],[2]), in particular when the matrix is the identity matrix. Thus one can expect small wrapping when the matrix is close to identity. When $\mathbf{P}_1$ is a selector of $\mathbf{P}$, the matrix $\mathbf{P}_1^*\hat{\cdot}\mathbf{P}$ contains identity. Thus, since the size of this matrix is usually small, it is close to identity and one can expect small wrapping.

A well known fact in numerical analysis is that inverting a matrix numerically is not a good procedure if the goal is to solve a linear equation. It turns out that finding numerically an inverse of $\mathbf{P}_1$ may be replaced in the rearrengment problem by solving a suitable interval linear equation. To explain this assume that $\mathbf{P}\in\mathbb{I}^{n\times n}$, $P_1\in\mathbb{R}^{n\times n}$ is an invertible selector of $\mathbf{P}$, $\mathbf{R}:=\mathbf{P}\hat{-}P_1$ and $\mathbf{r},\mathbf{v}\in\mathbb{I}^n$. Let $\mathbf{t}$ be an interval vector obtained by applying the Gauss elimination to the linear interval equation

$$P_1\hat{\cdot}\mathbf{x}=\mathbf{R}\hat{\cdot}\mathbf{r}\hat{+}\mathbf{v}.$$

Note that $\mathbf{t}$ in particular encloses the set of solutions of all linear equations

$$P_1\cdot x=b,\ \text{where } x\in\mathbb{R}^n \text{ and } b\in\mathbf{R}\hat{\cdot}\mathbf{r}\hat{+}\mathbf{v}.$$

Put

$$\operatorname{Rearr}_2(\mathbf{P},P_1,\mathbf{r},\mathbf{v}):=\mathbf{r}\hat{+}\mathbf{t}, \tag{12}$$

We have the following lemma.

Lemma 8.2. *Assume P_1 is an invertible selector of $\mathbf{P}$. Then for $\mathbf{r}_1:=\operatorname{Rearr}_2(P,\mathbf{P}_1,\mathbf{r},\mathbf{v})$ we have the inclusion*

$$\mathbf{P}\cdot\mathbf{r}+\mathbf{v}\subset P_1\cdot\mathbf{r}_1.$$

Proof: Take $u \in \mathbf{P}\cdot\mathbf{r}+\mathbf{v}$. Then $u = P\cdot r + v$ for some $P \in \mathbf{P}$, $r \in \mathbf{r}$ and $v \in \mathbf{v}$. Let $R := P - P_1$. We have

$$u = P_1 \cdot P_1^{-1} \cdot (P \cdot r + v) = P_1 \cdot \left(P_1^{-1} \cdot (P_1 + R) \cdot r + P_1^{-1} \cdot v\right) =$$
$$P_1 \cdot (r + P_1^{-1} \cdot R \cdot r + P_1^{-1} \cdot v) =$$
$$P_1 \cdot \left(r + P_1^{-1} \cdot (R \cdot r + v)\right) \in P_1 \cdot (\mathbf{r} + \mathbf{t}) \subset$$
$$P_1 \cdot \mathbf{r}_1.$$

□

8.2. **Parallelepipeds.** We now turn to the Lohner method of parallelepipeds in enclosing $f(S)$.

We recall that we consider a function $f : \mathbb{R}^n \to \mathbb{R}^n$ such that for a certain rational map $g : \mathbb{R}^n \to \mathbb{R}^n$ and a small continuous map $w : \mathbb{R}^n \to \mathbb{R}^n$ the decomposition (1) holds.

Let $\mathbf{x} := x + \mathbf{B}\cdot\mathbf{r} \subset D$ be a fuzzy parallelepiped.

Lemma 8.3. *Assume $\mathbf{B}_1$ is an interval matrix, $\mathbf{B}_1^*$ is a weak inverse of $\mathbf{B}_1$, $(x_1, \mathbf{x}_1^0)$ is a decomposition of $\langle g\rangle(x)$ and*

$$\mathbf{r}_1 := \mathrm{Rearr}_1(J_{\mathbf{X}}g \,\hat{\cdot}\, B, \mathbf{B}_1, \mathbf{r}, \mathbf{w} \,\hat{+}\, \mathbf{x}_1^0)$$

then $f(x + \mathbf{B}\cdot\mathbf{r}) \subset x_1 + \mathbf{B}_1 \cdot \mathbf{r}_1$.

