Proceedings of the Allerton Conference on Communication, Control and Computing
pages 640–649, 28-30 September 2005.

Polynomial level-set methods for nonlinear dynamical systems analysis

Ta-Chung Wang[1,4] Sanjay Lall[2,4] Matthew West[3,4]

Submitted to Allerton Conference on Communication, Control, and Computing 2005

Abstract

In this paper, we present a method for computing the domain of attraction for non-linear dynamical systems. We propose a level-set method where sets are represented as sublevel sets of polynomials. The problem of flowing these sets under the advection map of a dynamical system is converted to a semidefinite program, which we use to compute the coefficients of the polynomials. We further address the related problems of constraining the degree of the polynomials and the connectedness of the associated sets.

Keywords: Domain of attraction, semi-definite programming, level-set methods.

1 Introduction

For nonlinear control systems, one would often like to know the region of attraction of an equilibrium point. Often, this region is difficult to both find and represent computationally. In this paper, we present an approach using polynomials to represent the domain of attraction, and semidefinite programming to perform the computation. The algorithm is iterative, and proceeds by advecting the sublevel set of the polynomial under the inverse flow map.

The usual mathematical tool used for analysis of the region of attraction is Lyapunov's method. This gives us a sufficient condition for local stability, although it is often difficult to find a Lyapunov function that can be used as a certificate for the whole domain of attraction. Several prior approaches have used quadratic functions, for example [1, 7, 2]. In particular, the approach of [2] makes use of semidefinite programming to find a quadratic function whose sublevel-set is a good inner approximation to the region of attraction. For system in which the region is complicated, an ellipsoid may not provide a good approximation, and the above methods leave a large unexplored region within the domain of attraction.

With recent developments in algebra and sum-of-squares techniques, it is now possible to solve for a Lyapunov function with a more general polynomial form [12, 10]. Positive definiteness properties are replace by sum-of-squares constraints which can be efficiently solved using convex optimization. The SOSTOOLS [13] toolbox for MATLAB has been developed as an easy computational tool to solve problems that utilizes the sum-of-squares techniques. This approach has also allowed finding a Lyapunov function within some specified semi-algebraic region [11, 3]. However, while this provides a method to certify a given inner approximation to the region of attraction, it does not immediately provide a way to find it.

In this paper, we make use of backward advection of a small initial neighborhood of the equilibrium in order to give an algorithm that in many cases converges to the true domain of attraction. The approach is similar in spirit to the level-set methods that have been used for computation of reachable sets [8, 9]. The key distinction is that most level-set methods represent the function on a mesh; we represent the function as a polynomial. The consequence of this is that the computational requirements may grow more slowly with dimension, if one may fix a-priori the required degrees of the polynomials. By contrast, a mesh-based

[1]Email: tachung@stanford.edu
[2]Email: lall@stanford.edu
[3]Email: westm@stanford.edu
[4]Department of Aeronautics and Astronautics 4035, Stanford University, Stanford CA 94305-4035, U.S.A.

The first two authors were partially supported by the Stanford URI *Architectures for Secure and Robust Distributed Infrastructures*, AFOSR DoD award number 49620-01-1-0365.

method has computational costs which grow exponentially with dimension. In this paper we give numerical examples in two and three dimensions, computed using the SOSCODE [5] toolbox. We also provide numerical comparisons of our results with several other techniques.

1.1 Preliminaries

Throughout this paper, we will use $\mathbb{R}_+$ to represent the set of nonnegative real numbers. For $z \in \mathbb{R}^n$ and $\varepsilon > 0$ we denote the open ball by $B_\varepsilon(z)$ given by

$$B_\varepsilon(z) = \left\{ x \in \mathbb{R}^n \,\middle|\, \|x - z\| < \varepsilon \right\}$$

Suppose $f : \mathbb{R}^n \to \mathbb{R}^n$. For $S \subset \mathbb{R}^n$, we say f is ***Lipschitz*** on S if there exists $c > 0$ such that

$$\|f(x) - f(y)\| \leq c\|x - y\| \qquad \text{for all } x, y \in S$$

We say f is ***locally Lipschitz*** on S if for all $z \in S$ there exists $\varepsilon > 0$ such that f is Lipschitz on $B_\varepsilon(z)$. Note that if f is locally Lipschitz on S then it is Lipschitz on any compact subset of S. For vector spaces X and Y, let $C(X, Y)$ be the set of functions mapping X to Y. For a function $g : \mathbb{R}^n \to \mathbb{R}$, the derivative $Dg(x)$ is a linear map $Dg(x) : \mathbb{R}^n \to \mathbb{R}$ at each point x.

