

COMPUTING CONVEX HULLS OF TRAJECTORIES

DANIEL CIRIPOI, NIDHI KAIHNSA, ANDREAS LÖHNE, AND BERND STURMFELS

ABSTRACT. We study the convex hulls of trajectories of polynomial dynamical systems. Such trajectories include real algebraic curves. The boundaries of the resulting convex bodies are stratified into families of faces. We present numerical algorithms for identifying these patches. An implementation based on the software Bensolve Tools is given. This furnishes a key step in computing attainable regions of chemical reaction networks.

1. INTRODUCTION

Dynamics and convexity are ubiquitous in the mathematical sciences, and they come together in applied questions in numerous ways. We explore an interaction of dynamics and convexity that is motivated by reaction systems in chemical engineering [9, 18]. Consider an autonomous system of ordinary differential equations

$$\frac{d}{dt}x(t) = \phi(x(t)), \quad (1.1)$$

where $x : \mathbb{R} \rightarrow \mathbb{R}^n$ is an unknown function and $\phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a given polynomial map. By the Picard–Lindelöf Theorem, each initial value problem for (1.1) has a unique solution on a local interval. Although ϕ is assumed to be polynomial, for most of the techniques we develop it suffices for ϕ to be locally Lipschitz continuous. Any starting point $x(0)$ in \mathbb{R}^n gives rise to a unique trajectory $\mathcal{C} := \{x(t) \mid t \in [0, a) \text{ for } a > 0\}$. This curve may or may not converge to a stationary point, and its dynamics can be chaotic. We consider the case where the trajectory \mathcal{C} is bounded. If it is not bounded, we restrict time t to a finite interval.

Our object of study is the convex hull in \mathbb{R}^n of the trajectory starting from $t = 0$:

$$\text{convtraj}(x(0)) := \text{conv}(\mathcal{C}). \quad (1.2)$$

We call this set the *convex trajectory* of the point $x(0)$. By definition, it is the smallest convex set containing the trajectory. The convex trajectory need not be closed. In that case, we usually replace $\text{convtraj}(x(0))$ by its topological closure, so that the convex trajectory becomes a convex body, that is, a compact convex set with nonempty interior. We ensure full-dimensionality of the convex trajectory by restricting to the space in which this curve lies.

Nidhi Kaihnsa was funded by the International Max Planck Research School *Mathematics in the Sciences* (IMPRS). Bernd Sturmfels was partially supported by the US National Science Foundation (DMS-1419018).

This article introduces numerical methods that solve the following problems for the dynamical system (1.1) with respect to a given starting point $y = x(0)$ in \mathbb{R}^n :

- (i) Compute $\text{convtraj}(y)$. This is a convex body in \mathbb{R}^n . The output should be in a format that represents the boundary as accurately as possible.
- (ii) Decide whether the convex trajectory $\text{convtraj}(y)$ is forward closed. This happens if and only if the vector field $\phi(z)$ at every boundary point z points inwards, for every supporting hyperplane.

A subset $S \subset \mathbb{R}^n$ is *forward closed* if every trajectory of the dynamical system (1.1) that starts in S remains in S . In the literature, forward closed sets are also referred to as forward invariant sets. If $\text{convtraj}(y)$ is forward closed then it equals the *attainable region* of the point y for (1.1). The attainable region is the smallest subset of \mathbb{R}^n that contains y and is both convex and forward closed. We refer to [9, 14, 18] for motivation and many details.

This article enlarges the repertoire of *convex algebraic geometry*, a field that studies convex semialgebraic sets, and their role in polynomial optimization [3]. Indeed, every algebraic curve is locally the trajectory of a polynomial dynamical system. Hence, our results apply to convex hulls of algebraic curves [19]. While Problem (ii) is specific to dynamical systems, Problem (i) makes sense for any smooth curve that can be approximated sufficiently well. In particular, this is the case for parametric curves $t \mapsto x(t)$. See Figures 1, 2 and 4.

Approximations by polygonal curves are crucial for our approach. The boundary of the convex hull of a generic curve has a distinct structure that is characterized by families of polyhedral faces spanned by curve points. Our algorithms account for this by constructing polyhedral approximations of convex trajectories, using points that are sampled from the curve of interest. We will discuss conditions on \mathcal{C} and the sampling that assure the significance of our methods. More specifically, our results establish a connection between the facial structure of the convex bodies we study and the facets of the approximating polytopes.

Our presentation is organized as follows. Section 2 treats the planar case ($n = 2$). Example 2.3 illustrates our solution for the Hamiltonian system associated with the Trott curve. Section 3 develops a general geometric theory for taking limits of convex polytopes. This will be applied to identify the desired convex body (1.2) from a sequence of inner approximations by polytopes. We compute these polytopes using the `Matlab/Octave` package `Bensolve Tools` [5]. The optimization methodology that underlies this approach is explained in Section 4. Section 5 describes a stratification of a piecewise smooth convex body of dimension n . The strata are $(n - k - 1)$ -dimensional families of k -dimensional faces, called *patches*, for various k . Section 6 concerns dynamical systems (1.1) whose trajectories are algebraic or trigonometric curves. This includes linear dynamical systems. Table 1 offers data on their patches for $n = 3$. Section 7 presents our solution to Problem (ii). Algorithm 7.1 decomposes each patch of the convex trajectory (1.2) into two subsets, depending on whether the vector field $\phi(z)$ points inward or outward. If the outward set is always empty then (1.2) is forward closed. In Section 8 we focus on chemical reaction networks with mass action kinetics [11, 13]. These motivated

our study. We characterize planar algebraic curves that are trajectories of chemical reaction networks, we study the *van de Vusse system* [9, 18], and we exhibit a toric dynamical system [6] with $\text{convtraj}(y)$ not forward closed. This resolves [14, Conjecture 4.1].

2. PLANAR SCENARIOS

In this section we study the convex trajectories of dynamical systems in the plane ($n = 2$). Our input is a pair of polynomials $\phi = (\phi_1, \phi_2)$ along with a starting point $y = (y_1, y_2)$ in \mathbb{R}^2 . The resulting trajectory of (1.1) is a smooth plane curve \mathcal{C} that is parametrized by time t . We write $C = \text{conv}(\mathcal{C})$ for the associated convex trajectory. This is a convex region in \mathbb{R}^2 .

The boundary ∂C of the convex trajectory C consists of arcs on the curve \mathcal{C} and of edges that connect them. Each edge of C is a line segment between two points on \mathcal{C} . These are either points of tangency or endpoints of the curve. This partitions ∂C into *patches*, described in general in Section 5. Here, k -patches are edges (for $k = 1$) and arcs (for $k = 0$).

If we had an exact algebraic representation of the curve \mathcal{C} then we could use symbolic methods to compute its bitangents and derive from this a description of ∂C . For instance, if \mathcal{C} is an algebraic curve of degree four, as in Example 2.3, then it has 28 bitangent lines (over \mathbb{C}) which can be computed using Gröbner bases. In [2] and [15], methods for reducing the number of bitangent line computations are introduced for convex hull computations in the planar case. But, such algebraic representations are not available when we study dynamical systems. Each trajectory is an analytic curve $t \mapsto x(t)$. This parametrization is given indirectly, namely by the differential equation (1.1) it satisfies. Furthermore, our aim is not only to obtain a representation for the convex hull of a curve but also to characterize the structure of its boundary. We are not aware of any reports on such studies.

Algorithm 2.1. (Detection of edges and arcs for $n = 2$)

input : A list \mathcal{A} of points on a curve \mathcal{C} in \mathbb{R}^2 ; a threshold value $\delta > 0$.

output: The numbers $\#_0$ and $\#_1$ of arcs and edges of $C = \text{conv}(\mathcal{C})$.

For each i : list of curve points that represent the i th arc of ∂C .

List of line segments that represent the edges of C .

- 1 Compute the vertices \mathcal{V} and edges \mathcal{H} of $A = \text{conv}(\mathcal{A})$.
- 2 Build a graph G with node set \mathcal{H} such that two distinct edges H_1, H_2 of A form an edge of G if $H_1 \cap H_2 \neq \emptyset$ and both H_1 and H_2 have length $\leq \delta$.
- 3 Output the number $\#_1$ of isolated nodes of G and the number $\#_0$ of remaining connected components G_i .
- 4 **foreach** *nonsingleton connected component* G_i **do**
- 5 | Output a list of curve points that are endpoints of those edges of A , that
 | belong to G_i . This represents the i th arc of ∂C .
- 6 **end**
- 7 Edges H_j of A that correspond to isolated nodes of G represent edges of C .

