Jagiellonian University

Faculty of Mathematics and Computer Science

Institute of Computer Science and Computational Mathematics

A dissertation submitted for the degree
Doctor of Philosophy in Computer Science

Robert Szczelina

Rigorous integration of Delay Differential Equations

Supervisor:
prof. dr hab. Piotr Zgliczynski

Kraków, 2014

Acknowledgments

I would like to thank my supervisor, Professor Piotr Zgliczyński, for all his invaluable advices, discussions and effort he put into leading me to my PhD dissertation.

I am extremely grateful to my Family, especially my Mom and Dad for their constant support thorough all the way of my education.
I would like also to thank Marzena for her love, patience and a lot of support during all years of the PhD studies.

This research was supported in part by NCN grant no. 2011/03/B/ST1/0478 and by the SSDNM joint PhD programme co-financed by the European Social Fund through the Operational Programme Human Capital.

Podziękowania

Chciałbym podziękować mojemu opiekunowi, profesorowi Piotrowi Zgliczyńskimu, za wszystkie cenne rady, wskazówki i dyskusje oraz za wysiłek włożony w poprowadzenie mnie w tej pracy.

Chciałbym wyrazić niezmierną wdzięczność dla mojej rodziny, szczególnie moim rodzicom, za ich nieustające wsparcie w trakcie trwania całej mojej edukacji.
Chciałbym także podziękować Marzenie za jej miłość, cierpliwość i wiele wspracia podczas trwania całych studiów doktorskich.

Ta praca została w części sfinansowana z grantu NCN o numerze 2011/03/B/ST1/0478 oraz z programu Środowiskowych Studiów Doktoranckich z Nauk Matematycznych (SSDNM) współfinansowanego przez Unię Europejską w ramach Programu Operacyjnego Kapitał Ludzki.

to my parents

---

moim rodzicom

Contents

Chapter 1

Introduction

1.1 Motivation

Delay Differential Equations (DDEs) are a type of differential equations in which the derivative of the function in a current time depends on one or more values of the function in the past. DDEs gained wide interest because they are perfectly suited to model systems with aftereffect (time lag) - systems which play an important role in many applied fields of science like biology, engineering and the control theory. In particular, they may be used to model real life phenomena such as: transport lags between two compartments in living cells; population dynamics that include maturation times; aspects of control theory where control signals must pass some distance between the controller and the controlled device; to name only the few. One can find large number of examples in the literature ([9, 11] and references therein): from describing several aspects of infectious disease dynamics [5], drug therapy [17], immune response [6, 31], chemostat models [36], circadian rhythms [20], epidemiology [7], the respiratory system [25], tumor growth [26] and neural networks [4, 30]. Statistical analysis of ecological data [23, 24] has shown that there is evidence of delay effects in the population dynamics of many species.

The advantage of using DDEs lies in the fact that one can create very simple models (even with a single variable) which produce rich dynamical behavior that can be reasonably explained using small number of parameters. The reason for this is that the dynamical behaviour generated by DDEs can be seen as *infinite dimensional*. Thus, by introducing a few variables and parameters one can obtain dynamical system in a very big phase-space. Moreover, we can tract solutions in the basic variables which is advantageous in contrast to simplifying methods for large systems of ODEs [10], which may lose important informations.

There are many extensively studied classes of solutions to DDEs such as stationary solutions and periodic orbits. It is also important to investigate other dynamical properties of DDEs such as connecting orbits (between stationary points and/or periodic orbits) that can reveal (part of) the structure of (global) attractors and invariant manifolds, which

together constitute the image of the full dynamics of the system and dictates its possible behaviour for the long time evolution. Those properties are usually studied numerically, but rigorous, mathematical proofs for their existence is often limited to the simplest cases, where solutions can be computed analytically.

In recent years there were many important applications of computer assisted proofs of various dynamical properties in discrete maps, ordinary differential equations (ODEs) and (dissipative) partial differential equations (PDEs), see for example [29, 28, 12, 21, 34, 27] and references therein. By the computer assisted proof we mean a computer program which rigorously check assumptions of abstract theorems about the existence of some dynamical property. In this work we are going to extend this rich theory to the case of DDEs by creating a rigorous integration scheme for DDEs with bounded delays. By the rigorous integration we understand a computer procedure which produces strict bounds for the solution - in this case a finite set of representable numbers $\bar{B}$ which describes a subset B of some functional space with a property that a real solution $x(t)$ to a given DDE system belongs to B. Such a rigorous integrator may then be used in computer assisted proofs of many dynamical properties such as the existence of stationary points, periodic orbits, homo- or hetero- clinic connections and chaos.

1.2 Results of this work

We have created rigorous numerical algorithms to compute enclosures for the solutions to scalar DDEs with single, bounded, constant delay of the form $\dot{x} = f\left(x(t-1), x(t)\right)$. We have proved their correctness and we measured their performance on the exemplary DDEs. The methods are very promising, as they exhibit strong contraction on the estimates for high order derivatives of the solutions, and this allow to represent the solutions by the finite-dimensional representation. Moreover, we have developed algorithms to compute Poincaré maps associated with a given DDE, that may be used in the computer assisted proofs of periodic orbits both stable and unstable ones. To our knowledge the rigorous integration of DDEs is a new subject not studied extensively yet, and we do not know of currently available rigorous integrator designed for DDEs.

There are several works that deals with the rigorous numerics for periodic solutions to DDEs, see [13, 32] but the approach used there is very different from the presented work. Those mentioned works concentrate on the expansion of the periodic solutions into Fourier series and then on solving some algebraic equations. There is no notion of the rigorous integration and the applicability of those methods is restricted to prove the existence of periodic solutions. The rigorous integration presented in this work may be used to prove the existence of periodic solutions but also for connecting orbits and other dynamical properties, for which it is necessary to integrate solutions forward in time.

1.3 Organization of this work

The paper is organized as follows: in Chapter 2 we gather all necessary notations regarding DDEs and we present some well-known facts about solutions to DDEs, together with a short review of currently available theoretical literature on this subject. We also present interesting examples that were shown to generate complex dynamical behaviour. Those examples will be used to test our methods. Chapter 3 is devoted to the fixed step size method for fast and rigorous computation of numerical solutions to DDEs with bounded delays. We present methods and discuss its performance on exemplary systems. In Chapter 4 we present strategy for reducing the wrapping effect during long integrations, based on similar strategy for ODEs proposed by Lohner. We also compare the performance of Lohner sets and basic interval set arithmetic. In Chapter 5 we present two alternative algorithms for computing representations after an arbitrary time step smaller than basic, fixed step size of the integrator from Chapter 3 - those methods are used to construct Poincaré maps. We discuss advantages and disadvantages of those methods and we assess their performance. In the last Chapter 6 the work is summarized and possible future directions for development are proposed.

1.4 Notation

In this work we use the following notation. For a function $f : \mathbb{R} \to \mathbb{R}$, By $f^{(k)}$ we denote k-th derivative of f. By $f^{[k]}$ we denote the term $\frac{1}{k!} \cdot f^{(k)}$. By $f'(t^-)$ and $f'(t^+)$ we denote the left-hand side and right-hand side derivative of f w.r.t. t respectively.

For $F : \mathbb{R}^m \to \mathbb{R}^n$ by $DF(z)$ we denote the matrix $\left(\frac{\partial F_i}{\partial x_j}(z)\right)_{i\in\{1,..,n\},j\in\{1,..,m\}}$ i.e. a Jacobian matrix of the function F computed at the point $z \in \mathbb{R}^m$.

Let $A = \Pi_{i=1}^n [a_i, b_i]$, for $a_i \leq b_i$, $a_i, b_i \in \mathbb{R}$. We call A *an interval set* (a product of closed intervals in $\mathbb{R}^n$). For two sets $A \subset \mathbb{R}^n$ and $B \subset \mathbb{R}^n$ we denote by $[A, B]$ a minimal interval set, such that $A \cup B \subset [A, B]$. In case of A and B being closed intervals in $\mathbb{R}$, then $[A, B]$ is an interval $[\min(A \cup B), \max(A \cup B)]$. For sets $A \subset \mathbb{R}$, $B \subset \mathbb{R}$, $a \in \mathbb{R}$ and for some binary operation $\diamond : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ we define $A \diamond B = \{a \diamond b : a \in A, b \in B\}$ and $a \diamond A = A \diamond a = \{a\} \diamond A$. Analogously, for $f : \mathbb{R} \to \mathbb{R}$ and a set $A \in \mathbb{R}$ we define $f(A) = \{f(a) : a \in A\}$.

Let $D \subset \mathbb{R}$ be a compact set and let denote by $C^r(D, \mathbb{R})$ the space of all real valued functions of class C^r over D equipped with standard supremum norm: $\|f\| = \sum_{i=0}^r \sup_{x\in D} |f^{(i)}(x)|$.

Let $\tau \in \mathbb{R}$ such that $\tau > 0$. By $C^k(\tau)$, $k \in \{0, 1, 2..., \infty\}$, we denote space $C^k([-\tau, 0], \mathbb{R})$. In case of $\tau = 1$ we will abuse notation and we will denote $C^k(1)$ by C^k for simplicity.

For a given function $x : [-\tau, +\infty) \to \mathbb{R}^n$ we denote by x_t a function in $C^k([-\tau, 0], \mathbb{R})$ such that $x_t(s) = x(t + s)$ for all $s \in [-\tau, 0]$.

For $v \in \mathbb{R}^n$ by $\pi_i v$ for $i \in \{1, 2, .., n\}$ we denote the projection of v onto the i-th coordinate. For vectors $u, v \in \mathbb{R}^n$ by $\langle u, v\rangle$ we denote the standard scalar product: $\langle u, v\rangle = \sum_{i=1}^n \pi_i v \cdot \pi_i u$

Chapter 2

Preliminaries

2.1 Delay Differential Equations

In this work we are going to deal only with bounded delays and we will work with scalar equations. All theorems may be extended to higher dimensions.

The most general form of the DDE with bounded delays is [3, 8]:

$$\dot{x} = f(t, x_t) \tag{2.1}$$

where for $0 < \tau \in \mathbb{R}$, we have $x_t \in C^0(\tau)$ and $f : \mathbb{R} \times C^0(\tau) \to \mathbb{R}$ is a given function. In this context $\dot{x}$ stands for a right-hand-side derivative of x w.r.t. time t.

The initial value problem (IVP) is naturally given by:

$$\begin{cases} \dot{x} = f(t, x_t), & t \geq 0, \\ x(t) = \psi(t), & t \in [-\tau, 0], \end{cases} \tag{2.2}$$

where $\psi : [-\tau, 0] \to \mathbb{R}$ is called an *initial function*. Please notice that, in general, function ψ need not to be continuous. This is the case in many areas of research like control theory or neural networks where initial conditions are often step functions.

We assume that for each interval $I \subset \mathbb{R}$, $I = [a, b]$, $a \leq b$ and each continuous function $u : [-\tau + a, b] \to \mathbb{R}$ the function $\tilde{f}(t) = f(t, u_t)$ is integrable on the domain I.

The following definitions are natural.

Definition 1 *The function $u : [-\tau + a, b] \to \mathbb{R}^n$ is called a* solution of (2.1) on the interval $I = [a, b]$ *if u is continuous on I and*

$$u(t) = u(a) + \int_a^t f(s, u_s) ds \tag{2.3}$$

holds for all $t \in [a, b]$.

Definition 2 *The function $u : \mathbb{R} \to \mathbb{R}$ is called a* full solution *of* (2.2) *if u is a solution of* (2.1) *on each interval $[a, b] \subset \mathbb{R}$, $a \leq b$ and $u_0 = \psi$.*

For a given ψ we denote by x^ψ the solution of (2.2) for a given initial data ψ, that is a solution to (2.1) interval $[-\tau, b]$ for the largest possible $b \in \mathbb{R}$.

As in case of ODEs, stationary and periodic solutions are an important classes of solutions. In the case of stationary solutions we look for functions $x : \mathbb{R} \to \mathbb{R}$ for which we have:

$$f(x, x_t) = 0, \quad \forall t \in \mathbb{R}. \tag{2.4}$$

If $f(u, u_t) = 0$ then $u_t \equiv \hat{u}$ for some $\hat{u} \in \mathbb{R}$ and all $t \geq 0$, but we are not able, in general, to expand it to $t < 0$. In spite of this, the following definition is natural:

Definition 3 *The function u is called* a stationary solution *of* (2.2) *if u is a full solution of* (2.2) *and for each $t \in \mathbb{R}$ $u(t) = \hat{u} \in \mathbb{R}$, $\hat{u} = const$.*

We see that if a solution $u(t) = \hat{u}$ for all $t \in \mathbb{R}$, then it must be that $f(u, u_t) = 0$.

In the case of periodic solutions we have:

Definition 4 *The function u is called* a periodic solution *of* (2.2) *if u is a full solution of* (2.2) *and there exist $T > 0$ (called a* period*) such that $u(t) = u(t + T)$ for each $t \in \mathbb{R}$. The smallest period T for a periodic solution u is called a* basic period*.*

Let assume now that $\psi \in C^0(\tau)$. Then we can define natural (local) semiflow φ in $C^0(\tau)$ induced by (2.1) by

$$\varphi : C^0(\tau) \times \mathbb{R}_+ \ni (\psi, t) \mapsto x_t^\psi \in C^0(\tau). \tag{2.5}$$

Thus the problem of finding solutions to DDEs reduces to investigating (local) semiflows in an infinite dimensional phase-space $C^0(\tau)$.

Remark 1 *If $u(t)$ is a stationary solution of* (2.2) *then u_0 is a fixed point of the corresponding semiflow φ. If $u(t)$ is a periodic solution of* (2.2) *then $\{u_t\}_{t \in \mathbb{R}}$ is a periodic orbit of the corresponding semiflow φ.*

Notice that the general form of problem (2.2) allows for various forms of the r.h.s $f(t, x_t)$. The function f may even consist of integrals e.g. $\int_{-\tau}^{0} x_t(s)ds$ as in Voltera integral formulas [19]. However in many practical and interesting cases it suffices to restrict investigations to the autonomous DDEs with several discrete delays $0 < \tau_i \leq \tau$ of the form:

$$\dot{x} = f(x(t), x(t - \tau_1), ..., x(t - \tau_m)), \quad f : \mathbb{R}^{m+1} \to \mathbb{R}, x \in \mathbb{R}. \tag{2.6}$$

The simplest form will be

$$\dot{x} = f(x(t), x(t - \tau)), \quad f : \mathbb{R}^2 \to \mathbb{R}, x \in \mathbb{R}. \tag{2.7}$$

This formulation of the problem (2.6) is until now one of the most studied. Yet, there can still be found systems with surprising dynamical behaviour [14]. Thus, it is still important to investigate such systems, especially for a strongly non-linear r.h.s f.

2.2 Statement of the problem

In this work we study the following question: *will it be possible to rigorously integrate* (2.7) *(or the corresponding semiflow) to obtain finite dimensional representation of the solution of quality good enough to verify assumptions of abstract theorems about existence of dynamical properties of the system?*

The exact meaning of this short sentence is as follows:

1. By a finite representation of a function f we understand a finite collection $\bar{B}$ of computer-representable numbers that unambiguously defines a subset B of some functional space such that $f \in B$;

2. By the rigorous integration we understand a computer procedure Φ_ε that, given a representation $\bar{A}$ of x_t, produces strict bounds in the same representation for the solution to (2.7) after time $t = \varepsilon$, that is if $\bar{B} = \Phi_\varepsilon(\bar{A})$ then $x_{t+\varepsilon} \in B$;

3. By the dynamical properties of the system we mean for example the existence of stationary points, periodic orbits, connecting orbits, symbolic dynamics, structure of the attractors, chaos, etc.;

4. By abstract theorems we mean theorems such as: the Brouwer theorem, the Conley Index, Covering Relations[33], etc. Usually, application of the mentioned theorems requires construction of the time-shift map (or a Poincaré map) $P(x) = \varphi(x, t_p(x))$, which is done by the recursive application of the single-step method Φ_ε;

To apply theorems mentioned above we need the *compactness* of the map P. It turns out that in the case of DDEs the compactness of a map P is not a problem at all as long as the solutions stays bounded. The real problem is to get apropriate bounds on the high-order derivatives of the solutions to the equation (2.7). Moreover, when it comes to investigate the dynamics of the map P by a finite, rigorous approximation it is also essential that we get contraction on their bounds (this is one of the requirements we mean by the sentence *the quality good enough* in the statement of the problem). In the following of this work we will focus on those underlined goals.

The compactness of the time shift map P comes from the smoothing of the solutions to DDEs and the following lemma which is a direct consequence of the Arzela-Ascoli Theorem:

Lemma 2 *Let $D \subset C^n(\tau)$, be closed, bounded. Assume that for $x \in D$, $x^{(n+1)}$ exists and $|x^{n+1}| \leq M$. Then D is compact (in C^n-norm).*

To simplify things, let assume that $t_p(x) \equiv \tau$ for all x, that is $P(x_0) = x_\tau$ where

$$x_\tau(t) = x_0(0) + \int_{-\tau}^{t} f(x_\tau(s), x_0(s))\, ds, \quad t \in [-\tau, 0], \tag{2.8}$$

wherever the integral exists. If the r.h.s. f of equation (2.7) is smooth enough then for $x_0 \in C^n(\tau)$ we have $P(x_0) = x_\tau \in C^{n+1}(\tau)$. Thus, if $x(t)$ is a *bounded* solution to (2.7), that is $|x(t)| \leq M_0$ for all t and some $M_0 \in \mathbb{R}$ then using time-shift map P we can guarantee that $x(t)$ has uniform bounds for derivatives up to $n+1$ for t sufficiently large. Thus, by Lemma 2, the map P is compact in any C^n-norm. The same is essentially true for any t_p as long as $t_p(x) \geq \tau$ but the considerations in this case are very technical. We will discuss this issue in more detail in Section 5.4.

2.3 Poincaré maps

A standard tool used in proving the existence of periodic orbits both in analytical consideration and in computer assisted proofs are the Poincaré maps, which are a special case of time-shift maps mentioned in previous section. Below we recall basic definitions concerning Poincaré sections and Poincaré maps. We also relate problem of finding periodic solutions in dynamical systems to finding stationary points for Poincaré maps defined for suitable sections. For simplicity we restrict ourselves to linear sections.

Definition 5 *Let V be a vector space over $\mathbb{R}$ and let $s : V \to \mathbb{R}$, $s \not\equiv 0$, be a linear map and let $a \in \mathbb{R}$. We call the set $S = \{x \in V : s(x) = a\}$ an* (s,a)-section. *If s and a are known from the context we will simply call the set S* the section.

Remark 3 *If $dim(V) = n$ then S is an $(n-1)$-dimensional hyper-surface in the space V.*

Definition 6 *Let V be a vector space over $\mathbb{R}$, let $\varphi(x,t)$ be a (semi)flow in V and let S be some (s,a)-section in V. Define $D \subset V$ such that for each $x_0 \in D$ there exist the smallest time $t_p(x_0) > 0$ such that*

$$\varphi\left(x_0, t_p(x_0)\right) \in S \tag{2.9}$$

We call $t_p : D \to \mathbb{R}$ a Poincaré return time.

Definition 7 *Let S be a (s,a)-section and let $B \subset S$ be such that t_p is defined on B. We say that φ is* transversal *to S on B iff there exist $\varepsilon > 0$ such that for each $x \in B$ the curve $\varphi\left(x, [t_p(x) - \varepsilon, t_p(x) + \varepsilon]\right)$ intersects B transversally, that is $\frac{d}{dt} s\left(\varphi(x,t)\right) > 0$ (respectively < 0) for all $t \in [t_p(x) - \varepsilon, t_p(x) + \varepsilon]$.*

Definition 8 *Let V be a vector space over $\mathbb{R}$, let $\varphi(x,t)$ be a (semi)flow in V and let S be some (s,a)-section in V. Let $B \subset S$ be such that t_p is defined on B and φ is transversal to S on B. Then the map*

$$P_{\varphi,B} : B \to S \tag{2.10}$$

defined by

$$P_{\varphi,B}(x) = \varphi(x, t_p(x)), \quad x \in B \tag{2.11}$$

is called a Poincaré map. *We skip subscripts φ and B when they are known from the context and we simply write $P(x) = \varphi(x, t_p(x))$.*

We see that the question of the existence of periodic orbits in (semi)flows may be stated as a problem of existence of fixed points of some Poincaré maps. Namely if we have a fixed point x_0 of the Poincaré map P, $P(x_0) = x_0$, then the function $x(t) = \varphi(x_0, t)$ is periodic with the basic period $T = t_p(x_0)$.

In the next section we present an interesting application of the Poincaré map to a problem of finding periodic solutions in some DDE and we present sample problems on which we are going to test our rigorous integration procedure.

2.4 Exemplary problem for consideration

Here we will present a sample problem investigated by Krisztin and Vas in their work [14]. We propose variant of this problem with smooth r.h.s for which analytical computations are too difficult to carry out by hand and for which we may use the rigorous integrator presented in the remainder of this work.

Consider the following equation:

$$x'(t) = -x(t) + f\left(x(t-1)\right), \tag{2.12}$$

where:

$$f(x) = \begin{cases} -7 & \text{if } x \leq -1 \\ 0 & \text{if } x \in (-1, 1) \\ 7 & \text{if } x \geq 1 \end{cases} \tag{2.13}$$

Krisztin and Vas in their work [14] showed that for this system there exist periodic solutions O_p and O_q such that:

- the solutions oscillate around stationary solutions $x \equiv 1$ and $x \equiv -1$, that is $\{-1, 1\} \subset O_{p/q}(\mathbb{R})$,
- for any interval $[-1 + a, a]$ the graph of $O_{p|[-1+a,a]}$ and $O_{q|[-1+a,a]}$ crosses $x = 0$ at most 2 times.

Periodic solutions with such properties are called Large-Amplitude Slowly Oscilatory Periodic (LSOP) solutions.

Krisztin and Vas have constructed a Poincaré map in a suitable three-dimensional invariant subspace D of the space C^0 and showed that this map has two hyperbolic stationary points o_p and o_q (with one and two unstable directions respectively) that corresponds to LSOP solutions O_p and O_q. Their approach however required the system to be in form simple enough to allow analytical computation of the explicit solutions to (2.12) in the subspace D. This is (almost) impossible when we work with smooth, highly nonlinear r.h.s. In this work we are going to test our approach with the following exemplary equations:

$$x'(t) = -x(t) + \frac{1}{2} \cdot x(t-1)^2 \tag{2.14}$$

and

$$x'(t) = -x(t) - 3.2 \cdot x(t-1) + x(t-1)^3. \tag{2.15}$$

Numerical analysis suggests that system (2.14) has a stable stationary solution $x \equiv 0$ and system (2.15) has a stable periodic solution presented in Figure 2.1 (numerical approximation).

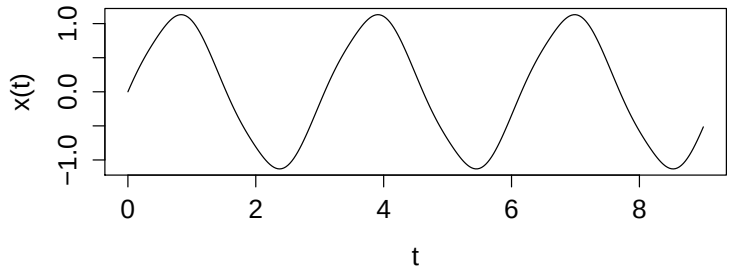

Figure 2.1: Numerical approximation to the stable periodic solution to system (2.15).

The future goal is to be able to present computer assisted proof of existence of a LSOP solution to the system (2.16), for which we have found a numerical candidate presented in Figure 2.2.

$$x'(t) = -x(t) + x(t-1) \cdot \left(-3.6 + x(t-1)^2 \cdot \left(4.7 - 0.1 \cdot x(t-1)^2\right)\right) \tag{2.16}$$

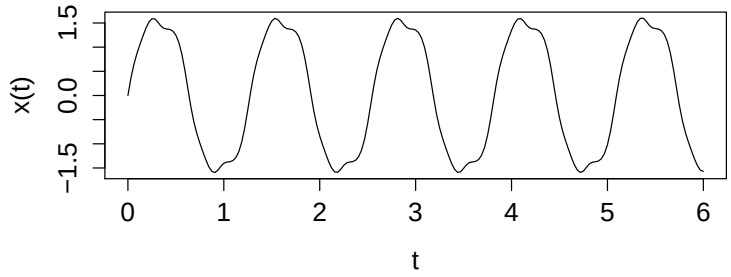

Figure 2.2: Numerical approximation to the LSOP solution to system (2.16).

Chapter 3

Rigorous integration of Delay Differential Equation

We are interested in the rigorous integration of the DDE (2.7). Without loss of generality and for the simplicity of presentation we can restrict ourselves to the case of the following equation:

$$x'(t) = f(x(t-1), x(t)), \quad x \in \mathbb{R} \tag{3.1}$$

By rigorous integration we mean a procedure that gives bounds on the solution (value and derivatives of it) at each time t in some interval $[-1, T]$. We assume that f is a 'nice' function, i.e. it is sufficiently smooth for the representation we will use (for simplicity we usually assume $f \in C^\infty$). All presented algorithms and procedures can be generalized to any dimensions.

3.1 Taylor representation of piecewise smooth functions

To create a rigorous integrator for the problem (3.1) we need a finite representation of some function spaces. We are going to work with smooth or piecewise smooth functions.

Definition 9 *Let $n > 0$ and $p > 0$ be fixed integers and let $h = \frac{1}{p}$. By C_p^n we denote the set of all functions $f : [-1, 0] \to \mathbb{R}$ such that f is of class C^{n+1} on each interval $[-i \cdot h, (-i+1) \cdot h)$ for $i \in \{1, .., p\}$.*

Remark 4 *Not all functions in C_p^n need to be continuous on the whole interval $[-1, 0]$.*
Example: *A function*

$$x(t) = \begin{cases} 1 & t \in \left[-1, -\frac{1}{2}\right) \\ 0 & t \in \left[-\frac{1}{2}, 0\right] \end{cases} \tag{3.2}$$

is by definition in C_2^n for any n but it is not continuous.

Definition 10 *We assume that all the derivatives in this definition are computed as forward in time derivatives w.r.t. time t. Let $n > 0$ and $p > 0$ be fixed integers, let $h = \frac{1}{p}$ and let $f \in C_p^n$ be given. An indexed collection $\bar{f}$ of closed intervals $\bar{f}^{i,[k]} \subset \mathbb{R}$, $(i,k) \in \{(i,k) : i \in \{1,..,p\}, k \in \{0,..,n+1\}\} \cup \{(0,0)\}$ such that:*

- $f(0) \in \bar{f}^{0,[0]}$,
- $f^{[k]}\left(-\frac{i}{p}\right) \in \bar{f}^{i,[k]}$ *for* $1 \leq i \leq p$, $0 \leq k \leq n$,
- $f^{[n+1]}\left(-\frac{i}{p} + \xi\right) \in \bar{f}^{i,[n+1]}$ *for all* $\xi \in [0,h)$ *and* $1 \leq i \leq p$,

is called the (interval) (p,n)-representation of f. *We also say that f is* (p,n)-representable with $\bar{f}$.

We call $\bar{f}^{i,[k]}$ the (i,k)-th coefficient of the representation *and we call $\bar{f}^{i,[n+1]}$* the i-th remainder of the representation. *The collection of all $\bar{f}^{i,[n+1]}$ is called* the remainder of the representation.

When parameters n and p are known from the context we will omit them and we will simply call $\bar{f}$ *the representation* of f.

Remark 5 *Not all functions in C_p^n have a (p,n)-representations. A function*

$$x(t) = \begin{cases} ln(-t) & t \in [-1,0) \\ 0 & t = 0 \end{cases} \tag{3.3}$$

is C_p^n but it has non-bounded derivative on the interval $[-1,0)$.

Remark 6 *In this work, we will slightly abuse notation and we will write for the remainder part that:*

$$f^{[n+1]}\left(-\frac{i}{p} + \xi\right) \in \bar{f}^{i,[n+1]}, \quad \forall \xi \in [0,h]. \tag{3.4}$$

Here, we have computed the remainder on wider interval $[0,h]$ than $[0,h)$ form the definition. Such representations are perfectly valid (p,n)-representations, but have usually slightly bigger remainder terms. This will be no issue in the rigorous integration.

Example: *For a function $f(t) \in C_2^1$ such that:*

$$f(t) = \begin{cases} \frac{1}{4} + t^2 & t \in \left[-1, \frac{1}{2}\right) \\ 2 \cdot t^2 & t \in \left[\frac{1}{2}, 0\right] \end{cases} \tag{3.5}$$

We can produce two sample (2,1)-representations $\bar{f}_1$ *and* $\bar{f}_2$ *such that:*

$$\begin{aligned}
\bar{f}_1^{2,[0]} &= \bar{f}_2^{2,[0]} = \left\{\frac{5}{4}\right\} \\
\bar{f}_1^{2,[1]} &= \bar{f}_2^{2,[1]} = \{-1\} \\
\bar{f}_1^{1,[0]} &= \bar{f}_2^{1,[0]} = \left\{\frac{1}{2}\right\} \\
\bar{f}_1^{1,[1]} &= \bar{f}_2^{1,[1]} = \{-1\} \\
\bar{f}_1^{0,[0]} &= \bar{f}_2^{0,[0]} = \{0\}
\end{aligned}$$

and the remainders are

$$\begin{aligned}
\bar{f}_1^{2,[2]} &= \{1\} \\
\bar{f}_1^{1,[2]} &= \{2\}
\end{aligned}$$

and

$$\begin{aligned}
\bar{f}_2^{2,[2]} &= [1,2] \\
\bar{f}_2^{1,[2]} &= \{2\}
\end{aligned}$$

The remainder in $\bar{f}_1$ *is computed on the intervals* $[-i \cdot h, -i \cdot h + h)$*, while in the case of* $\bar{f}_2$ *it is computed on the intervals* $[-i \cdot h, -i \cdot h + h]$*, where* $h = \frac{1}{2}$*,* $i \in \{1, 2\}$*.*

It is clear from Definition 10 that for $t = -i \cdot h + \xi$ where $1 \leq i \leq p$ and $0 \leq \xi < h$ we have the following:

$$f(t) \in \sum_{k=0}^{n+1} \bar{f}^{i,[k]} \cdot \xi^k. \tag{3.6}$$

In other words, a representation of f is simply a collection of (forward) Taylor expansion coefficients of f up to order n computed in equally spaced points $-i/p$, $i \in \{1, .., p\}$, the value of the function at $t = 0$ and the enclosures for the $(n+1)$-st Taylor coefficients of f on the whole intervals $I_i = [-\frac{i-1}{p}, -\frac{i}{p})$. Although the value of the function at $t = 0$ may be derived from the other coefficients (using equation (3.6)), incorporating it as $\bar{x}_0^{f,[0]}$ will be important in the context of the rigorous integrator developed in the next section. The graphical idea of the representation is given in Figure 3.1.

The following definition may be regarded as a definition of the 'best' representation for a given function f:

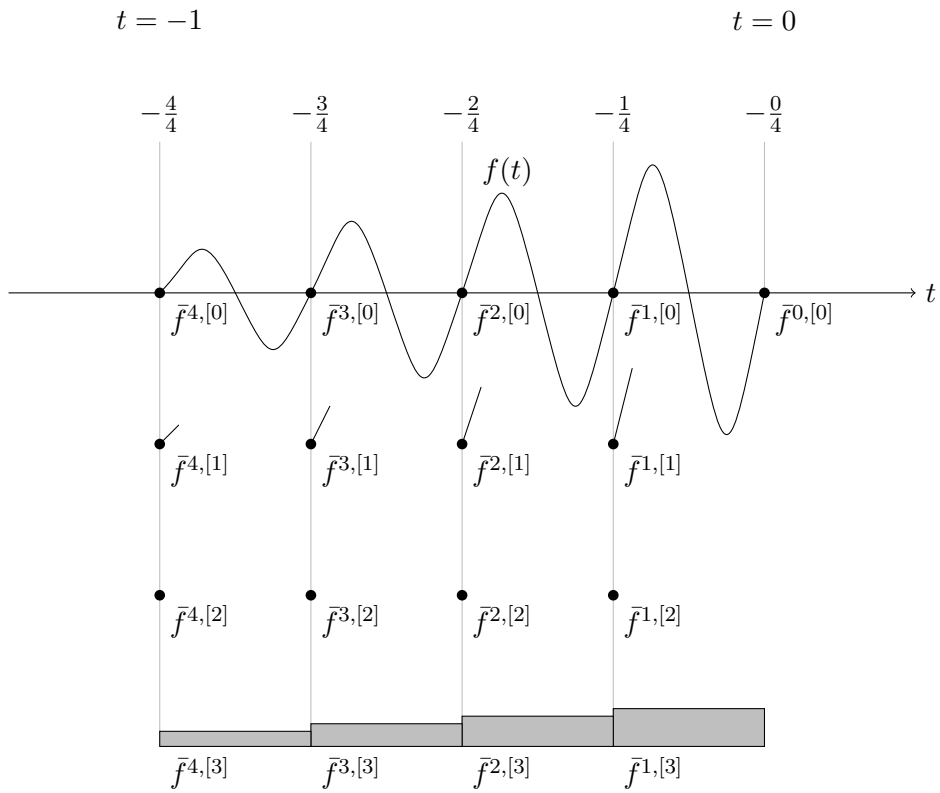

Figure 3.1: The graphical idea of the representation $\bar{f}$ of some function f. The function f is presented as a black, solid, sinusoidal line. In our example $n = 2$ and $p = 4$. Grid points are located at the times $t_i = -\frac{i}{p}$ for $i \in \{0, .., p\}$. The representation $\bar{f}$ consists of coefficients $\bar{f}^{0,[0]}$ and $\bar{f}^{i,[k]}$ for $i \in \{1, ..., p\}$, $k \in \{0, ..., n+1\}$. The coefficient $\bar{f}^{i,[0]}$ is simply a (bound on) value of the function f at point $-\frac{i}{p}$ (this is also true for $\bar{f}^{0,[0]}$), the coefficients $\bar{f}^{i,[1]}$ are (bound on) value of derivative at a given time (presented in picture as line stretching from the point) and so on. We draw a small circle for a given coefficient to stress that this is a (bound on) value at the point. The coefficients $\bar{f}^{i,[n+1]}$ are bounds on the $(n+1)$-st derivative on the whole intervals $\left[-\frac{i}{p}, -\frac{i-1}{p}\right)$ and to stress that we draw them as gray-shaded boxes stretching over the whole length of the appropriate interval.

Definition 11 *We define* the canonical (p,n)-representation *of* $f \in C_p^{n+1}$ *as a minimal (p,n)-representation of* f*, in the sense that for each* $1 \leq i \leq p$ *the following is true:*

- $\bar{f}^{0,[0]} = \{f(0)\}$,
- $\bar{f}^{i,[k]} = \left\{ f^{[k]}\left(-\frac{i}{p}\right) \right\}$ *for* $0 \leq k \leq n$,
- $\bar{f}^{i,[n+1]} = \left\{ f^{[n+1]}\left(-\frac{i}{p} + \xi\right) : 0 \leq \xi \leq h \right\}$.

Remark 7 *The representation described in this chapter is an adaptation of the representation given in [2], where the term* $\bar{f}^{i,[k]}$ *was denoted by* a_i^k *and term* $\bar{f}^{i,[n+1]}$ *was denoted by* b_i.

It is convenient to use notation of the coefficients of (p,n)-representations in the form $\bar{f}^{i,[k]}$, as this clearly link the coefficient to its place and meaning in equation (3.6). However, sometimes we would like to treat collection of coefficients $\bar{x}_k^{f,[i]}$ as a vector in some, M dimensional, euclidean space $\mathbb{R}^M$, for some M depending on p and n. For this, the following definition is natural:

Definition 12 *For given integers* $p > 0$ *and* $n > 0$ *let* $P = \{1, .., p\}$ *and* $N = \{0, .., n+1\}$. *Then we define* (p, n)-index function $\mathbb{I}_{(p,n)} : P \times N \cup \{(0,0)\} \to \mathbb{N}_+$ *as*

$$\mathbb{I}_{(p,n)}(0,0) := 1, \tag{3.7}$$
$$\mathbb{I}_{(p,n)}(i,k) := ((i-1)\cdot(n+2)+k)+2, \tag{3.8}$$

If n and p are known from context we will omit subscripts and we will call $\mathbb{I}$ simply the *index function*. From Definition 12 of $\mathbb{I}_{(p,n)}$ it is clear that:

Lemma 8 $\mathbb{I}_{(p,n)}$ *is an injection, and* $Dom(\mathbb{I}_{(p,n)}) = \{1, .., p\cdot(n+2)+1\}$.

■

From now one, using the index function, we can think of a representation $\bar{f}$ as a subset of $\mathbb{R}^M$, where $M = p\cdot(n+2)+1$. Namely, we can construct a set of vectors $\bar{\mathbf{f}} \subset \mathbb{R}^M$ such that $\pi_{\mathbb{I}(i,k)}\bar{\mathbf{f}} = \bar{f}^{i,[k]}$. We will use the same symbol $\bar{f}$ to describe both $\bar{\mathbf{f}}$ and $\bar{f}$, as it will be clear from the context which one we are using. Index function will allow us to use standard techniques for representing interval sets in computations, most notable the Lohner sets from CAPD library. Also we can construct inclusion relation between representations by $\bar{f} \subset \bar{g} \iff \bar{\mathbf{f}} \subset \bar{\mathbf{g}}$. We will write $x \in \bar{f}$ to indicate a vector $x \in \mathbb{R}^{(n+2)\cdot p+1}$ such that $x \in \bar{\mathbf{f}}$.

The representation of a given function f usually can describe more than a single function f. Therefore we define:

Definition 13 The support of (p,n)-representation $\bar{f}$ *is a set:*

$$Supp(\bar{f}) := \left\{ g \in C_p^{n+1} : \bar{f} \text{ is a (p,n)-representation of } g \right\} \tag{3.9}$$

By Definition 13 we see that $\bar{f}$ is a (p,n)-representation of each $g \in Supp(\bar{f})$. We also know that the set $Supp(\bar{f})$ is contained in C_p^{n+1}. Sometimes however, for a given representation $\bar{f}$, we will need to work only on a subset of functions in $Supp(\bar{f})$ of some continuity class C^k on the whole interval $[-1, 0]$. Therefore we define:

Definition 14 The C^k-support of (p,n)-representation $\bar{f}$, $k \in \{0, 1, 2, ..., \infty\}$, *is a set:*

$$Supp^{(k)}(\bar{f}) := Supp(\bar{f}) \cap C^k \tag{3.10}$$

As we seen in example in Remark 4, not all functions in $Supp(\bar{f})$ are continuous. Therefore it is reasonable to state the following:

Definition 15 *We say that* $\bar{f}$ *is* C^k-admissible *iff* $Supp^{(k)}(\bar{f}) \neq \emptyset$.

The notion of C^k-admissible (p,n)-representations will be used in Chapter 5.

