Dolomites Research Notes on Approximation

Proceedings of *Multivariate Approximation: Theory, Algorithms & Applications 2017*, Volume 11 · 2018 · Pages 15–22

Meshless partition of unity method for attraction basins of periodic orbits: Fast detection of separatrix points

Alessandra De Rossi[a] · Emma Perracchione[b] · Ezio Venturino[a]

Abstract

Given a dynamical system, the problem of reconstructing the attraction basins can effectively be solved by means of meshfree interpolation tools. We propose a novel approach to calculate separatrix manifolds by detecting points via an efficient bisection-like scheme. Such data are then interpolated by means of the implicit Partition of Unity (PU) method. The detection algorithm works for dynamical systems of two equations, speeds up the computational time of the previously studied routines and allows the treatment of periodic orbits.

1 Introduction

Over the last years numerical approximation of multivariate data has gained popularity in various disciplines, such as numerical solution of Partial Differential Equations (PDEs), image registration, neural networks, optimization, statistics, finance and modeling of 3D objects (see e.g. [4, 11, 24, 25, 26]). In this paper, we analyze an application in the field of population dynamics [9, 15].

The importance of an investigation in this context follows from the fact that mathematical modeling is still a fruitful research area despite its ancient origins. Specifically, in 1798 T.R. Malthus proposed the first model for the dynamics of a population [22]. Anyway, in this discipline the true breakthrough is due to V. Volterra and A.J. Lotka who in the 1920s, independently of each other, studied the dynamics of interacting populations [21, 27].

Nowadays, mathematical modeling is commonly applied to major disciplines, such as biology, medicine and social sciences. More precisely, the prediction of the temporal evolution of the considered quantities is determined via systems of ODEs. For comprehensive treatments about the theory of mathematical modeling see e.g. [3, 23].

In an initial value problem involving a set of ODEs, a particular solution of the system is completely determined by the Initial Condition (IC). Depending on the initial state of the system and on conditions involving the model parameters, the trajectories may tend toward different asymptotically stable equilibria or periodic orbits. In this case, the phase state of the dynamical system is divided by separatrix manifolds. Such manifolds, defining the domains of attraction, are determined locally (by linearization) in well-known cases (see e.g. [23]). Furthermore, numerical tools based on characterizing a Lyapunov function as a solution of a suitable linear first order PDE have already been developed. The PDE is then approximated using meshless collocation methods [17].

The detection of the points lying on the separatrix manifolds can be performed by means of an efficient backward integration in time (see [14]) or by using a bisection routine [6]. Focusing on the latter, in [9] a scheme to detect the separatrix points via a bisection routine is built to give graphical representations of the separatrix manifolds for asymptotically stable equilibria. Then the points are interpolated with the implicit PU method [13, 28], since attraction basins can be described by implicit equations. However, the detection approach constructed in this way is computationally expensive and is not robust enough to work with periodic orbits. Here, for 2-dimensional dynamical systems, we construct an *ad hoc* bisection-like algorithm that is able to speed up the procedure. Further, we extend the scheme so that it can accommodate also the case of periodic orbits.

The outline of the paper is as follows. Section 2 is devoted to the presentation of the designed algorithms for the detection of the points lying on the separatrix curves. In Section 3 we present the meshless interpolation scheme. The complexity analysis of the method is presented in Section 4. Finally, Section 5 contains several numerical results.

2 Fast detection of separatrix points for periodic orbits

In this section, given two different steady states, we are interested in finding the manifold separating the ICs evolving toward the two different stationary states. The case of an asymptotically stable equilibrium has already been studied in [9], while we now focus on systems that for a given IC may also stabilize toward periodic orbits. Moreover, this novel procedure is able to considerably reduce the computational cost of the algorithm adopted in [9].

[a] Dipartimento di Matematica "G. Peano", Università di Torino, Italia
[b] Dipartimento di Matematica "Tullio Levi-Civita", Università di Padova, Italia

2.1 The criteria for periodic orbits and asymptotically stable equilibria

Let

$$\frac{d\boldsymbol{Y}(t)}{dt} = \boldsymbol{g}(\boldsymbol{Y}(t), t), \tag{1}$$

be a 2-dimensional dynamical system. If (1) admits more than one steady state, the final configuration of the system depends on the IC $\boldsymbol{Y}(0) = \boldsymbol{Y}_0$. Thus, we now need a criterion to check where trajectories ultimately stabilize.

Let $\boldsymbol{E}_0 = E_0^k$, $k = 1, 2$, be an asymptotically stable equilibrium point or a centre of a periodic orbit. A criterion that allows to check where a trajectory finally stabilizes needs to be discussed. Let us first suppose that $\boldsymbol{E}_0$ is an asymptotically stable equilibrium point. In that case, we only need to check if

$$\lim_{t\to\infty} Y_k(t) = E_0^k, \quad k = 1, 2.$$

Of course in finite arithmetic, the system is integrated for a time interval $[0, t_M]$ and therefore we obtain a set of discrete values $y_k(t_i)$, $k = 1, 2$, $i = 1, \ldots, M$, as approximated solution. Then, if $\boldsymbol{E}_0$ is a stable equilibrium point, we only need to check if

$$|y_k(t_M) - E_0^k| \leq \tau_k, \quad k = 1, 2,$$

where τ_k are prescribed tolerances. To apply this criterion, we assume that t_M is large enough so that the trajectory stabilizes for an integration time t^*, with $t^* \leq t_M$. We are not able to provide suitable values of t_M for any dynamical system. However, for the given system and integration time, the proposed algorithm checks if t_M is effective for applying the criterion (see [9] for further details).

