

# Middle Point Reduction of the Chain-recurrent Set

Carlos Argáez<sup>1</sup><sup>a</sup>, Peter Giesl<sup>2</sup><sup>b</sup> and Sigurdur Freyr Hafstein<sup>1</sup><sup>c</sup>

<sup>1</sup>Science Institute, University of Iceland, Dunhagi 5, 107 Reykjavík, Iceland

<sup>2</sup>Department of Mathematics, University of Sussex, Falmer, BN1 9QH, U.K.

**Keywords:** Lyapunov Functions, Chain-recurrent Set, Programming, Algorithm, Mathematics, Dynamical Systems.

**Abstract:** Describing dynamical systems requires capability to isolate periodic behaviour. In Lyapunov's theory, the qualitative behaviour of a dynamical system given by a differential equation can be described by a scalar function that decreases along solutions: the Complete Lyapunov Function. The chain-recurrent set will produce constant values of an associated complete Lyapunov function and zero values of its orbital derivative. Recently, we have managed to isolate the chain-recurrent set of different dynamical systems in 2- and 3- dimensions. An overestimation, however, is always obtained. In this paper, we present a method to reduce such overestimation based on geometrical middle points for 2-dimensional systems.

## 1 INTRODUCTION

The study of dynamical systems offers different methodologies to approach time-evolving phenomena expressed as ordinary differential equations. Several disciplines find in its semantics a powerful descriptive language and in its results, useful tools to obtain solutions. From a geometrical point of view, a dynamical system describes the time-evolution of a point in a state-space or geometrical manifold. Practically, time-evolving phenomena can often be described as a time-autonomous system of differential equations given by the expression (1),

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}), \quad (1)$$

where  $\mathbf{x} \in \mathbb{R}^n$  and  $n \in \mathbb{N}$ . Furthermore, while seen as function of its components,  $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is a vector field while  $\dot{\mathbf{x}}$  represents the derivative against time. Systems like (1), in which  $\mathbf{f}(\mathbf{x})$  does not explicitly depend on the time  $t$ , are called autonomous. However, the solutions to (1) lead to time-dependent solutions unless  $\mathbf{f}(\mathbf{x})$  is identically zero.

The methodology to study and describe the behaviour of dynamical systems can vary. Since an analytical solution is only available in the simplest cases, one could attempt to solve numerically the differential equation system (1) for a large collection of ini-


tial conditions. That would indeed describe the system but only under a copious effort and would require to find a solution curve  $S_t\xi$ , defined as  $S_t\xi := \mathbf{x}(t)$  for numerous initial condition  $\mathbf{x}(0) = \xi \in \mathbb{R}^n$ .


Computing the invariant manifolds which form the boundaries of the attractors' basins of attraction (Krauskopf et al., 2005) is also a well-known methodology along with set oriented methods (Dellnitz and Junge, 2002) or the cell mapping approach (Hsu, 1987). All these methods require large computational effort.


When the stability of a dynamical system is of interest, one could turn to what today is known as Lyapunov's second method for stability. This was proposed by Aleksandr Lyapunov (Lyapunov, 1992) in 1893 and consists in constructing a scalar function, nowadays known as a Lyapunov function. It is built around an equilibrium point, or more generally an attractor of the dynamical system. The explicit solution to the differential equation is not required.

The equilibrium points are the zeros of the right-hand side, i.e. points  $\mathbf{x}_0$  with  $\mathbf{f}(\mathbf{x}_0) = 0$ , implying that any solution taking it as an initial condition is an equilibrium, i.e. will not change in time for it is a constant solution. An attractor, however, is a set whose neighbourhood's points will provide solutions that tend to the attractor as time grows.

An equilibrium point is called stable if close points produce close-remaining solutions. It is called attractive if all adjacent solutions will tend to it as time grows. That means that for an attractive equi-

<sup>a</sup> <https://orcid.org/0000-0002-0455-8015>

<sup>b</sup> <https://orcid.org/0000-0003-1421-6980>

<sup>c</sup> <https://orcid.org/0000-0003-0073-2765>

librium  $\mathbf{x}_0$  one can find a ball  $B_\delta(\mathbf{x}_0)$ , with  $\delta > 0$  such that  $\|S_t \mathbf{x} - \mathbf{x}_0\| \rightarrow 0, t \rightarrow \infty$  for all  $\mathbf{x} \in B_\delta(\mathbf{x}_0)$ .

The attractivity and stability are two different concepts and do not imply one another, however, when both occur, then we talk about an asymptotically stable equilibrium.

Solutions could also have the possibility of coming back to a given point after a certain time  $T$ . A set of points in which the behaviour of a dynamical system will repeat after a certain time  $T$ , i.e. is recurrent, is called a periodic orbit.

A Lyapunov function is an auxiliary scalar-valued function whose domain is a subset of the state-space, and which is strictly decreasing along all solution trajectories in a neighbourhood of an attractor, such as an asymptotically stable equilibrium point or periodic orbit. Furthermore, it attains its minimum on the attractor.

This is the classical definition of a strict Lyapunov function and it can be understood in the physical word using a simple paragon, i.e. a system with dissipative energy.

However, this function is only defined in the neighbourhood of one attractor and serves to describe the set of points whose solutions converge to the it: the basin of attraction.

For a long time it was hard to obtain a Lyapunov function for a given dynamical system. If the system under analysis is linear, it is possible to obtain a classical Lyapunov function relatively easily. However, if the system is not linear then, in general, one could not hope to be able to obtain a Lyapunov function easily.

An extension defined on the whole state space, a *complete Lyapunov function*, was introduced in (Conley, 1978; Conley, 1988; Hurley, 1995; Hurley, 1998).

A complete Lyapunov function characterizes the complete qualitative behaviour of the dynamical system on the whole phase space and not just in a neighbourhood of one particular attractor. Therefore, it allows to describe the different basins of attraction for the different attractors the dynamical system could have. It divides the state-space into two disjoint areas: The first is the gradient-like flow, where the system's trajectories flow through. The second is where infinitesimal perturbations can make the system recurrent. These two areas describe fundamentally different behaviours.

The regions in which the system is recurrent is usually referred to as the chain-recurrent set.

An  $\varepsilon$ -trajectory is arbitrarily close to a true system's solution and a point in the chain-recurrent set is recurrent or almost recurrent; for a precise definition see, e.g. (Conley, 1978). The dynamics outside of the chain-recurrent set are similar to a gra-

dient system, i.e. a system (1) where the right-hand side  $\mathbf{f}(\mathbf{x})$  is given by the gradient  $\nabla U(\mathbf{x})$  of a function  $U: \mathbb{R}^n \rightarrow \mathbb{R}$ .

**Definition 1.1** (Complete Lyapunov Function). *Let us consider an autonomous system of differential equations described by,*

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^n$$

*A complete Lyapunov function is a continuous scalar function,  $V: \mathbb{R}^n \rightarrow \mathbb{R}$  that:*

- *Is constant in each chain-transitive component of the chain-recurrent set*
- *Decreases strictly along solution trajectories in other places*

These conditions imply that, if  $V$  is differentiable, its time-derivative should be zero or decreasing:

$$\frac{dV(\mathbf{x})}{dt} = \nabla V(\mathbf{x}) \cdot \frac{d\mathbf{x}}{dt} = \nabla V(\mathbf{x}) \cdot \dot{\mathbf{x}} = \nabla V(\mathbf{x}) \cdot \mathbf{f}(\mathbf{x}) \leq 0.$$

This quantity is called orbital derivative.