As a last thing in this section we would like to show a procedure that, given a (p,n)-representation $\bar{f}$ of some function f, will return a rigorous bounds for each derivative $f^{[k]}(t)$ for each $t \in [-1, 0]$. This procedure will be an essential component of the rigorous integrator presented in the next section. Using equation (3.6) we can prove that:

Lemma 9 *Assume* $f \in C_p^{n+1}$ *and its (p,n)-representation* $\bar{f}$ *are given. Let define*

$$c^{i,[k]}(\varepsilon) = \sum_{l=k}^{n+1} \binom{l}{k} \cdot \varepsilon^{l-k} \cdot \bar{f}^{i,[l]}, \tag{3.11}$$

for $0 \leq \varepsilon < \frac{1}{p}$. *Then*

$$f^{[k]}\left(-\frac{i}{p} + \varepsilon\right) \in c^{i,[k]}(\varepsilon). \tag{3.12}$$

Proof: Let first look at the Taylor expansion of $f\left(-\frac{i}{p} + \varepsilon\right)$:

$$f\left(-\frac{i}{p} + \varepsilon\right) = f\left(-\frac{i}{p}\right) + \sum_{l=1}^{n} \frac{\varepsilon^l}{l!} \cdot f^{(l)}\left(-\frac{i}{p}\right) + \frac{\varepsilon^{n+1}}{(n+1)!} \cdot f^{(n+1)}\left(-\frac{i}{p} + \xi_0\right), \tag{3.13}$$

where $\xi_0 = \xi_0(\varepsilon) \in [0, \varepsilon]$. Using Definition 10 of the representation we get the following:

$$f^{[0]}\left(-\frac{i}{p} + \varepsilon\right) = f\left(-\frac{i}{p} + \varepsilon\right) \in \sum_{l=0}^{n+1} \varepsilon^l \cdot \bar{f}^{i,[l]} = c^{i,[0]}(\varepsilon). \tag{3.14}$$

Now, similarly, we can express each $f^{(k)}\left(-\frac{i}{p}+\varepsilon\right)$ by:

$$\begin{aligned} f^{(k)}\left(-\frac{i}{p}+\varepsilon\right) &= \sum_{l=k}^{n} \varepsilon^{l-k} \cdot \frac{f^{(l)}\left(-\frac{i}{p}\right)}{(l-k)!} + \\ &+ \varepsilon^{n+1-k} \cdot \frac{x^{(n+1)}\left(-\frac{i}{p}+\xi_k\right)}{(n+1-k)!}, \end{aligned}$$

for some $\xi_k = \xi_k(\varepsilon) \in [0, \varepsilon]$. Dividing both sides by $k!$ and setting $f^{(l)} = l! \cdot f^{[l]}$ we get the following:

$$f^{[k]}\left(-\frac{i}{p}+\varepsilon\right) \in \sum_{l=k}^{n+1} \frac{l!}{k! \cdot (l-k)!} \cdot \varepsilon^{l-k} \cdot \bar{f}^{i,[l]} = c^{i,[k]}(\varepsilon). \tag{3.15}$$

■

A single coefficient $c^{i,[k]}(\varepsilon)$ for a given k may be computed using an iterative procedure presented in the algorithm 1. We usually would like to compute all coefficients at once. For

Algorithm 1 compute-c-k

Input: $\bar{f}$, i, k, ε

Output: $c^{i,[k]}(\varepsilon)$

Require: $\bar{f}$ is a (p,n)-representation, $1 \le i \le p$, $0 \le k \le n+1$, $0 \le \varepsilon \le \frac{1}{p}$

1: $\hat{c}^{i,[k]}_{n+1} \leftarrow \bar{f}^{i,[n+1]}$
2: **for** $l = n, n-1, .., k$ **do**
3: $\quad \hat{c}^{i,[k]}_{l} \leftarrow \bar{f}^{i,[l]} + \varepsilon \cdot \frac{l+1}{l+1-k} \cdot \hat{c}^{i,[k]}_{l+1}$
4: **end for**
5: $c^{i,[k]}(\varepsilon) \leftarrow \hat{c}^{i,[k]}_{k}$

this we introduce Algorithm 2.

This ends the preparative this section, where we have described in details the (p,n)-representations of (sufficiently) smooth functions and discussed some properties of those representations. Now we are ready to describe rigorous integration method for DDEs of the form (3.1).

3.2 Integrator

Let assume that we are at time $t = 0$ and we have some a priori representation $\bar{x}_0$ of the initial function x_0. We denote by $x = x(t)$ the solution of (3.1) with initial condition x_0.

Algorithm 2 compute-c

Input: $\bar{f}$, i, ε

Output: $\left\{c^{i,[k]}(\varepsilon)\right\}_{k\in\{0,..,n+1\}}$

Require: $\bar{f}$ is a (p,n)-representation, $1 \leq i \leq p$, $0 \leq \varepsilon \leq \frac{1}{p}$

1: **for** $k : 1 \leq k \leq n$ **do**

2: $\quad c^{i,[k]}(\varepsilon) \leftarrow$ COMPUTE-C-K($\bar{f}, i, k, \varepsilon$)

3: **end for**

The goal: we want to compute $\bar{x}_h$ - the representation of x_h. For this we will use Taylor-type integration algorithm.

The goal which we want to achieve is graphically presented in Figure 3.2. We can see that most of the coefficients overlap in the new representation at $t = h$ with those in the representation at $t = 0$. The procedure of moving them from x_0 to the new representation x_h is called *the shift part*. The procedure of computation of new elements (empty circles and an empty rectangle in Figure 3.2) is called *the forward part*. This part is more involved than the shift part so we will divide it into three separate algorithms, which will be executed in the following order:

- computing coefficients $\bar{x}_h^{1,[k]}$ for $k \in \{1, .., n\}$,
- computing the remainder $\bar{x}_h^{1,[n+1]}$,
- computing the value of x_h at $t = 0$ (stored in $\bar{x}_h^{0,[0]}$).

In the forward part we will use the r.h.s f of equation (3.1) and its derivatives w.r.t. time. As in [33] we will use an automatic method for obtaining $x^{(k)}(t)$ using equation (3.1). The details may be found in [18, 16]. From equation (3.1) we have that:

$$x^{(k)}(t) \quad = \quad \frac{d^{k-1}}{dt^{k-1}} f(x(t-1), x(t)). \tag{3.16}$$

Let $z = (z_1, z_2)$, $z_1, z_2 \in \mathbb{R}$ and $f(z) = f(z_1, z_2)$. For $k = 1, 2$ and $z(t) = (x(t-1), x(t))$ we get:

$$x^{(1)}(t) \quad = \quad f(z(t)) \quad = \quad f(x(t-1), x(t)), \tag{3.17}$$

$$x^{(2)}(t) \quad = \quad \frac{d}{dt} f\left(z(t)\right) \quad = \tag{3.18}$$

$$= \quad \frac{\partial f}{\partial z}\left(z(t)\right) \cdot \frac{\partial z}{\partial t}(t) \tag{3.19}$$

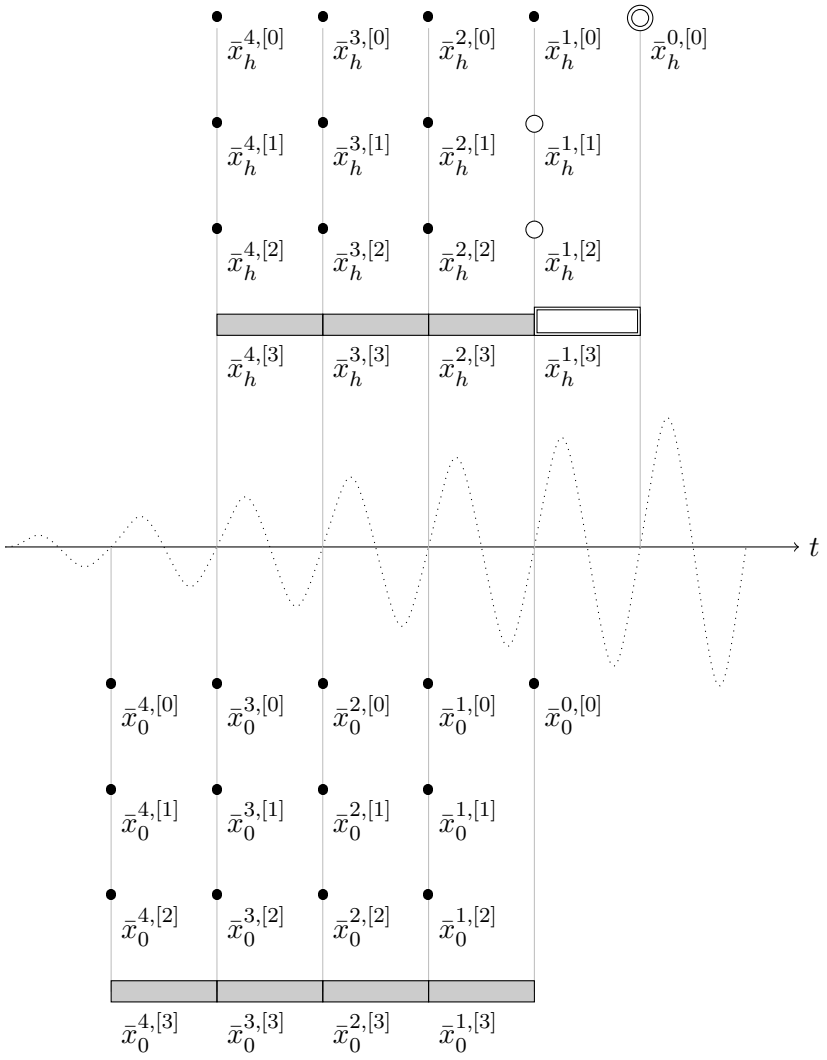

Figure 3.2: A graphical presentation of the integrator scheme. On top we see a representation after one step of size $h = \frac{1}{p}$. Black solid dots and gray rectangles represent the values we need not to compute - this is *the shift part*. The elements that need to be constructed are presented as empty dots and an empty rectangle - this is *the forward part*. The doubly bordered dot is a (bound on) value of the function at the time $t = h = \frac{1}{p}$. The doubly bordered empty rectangle is an enclosure for the $n+1$-st derivative on the whole interval $\left[0, \frac{1}{p}\right]$.

Where $\cdot$ in equation (3.19) is a scalar product. Thus we get:

$$x^{(2)}(t) = \frac{\partial f}{\partial z_1} f\left(x(t-1), x(t)\right) \cdot x^{(1)}(-1) + \tag{3.20}$$

$$+ \frac{\partial f}{\partial z_2}\left(x(t-1), x(t)\right) \cdot x^{(1)}(0) \tag{3.21}$$

We see that the second derivative with respect to t of $x(t)$ is a function of four variables: $x(t-1)$, $x^{(1)}(t-1)$, $x(t)$ and $x^{(1)}(t)$. Thus, there exist function $F^{(1)} : \mathbb{R}^4 \to \mathbb{R}$ such that:

$$F^{(1)}\left(x(t-1), x^{(1)}(t-1), x(t), x^{(1)}(t)\right) = x^{(2)}(t). \tag{3.22}$$

This procedure can be applied recursively to obtain a family $\left\{F^{(k)} : \mathbb{R}^{2\cdot(k+1)} \to \mathbb{R}\right\}_{k\in\mathbb{N}}$ of functions such that:

$$x^{(k+1)}(t) = F^{(k)}\left(x(-1+t), .., x^{(k)}(-1+t), x(t), .., x^{(k)}(t)\right). \tag{3.23}$$

where

$$F^{(0)}\left(z_1, z_2\right) = f(z_1, z_2). \tag{3.24}$$

We also introduce the following notation for the simplicity of the implementation of the algorithms:

$$F^{[k]}(z_1, .., z_{2\cdot(k+1)}) = \frac{1}{k!} F^{(k)}\left(0!\cdot z_1, .., k!\cdot z_{k+1}, 0!\cdot z_{k+2}, .., k!\cdot z_{2\cdot(k+1)}\right), \tag{3.25}$$

for which we have:

$$x^{[k+1]}(t) = \frac{1}{k+1} \cdot F^{[k]}\left(x^{[0]}(-1+t), .., x^{[k]}(-1+t), x^{[0]}(t), .., x^{[k]}(t)\right). \tag{3.26}$$

This notation is important in the context of computer rigorous numerics as the functions $F^{[k]}$ can be efficiently computed using the automatic differentiation (AD) algorithm [18, 16], provided we know the bounds on the Taylor expansion of x in the t variable up to order k.

Now we are ready to describe all subroutines that will be used in the integrator.

3.2.1 The shift part

The shift part is simply a direct substitution, realized by Algorithm 3. We assume that the shift part is always executed before the forward part, as the forward part will use the coefficient $\bar{x}_h^{1,[0]}$ in all subsequent computations.

Algorithm 3 shift

Input: $\bar{x}_0$

Output: $\bar{x}_h^{i,[k]}$ for $2 \leq i \leq p$ and $0 \leq k \leq n+1$

1: **for** $i, k : 2 \leq i \leq p$ and $0 \leq k \leq n+1$ **do**
2: $\quad \bar{x}_h^{i,[k]} \leftarrow \bar{x}_0^{i-1,[k]}$
3: **end for**
4: $\bar{x}_h^{1,[0]} \leftarrow \bar{x}_0^{0,[0]}$

3.2.2 The forward part - computing $\bar{x}_h^{1,[k]}$

The coefficient $\bar{x}_h^{1,[0]}$ is already present in the representation $\bar{x}_h$ after execution of the shift part. For $k > 0$ we get the following:

Lemma 10 *For $0 < k \leq n$ let define:*

$$\bar{x}_h^{1,[k]} = \frac{1}{k} \cdot F^{[k-1]} \left(\bar{x}_0^{p,[0]}, .., \bar{x}_0^{p,[k-1]}, \bar{x}_h^{1,[0]}, .., \bar{x}_h^{1,[k-1]} \right). \tag{3.27}$$

Then:

$$x^{[k]}(0) \in \bar{x}_h^{0,[k]} \tag{3.28}$$

Proof: Using equations (3.16), (3.25) and (3.26) we get for $1 \leq k \leq n$:

$$\begin{aligned} x^{[k]}(0) &= \frac{1}{k!} \cdot x^{(k)}(0) = \\ &= \frac{1}{k!} \cdot \frac{d^{k-1}}{dt^{k-1}} f\left(x(-1), x(0)\right) = \\ &= \frac{1}{k} \cdot \frac{1}{(k-1)!} \cdot F^{(k-1)} \left(x(-1), .., x^{(k-1)}(-1), x(0), .., x^{(k-1)}(0) \right) \\ &\in \frac{1}{k} \cdot F^{[k-1]} \left(\bar{x}_0^{p,[0]}, .., \bar{x}_0^{p,[k-1]}, \bar{x}_h^{1,[0]}, .., \bar{x}_h^{1,[k-1]} \right) \end{aligned}$$

■

Notice that, in Algorithm 4, we need to apply (3.27) iteratively, as all coefficients $\bar{x}_h^{1,[l]}$ for $l < k$ are needed for the computation of $\bar{x}_h^{1,[k]}$. Elements needed for computation of $\bar{x}^{1,[k]}$ for $1 \leq k \leq n$ are presented in Figure 3.3. The procedure itself is presented in Algorithm 4 and accepts a rather generic list of parameters, as we will use it in many applications thorough the text. We want to point out that for the computation of $\bar{x}_h^{1,[k]}$ for $1 \leq k \leq n$ given a representation $\bar{x}_0$ and the value $\bar{x}_h^{0,[0]}$ we need to call the procedure with the following list of arguments: $n-1$, $\bar{x}_h^{p,[0]}, ..., \bar{x}_h^{p,[n-1]}$, $\bar{x}_h^{0,[0]}$.

Algorithm 4 compute-rep-k

Input: n, $\left\{x^{[k]}(t-1)\right\}_{0\leq k\leq n}$, $x(t)=x^{[0]}(t)$

Output: $\left\{x^{[k]}(t)\right\}_{1\leq k\leq n+1}$

1: **for** $k:1\leq k\leq n+1$ **do**

2: $\quad x^{[k]}(t)\leftarrow\frac{1}{k}\cdot F^{[k-1]}\left(x^{[0]}(t-1),..,x^{[k-1]}(t-1),x^{[0]}(t),..,x^{[k-1]}(t)\right)$

3: **end for**

3.2.3 The forward part - computing remainder

Computing an enclosure for the remainder on the interval $\left[0,\frac{1}{p}\right]$ is the most involved part of the integration scheme.

From the mean value theorem we have:

$$\begin{aligned}\frac{1}{(n+1)!}\cdot x^{(n+1)}(\varepsilon) &= \frac{1}{(n+1)!}\cdot x^{(n+1)}(0)+\frac{1}{(n+1)!}\cdot x^{(n+2)}(\xi)\cdot\varepsilon= \\ &= \frac{1}{(n+1)}\cdot F^{[n]}\left(x(-1),..,x^{[n]}(-1),x(0),..,x^{[n]}(0),\right)\ + \\ &+ \frac{1}{(n+1)!}\cdot F^{[n+1]}\left(x(-1+\xi),..,x^{[n+1]}(-1+\xi),x(\xi),..,x^{[n+1]}(\xi)\right)\cdot\varepsilon\end{aligned}\tag{3.29}$$

for $0\leq\xi\leq\varepsilon\leq\frac{1}{p}$. From Definition 10 of the representation and from Lemma 9 we have for $0\leq k\leq n+1$:

$$x^{[k]}(-1+\xi) \in c^{p,[k]}_{\bar{x}_0}([0,h]) \tag{3.30}$$

$$x^{[k]}(-1) \in \bar{x}^{p,[k]}_0 \tag{3.31}$$

$$x^{[k]}(0) \in \bar{x}^{1,[k]}_h \tag{3.32}$$

The only things left to estimate are the values of $x^{[k]}(\xi)$ for $\xi\in\left[0,\frac{1}{p}\right]$.

Lemma 11 *Let $x(t)$ be a solution of (3.1) on the interval $\left[-1,\frac{1}{p}\right]$. Assume that $Y\subset\mathbb{R}$ is a set such that*

$$Z=x(0)+\left[0,\frac{1}{p}\right]\cdot f\left(x\left(\left[-1,-1+\frac{1}{p}\right]\right),Y\right)\subset int(Y),\tag{3.33}$$

Then

$$x\left(\left[0,\frac{1}{p}\right]\right)\subset Z\tag{3.34}$$

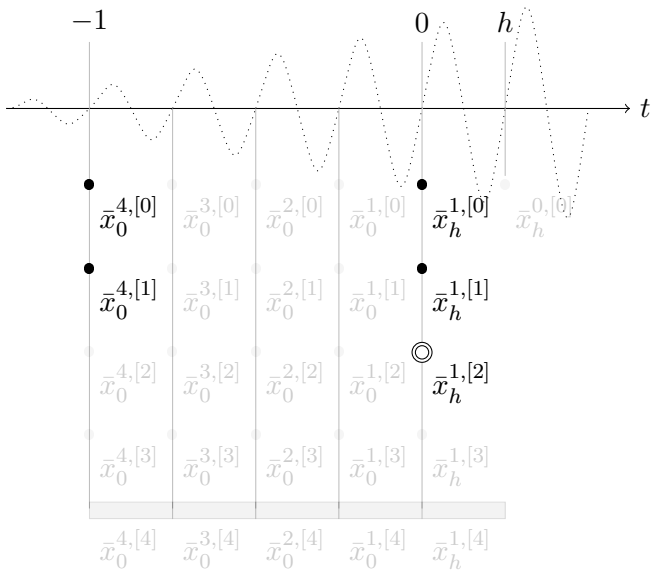

Figure 3.3: Elements needed to compute $\bar{x}_h^{1,[k]}$ (empty circle) for $k = 2$ are presented as bold, dark circles.

Proof: We can treat equation (3.1) on the interval $\left[0, \frac{1}{p}\right]$ as a non-autonomous ODE of the form:

$$x' = f(a(t), x), \tag{3.35}$$

where $a(t) = x(t)$ for $t \in [-1, -1+h]$ is a known function. Now the conclusion follows from the proof of the analogous theorem for ODEs. The proof can be found in [33, 35]. ∎

Definition 16 *The set Z in Lemma 11 is called a* rough enclosure *of x on the interval $\left[0, \frac{1}{p}\right]$.*

Remark 12 *A heuristic algorithm for finding the rough enclosure Z is described elsewhere [33, 35]. In our computations we use a standard procedure implemented in the CAPD library.*

Having rough enclosure Z we set $d^{[0]} = Z$ and we compute $d^{[k]}$ for $k > 0$ using equation (3.27). This procedure is summarized in Algorithm 5.

Now we can state the following lemma:

Lemma 13 *Let $\bar{x}_h^{1,[n+1]}$ be an output from Algorithm 5 executed for the representation $\bar{x}_0$ and $\bar{x}_h^{1,[0]}, .., \bar{x}_h^{1,[n]}$ such that $x^{[k]}(0) \in \bar{x}_h^{1,[k]}$. Then $x^{[n+1]}([0, h]) \subset \bar{x}_h^{1,[n+1]}$.*

Algorithm 5 compute-remainder

Input: $\bar{x}_0, \bar{x}_h^{1,[0]}, .., \bar{x}_h^{1,[n]}$

Output: $\bar{x}_h^{1,[n+1]}$, a^*, b^*

1: $\left\{c^{[k]}\right\}_{0\leq k\leq n+1} \leftarrow$ COMPUTE-C($\bar{x}_0^{p,[0]}, \bar{x}_0^{p,[n+1]}, [0,h]$)

2: $d^{[0]} \leftarrow$ ROUGH-ENCLOSURE($f(c^{[0]}, \cdot), \bar{x}_0^{0,[0]}, \frac{1}{p}$)

3: $\left\{d^{[k]}\right\}_{1\leq k\leq n+1} \leftarrow$ COMPUTE-REP-K($n, c^{[0]}, ..., c^{[n]}, d^{[0]}, [0,h]$)

4: $a^* \leftarrow \frac{1}{(n+1)} \cdot F^{[n]}(\bar{x}_0^{p,[0]}, .., \bar{x}_0^{p,[n]}, \bar{x}_h^{1,[0]}, .., \bar{x}_h^{1,[n]})$

5: $b^* \leftarrow F^{[n+1]}(c^{[0]}, .., c^{[n]}, c^{[n+1]}, d^{[0]}, .., d^{[n]}, d^{[n+1]})$

6: $\bar{x}_h^{1,[n+1]} \leftarrow a^* + [0,h] \cdot b^*$

Proof: The proof follows from the Lemmas 11 and 9 and from equation (3.29). ∎

Elements needed to compute the remainder are presented in Figure 3.4.

3.2.4 The forward part - computing value of x at $t = h$

Now, having all the coefficients and the remainder at $t = 0$, we can simply use Definition 10 of the representation and equation (3.6) to compute $\bar{x}_h^{0,[0]}$. Namely, for $h = \frac{1}{p}$ we have:

$$x(h) \in \bar{x}_h^{0,[0]} := \sum_{k=0}^{n+1} \bar{x}_h^{1,[k]} \cdot h^k \tag{3.36}$$

The elements needed to compute $\bar{x}_{t+h}^{0,[0]}$ are presented in Figure 3.5.

3.2.5 Altogether

In Algorithm 6 we present a full list of steps needed for one step of the rigorous integration. The graphical representation of the integrator is presented in Figure 3.2.

For simplicity, we will use $\Phi(\cdot)$ in the text of this manuscript instead of the name of Algorithm 6: `compute-Phi`. That is, we will write $\bar{x}_{t+h} = \Phi(\bar{x}_t)$, having in mind that Φ is computed using the numerical Algorithm 6.

Remark 14 *By equation* (2.5) *defining* φ *and from Algorithm 6 we can write* $\varphi(x_t, h) \in \Phi(\bar{x}_t) = \bar{x}_{t+h}$.

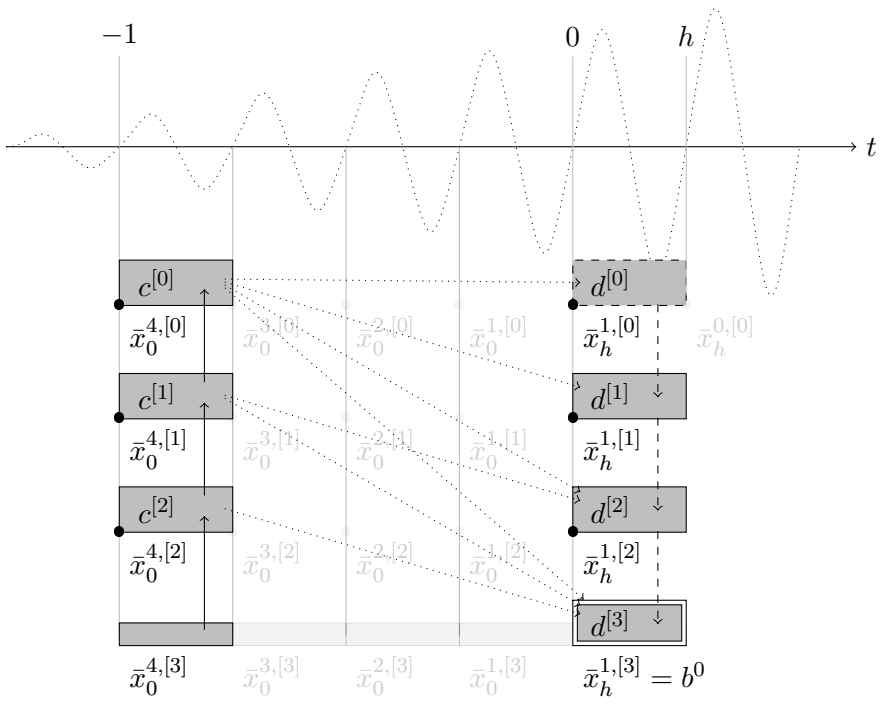

Figure 3.4: Elements needed to compute the remainder $\bar{x}_h^{1,[n+1]}$ (a rectangle with double black-white border) are presented as gray, dark rectangles and black dots. The enclosures $c^{[k]}$ over the interval $[-1, -1+h]$ are computed from the representation, using bottom-up procedure (solid arrows) for better performance. For $t = 0$ we need to use rough enclosure (a dashed rectangle) and then compute top-bottom (as pointed by dashed arrows). Dotted lines connects elements $c^{[\cdot]}$ needed in computation of a given element $d^{[k]}$ (remember that all $d^{[k']}$ for $k' < k$ are needed also). Notice, that we also need to compute $d^{[n+1]}$ which is here presented as gray rectangle inside $\bar{x}_h^{1,[n+1]}$. Having all those coefficients we can finally compute the remainder $\bar{x}_h^{1,[n+1]}$ (a white rectangle).

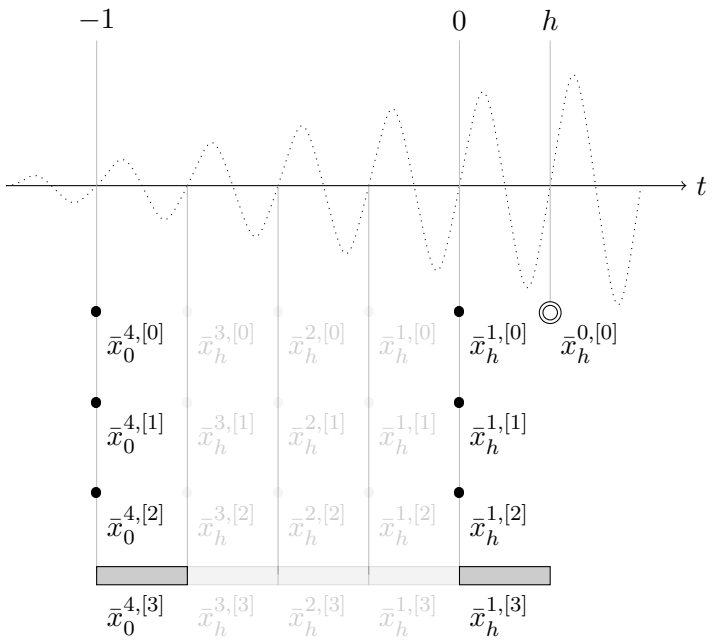

Figure 3.5: Elements needed to compute $\bar{x}_{t+h}^{0,[0]}$ (an empty circle) are presented as bold, dark circles and rectangles.

Algorithm 6 compute-Phi

Input: $\bar{x}_t$

Output: $\bar{x}_{t+h}$

1: $\{\bar{x}^{i,[k]}_{t+h}\}_{(i,k)\in\{(i,k):2\leq i\leq p, 0\leq k\leq n+1\}\cup\{(1,0)\}} \leftarrow$ SHIFT-PART($\bar{x}_t$)

2: $\left\{\bar{x}^{1,[k]}_{t+h}\right\}_{1\leq k\leq n} \leftarrow$ COMPUTE-REP-K($n-1, \bar{x}^{p,[0]}_t, .., \bar{x}^{p,[n-1]}_t, \bar{x}^{1,[0]}_{t+h}$)

3: $\{\bar{x}^{1,[n+1]}_{t+h}, a^*, b^*\} \leftarrow$ COMPUTE-REMAINDER($\bar{x}_t, \bar{x}^{1,[0]}_{t+h}, .., \bar{x}^{1,[n]}_{t+h}$)

4: $\bar{x}^{0,[0]}_{t+h} \leftarrow \sum_{k=0}^{n+1} \bar{x}^{1,[k]}_{t+h} \cdot h^k$

3.2.6 Expanding the representation

As it was pointed in Chapter 1 the imminent property of DDEs is the smoothing of solutions. If the r.h.s of equation (3.1) is smooth (i.e. of class C^∞) then the solution $x(t)$ gets smoother and smoother on each interval $[n, n+1]$, $n \in \mathbb{N}$. We can exploit this property in our integrator. For this purpose we return in procedure `compute-remainder` coefficients a^* and b^* separately to be able to remember them for future use. After p steps we can use this history to expand (p, n)-representation to (p, n+1)-representation as presented in the algorithm 7.

3.3 Numerical experiments

Here we discuss numerical experiments done for the integrator presented in this Chapter operating on the interval representation (i.e. the product of closed intervals) of the (p,n)-representation. We use two test cases:

1. a stable stationary solution $x \equiv 0$ for the system (2.14).
2. a stable periodic solution for the system (2.15). Its numerical approximation is presented in Figure 2.1.

Since the presentation of the exact output (lower and upper bounds) of the (p,n)-representation may be difficult, even for quite small (but reasonable) parameters p and n we have decided to present here only the diameters of the resulting set, which capture the contracting/expanding nature of the integration process. The full data and more figures are available on the authors home page[22].

Algorithm 7 expand-rep

Input: $\bar{x}_1, \{a_i^*\}_{i\in\{1,..,p\}}, \{b_i^*\}_{i\in\{1,..,p\}}$

Output: $\bar{y}_1$

Require: $\bar{x}_1$ is a (p,n)-representation of x_1, $\bar{x}_1 = \Phi^p(x_0)$, a_i^*, b_i^* are coefficients remembered in i-th step of computing $\Phi^p(\bar{x}_0)$.

Ensure: $\bar{y}_1$ is a (p,n+1)-representation of x_1

1: $\bar{y}_1^{0,[0]} \leftarrow \bar{x}_1^{0,[0]}$
2: **for** $k : 1 \le k \le n, i : 1 \le i \le p$ **do**
3: $\quad \bar{y}_1^{i,[k]} \leftarrow \bar{x}_1^{i,[k]}$
4: **end for**
5: **for** $i : 1 \le i \le p$ **do**
6: $\quad \bar{y}_1^{i,[n+1]} \leftarrow a_{p+1-i}^*$
7: $\quad \bar{y}_1^{i,[n+2]} \leftarrow b_{p+1-i}^*$
8: **end for**

For the stationary solution we have prepared an initial (p,n)-representations $\bar{x}_0$ such that:

$$\bar{x}_0^{0,[0]} = [-a, a] \tag{3.37}$$
$$\bar{x}_0^{i,[k]} = [-a, a], \quad i \in \{1, .., p\}, k \in \{0, .., n\} \tag{3.38}$$
$$\bar{x}_0^{i,[n+1]} = 10 \cdot [-a, a], , \quad i \in \{1, .., p\}. \tag{3.39}$$
$$\tag{3.40}$$

We have created five test cases, for three of them we have prepared variants with modified parameter a:

- Tests 1a, 1b, 1c: $p = 8$, $n = 7$, $a \in \{0.4, 0.2, 0.1\}$ respectively,
- Tests 2a, 2b, 2c: $p = 8$, $n = 9$, $a \in \{0.4, 0.2, 0.1\}$ respectively,
- Tests 3a, 3b, 3c: $p = 8$, $n = 11$, $a \in \{0.4, 0.2, 0.1\}$ respectively,
- Tests 4: $p = 16$, $n = 7$, $a = 0.1$,
- Tests 5: $p = 32$, $n = 7$, $a = 0.1$.

We have run an iteration process for each test for N $= p \cdot (n+2)$ steps to guarantee that the resulting representation contains solutions of class C^{n+2} - this will be important later. Then we have compared the diameters of the resulting (p, n)-representation, at each step $i \in \{0, ..., N\}$. Because of the rapid contraction in the first $2 \cdot p$ steps of the integration, we have decided to present in figures only the history after that step, to be able to see the contraction effect. Full history (after $2 \cdot p$ initial steps) of the integration process is presented in Figures 3.6, 3.8 and 3.10, and the history recorded every p steps (after $2 \cdot p$ initial steps) is presented in Figures 3.7, 3.9 and 3.11 for tests $1c, 2c, 3c$ respectively. Also we present the solution together with all derivatives up to order $k = 8$ over the whole time interval $t = [p, p \cdot (n+1)]$ in Figure 3.12. The dependence of the resulting set diameter on the diameter of the initial set in test 3 is presented in Figure 3.13. The influence of the diameter of the resulting set on the choice of the parameter p (the grid spacing) is presented in Figure 3.14, where we have used tests 1c, 4 and 5, and we have compared diameters of the corresponding representation coefficients. All other comparison charts may be found on the author's web page.

As we can see from Figures 3.6, 3.8 and 3.10 we get very good contraction along the whole history of the integration for this stationary solution. What is important, we get the strong contraction on the remainder part- it is what we required from our integrator in the statement of the problem in Section 2.2. The sudden drops in some coefficients that can be seen in the figures corresponds to the times $t \in \{1, 2, 3, ...\}$ where the solution get smoothed. We cannot now explain, why this smoothing property of the DDEs generates such a behaviour in the integrator - we will investigate this in our future work.

The dependence of the initial set diameter is also very good, as presented in Figure 3.13. We see that changing parameter a to $\frac{a}{2}$ results in decreasing the resulting set diameter two or more times. This is because we also decreased the diameter of the remainder which seems to play dominant role in the resulting set diameter.

The dependence of the diameter on the step size (grid size) is less visible. In Figure 3.14 we see that increasing grid spacing from p to $2 \cdot p$ gives slightly better results for coefficients $\bar{x}^{i,[k]}$ for $k \leq n$, but surprisingly it can produce bigger remainder coefficients.

Nevertheless, the integrator seems to be working very well in the case of the stationary solutions.

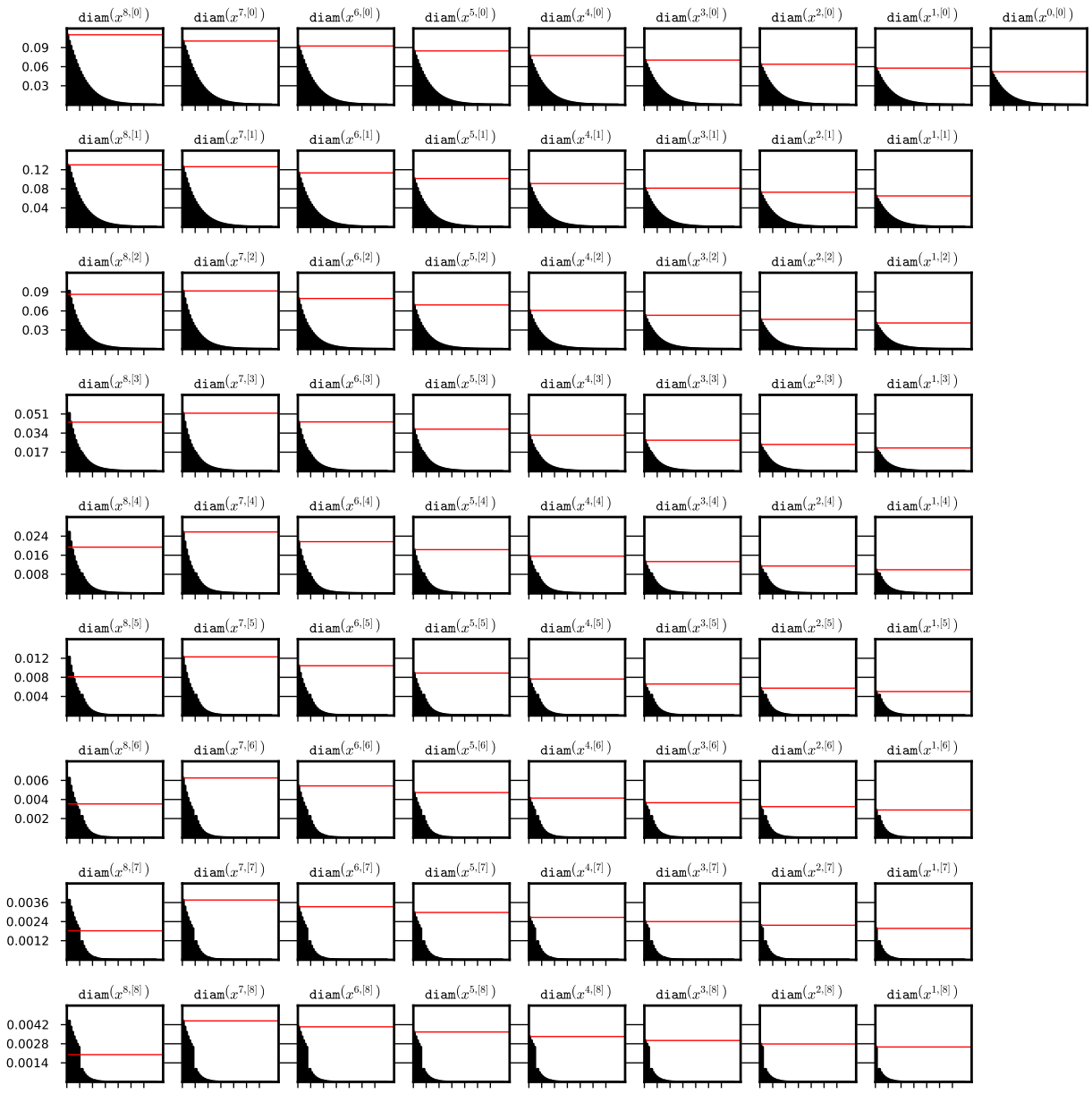

Figure 3.6: Diameters of the coefficients of a sequence $\{\Phi^n(\bar{x}_0)\}_{n\in\{16,..,8\cdot(7+2)\}}$ (a full history after $2 \cdot p$ steps) for some (8,7)-representation $\bar{x}_0$ of a stable stationary solution $x \equiv 0$ for system (2.14). Red horizontal line marks the diameter of the representation of the initial function. On the x-axis we have the iteration steps, each tick represents p steps of iteration. The data from test 1c was used. System (2.14), interval set representation and (8,7)-representation were used for the integration process. The data is stored in the file `steady_08_07_out_3/int_di.txt`.