However, since in case of periodic orbits a trajectory does not stabilize toward any equilibrium point, this simple criterion cannot be used to test if a given IC generates a periodic trajectory. In that case, we should define a different check test. Let us suppose that an IC $\boldsymbol{Y}_0$ generates a trajectory in the phase state that follows a periodic orbit. In general, neither the period nor the shape of the orbit are a priori known. However, thinking of the phase state, we can overcome this problem and state that the path originating at $\boldsymbol{Y}_0$ follows a periodic orbit around $\boldsymbol{E}_0$ if

$$\min_{t_i, i=M,\ldots,\tilde{M}} ||\boldsymbol{y}(t_i)||_2 < ||\boldsymbol{E}_0||_2 < \max_{t_i, i=M,\ldots,\tilde{M}} ||\boldsymbol{y}(t_i)||_2,$$

with $t_{\tilde{M}} = nt_M$, where the integer n is such that $n \geq 2$. Indeed, in order to apply this criterion, after integrating the system in $[0, t_M]$ with IC $\boldsymbol{Y}_0$, we need to use a time interval $[t_M, nt_M]$ and IC $\boldsymbol{Y}(t_M) = \boldsymbol{y}(t_M)$. This is due to the fact that we require that the trajectory at time t_M already reached the stationary state and that the time interval is large enough to test if the trajectory made a periodic cycle. In the sequel, for brevity, we use the notation $\boldsymbol{Y}_0 \to \boldsymbol{E}_0$, meaning that the path originating at $\boldsymbol{Y}_0$ follows a periodic orbit around $\boldsymbol{E}_0$ or asymptotically tends to $\boldsymbol{E}_0$.

Once these criteria are fixed, we can focus on the algorithm that finds the manifold identifying the attraction basins, given two different steady states $\boldsymbol{E}_0$ and $\boldsymbol{E}_1$ (which can be centres of cyclic solutions or asymptotically stable equilibria). Thus, our aim consists in finding the points lying on the separatrix curve and in interpolating/approximating them with a numerical tool. As in [9], a separatrix point is found by performing a detection routine, which essentially consists of a bisection-like algorithm. To construct the separatrix points in [9], a set of equispaced ICs on the boundary of the square $[0, \gamma]^2$, with $\gamma \in \mathbb{R}^+$, is considered. Then, a detection routine is performed between each pair of opposite points, which are seen as ICs of the system. Since we consider opposite points, the bisection algorithm usually takes a large number of steps for converging to a fixed tolerance.

2.2 Fast procedure to determine separatrix points

Here, we propose an algorithm that enables us to take ICs *close* to the separatrix. To start the process we apply the detection routine presented in [9] to those vertices of the square $[0, \gamma]^2, \gamma \in \mathbb{R}^+$, that evolve toward two different steady states $\boldsymbol{E}_0$ and $\boldsymbol{E}_1$. The bisection routine is iterated until the distance between two consecutive approximations is less than ε, where ε is a fixed tolerance. To be more precise, the bisection algorithm is applied to

$$\boldsymbol{Y}_0^1 = (0,0),\ \boldsymbol{Y}_0^2 = (0,\gamma),\ \text{if } \boldsymbol{Y}_0^1 \to \boldsymbol{E}_0 \text{ and } \boldsymbol{Y}_0^2 \to \boldsymbol{E}_1 \text{ or viceversa},$$

$$\boldsymbol{Y}_0^1 = (0,0),\ \boldsymbol{Y}_0^3 = (\gamma,0),\ \text{if } \boldsymbol{Y}_0^1 \to \boldsymbol{E}_0 \text{ and } \boldsymbol{Y}_0^3 \to \boldsymbol{E}_1 \text{ or viceversa},$$

$$\boldsymbol{Y}_0^2 = (0,\gamma),\ \boldsymbol{Y}_0^4 = (\gamma,\gamma),\ \text{if } \boldsymbol{Y}_0^2 \to \boldsymbol{E}_0 \text{ and } \boldsymbol{Y}_0^4 \to \boldsymbol{E}_1 \text{ or viceversa},$$

$$\boldsymbol{Y}_0^3 = (\gamma,0),\ \boldsymbol{Y}_0^4 = (\gamma,\gamma),\ \text{if } \boldsymbol{Y}_0^3 \to \boldsymbol{E}_0 \text{ and } \boldsymbol{Y}_0^4 \to \boldsymbol{E}_1 \text{ or viceversa}.$$

Since the separatrix intersects the perimeter of the square at two points on two different edges, either consecutive or opposite, and since the above ICs span all the sizes of the square, we are able to find two separatrix points, namely $\boldsymbol{s}_1 = (s_1^1, s_1^2)$ and $\boldsymbol{s}_2 = (s_2^1, s_2^2)$. One of these, let us say $\boldsymbol{s}_1 = (s_1^1, s_1^2)$, chosen in an arbitrary way, is used as starting seed of an iterative process that enables us to efficiently find the other points lying on the separating curve.