Proof: Let $u \in x + \mathbf{B}\cdot\mathbf{r}$. Then $u = x + B\cdot r$ for some $B \in \mathbf{B}$ and $r \in \mathbf{r}$. Thus we have

$$f(u) = g(u) + w(u) \in g(x) + J_{\mathbf{X}}g \cdot B \cdot r + w(u) \subset$$
$$\langle g\rangle(x) + J_{\mathbf{X}}g \cdot B \cdot r + \mathbf{w} \subset$$
$$x_1 + J_{\mathbf{X}}g \cdot B \cdot r + \mathbf{w} + \mathbf{x}_1^0 \subset$$
$$x_1 + (J_{\mathbf{X}}g \,\hat{\cdot}\, B) \cdot r + (\mathbf{w} \,\hat{+}\, \mathbf{x}_1^0)$$

and the conclusion follows from Lemma 8.1. □

Note that an analogous lemma may be formulated in terms of Rearr_2.

The above lemma guarantees the correctness of the enclosure for any choice of $\mathbf{B}_1$ but obviously we still try to choose $\mathbf{B}_1$ in such a way that the the resulting enclosure is possibly small. Unless there is a special reason to do otherwise we take $\mathbf{B}_1$ as a singleton B_1. Also, as we explained in the preceding paragraph a good idea is to take B_1 as a selector of $J_{\mathbf{X}}g \,\hat{\cdot}\, \mathbf{B}$.

However there is a problem with the described method. It requires constructing the weak inverse of $\mathbf{B}_1$. This may lead to very large interval matrices if $\mathbf{B}_1$ is ill-conditioned.

8.3. **Cuboids.** To avoid the problem with ill-conditioned matrices we can take cuboids instead of parallepipeds as our representable sets. A cuboid is a special case of a parallelepiped $\mathbf{q} := c + \mathbf{Q} \cdot \mathbf{r}$, where $c \in \mathbb{R}^n$, $\mathbf{r} \in \mathbb{I}^n$ and $\mathbf{Q}$ is an interval matrix which contains an orthogonal matrix. We have the following lemma.

Let $\mathbf{x} := x + \mathbf{Q} \cdot \mathbf{r} \subset D$ be a cuboid.

Lemma 8.4. *If $\mathbf{Q}_1$ be an interval matrix such that $\mathbf{Q}_1$ contains an orthogonal matrix, $(x_1, \mathbf{x}_1^0)$ is a decomposition of $\langle g \rangle(x)$ and*

$$\mathbf{r}_1 := \mathrm{Rearr}_1(J_{\mathbf{X}} g \,\hat{\cdot}\, \mathbf{Q}, \mathbf{Q}_1, \mathbf{r}, \mathbf{w} \,\hat{+}\, \mathbf{x}_1^0),$$

then $f(x + \mathbf{Q} \cdot \mathbf{r}) \subset x_1 + \mathbf{Q}_1 \cdot \mathbf{r}_1$.

Proof: Let Q_1 be an orthogonal selector of $\mathbf{Q}_1$. Then $\mathrm{id} = Q_1 \cdot Q_1^T \in \mathbf{Q}_1 \cdot \mathbf{Q}_1^T$, therefore $\mathbf{Q}_1^T$ is a weak inverse of $\mathbf{Q}_1$ and the result follows from the preceding lemma. □

To keep wrapping small Lohner proposes to take as $\mathbf{Q}_1$ the orthogonal factor in the orthogonal-triangular decomposition of a selector of $J_{\mathbf{X}} g \,\hat{\cdot}\, \mathbf{B}$.

8.4. **Doubletons.** The need for inverting matrices in the Lohner method arises from the fact that in course of computation there appear some extra interval vectors (for instance the nonlinear terms sitting in the function w), which have to be incorporated into a parallelepiped enclosure as in Lemma 8.1. However, if these interval vectors are small with respect to the size of the set S, then it is worth to keep the extra terms separately, so that the matrix inversion problem will be solved only with respect to small terms. In other words we use the sets of the form $x + C \cdot \mathbf{r}_0 + \mathbf{s}$, where x is a vector, C is a matrix, and $\mathbf{r}_0$ is an interval vector and $\mathbf{s}$ is a small set from a certain class of representable sets. We call such sets doubletons (Lohner calls the corresponding method an inner enclosure). The encoding of a representable doubleton consists of $(x, C, \mathbf{r}_0)$ and the encoding of $\mathbf{s}$. Formally a doubleton is the algebraic sum of a parallelepiped and a representable set $\mathbf{s}$ and we want to keep them in this separated way. The parallelepiped, $x + C \cdot \mathbf{r}_0$ is used to store the main part of the image of the set S as if there were no additional terms present in the course of the computation. Here we avoid multiplying inverted matrices by a relatively large interval set $\mathbf{r}_0$. The set $\mathbf{s}$ collects all the remainders which appear during the computation. In practice we make take $\mathbf{s}$ in the form of an interval set, another parallelepiped, a cuboid or any other representable set in a class for which we have an algorithm encl which for a given set $\mathbf{s}$ in this class and a given $\mathbf{A} \in \mathbb{I}^{n \times n}, \mathbf{v} \in \mathbb{I}^n$ constructs a set $\mathrm{encl}(\mathbf{A}, \mathbf{v}, \mathbf{s})$ in this class such that

