Statistical analysis of an iterative algorithm class for dynamical systems

Carlos Argáez[1], Peter Giesl[2], and Sigurdur Freyr Hafstein[3]

[1] Science Institute, University of Iceland, Tæknigarður, Dunhagi 5, 107 Reykjavík, Iceland,
carlos@hi.is
[2] Department of Mathematics, University of Sussex, Falmer, BN1 9QH, UK,
p.a.giesl@sussex.ac.uk
[3] Science Institute, University of Iceland, Tæknigarður, Dunhagi 5, 107 Reykjavík, Iceland,
shafstein.is

Abstract. We present some statistical analyses of algorithms to compute complete Lyapunov functions for dynamical systems. The algorithms approximate a solution to a PDE using mesh-free collocation with radial basis functions over a collocation grid. The equation has no solution in the chain-recurrent set and that allows us to determine the location of the chain-recurrent set. To evaluate our approximation we use two types of evaluation grids: either we place evaluation points in a circle (or sphere) around each collocation point, or we place them in the direction of the flow. Further, in an iterative algorithm we use the average of the orbital derivative on the evaluation grids to re-iterate. In this paper we deal with two open questions: which of the two types of evaluation grids gives better results? Is the average the adequate quantity or is the median more effective?

Keywords: Dynamical systems, Complete Lyapunov functions, Chain-recurrent set, Radial basis functions, Iterative methods, Computational mathematics

1 Introduction

Dynamical systems are of fundamental importance in science and engineering because they offer a compact language to describe physical and other phenomena. In recent years, novel algorithms to construct complete Lyapunov functions describing the qualitative behaviour of dynamical systems have been introduced and improved [2,4,5,6,11]. The importance of Lyapunov functions lies in their ability to locate the systems' basins of attraction. They were introduced in 1893 by Lyapunov [16] to study the stability of equilibria of dynamical systems.

Let us consider

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

where $\mathbf{x} \in \mathbb{R}^n$, $n \in \mathbb{N}$. A Lyapunov function for the system (1) 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 equilibrium or a periodic orbit.

A Lyapunov function is only defined in the neighbourhood of one attractor. A natural extension is a function defined on the whole state space, a *complete Lyapunov function* (CLF), see [7,8,10,13,14]. A CLF allows for dividing the state-space into two disjoint areas: The area of gradient-like flow, where the solution trajectories flow through, and the chain-recurrent set, where infinitesimal perturbations can make the system recurrent. On these two areas, the system behaves in fundamentally different ways. On the gradient-like part, a CLF V is strictly decreasing along solutions, whereas it is constant on the chain-recurrent set. For a sufficiently smooth CLF the orbital derivative $V'(\mathbf{x}) = \nabla V(\mathbf{x}) \cdot \mathbf{f}(\mathbf{x})$, i.e. the derivative along solutions, in these two areas is strictly negative or zero, respectively.

Although it is very difficult to construct Lyapunov functions for non-linear dynamical systems, there has been considerable progress in the last decades, see the review [12]. While there are several different approaches to computationally study dynamical systems, c.f. [9,18,1,19], in this paper we are mainly concerned with the methods from [2,4,5,6,11]. They allow for the computation of a (complete) Lyapunov function for any system described by an autonomous ordinary differential equation of the form (1).

Inspired by a method to compute classical Lyapunov functions for one stable equilibrium using Radial Basis Functions (RBFs), c.f. [11], the authors of this paper have developed a computationally efficient method to describe dynamical systems through the construction of a CLF and the subsequent localisation of the chain-recurrent set. The general idea is to find a CLF with $v'(\mathbf{x}) \leq 0$ by approximating 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. A function v is computed using RBFs, a mesh-free collocation technique, such that $v'(\mathbf{x}) = -1$ is fulfilled at all points $\mathbf{x}$ in a finite set X of collocation points.

The discretised problem of computing v is well-posed and possesses a unique solution. However, the computed function v cannot fulfill 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 of our general algorithms to locate the chain-recurrent sets; to determine the chain-recurrent set, we use the area where $v'(\mathbf{x}) \approx 0$, c.f. [2,4,5].