Because of the choice of the ICs, one component of $\boldsymbol{s}_1$ is necessarily equal to 0 or to γ. In particular, if $s_1^2 = 0$, we define an increment $\Delta = \gamma/q$, $q \in \mathbb{N}^+$ and apply the bisection routine with either one of the following ICs

$$\boldsymbol{Y}_0^1 = (\min(s_1^1 + \Delta, \gamma), s_1^2),\ \boldsymbol{Y}_0^2 = (s_1^1, \min(s_1^2 + \Delta, \gamma)),\ \text{if } \boldsymbol{Y}_0^1 \to \boldsymbol{E}_0 \text{ and } \boldsymbol{Y}_0^2 \to \boldsymbol{E}_1 \text{ or viceversa},$$

or

$$\boldsymbol{Y}_0^2 = (s_1^1, \min(s_1^2 + \Delta, \gamma)),\ \boldsymbol{Y}_0^3 = (\max(s_1^1 - \Delta, 0), s_1^2),\ \text{if } \boldsymbol{Y}_0^2 \to \boldsymbol{E}_0 \text{ and } \boldsymbol{Y}_0^3 \to \boldsymbol{E}_1 \text{ or viceversa}.$$

Note that the separatrix can intersect only one of the two segments $\overline{\boldsymbol{Y}_0^1\boldsymbol{Y}_0^2}$ and $\overline{\boldsymbol{Y}_0^2\boldsymbol{Y}_0^3}$, so that only one of the above conditions is satisfied. In other words, with these ICs we find exactly one separatrix point $\boldsymbol{s}_2$ (refer to Figure 1 for an illustrative example).

The procedure can now be iterated as follows. Assume that $s_1^2 = 0$. In fact this is not restrictive because we can proceed similarly for $s_1^1 = 0$ or $s_1^1 = \gamma$ or $s_1^2 = \gamma$. In particular, we exclude the orthogonal directions that from $\boldsymbol{s}_2$ generate ICs which are not contained in $[0,\gamma]^2$.

In general, to find the k-th separatrix point we consider pairs of points that are close to $\boldsymbol{s}_{k-1}$. Precisely, assuming that $s_{k-1}^1 - s_{k-2}^1 \geq 0$ and $s_{k-1}^2 - s_{k-2}^2 \geq 0$, a bisection routine between the ICs that evolve toward different steady states, chosen among the following pairs:

$$\boldsymbol{Y}_0^1 = (\min(s_{k-1}^1 + \Delta, \gamma), s_{k-1}^2),\ \boldsymbol{Y}_0^2 = (s_{k-1}^1, \min(s_{k-1}^2 + \Delta, \gamma)),\ \text{if } \boldsymbol{Y}_0^1 \to \boldsymbol{E}_0 \text{ and } \boldsymbol{Y}_0^2 \to \boldsymbol{E}_1 \text{ or viceversa,}$$

or

$$\boldsymbol{Y}_0^1 = (\min(s_{k-1}^1 + \Delta, \gamma), s_{k-1}^2),\ \boldsymbol{Y}_0^3 = (s_{k-1}^1, \max(s_{k-1}^2 - \Delta, 0)),\ \text{if } \boldsymbol{Y}_0^1 \to \boldsymbol{E}_0 \text{ and } \boldsymbol{Y}_0^3 \to \boldsymbol{E}_1 \text{ or viceversa,}$$

or

$$\boldsymbol{Y}_0^2 = (s_{k-1}^1, \min(s_{k-1}^2 + \Delta, \gamma)),\ \boldsymbol{Y}_0^4 = (\max(s_{k-1}^1 - \Delta, 0), s_{k-1}^2),\ \text{if } \boldsymbol{Y}_0^2 \to \boldsymbol{E}_0 \text{ and } \boldsymbol{Y}_0^4 \to \boldsymbol{E}_1 \text{ or viceversa.}$$

This step produces the k-th separatrix point. Obviously, depending on $\boldsymbol{s}_{k-1}$ and $\boldsymbol{s}_{k-2}$, we can have different scenarios. For instance, if $s_{k-1}^1 - s_{k-2}^1 \geq 0$ and $s_{k-1}^2 - s_{k-2}^2 < 0$, we consider

$$\boldsymbol{Y}_0^1 = (\max(s_{k-1}^1 - \Delta, 0), s_{k-1}^2),\ \boldsymbol{Y}_0^2 = (s_{k-1}^1, \min(s_{k-1}^2 + \Delta, \gamma)),\ \text{if } \boldsymbol{Y}_0^1 \to \boldsymbol{E}_0 \text{ and } \boldsymbol{Y}_0^2 \to \boldsymbol{E}_1 \text{ or viceversa,}$$

or

$$\boldsymbol{Y}_0^1 = (\max(s_{k-1}^1 - \Delta, 0), s_{k-1}^2),\ \boldsymbol{Y}_0^3 = (s_{k-1}^1, \max(s_{k-1}^2 - \Delta, 0)),\ \text{if } \boldsymbol{Y}_0^1 \to \boldsymbol{E}_0 \text{ and } \boldsymbol{Y}_0^3 \to \boldsymbol{E}_1 \text{ or viceversa,}$$

or

$$\boldsymbol{Y}_0^3 = (s_{k-1}^1, \max(s_{k-1}^2 - \Delta, 0)),\ \boldsymbol{Y}_0^4 = (\min(s_{k-1}^1 + \Delta, \gamma), s_{k-1}^2),\ \text{if } \boldsymbol{Y}_0^3 \to \boldsymbol{E}_0 \text{ and } \boldsymbol{Y}_0^4 \to \boldsymbol{E}_1 \text{ or viceversa.}$$

Similarly, one can fix the other searching directions that depend on the previous two separatrix points.