$$\mathbf{A} \cdot \mathbf{s} + \mathbf{v} \subset \operatorname{encl}(\mathbf{A}, \mathbf{v}, \mathbf{s}).$$

Thus let $\mathbf{x} := x + C \cdot \mathbf{r}_0 + \mathbf{s} \subset D$ be a doubleton.

Lemma 8.5. *Assume $\mathbf{B}_1$ is an interval matrix, $\mathbf{B}_1^*$ is a weak inverse of $\mathbf{B}_1$, $C_1, \mathbf{C}_1^0$ is a decomposition of $J_{\mathbf{X}} g \hat{\cdot} C$, $(x_1, \mathbf{x}_1^0)$ is a decomposition of $\langle g \rangle(x)$. Then*

$$f(x + C \cdot \mathbf{r}_0 + \mathbf{s}) \subset x_1 + C_1 \cdot \mathbf{r}_0 + \operatorname{encl}(J_{\mathbf{X}} g, \mathbf{w} \hat{+} \mathbf{x}_1^0 \hat{+} \mathbf{C}_1^0 \hat{\cdot} \mathbf{r}_0, \mathbf{s}).$$

Proof: Let $u \in x + C \cdot \mathbf{r}_0 + \mathbf{s}$. Then $u = x + C \cdot r_0 + s$ for some $r_0 \in \mathbf{r}_0$ and $s \in \mathbf{s}$. Thus we have

$$\begin{gathered} f(u) = g(u) + w(u) \in \\ g(x) + J_{\mathbf{X}} g \cdot C \cdot \mathbf{r}_0 + J_{\mathbf{X}} g \cdot s + w(u) \subset \\ \langle g \rangle(x) + J_{\mathbf{X}} g \cdot C \cdot \mathbf{r}_0 + J_{\mathbf{X}} g \cdot s + \mathbf{w} \subset \\ x_1 + C_1 \cdot \mathbf{r}_0 + J_{\mathbf{X}} g \cdot s + (\mathbf{w} + \mathbf{x}_1^0 + \mathbf{C}_1^0 \cdot \mathbf{r}_0) \subset \\ x_1 + C_1 \cdot \mathbf{r}_0 + J_{\mathbf{X}} g \cdot s + (\mathbf{w} \hat{+} \mathbf{x}_1^0 \hat{+} \mathbf{C}_1^0 \hat{\cdot} \mathbf{r}_0) \subset \\ x_1 + C_1 \cdot \mathbf{r}_0 + \operatorname{encl}(J_{\mathbf{X}} g, \mathbf{w} \hat{+} \mathbf{x}_1^0 \hat{+} \mathbf{C}_1^0 \hat{\cdot} \mathbf{r}_0, \mathbf{s}). \end{gathered}$$

□

8.5. **Elliptic sets.** An elliptic set (an ellipsoid) in $\mathbb{R}^n$ may be written as $\mathbf{e} := c + L \cdot \mathbf{b}$, where $c \in \mathbb{R}^n$, $L \in \mathbb{R}^{n \times n}$ and $\mathbf{b}$ is a Euclidean ball. The elliptic set $\mathbf{e}$ is representable if c, L and $\mathbf{b}$ are representable.

Elliptic sets as a tool in the enclosing problem were introduced by A. Neumaier [13] and we refer the reader there for a very clear presentation of an enclosing problem algorithm based on elliptic sets.