In Section 2, we give more details about the algorithm to construct a CLF and to determine the chain-recurrent set, including an iterative method. Section 3 applies the method to an example and performs a detailed analysis with respect to two different evaluation grids used in the literature: we analyse the distribution of values in the evaluation grid and compare them using the median and the average in the iteration. We conclude in Section 4.

2 Description of the algorithm

We will present the methodology introduced in [5], which is included in the freely distributed software LyapXool, see [6]. We first substitute the dynamical system (1) by

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

with a small parameter $\delta > 0$ and where $\|\cdot\|$ denotes the Euclidean norm. Systems (1) and (2) have the same trajectories, but the speed of the solutions to (2) is nearly uniform. This was shown to deliver superior results, see [5].

2.1 Mesh-free Collocation

Mesh-free collocation methods, in particular with RBFs [11], are a powerful methodology for solving generalised interpolation problems. RBFs are certain real-valued functions, whose evaluation depends only on the distance from the origin; Gaussians, multiquadrics and Wendland functions are examples of such functions.

Mesh-free collocation enables us to locally use a high resolution of collocation points to solve PDEs. In our work we use the compactly supported Wendland functions $\psi_{l,k}$ as RBFs, see [17]. They are positive definite functions that are polynomials on their compact support. We consider the RBF $\psi(\mathbf{x}) := \psi_{l,k}(\|\mathbf{x}\|)$, where $k \in \mathbb{N}$ is a smoothness parameter and l is fixed as $l = \lfloor \frac{n}{2} \rfloor + k + 1$. Our code uses the C++ tool from [3] to compute the Wendland functions. The Reproducing Kernel Hilbert Space (RKHS) corresponding to ψ contains the same functions as the Sobolev space $W_2^{k+(n+1)/2}(\mathbb{R}^n)$ and the spaces are norm equivalent.

Mesh-free collocation uses a finite set of collocation points $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \subset \mathbb{R}^n$, where the PDE is satisfied. We choose X as a subset of the following hexagonal grid with fineness-parameter $\alpha_{\text{Hexa-basis}} \in \mathbb{R}^+$:

$$\left\{ \alpha_{\text{Hexa-basis}} \sum_{k=1}^{n} i_k \omega_k : i_k \in \mathbb{Z} \right\}, \text{ where } \omega_1 = (2\varepsilon_1, 0, 0, \ldots, 0), \omega_2 = (\varepsilon_1, 3\varepsilon_2, 0, \ldots, 0), \tag{3}$$

$$\ldots,\ \omega_n = (\varepsilon_1, \varepsilon_2, \varepsilon_3, \ldots, (n+1)\varepsilon_n) \text{ and } \varepsilon_k = \sqrt{\frac{1}{2k(k+1)}}, \quad k \in \mathbb{N}.$$

This grid has been shown to optimally balance the opposing aims of a small fill distance, i.e. good error estimates, and a large separation distance of collocation points, i.e. small condition numbers of the collocation matrices, see [15]. All equilibria, i.e. points $\mathbf{x}$ with $\mathbf{f}(\mathbf{x}) = \mathbf{0}$, need to be removed from the set of the collocation points X, since otherwise the collocation matrix would be singular.

The approximated v is the norm-minimal function in the corresponding RKHS that satisfies the PDE $v'(\mathbf{x}) = -1$ at all collocation points. Practically, we find v by solving a system of N linear equations, where N is the number of collocation points.

To evaluate the computed CLF, several different evaluation grids have been proposed; in this paper we will focus on two different choices, see Figure 1. The first one distributes points homogeneously in two circumferences of radius $r > 0$ and $r/2$, see [5].

$$\begin{aligned} Y_{\mathbf{x}_j} &= \{\mathbf{x}_j + r\alpha_{\text{Hexa-basis}}(\cos(\theta), \sin(\theta))\} \ \cup \{\mathbf{x}_j + \tfrac{r}{2}\alpha_{\text{Hexa-basis}}(\cos(\theta), \sin(\theta))\} \\ &\text{where } \theta \in \{\pi/16, 2\pi/16, 3\pi/16, \ldots, 2\pi\}. \end{aligned} \tag{4}$$