We use $\mathbb{R}[x]$ to represent the ring of polynomials in x with real coefficients. A polynomial $f \in \mathbb{R}[x]$ is called positive semidefinite (PSD) if $f(x) \geq 0$, for all $x \in \mathbb{R}^n$. A polynomial f is called a ***sum-of-squares*** (SOS) if there exist polynomials $g_1, ..., g_s \in \mathbb{R}[x]$ such that $f = g_1^2 + g_2^2 + \cdots + g_s^2$. Clearly if f is SOS then f is PSD. It is also well-known that the converse is not true. We use Σ to denote the set of all SOS polynomials in $\mathbb{R}[x]$.

Suppose $g : \mathbb{R}^n \to \mathbb{R}$ is C^1. Define 0-***sublevel set*** of g to be $\mathcal{C}(g) \subset \mathbb{R}^n$ given by $\mathcal{C}(g) = \{ x \in \mathbb{R}^n \mid g(x) \leq 0 \}$. Further define the ***variety*** of g by be $\mathcal{V}(g) \subset \mathbb{R}^n$ given by $\mathcal{V}(g) = \{ x \in \mathbb{R}^n \mid g(x) = 0 \}$. Then we have $\mathcal{V}(g) \supset \partial\mathcal{C}(g)$, and both $\mathcal{V}(g)$ and $\mathcal{C}(g)$ are closed when g is a continuous function.

2 Set advection and the region of attraction

Suppose $f : \mathbb{R}^n \to \mathbb{R}^n$. In this paper, we will consider the following autonomous system

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

We make the following assumptions about f, to which we will refer later.

Assumption 1. *The assumptions are as follows.*

(i) $f(0) = 0$*, i.e., the origin is an equilibrium point.*

(ii) f *is locally Lipschitz on* $\mathbb{R}^n$*.*

(iii) For any $z \in \mathbb{R}^n$ *there exists a unique differentiable function* $x : \mathbb{R} \to \mathbb{R}^n$ *such that* $x(0) = z$ *and*

$$\dot{x}(t) = f(x(t)) \qquad \textit{for all } t \in \mathbb{R}$$

Notice that we assume existence of solutions for both positive and negative time. We make the following standard definitions. For f satisfying Assumption 1, the ***flow map*** $\phi : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ is defined to be the unique solution of

$$\begin{aligned} \frac{\partial \phi_t(z)}{\partial t} &= f(\phi_t(z)) \quad \text{for all } t \geq 0, z \in \mathbb{R}^n \\ \phi_0(z) &= z \end{aligned}$$

We say the ***origin is stable*** if for all $\varepsilon > 0$ there exists $\delta > 0$ such that

$$\|\phi_t(z)\| < \varepsilon \qquad \text{for all } z \in B_\delta(0) \text{ and } t \geq 0$$

We say the ***origin is asymptotically stable*** if the origin is stable and there exists $\delta > 0$ such that

$$\lim_{t\to\infty} \phi_t(z) = 0 \qquad \text{for all } z \in B_\delta(0)$$

A set $S \subset \mathbb{R}^n$ is called ***positively invariant*** if $\phi_t(S) \subset S$ for all $t \geq 0$ and it is called ***invariant*** if $\phi_t(S) \subset S$ for all $t \in \mathbb{R}$.

Definition 2. *Suppose $f : \mathbb{R}^n \to \mathbb{R}^n$ satisfies Assumption 1. Define the **domain of attraction** of f to be $R \subset \mathbb{R}^n$ given by*

$$R = \left\{ x \in \mathbb{R}^n \;\middle|\; \lim_{t\to\infty} \phi_t(x) = 0 \right\}$$

Lemma 3. *The domain of attraction is invariant; that is*

$$\phi_t(R) \subset R \qquad \textit{for all } t \in \mathbb{R}$$

Proof. Suppose $x \in \phi_t(R)$. Then there exists $y \in R$ such that $x = \phi_t(y)$. Hence

$$\begin{aligned} \lim_{s\to\infty} \phi_s(x) &= \lim_{s\to\infty} \phi_{s+t}(y) \\ &= 0 \end{aligned}$$

and so $x \in R$ as desired. ∎

Lemma 4. *For any $t \in \mathbb{R}$, the map $\phi_t : \mathbb{R}^n \to \mathbb{R}^n$ is continuous, invertible and has a continuous inverse; that is it is a topological homeomorphism on $\mathbb{R}^n$.*

Proof. This is standard; see for example Theorem 3.5 in [4]. ∎

Lemma 5. *Suppose f satisfies Assumption 1, and $R \neq \emptyset$. Suppose also $S_1 \subset R$ and $0 \in S_1$, and S_1 is a connected closed positively invariant set. Let $h > 0$ be a positive constant, and define the **backwards advection** of S_1 to be S_2, given by*

$$S_2 = \phi_{-h}(S_1)$$

Then $S_1 \subset S_2 \subset R$, and S_2 is also connected, closed and positively invariant. Further $\partial S_2 = \phi_{-h}(\partial S_1)$.