We now assume that a polygonal approximation is given for the curve \mathcal{C} . Our input is a finite list \mathcal{A} of points $x(t_i)$ on \mathcal{C} . In our computations we solve the differential equation (1.1) numerically using the versatile solver `ode45` in `Matlab`. This generates the set \mathcal{A} of sample points which we assume to be reliably accurate. Using `ode45` also allows us to control the quality of the approximation. To ensure a certain precision of the approximation one could employ better methods for integrating dynamical systems like the one presented in [12].

As a first step we address Problem (i) in the Introduction. Algorithm 2.1 computes a representation of the boundary ∂C of the convex trajectory C from an approximation \mathcal{A} of the trajectory \mathcal{C} . A key idea is the identification of long edges in $A := \text{conv}(\mathcal{A})$. In Section 5 we generalize to curves in \mathbb{R}^n . Algorithm 2.1 is a special case of Algorithm 5.4.

We next solve Problem (ii) from the Introduction. For each point z on an arc of C , the vector $\phi(z)$ is tangent to the curve, so there is nothing to be checked at the arcs. To decide whether C is forward closed with respect to the dynamics (1.1), we must examine the edges of C . Suppose that $\text{conv}(x(t_i), x(t_j))$ is an edge of C , and consider its relative interior points

$$z = \lambda \cdot x(t_i) + (1 - \lambda) \cdot x(t_j) \quad \text{where } 0 < \lambda < 1. \quad (2.1)$$

The following 2×2 determinant is a polynomial in the parameter λ :

$$f(\lambda) = \det(\phi(z), x(t_i) - x(t_j)). \quad (2.2)$$

We compute all real zeros of the polynomial $f(\lambda)$ in the open interval $(0, 1)$. The zeros partition the edge of C into segments where $\phi(z)$ points either inward or outward, relative to the convex region C . If there are no zeros then the entire edge of C is either inward pointing or outward pointing. In this manner we partition ∂C , and thereby solve (ii). In order to compute the attainable region, we start new trajectories from a sample of outward points.

We tested our method on trajectories that are algebraic curves. Let \mathcal{C} be the curve in \mathbb{R}^2 defined by a polynomial equation $h(x, y) = 0$. The associated *Hamiltonian system* equals

$$\dot{x} = \frac{\partial h}{\partial y}(x, y) \quad \text{and} \quad \dot{y} = -\frac{\partial h}{\partial x}(x, y). \quad (2.3)$$

Henceforth, we only consider Hamiltonian systems associated with polynomials. Thus h is a polynomial in x and y . At any point that is not a critical point of h , the right hand side of (2.3) is orthogonal to the gradient vector of h . This means that the vector field is tangent to the *level curves* $h(x, y) = c$, where c ranges over \mathbb{R} . From this we infer the following well-known result.

Corollary 2.2. *Every trajectory of (2.3) is a piece of a level curve $\{h(x, y) = c\}$.*

We close this section by illustrating Hamiltonian systems and our solutions to Problems (i) and (ii) for $n = 2$, for a curve that is familiar in computational algebraic geometry.

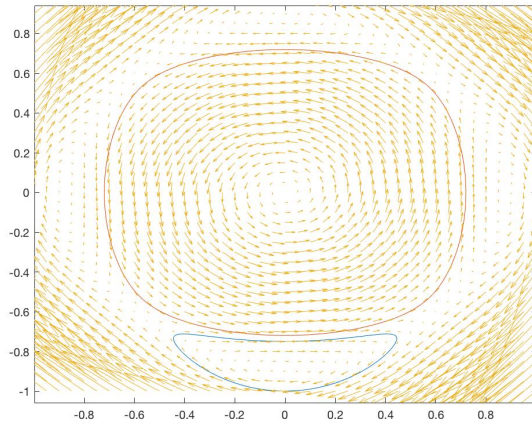


FIGURE 1. The Hamiltonian vector field defined by the Trott curve and two of its trajectories.

Example 2.3 ($n = 2$). The *Trott curve* is the quartic in the plane \mathbb{R}^2 defined by

$$h(x, y) = 144(x^4 + y^4) - 225(x^2 + y^2) + 350x^2y^2 + 81.$$

This curve consists of four nonconvex ovals. Hence it has 28 real bitangents.

The vector field for the Hamiltonian system (2.3) is shown in Figure 1, along with two of its trajectories. Fix any point (u, v) in \mathbb{R}^2 and set $c = h(u, v)$. Then the trajectory of (2.3) that starts at (u, v) travels on the quartic curve defined by $h(x, y) = c$. Consider the starting point $(0, -1)$, which lies on the original Trott curve $h(x, y) = 0$. Its trajectory is one of the four ovals, namely the oval at the bottom that is red in Figure 2 (left) and blue in Figure 1.

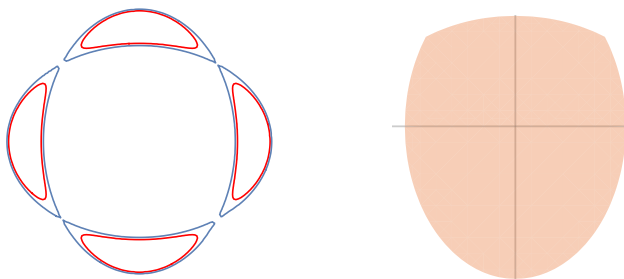


FIGURE 2. A pair of ellipses encloses the Trott curve and bounds the attainable region.

The region bounded by that oval is not convex. Its convex hull is the convex trajectory. It has one bitangent edge, namely the segment from $(-a, b)$ to (a, b) ,

where

$$a = 0.4052937596229429488 \quad \text{and} \quad b = -0.7125251813139792270.$$

This convex trajectory is not forward closed. This can be seen by taking any starting point (x, b) where $0 < x < 0.239173943$. The resulting trajectory is a convex curve that lies above the original oval. It is shown in red in Figure 1. The set of all trajectories as x ranges from 0 to a sweeps out the attainable region for $(0, -1)$. This region is a semialgebraic set. Its boundary consists of parts of two ellipses. Their union is the zero set of $h(x, y) - \frac{1053}{638}$. The containment of the Trott curve in the ellipse is shown on the left in Figure 2. The attainable region of the lower oval in the Trott curve is the convex set of the right in Figure 2.

3. LIMITING FACES IN POLYHEDRAL APPROXIMATIONS

We now study the limiting behavior of the facets of a sequence of polytopes. Each polytope is the convex hull of points sampled on a smooth, compact curve \mathcal{C} in \mathbb{R}^n whose convex hull $C = \text{conv}(\mathcal{C})$ has dimension n . Our aim is to derive information about faces of C from the facets of its polyhedral approximations. Even though there are lots of profound results on polytopal approximation of convex bodies (see e.g. [4] for a survey), to the best of our knowledge there is no study that gives results similar to the ones presented here. In addition to the approximation of C in terms of Hausdorff distance, in the limit we also approach the special facial structure of the convex hull C of the curve. Our approximation method, which is based on points sampled on that curve, accounts for this. We will elaborate on the structure of ∂C in Section 5. Here we establish a theoretical basis for our algorithms.

We wish to compute the boundary of C using a sequence of inner approximations by convex polytopes, each obtained as the convex hull of a path that approximates \mathcal{C} . The *Hausdorff distance* of two compact sets B_1 and B_2 in \mathbb{R}^n is defined as

$$d(B_1, B_2) = \max \left\{ \max_{x \in B_1} \min_{y \in B_2} \|x - y\|, \max_{y \in B_2} \min_{x \in B_1} \|x - y\| \right\}.$$

A sequence $\{B_\nu\}_{\nu \in \mathbb{N}}$ of compact sets is *Hausdorff convergent* to a fixed compact set B if $d(B, B_\nu) \rightarrow 0$ for $\nu \rightarrow \infty$.

A point x of a compact convex set B is *extremal* if it is not a proper convex combination of elements of B , that is, $\{x\}$ is a zero-dimensional face of B . Extremal points of a polytope are called *vertices*. Even if $\{B_\nu\}_{\nu \in \mathbb{N}}$ is a sequence of polytopes that Hausdorff converges to a polytope B , the limit of a convergent sequence of vertices x_ν of B_ν is not necessarily a vertex of B . For instance, consider $B_\nu = \text{conv}(\{(0, 0), (1, \frac{1}{\nu}), (2, 0)\})$. However, the converse holds.