Conley (Conley, 1978) gave the first mathematical existence proof for complete Lyapunov functions of a dynamical system defined on a compact metric space. Hurley (Hurley, 1998) extended these results to separable metric spaces. More recent results are found in (Fathi and Pageault, 2019; Bernhard and Suhr, 2018).

In this paper we use a computationally efficient algorithm to compute a complete Lyapunov function using Radial Basis Functions, which was introduced for classical Lyapunov functions in (Giesl, 2007). It has been developed in (Argáez et al., 2017; Argáez et al., 2018b; Argáez et al., 2018c; Giesl et al., 2018; Argáez et al., 2018a).

The general idea is to approximate a "solution" to the ill-posed problem  $V'(\mathbf{x}) = -1$ , where  $V'(\mathbf{x}) = \nabla V(\mathbf{x}) \cdot \mathbf{f}(\mathbf{x})$  is the derivative along solutions of the ODE, i.e. the orbital derivative. This approximation is inspired by the fact that the orbital derivative has to be negative everywhere except in areas in which the system has recurrent behaviour.

A function  $v$  is computed using Radial Basis Functions, a mesh-free collocation technique, such that  $v'(\mathbf{x}) = -1$  is fulfilled at all points  $\mathbf{x}$  in a finite set of collocation points  $X$ .

The discretized problem of computing  $v$  is well-posed and possesses a unique solution. However, the computed function  $v$  cannot fulfil the PDE  $V'(\mathbf{x}) = -1$  at all points of the chain-recurrent set, such as an equilibrium or a periodic orbit. This is the key component to locate the chain-recurrent set with our general algorithms; we use the area where  $v$  fails to fulfil the PDE to determine the chain-recurrent set. The

points in this set have vanishing values over the orbital derivative.

However, points in the neighbourhood of the chain-recurrent set, tend to have an orbital derivative near zero. That causes an overestimation of the chain-recurrent set. There have been systematic approaches to solve of this problem. First, in (Argáez et al., 2018b) it was observed that for systems, in which the speed varies greatly along solutions, the orbital derivative can vary greatly. This caused a considerable overestimation of the chain-recurrent set. A normalization condition was introduced to homogenize the speed at all points of the system. Later in Section 2 this step will be further explained.

Second, in (Argáez et al., 2019b), an effort was made to understand the importance of a clustering algorithm to classify different orbits and equilibria. That was introduced firstly as a test over circular orbits only. Based on the geometrical constraints, tolerance radii were defined to reduce the overestimation of the chain-recurrent set. The general idea is summarized as follows:

- Obtain an approximation to the chain-recurrent set
- For each circular orbit in the chain-recurrent set with radius  $r$ , define two new radii,  $r_{max}$  and  $r_{min}$ , then

$$r_1 = r_{min} + 0.52 * (r_{max} - r_{min})$$

$$r_2 = r_{max} - 0.52 * (r_{max} - r_{min})$$

- Remove from the chain-recurrent set all points with Euclidian norm  $r \notin [r_2, r_1]$

Obvious limitations arise with this approach as it only works with circular shapes, but it showed greatly improved results for adequate systems.

Later in (Argáez et al., 2019a), an algorithm capable of decomposing the chain-recurrent set into its connected components was presented. That was done regardless of the shapes of the components.

Now, a large chain-recurrent set containing equilibria and orbits of different shapes, would be classified in subsets. However, no algorithm to reduce the overestimation was given in that work.

Now, in this paper we will propose a reduction algorithm capable of reducing the overestimation of the chain-recurrent set. The idea for the reduction is based on exploiting the geometrical properties of the collocation grid used to compute the Lyapunov function. That means that we compute middle points of different cleverly-chosen sections of the chain-recurrent set. The union of these middle points is used as a reduced estimation of the chain-recurrent set.

In Section 2 a brief description of how to compute a complete Lyapunov function is given. Then in

Section 3 the novel algorithm to reduce the overestimation is explained. In Section 4 examples of its application are given, which are discussed in Section 5. Section 6 analyses the computation cost of this algorithm and finally Section 7 will summarize the results of this work.

## 2 CONSTRUCTION OF COMPLETE LYAPUNOV FUNCTIONS

### 2.1 Mesh-free Collocation

Radial Basis Functions (RBF) are a powerful methodology to construct complete Lyapunov functions when posed as a generalized interpolation problem (Wendland, 1998)

They are real-valued functions whose evaluation depends only on the distance from the origin. Common examples of RBFs are Gaussians and multi-quadrics.

In this work, we use Wendland functions as RBF, which are compactly supported and positive definite functions (Wendland, 1998). They have the advantage of being expressed as algebraic polynomials on their compact support.

Further, the corresponding Reproducing Kernel Hilbert Space  $H$  is norm-equivalent to a Sobolev space.

### 2.2 Wendland Functions

The general form of a Wendland function is  $\psi(\mathbf{x}) := \Psi_{l,k}(c\|\mathbf{x}\|)$ , where  $c > 0$  determines the size of the compact support and  $k \in \mathbb{N}$  is a smoothness parameter. For our application the parameter  $l$  is fixed as  $l = \lfloor \frac{n}{2} \rfloor + k + 1$ . The Reproducing Kernel Hilbert Space corresponding to  $\Psi_{l,k}$  contains the same functions as the Sobolev space  $W_2^{k+(n+1)/2}(\mathbb{R}^n)$  and the spaces are norm equivalent.

The functions  $\Psi_{l,k}$  are defined by the recursion: For  $l \in \mathbb{N}$  and  $k \in \mathbb{N}_0$ , we define

$$\Psi_{l,0}(r) = (1-r)_+^l, \tag{2}$$

$$\Psi_{l,k+1}(r) = \int_r^1 t \Psi_{l,k}(t) dt$$

for  $r \in \mathbb{R}_0^+$ , where  $x_+ = x$  for  $x \geq 0$  and  $x_+ = 0$  for  $x < 0$ .

### 2.3 Collocation Points

As collocation points, we use a subset,  $X = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \subset \mathbb{R}^n$ , of a hexagonal grid with fineness-parameter  $\alpha_{\text{Hexa-basis}} \in \mathbb{R}^+$  constructed according to the equation:

$$\begin{aligned} & \{\alpha_{\text{Hexa-basis}} \sum_{k=1}^n i_k \omega_k : i_k \in \mathbb{Z}\}, \\ & \omega_k = \sum_{j=1}^{k-1} \epsilon_j \mathbf{e}_j + (k+1) \epsilon_k \mathbf{e}_k \quad (3) \\ & \text{and } \epsilon_k = \sqrt{\frac{1}{2k(k+1)}}. \end{aligned}$$

Here  $\mathbf{e}_j$  is the usual  $j$ th unit vector.