Proof. Firstly, we will show $S_1 \subseteq S_2$. Since S_1 is positively invariant, we have

$$\phi_h(S_1) \subset S_1$$

and since ϕ_{-h} is a topological homeomorphism we have $S_1 \subset \phi_{-h}(S_1)$, that is $S_1 \subseteq S_2$ as desired. To show positive invariance of S_2, notice that for any $t \geq 0$

$$\begin{aligned} \phi_h\big(\phi_t(S_2)\big) &= \phi_t\big(\phi_h(S_2)\big) \\ &= \phi_t(S_1) \\ &\subset S_1 \end{aligned}$$

since S_1 is positively invariant. Taking ϕ_{-h} of both sides, we have $\phi_t(S_2) \subset S_2$ as desired. To show that $S_2 \subset R$, suppose $z \in S_2$. Then

$$\begin{aligned} \lim_{t\to\infty} \phi_t(z) &= \lim_{t\to\infty} \phi_{t-h}\big(\phi_h(z)\big) \\ &= 0 \end{aligned}$$

since $\phi_h(z) \in S_1$. Finally, connectedness, closedness and preservation of the boundary follow because ϕ_h is a topological homeomorphism on $\mathbb{R}^n$. ∎

2.1 Convergence of advection

Theorem 6. *Suppose f satisfies Assumption 1, and $h > 0$. Also suppose $S_0 \subset R$ is a closed connected positively invariant set, such that there exists $\varepsilon > 0$ such that $B_\varepsilon(0) \subset S_0$.*

Define the sequence of sets $S_0, S_1, S_2, \ldots$ by

$$S_{k+1} = \phi_{-h}(S_k) \qquad \text{for } k = 0, 1, 2, \ldots$$

Then this sequence converges to R in the following sense:

(i) $S_k \subset R$ for all $k \in \mathbb{N}$.

(ii) $S_k \subset S_{k+1}$ for all $k \in \mathbb{N}$.

(iii) For every $x \in R$, there exists n such that $x \in S_n$

Proof. Parts (i) and (ii) follow from Lemma 5. To show part (iii), notice that given $x \in R$, there exists T such that $\phi_t(x) \in B_\varepsilon(0)$ for all $t \geq T$. Pick n to be any integer such that $n > T/h$. Then $\phi_{nh}(x) \in S_0$ and hence $x \in S_n$. ∎

3 Level-set representations

We represent a subset of $\mathbb{R}^n$ as the sublevel-set of a continuous function. Given $h > 0$, we define the time h advection operator $A_h : C(\mathbb{R}^n, \mathbb{R}) \to C(\mathbb{R}^n, \mathbb{R})$ by

$$p = A_h q \quad \text{if} \qquad p(x) = q\big(\phi_{-h}(x)\big) \text{ for all } x \in \mathbb{R}^n$$

The map A_h is also called the Liouville operator associated with f. A very important property is that it is linear.

In this paper our objective is to use convex optimization to construct the advection $A_h q$ given q. To do this we need to specify a class of such functions with additional properties. We now turn to the first such property.

Definition 7. *A set $S \in \mathbb{R}^n$ is called **star-shaped** if for all $x \in S$,*

$$\lambda x \in S \qquad \text{for all } \lambda \in [0, 1]$$

*The set S is called **strictly star-shaped** if for all $x \in S$,*

$$\lambda x \in \operatorname{int}(S) \qquad \text{for all } \lambda \in [0, 1)$$

Note that a star shaped set S is connected. We now give a simple sufficient condition that ensures that a sublevel set is star-shaped. We make the following definition.

Definition 8. *Suppose $g : \mathbb{R}^n \to \mathbb{R}$. We call g **strictly star-shaped** if g is C^1 and further satisfies $g(0) < 0$ and*

$$Dg(x)x > 0 \qquad \text{for all } x \neq 0$$

Lemma 9. *Suppose $g : \mathbb{R}^n \to \mathbb{R}$ is strictly star-shaped. Then $\mathcal{C}(g)$ is star-shaped.*

Proof. Suppose $x \in \mathcal{C}(g)$. Let $y : \mathbb{R}_+ \to \mathbb{R}^n$ be the function

$$y(t) = e^{-t}x$$

We would like to show that $y(t) \in \mathcal{C}(g)$ for all $t \geq 0$. We have

$$\begin{aligned} \frac{d}{dt} g(y(t)) &= -Dg\big(y(t)\big)y(t) \\ &< 0 \end{aligned}$$

for all $t \geq 0$. Also

$$g\big(y(t)\big) - g\big(y(0)\big) = \int_0^t \frac{d}{dt} g\big(y(t)\big)\, dt$$

and since $y(0) = x$ we have $g\big(y(t)\big) < 0$ for all $t \geq 0$ as desired. ∎

For the purposes of this paper, we would like to construct a convex set of functions whose sublevel sets are connected. Although the convex set of all convex functions on $\mathbb{R}^n$ will suffice, using it would unnecessarily restrict the class of sets describable to be convex. One cannot simply use the set of all functions whose 0-sublevel set is connected, since this set of functions is not convex. We therefore choose the set of *strictly star-shaped* functions, which is a convex set. We will use strictly-star-shaped polynomials to represent sets. This is significantly more general than previous approaches using quadratic functions.