Lemma 3.1. *Let $\{B_\nu\}_{\nu \in \mathbb{N}} \rightarrow B$ be a Hausdorff convergent sequence of compact convex sets in \mathbb{R}^n . For every extremal point x of B there exist extremal points x_ν of B_ν converging to x .*

Proof. Let x be an extremal point of the limit body B . By Hausdorff convergence, there exists a sequence $\{x_\nu\}_{\nu \in \mathbb{N}}$ with $x_\nu \in B_\nu$ that converges to x . By

Carathéodory's Theorem, each x_ν is a convex combination of at most $n + 1$ extremal points $v_0^\nu, v_1^\nu, \dots, v_n^\nu$ of B_ν .

Fix some $\varepsilon > 0$. Assume there is an infinite subset N of \mathbb{N} such that

$$\forall \nu \in N \ \forall i \in \{0, 1, \dots, n\} : \|x - v_i^\nu\| \geq \varepsilon.$$

By compactness, there is a subsequence N' of N such that, for each i , the sequence $\{v_i^\nu\}_{\nu \in N'}$ converges to some v_j . Hence x is a proper convex combination of $v_0, v_1, \dots, v_n \in B$. This contradicts x being extremal in B . Therefore, for every $\varepsilon > 0$ there exists $\nu_0 \in \mathbb{N}$ such that

$$\forall \nu \geq \nu_0 \ \exists i \in \{0, 1, \dots, n\} : \|v_i^\nu - x\| < \varepsilon.$$

We thus obtain a sequence $\{x_\nu\}$ of extremal points x_ν of B_ν converging to x . \square

An ε -approximation of the given curve \mathcal{C} is a finite subset $\mathcal{A}_\varepsilon \subset \mathcal{C}$ such that

$$\forall y \in \mathcal{C} \ \exists x \in \mathcal{A}_\varepsilon : \|y - x\| \leq \varepsilon.$$

We consider a sequence $\{\mathcal{A}_\varepsilon\}_{\varepsilon \searrow 0}$ of ε -approximations, where $\varepsilon \searrow 0$ stands for a decreasing sequence $\{\varepsilon_\nu\}_{\nu \in \mathbb{N}}$ of positive real numbers ε_ν . The polytopes $A_\varepsilon = \text{conv}(\mathcal{A}_\varepsilon)$ can be described by their facets. Our goal is to study convergent sequences of facets F_ε of A_ε in order to get information about the facial structure of the convex hull C of the curve \mathcal{C} .

Proposition 3.2. *Let $\{F_\varepsilon\}_{\varepsilon \searrow 0}$ be a Hausdorff convergent sequence of proper faces F_ε of the polytopes A_ε . Then its limit F is contained in an exposed face of C .*

Proof. We write the face F_ε of the polytope A_ε in the form $F_\varepsilon = \{x \in A_\varepsilon \mid y_\varepsilon^T x = \gamma_\varepsilon\}$, where $\|y_\varepsilon\| = 1$ and $y_\varepsilon^T x \leq \gamma_\varepsilon$ for $x \in \mathcal{A}_\varepsilon$. Since \mathcal{C} is compact, the sequence $\{\gamma_\varepsilon\}$ is bounded. Choose accumulation points y and γ , respectively, of $\{y_\varepsilon\}_{\varepsilon > 0}$ and $\{\gamma_\varepsilon\}_{\varepsilon > 0}$. Since $y \neq 0$, $H := \{x \in \mathbb{R}^n \mid y^T x = \gamma\}$ is a hyperplane. By Lemma 3.1, any extremal point x of F is the limit of a sequence $\{x_\varepsilon\}_{\varepsilon \searrow 0}$ for x_ε a vertex of F_ε . Every vertex of F_ε belongs to \mathcal{C} . Thus F is contained in $C \cap H$. It remains to show that C is contained in the halfspace $H_- := \{x \in \mathbb{R}^n \mid y^T x \leq \gamma\}$. Assume there exists $x \in \mathcal{C}$ with $d := y^T x - \gamma > 0$. Then there exists a sequence $\{x_\varepsilon\}$ with $x_\varepsilon \in \mathcal{A}_\varepsilon$ converging to x and such that $\{y_\varepsilon^T x_\varepsilon - \gamma_\varepsilon\}$ converges to d . This contradicts that the halfspace $\{x \in \mathbb{R}^n \mid y_\varepsilon^T x \leq \gamma_\varepsilon\}$ contains A_ε . \square

The limit F in Proposition 3.2 may not be a face of C . This is shown in Figure 3 on the left. The following genericity assumptions on \mathcal{C} will ensure that a Hausdorff convergent sequence of proper faces F_ε of the polytopes A_ε converges to a proper face F of the body C :

- (H1) Every point on the curve \mathcal{C} that is in the boundary of C is an extremal point of C .
- (H2) Every polytope face of C is a simplex.
- (H3) Intersecting the curve \mathcal{C} with a hyperplane always results in a finite set.

We now give a sufficient condition that proper faces of C are polytopes.

Proposition 3.3. *If \mathcal{C} satisfies (H3), then every proper face of C is a polytope.*

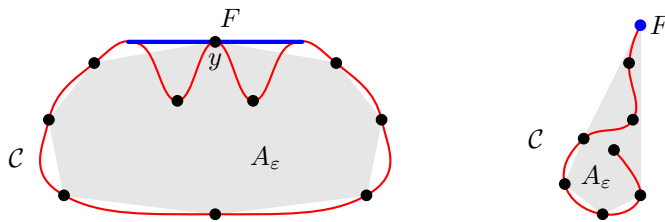


FIGURE 3. A Hausdorff convergent sequence of facets F_ε of A_ε need not converge to a face F of C . The face F on the left contains a curve point $y \in C$ which is not extremal in C . The endpoint F of the curve C on the right is an exposed face of C but it is not uniquely exposed. There is no sequence of facets F_ε of A_ε that Hausdorff converges to that face F .

Proof. A proper face F of C belongs to some hyperplane. By (H3), the set $C \cap F$ is finite. Since F is a face of C , an extremal point of F is also an extremal point of C . All extremal points of C belong to C , since they cannot be expressed as a proper convex combination of curve points. Thus, F is a polytope as it has only finitely many extremal points. \square

Proposition 3.4. *Suppose that (H1), (H2) and (H3) hold. Let $\{F_\varepsilon\}_{\varepsilon \searrow 0}$ be a Hausdorff convergent sequence of proper faces F_ε of A_ε . Then its limit F is a proper face of C .*

Proof. By Proposition 3.2, F is contained in an exposed face G of C ; in particular, in a proper face of C . Let G be the smallest face of C containing F . By assumption (H3) and Proposition 3.3, G is a polytope. The extremal points of F belong to C . The subset F of G is in the boundary of C . By assumption (H1), the extremal points of F are extremal points of C . Since $F \subset G \subset C$, they are also extremal points of G . Thus either $F = G$ or F is a sub-simplex of G by assumption (H2). The latter case contradicts the minimality of G . \square

Let F be a proper face of C . We seek a sequence F_ε of facets of A_ε that are Hausdorff converging to F . In general, such a sequence does not exist, even under the assumptions (H1), (H2), (H3). We need to additionally require the face F to be *uniquely exposed*, that is, there is a unique halfspace H^+ with $C \subset H^+$ and $F = C \cap -H^+$. For an example see Figure 3 (right).

Theorem 3.5. *Assume (H1) and let F be a simplex which is a uniquely exposed face of C . Then F is the Hausdorff limit of a sequence $\{F_\varepsilon\}_{\varepsilon \searrow 0}$ of facets of A_ε .*

Proof. Let v_0, \dots, v_k be the vertices of F . Without loss of generality, $\frac{1}{k+1} \sum_{i=0}^k v_i = 0$. The halfspace which defines F in C has the form $H_\gamma^+ := \{x \in \mathbb{R}^n \mid h^T x \geq \gamma\}$, where $\gamma = 0$ and $h \in \mathbb{R}^n$ with $\|h\| = 1$ is unique. We claim that for every $\delta > 0$

there exists $\gamma > 0$ such that

$$\forall y \in \mathcal{C} \setminus H_\gamma^+ \exists i \in \{0, 1, \dots, k\} : \|y - v_i\| < \delta. \tag{3.1}$$

We prove this by contradiction. Suppose there exists $\delta > 0$ such that for all $\gamma > 0$ there exists $y \in \mathcal{C} \setminus H_\gamma^+$ with $\|y - v_i\| \geq \delta$ for all vertices v_i of F . Thus we can construct a sequence of curve points approaching $-H^+ = -H_0^+$ but maintain a distance of at least δ from each v_i . By compactness of \mathcal{C} , this sequence can be assumed to converge to some $z \in -H^+ \cap \mathcal{C} \subset F$. Since the curve point z belongs to the boundary of C , assumption (H1) ensures that z is an extremal point of C . Hence, z is a vertex of F different from v_0, \dots, v_k , a contradiction.