These basis vectors are shown in red colour in Figure 1 in  $\mathbb{R}^2$ , while the canonical vectors are shown in black.

Basis sets: Canonical vs Hexagonal

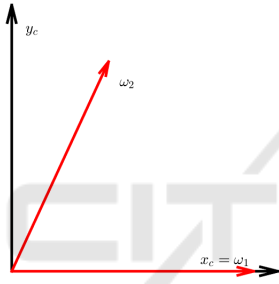


Figure 1: Black: Canonical basis. Red: Hexagonal basis set.

Although any scattered points could be used as collocation points, the hexagonal grid has been shown to minimize the condition numbers of the collocation matrices for a fixed fill distance, i.e. a measure of the density of the collocation grid (Iske, 1998).

Since  $\mathbf{f}(\mathbf{x}) = \mathbf{0}$  for all equilibria  $\mathbf{x}$ , we remove all equilibria from the set of collocation points  $X$ ; not doing so would cause the collocation matrix to be singular.

The first approximation  $v$  to a complete Lyapunov function is then given by the function that satisfies the PDE  $v'(\mathbf{x}) = -1$  at all collocation points and is norm minimal in the corresponding Reproducing Kernel Hilbert space  $H$ .

In later iterations we solve  $v'(\mathbf{x}) = r_j$  for different  $r_j \leq 0$ , determined by previous approximations.

Practically, we compute  $v$  by solving a system of  $N$  linear equations, where  $N$  is the number of collocation points.

The explicit formulas for  $v$  and its orbital derivatives are

$$\begin{aligned} v(\mathbf{x}) &= \sum_{j=1}^N \beta_j \langle \mathbf{x}_j - \mathbf{x}, \mathbf{f}(\mathbf{x}_j) \rangle \psi_1(\|\mathbf{x} - \mathbf{x}_j\|), \\ v'(\mathbf{x}) &= \sum_{j=1}^N \beta_j \left[ -\psi_1(\|\mathbf{x} - \mathbf{x}_j\|) \langle \mathbf{f}(\mathbf{x}), \mathbf{f}(\mathbf{x}_j) \rangle \right. \\ & \quad \left. + \psi_2(\|\mathbf{x} - \mathbf{x}_j\|) \langle \mathbf{x} - \mathbf{x}_j, \mathbf{f}(\mathbf{x}) \rangle \cdot \langle \mathbf{x}_j - \mathbf{x}, \mathbf{f}(\mathbf{x}_j) \rangle \right], \end{aligned}$$

where  $\psi_1, \psi_2$  are functions computed from the Wendland function  $\psi_{l,k}$  used,  $\langle \cdot, \cdot \rangle$  denotes the standard scalar product and  $\|\cdot\|$  the Euclidean norm in  $\mathbb{R}^n$ ,  $\beta \in \mathbb{R}^N$  is the solution to  $A\beta = \mathbf{r}$ ,  $r_j = r(\mathbf{x}_j)$  and  $A$  is the  $N \times N$  matrix with entries

$$a_{ij} = \psi_2(\|\mathbf{x}_i - \mathbf{x}_j\|) \langle \mathbf{x}_i - \mathbf{x}_j, \mathbf{f}(\mathbf{x}_i) \rangle \langle \mathbf{x}_j - \mathbf{x}_i, \mathbf{f}(\mathbf{x}_j) \rangle - \psi_1(\|\mathbf{x}_i - \mathbf{x}_j\|) \langle \mathbf{f}(\mathbf{x}_i), \mathbf{f}(\mathbf{x}_j) \rangle \quad (4)$$

for  $i \neq j$  and

$$a_{ii} = -\psi_1(0) \|\mathbf{f}(\mathbf{x}_i)\|^2.$$

More detailed explanations on this construction are given in (Giesl, 2007, Chapter 3).

If no collocation point  $\mathbf{x}_j$  is an equilibrium for the system, i.e.  $\mathbf{f}(\mathbf{x}_j) \neq \mathbf{0}$  for all  $j$ , then the matrix  $A$  is positive definite and the system of equations  $A\beta = \mathbf{r}$  has a unique solution.

The last assertion will hold true independently of whether the underlying discretized PDE has a solution or not, while error estimates are obviously only available if the PDE has a solution.

Note that in the context of RBF, a positive definite function  $\psi$  refers to the matrix  $(\psi(\|\mathbf{x}_i - \mathbf{x}_j\|))_{i,j}$  being positive definite for  $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ , where  $\mathbf{x}_i \neq \mathbf{x}_j$  if  $i \neq j$ .

As explained in (1), in (Argáez et al., 2018b) we introduced an “almost” normalized approach, i.e., the original dynamical system (1) was substituted by

$$\dot{\mathbf{x}} = \hat{\mathbf{f}}(\mathbf{x}), \quad \text{where } \hat{\mathbf{f}}(\mathbf{x}) = \frac{\mathbf{f}(\mathbf{x})}{\sqrt{\delta^2 + \|\mathbf{f}(\mathbf{x})\|^2}} \quad (5)$$

with a small parameter  $\delta > 0$ . This normalization already reduces significantly the overestimation of the chain-recurrent set.

### 2.4 Evaluation Grid

After we have solved the PDE numerically, i.e. forced  $V'(\mathbf{x}_j) = -r_j$  at every collocation point  $\mathbf{x}_j$ , we use an evaluation grid  $Y_{\mathbf{x}_j}$  around each collocation point  $\mathbf{x}_j$  to evaluate the results.

Placing all evaluation points aligned to the flow of the ODE system, as introduced in (Argáez et al., 2018c; Argáez et al., 2018a), has proved successful. The formula for the evaluation grid is

$$Y_{\mathbf{x}_j} = \left\{ \mathbf{x}_j \pm \frac{r \cdot k \cdot \alpha_{\text{Hexa-basis}} \cdot \hat{\mathbf{f}}(\mathbf{x}_j)}{m \|\hat{\mathbf{f}}(\mathbf{x}_j)\|} : k \in \{1, \dots, m\} \right\},$$



where  $\alpha_{\text{Hexa-basis}}$  is the parameter used to build the hexagonal grid defined above,  $r \in (0, 1)$  is the ratio up to which the evaluation points will be placed and  $m \in \mathbb{N}$  denotes the number of points in the evaluation grid that will be placed on both sides of the collocation points aligned to the flow.

Using this evaluation grid means that there will not be any evaluated points to provide information about the dynamical system other than in the direction of the flow. However, this evaluation grid avoids exponential growth of evaluation points as the system's dimension becomes higher and the results are usually good, i.e. enough information about the dynamics is gained.

We start by computing the approximate solution  $v_0$  to  $V'(\mathbf{x}) = -1$  with collocation points  $X$ . As we have previously done in (Argáez et al., 2017; Argáez et al., 2018b), we define a tolerance parameter  $-1 < \gamma \leq 0$ . In each step  $i$  of the iteration we mark a collocation point  $\mathbf{x}_j$  as being in the chain-recurrent set ( $\mathbf{x}_j \in X^0$ ) if there is at least one point  $\mathbf{y} \in Y_{\mathbf{x}_j}$  such that  $v'_i(\mathbf{y}) > \gamma$ . The points for which the condition  $v'_i(\mathbf{y}) \leq \gamma$  holds for all  $\mathbf{y} \in Y_{\mathbf{x}_j}$  are considered to belong to the gradient-like flow ( $\mathbf{x}_j \in X^-$ ).