The grid (4) can be generalised to higher dimensions. However, the growth of the evaluation points cardinality is exponential in the dimension. To avoid that, we use evaluation

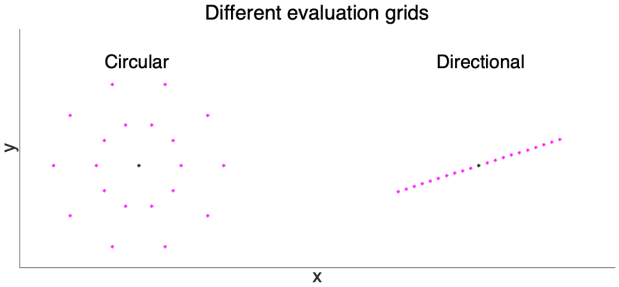

Fig. 1. Circular evaluation grid points around a collocation point (black) according to (4) and the directional evaluation grid points (5) around another collocation point (black) .

points that are aligned along the flow of the ODE system at the collocation point in [4], see (5). Here, $r \in (0,1)$ determines how far the evaluation points will be placed and $m \in \mathbb{N}$ denotes the number of points on each side of the collocation points.

$$Y_{\mathbf{x}_j} = \left\{ \mathbf{x}_j \pm \frac{r \cdot k \cdot \alpha_{\text{Hexa-basis}} \cdot \hat{\mathbf{f}}(\mathbf{x}_j)}{m} : k \in \{1, \ldots, m\} \right\} \tag{5}$$

2.2 Algorithm

Since the PDE $V'(\mathbf{x}) = -1$ does not have a solution for all $\mathbf{x}$ in the chain-recurrent set, we use an iterative algorithm to update the right-hand side in each step, using the results of the previous approximation; note that we only need to prescribe the value $V'(\mathbf{x}_j)$ at each collocation point $\mathbf{x}_j$. Our general algorithm to compute CLFs, given a set of collocation points X, is:

1. Compute the approximate solution v_0 to $V'(\mathbf{x}) = -1$; set $i = 0$
2. To approximate the chain-recurrent set by X^0, for each collocation point $\mathbf{x}_j$, compute $v_i'(\mathbf{y})$ for all $\mathbf{y} \in Y_{\mathbf{x}_j}$, see (4) or (5). If $v_i'(\mathbf{y}) > \gamma$ for an $\mathbf{y} \in Y_{\mathbf{x}_j}$, then $\mathbf{x}_j \in X^0$, else $\mathbf{x}_j \in X^-$, where $\gamma \leq 0$ is a chosen critical value
3. Define $\tilde{r}_j = \left(\frac{1}{|Y_{\mathbf{x}_j}|} \sum_{\mathbf{y} \in Y_{\mathbf{x}_j}} v_i'(\mathbf{y}) \right)_-$, where $x_- = x$ for $x \leq 0$ and $x_- = 0$ for $x > 0$
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 $V'(\mathbf{x}_j) = r_j$
6. Set $i \to i+1$ and repeat Steps 2. to 5. until some predefined criterion is reached

The aim of this paper is to investigate two aspects of the algorithm. Firstly, we investigate if using the average or the median in 3. is more appropriate for reiterating. Secondly, we compare the results using the circular or the directional evaluation grid. To enable a fair comparison, we use the same number of evaluation points in both grids.

3 Results

Le us consider system (1) with the right-hand side

$$\mathbf{f}(x,y) = \begin{pmatrix} -x(x^2+y^2-1/4)(x^2+y^2-1)-y \\ -y(x^2+y^2-1/4)(x^2+y^2-1)+x \end{pmatrix}. \tag{6}$$

This system has an asymptotically stable equilibrium at the origin and 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 $\Omega_2 = \{(x,y) \in \mathbb{R}^2 \mid x^2+y^2 = 1/4\}$.