Note that if the separatrix curve passes through more than one quadrant generated by the above ICs, we only need to reduce the stepsize Δ until such drawback is overcome. In the end, the method returns a set of separatrix points $\mathcal{S}_{q+2} = \{\boldsymbol{s}_i, i = 1,\dots,q+2\}$. Then, such points will be interpolated via the method described in the next section.

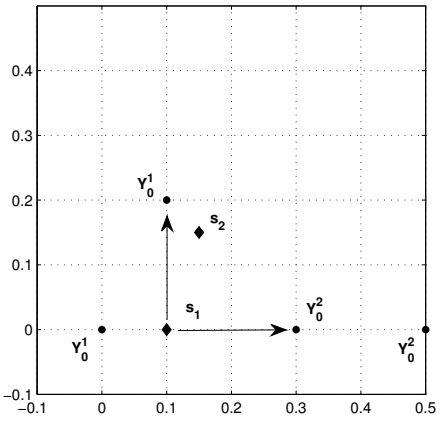

Figure 1: Illustrative example of the iterative process that enables us to find separatrix points. The ICs are plotted with dots and the diamonds identify the points on the separatrix. As first step, we apply the bisection with the vertices of the square. In this specific example the ICs $\boldsymbol{Y}_0^1 = (0,0)$ and $\boldsymbol{Y}_0^2 = (\gamma,0)$ provide us the point $\boldsymbol{s}_1$. Then, at the second step, we consider as new ICs $\boldsymbol{Y}_0^1 = (s_1^1 + \Delta, s_1^2)$ and $\boldsymbol{Y}_0^2 = (s_1^1, s_1^2 + \Delta)$. This because after integrating, we know that in this case the separatrix intersects the segment joining these points. Consequently, we discard the IC $\boldsymbol{Y}_0^3 = (\max(s_1^1 - \Delta, 0), s_1^2)$.

3 The reconstruction via implicit PU

In general, the basins of attraction might be described by implicit equations. Therefore the use of an implicit reconstruction scheme is essential. Moreover, if a large number of interpolation nodes is involved, the RBF system may suffer from ill-conditioning. To avoid this problem, local approximation schemes, such as the PU method can be used. In the sequel, we focus on reconstruction of implicit curves, but the proposed method can easily be extended to higher dimensions.

3.1 The implicit PU approach

The set of separatrix points $\mathcal{S}_{q+2} = \{\boldsymbol{s}_i, i = 1,\dots,q+2\} \subseteq [0,\gamma]^2$ is seen as a set belonging to an unknown one dimensional manifold $\mathscr{M}$, namely a curve in $\mathbb{R}^2$. Of course, in this case $\mathscr{M}$ identifies all points $\boldsymbol{x} \in \mathbb{R}^2$ satisfying the implicit equation $f(\boldsymbol{x}) = 0$, for some function f. Then, in order to reduce the problem to a standard explicit form, we add further interpolation conditions [13].

Let us suppose that, for each point $\boldsymbol{s}_i$, the oriented normal $\boldsymbol{n}_i \in \mathbb{R}^2$ is given. We construct extra points along the normals [20]. Precisely, we obtain for each data point $\boldsymbol{s}_i$ two additional points, namely $\boldsymbol{s}_{q+2+i} = \boldsymbol{s}_i + \omega \boldsymbol{n}_i$, and $\boldsymbol{s}_{2(q+2)+i} = \boldsymbol{s}_i - \omega \boldsymbol{n}_i$, where $\omega \in \mathbb{R}$ is the stepsize. Then, to each data we assign the function values $\mathcal{F}_{3(q+2)} = \{f(\boldsymbol{s}_i) \in \mathbb{R}, i = 1,\dots,3(q+2)\}$ as follows

$$\begin{aligned} f(\boldsymbol{s}_i) &= 0, && i = 1,\dots,q+2,\\ f(\boldsymbol{s}_i) &= 1, && i = q+3,\dots,2q+4,\\ f(\boldsymbol{s}_i) &= -1, && i = 2q+5,\dots,3q+6. \end{aligned}$$