As has been done in (Argáez et al., 2018c), in this paper we replace the right-hand side  $-1$  by the average of the values  $v'_i(\mathbf{y})$  over the evaluation grid  $\mathbf{y} \in Y_{\mathbf{x}_j}$  at each collocation point  $\mathbf{x}_j$  for averages that are negative. In cases in which the average is positive we use 0. In formulas, we calculate the approximate solution  $v_{i+1}$  of  $V'(\mathbf{x}_j) = \tilde{r}_j$  with

$$\tilde{r}_j = \left( \frac{1}{2m} \sum_{\mathbf{y} \in Y_{\mathbf{x}_j}} v'_i(\mathbf{y}) \right)_{-}$$

where  $x_{-} = \min(0, x)$ . We will refer to this as the “non-scaled” version.

However, this approach can lead to a continuous decrease of “energy” over the iterations.

Recall that the original value of the orbital derivative condition in the first iteration is  $-1$ , but the new value is obtained by averaging and bounding by 0. This can cause it to converge to zero and thus force the total “energy” of the Lyapunov function to decrease. To avoid this, we scale the condition of the orbital derivative after the first iteration onwards so that the sum of all  $r_j$  over all collocation points is constant for all iterations; we will refer to this as the “scaled” method.

Our algorithm to compute complete Lyapunov functions and classify the chain-recurrent set can be summarized as follows:

1. Create the set of collocation points  $X$  and compute the approximate solution  $v_0$  to  $V'(\mathbf{x}) = -1$ ; set  $i = 0$

2. For each collocation point  $\mathbf{x}_j$ , compute  $v'_i(\mathbf{y})$  for all  $\mathbf{y} \in Y_{\mathbf{x}_j}$ : if  $v'_i(\mathbf{y}) > \gamma$  for a point  $\mathbf{y} \in Y_{\mathbf{x}_j}$ , then  $\mathbf{x}_j \in X^0$ , otherwise  $\mathbf{x}_j \in X^-$ , where  $\gamma \leq 0$  is a chosen critical value
3. Define  $\tilde{r}_j = \left( \frac{1}{2m} \sum_{\mathbf{y} \in Y_{\mathbf{x}_j}} v'_i(\mathbf{y}) \right)_{-}$
4. Define  $r_j = \frac{N}{\sum_{l=1}^N |\tilde{r}_l|} \tilde{r}_j$ ,
5. Compute the approximate solution  $v_{i+1}$  to  $V'(\mathbf{x}_j) = r_j$  for  $j = 1, \dots, N$ ; this is the scaled version, while approximating the solution of  $V'(\mathbf{x}_j) = \tilde{r}_j$  would be the non-scaled version
6. Set  $i \rightarrow i + 1$  and repeat steps 2. to 5. until no more points are added to  $X^0$ .

Note that the sets  $X^0$  and  $X^-$  may change at each step of the algorithm.

### 3 MIDDLE POINT REDUCTION

#### 3.1 Middle Point Algorithm

Once we have obtained our approximations  $X^0$  and  $X^-$  to the chain-recurrent and gradient-like flow respectively, we want to classify the points  $X^0$  in the chain-recurrent set into its different connected components, e.g. different orbits and equilibria.

The way of doing it follows the algorithm explained in (Argáez et al., 2019a).

A brief description is given below:

Algorithm 3.1: Once the chain-recurrent set is obtained.

1. Measure all distances from the origin to the different failing points
2. Sort all points in an increasing order according to their distance from the origin
3. Measure the difference in radii-length between every two consecutive points. Gaps are considered to happen when difference in distance is greater than  $\alpha_{\text{Hexa-basis}}$  for two consecutive points
4. All points before the first gap are considered to be a part of the first set of connected components. Between the first and the second gap, all points are considered to be part of the second set of connected component, etc. After we have found all gaps, the last one of them and the longest radius length define the last set of connected components
5. Each set of component is decomposed into its connected components

Let us point out that the chain-recurrent set is given by the points in the collocation grid that failed the approximation.

Therefore, they will all be aligned to the basis set of vectors of the hexagonal grid, see Figure 2.

For each point in this classification of the chain-recurrent set, we will build two groups. One will consist of all its neighbours and neighbours of its neighbours, etc., aligned to the first basis vector. The second will consist of all its neighbours and neighbours of its neighbours, etc., aligned to the second basis vector.

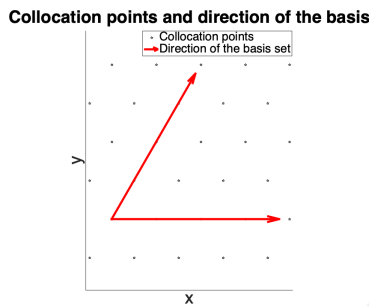


Figure 2: Points in the collocation grid aligned to the hexagonal basis set. Distance between two consecutive points is  $\alpha_{Hexa-basis}$ .

Given a point  $\mathbf{x}_c$  in a sub-classification of the chain-recurrent set, a way of obtaining the most direct neighbours in a direction of a vector  $\omega_k$  in the hexagonal basis (3) is to “walk” positively and negatively in the sense of that direction, with steps of size  $\alpha_{Hexa-basis}$ . That is, consider the points  $\mathbf{x}_c + j\alpha_{Hexa-basis}\omega_k$  for integer  $j$ . Then, the neighbours of its neighbours will be found with the “next step”, etc. When a next  $p$ th-step is given and no point is found, then the search in that direction’s sense is stopped. The total search of neighbours for that point stops when both direction’s senses, the positive and the negative, reach a point that was not originally in the chain-recurrent set. Then, the new sub-classification group continues with another point.

Once we have grouped all the points in the sub-classification of chain-recurrent set, then each individual group is organized by its different distances between points. The two most far-apart points in each group are the boundaries of that group in that direction. These two points give, in that particular direction, the middle point that we are looking for to obtain the reduction.

We consider that the reduced chain-recurrent set is the total union of the middle points for each point’s group in both directions.

Let us recall that the middle point of two points  $(x_1, y_1)$  and  $(x_2, y_2)$  is given by,

$$x_m = \frac{x_1 + x_2}{2}, \quad y_m = \frac{y_1 + y_2}{2} \quad (6)$$

Single points in the orbit will be themselves after the reduction as they are, themselves, their middle-points. For one dimensional chain-recurrent set, the algorithm is listed bellow.

Begin

- 1) Construct the Lyapunov function and its orbital derivative;
  - 2) Obtain the chain-recurrent set;
  - 3) Divide it into its connected components
  - 4) For each point in the chain-recurrent set build a group of adjacent collocation points in direction  $\omega_1$  and one in direction  $\omega_2$
  - 5) Identify groups that are equal
  - 6) Find the extreme points in each group and replace the points in the group by the middle point
  - 7) The reduced chain-recurrent set is the collection of all middle points
  - 8) Merge all groups aligned with  $\omega_2$  into a another set of groups
  - 9) For each group apply the middle points formula
  - 10) The reduced chain-recurrent set is the collection of all middle points
- End.