Theorem 10. *Suppose g is strictly star-shaped. Then $\mathcal{C}(g)$ is strictly star-shaped.*

Proof. We know $\mathcal{C}(g)$ is star-shaped. We need to show that if $y \in \partial\mathcal{C}(g)$ then there does not exist $\lambda \in [0, 1)$ such that $\lambda y \in \partial\mathcal{C}(g)$. Suppose for the sake of a contradiction that there does exist such a y and λ. We know that $\lambda > 0$, since $g(0) < 0$. Define the function $h : [0, 1] \to \mathbb{R}$ by

$$h(\theta) = g(\theta y) \qquad \text{for } \theta \in [0, 1]$$

Then the derivative of h is

$$\begin{aligned} h'(\theta) &= \frac{1}{\theta} Dg(\theta y)(\theta y) \\ &> 0 \end{aligned}$$

for all $\theta \in (0, 1)$. From the assumptions we know $h(\lambda) = 0$ and $h(1) = 0$. Since h is C^1 on $[\lambda, 1]$, by the mean-value theorem there must exist $\theta \in (\lambda, 1)$ such that $h'(\theta) = 0$, which is a contradiction. ∎

Lemma 11. *Suppose $S \subset \mathbb{R}^n$ is bounded and $0 \in S$. Suppose $x \in \mathbb{R}^n$ and $x \neq 0$, and define the half-line $L(x)$ by*

$$L(x) = \big\{\, \lambda x \mid \lambda \geq 0 \,\big\}$$

Then $L(x) \cap \partial S \neq \emptyset$.

Proof. One may construct a point in the intersection as follows. Let

$$\theta = \sup\big\{\, \lambda \in \mathbb{R} \mid \lambda x \in S \,\big\}$$

Then θ is finite since $x \neq 0$ and S is bounded. Also $\theta x \in L(x) \cap \partial S$ as required. ∎

Theorem 12. *Suppose g is strictly star shaped and $\mathcal{C}(g)$ is bounded. Then $\mathcal{V}(g) = \partial\mathcal{C}(g)$.*

Proof. We know that $\mathcal{V}(g) \supset \partial\mathcal{C}(g)$, so all we need to show is that $\mathcal{V}(g) \subset \partial\mathcal{C}(g)$. Suppose for the sake of a contradiction that this is not so, that is there exists $x \in \mathcal{V}(g)$ such that $x \notin \partial\mathcal{C}(g)$. Then $x \neq 0$, since g is strictly star shaped, and by Lemma 11 there exists $q \in L(x) \cap \partial\mathcal{C}(g)$. Also $q \neq x$ since $x \notin \partial\mathcal{C}(g)$. Now define the function $h : [0, 1] \to \mathbb{R}$ by

$$h(\theta) = g(\theta q) \qquad \text{for all } \theta \in [0, 1]$$

There exists $\lambda \in (0, 1)$ such that $x = \lambda q$, since $q \in L(x)$. As in the proof of Theorem 10, we know $h'(\theta) > 0$ for all $\theta \in (\lambda, 1)$, but $h(\lambda) = 0$ and $h(1) = 0$, and so the mean-value theorem implies that there exists $\theta \in (\lambda, 1)$ such that $h'(\theta) = 0$, a contradiction. ∎

The following simple lemma relates the advection operator to the advection of sets.

Lemma 13. *Suppose g_1, g_2 are functions mapping $\mathbb{R}^n$ to $\mathbb{R}$. If $g_2 = A_h g_1$ then $\mathcal{C}(g_2) = \phi_h\big(\mathcal{C}(g_1)\big)$.*

Proof. The proof follows from the definitions above. ∎

4 Computation

We use the following approach [11, 6] to computation with semialgebraic sets. In particular, we would like to computationally find a polynomial p, satisfying additional linear constraints (such as $p = A_h q$), such that $\mathcal{C}(p) \subset \mathcal{C}(q)$. The following result shows that this may be achieved using semidefinite programming.

Lemma 14. *Given $p, q \in \mathbb{R}[x]$, suppose there exist $s_0, s_1 \in \Sigma$ such that*

$$s_0 - s_1 q + p = 0 \qquad \text{for all } x \in \mathbb{R}^n \tag{2}$$

Then $\mathcal{C}(q) \subset \mathcal{C}(p)$. Further, given q, the set of coefficients of p, s_0, and s_1 satisfying (2) is the feasible set of a semidefinite program.