For $\varepsilon > 0$, we consider the linear program

$$\min \mu \quad \text{s.t.} \quad \mu h \in A_\varepsilon. \tag{3.2}$$

We claim that the following holds for sufficiently small $\varepsilon > 0$:

$$\text{span } h \cap \text{int } A_\varepsilon \neq \emptyset. \tag{3.3}$$

Assume the contrary. Then, for all $\varepsilon > 0$, $\text{span } h$ and A_ε can be separated weakly by a hyperplane $H(\varepsilon) = \{x \in \mathbb{R}^n \mid h_\varepsilon^T x = \gamma_\varepsilon\}$ with $\|h_\varepsilon\| = 1$. By a compactness argument, a subsequence of $\{(h_\varepsilon, \gamma_\varepsilon)\}_{\varepsilon \searrow 0}$ converges to $(\bar{h}, \bar{\gamma})$ with $\|\bar{h}\| = 1$. The hyperplane $\bar{H} = \{x \in \mathbb{R}^n \mid \bar{h}^T x = \bar{\gamma}\}$ weakly separates $\text{span } h$ and C . Since 0 is contained in both C and $\text{span } h$, we have $\bar{\gamma} = 0$. Since 0 is a relative interior point of F , we must have $F \subset \bar{H}$. Hence, F is exposed with respect to a halfspace $\bar{H}^+ := \{x \in \mathbb{R}^n \mid \bar{h}^T x \geq \bar{\gamma}\}$ corresponding to \bar{H} . Since $H^+ \neq \bar{H}^+$, this contradicts the assumption that F is uniquely exposed with respect to H^+ .

Let μ_ε be the optimal value of (3.2). We have $\{\mu_\varepsilon\}_{\varepsilon \searrow 0} = 0$. We will use linear programming duality to show that, for $\varepsilon > 0$ sufficiently small, $\mu_\varepsilon h$ belongs to a facet of the form $F_\varepsilon = A_\varepsilon \cap \{x \in \mathbb{R}^n \mid y_\varepsilon^T x = \mu_\varepsilon\}$, where $h^T y_\varepsilon = 1$. Let M_ε be the matrix with columns \mathcal{A}_ε and $e = (1, \dots, 1)^T$. The linear program dual to (3.2) is

$$\max \eta \quad \text{s.t.} \quad M_\varepsilon^T y - e\eta \geq 0 \quad \text{and} \quad h^T y = 1. \tag{3.4}$$

Let $(y_\varepsilon, \eta_\varepsilon)$ denote an optimal solution of (3.4). By duality, $\mu_\varepsilon = \eta_\varepsilon$. We conclude that the set F_ε is a face of the polytope A_ε .

To see that F_ε is a facet of A_ε , we replace (3.2) and (3.4) by the pair of dual problems

$$\min 0\mu \quad \text{s.t.} \quad \mu h \in A_\varepsilon, \tag{3.5}$$

$$\max \eta \quad \text{s.t.} \quad M_\varepsilon^T y - e\eta \geq 0 \quad \text{and} \quad h^T y = 0. \tag{3.6}$$

Using complementary slackness, we conclude from (3.3) that $(y, \eta) = (0, 0)$ is the unique optimal solution of (3.6). Hence the set of optimal solutions of (3.4) is bounded, and we can choose $(y_\varepsilon, \eta_\varepsilon)$ to be a vertex. At least n linearly independent inequalities in (3.4) hold with equality at $(y, \eta) = (y_\varepsilon, \eta_\varepsilon)$. These correspond to n affinely independent points in \mathcal{A}_ε , all belonging to the hyperplane $\{x \in \mathbb{R}^n \mid y_\varepsilon^T x = \mu_\varepsilon\}$. This shows that F_ε is a facet of A_ε .

From (3.1), we conclude that, for sufficiently small $\varepsilon > 0$, the point $\mu_\varepsilon h \in F_\varepsilon$ (which approaches the mean of the vertices of F) can be represented only by elements $y \in \mathcal{A}_\varepsilon$ with $\|y - v\| < \delta$ for some vertex v of F . Since F is a simplex,

each vertex v of F is used in this representation. Hence, for each vertex v of F there exists a vertex y of F_ε with $\|y - v\| < \delta$.

We claim that if $\varepsilon > 0$ is sufficiently small then for every vertex y of F_ε there exists a vertex v of F with $\|y - v\| < \delta$. Assume the contrary. Then, by (3.1), for any small $\varepsilon > 0$, there is a vertex y_ε of F_ε with $y_\varepsilon \in \mathcal{C} \cap H_\gamma^+$. By compactness of \mathcal{C} , we may assume that $\{y_\varepsilon\}_{\varepsilon \searrow 0}$ converges to some $\bar{y} \in \mathcal{C} \cap H_\gamma^+$. By Proposition 3.2, $\text{conv}(F \cup \{\bar{y}\})$ belongs to an exposed face of C . Since $\bar{y} \notin -H^+$, this contradicts the assumption that F is uniquely exposed. We conclude that for every $\delta > 0$ we find $\varepsilon_0 > 0$ such that $d(F_\varepsilon, F) < \delta$ for all $0 < \varepsilon \leq \varepsilon_0$. Hence, the simplex face F of C is the Hausdorff limit of the sequence $\{F_\varepsilon\}_{\varepsilon \searrow 0}$, as desired. \square

Remark 3.6. In the proof of Theorem 3.5, we construct a sequence $\{F_\varepsilon\}_{\varepsilon \searrow 0}$ of facets of A_ε whose vertices converge to the vertices of F . This is stronger than Hausdorff convergence.

Remark 3.7. In our proofs we had assumed, for simplicity, that the curve \mathcal{C} is smooth and that the sampled points lie exactly on \mathcal{C} . Our results should extend to nonsmooth curves and to points that are sampled around \mathcal{C} with a given accuracy. The investigation of polyhedral approximations of $C = \text{conv}(\mathcal{C})$ under such weaker assumptions is left for future work.

4. CONVEX HULLS IN BENSOLVE

The key step in our solution to Problem (i) is the computation of the convex hull of an ε -approximation of a curve \mathcal{C} . There are many methods and implementations for convex hulls. For this paper, the software `Bensolve Tools` [5] was used. It is based on Benson's algorithm; see e.g. [8]. One reason for that choice is the output sensitivity of the underlying method. This means that the runtime is mainly dependent on the number of facets and vertices of the polytope that is the convex hull and its dimension. In particular, the number of sampled points in the interior of $C_{\mathcal{A}}$ only marginally influences the computation time. Another advantage of Benson's algorithm is the possibility to set the parameter ε in Algorithm 4.4. This feature enables the approximative representation of highly complex convex hulls in a reasonable amount of time. In addition, the process can be aborted at any point while still providing an outer approximation. For small values of ε , we obtain exact solutions, up to numerical inaccuracy of the vertex enumeration routine.

We next discuss this software, its underlying methodology, and how we apply it. `Bensolve` [17] is a solver for multiple objective linear programs (MOLP). In `Bensolve Tools` it is utilized to perform many polyhedral calculus operations, among them computing the convex hull. The key insight behind this is that multiple objective linear programming is equivalent to polyhedral projection [16]. Convex hull computation is a special case of polyhedral projection. This follows from standard facts in Ziegler's textbook [22, Chapter 1]. We state it as follows:

Lemma 4.1. *The convex hull of $\mathcal{V} = \{v^1, v^2, \dots, v^k\}$ in \mathbb{R}^n is the polytope*

$$\text{conv}(\mathcal{V}) = \{y \in \mathbb{R}^n \mid \exists \lambda \in \mathbb{R}^k : \lambda \geq 0, e^T \lambda = 1, y = V\lambda\},$$

where $V \in \mathbb{R}^{n \times k}$ is the matrix with set of columns \mathcal{V} and $e = (1, 1, \dots, 1)^T$. Hence, the convex hull of \mathcal{V} is a projection into \mathbb{R}^n of the polytope

$$Q = \{(y, \lambda) \in \mathbb{R}^n \times \mathbb{R}^k \mid \lambda \geq 0, e^T \lambda = 1, y = V\lambda\}.$$

To understand the computation of $\text{conv}(\mathcal{V})$ from Q , let us turn to an arbitrary polyhedral projection problem. By Fourier–Motzkin elimination, every linear projection of a polyhedron is a polyhedron. This leads to the concept of a *P-representation* of a polyhedron. Let $M \in \mathbb{R}^{n \times k}$, $B \in \mathbb{R}^{m \times k}$, $a \in \mathbb{R}^m$ be given. The triple (M, B, a) represents the polyhedron

$$P = \{Mx \mid Bx \geq a\} = \{y \in \mathbb{R}^n \mid \exists x \in \mathbb{R}^k : y = Mx, Bx \geq a\}. \tag{4.1}$$

In what follows, we restrict to polytopes (bounded polyhedra). Given a P-representation (4.1) of a polytope, the polyhedral projection problem is to compute an irredundant *V-representation*, i.e. a representation as convex hull of finitely many points, and an irredundant *H-representation*, i.e. a representation by finitely many linear inequalities (cf. [22]).