The Wendland function used for computing a CLF to system (6) is $\psi_{5,3}$ with $c = 1$. The collocation points were set in the region $[-1.5,1.5] \times [-1.5,1.5] \subset \mathbb{R}^2$ and we used a hexagonal grid (3) with $\alpha_{\mathrm{H}exa-basis} = 0.09164$. We used both the circular grid with 10 different angles for two concentric circumferences and the directional grid (5) with parameters $m = 10$ and $r = 0.5$, resulting in 20 evaluation points per collocation point in both cases. We used $\delta^2 = 10^{-8}$, see (2), and the critical value $\gamma = -0.25$ in Step 2. of the Algorithm. For a plot of this CLF see [2].

3.1 Average and median

For the average and the median, we use the classical definitions. The average is defined and used in the algorithm in Section 2.2. When using the median, the algorithm in Section 2.2 is modified accordingly. The median of numbers, assembled in a vector R in ascending order is defined as $\mathrm{median}(R) = \frac{1}{2}\left(R_{\lfloor \frac{(\#R+1)}{2} \rfloor} + R_{\lceil \frac{(\#R+1)}{2} \rceil}\right)$. The median thus takes into account only the middle number(s).

3.2 Distribution

Let us consider the distribution of the values of the CLF's orbital derivative for some random collocation points in iteration 0, using a circular evaluation grid, see Figure 2. As none of these is Gaussian, the average value of the orbital derivative in Step 3. is not necessarily the optimal choice for a new value on the right-hand side in Step 5. in the algorithm for the next iteration.

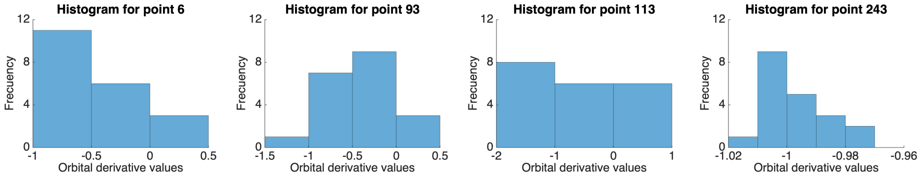

Fig. 2. Orbital derivative distribution around collocation points: circular evaluation grid. The points were chosen randomly. None of the distributions is Gaussian.

We will now compare the distributions of the orbital derivatives when using circular and directional evaluation grids. In contrast to Figure 2, we will now display the precise spatial distribution. We have chosen one random point in the chain-recurrent set (first two figures), and one random point, where the flow is gradient-like (last two figures), see Figure 3.

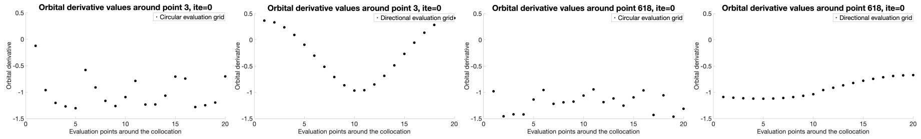

Fig. 3. Distribution of the orbital derivative around two collocation points in iteration 0. First: collocation point in chain-recurrent set, circular grid, second: collocation point in chain-recurrent set, directional grid, third: collocation point in gradient-like flow, circular grid, fourth: collocation point in gradient-like flow, directional grid.

For the chain-recurrent point with the circular grid (first figure), the values are random and vary between -1.3 and -0.1; the values for the gradient-like point (third figure) are negative between -1.5 and -1. For the same collocation points, the results using the directional evaluation grid are very different. For the chain-recurrent point (second figure), the orbital derivative is clearly forced down to -1 at the collocation point, while being positive far away with maximum at 0.4. The values have a distinct pattern with minimum at the collocation point with value -1. Since the integral of the orbital derivative over the whole periodic orbit must be zero and we force it to be negative on some points, it has to be positive elsewhere. For the gradient-like point (fourth figure), the orbital derivative is between -1.1 and -0.7.

While for points where the flow is gradient-like the difference between the circular and directional evaluation grids is not significant, it becomes clear that for points in the chain-recurrent set, the directional grid contains much more relevant information. The circular grid diffuses the information by considering many evaluation points out of the chain-recurrent set. The directional grid, however, only contains points which are of the same type as the collocation point (chain-recurrent/gradient-flow) and thus amplifies and clarifies the information: for points with gradient-like flow, the orbital derivative is close to -1 under the directional grid, while for points in the chain-recurrent set it displays a characteristic shape with a minimum at -1 and maxima at 0.4 for those points away from the collocation point.