For the augmented data set, namely $\mathcal{S}_{3(q+2)} = \{\boldsymbol{s}_i \in \mathbb{R}^2, i = 1,\dots,3(q+2)\} \subseteq \Omega$ we compute an interpolant $\mathcal{I}$ such that $\mathcal{I} = 0$ interpolates the given data. The approximant is constructed via the PU method, i.e. at first we build a covering of the domain Ω with d circular patches, also called subdomains, Ω_j, such that $\Omega \subseteq \cup_{j=1}^{d} \Omega_j$, with some mild overlap among the Ω_j's. Then, the interpolant assumes the form [13]

$$\mathcal{I}(\boldsymbol{x}) = \sum_{j=1}^{d} R_j(\boldsymbol{x}) W_j(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega, \tag{2}$$

where $\{W_j\}_{j=1}^{d}$ is a partition of unity, i.e. a family of compactly supported, non-negative, continuous functions W_j, with $\operatorname{supp}(W_j) \subseteq \Omega_j$ and R_j are local approximants. Here, we take RBF local interpolants and thus

$$R_j(\boldsymbol{x}) = \sum_{k=1}^{q_j} \alpha_k^{(j)} \phi(||\boldsymbol{x} - \boldsymbol{s}_k^{(j)}||_2), \tag{3}$$

where q_j indicates the number of points on Ω_j and $\boldsymbol{s}_k^{(j)} \in \mathcal{S}_{q_j} = \mathcal{S}_{3(q+2)} \cap \Omega_j$, with $k = 1,\dots,q_j$. The coefficients $\{\alpha_k^{(j)}\}_{k=1}^{q_j}$ in (3) are determined by enforcing the q_j local interpolation conditions

$$R_j(\boldsymbol{s}_i^{(j)}) = f_i^{(j)}, \quad i = 1,\dots,q_j.$$

Thus, in case of strictly positive definite functions, the problem of finding the PU interpolant (2) reduces to solving d linear systems of the form

$$A_j \boldsymbol{\alpha}_j = \boldsymbol{f}_j,$$

where $\boldsymbol{\alpha}_j = (\alpha_1^{(j)},\dots,\alpha_{q_j}^{(j)})^T$, $\boldsymbol{f}_j = (f_1^{(j)},\dots,f_{q_j}^{(j)})^T$ and $A_j \in \mathbb{R}^{q_j \times q_j}$ is

$$A_j = \begin{pmatrix} \phi(||\boldsymbol{s}_1^{(j)} - \boldsymbol{s}_1^{(j)}||_2) & \cdots & \phi(||\boldsymbol{s}_1^{(j)} - \boldsymbol{s}_{q_j}^{(j)}||_2) \\ \vdots & \ddots & \vdots \\ \phi(||\boldsymbol{s}_{q_j}^{(j)} - \boldsymbol{s}_1^{(j)}||_2) & \cdots & \phi(||\boldsymbol{s}_{q_j}^{(j)} - \boldsymbol{s}_{q_j}^{(j)}||_2) \end{pmatrix}.$$

3.2 Normals estimation

To implement the implicit PU method, for each point $\boldsymbol{s}_i$, we need to find the oriented normal $\boldsymbol{n}_i$. The normals can easily be estimated following [13]. Then, to orient them consistently, we adopt the technique presented in [20]. To this aim, we model this problem via graph optimization [20]. At first, we build the *Riemann graph* $G = \{\mathcal{V}, \mathcal{E}\}$, with each node in $\mathcal{V}$ corresponding to one of the data points. We remark that the Riemann graph is defined as the undirected graph among which there exists an edge e_{ik} in $\mathcal{E}$ if v_k is one of the K-nearest neighbors of v_i and viceversa. In our case, the graph G has a vertex for every normal $\boldsymbol{n}_i$ and an edge e_{ik} between the vertices of $\boldsymbol{n}_i$ and $\boldsymbol{n}_k$ if and only if $i \in \mathscr{K}(\boldsymbol{s}_k)$ or $k \in \mathscr{K}(\boldsymbol{s}_i)$, where $\mathscr{K}(\boldsymbol{s}_k)$ denotes the set of K-nearest neighbour points of $\boldsymbol{s}_k$. We assign to each edge e_{ik} the cost $w(e_{ik}) = 1 - |\boldsymbol{n}_i \boldsymbol{n}_k^T|$, as suggested in [20]. A favourable propagation order can therefore be achieved by traversing the *minimal spanning tree* of the Riemann graph. The advantage of this order consists in propagating the orientation along directions of low curvature in the data.

To such scope, we need some preliminary definitions (see e.g. [1]) for further details.

Definition 3.1. In any connected graph G, a spanning tree is a subgraph of G having the following two properties:

i. the subgraph is a tree,

ii. the subgraph contains every vertex of G.

Definition 3.2. The weight of a tree is defined to be the sum of the weights of all edges in the tree.

Definition 3.3. Given a connected weighted graph G the minimal spanning tree is the one having minimum weight among all spanning trees in the graph.

We now want to determine how to construct a minimum weight spanning tree. As suggested by [20], we use the Kruskal's algorithm (e.g. refer to [18] for further details). Precisely, we begin by choosing an edge of minimum weight in the graph and we then continue by selecting an edge of minimum weight from the remaining edges until a spanning tree is formed.