8.6. **Large or irregular sets.** When the set S is very large or has an irregular shape, a covering strategy may be used to obtain a good upper bound. We start with a family $\mathcal{C}$ of certain representable sets which forms a locally finite minimal covering of $\mathbb{R}^n$. We call the sets in $\mathcal{C}$ elementary representable sets. Possible choices of elementary representable set are for example: a family of small representable intervals in $\mathbb{R}^n$ forming a grid in $\mathbb{R}^n$ or a collection of balls in $\mathbb{R}^n$ of fixed representable radius and centers at a grid of representable vectors in $\mathbb{R}^n$. We then define representable sets as sets which are finite unions of elements in $\mathcal{C}$. The encoding for such sets is just the list in a certain fixed order of the elements of the union. If S is represented by the union of $\mathbf{s}_1, \mathbf{s}_2, \ldots \mathbf{s}_n$ then $f(S) \subset f(\mathbf{s}_1) \cup f(\mathbf{s}_2) \cup \ldots \cup f(\mathbf{s}_n)$ and the problem of enclosing $f(S)$ is reduced to enclosing the sets $f(\mathbf{s}_i)$ for $i = 1, 2, \ldots, n$.

9. Numerical examples.

We tested the methods discussed in this paper on dynamical systems obtained from:

1 the classical Rössler equations
2 the classical Lorenz equations
3 a three dimensional ODE derived form the Kuramoto-Sivashinsky equations
4 the Hénon map

In the first three cases we took as f the h-translation along the trajectories of the flow and as its rational approximation g the 5-th order Taylor method. In case of the Hénon map we took the Hénon map itself as both f and g.

We performed two kinds of tests: *blowUp* and *Poincare map*.

BlowUp test We iterated the process of enclosing a small initial set until the ratio of the size of enclosure to the size of the initial set passed above a prescribed value (called later *blowUp limit*).

Poincare map test. We compute a Poincare map P for a flow induced by an ODE on some section Θ, which was relevant for the computer assisted proofs [4], [7], [16]. We are interested in two quantities. First one is L, which is given as the ratio of the size of enclosure of $P(\mathbf{x})$ to the size of the initial set $\mathbf{x}$. We will call L a computed Lipschitz constant.

Another quantity is a local computation cost for a given method denoted by C_l. We define C_l as follws: Let c be a time (cost) of computation of $\mathbf{x}$. We set

$$C_l = Lc \tag{13}$$

We justify the above definition as follows. Our goal is to compute $P(S)$ for some set S with a prescribed error $\epsilon > 0$. Let $S \subset \bigcup_{i=1}^k \mathbf{x}_i$. C a total cost of computation of $P(S)$ is given by

$$C = kc$$

Observe that the size of $\mathbf{x}_i$ is given by the following error equation

$$\frac{\mathrm{diam}(\mathbf{x}_i)L}{2} < \epsilon \tag{14}$$

From this we see that we have the following relation

$$k \approx \frac{\mathrm{diam}(S)^d}{\mathrm{diam}(\mathbf{x})^d} \geq L^d \left(\frac{\mathrm{diam}(S)}{2\epsilon}\right)^d \tag{15}$$

where d is a dimension of S. Hence we see that

$$C \sim L^d c \tag{16}$$

In out test examples we have $d = 1$.

Observe that in (14) we silently assumed that the only source of error in computation $P(S)$ is related to space discretization. In real computations we have also round–off errors and errors related to time discretization. For discussion of these errors and their dependence upon various numerical methods the reader is referred to [18].

9.1. **Rössler equations.** Rössler equations are given by

$$\begin{aligned} x' = & \quad -y - z \\ y' = & \quad x + by \\ z' = & \quad b + y(x - a) \end{aligned} \tag{17}$$

where $a = 5.7$ and $b = 0.2$.

Let P be a Poincaré map for a section $\Theta = \{x = 0, y < 0, -y - z > 0\}$. This is a map studied in [17]. The observed Poincaré return time belongs to $(4.5, 6.5)$.

Tests are performed with time step $h = 0.0025$ for interval sets $\mathbf{v}_1 = (0, -10.2, 0) + \{0\} \times [-5 \cdot 10^{-6}, 5 \cdot 10^{-6}]^2$, $\mathbf{v}_2 = (0, -3.8, 0) + \{0\} \times [-0.001, 0.001]^2$. For blowUp test we set *blowUpLimit*=100 for $\mathbf{v}_1$ and *blowUpLimit*=10 for $\mathbf{v}_2$.

The results of tests are in contained in tables 1 and 2. Symbol ∞ in Tab. 2 means that due to blow up of the size of the set during computations we were unable to compute Poincaré map.