Proof. See, for example, [11] or [12]. ∎

4.1 Time-stepping

Since we are performing advection, we must using an approximation to the flow map ϕ_h. Many such approximations are possible, and are provided by the theory of numerical integration. Here we present two simple approximations. The first-order Taylor approximation to $p = A_h q$ is the map $B_h : C(\mathbb{R}^n, \mathbb{R}) \to C(\mathbb{R}^n, \mathbb{R})$ given by

$$p = B_h q \qquad \text{if } p(x) = q(x) - hDq(x)f(x)$$

An alternative approximation is C_h, given by

$$p = C_h q \qquad \text{if } p(x) = q\big(x - hf(x)\big)$$

Both of these have the following properties; they are both linear maps, and if q and f are polynomials then so is p. In this paper, we concentrate on B_h, since typically $B_h q$ is a polynomial whose degree is less than that of $C_h q$.

4.2 An algorithm for backward advection

Given a strictly star-shaped polynomial g_k such that $\mathcal{C}(g_k) \subset R$, and $\mathcal{C}(g_k)$ is bounded and positively invariant, we compute an approximation g_{k+1} such that $g_{k+1} \approx A_{-h} g_k$ as follows.

Pick $\alpha > 0$ and positive integer d. Solve, using semidefinite programming, the following feasibility problem for $g_{k+1} \in \mathbb{R}[x]$, $s_1, s_2, s_3, s_4 \in \Sigma$.

$$\begin{aligned} g_{k+1}(0) &= -1 \\ Dg_{k+1}(x)x &> 0 \\ s_3 - s_4 g_k + B_h g_{k+1} &= 0 \\ s_1 + s_2 g_k - B_{h+\alpha} g_{k+1} &= 0 \\ \deg(g_{k+1}) &\le d \end{aligned}$$

This algorithm is an *implicit method*; it is approximately solving $B_h g_{k+1} = g_k$. Alternatively one could form an *explicit* version by constructing $g_{k+1} = B_{-h} g_k$.

One parameter that must be chosen is d, the degree of the desired approximations; how large a d is necessary depends on the complexity of the system and its advected sets. The cost of computation grows polynomially with d, and so typically d is chosen as small as possible. Once the degree has been chosen, the map B_h is a matrix, operating on the coefficients of the polynomials.

The other important parameter is α, which we think of as follows. The above algorithm finds a degree d polynomial g_{k+1} such that g_{k+1} is strictly star shaped, $\phi_{h+\alpha}\mathcal{C}(g_{k+1}) \subset \mathcal{C}(g_k)$, and $\phi_h \mathcal{C}(g_{k+1}) \supset \mathcal{C}(g_k)$. Hence the parameter α may be thought of as a tolerance that allows for the constraint that g_{k+1} is required to have degree d or less.

4.3 Finding the initial semi-algebraic set

We find a local Lyapunov function in order to construct an initial star-shaped positively invariant set. The following result is standard.

Proposition 15. *Suppose $f : \mathbb{R}^n \to \mathbb{R}^n$ satisfies Assumption 1, $V : \mathbb{R}^n \to \mathbb{R}$ is a C^1 function, $\gamma > 0$, and the set*

$$D = \left\{ x \in \mathbb{R}^n \mid V(x) \leq \gamma \right\}$$

is compact. Further suppose

$$\begin{aligned} V(x) &> 0 \qquad \text{for all } x \neq 0 \\ V(0) &= 0 \\ DV(x)f(x) &< 0 \qquad \text{for all } x \neq 0, x \in D \end{aligned}$$

Let $g_0(x) = V(x) - \gamma$. Then $\mathcal{C}(g_0)$ is positively invariant, and $\mathcal{C}(g_0) \subset R$.

One simple approach to finding an initial sublevel set is to find a quadratic Lyapunov function for the linearization of the system, and use a small sublevel-set of this quadratic for the polynomial g_0.

An alternative method which often gives a much large initial set is as follows. Choose a polynomial $p \in \mathbb{R}[x]$ such that $\mathcal{C}(p) \subset R$. We then solve the following convex feasibility problem. Find $V \in \mathbb{R}[x]$ and $s_0, s_1 \in \Sigma$ such that

$$\begin{aligned} DV(x)x &> 0 \qquad \text{for all } x \neq 0 \\ V(x) &> 0 \qquad \text{for all } x \neq 0 \\ V(0) &= 0 \\ DV(x)f(x) + s_0 - s_1 p &= 0 \qquad \text{for all } x \neq 0 \end{aligned}$$

Similar methods for finding local Lyapunov functions may be found in [3, 10], along with details on the construction of the associated semidefinite program. Here we have added the first constraint to ensure that if $\gamma > 0$ then $V - \gamma$ is strictly star-shaped. Note that these constraints imply that all sublevel sets of V are compact. Given V, we then solve the convex program

$$\begin{aligned} \text{maximize} \quad & \gamma \\ \text{subject to} \quad & V - \gamma - s_0 - s_1 p - \varepsilon = 0 \qquad \text{for all } x \\ & s_0, s_1 \in \Sigma \end{aligned}$$

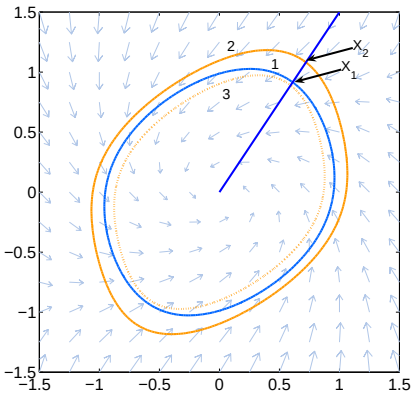

Figure 1: Example of stopping conditions.

where $\varepsilon > 0$ is small. The optimal γ satisfies $\mathcal{C}(V - \gamma) \subset \mathcal{C}(p)$. Then V and γ satisfy the assumptions of Proposition 15 and so we may use $g_0 = V - \gamma$ as the function defining our initial level set.