## 4 RESULTS

### 4.1 Two Periodic Orbits

We consider system (1) with right-hand side

$$\mathbf{f}(x, y) = \begin{pmatrix} -x(x^2 + y^2 - 1/4)(x^2 + y^2 - 1) - y \\ -y(x^2 + y^2 - 1/4)(x^2 + y^2 - 1) + x \end{pmatrix}. \quad (7)$$

This system has an asymptotically stable equilibrium at the origin. Moreover, the system has two periodic circular orbits: an asymptotically stable periodic orbit at  $\Omega_1 = \{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 = 1\}$  and a repelling periodic orbit at  $\Omega_2 = \{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 = 1/4\}$ .

To compute the complete Lyapunov function with our method we used the Wendland function  $\psi_{5,3}$ . The collocation points were set in a region  $[-1.5, 1.5] \times [-1.5, 1.5] \subset \mathbb{R}^2$  and we used a hexagonal grid (3) with  $\alpha_{Hexa-basis} = 0.0162$ . The evaluation grid was computed with the directional grid (6) with parameter  $m = 50$ .

We computed this example with the almost-normalized method  $\dot{\mathbf{x}} = \hat{\mathbf{f}}(\mathbf{x})$  with  $\delta^2 = 10^{-8}$  and  $\gamma = -0.8$ .

Figure 4 shows the complete Lyapunov function and its orbital derivative computed using 3 iterations. The two orbits with radii 1/2 and 1 can clearly be

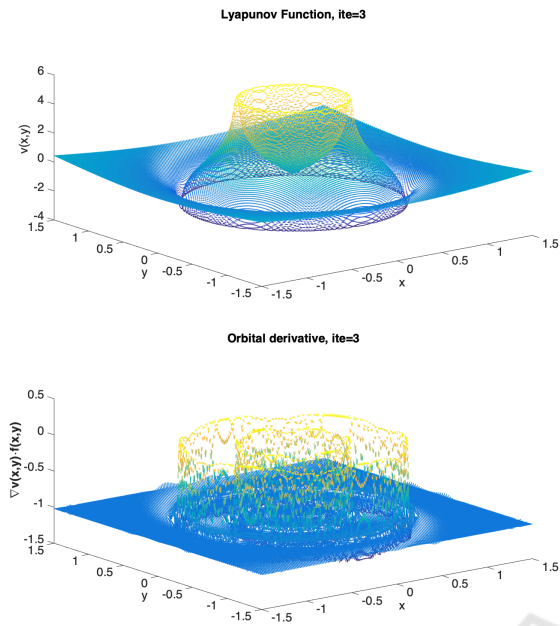


Figure 3: Upper: Complete Lyapunov function for system 7. Lower: Orbital derivative for system 7.

identified. The first one is a repeller and the second is attractive. We see that beyond the second orbit with radius 1, the behaviour of the Lyapunov function is well behaved and its orbital derivative whose value approximated  $-1$  nicely. However, at the orbits and close to the equilibrium at zero, we can see that the orbital derivative fails the approximation. The collocation points associated to the failing point are shown in the last figure of Figure 3 in dark-yellow colour. That is our approximation to the chain-recurrent set. The middle-point reduced the chain-recurrent set over the vector basis  $(1,0)$  is shown in the upper figure of 3. The two dots representing the equilibrium point at the origin, the middle point reduction, are the same as the original ones before reduction. They did not have neighbours within a  $\alpha_{Hexa-basis}$  distance. As such, the middle-up figure shows the middle-point reduced chain-recurrent set over the vector basis  $(1/2, \sqrt{3}/2)$ , Figure 3. It is interesting to notice that in different directions there will be different areas without points. It is the union of both groups, lowest figure in Figure 3, that gives the total reduction.

The final reduction is shown over the first overestimated chain-recurrent set at the bottom of Figure 3. Let us show in numbers what the reduction means, Table 1.

Table (1) shows the number of elements in the chain-recurrent set in both cases: The non-reduced and the reduced one. As can be seen, the number of elements after the reduction can be a half of the amount of numbers before the reduction.

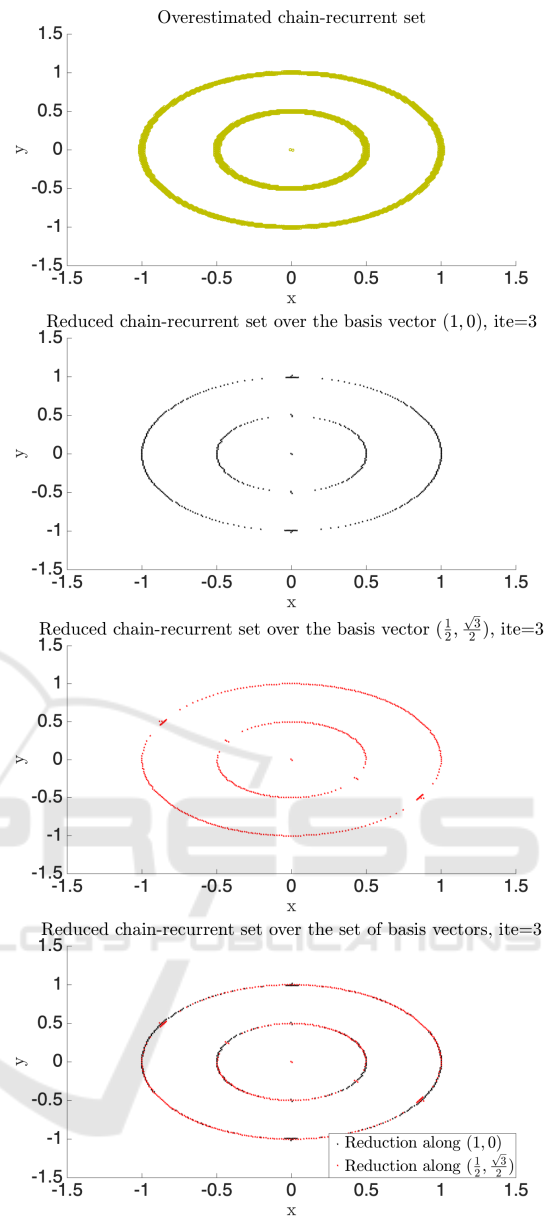


Figure 4: Upper: Overestimated chain-recurrent set with  $\gamma = 0.8$ . Middle up: Middle point reduction over the hexagonal basis vector  $(1,0)$ . Middle down: Middle point reduction over the hexagonal basis vector  $(\frac{1}{2}, \frac{\sqrt{3}}{2})$ . Bottom: Both reductions. All figures for system (7).

Table 1: Difference in the total amount of points per section of the chain-recurrent set for example (7).