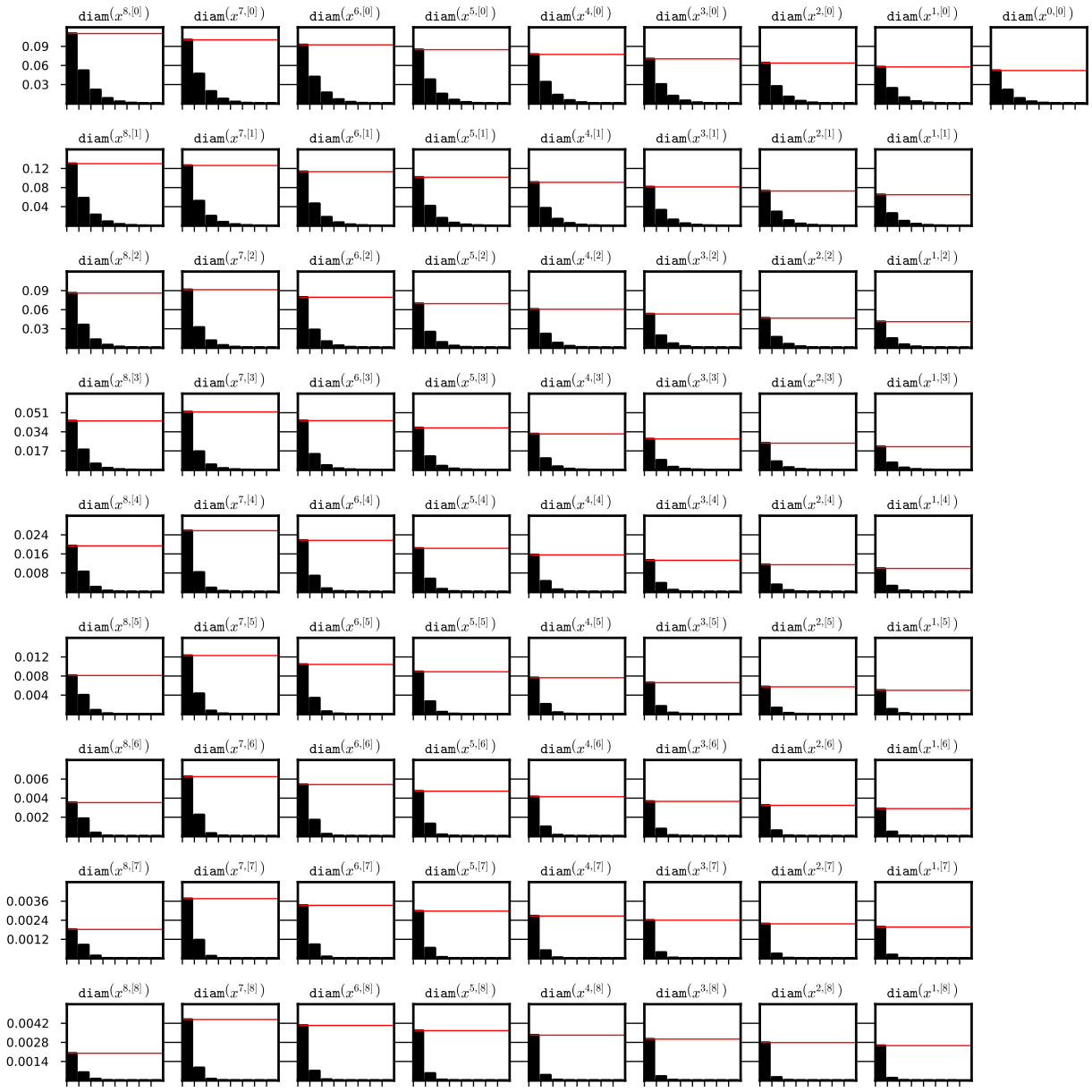

Figure 3.7: Diameters of the coefficients of a sequence $\left\{\Phi^{8\cdot n}(\bar{x}_0)\right\}_{n\in\{2,..,(7+2)\}}$ for some (8,7)-representation $\bar{x}_0$ of a stable stationary solution $x \equiv 0$ for system (2.14). Red horizontal line marks the diameter of the representation of the initial function after $2 \cdot p$ steps. On the x-axis we have the iteration steps, each tick represents p steps of iteration. The data from test 1c was used. System (2.14), interval set representation and (8,7)-representation were used for the integration process. The data is stored in the file `steady_08_07_out_3/int_di.txt`.

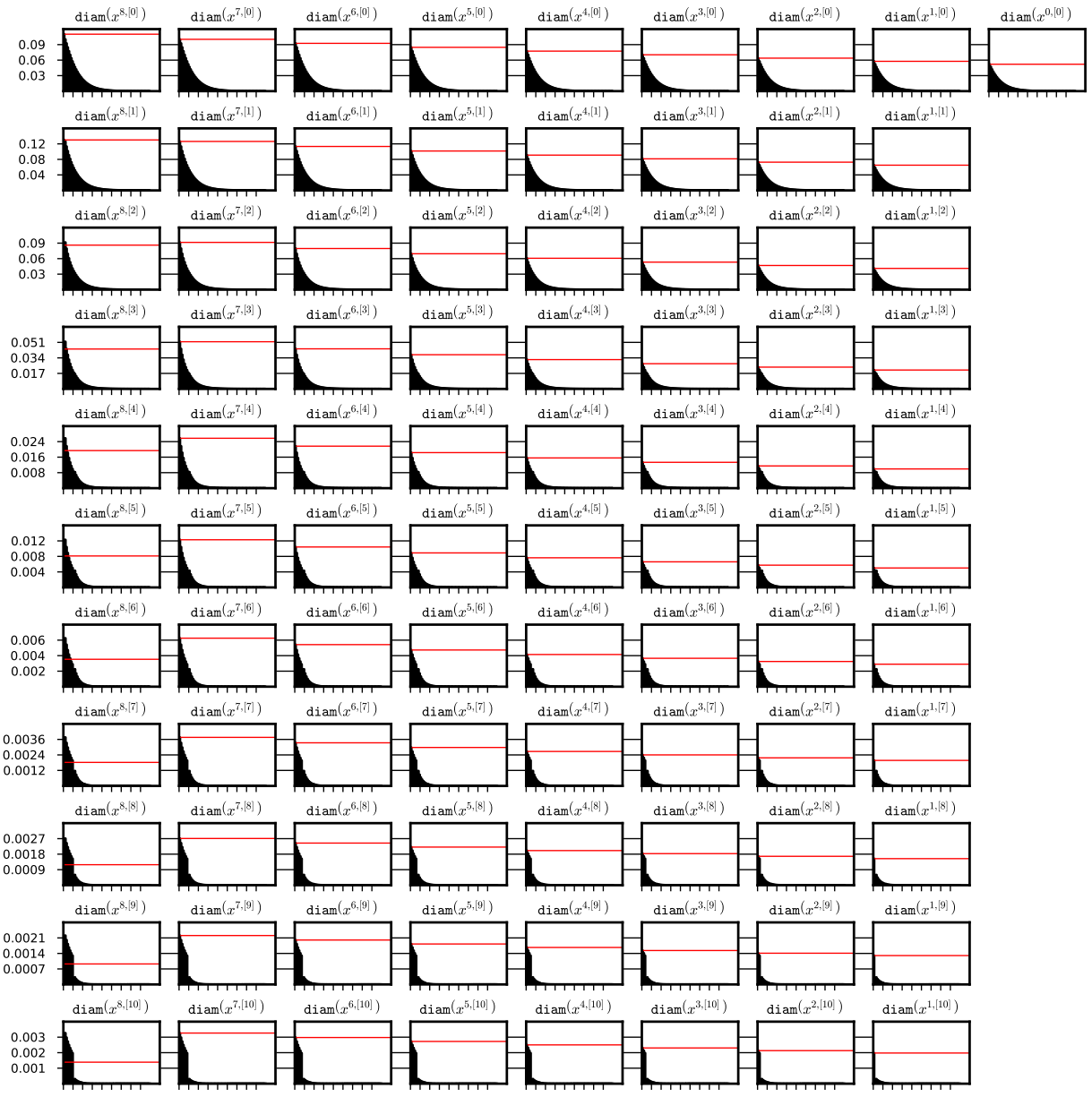

Figure 3.8: Diameters of the coefficients of a sequence $\{\Phi^n(\bar{x}_0)\}_{n\in\{16,..,8\cdot(9+2)\}}$ (a full history after $2 \cdot p$ steps) for some (8,9)-representation $\bar{x}_0$ of a stable stationary solution $x \equiv 0$ for system (2.14). Red horizontal line marks the diameter of the representation of the initial function. On the x-axis we have the iteration steps, each tick represents p steps of iteration. The data from test 2c was used. System (2.14), interval set representation and (8,9)-representation were used for the integration process. The data is stored in the file `steady_08_09_out_3/int_di.txt`.

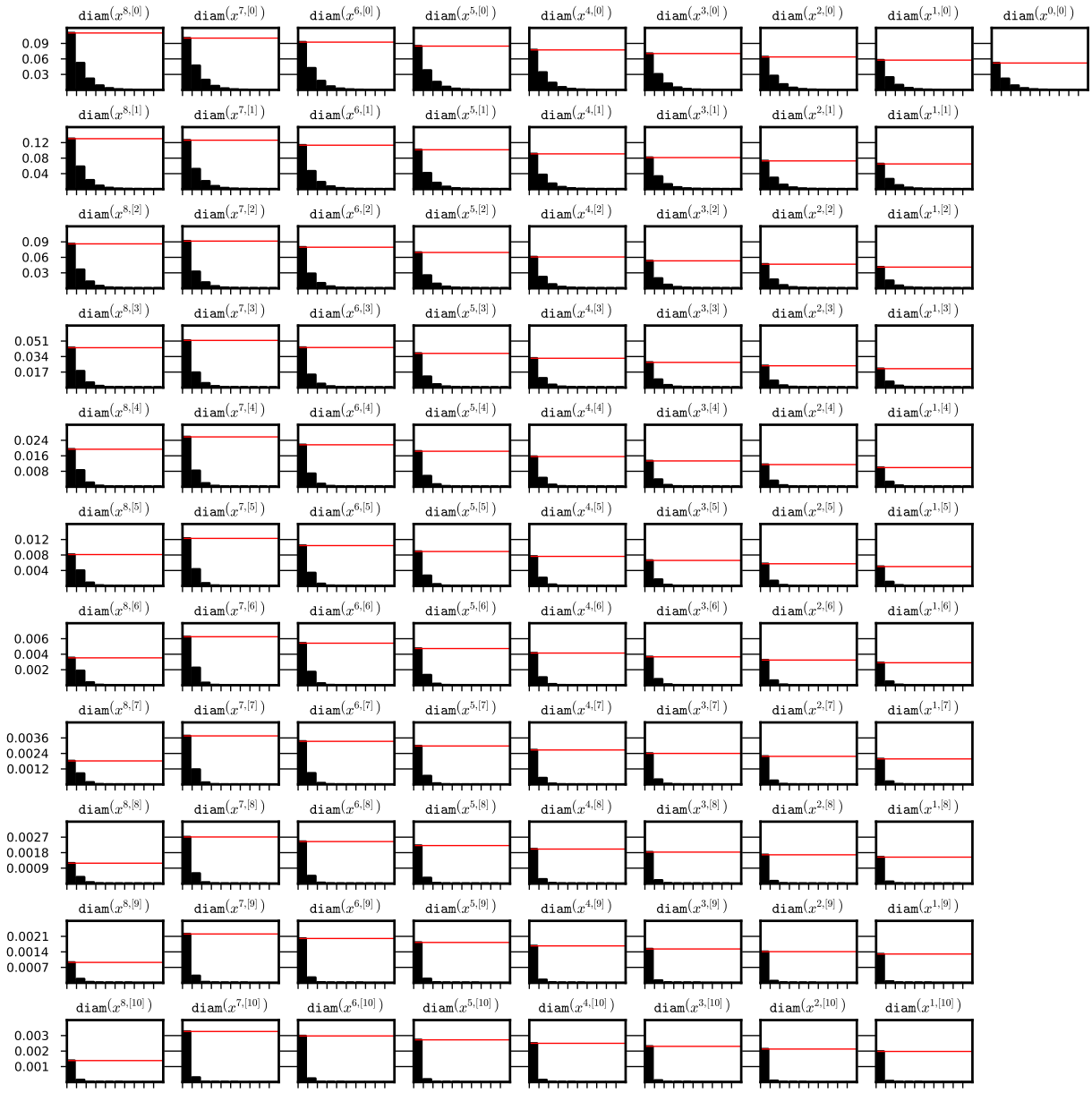

Figure 3.9: Diameters of the coefficients of a sequence $\left\{\Phi^{8\cdot n}(\bar{x}_0)\right\}_{n\in\{2,..,(9+2)\}}$ for some (8,9)-representation $\bar{x}_0$ of a stable stationary solution $x \equiv 0$ for system (2.14). Red horizontal line marks the diameter of the representation of the initial function after $2 \cdot p$ steps. On the x-axis we have the iteration steps, each tick represents p steps of iteration. The data from test 2c was used. System (2.14), interval set representation and (8,9)-representation were used for the integration process. The data is stored in the file `steady_08_09_out_3/int_di.txt`.

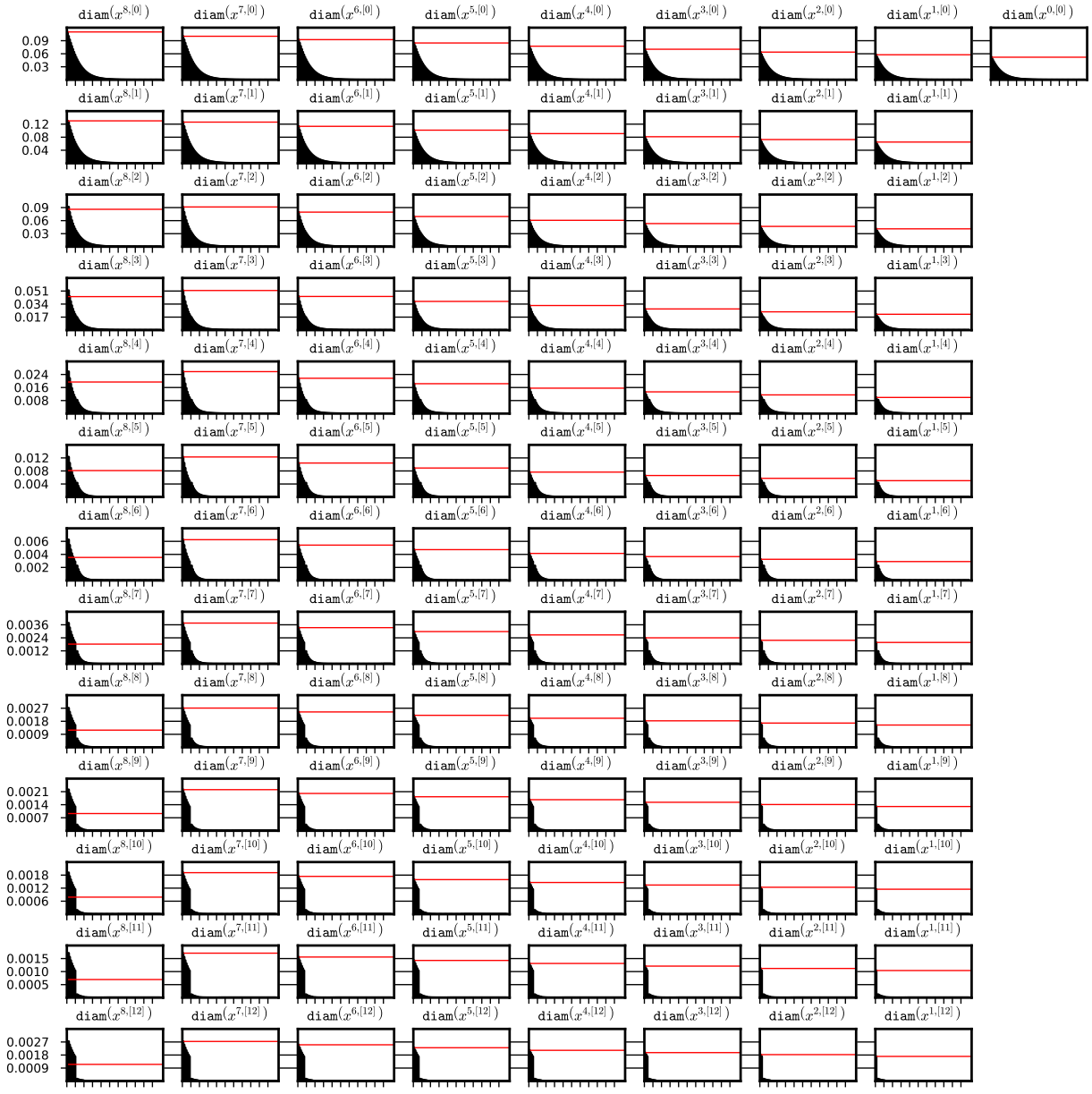

Figure 3.10: Diameters of the coefficients of a sequence $\{\Phi^n(\bar{x}_0)\}_{n\in\{16,..,8\cdot(11+2)\}}$ (a full history after $2\cdot p$ steps) for some (8,11)-representation $\bar{x}_0$ of a stable stationary solution $x \equiv 0$ for system (2.14). Red horizontal line marks the diameter of the representation of the initial function. On the x-axis we have the iteration steps, each tick represents p steps of iteration. The data from test 3c was used. System (2.14), interval set representation and (8,11)-representation were used for the integration process. The data is stored in the file `steady_08_11_out_3/int_di.txt`.

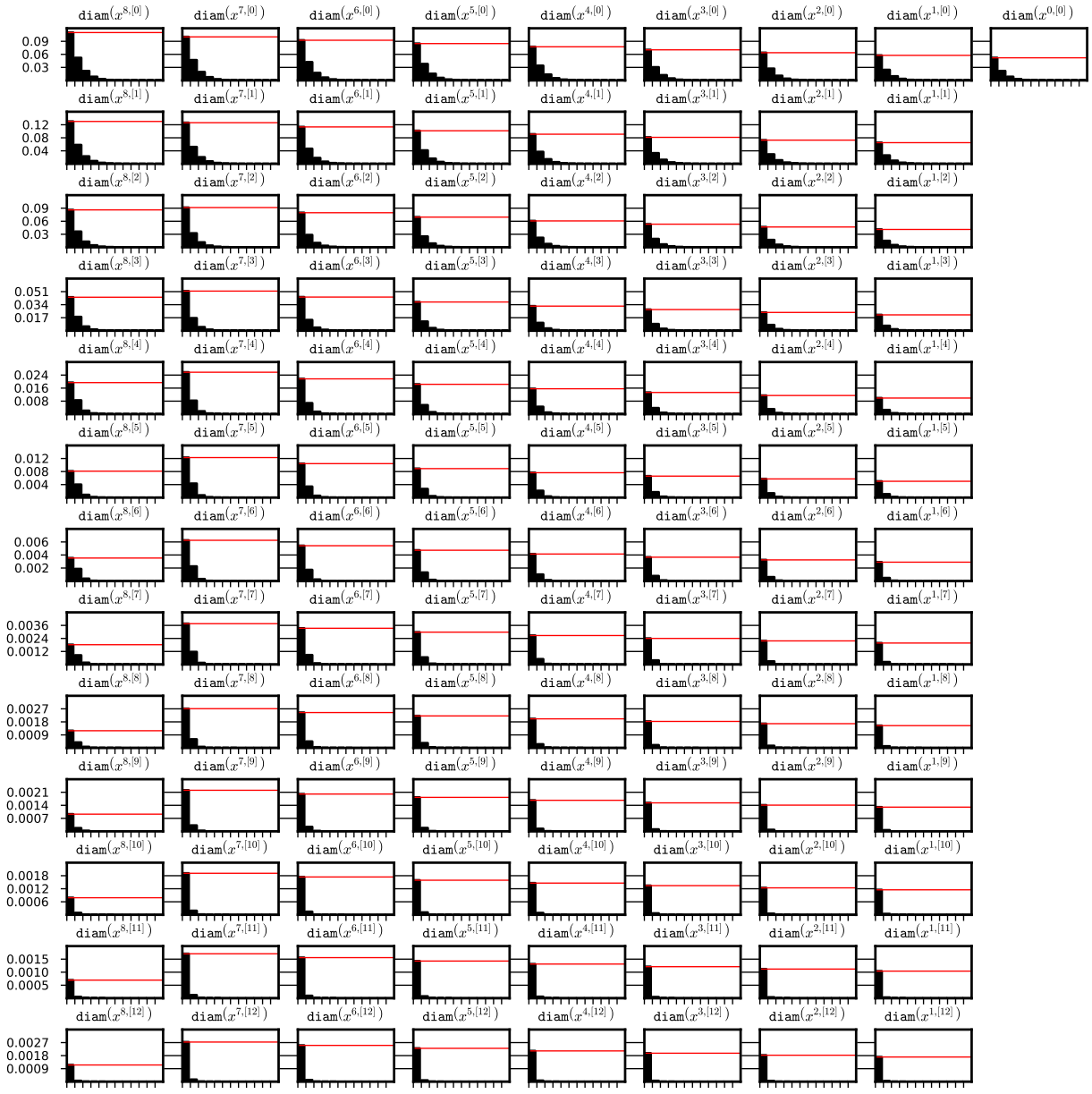

Figure 3.11: Diameters of the coefficients of a sequence $\left\{\Phi^{8\cdot n}(\bar{x}_0)\right\}_{n\in\{2,..,(11+2)\}}$ for some (8,11)-representation $\bar{x}_0$ of a stable stationary solution $x \equiv 0$ for system (2.14). Red horizontal line marks the diameter of the representation of the initial function after $2 \cdot p$ steps. On the x-axis we have the iteration steps, each tick represents p steps of iteration. The data from test 3c was used. System (2.14), interval set representation and (8,11)-representation were used for the integration process. The data is stored in the file `steady_08_11_out_3/int_di.txt`.

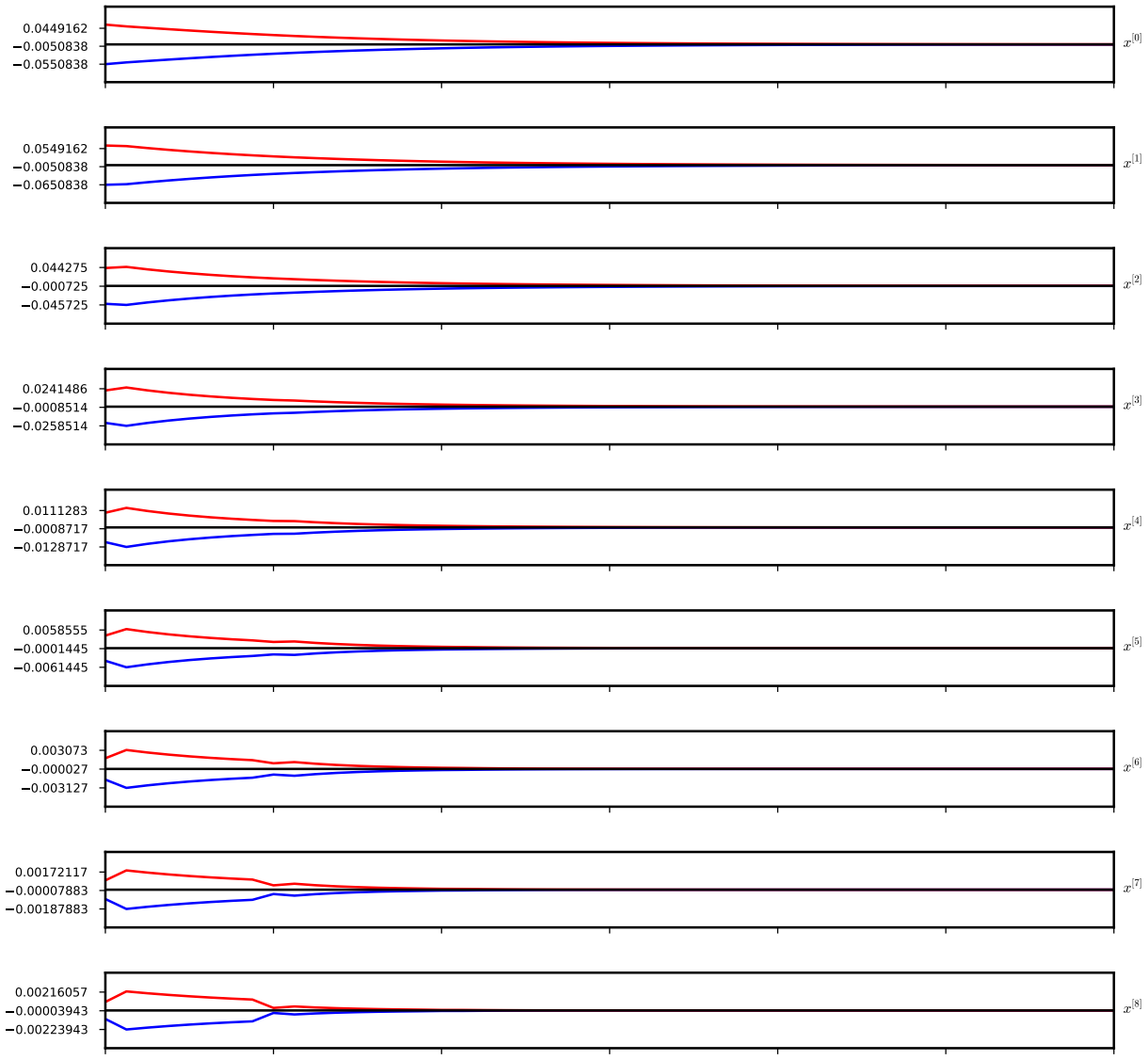

Figure 3.12: A graph of the solution $x(t) \equiv 0$ to (2.14), together with the all derivatives up to order $k = 7$ (black lines). Blue and red lines present lower and upper bounds respectively. For $k = 8$ we present the bound on the 8-th derivative on the intervals of the length $h = \frac{1}{p}$. On x axis we have time t. The data is as in Figure 3.6.

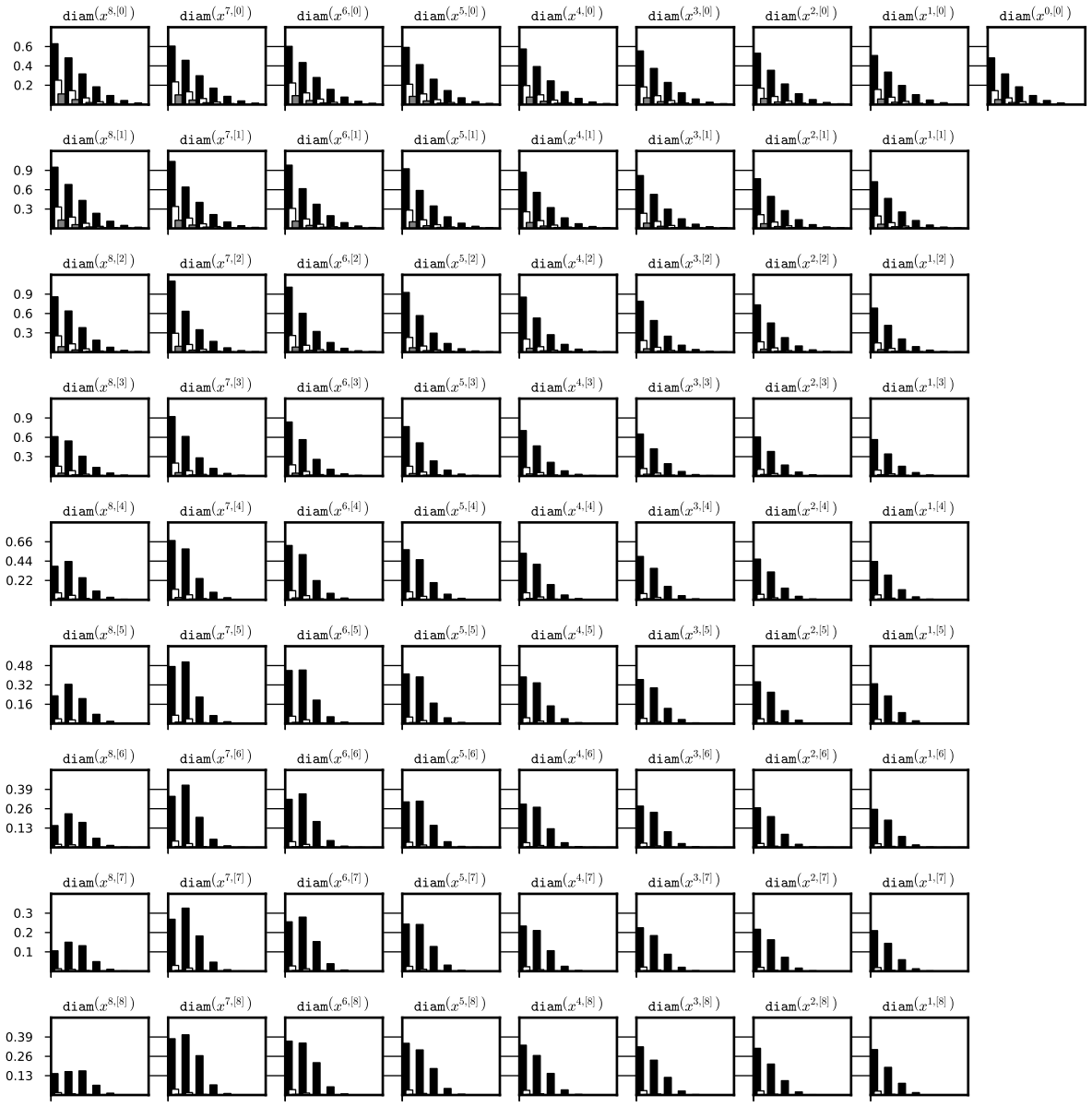

Figure 3.13: Dependence of the diameter of the interval set on the initial data diameter. After initial $2 \cdot p$ steps a history of the integration of some neighbourhood of a stable stationary solution $x \equiv 0$ to system (2.14) was recorded every p steps for three runs with initial data of decreasing diameter. On the x-axis we have the iteration steps, each bar is a diameter of the representation coefficient after p steps of iteration. The data is generated for tests 1a, 1b, 1c (black, white, gray respectively). System (2.14), doubleton Lohner set representation and (8,7)-representation were used for the integration process. The data is stored the files `steady_08_07_out_1/int_di.txt`, `steady_08_07_out_2/int_di.txt` and `steady_08_07_out_3/int_di.txt` respectively.

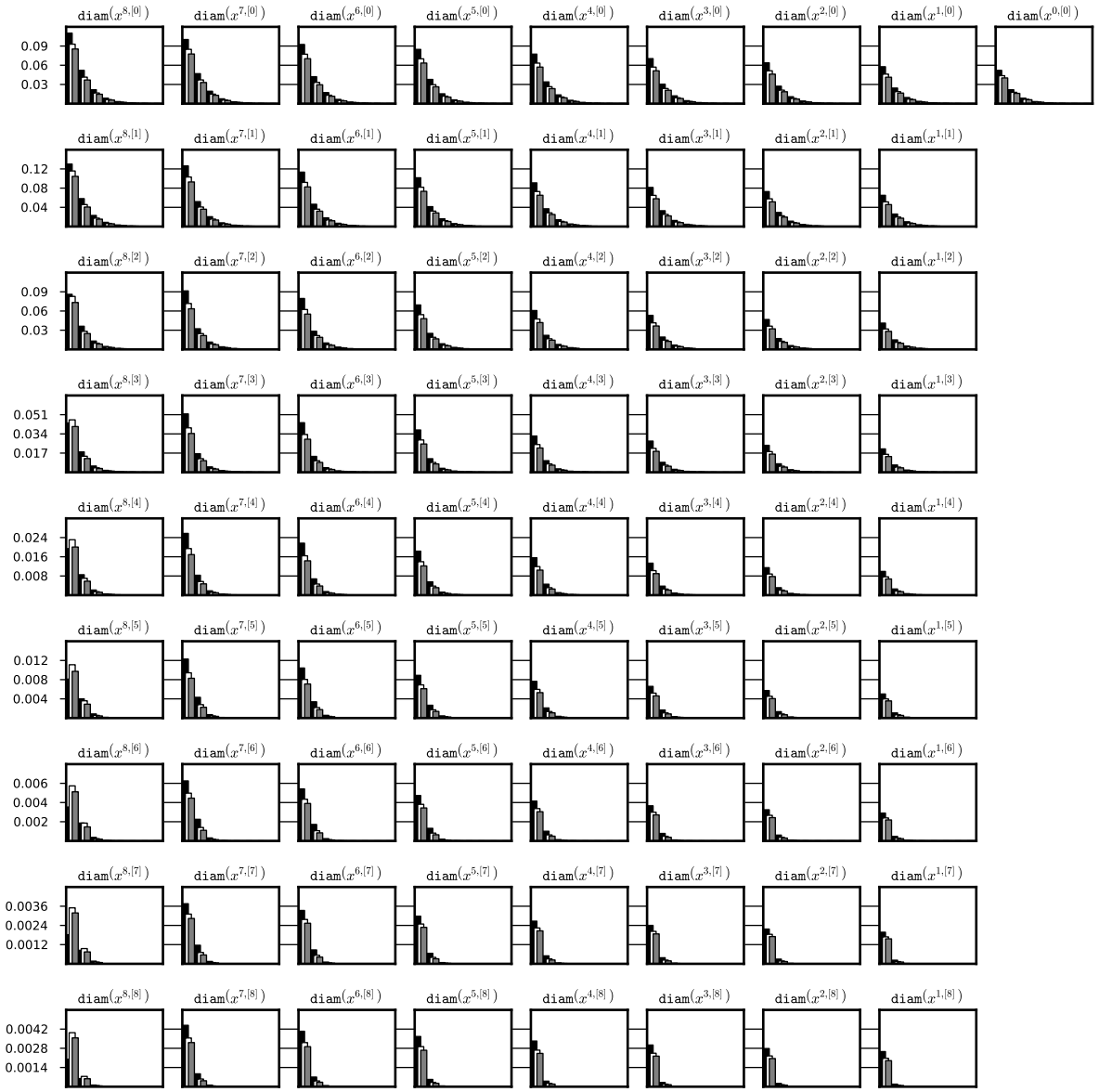

Figure 3.14: Dependence of the diameter of the representation on the grid size p. A history of one integration of three representations of a stable stationary solution $x \equiv 0$ for system (2.14) for parameter $p \in \{8, 16, 32\}$ was recorded every p steps (black, white, gray respectively). The diameters of corresponding representation coefficients (i.e. that represents appropriate derivative at the same time t) are drawn for comparision, i.e. $x^{i,[p]}$ for $p = 8$, $x^{2 \cdot i,[p]}$ for $p = 16$ and $x^{4 \cdot i,[p]}$ for $p = 32$. Each bar is a diameter of the representation coefficient after p steps of iteration. The tests used are: 1c, 4, 5 (black, white, gray respectively). For all integrations the system (2.14) and interval set representation were used. The data is stored the files `steady_08_07_out_3/int_di_p.txt`, `steady_16_07_out/int_di_p.txt` and `steady_32_07_out/int_di_p.txt` respectively.

Now we set our attention to a more involved case - a stationary periodic solution. For the periodic solution to equation (2.15) we have prepared an initial (p,n)-representations $\bar{x}_0$ using non-rigorous version of the integrator. We have integrated an initial function $x(t) = sin\left(\frac{\pi}{2} \cdot t\right)$ for $p \cdot 4$ steps and then we refined the resulting solution with the Newton method applied for a suitably created non-rigorous Poincaré map until the accuracy of about 10^{-14} was achieved. We denote the resulting representation with $\bar{p}$ and then we define:

$$\begin{aligned} \bar{x}_0^{0,[0]} &= \bar{p}^{0,[0]} \\ \bar{x}_0^{i,[k]} &= \bar{p}^{i,[k]} + [-a \cdot (p \cdot b)^k, b \cdot (p \cdot b)^k], \quad i \in \{1, .., p\}, k \in \{0, .., n\} \qquad (3.41) \\ \bar{x}_0^{i,[n+1]} &= 10 \cdot [-a \cdot (p \cdot b)^{n+2}, a \cdot (p \cdot b)^{n+2}], , \quad i \in \{1, .., p\}. \end{aligned}$$

(3.42)

The intuition behind this choice of the initial diameters is as follows: in the integration process each coefficient $\bar{x}_{i \cdot h}^{i,[k]}$ contribute to the coefficient $\bar{x}_{i \cdot h+h}^{i,[0]}$ with additional factor $h^k = \frac{1}{p^k}$ (we call it contribution factor). So, to achieve a equal contributions we should choose $\bar{x}_{i \cdot h}^{i,[k]} \approx (h^k)^{-1} = p^k$. For the remainder at $k = n + 1$ from equation (3.29) we get that the contribution factor is h^{n+2}. The parameters b may be chosen arbitrarily to get better results. We created 5 test cases, three of them have diameter variants, exactly as in case of stable stationary solutions discussed previously:

- Tests 1: $p = 8$, $n = 7$, $a \in \{0.4, 0.2, 0.1\}$ respectively,
- Tests 2a, 2b, 2c: $p = 8$, $n = 9$, $a \in \{0.4, 0.2, 0.1\}$ respectively,
- Tests 3a, 3b, 3c: $p = 8$, $n = 11$, $a \in \{0.4, 0.2, 0.1\}$ respectively,
- Tests 4: $p = 16$, $n = 7$, $a = 0.1$,
- Tests 5: $p = 32$, $n = 7$, $a = 0.1$.

Our numerical experiments (see author's web page for data files) suggests that for system (2.15) it is best to choose $b = 0.5$.

Remark 15 *We have also used heuristics from equation* (3.41) *for the stable stationary solution to system* (2.14) *and it resulted in very rapid contraction, from the first iteration of the integration process. Hovewer, due to this rapid contraction, the resulting figures were non-informative and not suitable for presentation. We reffer the author's web page for the results from those numerical experiments.*

We have run an iteration process as for stable stationary solution. Full history of the integration is presented in Figures 3.15, 3.17 and 3.19, and the history recorded every p

steps is presented in Figures 3.16, 3.18 and 3.20 for tests $1c, 2c, 3c$ respectively. In this case, contrary to the stationary solution presented earlier, we have included all representations from the history of the integration, including initial representation $\bar{x}_0$. Also we present the solution together with all derivatives up to order $k = 8$ over the whole time interval $t = [-1, p \cdot (n+1)]$ in Figures 4.11 and 4.12. The first one presents the numerical solution together with the rigorous bounds, while the second one presents only the bounds, centered at 0. The dependence of the resulting set diameter on the dimater of the initial set in test 3 is presented in Figure 3.23. The influence of the diameter of the resulting set on the choice of the parameter p (the grid spacing) is presented in Figure 3.24, where we have used tests 1c, 4 and 5, and we have compared diameters of the corresponding representation coefficients. All other comparision charts may be found on the author's web page.

As we can see from Figures 3.15, 3.17 and 3.19 we have obtained a strong contraction on the remainder, and some of the higher order derivative coefficients. However we have failed to obtain contraction for coefficients with $k \leq 4$, and we can see that the further integration may result in the blow-up of the representation. We also see that the oscillatory properties of the system are reflected in the diameter of the set. This is due to the fact that we have represented each set as a product of closed intervals. When rotated, the representation can no longer be represented by the product of intervals, thus we must enclose it, introducing overestimates. This is an example of the phenomena called *wrapping effect*. We will show in Chapter 4 how to reduce this problem and we will analyze the performance of proposed algorithms.