Given a triple (M, B, a) as above, the associated *multiple objective linear program* is

$$\min Mx \quad \text{s.t.} \quad Bx \geq a. \tag{MOLP}$$

The *upper image* of the program (MOLP) is the polyhedron

$$\mathcal{P} = \{y \in \mathbb{R}^n \mid \exists x \in \mathbb{R}^k : y \geq Mx, Bx \geq a\}. \tag{4.2}$$

A solution of (MOLP) consists of irredundant V- and H-representations of \mathcal{P} . This concept of solution can be used to address the polyhedral projection problem:

Proposition 4.2 (cf. [16, Theorem 3]). *The solution of the MOLP*

$$\min \begin{pmatrix} M \\ -e^T M \end{pmatrix} x \quad \text{s.t.} \quad Bx \geq a \tag{4.3}$$

yields an irredundant V- and H-representation of the P-represented polyhedron (4.1).

The upper image of the MOLP in (4.3) is the polyhedron

$$\bar{\mathcal{P}} = \{(y, z) \in \mathbb{R}^n \times \mathbb{R} \mid y \geq Mx, z \geq -e^T Mx, Bx \geq a\}. \tag{4.4}$$

Corollary 4.3. *The polytope $P = \{Mx \mid Bx \geq a\}$ is obtained from $\bar{\mathcal{P}}$ by setting*

$$P = \{y \in \mathbb{R}^n \mid \exists z : (y, z) \in \bar{\mathcal{P}}, e^T y + z = 0\}.$$

An irredundant V-representation of P derives from the set of vertices of $\bar{\mathcal{P}}$ by deleting their last components. An H-representation of $\bar{\mathcal{P}}$ gives an H-representation of P by adding the equation $z = -e^T y$.

Bensolve computes V- and H-representations of the upper image (4.4) using Algorithm 4.4. This is a version of *Benson’s algorithm*. It applies to upper images satisfying $\mathcal{P} \subseteq y + \mathbb{R}_{\geq 0}^n$ for some $y \in \mathbb{R}^n$. This version suffices for handling projections of polytopes including the representation of the convex hull of finitely many points. Since the algorithm is numerical, we work with a prescribed tolerance

$\varepsilon > 0$. The output is an ε -approximation to the upper hull \mathcal{P} , i.e. it is a polyhedron \mathcal{O} that is ε -close to \mathcal{P} in the sense that $\varepsilon e + \mathcal{O} \subseteq \mathcal{P} \subseteq \mathcal{O}$.

Algorithm 4.4. (Benson's algorithm)

input : (MOLP) given by the matrices M , B and vector a ; a tolerance $\varepsilon \geq 0$.

output: ε -close V-representation \mathcal{V} and H-representation \mathcal{H} of \mathcal{P} in (4.2).

```

1  $T \leftarrow \emptyset$ 
2 Compute the H-representation  $\mathcal{H}$  of an outer approximation of  $\mathcal{P}$  having the
  same recession cone as  $\mathcal{P}$ . Compute the corresponding V-representation  $\mathcal{V}$ .
3 while  $(\mathcal{V} \setminus T) \neq \emptyset$  do
4   Choose a vertex  $v \in \mathcal{V} \setminus T$ .
5   Compute the solution  $t^*$  of the linear program  $\min \{t \mid v + te \in \mathcal{P}\}$ .
6   Compute the solution  $(u^*, w^*)$  of the dual linear program
      $\max \{a^T u - v^T w \mid B^T u = M^T w, e^T w = 1, w \geq 0, u \geq 0\}$ .
7   if  $t^* \geq \varepsilon$  then
8     Refine  $\mathcal{H}$  by adding  $\{y \mid (w^*)^T y \geq a^T u^*\}$  to the description.
9     Update  $\mathcal{V}$  by performing vertex enumeration on  $\mathcal{H}$ .
10  else
11     $T \leftarrow T \cup \{v\}$ 
12  end
13 end

```

One starts with an initial outer polyhedral approximation of \mathcal{P} . Both an H-representation and a V-representation are stored. Until the tolerance ε is reached, each iteration adds a linear inequality to refine the outer approximation of \mathcal{P} . An iteration step starts by choosing a vertex v of the current polyhedron. The V-representation is updated after adding an inequality. From v one moves in direction $e = (1, \dots, 1)^T$ to the boundary point $y = v + t^*e$ of \mathcal{P} . To this end, a linear program has to be solved. The solution of the dual linear program yields the desired linear inequality which cuts off v and holds with equality in y . Algorithm 4.4 terminates and computes both V- and H-representation of an ε -approximation of \mathcal{P} . For computations in this paper we use the *dual Benson algorithm* [8]. It is dual to Benson's algorithm and provides an inner approximation for the upper hull \mathcal{P} .

We employ **Bensolve** for obtaining a polyhedral approximation of the convex hull of a smooth curve \mathcal{C} in \mathbb{R}^n . This is done by computing the convex hull of a sufficiently large finite subset \mathcal{A} of \mathcal{C} . The output gives both an irredundant H- and V-representation of an inner ε -approximation $\mathcal{C}_{\mathcal{A}}$ of $\text{conv}(\mathcal{A})$, and this is our approximation to $\text{conv}(\mathcal{C})$. All facets and all vertices of $\mathcal{C}_{\mathcal{A}}$ are known after such a computation. The output also contains the *incidence matrix* $I_{\mathcal{A}}$ for facets and vertices of $\mathcal{C}_{\mathcal{A}}$ and the *adjacency matrix* $A_{\mathcal{A}}$ for vertices of $\mathcal{C}_{\mathcal{A}}$.

Example 4.5. Let \mathcal{C} be the trigonometric space curve parametrically given by

$$\theta \mapsto (\cos(\theta), \sin(2\theta), \cos(3\theta)). \quad (4.5)$$

Its convex hull $C = \text{conv}(\mathcal{C})$ is shown in [19, Figure 1]. We select the sample points

$$\mathcal{A} = \left\{ \left(\cos \left(\frac{2k\pi}{N} \right), \sin \left(\frac{4k\pi}{N} \right), \cos \left(\frac{6k\pi}{N} \right) \right) \mid k = 0, \dots, N - 1 \right\}. \quad (4.6)$$

Using `Bensolve`, as described above, we can compute irredundant V- and H-representations of the inner approximation $C_{\mathcal{A}}$ of the polytope $\text{conv}(\mathcal{A})$ for various values of N and with specified accuracy ε . For instance, let $N = 100$ and $\varepsilon = 10^{-9}$. The sample (4.6) is shown on the left in Figure 4. Its convex hull is the polytope $C_{\mathcal{A}}$ on the right in Figure 4. It has 70 vertices and 102 facets, so, by Euler’s relation, it has 170 edges. Thus the incidence matrix $I_{\mathcal{A}}$ is of size 102×70 and has 340 nonzero entries. The 70×70 adjacency matrix $A_{\mathcal{A}}$ also has 340 nonzero entries. The polytope $C_{\mathcal{A}}$ in Figure 4 already looks like [19, Figure 1] and Figure 6. The picture of $C_{\mathcal{A}}$ reveals the edge surfaces of C and the two triangles in ∂C . The identification of such patches from the `Bensolve` output is our theme in Section 5.