4.4 Stopping conditions

By using the proposed level-set method, we can successfully propagate the system states backward in time. However, we still need a stopping criterion for this iteration. We use the following method to measure the closeness of two semi-algebraic sets.

Theorem 16. *Suppose g_1 and g_2 are strictly star-shaped functions, $\mathcal{C}(g_1) \subset \mathcal{C}(g_2)$, and $\mathcal{C}(g_2)$ is bounded. Suppose $x_1, x_2 \in \mathbb{R}^n$ are two points such that*

$$x_1 \in \mathcal{V}(g_1) \qquad \text{and} \qquad x_2 \in \mathcal{V}(g_2)$$

and $x_1 = \alpha x_2$ for some $\alpha \geq 0$. Define the function $q : \mathbb{R}^n \to \mathbb{R}$ by

$$q(x) = g_2(\lambda x) \qquad \text{for all } x \in \mathbb{R}^n$$

where $\lambda > 1$. Then if $\mathcal{C}(q) \subset \mathcal{C}(g_1)$, we have

$$\frac{\|x_2 - x_1\|}{\|x_2\|} \leq 1 - \lambda^{-1}$$

Proof. We know q is strictly star-shaped since g_2 is, and therefore by Lemma 11 there is a unique $y \in L(x_1) \cap \mathcal{V}(q)$. We can construct this explicitly; it is $y = \lambda^{-1} x_2$. Since $\mathcal{C}(q) \subset \mathcal{C}(g_1) \subset \mathcal{C}(g_2)$ we have $\lambda^{-1} \leq \alpha \leq 1$. Therefore we have $\|x_2 - x_1\| = (1-\alpha)\|x_2\|$ and hence $\|x_2 - x_1\| \leq (1-\lambda^{-1})\|x_2\|$ as desired. ■

Hence to determine when the algorithm should terminate, one uses Lemma 14 to determine the smallest $\lambda > 1$ such that $\mathcal{C}(q) \subset \mathcal{C}(g_k)$, where $q(x) = g_2(\lambda x)$. Again, this may be evaluated using semidefinite programming. Note that if one has a known bound on the size of $\|x_2\|$, for example given by an outer bound on the region of attraction R, then this stopping criterion gives an absolute bound of λR. In practice, one picks a $\lambda > 1$ in advance, and checks this condition after each iteration. Figure 1 shows an example; here curve 1 is $\mathcal{C}(g_1)$, curve 2 is $\mathcal{C}(g_2)$ and curve 3 is $\mathcal{C}(q)$. The largest radial deviation between curves 1 and 2 is less than 0.3.

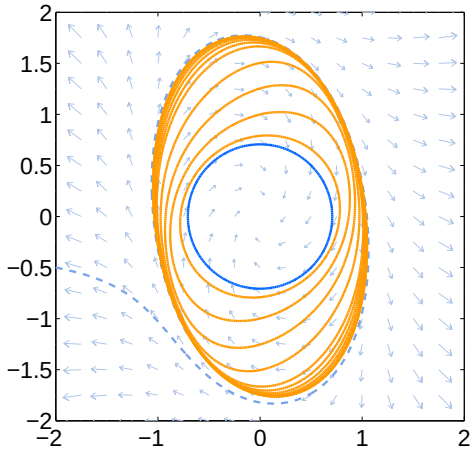

Figure 2: Successive iterates of the level-set algorithm

5 Numerical examples

Example 1. Consider the following dynamical system

$$\dot{x} = 0.5y - x(1 - x^2 - 0.25y^2)$$
$$\dot{y} = -x - 0.5y(1 - x^2 - 0.25y^2)$$

The origin is a locally stable equilibrium point. Here we start with the initial polynomial $g_0 = 2x^2 + 2y^2 - 1$. The results of the level-set method are shown in Figure 2, using time step $h = 0.2$. It can be seen that the successive iterates approach the true domain of attraction.