The dependence of the initial set diameter is very good, as presented in Figure 3.13. We see that changing parameter a to $\frac{a}{2}$ results in decreasing the resulting set diameter about two times as expected.

The dependence of the diameter on the step size (grid size) is better visible for the periodic solution than in the case of stationary solution. We see that changing p from 8 to 16 leads to approximately 2 times smaller diameters of the resulting sets. Changing p from 16 to 32 is less visible, but it still gives better estimates.

Despite the fact that we do not get the contraction on all coefficients, we see that the integrator reflects the strong contraction for the efficients of high order, especially for the remainders, that is what we wanted to achieve. In the Chapter 4 we will show how to obtain better estimates for low order coefficients (for small k).

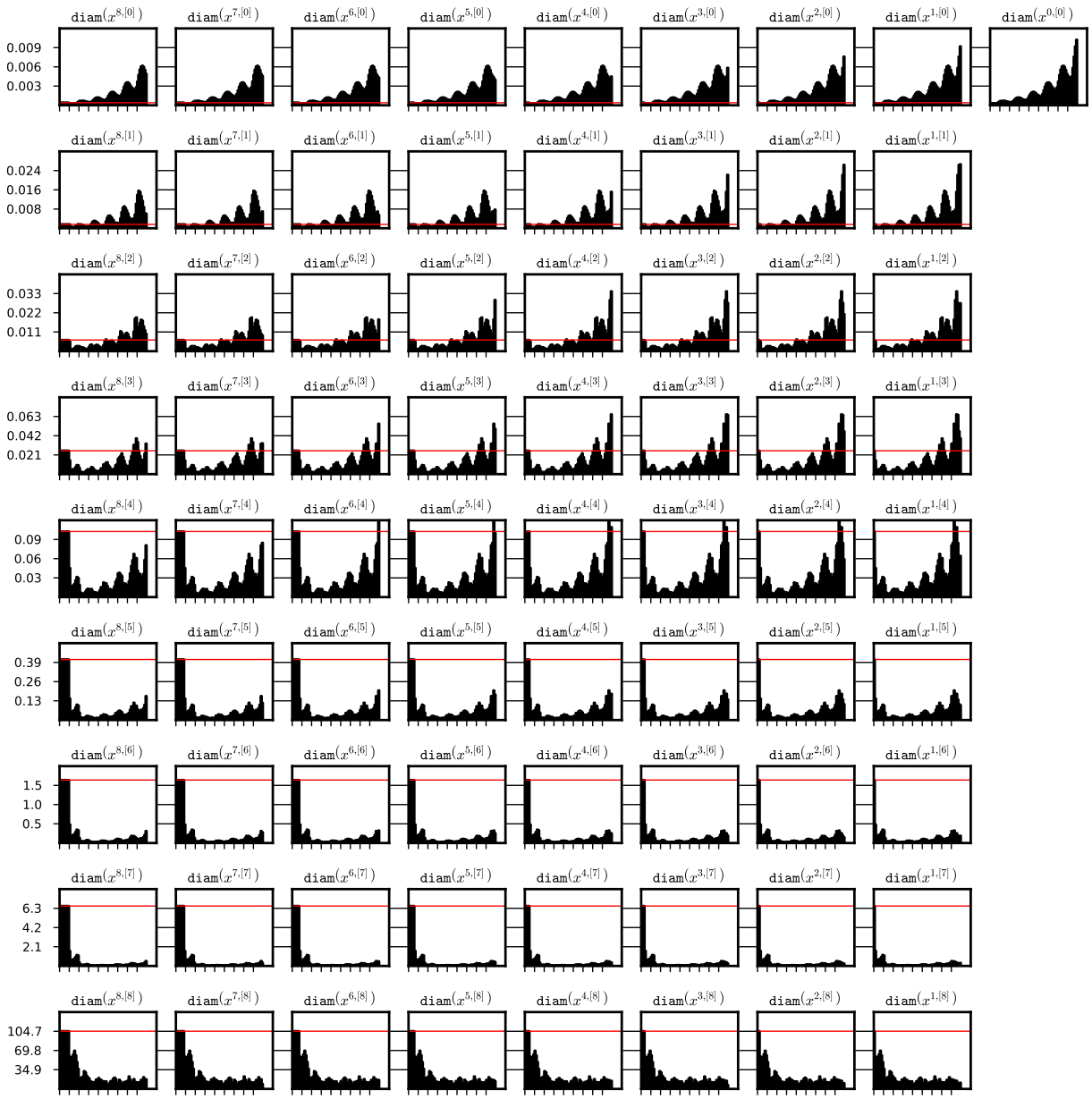

Figure 3.15: Diameters of the coefficients of a sequence $\{\Phi^n(\bar{x}_0)\}_{n\in\{0,..,8\cdot(7+2)\}}$ (a full history) for some (8,7)-representation $\bar{x}_0$ of a stable periodic orbit for system (2.15). Red horizontal line marks the diameter of the representation of the initial function. On the x-axis we have the iteration steps, each tick represents p steps of iteration. The data from test 1c was used. System (2.15), interval set representation and (8,7)-representation were used for the integration process. The data is stored in the file `periodic_08_07_out_3/int_di.txt`.

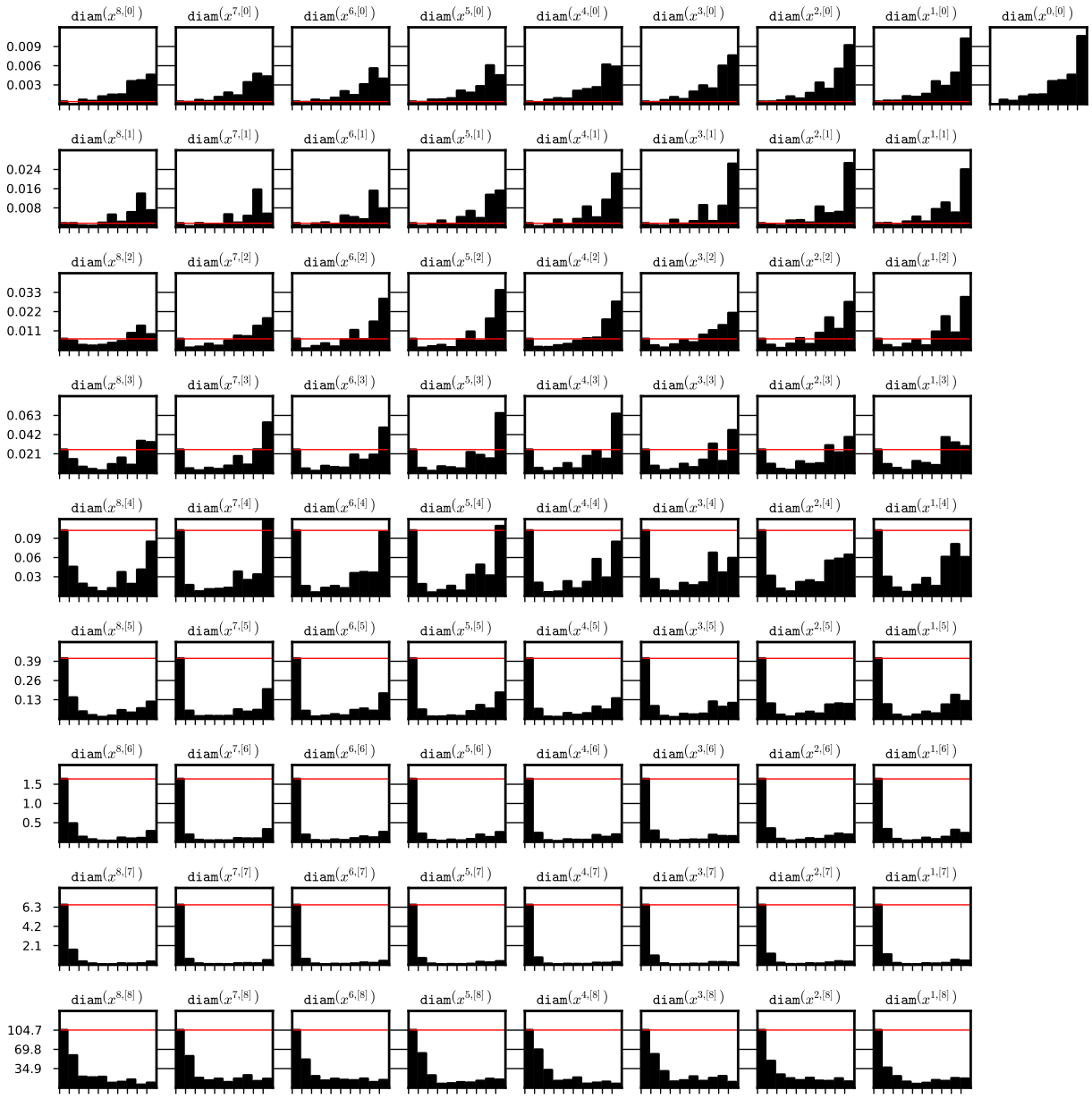

Figure 3.16: Diameters of the coefficients of a sequence $\left\{\Phi^{8\cdot n}(\bar{x}_0)\right\}_{n\in\{0,..,(7+2)\}}$ for some (8,7)-representation $\bar{x}_0$ of a stable periodic orbit for system (2.15). Red horizontal line marks the diameter of the representation of the initial function. On the x-axis we have the iteration steps, each tick represents p steps of iteration. The data from test 1c was used. System (2.15), interval set representation and (8,7)-representation were used for the integration process. The data is stored in the file `periodic_08_07_out_3/int_di.txt`.

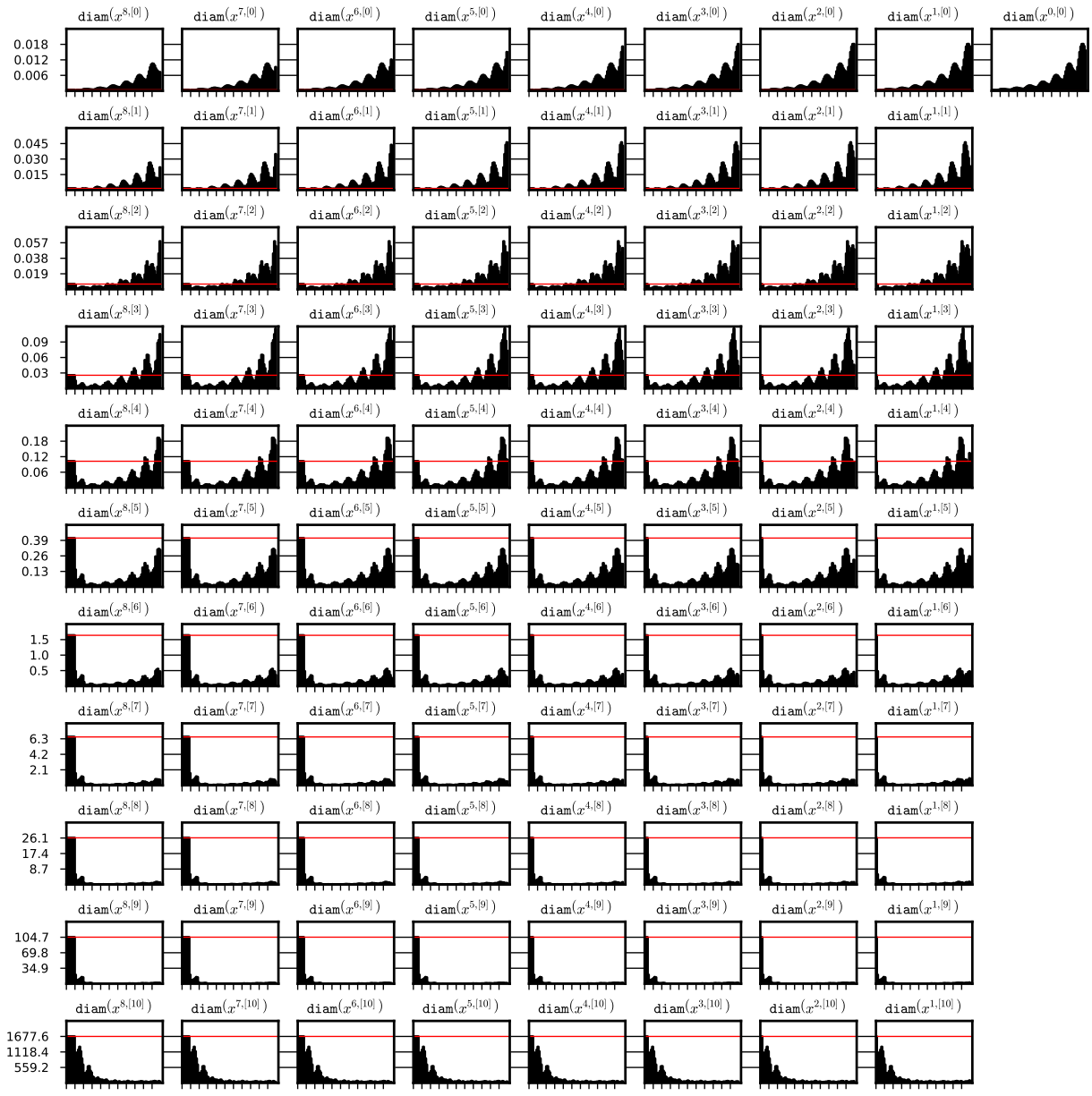

Figure 3.17: Diameters of the coefficients of a sequence $\{\Phi^n(\bar{x}_0)\}_{n\in\{0,\ldots,8\cdot(9+2)\}}$ (a full history) for some (8,9)-representation $\bar{x}_0$ of a stable periodic orbit for system (2.15). Red horizontal line marks the diameter of the representation of the initial function. On the x-axis we have the iteration steps, each tick represents p steps of iteration. The data from test 2c was used. System (2.15), interval set representation and (8,9)-representation were used for the integration process. The data is stored in the file `periodic_08_09_out_3/int_di.txt`.

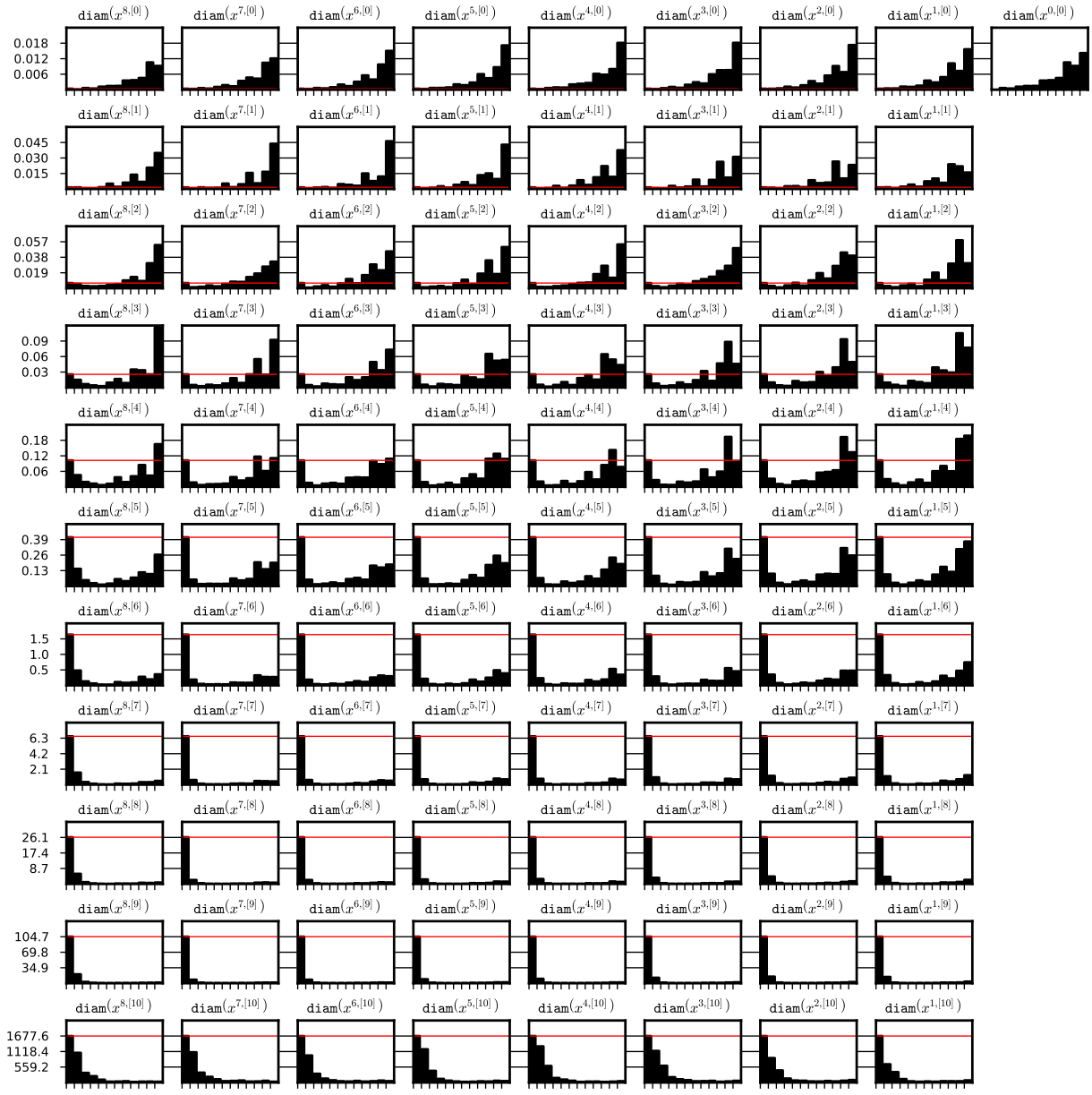

Figure 3.18: Diameters of the coefficients of a sequence $\left\{\Phi^{8\cdot n}(\bar{x}_0)\right\}_{n\in\{0,..,(9+2)\}}$ for some (8,9)-representation $\bar{x}_0$ of a stable periodic orbit for system (2.15). Red horizontal line marks the diameter of the representation of the initial function. On the x-axis we have the iteration steps, each tick represents p steps of iteration. The data from test 2c was used. System (2.15), interval set representation and (8,9)-representation were used for the integration process. The data is stored in the file `periodic_08_09_out_3/int_di.txt`.

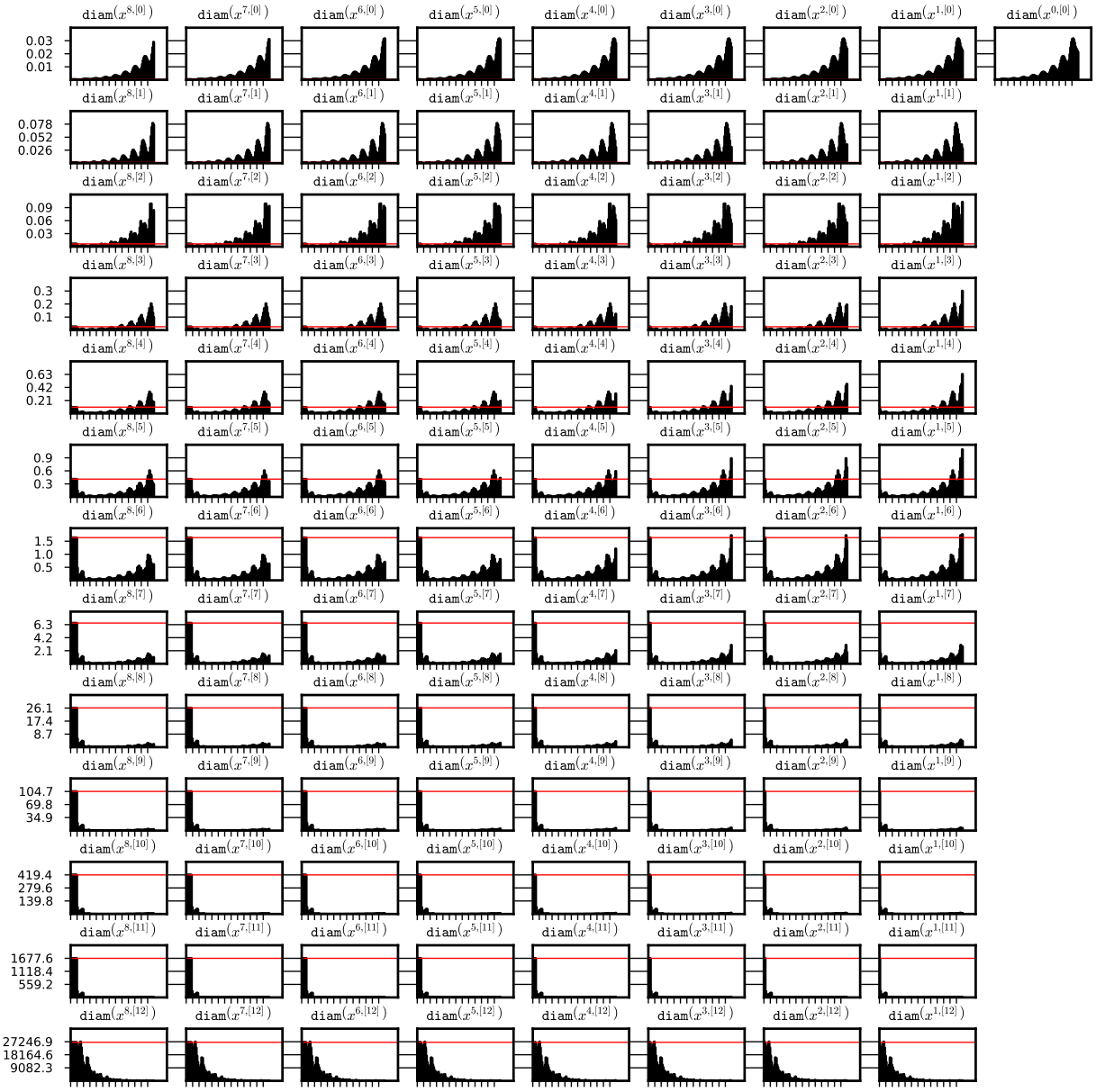

Figure 3.19: Diameters of the coefficients of a sequence $\{\Phi^n(\bar{x}_0)\}_{n\in\{0,..,8\cdot(11+2)\}}$ (a full history) for some (8,11)-representation $\bar{x}_0$ of a stable periodic orbit for system (2.15). Red horizontal line marks the diameter of the representation of the initial function. On the x-axis we have the iteration steps, each tick represents p steps of iteration. The data from test 3c was used. System (2.15), interval set representation and (8,11)-representation were used for the integration process. The data is stored in the file `periodic_08_11_out_3/int_di.txt`.

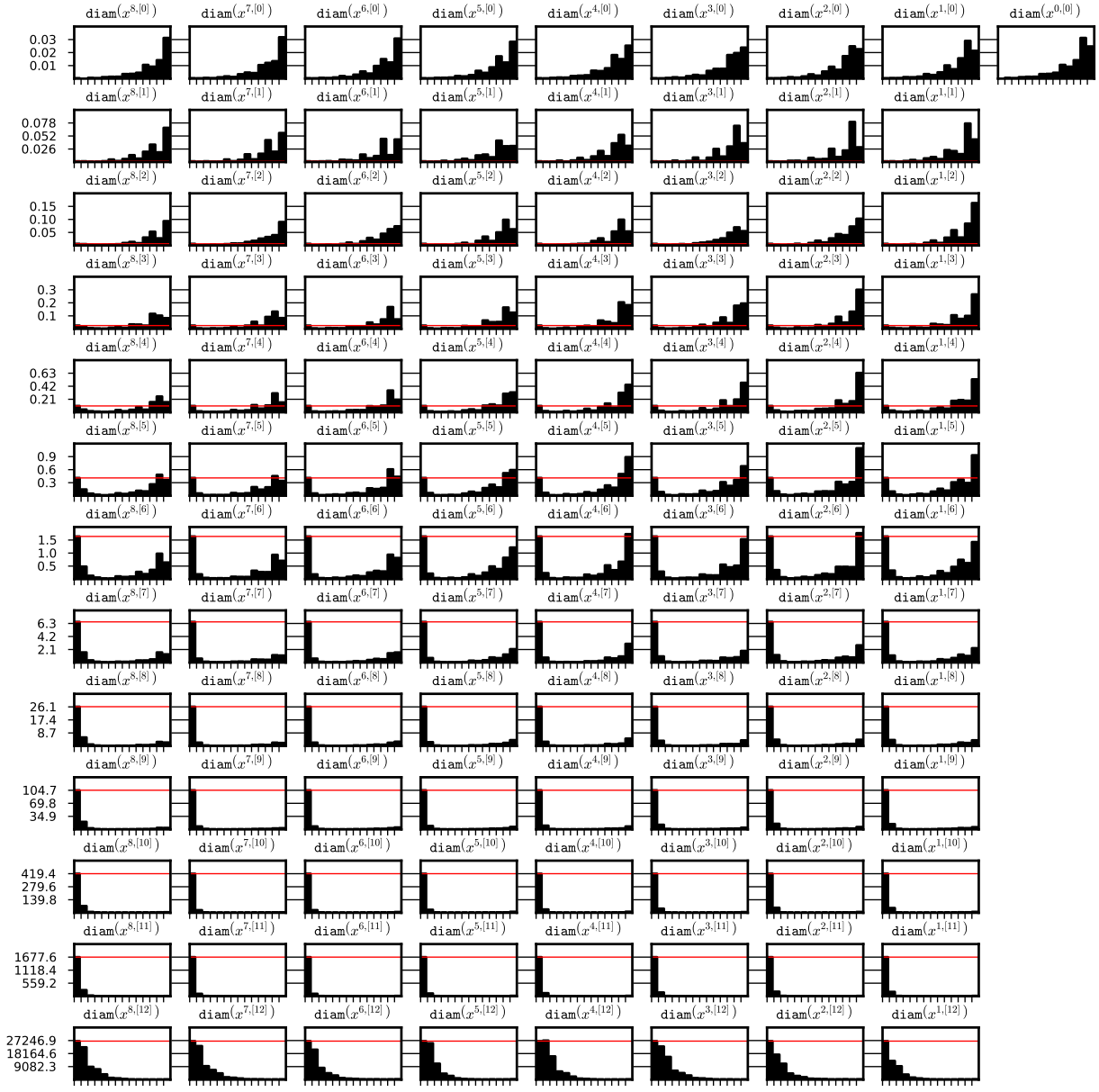

Figure 3.20: Diameters of the coefficients of a sequence $\left\{\Phi^{8\cdot n}(\bar{x}_0)\right\}_{n\in\{0,..,(11+2)\}}$ for some (8,11)-representation $\bar{x}_0$ of a stable periodic orbit for system (2.15). Red horizontal line marks the diameter of the representation of the initial function. On the x-axis we have the iteration steps, each tick represents p steps of iteration. The data from test 3c was used. System (2.15), interval set representation and (8,11)-representation were used for the integration process. The data is stored in the file `periodic_08_11_out_3/int_di.txt`.

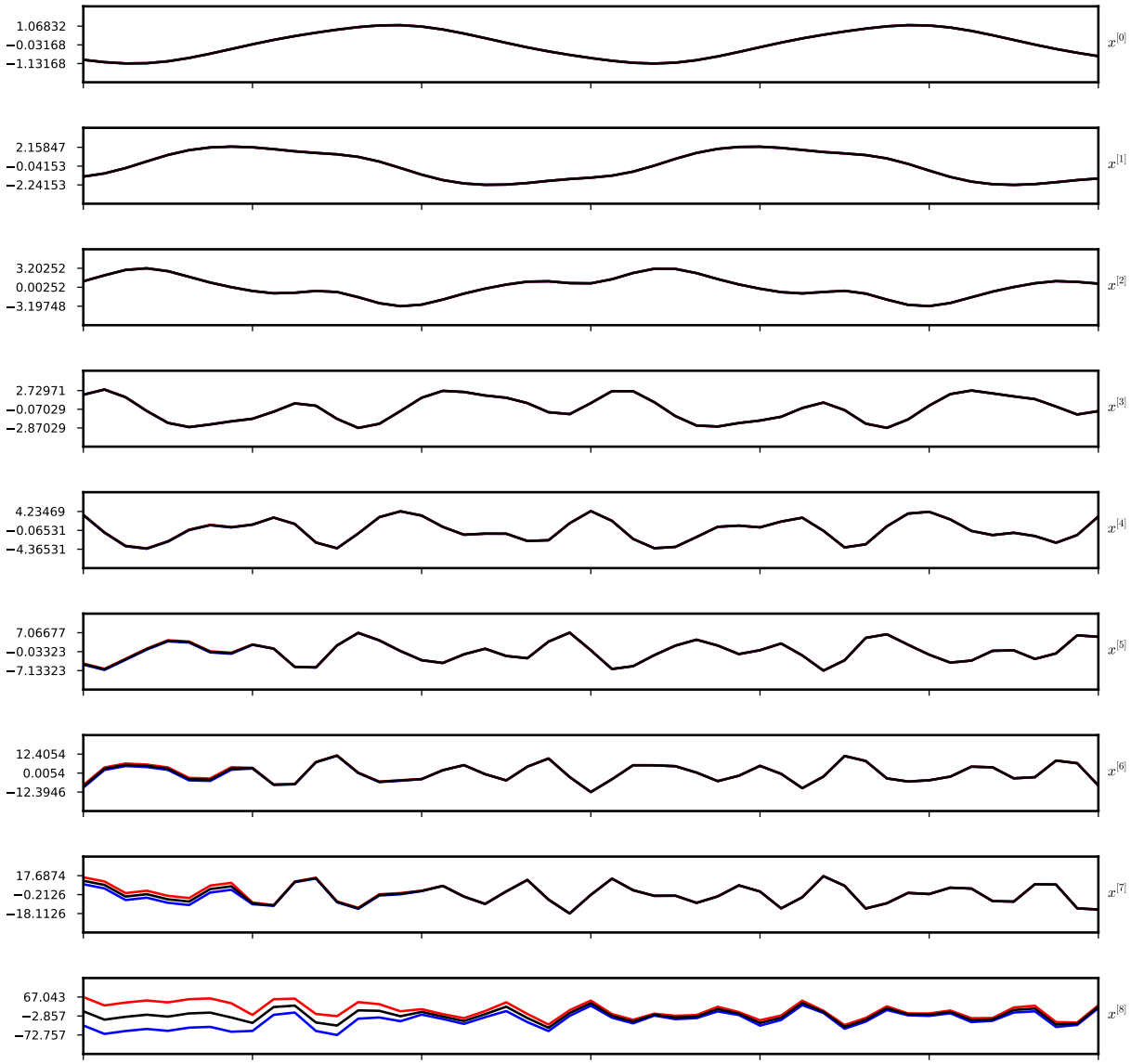

Figure 3.21: A graph of the stable periodic solution $x(t)$ to (2.15), together with the all derivatives up to order $k = 7$ (black lines). Blue and red lines present lower and upper bounds respectively. For $k = 8$ we present the bound on the 8-th derivative on the successive intervals of the length $h = \frac{1}{p}$. On x axis we have time t. The data is as in Figure 3.15.

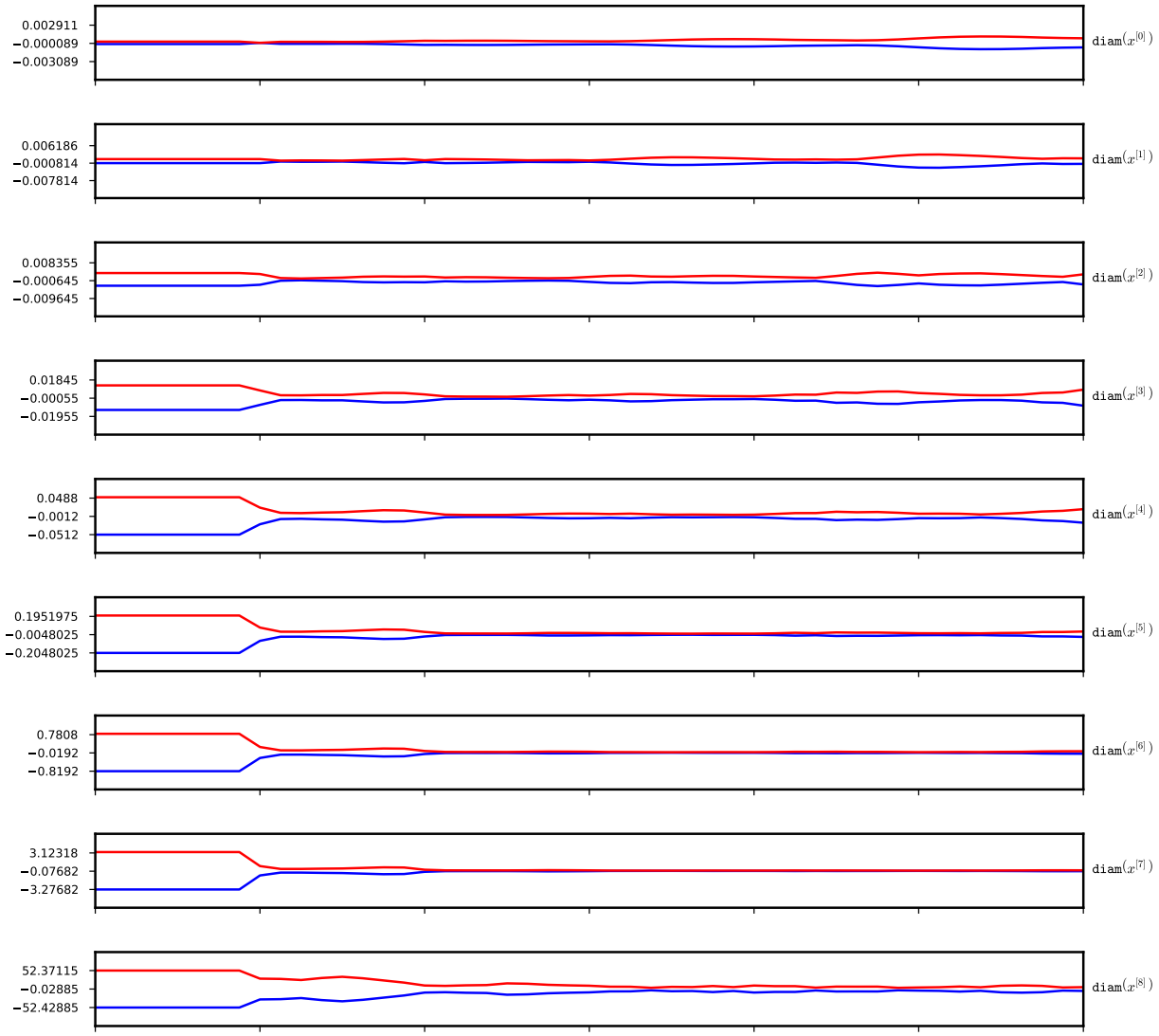

Figure 3.22: A graph of the lower and upper bounds for the stable periodic solution $x(t)$ to (2.15) and its derivatices shifted by the numerical approximation to the solution. On x axis we have time t. The data is as in Figure 3.15.

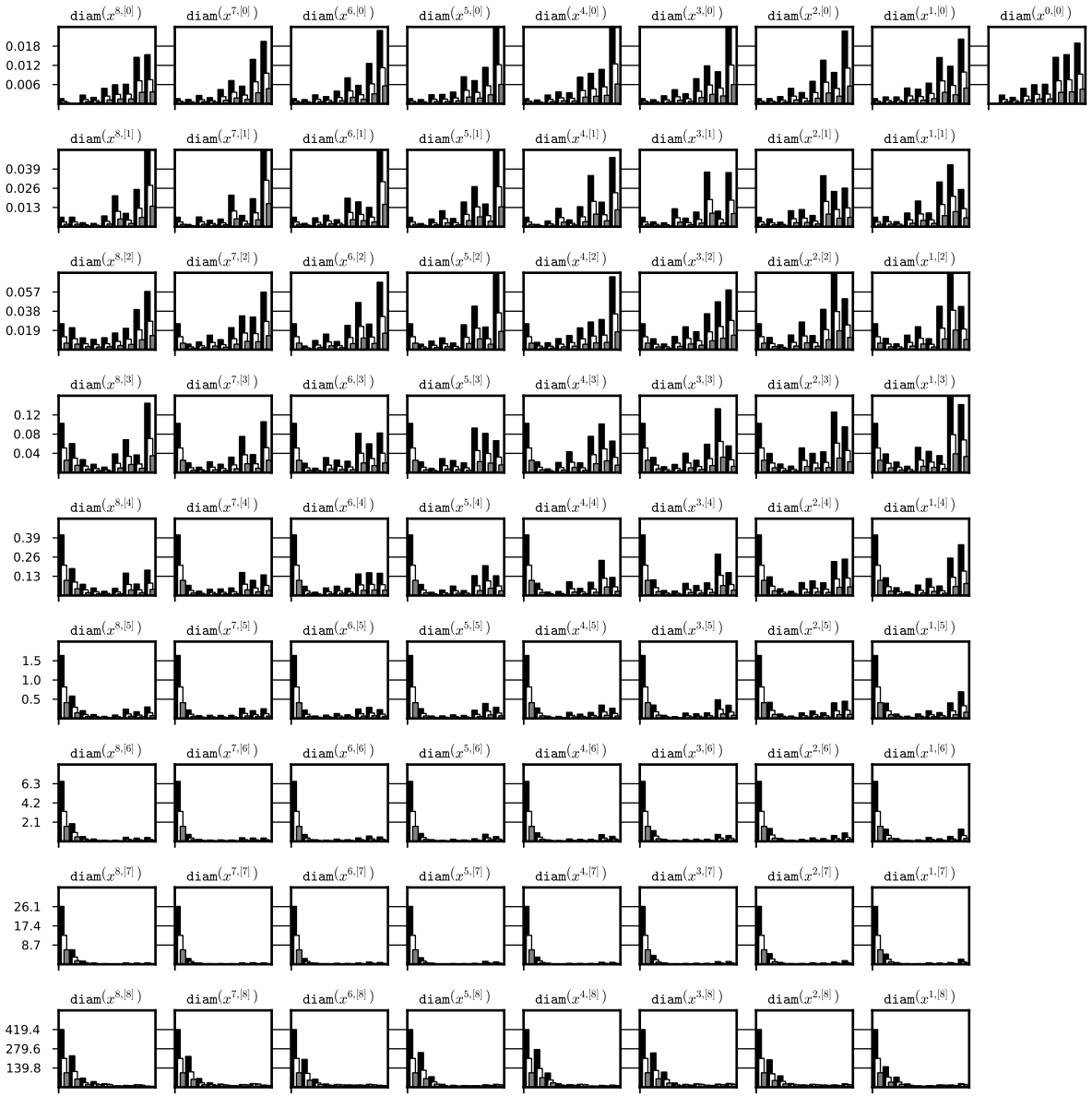

Figure 3.23: Dependence of the diameter of the interval set on the initial data diameter. A history of the integration of some neighbourhood of a stable periodic orbit for system (2.15) was recorded for three runs with initial data of decreasing diameter. On the x-axis we have the iteration steps, each bar is a diameter of the representation coefficient after p steps of iteration. The data is generated for tests 1a, 1b, 1c (black, white, gray respectively). System (2.15), interval set representation and (8,7)-representation were used for the integration process. The data is stored the files `periodic_08_07_out_1/int_di.txt`, `periodic_08_07_out_2/int_di.txt` and `periodic_08_07_out_3/int_di.txt` respectively.