3.3 Iterations

Now we consider iterations: the goal of these iterations is to determine the right-hand side of the PDE used to construct the CLF. In particular, the right-hand side should become 0 for points in the chain-recurrent set, while it should stay strictly negative in the gradient-like flow. We use either the circular or the directional evaluation grids combined with either the average or the median. The results for iteration 4 are shown in Figure 4 for the circular (upper row) and the directional (bottom row) evaluation grids.

Let us first consider a point in the chain-recurrent set (first column). While the circular evaluation grid (up) displays a large variation of values between -1.3 and -0.2

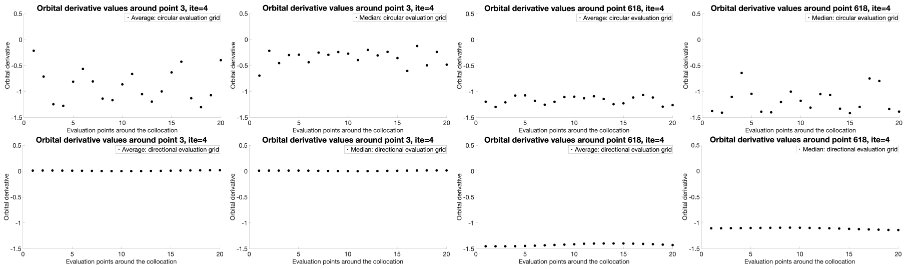

Fig. 4. Orbital derivative distribution around collocation points in iteration 4. Top row: circular, bottom row: directional evaluation grids. Column 1: chain-recurrent point, average, column 2: chain-recurrent point, median, column 3: gradient-like point, average, column 4: gradient-like point, median.

for the average values (column 1), and between -0.7 and -0.1 for the median values (column 2), the directional grid has a constant orbital derivative close to 0 for both average and median (bottom, columns 1 and 2) with values between 0 and 0.02 in both cases. The directional grid is thus better than the circular one; compared to iteration 0, the iterations have succeeded in identifying the correct right-hand side for the directional grid. When using the circular evaluation grid, the median gives better results than the average, as it has values closer to 0.

Let us now consider a point with gradient-like flow (columns 3 and 4). Again, the circular evaluation grid displays a larger variation of values, namely between -1.3 and -1 for the average values (up, column 3), and between -1.4 and -0.6 for the median value (up, column 4). The directional grid has a nearly constant orbital derivative at -1.4 for the average (bottom, column 3) and -1.1 for the median (bottom, column 4). The directional grid is again superior to the circular one; in this case the median results in a value closer to the original derivative at -1. For the circular evaluation grid, the average gives better results than the median as there is less variation; however, also the median delivers only negative values, i.e. classifying the collocation point correctly as gradient-like flow.

3.4 Determination of the chain-recurrent set

Let us now investigate how the different methods manage to identify the chain-recurrent set after iterations 4 and 49. Figure 5 displays the approximation to the chain-recurrent set by plotting the collocation points $\mathbf{x}_j \in X^0$ (see Step 2. of the algorithm), i.e. $\mathbf{x}_j$ where at least one point in corresponding evaluation grid $Y_{\mathbf{x}_j}$ satisfies $v'(\mathbf{x}) > \gamma$.

The results in Figure 5 use the circular grid in the upper row. The figures in column 1 and 2 display the approximation to the chain-recurrent set in iterations 4 and 49, respectively, using a circular evaluation grid and the average. The two periodic orbits and the equilibrium at the origin are clear and completely displayed. However, they are over-approximated. In contrast, the figures in columns 3 and 4 showing the approximation to the chain-recurrent set in iterations 4 and 49, respectively, for the median and a circular grid, display gaps in the periodic orbits in iteration 49.

We continue with the directional grid in the bottom row. The figures in column 1 and 2, bottom row, display the approximation to the chain-recurrent set in iterations 4 and 49, respectively, using a directional evaluation grid and the average. The two periodic orbits and the equilibrium at the origin are clearly and completely displayed, however, they are slightly over-approximated in iteration 49. The figures in columns 3 and 4, using a directional evaluation grid and the median, show a very similar picture with slightly less over-approximation.