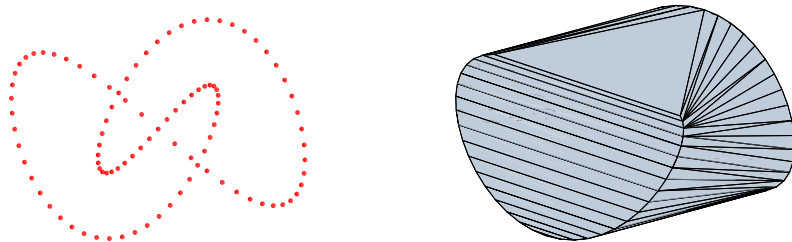


FIGURE 4. A sample of points (left) from a space curve and its convex hull (right).

5. DETECTION OF PATCHES

Let C be a full-dimensional compact convex body in \mathbb{R}^n . The boundary of ∂C is an $(n - 1)$ -dimensional set whose subset ∂C_{sm} of smooth points is dense. We shall stratify ∂C_{sm} into finitely many manifolds we call *patches*. Each patch is an $(n - k - 1)$ -dimensional family of k -faces of C . For a typical convex body of dimension $n = 3$, the boundary is comprised of surfaces of extreme points ($k = 0$), curves of edges ($k = 1$), and finitely many facets ($k = 2$). For the general definition, we use the concept of the normal cycle of a convex body. Let \mathbb{S}^{n-1} denote the unit $(n - 1)$ -sphere. Following [10, eqn (10)], the *normal cycle* of C equals

$$N(C) = \{ (u, v) \in \mathbb{R}^n \times \mathbb{S}^{n-1} : v \cdot (u - u') \geq 0 \text{ for all } u' \in C \}.$$

If ∂C is smooth then $N(C)$ is a Legendrian submanifold of dimension $n - 1$. If C is not smooth then we can approximate C by nearby smooth convex bodies C_ε , for $\varepsilon > 0$. By [10, Theorem 3.1], the normal cycle $N(C)$ is the Hausdorff limit of the manifolds $N(C_\varepsilon)$ for $\varepsilon \rightarrow 0$. The normal cycle $N(C)$ is pure $(n - 1)$ -dimensional, and its smooth points are dense.

There are several other ways of defining the normal cycle. The one we like best uses the dual convex body C^\vee . Assuming that the origin is in the interior of C ,

$$N(C) = \{(u, v) \in \partial C \times \partial C^\vee : v \cdot (u - u') \geq 0 \text{ for all } u' \in C\}. \quad (5.1)$$

The normal cycle comes naturally with two surjective maps

$$\pi_1 : N(C) \rightarrow \partial C, (u, v) \mapsto u \quad \text{and} \quad \pi_2 : N(C) \rightarrow \partial C^\vee, (u, v) \mapsto v.$$

Let $\mathcal{E} \subseteq \partial C^\vee$ be the set of exposed points of C^\vee . We have $v \in \mathcal{E}$ if and only if there exists $u \in C$ such that $\pi_1^{-1}(u) = \{(u, v)\}$. A subset ψ of $N(C)$ with $\pi_2(\psi) \subset \mathcal{E}$ is called a *patch* if ψ is a connected differentiable manifold, $\dim(\pi_1(\psi)) = n - 1$, the fibers of π_2 vary continuously in the Hausdorff metric, and ψ is maximal with these properties. We say that ψ is a k -patch if $\dim(\pi_2(\psi)) = n - k - 1$. This means that $\pi_2(\psi)$ is an $(n - k - 1)$ -dimensional manifold of exposed points of C^\vee , and these exposed points support continuously varying k -faces of C .

Remark 5.1. If the trajectory \mathcal{C} is algebraic then its convex hull C is semialgebraic. Also the normal cycle $N(C)$ and all its patches ψ are semialgebraic. This follows from Tarski's theorem on quantifier elimination, and we find that the number of patches of C is finite. We believe that finiteness holds more generally for compact trajectories. But we do not yet know the precise statement. Real analytic geometry is much more delicate than real algebraic geometry. For instance, the family of semianalytic sets is not closed under projection. We refer to [1] for a recent account. The concept of \mathbb{C} -semianalytic sets, introduced in [1] and named after Cartan, might be appropriate for our setting. One can hope that the convex trajectories and their patches are \mathbb{C} -semianalytic when ϕ in (1.1) is polynomial.

We now study the following computational problem. The input is a smooth curve \mathcal{C} in \mathbb{R}^n , typically arising as trajectory of a dynamical system (1.1). We seek the boundary of the convex trajectory $C = \text{conv}(\mathcal{C})$. The output is the list of all patches.

Example 5.2 ($n = 3$). The convex body in Examples 4.5 and 6.2 has six patches. It has two 2-patches, namely the two triangles. It has two irreducible edge surfaces (cf. [19]). These have degrees 3 and 16. Each contributes two 1-patches to ∂C . All six patches are visible from the edges and triangles of the polytope in Figure 4.

Example 5.3 ($n = 4$). If C is the convex hull of a curve \mathcal{C} in \mathbb{R}^4 then its 1-patches are surfaces of edges, its 2-patches are curves of 2-faces, and its 3-patches are the facets of C .

Our goal is to identify all patches of $C = \text{conv}(\mathcal{C})$ from ε -approximations, using the results in Section 3. We assume that \mathcal{C} is a simplicial curve, by which we mean that it satisfies the hypotheses (H1), (H2) and (H3), and the number of patches of C is finite (cf. Remark 5.1).

The special case $n = 2$ is solved by Algorithm 2.1. Algorithm 5.4 computes the patches for $n \geq 3$. We implemented this algorithm for $n = 3$ and $n = 4$. A detailed theoretical analysis of this algorithm is left for future work. The task is

to identify the precise conditions under which the output detects the true patches when applied to ε -approximations with $\varepsilon \rightarrow 0$.

In the next section we report on some experiments with our methods. The code for dimensions 2, 3 and 4 is made available at

<http://tools.bensolve.org/trajectories>.

Algorithm 5.4. (Detection of patches for $n \geq 3$)

- input** : Finite list \mathcal{A} of points on a curve C in \mathbb{R}^n ; a threshold value $\delta > 0$.
- output**: For each $k \geq 1$: the expected number $\#_k$ of k -patches of $C = \text{conv}(C)$.
 For each i : list of k -polytopes that represent the k -patch G_i .
- 1 Compute vertices \mathcal{V} , facets \mathcal{H} , incidence list $I_{\mathcal{A}}$ and adjacency list $A_{\mathcal{A}}$ of $\text{conv}(\mathcal{A})$.
- 2 Build a graph G with node set \mathcal{H} as follows: two facets H_1, H_2 form an edge if their unit normals $v(H_i)$ have distance $\leq \delta$, $\dim(H_1 \cap H_2) = n - 2$, and $d(H_1, H_2) \leq \delta$.
- 3 **foreach** *connected component* G_i *of the graph* G **do**
- 4 **foreach** *facet* $H \in G_i$ **do**
- 5 Find representatives $U = \{u_0, \dots, u_k\}$ of the δ -proximity clusters
- 6 of vertices of H such that $F = \text{conv}(U)$ is a k -face of $\text{conv}(\mathcal{A})$.
- 7 Associate the tuple $(u_0, \dots, u_k; v)$ with that node of G_i .
- 8 **end**
- 9 G_i represents a k -patch of C if k is the largest index encountered in the loop above.
- 10 Adjust all tuples with smaller indices found in step 7 to that common value of k .
- 11 **end**
- 12 Output $(\#_1, \dots, \#_{n-1})$, where $\#_k$ is the number of graphs G_i representing k -patches.

We discuss the steps in Algorithm 5.4. Step 1 is executed using `Bensolve` as discussed in Section 4. Each facet H comes with its unit normal vector $v(H)$. Step 2 reflects the conditions in the definition of patches. For instance, the criterion $d(H_1, H_2) \leq \delta$, stating that H_1 and H_2 are close in Hausdorff distance, reflects the continuous variation of k -faces. The exposed points in $\pi_2(\psi) \subset \mathcal{E}$ are represented by the vectors $v(H_i)$. The requirement that they are δ -close along the edges of G is our discrete version of the smoothness of ψ . In step 3 we identify the connected components of G , and these represent the patches of C .

The inner loop in steps 5–7 reflects our results in Section 3. By Theorem 3.5, every k -face of C is approximated by a facet $H \in G$. Here, for each vertex of the k -face, the algorithm chooses a nearby vertex u_i of H . In the loop between steps 4 and 8, it can happen that a δ -proximity cluster corresponds to more than one vertex of the k -face. This happens for k -faces with an edge shorter than δ . For that reason, we take the maximum in step 9.