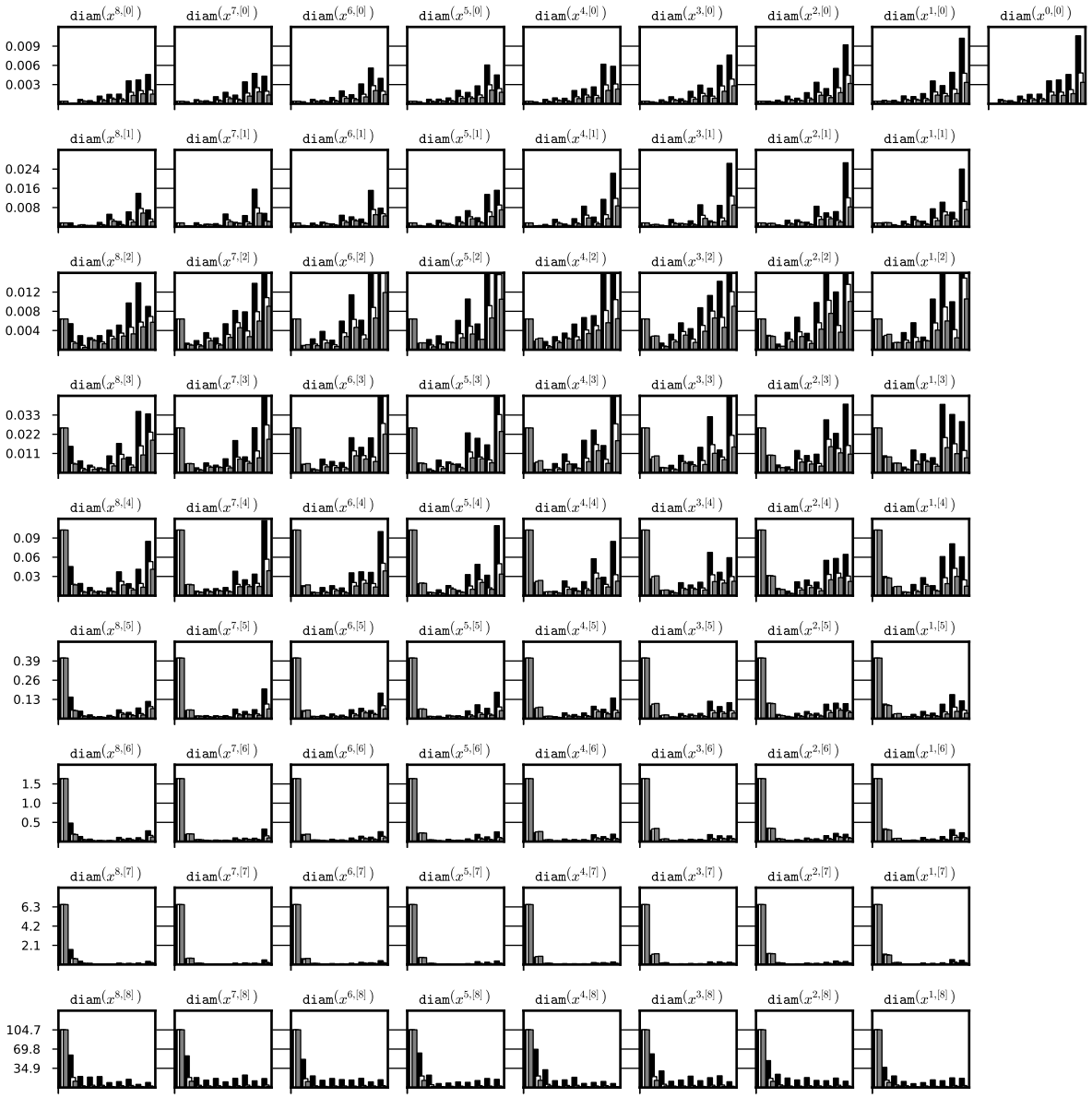

Figure 3.24: Dependence of the diameter of the representation on the grid size p. A history of one integration of three representations of a stable periodic orbit for system (2.15) for parameter $p \in \{8, 16, 32\}$ was recorded every p steps (black, white, gray respectively). The diameters of corresponding representation coefficients (i.e. that represents appropriate derivative at the same time t) are drawn for comparision, i.e. $x^{i,[p]}$ for $p = 8$, $x^{2 \cdot i,[p]}$ for $p = 16$ and $x^{4 \cdot i,[p]}$ for $p = 32$. Each bar is a diameter of the representation coefficient after p steps of iteration. The tests used are: 1c, 4, 5 (black, white, gray respectively). For all integrations the system (2.15) and interval set representation were used. The data is stored the files `periodic_08_07_out_3/int_di_p.txt`, `periodic_16_07_out/int_di_p.txt` and `periodic_32_07_out/int_di_p.txt` respectively.

Chapter 4

Reducing the „Wrapping Effect": Lohner set representation

Up till now we have only discussed basic algorithms which work theoretically in the set arithmetics, with no computer precisions errors, and with ideal representation of sets. When it comes to rigorous numerics implemented using the interval arithmetic we have two major limitations: (1) set representation by the product of intervals and (2) the finiteness of the computer representation of real numbers. The first of the limitations is a source of two imminent negative effects: *the wrapping effect* and *dependency problem*.

Dependency problem arises when the same interval is used several times in a complicated numerical formula, for example evaluating formula $x - x$ using interval arithmetic for $x = [-1, 1]$ produces $[-2, 2]$ instead of expected $[0, 0]$. This is very simple and rather artificial example, but similar pattern occurs for example when computing values of function using the Taylor expansion up to a given order, which is quite common in many applications.

The wrapping effect is a direct consequence of the fact that a result of a function computed on an interval set must again be an interval set. It is visible when there exist some kind of rotation in the system. The geometric intuition is simple in 2D: when we rotate a rectangle we can get a rhomboidal shape. To produce interval enclosure of this shape we need to introduce many unnecessary points - the corners. When the procedure is repeated many times (iteration of the map/function) the errors may grow exponentially, eventually 'blowing up' the whole solution. An illustration of this phenomenon is presented in Figure 4.1.

Wrapping effect is usually regarded as more dangerous in many applications than the dependency problem. It is even more dangerous in DDE setting as most of the eigenvalues of the system are imaginary, giving rise to many rotations, even though the eigenvalues associated with rotating directions are small. To reduce the wrapping effect during the integration we are going to use the Lohner algorithm and Lohner sets[15].

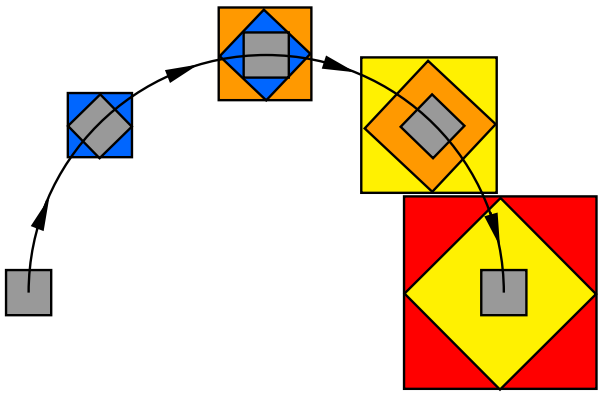

Figure 4.1: An illustration of the wrapping effect problem for a classical, idealized, mathematical pendulum ODE $\ddot{x} = -x$. The picture shows a set of solutions in a phase space $(x, \dot{x})$. The gray boxes present points of initial box moved by the flow. The color boxes presents the wrapping effect occurring at each step when we want to enclose the moving points in a product of intervals in the basic coordinate system.

4.1 The Lohner algorithm for DDEs

To deal with the wrapping effect one usually employ an algorithm to locally change the coordinate frame during the computations to minimize the size of overestimates introduced by the wrapping effect. In our work we will use the method of Lohner[33, 15]. We would like to briefly recall the procedure in the case of ODEs. In the following we use the notation $[A]$ to stress that set A is an interval set. We assume that we have IVP:

$$\begin{aligned} x'(t) &= G(x), \quad G : \mathbb{R}^M \to \mathbb{R}^M, \\ x(0) &= x_0, \end{aligned} \tag{4.1}$$

and for some step size $0 < h < 1$ we are interested in the enclosures $[x_k]$ of $x_k = x(i \cdot h)$ for $i \in \mathbb{N}$. We assume that we have a numerical method $\Phi_{G,h} : \mathbb{R}^M \to \mathbb{R}^M$ such that $\varphi_G(x, h) = \Phi_{G,h}(x) + Rem_{G,h}(x)$ for some $Rem_{G,h} : \mathbb{R}^M \to \mathbb{R}^M$, $Rem_{G,h}(x) \in O(h^{n+1})$. Assume that we have enclosure $[x_k]$ for x_k and we want to compute enclosure $[x_{k+1}]$ of x_{k+1} the lohner method is as follows:

1. compute rough enclosure W_1: let $[W_1]$ be such that $\varphi_G([x_k], [0, h]) \subset W_1$,
2. $A_k = \frac{\partial \Phi_{G,h}}{\partial x}([x_k])$,
3. let $m([x_k])$ be a midpoint of the set $[x_k]$, then:

$$[x_{k+1}] = \Phi_{G,h}(m([x_k])) + [A_k] \cdot ([x_k] - m([x_k])) + Rem_{G,h}([W_1]). \tag{4.2}$$

All the methods are also valid for non-autonomous r.h.s. $G = G(x,t)$. We see that there are three advantages of using Lohner method:

1. while the *Rem* is of order $n+1$ in h then the influence of rough enclosure W_1 should be small,

2. moving forward the enclosure is done by the numerical method only on the midpoint of $[x_k]$, so there are no errors introduced by the evaluation on the whole set (in the case of idealized numerics),

3. the interval matrix $[A_k]$ gives us opportunity to change the coordinate frame, which allows to control the wrapping effect. If we would represent $[x_k]$ as $x_k + [r_k]$, where $x_k = m([x_k])$ and $[r_k] = [x_k] - x_k$ then the equation (4.2) to be solved reduces to:

$$\begin{aligned} x_{k+1} &= \Phi_{G,h}(x_k) && (4.3)\\ [r_{k+1}] &= [A_k] \cdot ([r_k]) + Rem_{G,h}([W_1]). && (4.4) \end{aligned}$$

The quality of the wrapping effect control will depend on the way we will represent r_k. Namely:

1. *Interval set*: $[r_k] = \Pi_{i=1}^{M} I_j$, I_j - interval,

2. *Parallelepiped*: $[r_k] = B_k \cdot [\tilde{r_k}]$, B_k is a matrix, $[\tilde{r_k}]$ is an Interval set,

3. *Cuboid*: $[r_k] = Q_k \cdot [\tilde{r_k}]$, B_k is an orthogonal matrix, $[\tilde{r_k}]$ is an Interval set,

4. *Doubleton*: $[r_k] = C_k \cdot [r_0] + [\tilde{r_k}]$, C_k is a matrix, $[\tilde{r_k}]$ is either an interval set, cuboid or parallelepiped.

We are not going to discuss in details how to achieve conservation of the Lohner representation at each step, we only want to give an example for the case of Parallelepiped (method 2). We see that for this representation the r.h.s. of equation (4.2) is equivalent to evaluating:

$$B_{k+1} \cdot [\tilde{r_{k+1}}] = [A_k] \cdot B_k \cdot [\tilde{r_k}] + Rem_{G,h}([x_k]), \tag{4.5}$$

thus we usually set:

$$\begin{aligned} B_{k+1} &= m([A_k] \cdot B_k), && (4.6)\\ [\tilde{r_{k+1}}] &= \left(B_{k+1}^{-1} \cdot [A_k] \cdot B_k\right) \cdot [\tilde{r_k}] + B_{k+1}^{-1} \cdot Rem_{G,h}([x_k]). && (4.7) \end{aligned}$$

We see that the choosing of B_{k+1} leads (in ideal case) to $B_{k+1}^{-1} \cdot [A_k] \cdot B_k$ being close to identity, thus in the next step $[\tilde{r_{k+1}}]$ should not suffer from the wrapping effect.

The method 1 is equivalent to using a bare interval arithmetic. Other methods has their own advantages and disadvantages, but making our integration scheme compatible with

the standard Lohner notation (implementing Φ and Rem) will allow us to use the standard Lohner sets implementation in the CAPD library, thus it will allow us to exchange the representation as needed without need for reimplementing all Lohner sets. See the online documentation [22] for details.

Since in our DDE integration scheme we can use the Lohner algorithm, as for $t \in [0,1]$ we can regard solving equation (3.1) as solving non-autonomous ODE. In the case of DDEs we must modify Algorithm 6. We divide it into two parts, just like in the Lohner algorithm: the numerical method Φ in Algorithm 8 and the remainder Rem in Algorithm 9. The Jacobian of the Φ at x will be computed using Automatic Differentiation. As you see we abused notation here and used Φ to denote the numerical method without remainder. From now on, in this chapter, the symbol Φ will be used as a name of the procedure in Algorithm 8 and symbol Rem will mean Algorithm 9.

Algorithm 8 Φ

Input: $\bar{x}_t$

Output: $\bar{x}_{t+h}$

Require: $\bar{x}_t$ is a (p,n)-representation

Ensure: $\bar{x}_{t+h}$ is a (p,n)-representation containing nonzero entries on (i,k) and $(0,0)$ for $1 \le i \le p$ and $0 \le k \le n$, `compute-Phi`$(x) = \Phi(x) + Rem(x)$.

1: $\{\bar{x}_{t+h}^{i,[k]}\}_{(i,k)\in\{(i,k):2\le i\le p, 0\le k\le n+1\}\cup\{(1,0)\}} \leftarrow$ SHIFT-PART$(\ \bar{x}_t\ )$

2: $\left\{\bar{x}_{t+h}^{1,[k]}\right\}_{1\le k\le n} \leftarrow$ COMPUTE-REP-K$(\ n-1, \bar{x}_t^{p,[0]}, .., \bar{x}_t^{p,[n-1]}, \bar{x}_{t+h}^{1,[0]})$

3: $\bar{x}_{t+h}^{0,[0]} \leftarrow \sum_{k=0}^{n} \bar{x}_{t+h}^{1,[k]} \cdot h^k$

Algorithm 9 Rem

Input: $\bar{x}_t$, $\bar{x}_{t+h}$

Output: $\bar{r}_{t+h}$

Require: $\bar{x}_t$ is a (p,n)-representation, $\bar{x}_{t+h} = \Phi(\bar{x}_t)$

Ensure: $\bar{r}_{t+h}$ is a (p,n)-representation containing nonzero entries on $(i,n+1)$ and $(0,0)$ for $1 \le i \le p$, `compute-Phi`$(x) = \Phi(x) + Rem(x)$.

1: $\{\bar{r}_{t+h}^{1,[n+1]}, a^*, b^*\} \leftarrow$ COMPUTE-REMAINDER$(\ \bar{x}_t, \bar{x}_{t+h}^{1,[0]}, .., \bar{x}_{t+h}^{1,[n]}\ )$

2: $\bar{r}_{t+h}^{0,[0]} \leftarrow \bar{r}_{t+h}^{1,[n+1]} \cdot h^{n+1}$

As it was show in [33] using the Lohner algorithm gives us chance to do C^1 computations, that is, to compute derivatives of the flow w.r.t. initial conditions. This is due to the fact that in each step of the Lohner algorithm we need to obtain the Jacobian of Φ. Using this C^1 data in the case of computing enclosures for Poincaré map images one can obtain the Jacobian of this Poincaré map. The knowledge of the Jacobian may be used to find a better first approximation to the stationary point via Newton method and/or to choose good initial coordinate frame for the Lohner set, which might be important in computer assisted proofs of covering relations. In the few next sections we want to give some formulas that will be helpful in faster computation of the Jacobian for a single step of integration and we will present method for obtaining the derivative of the flow (non-rigorously).

4.2 Jacobian of the numerical method Φ

As it was shown, construction of robust rigorous integrators based on the Lohner algorithm that avoids wrapping effect needs some method for computing Jacobian of the map $\Phi(\cdot)$ at the current point. Moreover those one-step Jacobians may be used to construct (non-rigorous) derivatives of the flow $\frac{\partial \varphi_I}{\partial x_J}(x, t_p(x))$ needed in the computation of the Jacobian of the flows and/or Poincaré Maps. A straightforward method to compute $D\Phi(a)$ is to use Automatic Differentiation for this task, but we will show that in the case of the rigorous integrator of DDE it may be a huge loss of computing time, as the form of the $\Phi(\cdot)$ allows for vast optimization. Here we will present a faster method for computing the Jacobian of Φ for a single step $h = \frac{1}{p}$ than by using Automatic Differentiation on all coefficients of the representation.

From now we will use the following notation in subscripts to simplify notation: $(i,k) := index(i,k)$ as defined by the *index* function from Definition 12. For a representations $\bar{x}$ and $\bar{x}_h = \Phi(\bar{x})$ we will write $\Phi_{(i,k)}(\bar{x})$ to indicate $\bar{x}_h^{i,[k]}$. Moreover, we will use capital letters (eg. I, J) as the indices where it is not so important to remember grid point i and order k. However, we will remember that $I = index(i,k)$ for some unique i and k - the value of i and k may be obtained from I via inverse function $index^{-1}$:

$$\begin{aligned} index^{-1}(1) &= (0,0), \\ index^{-1}(I) &= ((I-2) \div (n+2) + 1, (I-2) \ \% \ (n+2)), \quad I > 1, \end{aligned} \tag{4.8}$$

where $\%$ and $\div$ are integer modulo and division operators:

$$a \div b = \left\lfloor \frac{a}{b} \right\rfloor, \quad a, b \in \mathbb{N} \tag{4.9}$$

$$a \ \% \ b = a - b \cdot (a \div b), \quad a, b \in \mathbb{N} \tag{4.10}$$

We will write N as the size of the representation, that is $N := (n+2) \cdot p + 1$. The Jacobian of Φ at $\bar{x}$ (denoted by $D\Phi(\bar{x})$) is a $N \times N$ matrix of partial derivatives $\frac{\partial \Phi_I}{\partial x_J}(\bar{x})$ for $I, J \in \{1, ..., N\}$.

The integrator method $\Phi(\cdot)$ presented in algorithm 8 was divided into two parts: the *shift* part presented in Algorithm 3 and the *forward* part that uses procedure `compute-rep-k` from the algorithm 4 and the Taylor sum that computes $\bar{x}_h^{0,[0]}$. From Algorithm 3 it is clear that:

$$\frac{\partial \Phi_{(i,k)}}{\partial x_{(j,l)}}(\bar{x}) = 1,\ j = i-1,\ l = k, \tag{4.11}$$

$$\frac{\partial \Phi_{(i,k)}}{\partial x_{(j,l)}}(\bar{x}) = 0, \quad otherwise, \tag{4.12}$$

for $i \geq 2$ and:

$$\frac{\partial \Phi_{(1,0)}}{\partial x_{(j,l)}}(\bar{x}) = 1,\ j = 0,\ l = 0, \tag{4.13}$$

$$\frac{\partial \Phi_{(1,0)}}{\partial x_{(j,l)}}(\bar{x}) = 0, \quad otherwise. \tag{4.14}$$

Using those equalities we saved computational time for many of the coefficients of the Jacobian $D_x\Phi(\bar{x})$ reducing the size of the problem from $O\left((p \cdot n)^2\right)$ to $O(p \cdot n^2)$. Namely, we need only to compute $\frac{\partial \Phi_{(1,k)}}{\partial x_{(j,l)}}(\bar{x})$ for $k > 1$ (that is, $(n+1) \cdot N$ elements) and $\frac{\partial \Phi_{(0,0)}}{\partial x_{(j,l)}}(\bar{x})$ (that is, $1 \cdot N$ elements), which comes from the forward part. We will further show that we may again set some of them explicitly to 0.

From the formulation of Algorithm 4 we see that $\Phi_{(1,k)}$ depends on $k+1$ variables, namely $\bar{x}_0^{0,[0]}$ and k values in the past; $\bar{x}_0^{p,[0]}, ..., \bar{x}_0^{p,[k-1]}$. For $k = n$ it gives maximum of $n+1$ variables $x_{(0,0)},\ x_{(p,0)}, ..., x_{(p,n-1)}$. For all others, we can set $\frac{\partial \Phi_{(1,k)}}{\partial x_{(j,l)}}(\bar{x})$ to 0, thus reducing number of elements to $(n+1) \cdot (n+1) \in O(n^2)$.

Finally, the formula for $\frac{\partial \Phi_{(0,0)}}{\partial x_{(j,l)}}(\bar{x})$ comes from the fact that $\Phi_{(0,0)}(\bar{x})$ is simply a Taylor summation of $\bar{x}_h^{1,[k]}$ and from the fact that $\bar{x}_h^{1,[0]} = \Phi_{(1,0)}(\bar{x}_0) = \bar{x}_0^{0,[0]}$ (see Algorithm 3). As we stated previously, each $\bar{x}_h^{1,[k]}$ depends only on $\bar{x}_0^{0,[0]}$ and $\bar{x}_0^{p,[0]}, ..., \bar{x}_0^{p,[k-1]}$, so the value $\Phi_{(0,0)}(\bar{x})$ depends also on $\bar{x}_0^{0,[0]}$ and $\bar{x}_0^{p,[0]}, ..., \bar{x}_0^{p,[n-1]}$. Finally, the a^* coefficient in the sum is dependent also on $\bar{x}_0^{p,[n]}$. Thus, we obtain $O(n^2)$ coefficients in the Jacobian that are needed to be computed.

The graphical idea of the dependence of the elements of the Jacobian for a single step method is shown in Figure 4.2.

All those computations allowed us to reduce the input and output size to the Automatic Differentiation procedure to the $O(n)$ and $O(n^2)$ respectively. Thus, the problem of computing the Jacobian in the each step of the integration process does not depend on the number of interpolation points but only on the order of the interpolation. Still, it has an impact on the size of the Jacobian matrix, and thus on the complexity of basic matrix operations performed in each step of the Lohner algorithm.

	(0,0)	(1,0)	(1,1)	(1,2)	(1,3)	(2,0)	(2,1)	(2,2)	(2,3)	(3,0)	(3,1)	(3,2)	(3,3)
(0,0)	⋆		⋆	⋆						⋆	⋆	⋆	
(1,0)	⋆												
(1,1)	⋆									⋆			
(1,2)	⋆									⋆	⋆		
(1,3)													
(2,0)		1											
(2,1)			1										
(2,2)				1									
(2,3)					1								
(3,0)						1							
(3,1)							1						
(3,2)								1					
(3,3)									1				

Figure 4.2: Dependence of the elements of the Jacobian for the single step method $\Phi(\cdot)$ for a (3,2)-representation. Star ($\star$) represents some meaningful value, empty space represent value 0. As we can see the matrix is singular, as it has empty last column and $n+2$-nd row. This is due to the fact that those elements corresponds to the remainders $\bar{x}_0^{p,[n+1]}$ (which is not used in the Φ procedure) and $\bar{x}_h^{1,[n+1]}$ (which is computed in the *Rem* procedure), respectively.

4.3 Jacobian of a Poincaré map

4.3.1 Reduced Poincaré map and it's Jacobian matrix

As a side effect of using the Lohner set representation, we are able to compute Jacobian of the Poincaré map associated with the semiflow φ and some section S. We will assume that S is representable, in the sense of Definition 17:

Definition 17 *We say that an (s,a)-section S is* (p,n)-representable *if there exist a finite dimensional mapping $\bar{s} : \mathbb{R}^M \to \mathbb{R}$ for $M = 1 + p \cdot (n+2)$ such that for any (p,n)-representation $\bar{f}$ we have:*

$$\bar{s}\left(\bar{f}\right) = s\left(Supp(\bar{f})\right) \tag{4.15}$$

We call the set $\bar{S} = \{x \in \mathbb{R}^M : \bar{s}(x) = a\}$ the representation of the (s,a)-section S.

Remark 16 *An example of a (p, n)-representable section for any n and p is an (s,a)-section S defined with $s(f) := f(0)$ and with any $a \in \mathbb{R}$. The corresponding finite dimensional section is then defined simply with $\bar{s}(x) = \pi_1(x)$.*

An example of a (2, n)-representable but not (3, n)-representable section for any $n > 1$ is an (s, a)-section S defined with $s(f) := f\left(\frac{1}{2}\right)$ and with any $a \in \mathbb{R}$.

For the simplicity of notation from now on we are using symbol P to denote Poincaré map $P_{S,\varphi}$ and we assume that S and φ are given and we assume that we have some (n,p)-representable (s,a)-section S with representation $\bar{s}$. We assume that we are working in some neighbourhood $\bar{f}$ of a function g such that $P_{\varphi,S}(g) = g$ and we assume that $t_p(f)$ exists for any $f \in Supp(\bar{f})$ and $t_p(f) \geq 1$.

Let $g \in \bar{g}$ and let x^g be a solution to the equation (3.1) with initial function g. We have $P_{\varphi,S}(g) = \varphi\left(g, t_p(g)\right) = x^g_{t_p(g)}$ and we can define $x \in \mathbb{R}^M$ and $P(x) \in \mathbb{R}^M$ for $M = (n+2) \cdot p + 1$ such that:

$$\begin{aligned}
x_{(0,0)} &= g(0) && (4.16)\\
x_{(i,k)} &= g^{[k]}\left(-i \cdot h\right) && (4.17)\\
P(x)_{(0,0)} &= x^g_{t_p(g)}(0) && (4.18)\\
P(x)_{(i,k)} &= \frac{1}{k} \cdot \frac{\partial x^g_{t_p(g)}}{\partial t}\left(-i \cdot h\right) && (4.19)
\end{aligned}$$

for $1 \leq i \leq p$, $0 \leq k \leq n$. We call the map $P(\cdot)$ a *reduced Poincaré map*. We see that the reduced Poincaré map is hard to compute explicitly, but we can compute it rigorously using the integrator scheme. Since it closely resembles real Poincaré map $P_{\varphi,S}$, but exists in a finite dimensional space closely resembling our space of (p,n)-representations, we can use its Jacobian matrix to achieve two important goals:

- find better approximation to a priori chosen initial set $\bar{g}$, for example using nonrigorous Newton algorithm,
- choose better initial coordinate frame for the Lohner set representation, by computing (nonrigorously) eigen vectors of the Jacobian matrix DP.

Remark 17 *The value $t_p(x)$ usually is not a multiplicity of base step h, thus it is neccessary to allow other time steps in the integration process. We will deal with this problem in Chapter 5.*

In this section, we denote by x the element of $\mathbb{R}^M$ and by x_J the J-th element of the vector x, that is $x_J = \pi_J(x)$, $J \in \{1, ..., M\}$. For the Poincaré map P we will denote by P_I the I-th component of P, that is a function $P_I : \mathbb{R}^M \to \mathbb{R}$, $P_I(x) = \pi_I(P(x))$. Using equation (4.8) we know that I and J are given by:

$$I = index(i,k) \tag{4.20}$$
$$J = index(j,l) \tag{4.21}$$

for some i, k, j, l. We will use indices I, J to simplify notation where it is appropriate. For an index I we denote by

$$\varphi_I(x,t) = x_t(0), \quad I = index(0,0) \tag{4.22}$$
$$\varphi_I(x,t) = \frac{1}{k} \cdot \frac{\partial x_t}{\partial t}\left(-i \cdot \frac{1}{p}\right), \quad I = index(i,k), \tag{4.23}$$

thus we can simply write:

$$P(x)_I = \varphi_I\left(x, t_p(x)\right). \tag{4.24}$$

Now, for I and J, $1 \leq I, J \leq M$ we want to compute:

$$\frac{\partial P_I}{\partial x_J} = \frac{\partial \varphi_I}{\partial x_J}\left(x, t_p(x)\right) + \frac{\partial \varphi_I}{\partial t}\left(x, t_p(x)\right) \cdot \frac{\partial t_p}{\partial x_J}\left(x\right) \tag{4.25}$$

This equation resembles the standard formula in the ODE setting [33]. The first term $\frac{\partial \varphi_I}{\partial x_J}\left(x, t_p(x)\right)$ may be computed numerically using single-step Jacobians of the integrator - we call it derivative of the flow. The second term $\frac{\partial \varphi_I}{\partial t}\left(x, t_p(x)\right) \cdot \frac{\partial t_p}{\partial x_J}\left(x\right)$ is called a correction term. It can be computed using the section S and the *implicit function theorem*. Here we would like to reformulate the calculations presented in [33] to the case of DDEs as there are some minor subtleties in the equations.

4.3.2 Correction to the Jacobian of the Poincaré Map

We would like to compute the correction term $\frac{\partial \varphi_I}{\partial t}(x, t_p(x)) \cdot \frac{\partial t_p}{\partial x_J}(x)$.

The term $\frac{\partial \varphi_I}{\partial t}(x, t_p(x))$ may be obtained from equation (3.1). In the ODE setting, it corresponds to the vector field where simply we have $\frac{\partial \varphi_I}{\partial t}(x, t_p(x)) = F_I(P(x_0))$. In DDE case it is more involved as in our case r.h.s of equation $x' = f(x(t-1), x(t))$ depends only on the past and present value of the function and it returns only single scalar (so we do not in fact have f_I). We need to look closer at the $\varphi_I(x, t_p(x))$.

By equation (4.24) we have:

$$\varphi_I(x_0, t_p(x_0)) = \frac{1}{k!} \cdot \frac{\partial^k x}{\partial t^k}\left(t_p(x_0) - \frac{i}{p}\right) \tag{4.26}$$

where $I = index(i,k)$ and x on the r.h.s is a solution to DDE on $[-1, t_p(x_0)]$, with the initial function x_0. Now we have:

$$\frac{1}{k!}\frac{\partial^k x}{\partial t^k}\left(t_p(x_0) - \frac{i}{p}\right) = \frac{1}{k!} \cdot \frac{\partial^k x_{t_p(x_0)}}{\partial t^k}\left(-\frac{i}{p}\right), \tag{4.27}$$

that is, we have changed a solution x to a point $x_{t_p(x_0)}$ in the phase-space C^0, and we changed time variable accordingly. We remember that $P(x) = x_{t_p(x_0)}$. Now we compute:

$$\frac{\partial \varphi_I}{\partial t}(x_0, t_p(x_0)) = \frac{\partial}{\partial t}\frac{1}{k!} \cdot \frac{\partial^k x_{t_p(x_0)}}{\partial t^k}\left(-\frac{i}{p}\right) = \tag{4.28}$$

$$= \frac{1}{k!} \cdot \frac{\partial^{k+1} x_{t_p(x_0)}}{\partial t^{k+1}}\left(-\frac{i}{p}\right) = \tag{4.29}$$

$$= (k+1) \cdot \frac{1}{(k+1)!} \cdot \frac{\partial^{k+1} x_{t_p(x_0)}}{\partial t^{k+1}}\left(-\frac{i}{p}\right). \tag{4.30}$$

Now, for $k < n$ we have:

$$\frac{\partial \varphi_I}{\partial t}(x_0, t_p(x_0)) = (k+1) \cdot \varphi_{I+1}(x_0, t_p(x_0)) = (k+1) \cdot P_{I+1}(x_0) \tag{4.31}$$

because $index(i, k+1) = index(i,k) + 1 = I + 1$. For $k = n$ we may use:

$$\frac{\partial \varphi_I}{\partial t}(x_0, t_p(x_0)) = \frac{1}{n} \cdot F^{[n]}\left(x^{[0]}(t_p(x_0) - 1), .., x^{[n]}(t_p(x_0) - 1), x^{[0]}(t_p(x_0)), x^{[n]}(t_p(x_0))\right) \tag{4.32}$$

This is the reason why we have assumed that $t_p(x) \geq 1$ at the beginning of this section.

As in [33] to compute $\frac{\partial t_p}{\partial x_J}(x)$ we use section S and the implicit function theorem. We assume that S has a (p,n)-representation $\bar{s}$. We have:

$$\bar{s}(P(x_0)) = 0 \Rightarrow \bar{s}(\varphi(x_0, t_p(x_0))) = 0 \tag{4.33}$$

taking derivative $\frac{\partial}{\partial x_J}$ we get:

$$\frac{\partial}{\partial x_J}\bar{s}\left(\varphi(x_0, t_p(x_0))\right) = 0 \tag{4.34}$$

$$\left\langle \nabla\bar{s}\left(P(x_0)\right), \left(\frac{\partial}{\partial x_J}\varphi\left(x_0, t_p(x_0)\right)\right)\right\rangle = 0 \tag{4.35}$$

The . in the above equation is a standard scalar product in $\mathbb{R}^M$. We use (4.25) to get:

$$\left\langle \nabla\bar{s}\left(P(x_0)\right), \left(\frac{\partial\varphi}{\partial x_J}\left(x, t_p(x)\right) + \frac{\partial\varphi}{\partial t}\left(x, t_p(x)\right)\cdot\frac{\partial t_p}{\partial x_J}\left(x\right)\right)\right\rangle = 0, \tag{4.36}$$

where by $\frac{\partial\varphi}{\partial x_J}$ we mean a vector of derivatives $\left(\frac{\partial\varphi_1}{\partial x_J}, ..., \frac{\partial\varphi_N}{\partial x_J}\right)$. After reorganization:

$$\left\langle \nabla\bar{s}\left(P(x_0)\right), \frac{\partial\varphi}{\partial t}\left(x, t_p(x)\right)\right\rangle \cdot \frac{\partial t_p}{\partial x_J}\left(x\right) = -\left\langle \nabla\bar{s}\left(P(x_0)\right).\frac{\partial\varphi}{\partial x_J}\left(x, t_p(x)\right)\right\rangle, \tag{4.37}$$

finally:

$$\frac{\partial t_p}{\partial x_J}\left(x\right) = -\frac{\left\langle \nabla\bar{s}\left(P(x_0)\right), \frac{\partial\varphi}{\partial x_J}\left(x, t_p(x)\right)\right\rangle}{\left\langle \nabla\bar{s}\left(P(x_0)\right), \frac{\partial\varphi}{\partial t}\left(x, t_p(x)\right)\right\rangle}. \tag{4.38}$$

4.4 Doubleton Lohner set performance in rigorous DDE integration

We have performed the same tests as in Section 3.3, but instead using interval sets we have used the doubleton Lohner set (Lohner representation 4): $x = x_0 + C \cdot r_0 + r$, where r is an interval set.

We run tests as in the case of basic tests in Section 3.3. Here we briefly present some of the results, rest of them can be found on the author's web page. Full history (after $2 \cdot p$ initial steps in case of stationary solution) of the integration process for (8,7)-representation for both stationary and periodic solution is presented, respectively, in Figures 4.3 and 4.9 The history recorded every p steps (after $2 \cdot p$ initial steps in case of stationary solution) is presented in Figures 4.4 and 4.10. Also we present the solutions together with all derivatives up to order $k = 8$ over the whole time interval $t = [-1, p \cdot (n+1)]$ ($t = [p, p \cdot (n+1)]$ in the case of stationary solution) in Figures 4.5, 4.11 and 4.12. The first one is for the stationary solution to equation (2.14), the two others are for periodic solution to equation (2.15). The dependence of the resulting set diameter on the diameter of the initial set in test 3 is presented in Figures 4.6 and 4.13. The influence of the diameter of the resulting set on the choice of the parameter p (the grid spacing) is presented in Figures 4.7 and 4.14.

We see that the strong contraction on the high order coefficients is maintained and we also get the contraction for all or almost all other coefficients, contrary to the interval set representation. To better see the advantage of the Lohner set over the interval set we also included comparison of the set representations in Figures 4.8 and 4.15. The advantage of using Lohner sets is more clear in the case of periodic orbit than in the case of strongly attracting stationary solution. Still, in both cases, we obtain better estimates on the diameter of the set if we use doubleton Lohner set representation.

Remark 18 *We have also done numerical experiments for doubleton Lohner sets, where the error part r was represented by the cuboid set (QR decomposition was used) instead of interval set. Surprisingly, this approach gave worse results and sometimes behaved as classical interval set. We attribute this strange behaviour to the problem of computing QR decomposition for matrices of very big dimensions (in our case the dimension of the Jacobian matrix is $p \cdot (n+2)+1$). This phenomena should be studied in the future.*

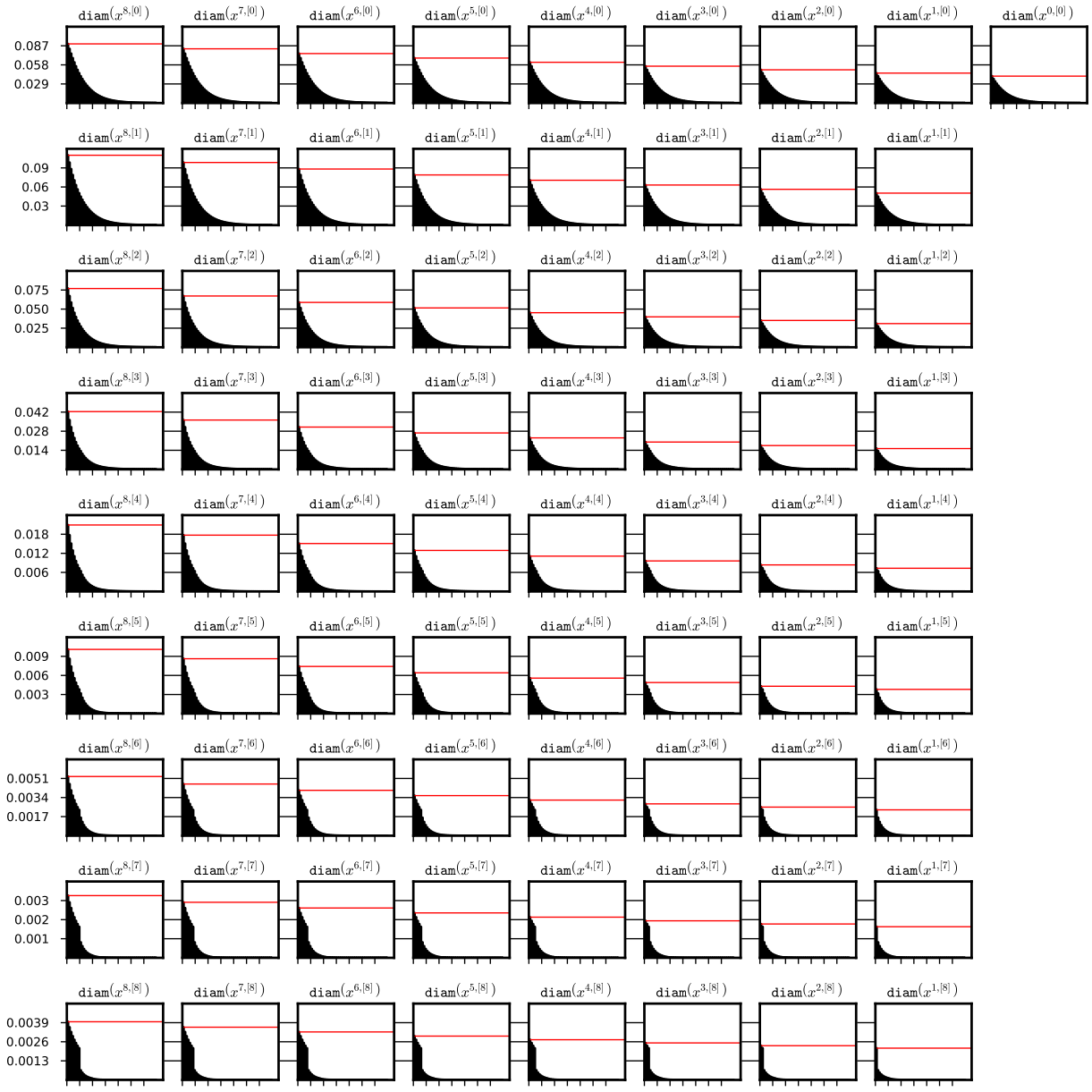

Figure 4.3: Diameters of the coefficients of a sequence $\{\Phi^n(\bar{x}_0)\}_{n\in\{16,..,8\cdot(7+2)\}}$ (a full history after $2\cdot p$ steps) for some (8,7)-representation $\bar{x}_0$ of a stable stationary solution $x \equiv 0$ for system (2.14). Red horizontal line marks the diameter of the representation of the initial function. On the x-axis we have the iteration steps, each tick represents p steps of iteration. The data from test 1c was used. System (2.14), doubleton Lohner set representation and (8,7)-representation were used for the integration process. The data is stored in the file `steady_08_07_out_3/rect_di.txt`.