4 Computational complexity

In a standard setting, the efficiency of the bisection method trivially depends on the length of the initial searching interval, in our case $||\boldsymbol{Y}_0^1 - \boldsymbol{Y}_0^2||_2$. Denoting by $\boldsymbol{s}$, the j-th approximation of a separatrix point $\boldsymbol{s}^*$, we have that

$$||\boldsymbol{s} - \boldsymbol{s}^*||_2 \leq \frac{||\boldsymbol{Y}_0^1 - \boldsymbol{Y}_0^2||_2}{2^{j+1}}.$$

In particular, since the method previously explained is built ad hoc in order to start close to the separatrix, we expect a saving in terms of computational complexity with respect to the scheme outlined in [9]. This is confirmed by the fact that, in [9], $||\boldsymbol{Y}_0^1 - \boldsymbol{Y}_0^2||_2 = \gamma$, while currently, since $\Delta = \gamma/q$, we obtain

$$||\boldsymbol{s} - \boldsymbol{s}^*||_2 \leq \frac{||\boldsymbol{Y}_0^1 - \boldsymbol{Y}_0^2||_2}{2^{j+1}} = \frac{\sqrt{2}\gamma}{q2^{j+1}} \ll \frac{\gamma}{2^{j+1}}.$$

Concerning the interpolation method, given $q+2$ data and the space dimension 2 [13, 28], we need to:

i. Compute and consistently orient the normals to the curve. This consists in calculating the minimum spanning tree of a weighted graph using the Kruskal's algorithm which requires $\mathcal{O}(e \log v)$ time, where e is the number of edges and v the number of vertices [20]. In our case, we have $v = q+2$ and $e = K(q+2)$, where K is the number of nearest neighbour points.

ii. Organize the points among the different patches. This can be done with the use of the so-called kd-trees data structures [13]. Taking into account that the augmented data set for the implicit approach consists of $3(q+2)$ points, the construction of a tree requires $\mathcal{O}(6(q+2)\log 3(q+2))$ time. Once the tree is built, we can search for all the points lying in a given patch in $\mathcal{O}(\log(3q+6))$ time [5]. We can eventually use the so-called integer-based data structure [7, 8] that needs $\mathcal{O}(3(q+2))$ time to organize the points among the different patches and $\mathcal{O}(1)$ time complexity for the searching routine.

iii. Construct the PU interpolant. This consists in solving d linear systems of size $q_j \times q_j$, with $q_j \ll 3(q+2)$, thus requiring an $\mathcal{O}(q_j^3)$ running time for each patch.

5 Numerical results

To test the method described in the previous sections, let us consider the following model describing a population affected by a disease [19]:

$$\frac{dP}{dt} = r(1-P)(P-u)P - \alpha I,$$

$$\frac{dI}{dt} = [-\alpha - d - ru + (\sigma - 1)P - \sigma I]I,$$

where P is the dimensionless total population that is composed of infected individuals I and susceptible individuals $P - I$. Referring to [19], we set $r = 0.2$, $u = 0.1$, $d = 0.25$ and $\alpha = 0.1$; furthermore we fix $\sigma = 4.08$. With this choice there exists exactly one endemic steady state $\boldsymbol{E}_1$ which is a centre of periodic orbits. Moreover the origin $\boldsymbol{E}_0$ is asymptotically stable (see Figure 2). This situation suggests the existence of a curve separating the paths tending to $\boldsymbol{E}_0$ from the trajectories following periodic cycles around $\boldsymbol{E}_1$.

Letting $q = 46$, the points describing such curve are plotted in Figure 3 (left). Moreover, to reconstruct the curve we consider the following basis functions

$$\begin{array}{lcll} \phi_1(r) & = & e^{-(cr)^2}, & C^\infty \quad \text{Gaussian (G)} \\ \phi_2(r) & = & e^{-cr}(15 + 15cr + 6(cr)^2 + (cr)^3), & C^6 \quad \text{Matérn (M6)} \\ \phi_3(r) & = & (1-cr)_+^4(4cr+1), & C^2 \quad \text{Wendland (W2)} \end{array}$$

where $c \in \mathbb{R}^+$ is the so-called shape parameter and r is the Euclidean distance. Note that ϕ_3 is compactly supported and this might be an advantage when the interpolation system is ill-conditioned.

To construct the curve, we consider 4 subdomains and we evaluate the interpolant on $\xi = 40$ equally spaced points, $\tilde{\boldsymbol{x}}_i$ $i = 1, \ldots, \xi$. The tolerance for the bisection algorithm is taken equal to 10^{-4} and $\gamma = 2$. Moreover, to point out the accuracy we estimate via cross-validation the Root Mean Square Error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{\xi}\sum_{i=1}^{\xi} |f(\tilde{\boldsymbol{x}}_i) - \mathcal{I}(\tilde{\boldsymbol{x}}_i)|^2}.$$

Finally, we also calculate the Maximum Condition Numbers (MCNs) of the local interpolation matrices, i.e.

$$\text{MCN} = \max_{j=1,\ldots,d} ||A_j||_2 ||A_j^{-1}||_2.$$

In the right frame of Figure 3 we plot the curve reconstructed by taking for the W2 function the shape parameter $c = 1$. This error is compared with the ones obtained with the G and M6 kernels (see Table 1). It is well-known that the G function can give arbitrarily high convergence orders, but dealing with real data, the ill-conditioning usually leads to inaccurate and meaningless

approximations for smooth RBFs. This is the case of the application at hand; in fact the G function is not able to fit the curve (this is the reason why we do not report any error in Table 1). On the contrary, taking advantage of the local support, the W2 function results particularly effective in this case. Of course, the choice of the shape parameter influences the accuracy of the approximation, but the only way to completely overcome the instability issue for the G kernel consists in using stable bases [2, 10, 12, 16].