Section	Before	After
Equilibrium point	2	2
Orbit $r = 1/2$	612	256
Orbit $r = 1$	1062	512

### 4.2 Van der Pol Oscillator

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \mathbf{f}(x,y) = \begin{pmatrix} y \\ (1-x^2)y-x \end{pmatrix}. \quad (8)$$

For computing the complete Lyapunov function associated to system (8), we set  $\alpha_{\text{Hexa-basis}} = 0.043$  over the area defined by  $[-4,4]^2 \subset \mathbb{R}^2$ . The Wendland function parameters used are  $(l,k,c) = (4,2,1)$ , the critical value  $\gamma = -0.5$ , and  $\delta^2 = 10^{-8}$ . The evaluation grid was computed with the directional grid with  $m = 10$ . The complete Lyapunov function at the initial iteration and the orbital derivative over the chain-recurrent set is shown in Figure 5.

As can be seen, the Lyapunov function has a repeller at the origin and it has a non-circular but still symmetric orbit around the origin. Away from the orbit, the orbital derivative satisfies the condition  $-1$  nicely but again the approximation fails in the orbit, the repeller and the points nearby.

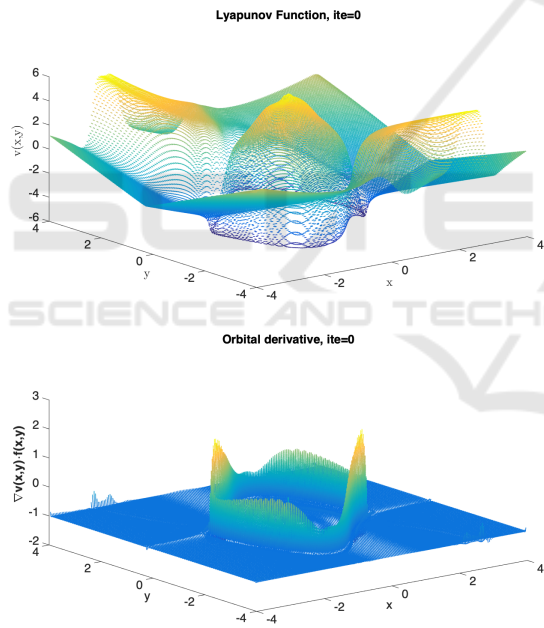


Figure 5: Upper: Complete Lyapunov function for system 8. Lower: Orbital derivative for system (8).

Our approximated chain-recurrent set is shown in the last figure of Figure 5 in dark-yellow colour. Again the upper figure shows the reduction along  $(1,0)$  and middle-upper figures the reduction along  $(\frac{1}{2}, \frac{\sqrt{3}}{2})$ . The total middle-point reduction is again shown in figure middle-lower of Figure 6. The reduction becomes clear when seen over the original estimation of the chain-recurrent set, lower figure in Figure 6.

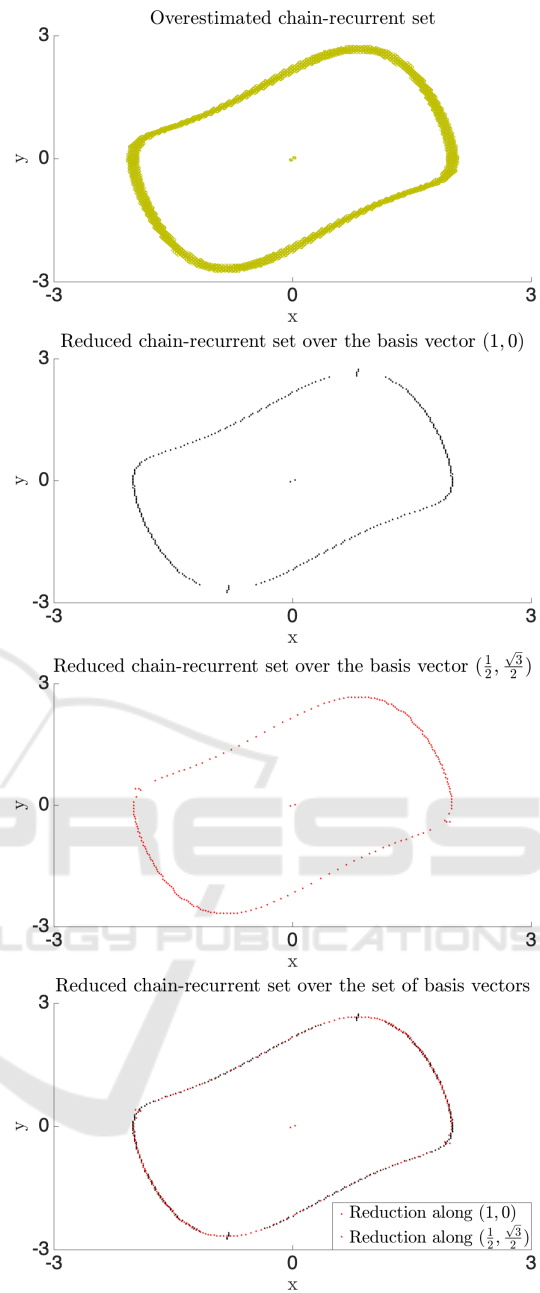


Figure 6: This figure shows both the overestimation and the reductions. It is divided in four subfigures. Upper: Overestimated chain-recurrent set with  $\gamma = 0.043$ . Middle up: Middle point reduction over the hexagonal basis vector  $(1,0)$ . Middle down: Middle point reduction over the hexagonal basis vector  $(\frac{1}{2}, \frac{\sqrt{3}}{2})$ . Bottom: Both reductions. All figures for system (8).

In Table 2 the reduction is quantified. The table shows that the number of elements is reduced to circa a third of its original overestimated value.



Table 2: Difference in the total amount of points per section of the chain-recurrent set for example (8).

Section	Before	After
Equilibrium point	2	2
Limiting orbit	1212	444

### 4.3 Homoclinic Orbit

As in (Argáez et al., 2017), we also consider here the following example

$$\begin{aligned} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} &= \mathbf{f}(x,y) \\ &= \begin{pmatrix} x(1-x^2-y^2) - y((x-1)^2 + (x^2+y^2-1)^2) \\ y(1-x^2-y^2) + x((x-1)^2 + (x^2+y^2-1)^2) \end{pmatrix}. \end{aligned} \quad (9)$$

The origin is an unstable focus and the system has an asymptotically stable homoclinic orbit at a circle centred at the origin and with radius 1, connecting the equilibrium  $(1,0)$  with itself. We used the Wendland function  $\psi_{4,2}$  for our computations. Our collocation points were defined in the region  $[-1.1, 1.1] \times [-1.1, 1.1] \subset \mathbb{R}^2$  with a hexagonal grid (3) with  $\alpha_{\text{Hexa-basis}} = 0.011$ . In this example, we have used the normalized method, i.e. we replaced  $\mathbf{f}$  by  $\hat{\mathbf{f}}$  as in (5) with  $\delta^2 = 10^{-8}$ , and we used  $\gamma = -0.75$ . As before, the evaluation grid was computed with the directional grid with  $m = 10$ . The complete Lyapunov function and its orbital derivative are shown in Figure 7.