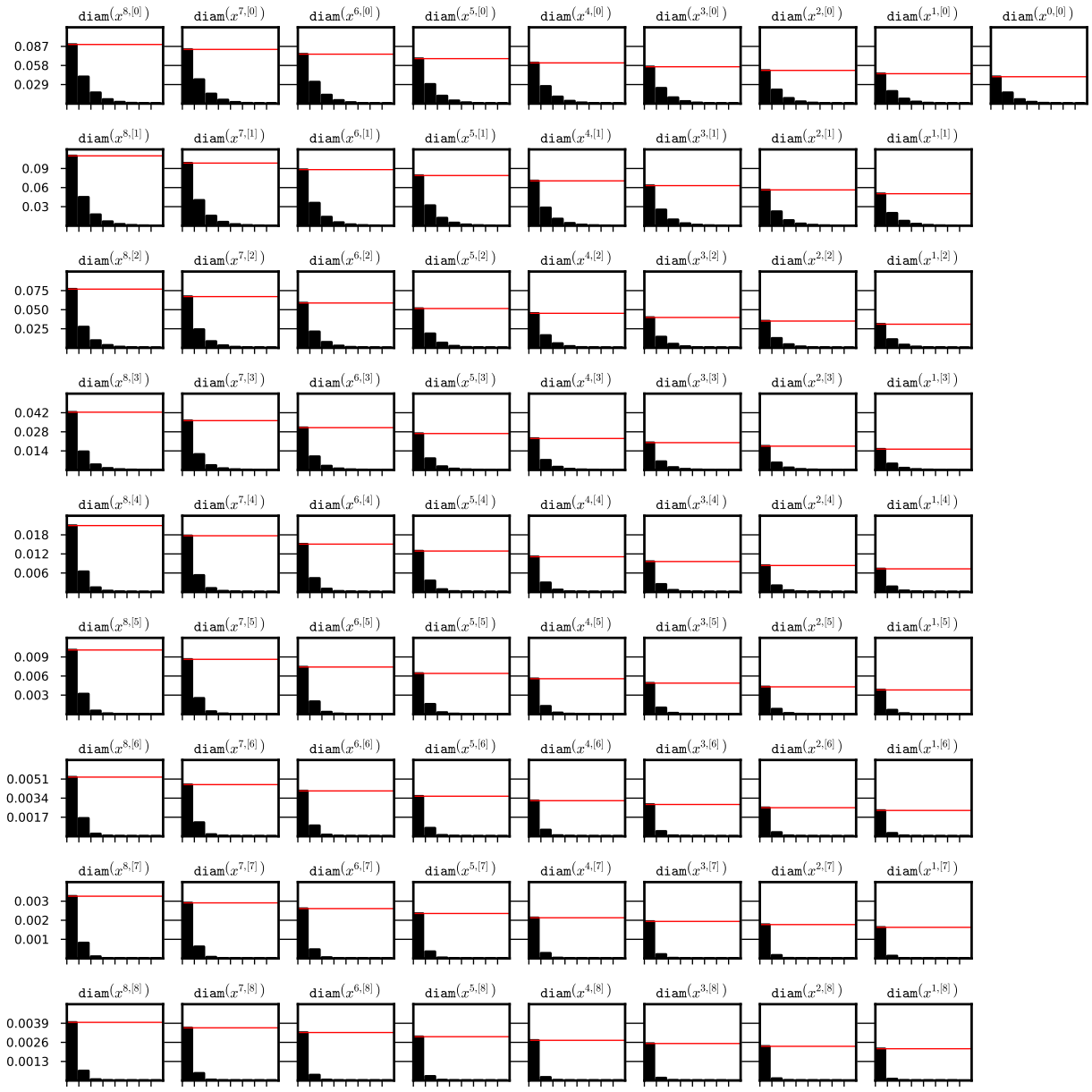

Figure 4.4: Diameters of the coefficients of a sequence $\left\{\Phi^{8\cdot n}(\bar{x}_0)\right\}_{n\in\{2,..,(7+2)\}}$ for some (8,7)-representation $\bar{x}_0$ of a stable stationary solution $x \equiv 0$ for system (2.14). Red horizontal line marks the diameter of the representation of the initial function after $2 \cdot p$ steps. On the x-axis we have the iteration steps, each tick represents p steps of iteration. The data from test 1c was used. System (2.14), doubleton Lohner set representation and (8,7)-representation were used for the integration process. The data is stored in the file `steady_08_07_out_3/rect_di.txt`.

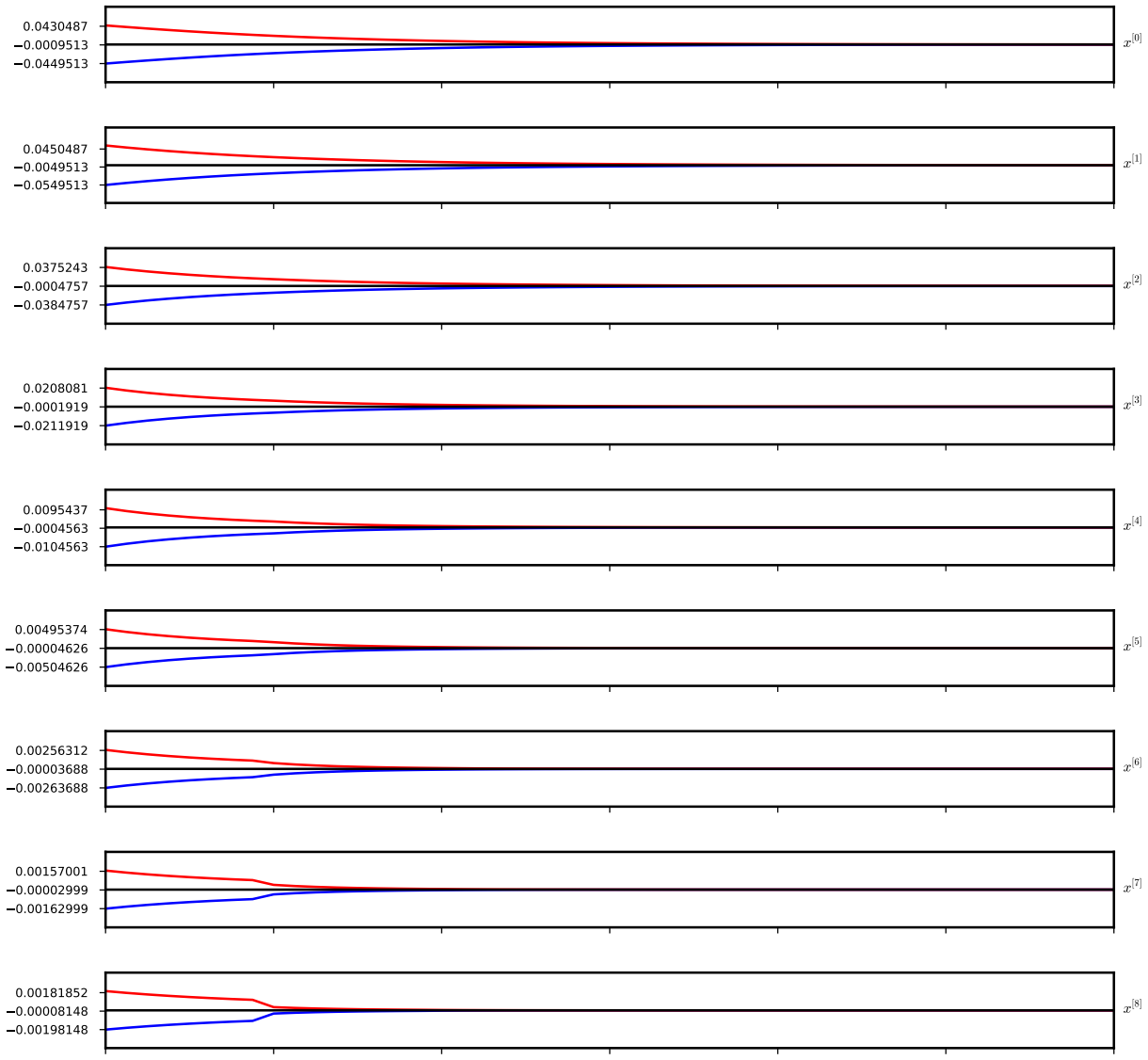

Figure 4.5: A graph of the solution $x(t) \equiv 0$ to (2.14), together with the all derivatives up to order $k = 7$ (black lines). Blue and red lines present lower and upper bounds respectively. For $k = 8$ we present the bound on the 8-th derivative on the intervals of the length $h = \frac{1}{p}$. On x axis we have time t. The data is as in Figure 4.3.

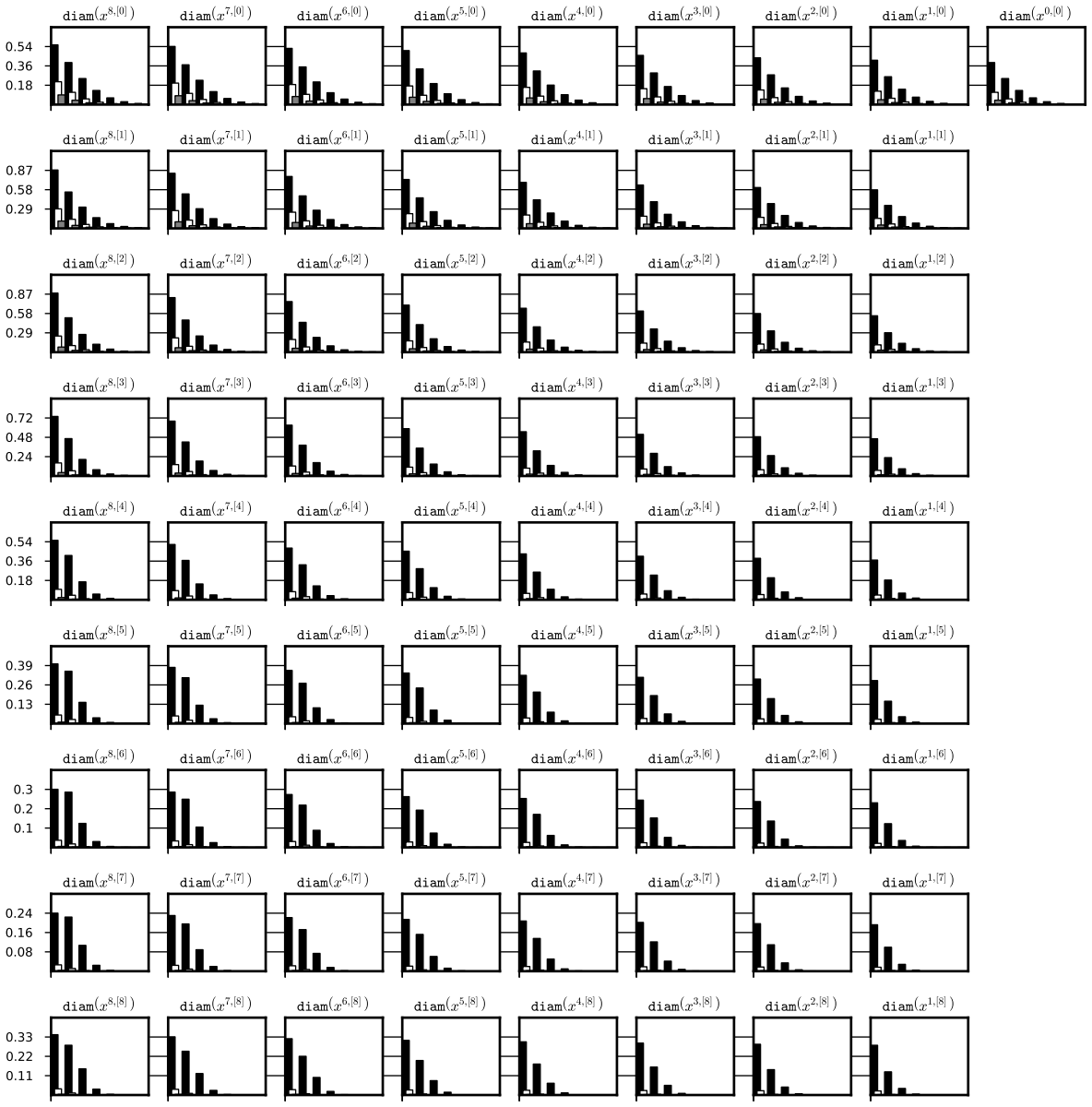

Figure 4.6: Dependence of the diameter of the interval set on the initial data diameter. After initial $2 \cdot p$ steps a history of the integration of some neighbourhood of a stable stationary solution $x \equiv 0$ to system (2.14) was recorded every p steps for three runs with initial data of decreasing diameter. On the x-axis we have the iteration steps, each bar is a diameter of the representation coefficient after p steps of iteration. The data is generated for tests 1a, 1b, 1c (black, white, gray respectively). System (2.14), doubleton Lohner set representation and (8,7)-representation were used for the integration process. The data is stored the files `steady_08_07_out_1/rect_di.txt`, `steady_08_07_out_2/rect_di.txt` and `steady_08_07_out_3/rect_di.txt` respectively.

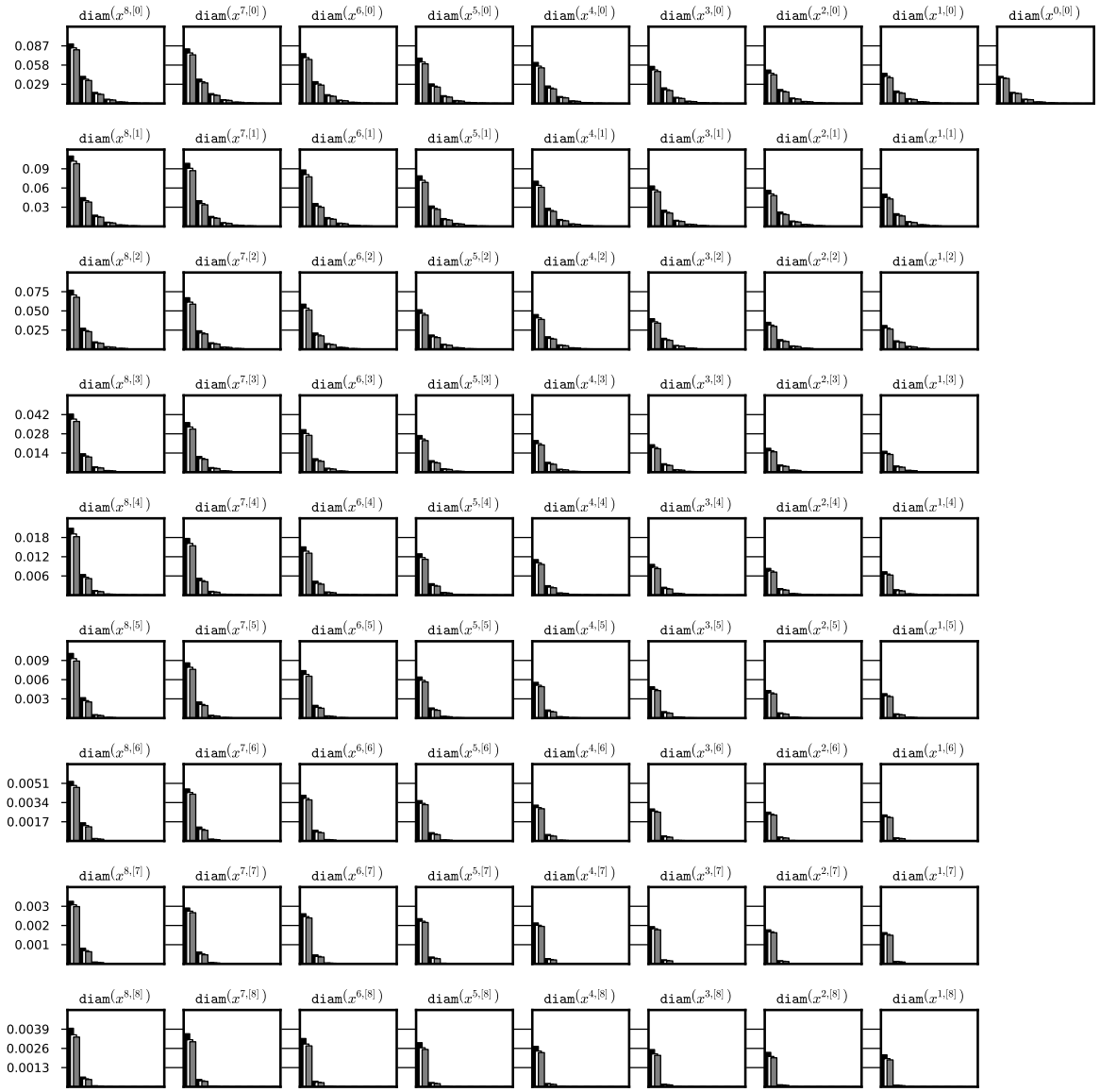

Figure 4.7: Dependence of the diameter of the representation on the grid size p. A history of one integration of three representations of a stable stationary solution $x \equiv 0$ for system (2.14) for parameter $p \in \{8, 16, 32\}$ was recorded every p steps after initial $2 \cdot p$ steps (black, white, gray respectively). The diameters of corresponding representation coefficients (i.e. that represents appropriate derivative at the same time t) are drawn for comparision. Each bar is a diameter of the representation coefficient after p steps of iteration. The tests used are: 1c, 4, 5 (black, white, gray respectively). For all integrations the system (2.14) and doubleton Lohner set representation were used. The data is stored the files `steady_08_07_out_3/rect_di_p.txt`, `steady_16_07_out/rect_di_p.txt` and `steady_32_07_out/rect_di_p.txt` respectively.

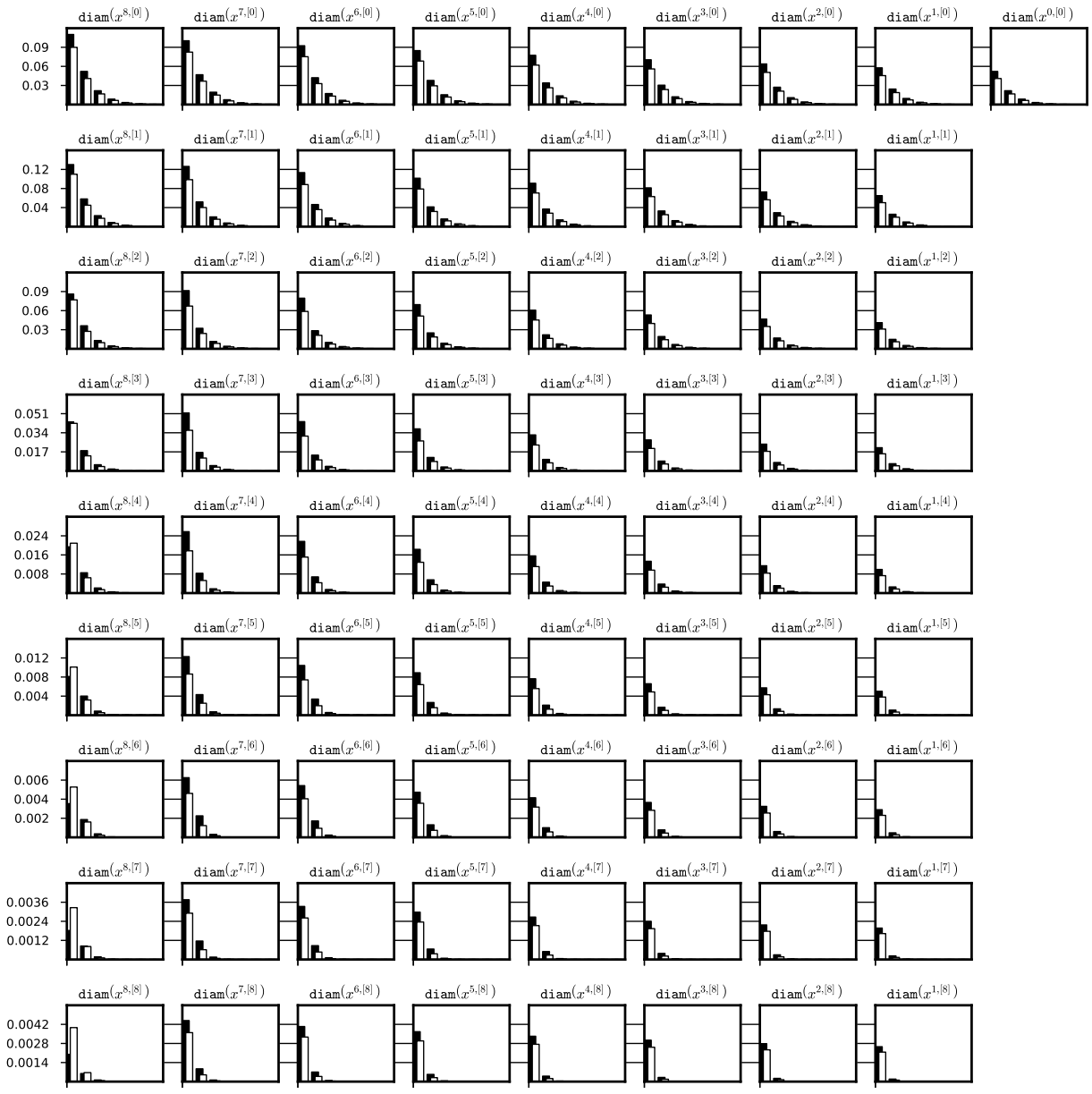

Figure 4.8: Comparision between basic interval numeric method (black) and a Lohner set representation by the doubleton $x_0 + C \cdot r_0 + B \cdot r$ (white). In both cases we have integrated the same initial representation of a stable stationary solution $x \equiv 0$ to system (2.14) and we have used interval set representation and (8,7)-representation. On the chart we present the diameter of the interval hull of each representation coefficient every 8 steps of the integration. The data from test 1c was used. The data is stored in the file `steady_08_07_out_3/int_di.txt`.

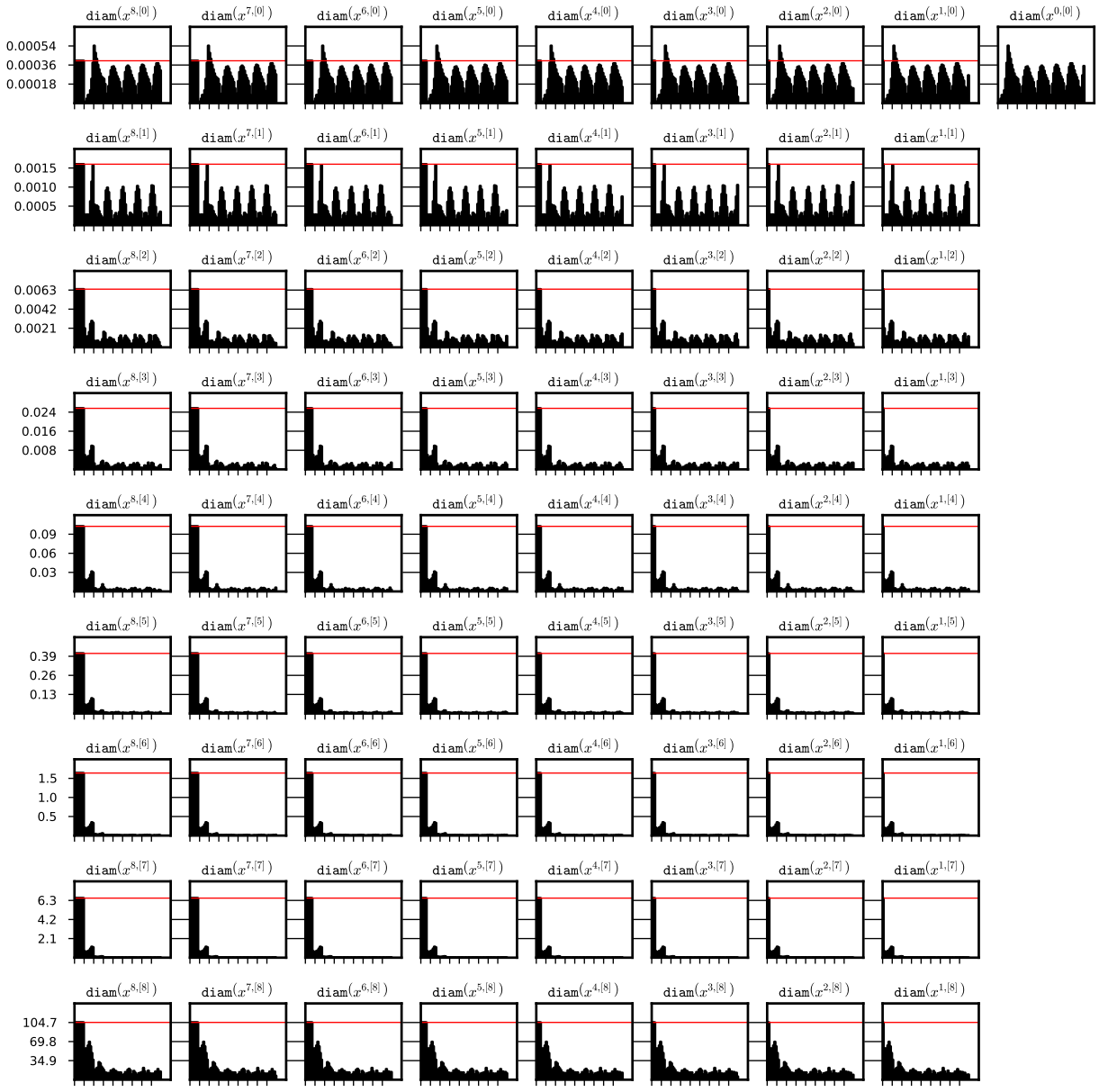

Figure 4.9: Diameters of the coefficients of a sequence $\{\Phi^n(\bar{x}_0)\}_{n\in\{0,..,8\cdot(7+2)\}}$ (a full history) for some (8,7)-representation $\bar{x}_0$ of a stable periodic orbit for system (2.15). Red horizontal line marks the diameter of the representation of the initial function. On the x-axis we have the iteration steps, each tick represents p steps of iteration. The data from test 1c was used. System (2.15), doubleton Lohner set representation and (8,7)-representation were used for the integration process. The data is stored in the file `periodic_08_07_out_3/rect_di.txt`.

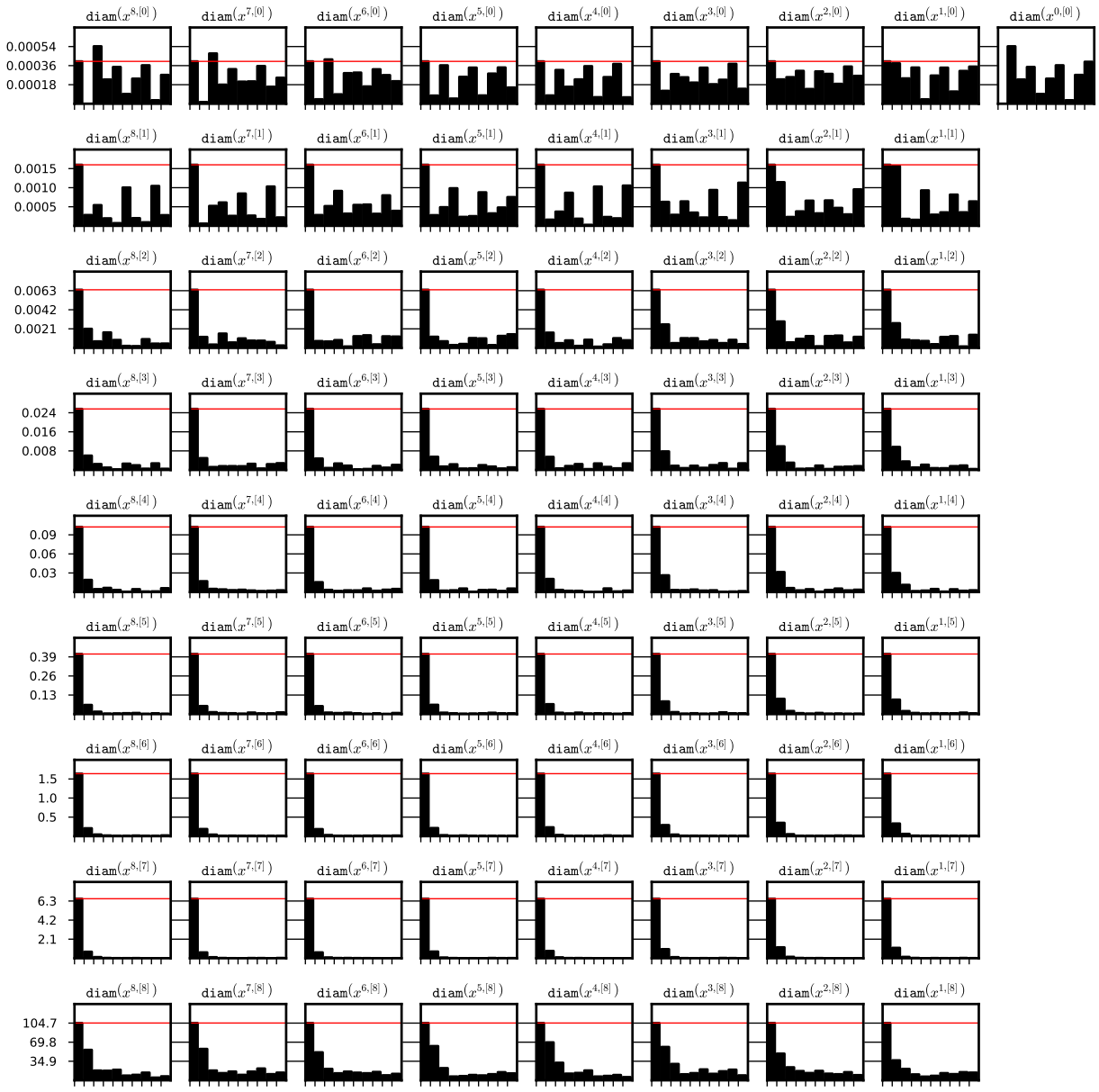

Figure 4.10: Diameters of the coefficients of a sequence $\left\{\Phi^{8\cdot n}(\bar{x}_0)\right\}_{n\in\{0,..,(7+2)\}}$ for some (8,7)-representation $\bar{x}_0$ of a stable periodic orbit for system (2.15). Red horizontal line marks the diameter of the representation of the initial function. On the x-axis we have the iteration steps, each tick represents p steps of iteration. The data from test 1c was used. System (2.15), doubleton Lohner set representation and (8,7)-representation were used for the integration process. The data is stored in the file `periodic_08_07_out_3/rect_di.txt`.

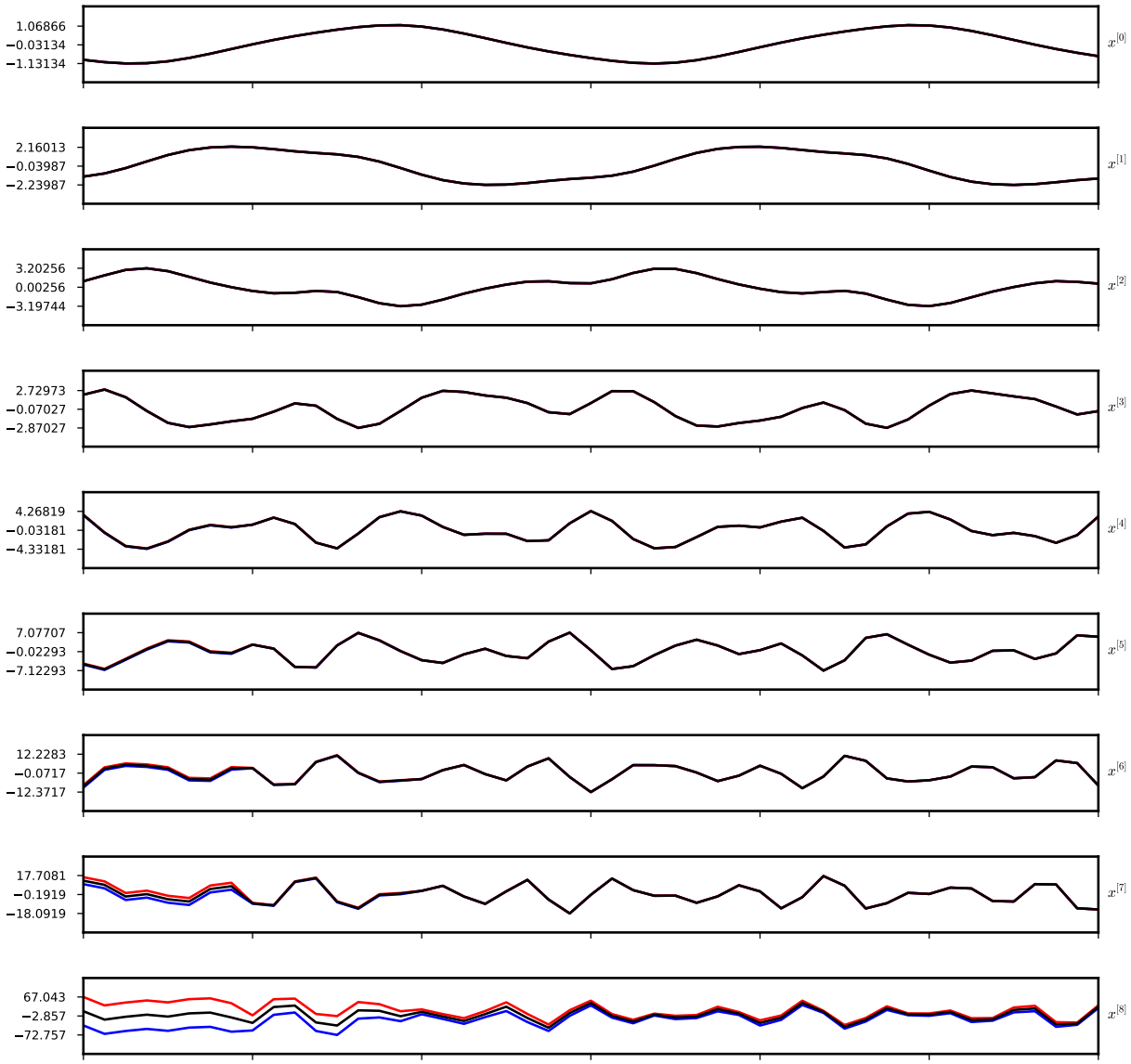

Figure 4.11: A graph of the stable periodic solution $x(t)$ to (2.15), together with the all derivatives up to order $k = 7$ (black lines). Blue and red lines present lower and upper bounds respectively. For $k = 8$ we present the bound on the 8-th derivative on the successive intervals of the length $h = \frac{1}{p}$. On x axis we have time t. The data is as in Figure 4.9.

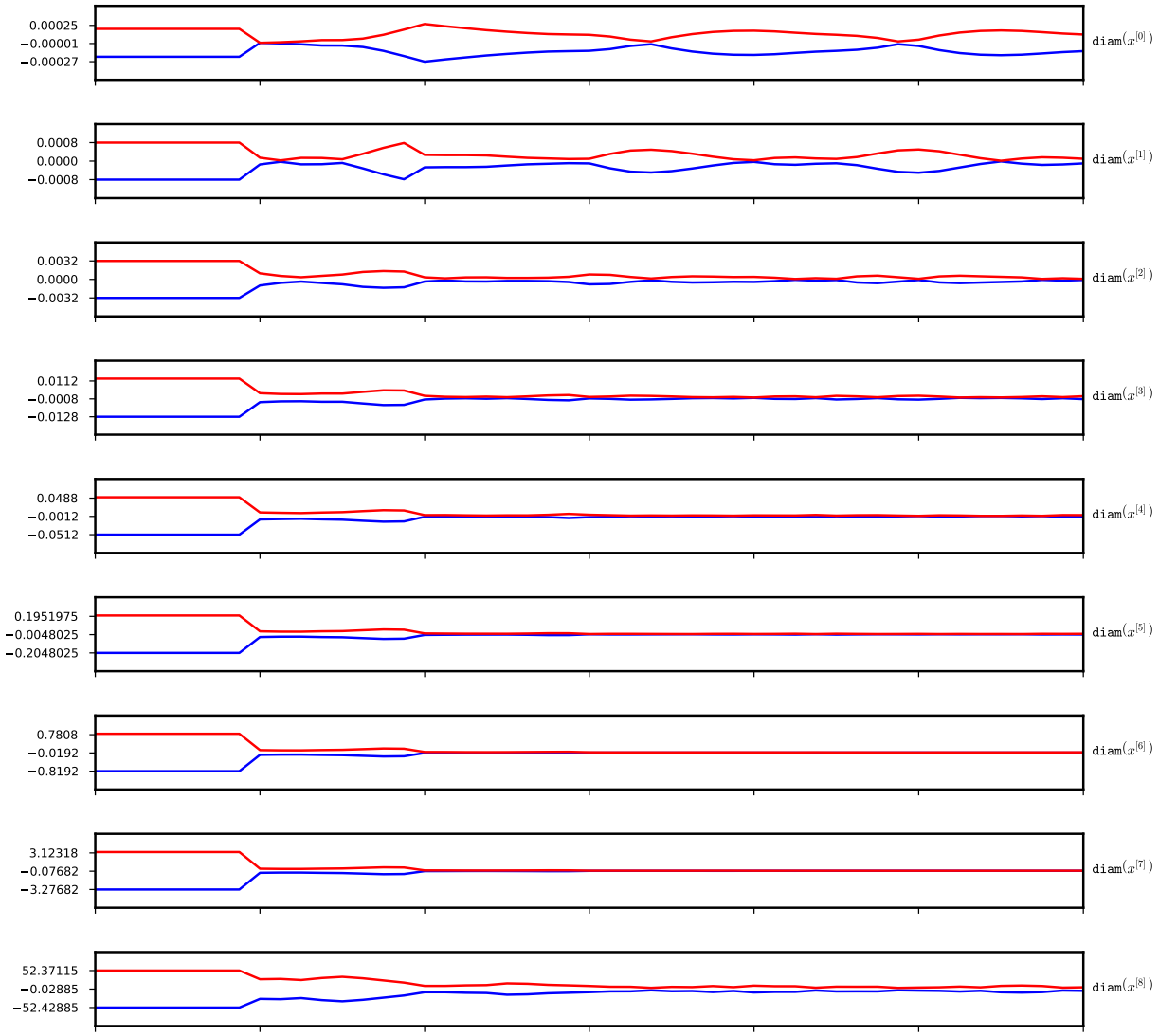

Figure 4.12: A graph of the lower and upper bounds for the stable periodic solution $x(t)$ to (2.15) and its derivatices shifted by the numerical approximation to the solution. On x axis we have time t. The data is as in Figure 4.9.