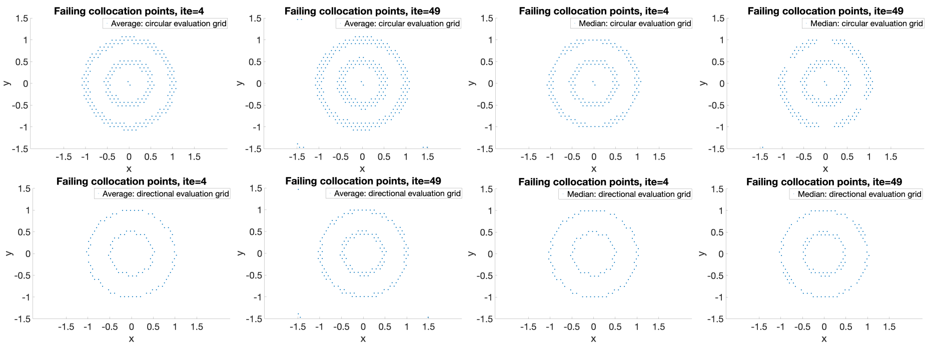

Fig. 5. Comparison of the chain-recurrent set in iterations 4 and 49. Top row: circular, bottom row: directional evaluation grids. Column 1: iteration 4, average, column 2: iteration 49, average, column 3: iteration 4, median, column 4: iteration 49, median.

3.5 Values on the chain-recurrent set

Let us analyse the values of the orbital derivative on the circular periodic orbit with radius 1. In particular, we study their dependence on the circular and directional grids as well as using the average or the median for the iterations, see Figure 6. The top figure shows the orbital derivative after iteration 0 on the orbit. It is plotted as a function of the angle on the positive x-axis, i.e. θ corresponds to the point $(\cos\theta, \sin\theta)$ on the orbit. After iteration 0 neither the evaluation grid nor the selection of average or median, matters. The values oscillate heavily between -1 and 0.7.

The remaining four figures show the orbital derivative after iteration 49 on the orbit. In the left column we used the circular evaluation grid and in the right column the directional evaluation grid. In both cases we iterated using the average in the middle row and using the median in the bottom row. In all cases we would like to achieve values of the orbital derivative close to zero. We clearly see that iterating using the average (middle row) is better as the orbital derivative is closer to zero. Further, the values oscillate much more if we use the circular evaluation grid. Indeed, if we use the directional evaluation grid and the average for iterations most of the values are between -0.02 and 0, the only exceptions being two small spikes close to the angles 0 and π.

When using the circular evaluation grid and the median, we observe that the values oscillate considerably, and there are large areas, where the orbital derivative is entirely

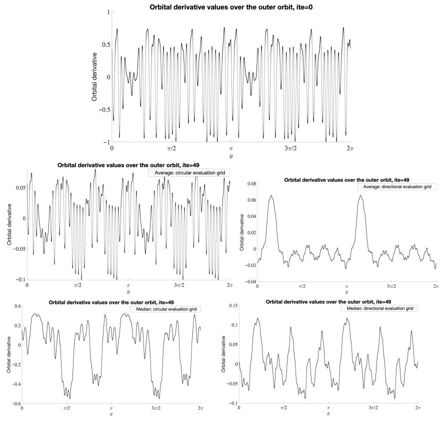

Fig. 6. Orbital derivative on the circular periodic orbit with radius 1. Top figure: iteration 0. All other figures: iteration 49. Left column: circular, right column: directional evaluation grid. Middle row: average, bottom row: median.

negative and below the threshold γ; hence, they do not appear as failing in Figure 5. Let us discuss why this can develop over the iterations: the circular grid takes adjacent points in the gradient-flow into account, which have negative orbital derivative, while the points of the directional evaluation grid lie on the periodic orbit with derivatives close to 0. While the average over the circular evaluation grid takes some positive values on the periodic orbit into account, the median just considers the middle number(s).