Step 10 is very important and requires some explanation. In a connected component of G , some facets might be misclassified: they represent k -faces of C , but

step 5 identifies less than $k + 1$ proximity clusters at the fixed tolerance level δ . For such facets, step 10 adds additional points u_i from an existing cluster to get up to the correct value of k for that patch.

Example 5.5 ($n = 4$). We illustrate the output of Algorithm 5.4 when \mathcal{C} is a random trigonometric curve of degree six in \mathbb{R}^4 , as in Section 6, and $\mathcal{A} \subset \mathcal{C}$ is a finite approximation. Figure 5 shows the graph G . Each node in G is a face of the polytope $\text{conv}(\mathcal{A})$. We find $\#_3 = 0$. There are $\#_1 = 3$ patches for $k = 1$, represented by the three connected components of G on the right in Figure 5. These three connected graphs encode surfaces worth of edges. The number of patches for $k = 2$ is $\#_2 = 2$. These two components of G are shown on the left in Figure 5. Each node represents a triangle face of $\text{conv}(\mathcal{C})$. So, the picture on the left shows two curves of triangle faces in the boundary of our 4-dimensional convex body.

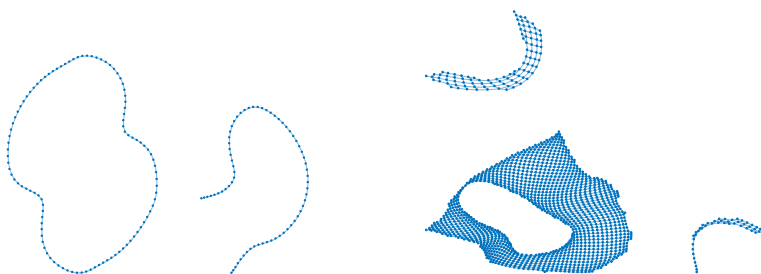


FIGURE 5. Two 2-patches (left) and three 1-patches (right) in the boundary of a 4-dimensional convex body. It is the convex hull of a trigonometric curve of degree six. The picture shows the graph G , with five connected components G_i , found by Algorithm 5.4.

6. ALGEBRAIC AND TRIGONOMETRIC CURVES

In what follows we note that every algebraic curve can be realized locally as the trajectory of an autonomous polynomial dynamical system. This generalizes the Hamiltonian systems (2.3) seen in Section 2. Hence, the computation of the convex hull of a real algebraic curve in \mathbb{R}^n , discussed in e.g. [19], is a special case of the problem we addressed in Sections 3–5.

Let \mathcal{C} be an algebraic curve in \mathbb{R}^n and let z be a regular point on \mathcal{C} . We construct an appropriate vector field $\phi(x)$ on \mathbb{R}^n as follows. Let f_1, f_2, \dots, f_{n-1} be polynomials in $x = (x_1, x_2, \dots, x_n)$ that cut out the curve \mathcal{C} locally near its point z . Let J denote their Jacobian matrix. Thus J is the $(n - 1) \times n$ matrix whose entry in row i and column j is the partial derivative $\partial f_i / \partial x_j$. Let J_i be $(-1)^{i+1}$ times the determinant of the submatrix of J obtained by deleting the i th column. Fix the vector of polynomials $\phi = (J_1, J_2, \dots, J_n)^T$. Locally at z , the kernel of J is the line spanned by the vector ϕ . This follows from Cramer's rule, and it implies that $\phi(z)$ is a tangent vector to the curve \mathcal{C} at its point z . We are interested in the dynamics of the system $\dot{x} = \phi(x)$ when the starting point is $z \in \mathcal{C}$.

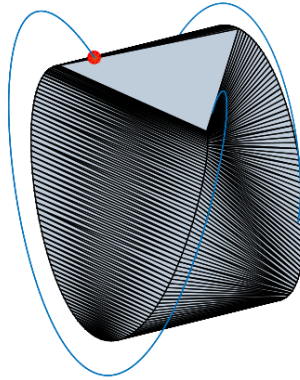


FIGURE 6. The red boundary point shows that the convex trajectory is not forward closed.

Proposition 6.1. *The trajectory of the dynamical system $\dot{x} = \phi(x)$ that starts at a point z on the algebraic curve \mathcal{C} remains on the curve \mathcal{C} . It either cycles around one nonsingular oval of \mathcal{C} , or it diverges towards infinity in \mathbb{R}^n , or it converges to a singular point of \mathcal{C} .*

Proof. The proof is analogous to Corollary 2.2 which dealt with the case $n = 2$. \square

Example 6.2 ($n = 3$). Let \mathcal{C} be the trigonometric curve (4.5) in Example 4.5. By [19, §1], this is an algebraic curve, namely it is the zero set of the two polynomials

$$f_1 = x^2 - y^2 - xz \quad \text{and} \quad f_2 = z - 4x^3 + 3x. \tag{6.1}$$

With these two polynomials we associate the dynamical system

$$\dot{x} = -2y \quad \text{and} \quad \dot{y} = 12x^3 - 5x + z \quad \text{and} \quad \dot{z} = -24x^2y + 6y.$$

Suppose we start this at a point on the curve \mathcal{C} , such as $(1, 0, 1)$. The trajectory travels on \mathcal{C} and it stops at the singular point $(0, 0, 0)$. To get $\text{conv}(\mathcal{C})$, we compute the convex hull of two trajectories obtained by using two different starting points on the curve given by (6.1). The convex body has six patches, as shown in [19, Figure 1] and in our Figures 4 and 6.

Using the methods in Section 7, we analyzed the vector field on these facets and patches, and we found points with both inward and outward pointing directions on each of them. Figure 6 shows a point in a triangle facet with outward pointing direction. The resulting trajectory is also depicted. This solves Problem (ii) from the Introduction for this example.

Trigonometric curves also arise from linear dynamical systems. Here (1.1) takes the form $\dot{x} = Ax$, where A is a real $n \times n$ -matrix. We tested our convex hull algorithms on linear systems for $n = 3, 4$. We sampled matrices A with no real eigenvalues. This ensures that the trajectories are bounded in \mathbb{R}^n . They can be

written in terms of trigonometric functions. It was shown in [14] that every convex trajectory of a linear dynamical system is forward-closed. Thus, computing the convex trajectory is equivalent to computing the attainable region.

Consider the *generalized moment curve*, whose convex hull was studied in [21, Theorem 1]. Let $z = (1, 0, 1, 0)$ and consider the linear dynamical system given by

$$A = 2\pi \cdot \begin{pmatrix} 0 & -p & 0 & 0 \\ p & 0 & 0 & 0 \\ 0 & 0 & 0 & -q \\ 0 & 0 & q & 0 \end{pmatrix},$$

where p and q are relatively prime positive integers. The trajectory is the curve

$$x(t) = (\cos(2\pi pt), \sin(2\pi pt), \cos(2\pi qt), \sin(2\pi qt)).$$

The curve is closed, and we can restrict to $0 \leq t < 1$. The convex hull of the curve is a 4-dimensional convex body. By [21, Theorem 1], there are no 3-dimensional faces. Assuming $p, q \geq 3$, there are two 1-patches and two 2-patches. The explicit description given by Smilansky in [21] makes this a useful test case.

Example 6.3 ($p = 3, q = 4$). The segment $\text{conv}\{x(s), x(t)\}$ is an edge if and only if

$$\frac{1}{4} < |s - t| < \frac{1}{3} \quad \text{or} \quad \frac{2}{3} < |s - t| < \frac{3}{4}.$$

In addition to this surface of edges, there are two curves of 2-faces, namely the triangles

$$\text{conv}\{x(t), x(t + \frac{1}{3}), x(t + \frac{2}{3})\} \quad \text{for } 0 \leq t < \frac{1}{3}$$

and the squares

$$\text{conv}\{x(t), x(t + \frac{1}{4}), x(t + \frac{1}{2}), x(t + \frac{3}{4})\} \quad \text{for } 0 \leq t < \frac{1}{4}.$$

These are all the exposed faces of the convex trajectory. Even though the curve is not simplicial, Algorithm 5.4 works well, and we verified Smilansky’s findings using our software.

We experimented with our **Bensolve**-based code for random trigonometric curves $x : [0, 1] \rightarrow \mathbb{R}^n$. The coordinates of x are trigonometric polynomials of the form

$$x_j(t) = \sum_{k=1}^d A_{jk} \cdot \cos(2\pi kt) + \sum_{k=1}^d B_{jk} \cdot \sin(2\pi kt) + C_j \quad \text{for } j = 1, \dots, n.$$

We write the coefficients as a pair of $n \times d$ matrices A and B together with a column vector C , all filled with real numbers. For general matrices, the resulting curve is an algebraic curve of degree $2d$ in \mathbb{R}^n . We computed many examples and recorded the features seen in the boundary. We were most interested in the maximal numbers of facets that were observed.