While the method presented here and the one shown in [9] are comparable in terms of accuracy, the difference in terms of CPU times is evident from Table 2. Indeed, by using the fast detection procedure presented here, we register a sizeable saving in terms of computational time.

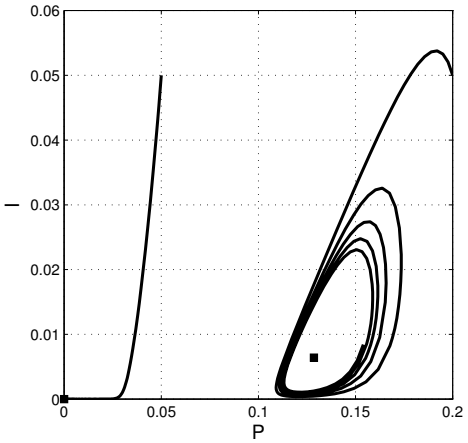

Figure 2: An illustrative example of two different trajectories evolving toward $\boldsymbol{E}_0$ and $\boldsymbol{E}_1$ (represented by squares).

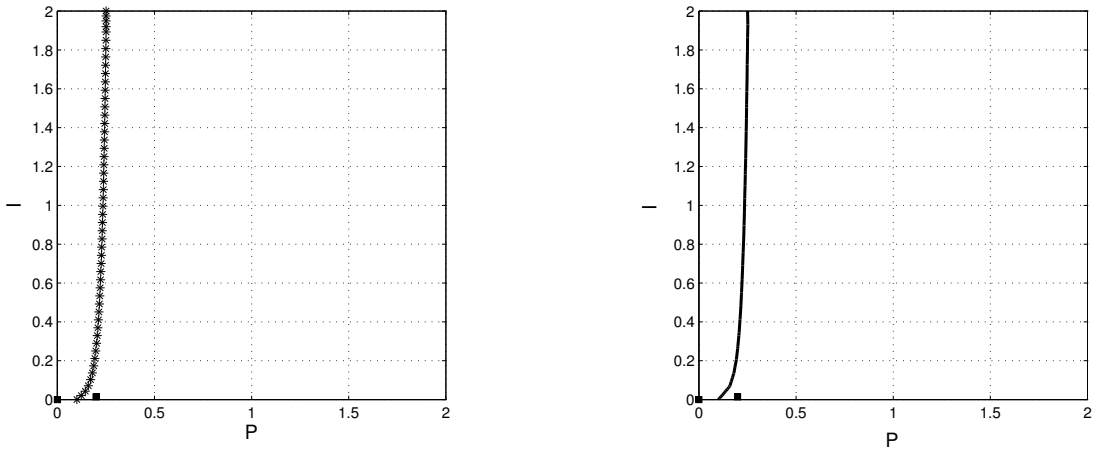

Figure 3: Set of points lying on the curve separating the domains of attraction of $\boldsymbol{E}_0$ and $\boldsymbol{E}_1$ (left) and the reconstruction of the separatrix curve (right). The squares represent $\boldsymbol{E}_0$ and $\boldsymbol{E}_1$.

	G	M6	W2
RMSE	——	6.28E−03	2.92E−03
MCN	9.27E+19	6.616E+12	1.08E+06

Table 1: RMSEs and MCNs for different kernels with shape parameter $c = 1$. Note that the measure of the error for the G case is not available because the approximation fails in view of the high condition number.

$q+2$	12	24	48	96
t_1	12.73	18.50	35.76	72.58
t_2	4.08	7.30	14.13	27.13

Table 2: CPU times (in seconds). With t_1 we denote the time obtained with the algorithm presented in [9], while t_2 identifies the CPU time of the new detection algorithm. Tests have been carried out with the MATLAB software on a Intel(R) Core(TM) i7 CPU 4712MQ 2.13 GHz processor.

6 Acknowledgements

We sincerely thank the reviewer for helping us to significantly improve the paper. This research has been accomplished within Rete ITaliana di Approssimazione (RITA). The first and third authors are partially supported by the 2016-2017 project "Metodi numerici e computazionali per le scienze applicate" of the Department of Mathematics of the University of Torino. Moreover, this research was supported by GNCS–INdAM 2016-2017 project. The second author is supported by the research project "Radial basis functions approximations: stability issues and applications" of the Department of Mathematics of the University of Padova, No. BIRD167404.

References

[1] M. BEHZAD, G. CHARTRAND, L. LESNIAK-FOSTER, *Graphs and Digraphs*, Prindle, Weber and Schmidt, Boston, 1979.

[2] M. BOZZINI, L. LENARDUZZI, M. ROSSINI, R. SCHABACK, *Interpolation with variably scaled kernels*, IMA J. Numer. Anal. **35** (2015), pp. 199–219.

[3] F. BRAUER, C. CASTILLO-CHAVEZ, *Mathematical Models in Population Biology and Epidemiology*, Springer-Verlag, New York, 2001.

[4] J.C. CARR, R.K. BEATSON, J.B. CHERRIE, T.J. MITCHELL, W.R. FRIGHT, B.C. MCCALLUM, T.R. EVANS, *Reconstruction and representation of 3D objects with radial basis functions*, in: Proceedings of the 28-th Annual Conference on Computer Graphics and Interactive Techniques, ACM press, New York, 2001, pp. 67–76.