The Lyapunov function has a maximum at the origin which is a repelling equilibrium point. The orbit can be seen at radius 1. The orbital derivative condition approximates very well the condition  $-1$  everywhere except in the orbit and in the origin. The approximated chain-recurrent set is shown at the lower figure of Figure 7 in dark-yellow colour. Figure 8 has four subfigures. The upper figure shows the overestimation. The middle-upper one is the middle-point reduction over  $(1,0)$ . The middle-lower is the middle-point reduction over  $(\frac{1}{2}, \frac{\sqrt{3}}{2})$ . The lowest is the union of both.

Let us show in numbers what the reduction means, table 3.

Table 3: Difference in the total amount of points per section of the chain-recurrent set for problem 9.

Section	Before	After
Equilibrium point	255	86
Homoclinic orbit	1579	941

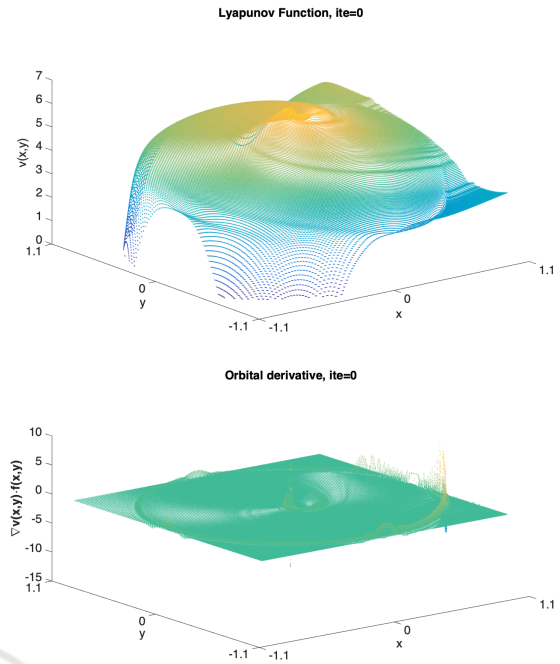


Figure 7: Upper: Complete Lyapunov function for system 9. Lower: Orbital derivative for system 9. These figures were obtained after just one iteration.

## 5 DISCUSSION

Let us discuss the results by analysing first the homoclinic orbit example. The orbit has a circumference of radius  $r = 1$ . The  $\alpha_{\text{Hexa-basis}}$  parameter for that problem was set to be  $\alpha_{\text{Hexa-basis}} = 0.011$ . That would let us assume that the total amount of collocation points that should be given in the orbit is the total perimeter divided by  $\alpha_{\text{Hexa-basis}}$ , that is:  $P/\alpha_{\text{Hexa-basis}} = 2\pi r/0.011 = 571$  points. In that calculation  $P$  represents the perimeter of the orbit. The first overestimated approximation we obtained had 1579 on the orbit with radius  $r = 1$ , so it was 276% overestimated. The middle point reduction gave 165% overestimation. So, 111% less.

One could think that these new results happen to be overestimated as well. However, this is not the case. They are in fact good results because the middle point reduction is placing points over the orbit where collocation points were not even defined. That is, between two different collocation points. Figure 9 shows this in a pedagogical way:

Figure 9 shows middle-point reductions in between different collocation points. So, the distance between two consecutive points in the reduction is now,  $\alpha_{\text{Hexa-basis}}/2 = 0.0055$ . In the ideal case in which all points were placed equidistantly that would make

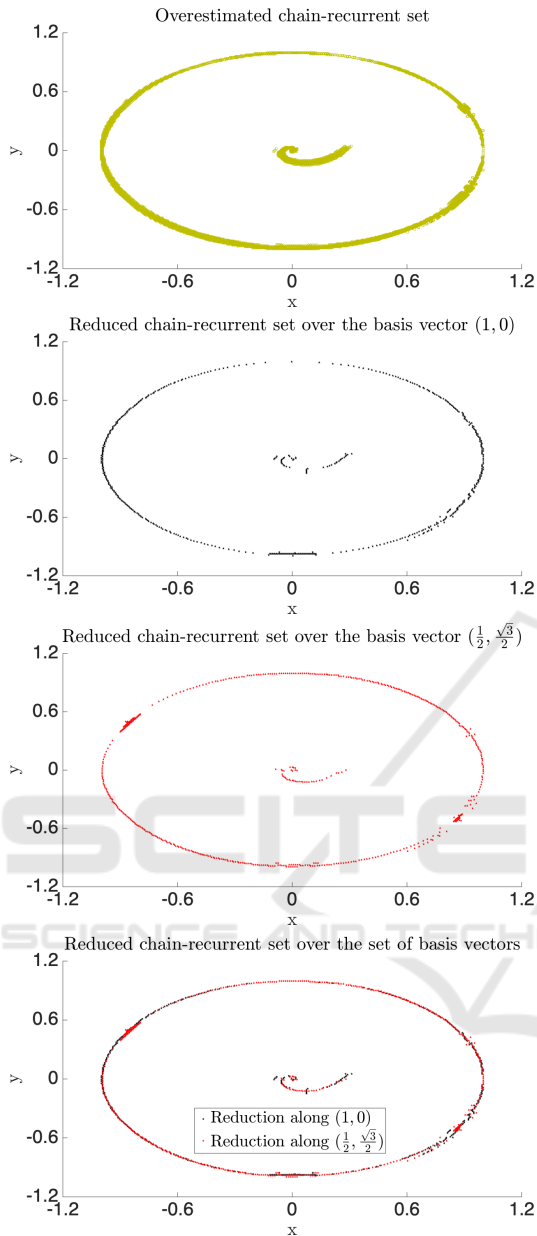


Figure 8: Upper: Overestimated chain-recurrent set with  $\gamma = -0.75$ . Middle up: Middle point reduction over the hexagonal basis vector  $(1,0)$ . Middle down: Middle point reduction over the hexagonal basis vector  $(\frac{1}{2}, \frac{\sqrt{3}}{2})$  Bottom: Both reductions. All figures for example (9).

a total number of  $P/\alpha_{\text{Hexa-basis}} = 2\pi r/0.0055 = 1142$  points. However, not all points in the middle point reduction are placed equidistantly.

Now, the equilibrium point is overestimated with 86 points. That is because the chain-recurrent set approximation failed from the beginning with 255 points. That makes a reduction of over 70%.

Similar analysis can be made for examples (7) and

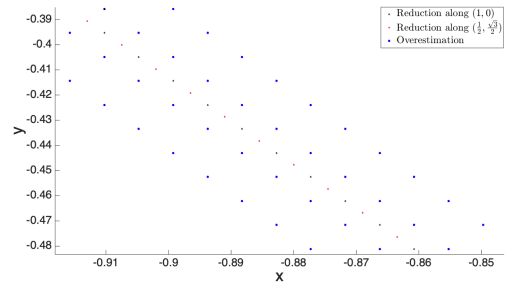


Figure 9: Middle point reductions seen around the collocation points.

(8). However, in these last cases, the equilibrium point is better estimated.

These results lead us to enunciate the following lemma.

**Lemma 5.1** (Middle point reduction). *Under the middle-point reduction to a closed orbit of a Lyapunov function constructed with the algorithm described in Sec. 2. The cardinality of elements in the reduction will be bounded below by  $P_o/\alpha_{\text{Hexa-basis}}$ .*

Where  $P_o > 1$  is the perimeter of the closed orbit and  $\alpha_{\text{Hexa-basis}} < 1$  is the parameter to build collocation grid.