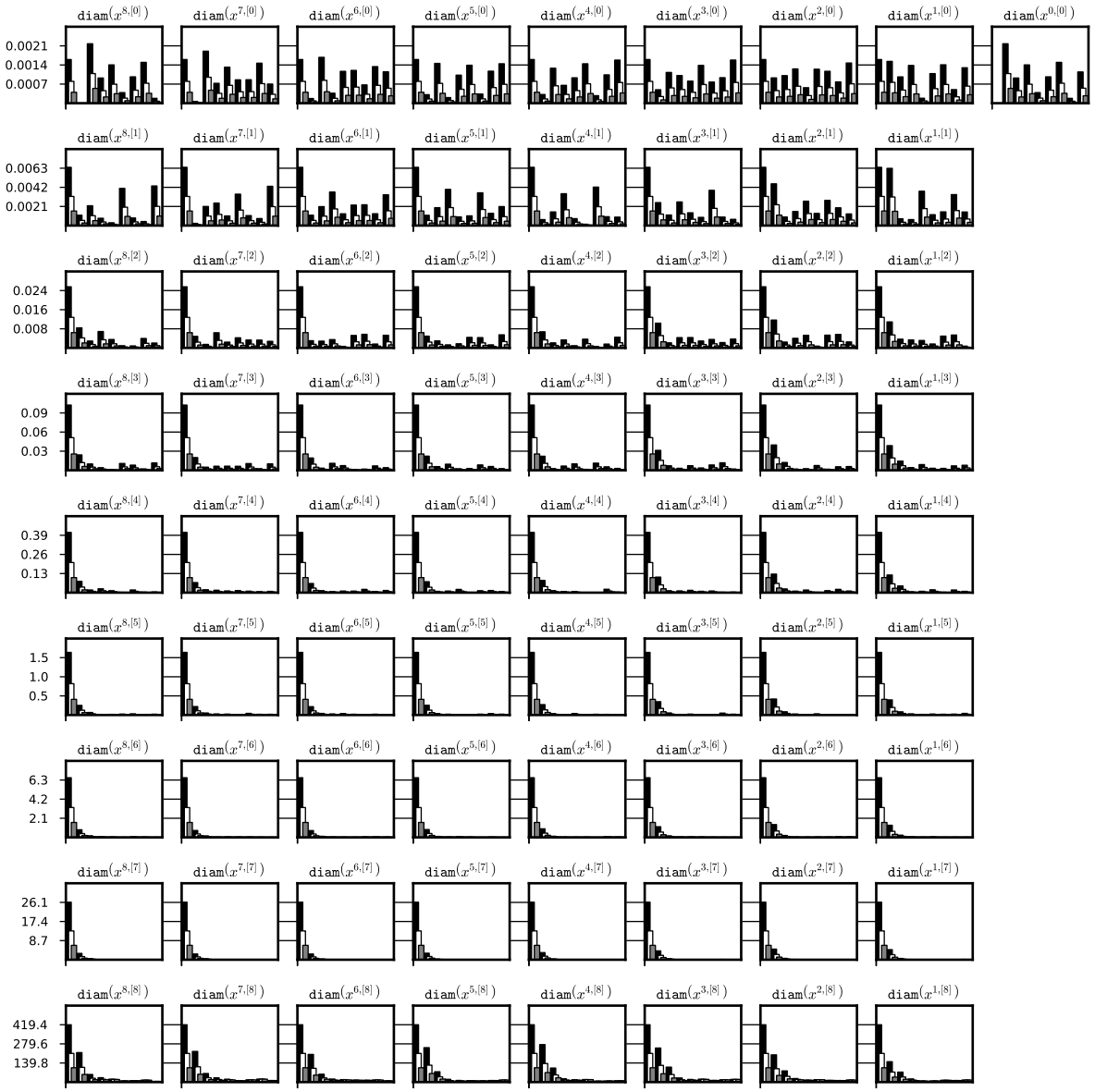

Figure 4.13: Dependence of the diameter of the interval set on the initial data diameter. A history of the integration of some neighbourhood of a stable periodic orbit for system (2.15) was recorded for three runs with initial data of decreasing diameter. On the x-axis we have the iteration steps, each bar is a diameter of the representation coefficient after p steps of iteration. The data is generated for tests 1a, 1b, 1c (black, white, gray respectively). System (2.15), doubleton Lohner set representation and (8,7)-representation were used for the integration process. The data is stored the files `periodic_08_07_out_1/rect_di.txt`, `periodic_08_07_out_2/rect_di.txt` and `periodic_08_07_out_3/rect_di.txt` respectively.

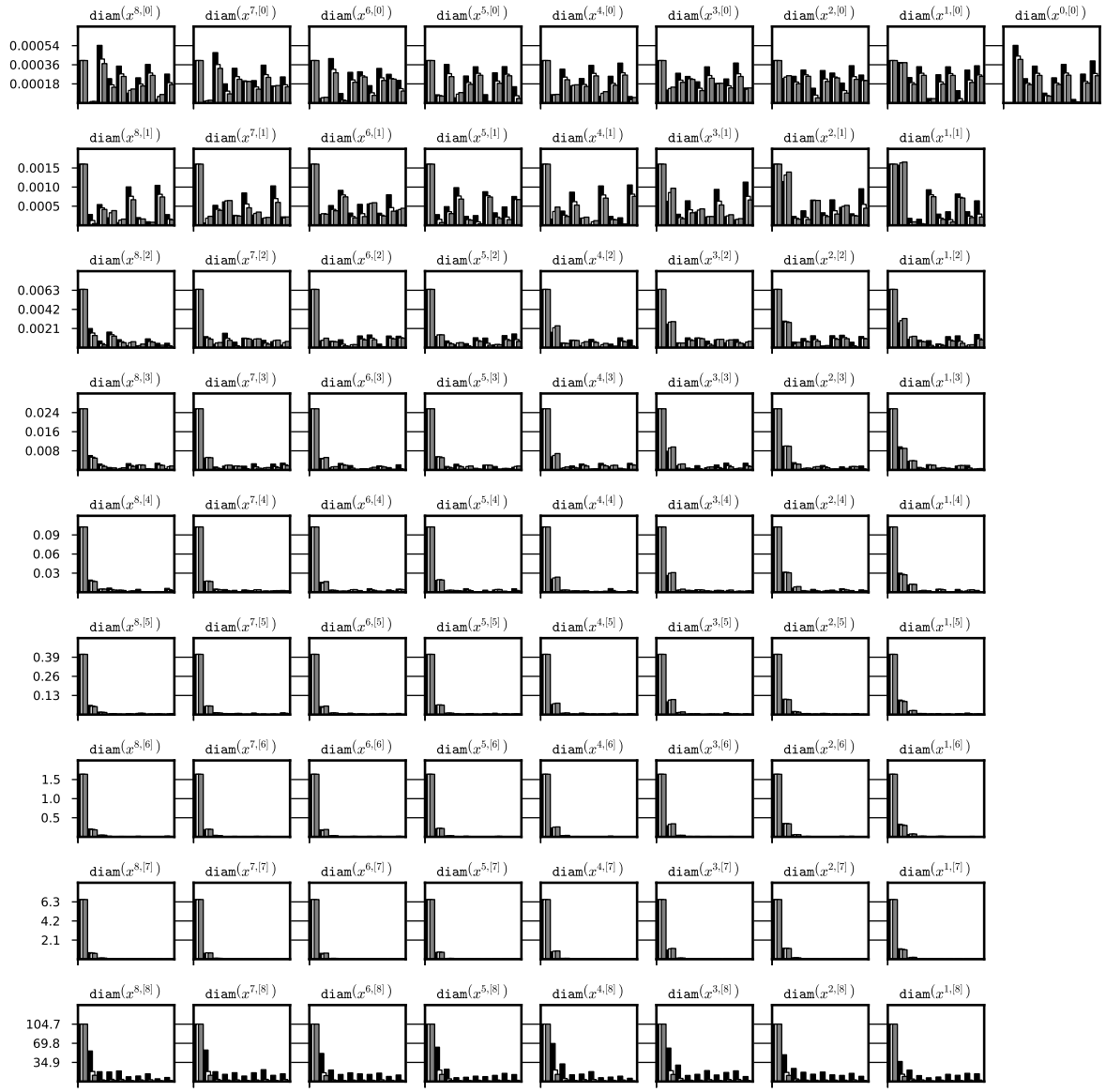

Figure 4.14: Dependence of the diameter of the representation on the grid size p. A history of one integration of three representations of a stable periodic orbit for system (2.15) for parameter $p \in \{8, 16, 32\}$ was recorded every p steps (black, white, gray respectively). The diameters of corresponding representation coefficients (i.e. that represents appropriate derivative at the same time t) are drawn for comparision, i.e. $x^{i,[p]}$ for $p = 8$, $x^{2 \cdot i,[p]}$ for $p = 16$ and $x^{4 \cdot i,[p]}$ for $p = 32$. Each bar is a diameter of the representation coefficient after p steps of iteration. The tests used are: 1c, 4, 5 (black, white, gray respectively). For all integrations the system (2.15) and doubleton Lohner set representation were used. The data is stored the files `periodic_08_07_out_3/rect_di_p.txt`, `periodic_16_07_out/rect_di_p.txt` and `periodic_32_07_out/rect_di_p.txt` respectively.

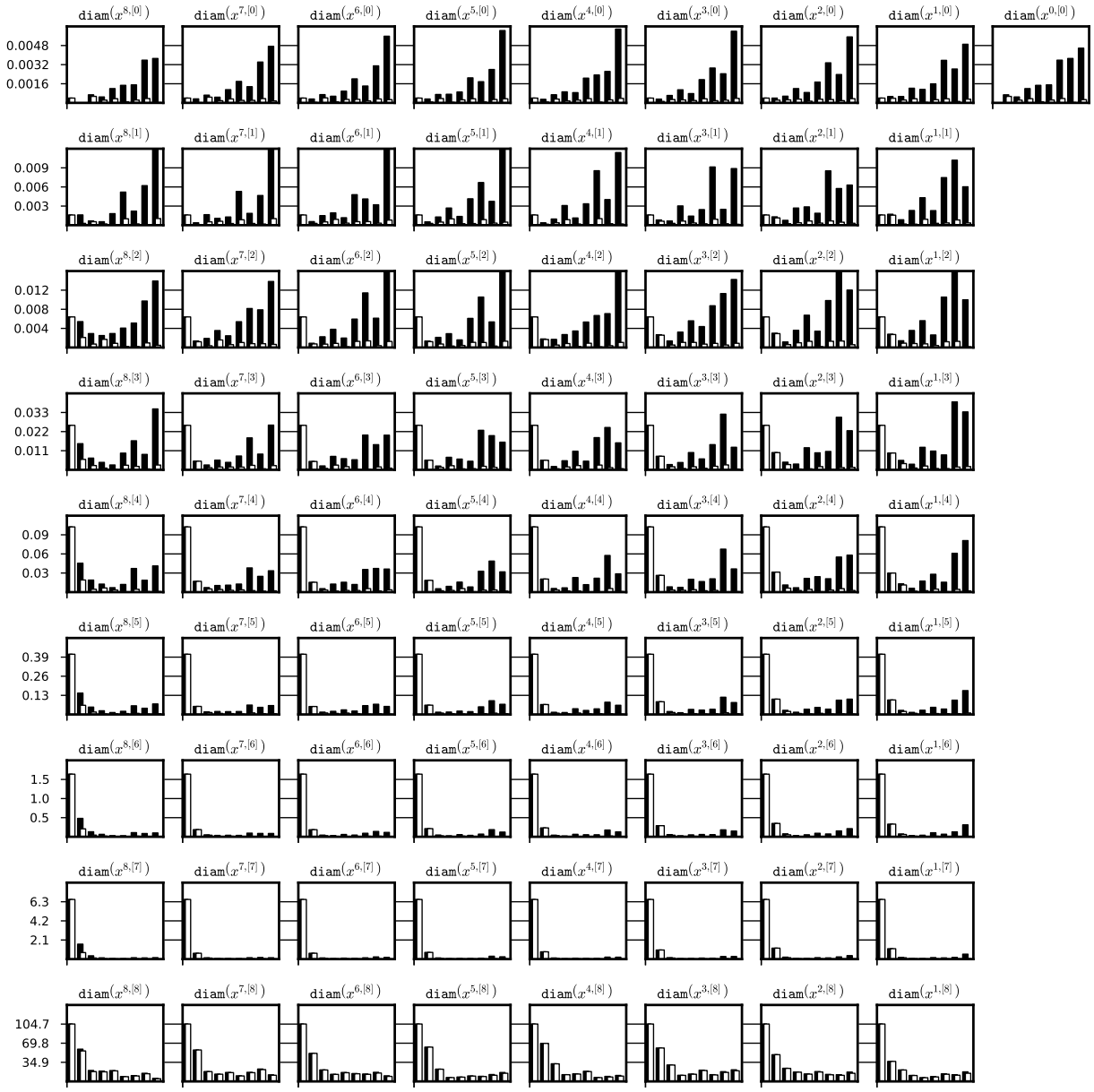

Figure 4.15: Comparision between basic interval numeric method (black) and a Lohner set representation by the doubleton $x_0 + C \cdot r_0 + B \cdot r$ (white). In both cases we have integrated the same initial representation of a stable periodic solution to system (2.15) and we have used interval set representation and (8,7)-representation. On the chart we present the diameter of the interval hull of each representation coefficient every 8 steps of the integration. The data is stored in the file `periodic_08_07_out_3/int_di.txt`.

Chapter 5

Poincaré maps for DDEs

In this chapter we consider the following question: how we can rigorously compute the Poincaré map $P_{S,\varphi}$ associated with the system (3.1)? Namely, we consider the following problem: given the representation $\bar{x}_0$ such that $t_p(x_0)$ exists for any $g \in Supp(\bar{x})$ find a representation $\bar{x}_{t_p(x_0)}$ such that $P_{S,\varphi}(x_0) \in Supp(\bar{x}_{t_p(x_0)})$.

For this purpose we will use the rigorous integrator discussed in Chapter 3. We must however notice, that there is a problem with the integrator: a fixed step size. The fixed step size disallow to move a representation as close to section as possible and, in turn, it may prevent to produce small image of the initial set on the section, unless we choose a very small h at the beginning of the iteration process. But a very small step increases size of the representation and the number of iteration steps needed to reach the section, thus making the integration very slow. An alternative would be to construct a procedure that can create representation of a solution after arbitrary step size $0 < \varepsilon < \frac{1}{p}$. Such a procedure would be used to do the last step of the computation of map P, after moving close enough to the section using basic integrator from Section 3.2.

In this chapter we propose two procedures (called for convenience the ε-methods) that given a representation of x compute a representation of $\varphi(x, \varepsilon)$ for step size $0 < \varepsilon < \frac{1}{p}$. We compare their effectiveness in integration of the neighbourhood of exemplary periodic solutions both stable and hyperbolic with one or two unstable directions. We also discuss application of those methods to compute Poincaré maps, their limitations and the problem with discontinuities imminent in the DDEs (see Remark 19).

Remark 19 *Here we want to stress that, in general, we will not be able to guarantee that $\varphi(x_0, t_p(x_0)) \in \bar{x}_{t_p(x_0)}$ for each $x_0 \in Supp(\bar{x}_0)$! Below we present two examples where (p,n)-representable functions cannot be represented with the same (p,n)-representation after arbitrary time step.*

- ***Example 1:*** *Assume that r.h.s. f of (3.1) is*

$$f(x, y) = 0, \quad \forall x, y \in \mathbb{R} \tag{5.1}$$

and let $x_0(t)$ be an initial function such that:

$$x_0(t) = \begin{cases} \frac{1}{2} + t & -1 \leq t \leq -\frac{1}{2} \\ 0 & -\frac{1}{2} \leq t \leq 0 \end{cases} \tag{5.2}$$

We see that $x(t)$ is continuous on $[-1, 0]$ and C^∞ on each interval $\left[-i \cdot \frac{1}{2}, -i \cdot \frac{1}{2} + \frac{1}{2}\right]$ thus we can construct a (2,n)-representation for any $n \in \mathbb{N}$. We cannot however construct such representations for $x_\varepsilon = \varphi(x, \varepsilon)$ for $0 < \varepsilon < h$, due to the existing discontinuity at $t = -\frac{1}{2}$ in $x(t)$ which propagates to a discontinuity at $t = -\frac{1}{2} - \varepsilon$ in $x_\varepsilon(t)$.

- ***Example 2:*** *Assume r.h.s. f of (3.1) as in Example 1 and assume two initial functions:*

$$x_0(t) = 1 \tag{5.3}$$

and

$$y_0(t) = t \tag{5.4}$$

which give rise to solutions $x(t)$ and $y(t)$ respectively. Let $x_\varepsilon = \varphi(x_0, \varepsilon)$ and $y_\varepsilon = \varphi(y_0, \varepsilon)$ for $0 < \varepsilon < h$, namely:

$$x_\varepsilon(t) = 1, \quad t \in [-1, 0] \tag{5.5}$$

$$y_\varepsilon(t) = \begin{cases} t & t \in [-1, -\varepsilon] \\ 0 & t \in [-\varepsilon, 0] \end{cases}. \tag{5.6}$$

We see that x_ε can be represented with a (p,n)-representation for any p and n while y_ε cannot as its derivative has a discontinuity at $t = -\varepsilon$. This phenomenon is possible as $x_0(t)$ is aligned to φ in the sense that $x'(0^-) = f\left(x(-1), x(0)\right) = x'(0^+)$. In the case of $y(t)$ we have $y'(0^-) = 1 \neq 0 = f\left(y(-1), y(0)\right) = y'(0^+)$, thus the solution y_ε is only continuous.

Having those examples in mind, we see that we will only be able to investigate a restriction of P to some subset of phase-space of regularity high enough for our methods to work (such as $x(t)$ in Example 2). This however will be no issue in computer assisted proofs, as we assume high regularity of r.h.s. of equation (3.1) and we will be applying our methods to investigate solutions for which we will be able to assure their regularity a priori *using smoothing property of the DDE. We discuss this issue further in this chapter in the Section 5.4.*

Remark 20 *To compute Poincaré maps we also need to estimate the final time step $\varepsilon = [\varepsilon_1, \varepsilon_2]$ such that $t_p(x) \in h \cdot N + \varepsilon$ for some $N \in \mathbb{N}$ and all $x \in Supp(\bar{x}_0)$. We do not discuss this issue here, as it is rather simple matter and may be resolved using various approaches like Newton method, binsearch algorithm or other heuristic algorithms.*

5.1 First ε-method

Formula for $c_t^{i,[k]}(\varepsilon)$ in equation (3.11) from Lemma 9 together with a simple estimation of the remainder part of the representation immediately brings the first ε-method H_1 presented in the algorithm 10.

Algorithm 10 H_1

Input: ε, $\bar{x}_0, \bar{x}_h$

Output: $\bar{x}_\varepsilon$

Require: $0 < \varepsilon < \frac{1}{p}$ $\bar{x}_h = \Phi(x_0)$,

1: **for** $k : 0 \leq k \leq n, i : 1 \leq i \leq p$ **do**
2: $\quad \bar{x}_\varepsilon^{i,[k]} \leftarrow$ COMPUTE-C-K($\bar{x}_0, i, k, \varepsilon$)
3: **end for**
4: **for** $i : 1 \leq i \leq p$ **do**
5: $\quad \bar{x}_\varepsilon^{i,[n+1]} \leftarrow$ INTERVALHULL($\bar{x}_0^{i,[n+1]}, \bar{x}_h^{i,[n+1]}$)
6: **end for**
7: $\bar{x}_\varepsilon^{0,[0]} \leftarrow \sum_{k=0}^{n+1} \bar{x}_h^{1,[k]} \cdot \varepsilon^k$

The idea behind Algorithm 10 is very simple. Let $x_0 \in \bar{x}_0$ and let $x(t)$ be a solution to (3.1) with the initial function x_0. Then from Lemma 9 we get that:

$$x_\varepsilon^{[k]}\left(-\frac{i}{p}\right) = x^{[k]}\left(-\frac{i}{p} + \varepsilon\right) \in c_0^{i,[k]}(\varepsilon) \tag{5.7}$$

for any $1 \leq i \leq p$ and $0 \leq k \leq n$. For the $\bar{x}_\varepsilon^{0,[0]}$ we use simply equation (3.6) on the representation $\bar{x}_h = \Phi(\bar{x}_0)$. What is left to compute are the remainders, but for them, as we said in the introduction to this chapter, we need to restrict ourselves to a subset of functions in $Supp(\bar{x}_0)$ of regularity high enough.

Lemma 21 *Let $\bar{x}_0$ be a representation and let $\bar{x}_\varepsilon$ be an output from Algorithm 10 executed for arguments ε, x_0, $\Phi(x_0)$. Assume that function $x(t)$ of class C^{n+1} on the interval $[-1, h]$ is a solution to* (3.1) *with smooth r.h.s. such that $x_0 \in \bar{x}_0$. Then $\varphi(x_0, \varepsilon) \in Supp(\bar{x}_\varepsilon)$.*

Proof: the proof is obvious. All the coefficients of the representation $\bar{x}_\varepsilon$ were computed with formulas valid for any function $x_0 \in \bar{x}_0$, except for the remainders. Now, if $x(t)$ is C^{n+1} on $[-1, h]$ then the $n+1$-st derivative of x exists on each interval $[-i \cdot h + \varepsilon, -i \cdot h + h + \varepsilon]$,

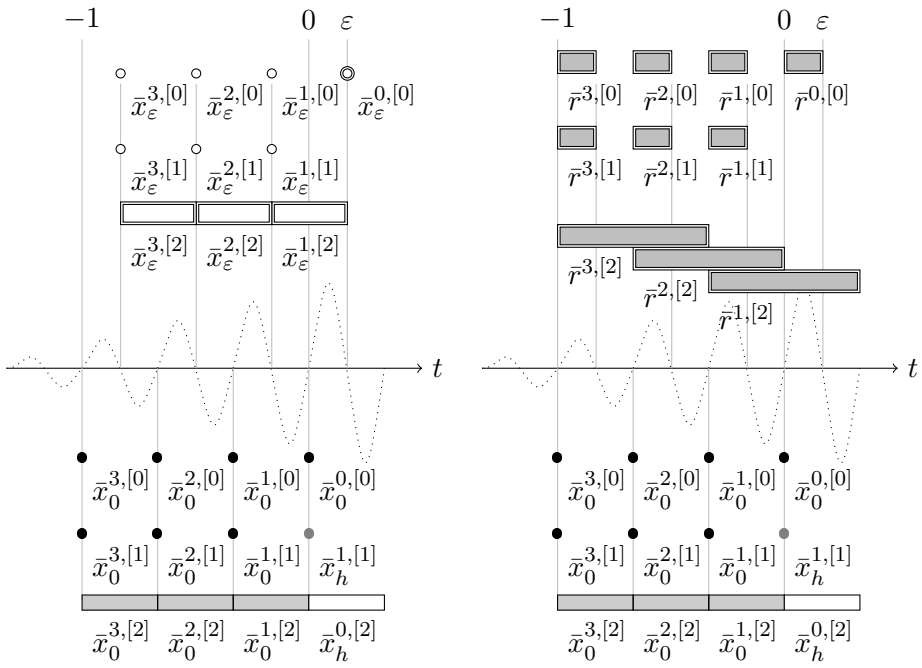

Figure 5.1: The idea of the first ε-method H_1. On the left: what we want to compute (above) and what we have on the input (below). On the right: what is really computed by the procedure H_1.

$1 \leq i \leq p$. From the fact that $0 \leq \varepsilon \leq h$ we have:

$$\begin{aligned} x_\varepsilon^{[n+1]}(-i \cdot h + [0,h]) &= x^{[n+1]}(-i \cdot h + \varepsilon + [0,h]) \subset & (5.8)\\ &\subset x^{[n+1]}(-i \cdot h + [0,h]) \cup x^{[n+1]}(-i \cdot h + h + [0,h]) \subset & (5.9)\\ &\subset \left[\bar{x}_0^{i,[n+1]}, \bar{x}_h^{i,[n+1]}\right] & (5.10) \end{aligned}$$

■

The graphical presentation of the idea of the ε-method H_1 is shown in Figure 5.1.

Remark 22 *In fact, procedure H_1 uses r.h.s. of equation* (3.1) *only to compute coefficient $\bar{x}_\varepsilon^{0,[0]}$ (as it sums coefficients from the representation $x_h = \Phi(x_0)$). Apparently, when we analyze the formula for $c_t^{i,[k]}(\varepsilon)$ from Lemma 9 (which is used in procedure H_1 to obtain rigorous bounds on almost all coefficients) we can draw a conclusion that ε-method H_1 may produce poor estimates for the solution because of the 'bottom-up' computation approach which builds all $c_t^{i,[k]}(\varepsilon)$ based on the remainder estimation $\bar{x}^{i,[n+1]}$. The remainder is often big, because it holds bounds on the $n+1$-st derivative on the whole interval $[-i \cdot h, -i \cdot h + h]$.*

The effect of this overestimation is not so important in the standard integration procedure Φ, as it is only used to estimate a remainder in Taylor series of usually high order $n+1$. But it may have great impact on coefficients $c_t^{i,[k]}(\varepsilon)$, especially for $k \approx n$ or $k \approx \frac{n}{2}$, as for those k's the approximation given by $c_t^{i,[k]}(\varepsilon)$ is either of a low order in the variable ε (k close to n) or it has quite big coefficients $\binom{l}{k}$ in (3.11) (k close to $\frac{n}{2}$). Thus it would be advisable to find a better procedure for computing ε-method - a method that will use r.h.s. of equation (3.1) to compute more coefficients, and thus it would be able to „exploit" inner dynamics of the system.

5.2 Second ε-method

Here, we will use the fact that remembering the history on an interval $[-2, 0]$ instead only on $[-1, 0]$ gives us chance to compute all coefficients of $\bar{x}_\varepsilon$ using an analogue of the forward-part algorithm from Section 3.2. This will allow to exploit the inner dynamics of the equation (3.1) to (probably) get better numerical results on all coefficients than it can be done in the case of procedure H_1. The idea is simple: we would be able to compute coefficients $\bar{x}_\varepsilon^{i,[k]}$ using equation (3.27) if we have two ingredients:

- value for $\bar{x}_\varepsilon^{i,[0]}$, which can be simply obtained using equation (3.6) on coefficients $\bar{x}_h^{1,[k]}$,
- the derivatives in the past, namely estimates for $x^{[k]}(\varepsilon - 1)$, $0 \leq k \leq n$. For this, we will simply use already presented procedure H_1.

We hope, that this procedure will be better than H_1 because equation (3.27) uses r.h.s. f of the equation (3.1), thus it can take advantage of the contracting properties of the system to compensate for overestimates given by H_1. The idea behind the second ε-method H_2 is presented in the series of figures 5.2-5.5 and the numerical procedure itself is presented in Algorithm 12.

In the procedure H_2 we will use an auxiliary method presented in Algorithm 11 for computing remainder on the interval of length ε instead of length h as it is in case of Algorithm 5.

The following lemma is true:

Lemma 23 *Let x be a solution to (3.1) with some initial function x_0.*

Let r^* be an output from Algorithm 11 executed for ε, i, $\left\{\bar{x}_{t-1}^{i,[k]}\right\}_{0\leq k\leq n+1}$, $\left\{\bar{x}_t^{i,[k]}\right\}_{0\leq k\leq n}$ such that:

- $\varepsilon \in [0, h]$,
- $1 \leq i \leq p$,
- $x_{t-1}^{[k]}(-i \cdot h) \in \bar{x}_{t-1}^{i,[k]}$, *for* $0 \leq k \leq n$

Algorithm 11 compute-remainder*

Input: ε, i, $\left\{\bar{x}_{t-1}^{i,[k]}\right\}_{0\le k\le n+1}$, $\left\{\bar{x}_{t}^{i,[k]}\right\}_{0\le k\le n}$

Output: r^*, a^*, b^*

1: $\left\{c^{[k]}\right\} \leftarrow$ COMPUTE-C$(x_t^{[0]}(-1), ..., x_t^{[n+1]}(-1), [0,\varepsilon])$

2: $d^{[0]} \leftarrow$ ROUGHENCLOSURE$(\ f(c^{[0]},\cdot), \bar{x}_0^{0,[0]}, \varepsilon\ )$

3: **for** $k : 1 \le k \le n+1$ **do**

4: $\quad d^{[k]} \leftarrow \frac{1}{k} \cdot F^{[k-1]}\left(c^{[0]}, .., c^{[k-1]}, d^{[0]}, .., d^{[k-1]}\right)$

5: **end for**

6: $a^* \leftarrow \frac{1}{(n+1)} \cdot F^{[n]}\left(x^{[0]}(-1), .., x^{[n]}(-1), x^{[0]}(0), .., x^{[n]}(0)\right)$

7: $b^* \leftarrow F^{[n+1]}(c^{[0]}, .., c^{[n]}, c^{[n+1]}, d^{[0]}, .., d^{[n]}, d^{[n+1]})$

8: $r^* \leftarrow a^* + [0,\varepsilon] \cdot b^*$

- $x_{t-1}^{[n+1]}(-i\cdot h + \xi) \in \bar{x}_{t-1}^{i,[n+1]}$, *for any* $\xi \in [0,\varepsilon]$,
- $x_t^{[k]}(-i\cdot h) \in \bar{x}_t^{i,[k]}$, *for* $0 \le k \le n$

Then $x^{[n+1]}([-i\cdot h, -i\cdot h + \varepsilon]) \subset r^*$.

Proof: The proof is analogous to the proof of Lemma 13. ∎

Now we are able to present the method H_2 in Algorithm 12.

Lemma 24 *Let $\bar{x}_{-1}$ be a representation and let $\bar{x}_\varepsilon$ be an output from Algorithm 12 executed for arguments ε, x_{-1}, $\Phi(x_{-1})$, $\Phi^p(x_{-1})$, $\Phi^{p+1}(x_{-1})$. Assume that function $x(t)$ of class C^{n+1} on the interval $[-2,h]$ is a solution to (3.1) with initial function $x_{-1} \in \bar{x}_{-1}$. Then $\varphi(x_{-1}, 1+\varepsilon) \in Supp(\bar{x}_\varepsilon)$.*

Below we present only sketch of a proof, as it is rather technical application of Definition 10, equation (3.6), equation (3.1), Lemma 21 and Lemma 11.

Proof: First of all, by Lemma 21, we have $x_{-1+\varepsilon} = \varphi(x_{-1},\varepsilon) \in Supp(\bar{r})$. Let $x_\varepsilon = \varphi(x_{-1}, 1+\varepsilon)$, then:

- we use equation (3.6) and Definition 10 to obtain that $x_\varepsilon(-i\cdot h) \in \bar{x}_\varepsilon^{i,[0]}$ for all $0 \le i \le p$
- we use equation (3.1) to obtain $x_\varepsilon^{[k]}(-i\cdot h) \in \bar{x}_\varepsilon^{i,[k]}$ for each $1 \le i \le p$ and $0 \le k \le n$,

Algorithm 12 H_2

Input: ε_1, ε_2, $\bar{x}_{past}$, $\bar{x}_{past'}$, $\bar{x}_{before}$, $\bar{x}_{after}$

Output: $\bar{x}_\varepsilon$

Require: $\varepsilon \in (0,h)$, $\bar{x}_{past'} = \Phi(\bar{x}_{past})$, $\bar{x}_{before} = \Phi^p(x_{past})$, $\bar{x}_{after} = \Phi^{p+1}(x_{past})$,

1: $\varepsilon \leftarrow [\varepsilon_1, \varepsilon_2]$

2: $\bar{r} \leftarrow H_1(\varepsilon_1, \varepsilon_2, \bar{x}_{past}, \bar{x}_{past'})$

3: **for** $i : 1 \leq i \leq p$ **do**

4: $r^{i,*} \leftarrow$ COMPUTE-REMAINDER*($[0,\varepsilon_2], i, \bar{x}^{i,[0]}_{past}, .., \bar{x}^{i,[n+1]}_{past}, \bar{x}^{i,[0]}_{before}, .., \bar{x}^{i,[n]}_{before}$)

5: $\bar{x}^{i,[0]}_\varepsilon \leftarrow \sum_{k=0}^{n} \bar{x}^{i,[k]}_{before} \cdot \varepsilon^k + r^{i,*} \cdot \varepsilon^{n+1}$

6: $\left\{\bar{x}^{i,[k]}_\varepsilon\right\}_{1\leq k\leq n} \leftarrow$ COMPUTE-REP-K($n-1, \bar{r}^{i,[0]}, .., \bar{r}^{i,[k-1]}, \bar{x}^{i,[0]}_\varepsilon$)

7: **end for**

8: **for** $k : 0 \leq k \leq n+1, i : 1 \leq i \leq p$ **do**

9: $c^{i,[k]}_{past} \leftarrow$ COMPUTE-C-K($\bar{x}_{past}, i, k, [0,h]$)

10: $c^{i,[k]}_{past'} \leftarrow$ COMPUTE-C-K($\bar{x}_{past'}, i, k, [0,\varepsilon_2]$)

11: $\tilde{c}^{i,[k]} \leftarrow$ INTERVALHULL($c^{i,[k]}_{past}, c^{i,[k]}_{past'}$)

12: **end for**

13: **for** $i : 1 \leq i \leq p$ **do**

14: $\tilde{d}^{i,[0]} \leftarrow$ ROUGHENCLOSURE($f, \tilde{c}^{i,[0]}, \bar{x}^{i,[0]}_\varepsilon, h$)

15: $\left\{\tilde{d}^{i,[k]}\right\}_{1\leq k\leq n+1} \leftarrow$ COMPUTE-REP-K($n, \tilde{c}^{i,[0]}, .., \tilde{c}^{i,[k-1]}, \tilde{d}^{i,[0]}$)

16: **end for**

17: $\bar{x}^{i,[n+1]}_\varepsilon \leftarrow \frac{1}{n+1} \cdot F^{[n]}\left(\bar{r}^{i,[0]}, .., \bar{r}^{i,[n]}, \bar{x}^{i,[0]}_\varepsilon, .., \bar{x}^{i,[n]}_\varepsilon\right) + [0,h] \cdot F^{[n+1]}\left(\tilde{c}^{i,[0]}, .., \tilde{c}^{i,[n+1]}, \tilde{d}^{i,[0]}, \tilde{d}^{i,[n+1]}\right)$

18: **for** $k : 0 \leq i \leq n+1$ **do**

19: $c^{p,[k]} \leftarrow$ COMPUTE-C-K($\bar{x}_{before}, i, k, [0,h]$)

20: **end for**

21: $\tilde{d}^{*,[0]} \leftarrow$ ROUGHENCLOSURE($f, c^{p,[0]}, \bar{x}^{0,[0]}_{before}, \varepsilon_2$)

22: $\left\{\tilde{d}^{*,[k]}\right\} \leftarrow$ COMPUTE-REP-K($n, c^{p,[0]}, .., c^{p,[k-1]}, \tilde{d}^{*,[0]}$)

23: $a^{**} \leftarrow \frac{1}{n+1} \cdot F^{[n]}\left(\bar{x}^{p,[0]}_{before}, .., \bar{x}^{p,[n]}_{before}, \bar{x}^{1,[0]}_{after}, .., \bar{x}^{1,[n]}_{after}\right)$

24: $b^{**} \leftarrow [0,\varepsilon_2] \cdot F^{[n+1]}\left(c^{p,[0]}, .., c^{p,[n+1]}, \tilde{d}^{*,[0]}, \tilde{d}^{*,[n+1]}\right)$

25: $\bar{x}^{0,[0]}_\varepsilon \leftarrow \sum_{k=0}^{n} \bar{x}^{1,[k]}_{after} \cdot \varepsilon^k + a^{**} \cdot \varepsilon^{n+1} + b^{**} \cdot \varepsilon^{n+2}$

- we use Lemma (11) and the fact that $x(t)$ is of class C^{n+1} on $[-2,h]$ to get $x^{[k]}_\varepsilon(-i \cdot h + [0,h]) \subset \bar{x}^{i,[n+1]}_\varepsilon$ for all $1 \leq i \leq p$ ∎

The idea of the method H_2 is presented in a series of figures. In Figure 5.2 we have a graphical representation of the setup to the procedure H_2. Figure 5.3 shows the usage of the method H_1 to compute a representation $\bar{r}$ of $x_{-1+\varepsilon}$. The representation $\bar{r}$ is then used in other procedures to obtain other coefficients using r.h.s. of equation (3.1). Computation of $\bar{x}_\varepsilon^{i,[k]}$ for $0 \leq k \leq n$ is presented in Figure 5.4. The final computation of the value of the function at time $t = \varepsilon$ is shown in Figure 5.5.

Remark 25 *The procedure H_2 in the form presented in Algorithm 12 has the following advantage: in the case of r.h.s. of equation (3.1) independent of $x(t-1)$, i.e. when $f\left(x(t-1), x(t)\right) = f\left(x(t)\right)$ we can show that the method is essentially equivalent to rigorous Taylor method of order n for an ODE of the form $x' = f(x)$ for any step size $\varepsilon \in [0, h]$. We set $f\left(x(t-1), x(t)\right)$ to $f\left(x(t)\right)$ and $F^{[k]}\left(x^{[0]}(t-1), .., x^{[k]}(t-1), x^{[0]}(t), .., x^{[k]}(t)\right)$ to $F^{[k]}\left(x^{[0]}(t), .., x^{[k]}(t)\right)$ to indicate that there is no dependence on the past. We see that all coefficients of the resulting representation $\bar{x}_\varepsilon$ are estimates given by the classical rigorous Taylor method for ODEs. Namely, in the past, we have:*

$$\begin{aligned}
\bar{x}_\varepsilon^{i,[0]} &= \sum_{k=0}^{n} \bar{x}_0^{i,[k]} \cdot \varepsilon^k + b^{i,*} \cdot \varepsilon^{n+1} \\
\bar{x}_\varepsilon^{i,[k]} &= \frac{1}{k} \cdot F^{[k-1]}\left(\bar{x}_\varepsilon^{i,[0]}, .., \bar{x}_\varepsilon^{i,[k-1]}\right) \qquad\qquad 1 \leq k \leq n,
\end{aligned}$$

Where $b^{i,*}$ becomes a Taylor remainder computed using classical rough enclosure for ODE: as we can see in Algorithm 11 f depends only on $x(t)$, which reduces computation of $d^{[0]}$ to computation of the classic rough enclosure for ODE $x' = f(x)$. Then, the coefficients $d^{[k]}$ depends only on $d^{[0]}$:

$$d^{[k]} = \frac{1}{k} \cdot F^{[k-1]}\left(d^{[0]}, .., d^{[k-1]}\right), \quad 1 \leq k \leq n+1 \tag{5.11}$$

exactly as in case of ODE Taylor method of order n.