The results of blowUp tests show that we have significant wrapping effect. No blow up was observed for interval and cuboid doubletons. The failure of performing so well for the parallelepiped doubleton is related to problems with inversion of ill-conditioned matrices.

In Poincaré map test we get a qualitatively different results for $\mathbf{v}_1$ and $\mathbf{v}_2$. From the point of view of rigorous numerics these sets differ significantly. $\mathbf{v}_2$ represents an example of an *easy set*, whereas $\mathbf{v}_1$ is an *difficult set*. This distinction as justified as follows: the wrapping effect during computation of $P(v_2)$ is small, whereas is very big for $P(v_1)$. As a result we see that for $\mathbf{v}_2$ the winner (minimal C_l) are euclidean logarithmic balls, but interval and parallelepiped doubletons are performing almost as good.

But for a difficult set $\mathbf{v}_1$ the doubletons cuboid and interval are nearly 10^3-times better then euclidean logarithmic balls.

In [17] Poincare map was computed using euclidean logarithmic balls and the total computation time was around 50 hours. From the table

Sets	$\mathbf{v}_1$	$\mathbf{v}_2$
logarithmic balls:		
euclidean	1.67	7.77
max	1.404	1.153
sum	1.449	1.92
balls:		
euclidean	1.359	0.47
max	1.313	0.48
sum	1.215	0.39
single sets		
interval	1.725	1.98
parallelepiped	4.595	3.73
cuboid	1.842	8.125
ellipsoid	4.36	3.39
doubletons		
interval	> 10	> 10
parallelepiped	4.595	3.73
cuboid	> 10	> 10

TABLE 1. BlowUp times for Rössler system.

2 it follows that using doubletons we can redo these computation in just few minutes.

9.2. **Tests for Lorenz equation, Kuramoto-Shivashinski equations and Hénon map.** Lorenz equations are given by

$$\begin{aligned} x' &= s(y-x) \\ y' &= (r-z)x - y \\ z' &= xy - qz \end{aligned} \tag{18}$$

where $s = 10$, $r = 28$ and $q = 8/3$. We set $\Theta = \{z = r - 1\}$. We made tests for the set $\mathbf{x} = (0.137, 0.378, 27) + [-0.0005, 0.0005]^2 \times \{0\}$. The time step h was equal to 0.003. This is an example of difficult point for computations from [4]. For blowUp test we set *blowUpLimit* equal to 100.

We made test also for the following three dimensional ODE derived from Kuramoto-Sivashinsky equations (see [12])

$$\begin{aligned} x' &= y \\ y' &= z \\ z' &= -d^2ly - x^2/2 + d^6 \end{aligned} \tag{19}$$

Sets	$L(\mathbf{v}_1)$	$C_l(\mathbf{v}_1)$	$L(\mathbf{v}_2)$	$C_l(\mathbf{v}_2)$
logarithmic balls:				
euclidean	$1.33 \cdot 10^4$	1	3.53	1
max	∞	∞	∞	∞
sum	∞	∞	∞	∞
balls:				
euclidean	∞	∞	∞	∞
max	∞	∞	∞	∞
sum	∞	∞	∞	∞
single sets:				
interval	4440	0.71	1530	950
parallelepiped	∞	∞	∞	∞
cuboid	435.7	0.073	3.23	2.04
ellipsoid	∞	∞	∞	∞
doubletons:				
interval	4.99	$8.1 \cdot 10^{-4}$	2.26	1.39
parallelepiped	∞	∞	∞	∞
cuboid	4.96	$8.4 \cdot 10^{-4}$	2.21	1.41

TABLE 2. Results of Poincare map test for Rössler system

where $d = 1$ and $l = 1$.

Hénon map is given by

$$H(x, y) = (1 - ax^2 + y, bx) \tag{20}$$

where $a = 1.4$ and $b = 0.3$. We do test for $\mathbf{x} = (1,1) + [-0.5 \cdot 10^{-5}, 0.5 \cdot 10^{-5}]^2$. For blowUp test we set *blowUpLimit* equal to 100.

The table 3 collects the results of blowUp test.

From the contents of tables 3 and 4 one can see that the winner for Lorenz equation is doubleton cuboid , but it is only three times better than euclidean logarithmic balls, which were used in [4].

REFERENCES

[1] R. Anguelov, S. Markov, Wrapping effect and wrapping function, *Reliable Computing* **4**(1998), 1-20.

[2] R. Anguelov, Wrapping function of the initial value problem for ODE: Applications, *Reliable Computing*, to appear.