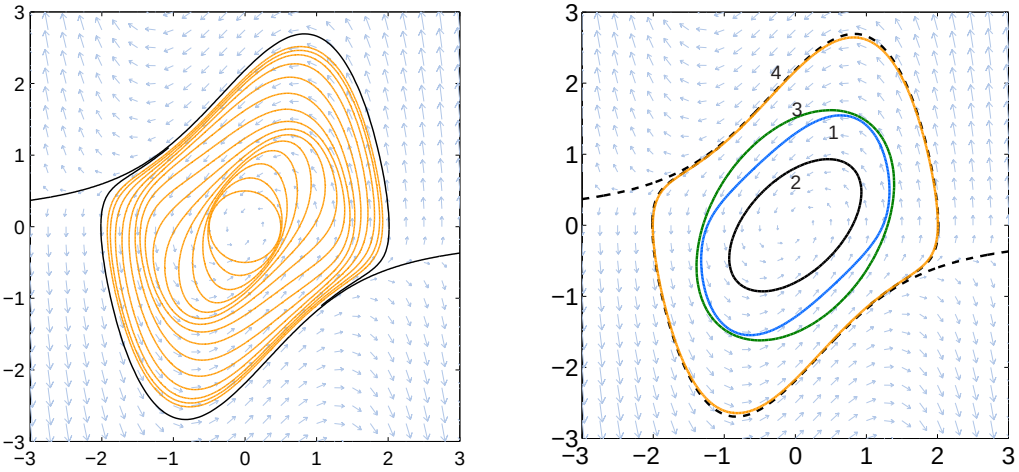

Figure 3: The Van der Pol oscillator. The left figure shows the sequence of iterations. The right figure shows the final iterate, and some other approximations.

Example 2. Consider the Van der Pol oscillator

$$\dot{x} = -y$$
$$\dot{y} = x - y(1 - x^2)$$

Again the system is locally stable around the origin. Here we use an initial sublevel set given by the quadratic polynomial $g_0 = 4x^2 + 4y^2 - 1$, which can be verified to be positively invariant. A time step of $h = 0.2$ is used. The even-numbered iterates $g_0, g_2, g_4, \ldots$ are shown in Figure 3. Some of the iterates are below, normalized to allow integer coefficients.

$$\begin{aligned}
p_2 &= -1000 + 2252\,y^2 - 88\,y^4 + 11\,y^6 - 907\,xy - 56\,xy^3 - 4\,xy^5 + 3883\,x^2 \\
&\quad + 360\,x^2y^2 - 57\,x^2y^4 + 660\,x^3y - x^3y^3 - 417\,x^4 + 21\,x^4y^2 + 81\,x^5y + 260\,x^6 \\
p_4 &= -1000 + 1614\,y^2 - 137\,y^4 + 16\,y^6 - 1654\,xy - 170\,xy^3 + 14\,xy^5 + 3162\,x^2 \\
&\quad + 480\,x^2y^2 - 43\,x^2y^4 + 94\,x^3y - 35\,x^3y^3 + 144\,x^4 - 2\,x^4y^2 + 192\,x^5y + 335\,x^6 \\
p_{28} &= -10000 + 2510\,y^2 - 56\,y^4 + 2\,y^6 - 4306\,xy + 42\,xy^3 + 4\,xy^5 + 4099\,x^2 \\
&\quad + 25\,x^2y^2 + 2\,x^2y^4 + 1103\,x^3y - 27\,x^3y^3 - 687\,x^4 - x^4y^2 + 2\,x^5y + 84\,x^6
\end{aligned}$$

It can also be seen that the iterates gradually approach the exact boundary of the domain of attraction. After 30 iterations, the solution covers most of the stable region.

After 40 iterations, the stopping criteria allowing an absolute radial change of 0.01 has been met. The final result is shown in Figure 3. Curve 1 is the boundary of $\mathcal{C}(g_0)$, when g_0 is obtained through the sdp-based procedure in Section 4.3. For comparison, curve 2 is the result of [7] and curve 3 is the result of [1].

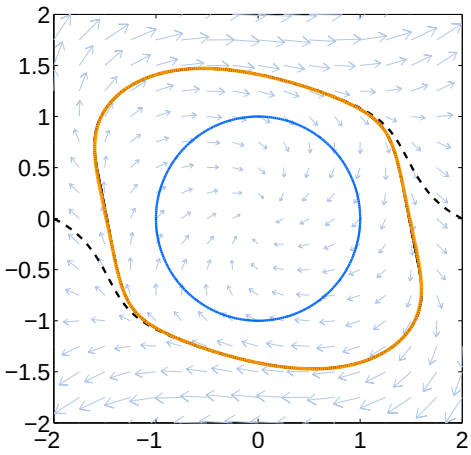

Figure 4: The sublevel sets of g_0 and g_{20}.

Example 3. The following dynamical system is Example S4, from [2].

$$\begin{aligned}
\dot{x} &= -2x + y + x^3 + y^5 \\
\dot{y} &= -x - y + x^2y^3
\end{aligned}$$

After 20 iterations, the estimated boundary of the DOA has reached the pre-specified bound. The final iterate, and the true boundary, are shown in Figure 4.