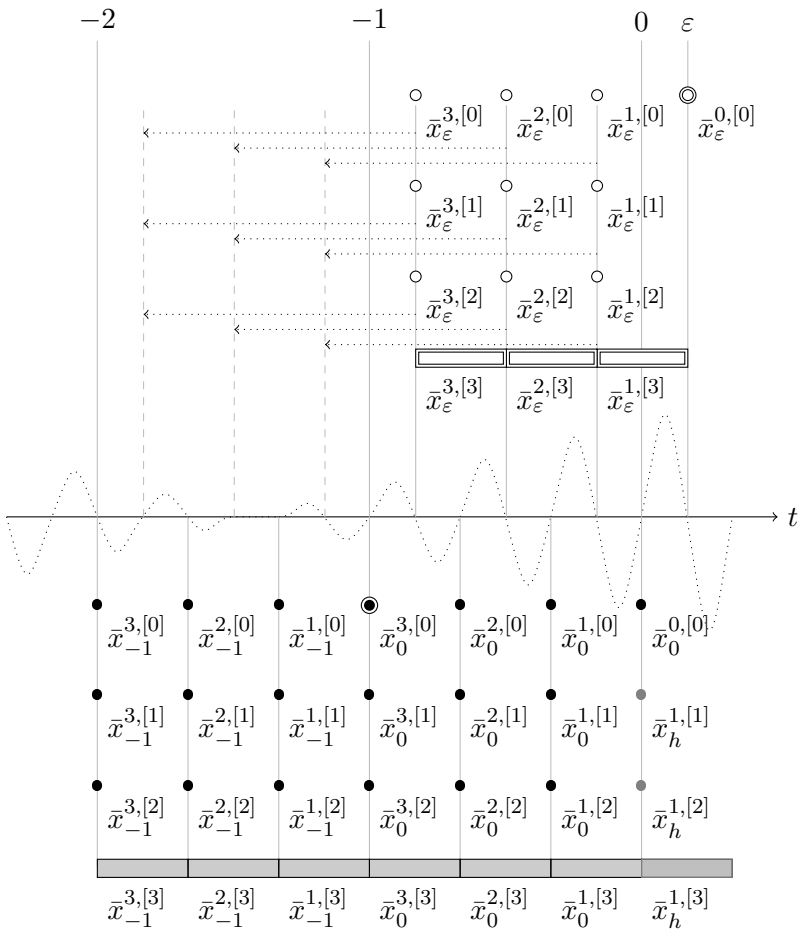

Figure 5.2: Computing a_ε with the second method - what we have (bottom) and what we want to have (up). On the bottom, marked with gray dots and rectangles, there is an additional column of representation a_h for $h = \frac{1}{p}$ which is needed to compute $\bar{x}_\varepsilon^{0,[0]}$ (double-bordered circle above timeline). The element $\bar{x}_0^{3,[0]} = \bar{x}_{-1}^{0,[0]}$ that is present in both representation a_-1 and a_0 is marked with border on the bottom. Dotted arrows shows time $t = \varepsilon - \frac{i}{p}$, where we need to look for past values when computing given coefficient.

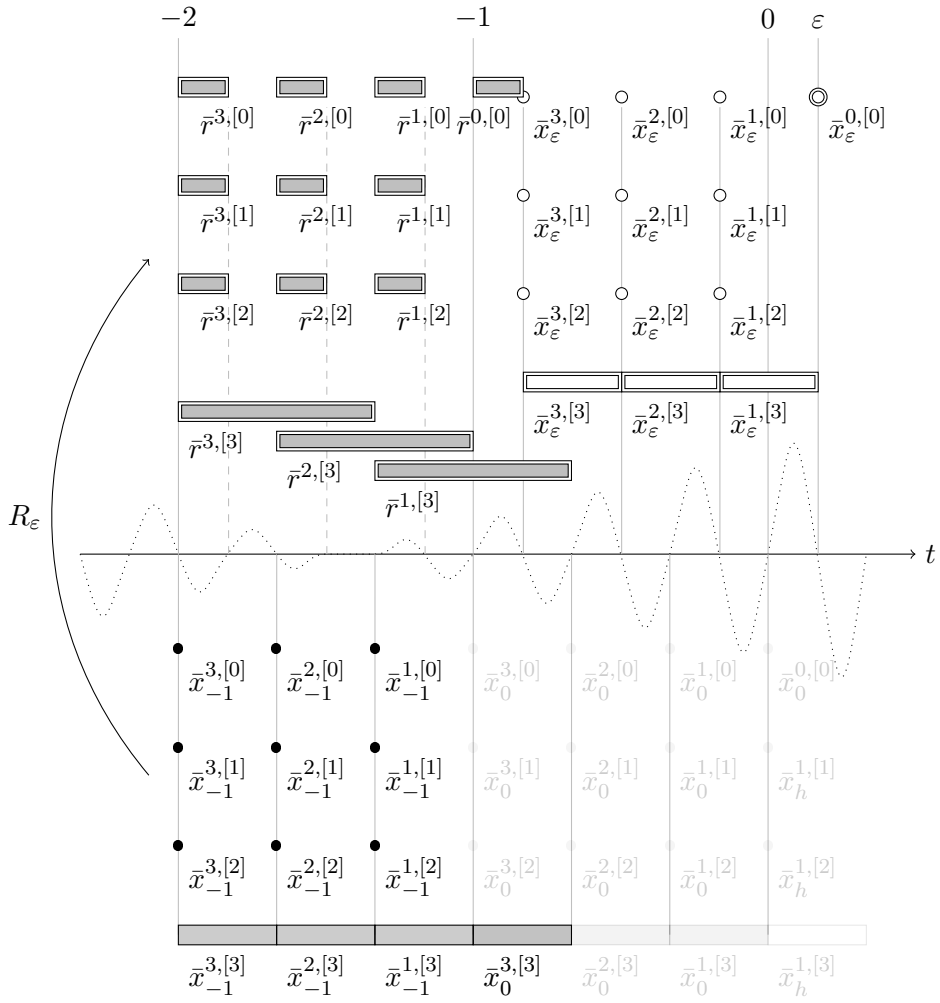

Figure 5.3: Here we have computed $r = R_\varepsilon(a_{-1})$ (double-bordered rectangles with gray interior). We will use it as values at times $\varepsilon - \frac{i}{p}$ (dashed lines). The elements needed to compute r are drawn as solid black dots and dark gray rectangles. Notice that $\bar{r}^{i,[n+1]}$ are much bigger than needed, but we are not able to overcome this (a pointed out in Section 5.1).

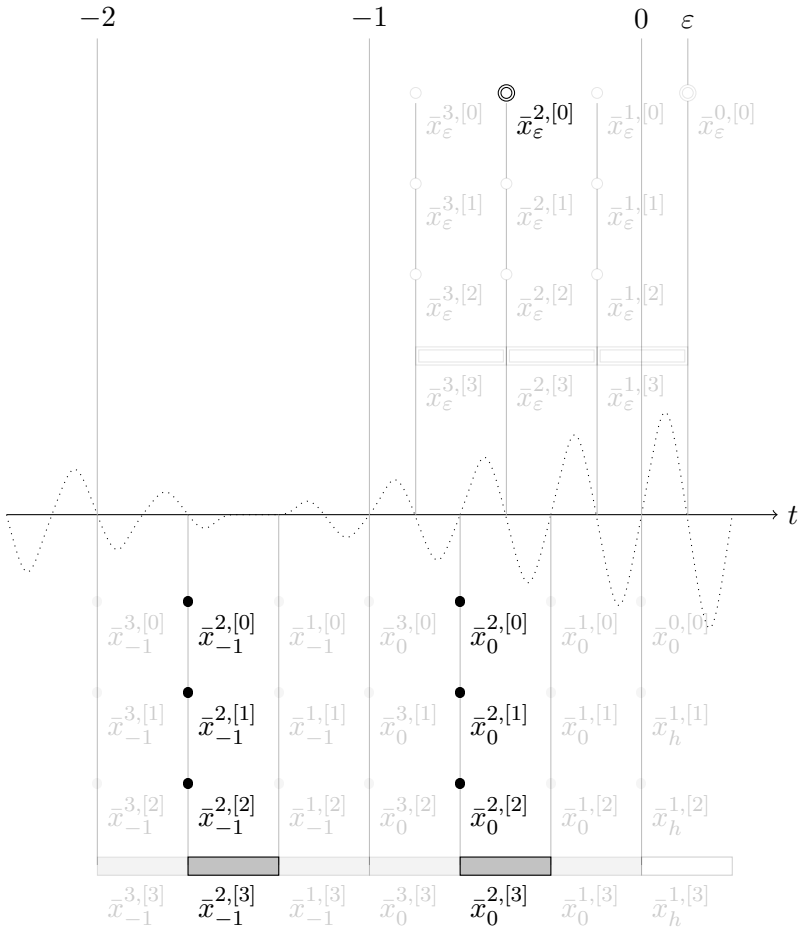

Figure 5.4: Computing $\bar{x}_\varepsilon^{i,[0]}$ in the second method . The elements needed to compute $\bar{x}_\varepsilon^{i,[0]}$ for $i = 2$ (highlighted double bordered element above the axis) are drawn highlighted below the axis.

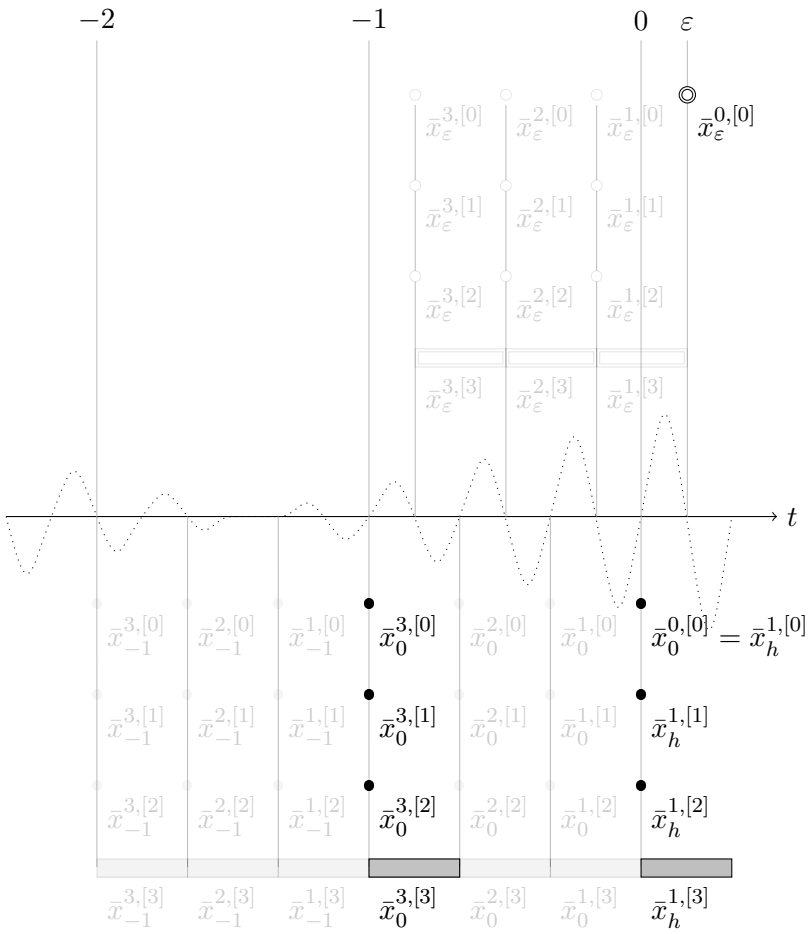

Figure 5.5: Computing $\bar{x}_\varepsilon^{0,[0]}$ in the second method (highlighted double bordered element above the axis). Elements needed to compute $\bar{x}_\varepsilon^{0,[0]}$ are drawn highlighted below the axis.

5.3 Comparison of the ε-step methods

Here we compare the performance of the ε-methods H_1 and H_2. The test are performed as follows: we use as the input to the ε-methods the representations after $N = p \cdot (n+2)$ steps from the integration process history in tests $3c$, both for stationary solution to system (2.14) and periodic solution to system (2.15). Here, the number of steps equal is important as we need another representation after $p \cdot (n+1)$ steps to compute procedure H_2 and we want to assure that this representation contains smooth enough functions. We present the comparison of the diameters of the sets obtained from H_1 and H_2 for three step sizes $\varepsilon \in \left\{\frac{1}{4} \cdot h, \frac{2}{4} \cdot h, \frac{3}{4} \cdot h\right\}$ to be able to asses the influence of the step size on the resulting set.

Figure 5.6 presents comparison of the ε-methods applied to a stationary solution $x \equiv 0$ to system (2.14). In the first column we see the diameter of the (8,7)-representation after $N = p \cdot (n+2)$ steps of integration ($p = 8$, $n = 7$, $h = \frac{1}{8}$). Three other columns present diameters of the resulting sets after applying ε-method for increasing step size $\varepsilon \in \left\{\frac{1}{4} \cdot h, \frac{1}{2} \cdot h, \frac{3}{4} \cdot h\right\}$. We see that the diameters do not differ significantly for coefficients $\bar{x}^{i,[k]}$ for $k < 7$. Closer inspection however reveals that the ε-method H_2 is always better or equal than the procedure H_1. For $k = 7$ method H_2 is evidently better than H_1, especially for longer step sizes. Moreover, it allowed to achieve contraction on the remainder part ($k = 8$), where method H_1 simply preserved the diameter of the set (this is consequence of the way in which we compute remainder in H_1, as a interval hull of two consecutive remainder terms).

Figure 5.7 presents comparison of the ε-methods applied to a stable periodic solution to system (2.15). First column presents the size of the initial function. Te second column presents the diameter of the (8,7)-representation after $N = p \cdot (n+2)$ steps of integration ($p = 8$, $n = 7$, $h = \frac{1}{8}$). Three other columns presents diameters of the resulting set after applying ε-method for increasing step size $\varepsilon \in \left\{\frac{1}{4} \cdot h, \frac{1}{2} \cdot h, \frac{3}{4} \cdot h\right\}$. For coefficients $\bar{x}^{i,[k]}$ for $k = 0$ we see that the results does not differ significantly, but surprisingly we see that the method H_2 is worse than H_1 for coefficients with $k = 1, 2$. From $k = 3$ the method H_2 starts to behave much better than H_1, especially for longer step sizes. Especially, method H_2 guarantee that the remainder part is contracting, where method H_1 fail, producing quite big overestimates.

We cannot now explain the surprising behaviour of the H_2 method for small values of k. It may be that the diameter of the set in the past, $\Phi^{N-p}(\bar{x}_0)$ used in the H_2 method to compute r^*, play a crucial role. But, as we see in Figure 5.8 the sets $\bar{x}_{past} = \Phi^{N-p}(\bar{x}_0)$ and $\bar{x}_{present} = \Phi^N(\bar{x}_0)$ cannot be so easily compared - as some coefficients are larger in past (p steps before the section), but some of them are larger at the present time (just before the section). So it is hard to draw a clear conclusion. This experiment suggests that the preferred way of computing the representation after ε step size is to use *both* methods and return intersection of the resulting representation.

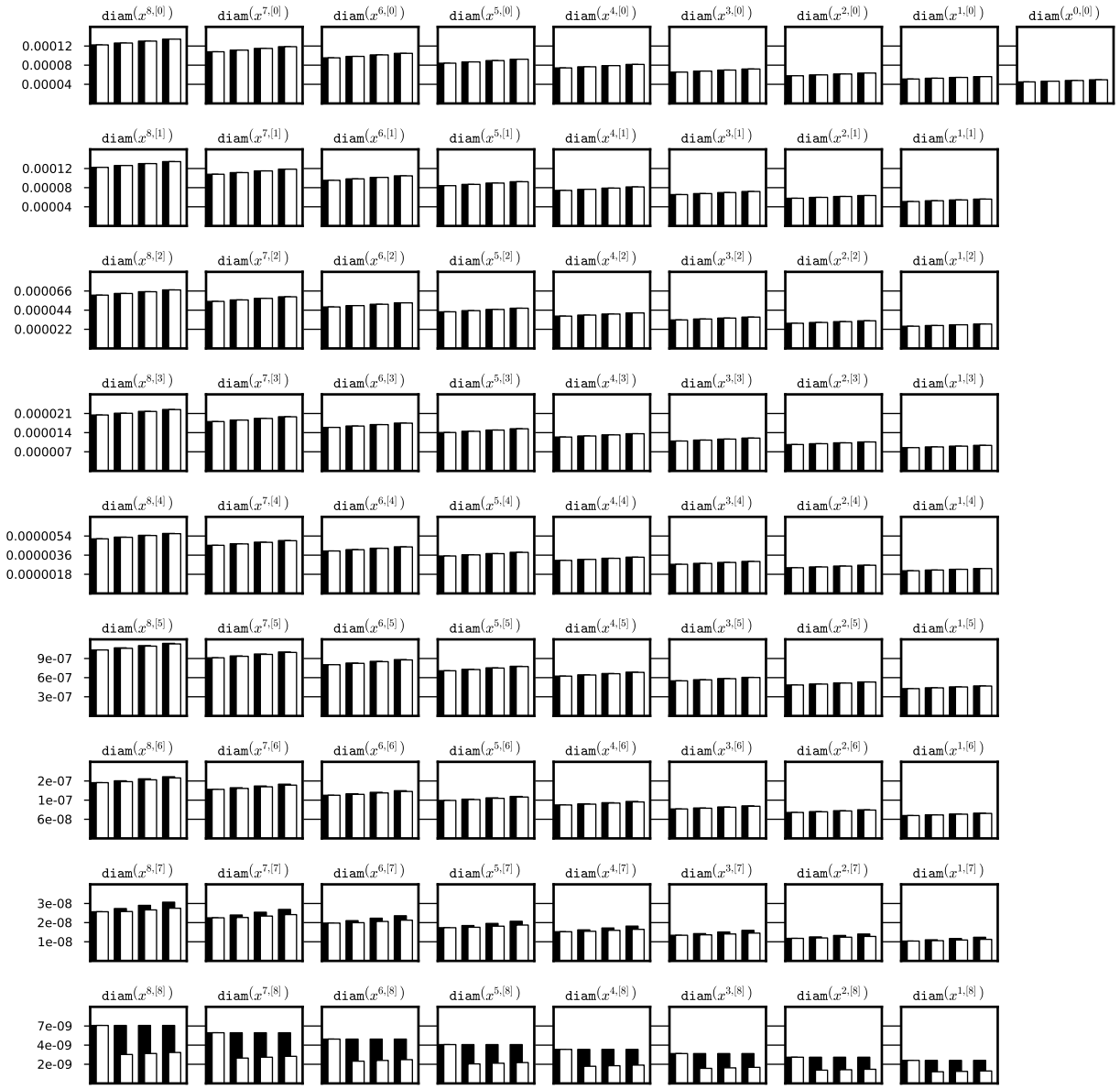

Figure 5.6: Performance of the ε-methods H_1 and H_2 (black and white respectively) computed for three values of the step size $\varepsilon \in \left\{\frac{1}{4}\cdot h, \frac{1}{2}\cdot h, \frac{3}{4}\cdot h\right\}$ (second, third and fourth column respectively). In the first column we have diameter of a set $\bar{x}_{before}$ used to compute ε-method, $\bar{x}_{before} = \Phi^{p\cdot(n+2)}(\bar{x}_0)$ for initial representation $\bar{x}_0$ of a stable stationary solution for system (2.14). System (2.14), doubleton Lohner set representation and (8,7)-representation were used for integration. The data from test 1c was used. Presented data for ε-methods H_1 and H_2 is stored in files `steady_08_07_out_3/di_epsi_1_rect.txt` and `steady_08_07_out_3/di_epsi_2_rect.txt` respectively.

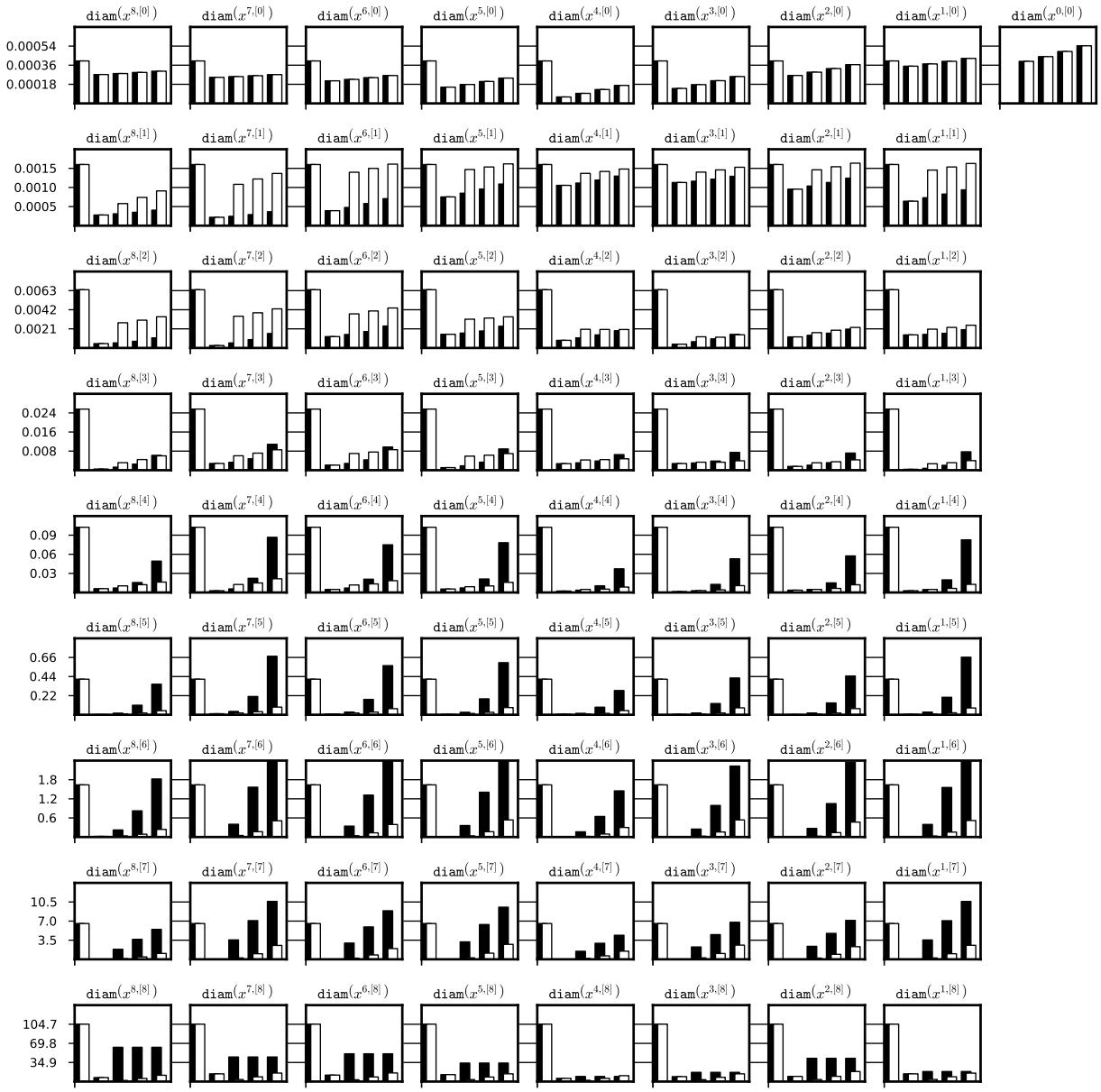

Figure 5.7: Performance of the ε-methods H_1 and H_2 (black and white respectively) computed for three values of the step size $\varepsilon \in \left\{\frac{1}{4}\cdot h, \frac{1}{2}\cdot h, \frac{3}{4}\cdot h\right\}$ (second, third and fourth column respectively). In the first column we have diameter of a set $\bar{x}_{before}$ used to compute ε-method, $\bar{x}_{before} = \Phi^{p\cdot(n+2)}(\bar{x}_0)$ for initial representation $\bar{x}_0$ of a stable periodic solution for system (2.15). System (2.15), doubleton Lohner set representation and (8,7)-representation were used for integration. The data from test 1c was used. Presented data for ε-methods H_1 and H_2 is stored in files `periodic_08_07_out_3/di_epsi_1_rect.txt` and `periodic_08_07_out_3/di_epsi_2_rect.txt` respectively.

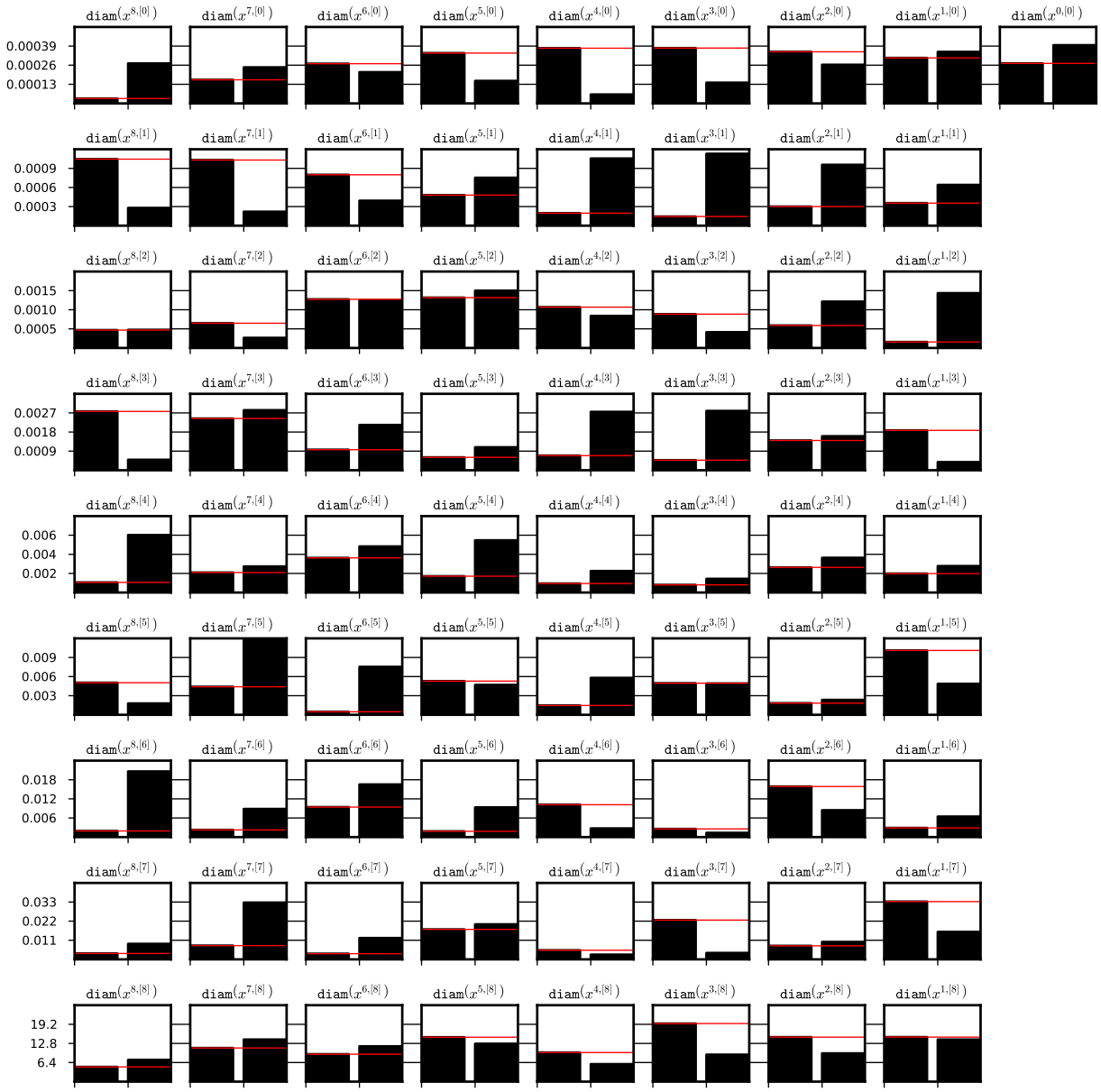

Figure 5.8: Comparison of diameters of the sets $\bar{x}_{past} = \Phi^{N-p}(\bar{x}_0)$ and $\bar{x}_{present} = \Phi^N(\bar{x}_0)$ (left and right bar respectively) for the periodic orbit in system (2.15). The sets $\bar{x}_{past}$ and $\bar{x}_{present}$ are used as an input to the H_2 method and $N = p \cdot (n+2)$.

5.4 Discontinuity issues of ε-methods

In Remark 19, we have presented a problem that arises when we want to produce a (p, n)-representation of a set $\varphi(Supp(\bar{x}_0), \varepsilon)$ for $0 < \varepsilon < h$. Namely, *we are not able to represent* $\varphi(x, \varepsilon)$ *for all functions* $x \in Supp(\bar{x}_0)$ because many of them, in general, have discontinuities in derivatives of various order at grid points $t = -i \cdot h$ (see examples in Remark 19). In Lemmas 21 and 24 we have established a classes of functions for which procedures H_1 and H_2 produces valid (p, n)-representations. It is advisable then to be able to assure or check *a priori* if those classes have nonempty intersection with the $Supp(\bar{x}_0)$ of the initial representation $\bar{x}_0$ and this section is devoted to this problem.

This problem is important in the case of rigorously proving the existence of periodic orbits in system (3.1) with r.h.s. f smooth enough. Let assume that f is C^∞, then we can show that any periodic solution $x(t)$ to (3.1) must be of class C^∞ (for details see [8]). Thus we can do rigorous computation using the integrator and the ε-methods to obtain estimates of $x(t)$ and to construct Poincaré map P in the vicinity of the initial function x_0.

In the examples in Remark 19 we have shown that even C^∞ initial function x_0 may give rise to a solution $x(t)$ that is only C^0. Only the solutions that are in some sense *aligned* to the flow φ will give rise to a smooth solutions. Definition 18 gives a rigorous definition of an *alignment* of an initial function and Definition 19 defines a subset of $Supp(\bar{f})$ of *aligned* functions.

Definition 18 *Let $x_0 \in C^k$. We say that x_0 is* C^k-aligned *to system* (3.1) *iff $x_0^{(j)}(0^-) = \frac{d^j}{dt^j} f\left(x_0(t-1), x_0(t)\right)|_{t=0}$ for all $0 \le j \le k$.*

In other words, x_0 is C^k-aligned if left and right derivatives of a solution $x(t)$ of system (3.1) with initial function x_0 are equal up to order k at $t = 0$.

Definition 19 *We say that a (p,n)-representation $\bar{x}_0$ is* C^k-consistent *iff there exist $x_0 \in Supp(\bar{x}_0)$ such that x_0 is C^k-aligned. By $Cons^{(k)}(\bar{x}_0)$ we denote the set of all C^k-aligned functions in $Supp^{(k)}(\bar{x}_0)$.*

Now, if the initial representation $\bar{x}_0$ is C^{n+1}-consistent then we can use ε-methods to rigorously estimate restriction of the Poincaré map $P_{\varphi,S}$ to a set $Cons^{(n+1)}(\bar{x}_0)$ - thus we will be able to use those estimates in computer assisted proofs. By the smoothing property discussed shortly in Section 2.2 we know that the simplest method for obtaining C^k-consistent representation is iteration of time shift by 1 map $P(x_0) = x_1$. If f in (3.1) is of class at least C^{n+1} and if $\bar{z}$ is any interval (p, n)-representation then we have that $\Phi^{p \cdot k}$ must be C^k-consistent. This approach may be applied in the case of computer assisted proofs of contracting periodic orbits but may fail when the orbit is repelling. Thus it is advisable to provide other means for assuring that a representation is C^k-consistent - we are going to address this issue in our future work.

Chapter 6

Conclusions and the future work

In this work we have presented rigorous numerical algorithms to compute enclosures for the solutions to scalar DDEs with single, bounded, constant delay of the form $\dot{x} = f\left(x(t-1), x(t)\right)$. We have proved their correctness and we have measured the performance of our implementation of those algorithms on the exemplary DDEs, moreover we have developed and implemented algorithms to compute Poincaré maps associated with the given DDE.

We have implemented the algorithms in a C++ library which extends the CAPD[1] package. The implementation heavily uses template-based approach to provide high performance during computations. The library, its documentation and the sample codes together with data from numerical experiments are available from the author's web page: `http://www.ii.uj.edu.pl/~szczelir`.

The results obtained in this work are promising for the future application to the proofs of dynamical phenomena existing in DDEs. However, to achieve this we need to overcome several difficulties and questions that we have encountered during our research:

- how to choose good coordinate frame for initial function representation to be able to create covering relations (needed in the proofs of unstable periodic orbits)? How to compute rigorously inverses of big and usually close to singular matrices, that comes from choosing (approximate) eigenvectors of the reduced Jacobian matrix of a Poincaré map as the coordinate frame?

- why is the doubleton Lohner set with the cuboid error term representation worse than the doubleton Lohner set with interval error term representation? What is the performance of the QR decomposition in this case?

- is it possible to implement C^1-computations from the work [33] (or some kind of their analog) in the context of DDEs to be able to prove uniqueness of the solutions and other important features of the solutions?

- test other strategies for computing ε-methods and access their performance.

- how to check *a priori* if a representation is C^k-consistent without need to do many steps of basic numerical method?

Besides overcoming those theoretical problems we would like also to extend our methods to include any number of equations (systems of DDE), any number of discrete delays and (if possible) time- and state- dependent, positive delays.

We hope that we will be able to overcome those problems and to provide in near future an unified framework to create computer assisted proofs in DDEs.

Bibliography

[1] CAPD library: `http://capd.ii.uj.edu.pl`, 2014.

[2] F Bartha and P. Zgliczyński. Notes on rigorous integration of DDEs. *Unpublished*.

[3] A. Bellen and M. Zennaro. *Numerical Methods for Delay Differential Equations*. Oxford University Press, New York, 2003.

[4] S. A. Campbell, R. Edwards, and P. van den Driessche. Delayed coupling between two neural network loops. *SIAM J. Appl. Math*, 65(1), 2004.

[5] S. M. Ciupe, B. L. de Bivort, D. M. Bortz, and P. W. Nelson. Estimates of kinetic parameters from HIV patient data during primary infection through the eyes of three different models. *Math. Biosci.*, 200(1), 2006.

[6] K. Cooke, Y. Kuang, and B. Li. Analyses of an antiviral immune response model with time delays. *Canad. Appl. Math. Quart.*, 6(4), 1998.

[7] K. L. Cooke, P. van den Driessche, and X. Zou. Interaction of maturation delay and nonlinear birth in population and epidemic models. *J. Math. Biol.*, 39, 1999.

[8] R.D. Driver. *Ordinary and Delay Differential Equations*. Springer-Verlag, New York, 1977.

[9] J.E. Forde. *Delay Differential Equation Models in Mathematical Biology*. Ph.D Thesis, The University of Michigan, 2005.

[10] H. Hardin. *Handling Biological Complexity: as simple as possible but not simpler*. Ph.D Thesis, VU University, Amsterdam, 2010.

[11] E. Jarlebring. *The spectrum of delay-differential equations: numerical methods, stability and perturbation*. Ph.D Thesis, 2008.

[12] T. Kapela and C. Simó. Computer assisted proofs for nonsymmetric planar choreographies and for stability of the Eight. *Nonlinearity*, 20(5):1241, 2007.

[13] G. Kiss and J. Lessard. Computational fixed-point theory for differential delay equations with multiple time lags. *Journal of Differential Equations*, 252(4):3093 – 3115, 2012.

[14] T. Krisztin and G. Vas. Large-Amplitude Periodic Solutions for Differential Equations with Delayed Monotone Positive Feedback. *J. Dyn. Diff. Eq.*, 23, 2011.

[15] R.J. Lohner and Gladwel (ed) I. Cach (ed), J.R. *Computation of Guaranteed Enclosures for the Solutions of Ordinary Initial and Boundary Value Problems: in* Computational Ordinary Differential Equations. Clarendon Press, Oxford, New York, 1992.

[16] R.E. Moore. *Interval Analysis*. Prentice Hall, 1966.

[17] P. W. Nelson, J. D. Murray, and A. S. Perelson. A model of HIV-1 pathogenesis that includes an intracellular delay. *Math. Biosci.*, 163, 2000.

[18] L.B. Rall. *Automatic Differentiation: Techniques and Applications. In: Lecture Notes in Computer Science, vol 120*. Springer Verlag, 1981.

[19] F. M. Scudo and J.R. Ziegler. The Golden Age of Theoretical Ecology:1923-1940, A collection of the Works of V. Volterra, V.A. Kostitzin, A.J. Lotka, and A.N. Kolmogoroff. *Lect. Notes in Biomathematics*, 22, 1978.

[20] P. Smolen, D. Baxter, and J. Byrne. A reduced model clarifies the role of feedback loops and time delays in the Drosophila circadian oscillator. *Biophys. J.*, 83, 2002.

[21] R. Szczelina and P. Zgliczyński. A Homoclinic Orbit in a Planar Singular ODE—A Computer Assisted Proof. *SIAM Journal on Applied Dynamical Systems*, 12(3):1541–1565, 2013.

[22] Robert Szczelina. Author's home page: `www.ii.uj.edu.pl/~szczelir`, 2014.

[23] P. Turchin. Rarity of density dependence or population regulation with lags. *Nature*, 1990.

[24] P. Turchin and Taylor A. D. Complex dynamics in ecological time series. *Ecology*, 73, 1992.

[25] B. Vielle and G. Chauvet. Delay equation analysis of human respiratory stability. *Math.Biosci.*, 152(2), 1998.

[26] M. Villasana and A. Radunskaya. A delay differential equation model for tumor growth. *J. Math. Biol.*, 47(3), 2003.

[27] D. Wilczak. Chaos in the Kuramoto-Sivashinsky equations-a computer-assisted proof. *Journal of Differential Equations*, 194(2):433 – 459, 2003.

[28] D. Wilczak. The Existence of Shilnikov Homoclinic Orbits in the Michelson System: A Computer Assisted Proof. *Foundations of Computational Mathematics*, 6(4):495–535, 2006.

[29] D. Wilczak and P. Zgliczyński. Computer Assisted Proof of the Existence of Homoclinic Tangency for the Hénon Map and for the Forced Damped Pendulum. *SIAM Journal on Applied Dynamical Systems*, 8(4):1632–1663, 2009.

[30] J. Ying, S. Guo, and Y. He. Multiple periodic solutions in a delay-coupled system of neural oscillators. 2011.

[31] C. Yu and J. Wei. Stability and bifurcation analysis in a basic model of the immune response with delays. *Chaos, Solitons and Fractals*, 41, 2009.

[32] M. Zalewski. Computer-assisteed proof of a periodic solution in a non-linear feedback DDE. *ArXiv Preprint*, 2007.

[33] P. Zgliczyński. C^1-Lohner algorithm. *Foundations of Computational Mathematic*, 2, 2002.

[34] P. Zgliczyński. Rigorous Numerics for Dissipative Partial Differential Equations II. Periodic Orbit for the Kuramoto-Sivashinsky PDE; A Computer-Assisted Proof. *Found. Comput. Math.*, 4(2):157–185, April 2004.

[35] P. Zgliczyński. Computer assisted proofs in dynamics - lecture notes: `www.ii.uj.edu.pl/~zgliczyn`, 2007.

[36] T. Zhao. Global periodic solutions for a differential delay system modeling a microbial population in the chemostat. *J. Math. Anal. Appl.*, 193, 1995.