*Proof.* Given the geometric conditions used to build the collocation points, the reduced points can be as much as  $\alpha_{\text{Hexa-basis}}$  apart. That will only happen if all the middle points reductions give a point that already belong to the collocation grid. If on the other hand, the middle point reduction gives only new points, then by construction the new points will be placed between elements of the collocation grid and again, they will be apart by  $\alpha_{\text{Hexa-basis}}$ .

Therefore, the total amount of points in the reduction will be always bounded below by  $P_o/\alpha_{\text{Hexa-basis}}$ .  $\square$

## 6 OBSERVATIONS

In (Argáez et al., 2019c), we made a large analysis of the cost of computing an approximation to a complete Lyapunov function with our method using  $I$  iterations. That analysis is carried out under the big-oh approach.

It was proven that our computation is as expensive as  $O(N^3 + IN^2nm)$  where  $N$  is the total number of collocation points,  $n$  is the dimension of the problem and  $m$  the total amount of points to be evaluated in the evaluation grid.

Generating the different groups aligned to the different vector basis set is built with two for-loops per orbit. However, the total amount of failing points,  $\mu$ , i.e. points in the chain-recurrent set, is much lower

than  $N$ . So, a for-loop is carried to all points in the chain-recurrent set, per each point,  $p$  steps are given in both directions of the basis set. Finally, each new point is checked to belong to the chain-recurrent set. So, the total cost is  $\mu^2 p$  with  $p < \mu < N$ . Therefore, the total cost of constructing the complete Lyapunov function remains  $O(N^3 + IN^2nm)$ .

It is to be pointed out that the example 8 is actually an example arising from engineering. It represents the limiting cycles in electrical circuits built with vacuum tubes. Therefore this is a real-life application of our methodology.

## 7 CONCLUSIONS

We have introduced an algorithm capable of reducing the overestimation of the chain-recurrent set. This algorithm is based on exploring the geometrical constraints used to construct the complete Lyapunov function in the first place. Grouping the elements of the chain-recurrent subsets (or orbits) into different small groups of points aligned to the hexagonal basis vectors allowed us to obtain the corresponding middle points. The new points obtained over the orbit can be added to the collocation points for further iterations and constraints. However, that will be done in future work.

An important observation to be made is that although the elements of the different groups are always elements of the hexagonal collocation grid, the middle points might not be part of it. However, it was to be expected that the continuous orbit would pass through the spaces between two consecutive collocation points. That enlightens the fact that the denser the collocation points grid is, the better the results.

Furthermore, since we know that these points have zero orbital derivative, one could now re-build the complete Lyapunov function with the right condition on that particular set without forcing extra points to be zero; optimizing our new approximation to the complete Lyapunov function.

However, a remaining problem to solve is an equivalent algorithm capable to work in higher dimensions.

Finally, our method has been applied to a real-world application problem arising from electrical engineering, as our results for equation (8) shows.

## ACKNOWLEDGEMENTS

The first author in this paper is supported by the Icelandic Research Fund (Rannís) grant number 163074-

052, Complete Lyapunov functions: Efficient numerical computation.

## REFERENCES

- Argáez, C., Giesl, P., and Hafstein, S. (2017). *Analysing dynamical systems towards computing complete Lyapunov functions*. In Proceedings of the 7th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH), pages 134–144. Madrid, Spain.
- Argáez, C., Giesl, P., and Hafstein, S. (2018a). *Computation of complete Lyapunov functions for three-dimensional systems*. In Proceedings IEEE Conference on Decision and Control (CDC), 2018, pages 4059–4064. Miami Beach, FL, USA.
- Argáez, C., Giesl, P., and Hafstein, S. (2018b). *Computational approach for complete Lyapunov functions*. In Dynamical Systems in Theoretical Perspective. Springer Proceedings in Mathematics & Statistics. ed. Awrejcewicz J. (eds), volume 248.
- Argáez, C., Giesl, P., and Hafstein, S. (2018c). *Iterative construction of complete Lyapunov functions*. In Proceedings of the 8th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH). Porto, Portugal.
- Argáez, C., Giesl, P., and Hafstein, S. (2019a). *Clustering algorithm for generalized recurrences using complete Lyapunov functions*. In International Conference on Informatics in Control, ICINCO. SUBMITTED.
- Argáez, C., Giesl, P., and Hafstein, S. (2019b). *Improved estimation of the chain-recurrent set*. In ECC 2019, Naples.
- Argáez, C., Giesl, P., and Hafstein, S. (2019c). *Iterative construction of complete lyapunov functions: Analysis of algorithm efficiency*. In Springer. SUBMITTED.
- Bernhard, P. and Suhr, S. (2018). *Lyapounov functions of closed cone fields: From Conley theory to time functions*. Commun. Math. Phys., 359:467–498.
- Conley, C. (1978). *Isolated Invariant Sets and the Morse Index*. CBMS Regional Conference Series no. 38. American Mathematical Society.
- Conley, C. (1988). *The gradient structure of a flow I*. Ergodic Theory Dynam. Systems, 8:11–26.
- Dellnitz, M. and Junge, O. (2002). *Set oriented numerical methods for dynamical systems*. In Handbook of dynamical systems, Vol. 2, pages 221–264. North-Holland, Amsterdam.
- Fathi, A. and Pageault, P. (2019). *Smoothing Lyapunov functions*. Trans. Amer. Math. Soc., 371:1677–1700.
- Giesl, P. (2007). *Construction of Global Lyapunov Functions Using Radial Basis Functions*. Lecture Notes in Math. 1904, Springer.
- Giesl, P., Argáez, C., Hafstein, S., and Wendland, H. (2018). *Construction of a complete Lyapunov function using quadratic programming*. In Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics (ICINCO). SIMULTECH 2018, Porto.

- Hsu, C. S. (1987). *Cell-to-cell mapping, volume 64 of Applied Mathematical Sciences*. Springer-Verlag, New York.
- Hurley, M. (1995). *Chain recurrence, semiflows, and gradients*. J Dyn Diff Equat, 7(3):437–456.
- Hurley, M. (1998). *Lyapunov functions and attractors in arbitrary metric spaces*. Proc. Amer. Math. Soc., 126:245–256.
- Iske, A. (1998). *Perfect centre placement for radial basis function methods*. Technical Report TUM-M9809, TU Munich, Germany.
- Krauskopf, B., Osinga, H., Doedel, E. J., Henderson, M., Guckenheimer, J., Vladimírsky, A., Dellnitz, M., and Junge, O. (2005). *A survey of methods for computing (un)stable manifolds of vector fields*. Internat. J. Bifur. Chaos Appl. Sci. Engrg., 15(3):763–791.
- Lyapunov, A. M. (1992). *The general problem of the stability of motion*. Internat. J. Control, 55(3):521–790. Translated by A. T. Fuller from Édouard Davaux's French translation (1907) of the 1892 Russian original.
- Wendland, H. (1998). *Error estimates for interpolation by compactly supported Radial Basis Functions of minimal degree*. J. Approx. Theory, 93:258–272.