4 Conclusion

We have investigated the differences between using a circular or directional evaluation grid as well as those between employing the median or the average to the iterative process. We have found that the values of the orbital derivative around a collocation point have no clear distribution; in particular, they are not Gaussian. Nevertheless, using the average provides better results in the iteration process for both types of evaluation grids. Further, the directional evaluation grid proved to be superior. This is presumably because the directional evaluation grid only considers points along the flow, which are of the same type as the collocation point; if the collocation point is on (or very close to) the chain-recurrent set, then so are the evaluation points. This improves the determination of the chain-recurrent set.

Summarising, using the directional evaluation grid with the average for iterations delivers the best results in localising the chain-recurrent set. For collocation points far from the chain-recurrent set, the graph is quite flat and its curvature will be (close to) zero, while it is non-zero for collocation points close to the chain-recurrent set. We anticipate that this criterion might perform better than the current one, namely that the orbital derivative is above a critical value.

References

1. Ao, P. (2004). *Potential in stochastic differential equations: novel construction. J. Phys. A Math. Theor.*, 37, 3, L25–L30.
2. Argáez, C., Giesl, P., and Hafstein, S. (2017a). *Analysing dynamical systems: Towards computing complete Lyapunov functions.* In *Proceedings of the 7th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH)*, 134–144. Madrid, Spain.
3. Argáez, C., Giesl, P., and Hafstein, S. (2017b). *Wendland functions: A C++ code to compute them.* In *Proceedings of the 7th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH)*, 323–330. Madrid, Spain.
4. Argáez, C., Giesl, P., and Hafstein, S. (2018a). *Computation of complete Lyapunov functions for three-dimensional systems.* In *Proceedings of the 57th IEEE Conference on Decision and Control (CDC)*, 4059–4064. Miami Beach, FL, USA.
5. 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, 1–11.
6. Argáez, C., Berthet, J.C., Björnsson, H., Giesl, P., and Hafstein, S.F. (2019). *LyapXool - a program to compute complete Lyapunov functions. SoftwareX*, 10, 100325.
7. Auslander, J. (1964). *Generalized recurrence in dynamical systems. Contr. to Diff. Equ.*, 3, 65–74.
8. Conley, C. (1978). *Isolated Invariant Sets and the Morse Index*. CBMS Regional Conference Series no. 38. American Mathematical Society.
9. Dellnitz, M. Froyland, G. Junge, O. (2001). *The algorithms behind GAIO - Set oriented numerical methods for dynamical systems*, in *Ergodic theory, analysis, and efficient simulation of dynamical systems. Springer. B. Fiedler (ed.)* 145–174.
10. Conley, C. (1988). *The gradient structure of a flow I. Ergodic Theory Dynam. Systems*, 8, 11–26.
11. Giesl, P. (2007). *Construction of Global Lyapunov Functions Using Radial Basis Functions*. Lecture Notes in Math. 1904, Springer.
12. Giesl, P. and Hafstein, S. (2015). *Review of computational methods for Lyapunov functions. Discrete Contin. Dyn. Syst. Ser. B*, 20(8), 2291–2331.
13. Hurley, M. (1995). *Chain recurrence, semiflows, and gradients. J. Dyn. Diff. Equat.*, 7(3), 437–456.
14. Hurley, M. (1998). *Lyapunov functions and attractors in arbitrary metric spaces. Proc. Amer. Math. Soc.*, 126, 245–256.
15. Iske, A. (1998). *Perfect centre placement for radial basis function methods.* Technical Report TUM-M9809, TU Munich, Germany.
16. 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.
17. Wendland, H. (1998). *Error estimates for interpolation by compactly supported Radial Basis Functions of minimal degree. J. Approx. Theory*, 93, 258–272.
18. Yuan, R., Wang, X., Ma, Y., Yuan, B., and Ao, P. (2013). *Exploring a noisy van der Pol type oscillator with a stochastic approach. Phys. Rev. E.*, 87, 6, 062109.
19. Zhu, X.-M. , Yin, L., and Ao, P. (2006) *Limit cycle and conserved dynamics Int. J. Mod. Phys. B*, 20, 7, 817–827.