[3] George F. Corliss, Robert Rihm, Validating an a priori enclosure using high-order Taylor series, *Scientific computing and validated numerics (Wuppertal, 1995)*, 228–238, Math. Res., 90, Akademie Verlag, Berlin, 1996.

Set	Lorenz	K-S	Hénon
interval	0.59	2.28	13
parallelepiped	0.42	2.71	7
cuboid	1.14	2.63	13
ellipsoid	0.44	2.84	8
doubletons:			
interval	1.23	2.87	13
parallelepiped	0.42	2.71	7
cuboid	2.54	2.86	13
balls:			
euclidean	0.28	1.71	8
max	0.23	1.56	5
sum	0.25	1.68	8
logarithmic balls:			
euclidean	0.61	2.14	
max	0.41	1.56	
sum	0.45	1.68	

TABLE 3. BlowUp test results

[4] Z. Galias and P. Zgliczyński, Computer assisted proof of chaos in the Lorenz system, *Physica D*, **115**, 1998,165-188.

[5] E. Hairer, S.P. Nørsett and G. Wanner, *Solving Ordinary Differential Equations I, Nonstiff Problems*, Springer-Verlag, Berlin Heidelberg 1987.

[6] R.J. Lohner, Computation of Guaranteed Enclosures for the Solutions of Ordinary Initial and Boundary Value Problems, in: *Computational Ordinary Differential Equations*, J.R. Cash, I. Gladwell Eds., Clarendon Press, Oxford, 1992.

[7] K. Mischaikow and M. Mrozek, Chaos in Lorenz equations: a computer assisted proof, *Bull. Amer. Math. Soc. (N.S.)*, **32**(1995), 66-72.

[8] K. Mischaikow and M. Mrozek, Chaos in the Lorenz Equations: A Computer Assisted Proof. Part II: Details, *Mathematics of Computation*, **67**(1998), 1023-1046.

[9] K. Mischaikow, M. Mrozek and A. Szymczak, Chaos in the Lorenz equations: a computer assisted proof. Part III: the classical parameter values, submitted.

[10] R.E. Moore, *Interval analysis*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1966.

[11] M. Mrozek, Topological invariants, multivalued maps and computer assisted proofs, *Computers & Mathematics*, **32**(1996),83-104.

[12] M. Mrozek and M. Żelawski, Heteroclinic connections in the Kuramoto-Sivashinsky equation, *Reliable Computing*, 3(1997), 277-285.

[13] A. Neumaier, The wrapping effect, ellipsoid arithmetic, stability and confidence regions, *Computing Supplementum*, **9** (1993), 175-190.

[14] M. Warmus, Calculus of approximations, *Bull. de l'Académie Polonaise des Sciences Cl. III*, **4**(1956), 253-259.

Set	Lorenz	K-S
logarithmic balls:		
euclidean	1.00	1.00
max	∞	2.14
sum	∞	3.39
sets:		
interval	62.9	5.08
parallelepiped	∞	2.86
cuboid	0.59	3.66
ellipsoid	∞	2.42
doubletons:		
interval	0.39	2.82
parallelepiped	∞	2.90
cuboid	0.31	2.93
balls:		
euclidean	∞	2.46
max	∞	2.17
sum	∞	3.47

TABLE 4. Results for Poincaré map tests, local computation cost comparison

[15] M. Warmus, Approximation and Inequalities in the Calculus of Approximations. Classification of Approximate Numbers, *Bull. de l'Académie Polonaise des Sciences, Série des sciences math., astr. et phys.*, **9**, (1961), 241-245.

[16] P. Zgliczyński, Rigorous verification of chaos in the Rössler equations, *Scientific Computing and Validated Numerics*, G. A1efe1d, A. Frommer and B. Lang editors, Akademie Ver1ag, Ber1in, 1996, 287-292

[17] P. Zgliczyński, Computer assisted proof of chaos in the Hénon map and in the Rössler equations, *Nonlinearity*, **10**(1997), 243-252.

[18] P. Zgliczyński, Remarks on computer assisted proof of chaotice behavior in ODE's, in preparation.

INSTYTUT INFORMATYKI, UNIWERSYTET JAGIELLOŃSKI, KRAKÓW, POLAND
E-mail address: `mrozek@ii.uj.edu.pl`

SCHOOL OF MATHEMATICS, GEORGIA INSTITUTE OF TECHNOLOGY, ATLANTA, GEORGIA, USA.
E-mail address: `piotrz@math.gatech.edu`