[5] R. CAVORETTO, *A numerical algorithm for multidimensional modeling of scattered data points*, Comput. Appl. Math. **34** (2015), pp. 65–80.

[6] R. CAVORETTO, S. CHAUDHURI, A. DE ROSSI, E. MENDUNI, F. MORETTI, M. RODI, E. VENTURINO, *Approximation of dynamical system's separatrix curves*, in: T.E. Simos et al. (Eds.), Proceedings of the International Conference of Numerical Analysis and Applied Mathematics, AIP Conference Proceedings, Melville, vol. 1389, 2011, pp. 1220–1223.

[7] R. CAVORETTO, A. DE ROSSI, *A trivariate interpolation algorithm using a cube-partition searching procedure*, SIAM J. Sci. Comput. **37** (2015), pp. A1891–A1908.

[8] R. CAVORETTO, A. DE ROSSI, E. PERRACCHIONE, *Optimal selection of local approximants in RBF-PU interpolation*, J. Sci. Comput. **74** (2018), pp. 1–22.

[9] R. CAVORETTO, A. DE ROSSI, E. PERRACCHIONE, E. VENTURINO, *Robust approximation algorithms for the detection of attraction basins in dynamical systems*, J. Sci. Comput. **68** (2016), pp. 395–415.

[10] R. CAVORETTO, G.E. FASSHAUER, M. MCCOURT, *An introduction to the Hilbert-Schmidt SVD using iterated Brownian bridge kernels*, Numer. Algorithms **68** (2015), pp. 393–422.

[11] S. DE MARCHI, A. ISKE, A. SIRONI, *Kernel-based image reconstruction from scattered radon data*, Dolomites Res. Notes Approx. **9** (2016), pp. 19–31.

[12] S. DE MARCHI, G. SANTIN, *Fast computation of orthonormal basis for RBF spaces through Krylov space methods*, BIT **55** (2015), pp. 949–966.

[13] G.E. FASSHAUER, *Meshfree Approximations Methods with* MATLAB, World Scientific, Singapore, 2007.

[14] E. FRANCOMANO, F.M. HILKER, M. PALIAGA, E. VENTURINO, *An effcient method to reconstruct invariant manifolds of saddle points*, Dolomites Res. Notes Approx. **10** (2017), pp. 25–30.

[15] E. FRANCOMANO, F.M. HILKER, M. PALIAGA, E. VENTURINO, *On basins of attraction for a predator-prey model via meshless approximation*, in: T.E. Simos et al. (Eds.), Proceedings of the International Conference and Summer School Numerical Computations: Theory and Algorithms, AIP Conference Proceedings, Melville, vol. 1776, 2016, pp. 070007-1–070007-4.

[16] B. FORNBERG, E. LARSSON, N. FLYER, *Stable computations with Gaussian radial basis functions*, SIAM J. Sci. Comput. **33** (2011), pp. 869–892.

[17] P. GIESL, H. WENDLAND, *Numerical determination of the basin of attraction for asymptotically autonomous dynamical systems*, Nonlinear Anal. Theor. **75** (2012), pp. 2823–2840.

[18] R.J. GOULD, *Graph Theory*, Dover Publications, Mineola, 2012.

[19] F.M. HILKER, M. LANGLAIS, H. MALCHOW, *The Allee effect and infectious diseases: Extinction, multistability, and the (dis-)appearance of oscillations*, Am. Nat. **173** (2009), pp. 72–88.

[20] H. HOPPE, T. DEROSE, T. DUCHAMP, J. MCDONALD, W. STUETZLE, *Surface reconstruction from unorganized points*, in: J.J. Thomas (Ed.), Proceedings of the 19-th annual conference on Computer graphics and interactive techniques, ACM press, New York, vol. 26, 1992, pp. 71–78.

[21] A.J. LOTKA, *Elements of Physical Biology*, Williams & Wilkins, Baltimore, 1925.

[22] T.R. MALTHUS, *An Essay on the Principle of Population*, J. Johnson in St. Paul's Churchyard, London, 1789.

[23] J.D. MURRAY, *Mathematical Biology*, Springer-Verlag, Berlin, 1993.

[24] E. PERRACCHIONE, I. STURA, *RBF kernel method and its applications to clinical data*, Dolomites Res. Notes Approx. **9** (2016), pp. 13–18.

[25] S.N. QASEM, S.M. SHAMSUDDIN, *Radial basis function network based on time variant multi-objective particle swarm optimization for medical diseases diagnosis*, Appl. Soft Comput. **11** (2011), pp. 1427–1438.

[26] V. SHCHERBAKOV, E. LARSSON, *Radial basis function partition of unity methods for pricing vanilla basket options*, Comput. Math. Appl. **71** (2016), pp. 185–200.

[27] V. VOLTERRA, *Variazioni e fluttazioni del numero d'individui in specie animali conviventi*, Mem. Acad. Sci. Lincei **2** (1926), pp. 31–113.

[28] H. WENDLAND, *Fast evaluation of radial basis functions: Methods based on partition of unity*, in: C.K. Chui et al. (Eds.), Approximation Theory X: Wavelets, Splines, and Applications, Vanderbilt Univ. Press, Nashville, 2002, pp. 473–483.