Example 4. The proposed level-set method also works in higher dimensions. In the derivation of the algorithm, we do not make any assumptions about the system dimensions. Consider the following 3D dynamical system.

$$\begin{aligned}
\dot{x} &= -x + x^3 + x^2y + 1.5xy^2 + xyz + 2xz^2 \\
\dot{y} &= -y + x^2y + xy^2 + 1.5y^3 + y^2z + 2yz^2 \\
\dot{z} &= -z + x^z + xyz + 1.5y^2z + yz^2 + 2z^3
\end{aligned}$$

This system has a known domain of attraction given by $\mathcal{C}(p)$ where

$$p = x^2 + xy + 1.5y^2 + yz + 2z^2 - 1$$

After 30 iterations with a time step of 0.2, the solution has reached the specified stopping condition (an absolute increment less than 0.01.) After removing the insignificant coefficients, this is

$$\begin{aligned} g_{30} = {} & 2.142z^2 - 0.2814z^4 + 0.01307z^6 + 1.005yz - 0.1522yz^3 + 0.007319yz^5 + 1.627y^2 \\ & - 0.4657y^2z^2 + 0.03183y^2z^4 - 0.1327y^3z + 0.01265y^3z^3 - 0.1878y^4 + 0.02689y^4z^2 + 0.005496y^5z \\ & + 0.008039y^6 - 0.01139xz + 0.02394xz^3 - 0.002257xz^5 + 1.014xy - 0.1576xyz^2 \\ & + 0.007450xyz^4 - 0.005287xy^2z - 0.0007966xy^2z^3 - 0.1460xy^3 + 0.01397xy^3z^2 + 0.0004539xy^4z \\ & + 0.006491xy^5 + 1.091x^2 - 0.3132x^2z^2 + 0.02162x^2z^4 - 0.07976x^2yz + 0.007274x^2yz^3 - 0.2705x^2y^2 \\ & + 0.03683x^2y^2z^2 + 0.006627x^2y^3z + 0.01680x^2y^4 + 0.01218x^3z - 0.002069x^3z^3 - 0.1011x^3y \\ & + 0.009490x^3yz^2 - 0.0006406x^3y^2z + 0.008830x^3y^3 - 0.08753x^4 + 0.01310x^4z^2 + 0.001690x^4yz \\ & + 0.01176x^4y^2 - 0.0007735x^5z + 0.003135x^5y + 0.002962x^6 - 1 \end{aligned}$$

We may compare this result with the known domain of attraction, using similar techniques to those for the stopping criterion. The maximum radial deviation from the exact bound is less than 1.42% of the longest radial distance of the true domain of attraction.

6 Conclusions

In this paper, we developed a level-set algorithm to advect subsets of the state-space using semidefinite programming. The sets are represented as semialgebraic sets, i.e., sublevel-sets of polynomials. We present an implicit time-stepping algorithm, whose solution converges to the domain of attraction when it is star-shaped. This level-set algorithm not only works for two-dimensional systems but also works for higher dimensional systems. We also provide a stopping criterion, and several numerical examples.

References

[1] E. J. Davision and E. M. Kurak. A computational method for determining quadratic Lyapunov functions for non-linear systems. *Automatica*, 7:627–636, 1971.

[2] G. Ghesi, A. Tesi, and A. Vicino. On optimal quadratic Lyapunov functions for polynomial systems. In *15th Int. Symp. on Mathematical Theory of Networks and Systems*, 2002.

[3] Z. W. Jarvis-Wloszek. *Lyapunov based analysis and controllers synthesis for polynomial systems using sum-of-squares optimization*. PhD thesis, University of California, Berkeley, 2003.

[4] H. Khalil. *Nonlinear systems*. Prentice-Hall Inc., third edition, 2000.

[5] S. Lall, M. Peet, and T. Wang. *SOSCODE: Sum of squares optimization toolbox for MATLAB*, 2005. Available from `http://www.stanford.edu/~lall/`.

[6] J. B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.

[7] A. Levin. An analytical method of estimating the domain of attraction for polynomial differential equations. *IEEE Transactions on Automatic Control*, 39(12):2471–2475, 1994.

[8] I. M. Mitchell and C. J. Tomlin. Level set methods for computation in hybrid systems. In *Hybrid Systems: Computation and Control*, LNCS 1790, pages 310–323. Springer Verlag, 2000.

[9] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer Verlag, 2002.

[10] A. Papachristodoulou and S. Prajna. On the construction of Lyapunov functions using the sum of squares decomposition. In *Proceedings of the 41st IEEE Conference on Decision and Control*, 2002.

[11] P. Parrilo. *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*. PhD thesis, California Institute of Technology, 2000.

[12] P. A. Parrilo and S. Lall. Semidefinite programming relaxations and algebraic optimization in control. *European Journal of Control*, 9(2–3):307–321, April 2003.

[13] S. Prajna, A. Papachristodoulou, and P. A. Parrilo. Introducing sostools: A general purpose sum of squares programming solver. In *Proceedings of the 41st IEEE Conference on Decision and Control*, December 2002.