We sampled random data (A, B, C) and computed the convex hull of the resulting curves. For $n = 3$ we recorded the number of triangles (= 2-patches). The second row in Table 1 shows the maximal number of triangles that was observed for given degree $2d$. Each triangle spans a real tritangent plane of the curve. The third

degree $2d$	6	8	10	12	14	16	18	20	22	24	26	28
max $\#_2$	6	10	16	17	20	21	24	26	28	30	34	34
tritangents	8	80	280	672	1320	2288	3640	5440	7752	10640	14168	18400
max $\#_1$	10	16	24	26	30	32	35	38	41	44	46	50
edge surface	30	70	126	198	286	390	510	646	798	966	1150	1350

TABLE 1. Census of random trigonometric curves in 3-space.

row lists the number of complex tritangent planes for this curve, which is a generic space curve of genus 0 of degree $2d$. The edge surface of the curve is an irreducible ruled surface that defines the nonlinear part of the boundary of the convex hull. Its degree is listed in the fifth row. This surface is the Zariski closure of any of the 1-patches. The fourth row shows the maximal number of observed 1-patches. The numbers in the third and fifth rows are taken from [19, Corollary 3.1].

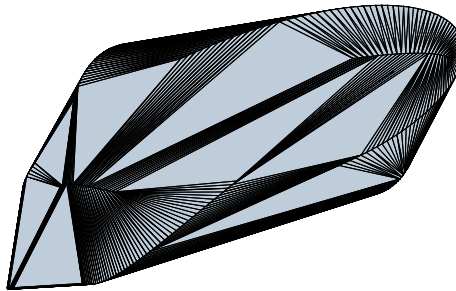


FIGURE 7. The convex hull of a trigonometric curve of degree 14 in 3-space. The boundary of this convex body consists of triangles and of 1-patches in a ruled surface of degree 286.

We illustrate our computational results in Table 1 for a curve of degree $2d = 14$.

Example 6.4 ($2d = 14$). We consider the curve defined by the 3×7 matrices

$$A = \begin{pmatrix} 0.28561 & -0.024204 & -0.07664 & 0.43593 & 0.15244 & -0.24464 & 0.41538 \\ -0.37439 & -0.30106 & 0.32118 & 0.38410 & 0.29990 & -0.14990 & -0.45481 \\ -0.17997 & -0.16046 & -0.23522 & 0.47912 & -0.08084 & 0.19628 & 0.46895 \end{pmatrix}$$

and

$$B = \begin{pmatrix} -0.39109 & 0.06742 & -0.12451 & 0.44073 & -0.20822 & -0.03646 & -0.01034 \\ 0.48646 & 0.38580 & -0.13216 & 0.36184 & 0.30633 & -0.14131 & 0.48650 \\ -0.15326 & 0.32591 & 0.02569 & 0.23351 & -0.34972 & 0.04772 & 0.42441 \end{pmatrix},$$

along with the vector $C = (0.39768 \quad 0.42346 \quad 0.23797)^T$. The convex hull of this curve has 20 triangle facets. It is shown in Figure 7. The planes that define the triangles are tritangent planes. The curve is generic and has 1320 tritangent planes

over \mathbb{C} . The nonlinear part of the boundary is the edge surface [19]. This is an irreducible ruled surface of degree 286.

7. PARTITIONING THE BOUNDARY

Problem (i) from the Introduction was addressed in the previous sections. In this section we propose a solution to Problem (ii). Our input now is the output of Algorithm 2.1 or Algorithm 5.4. If $n = 3$ then we are given all 2-patches (triangles) and all 1-patches (in the edge surface) of the convex hull $C = \text{conv}(\mathcal{C})$ of a trajectory \mathcal{C} of the dynamical system (1.1). For instance, the output of Algorithm 5.4 might be Figure 8. This is a variant of Figure 7, derived from a trigonometric curve \mathcal{C} of degree 14. Here \mathcal{C} is simplicial, and the boundary ∂C consists of 20 triangles (= 2-patches) and 30 1-patches, shown in different colors in Figure 8.

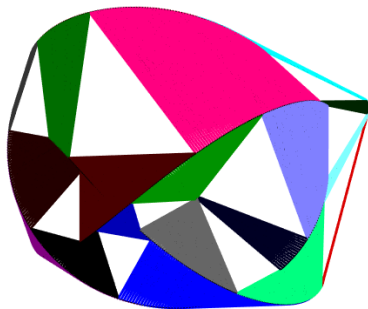


FIGURE 8. The convex hull of a trigonometric curve of degree 14.

We seek to partition the boundary ∂C into two regions. In one region, the vector $\phi(z)$ points inward and in the other it points outward, as for the red boundary point in Figure 6. For $n = 2$ this partition is determined by the formulas (2.1) and (2.2). In what follows we generalize this method to $n \geq 3$.

For a pair (u, v) in the normal cycle $N(C)$, v is an outward pointing unit normal vector at C . The right hand side of (1.1) *points inward* at $u \in \partial C$ if $\phi(u) \cdot v \leq 0$ for all $v \in \pi_2(\pi_1^{-1}(u))$. It *points outward* otherwise. Taking into account that the set $\{u \in \mathbb{R}^n \mid \pi_2(\pi_1^{-1}(u)) = \{v\}\}$ is dense in ∂C , we see that the boundary between inward and outward pointing vectors $\phi(z)$ is the image under the projection $\pi_1 : N(C) \rightarrow \partial C$ of the set $\{(u, v) \in N(C) : \phi(u) \cdot v = 0\}$. This image has dimension $n - 2$ inside the $(n - 1)$ -dimensional normal cycle $N(C)$.

Let ψ be a k -patch of C . The output of Algorithm 5.4 represents ψ by a connected graph G_i . Each node of G_i is a face $F = \text{conv}\{u_0, \dots, u_k\}$ along with a normal vector v at C . We are interested in the restriction of the boundary above to the patch ψ of interest:

$$\pi_1 \left(\{(u, v) \in \psi : \phi(u) \cdot v = 0\} \right). \quad (7.1)$$

We obtain an approximate representation of ∂C employing Algorithm 5.4. Algorithm 7.1 computes a partition of this approximation into inward and outward pointing regions. The simplex Δ_k is the convex hull of the unit vectors in \mathbb{R}^k .

Consider a fixed graph G_i in the loop started in step 1. The value of k is constant in step 2. This is ensured by step 10 in Algorithm 5.4. The graph G_i represents a k -patch of C . We algorithmically realize the restriction of the hypersurface (7.1) to the k -faces in that patch in step 4. If $k = 1$ then this results in a finite partition of a line segment. For $k = 2$ we obtain a curve in a triangle, and for $k = 3$ we obtain a surface in a tetrahedron. The latter case happens only for $n \geq 4$.

The paradigm for our computations is the algebraic case. Suppose that C is an algebraic curve, for instance obtained from a dynamical system as in Proposition 6.1. In that case, the equation $\phi(u) \cdot v = 0$ is a polynomial in k unknowns $\lambda_1, \dots, \lambda_k$, after setting $\lambda_0 = 1 - \sum_{j=1}^k \lambda_j$. To be precise, let $v = (v_1, \dots, v_n)$ be the normal vector of the k -face in question. In the situation of Proposition 6.1, the polynomial equation we are solving on Δ_k takes the form

$$\phi(u) \cdot v = \sum_{l=1}^n J_l(\lambda_0 u_0 + \dots + \lambda_k u_k) \cdot v_l = 0.$$

In some situations, we know the equation $f = 0$ of the hypersurface $\pi_1(\psi)$ in \mathbb{R}^n . Here f is analytic or polynomial, depending on the instance. With this, we write

$$v_l = \frac{\partial f}{\partial x_l}(\lambda_0 u_0 + \dots + \lambda_k u_k) \quad \text{for } l = 1, 2, \dots, n.$$

This formula allows us to solve the equation $\phi(u) \cdot v = 0$ simultaneously on the entire and exact k -patch, and not just on each approximated k -face of ψ individually, as it is done in step 4 of Algorithm 7.1.

Algorithm 7.1. (Partitioning the boundary of a convex trajectory)

input : The graphs G_i representing the patches of a convex trajectory of (1.1).
output: Partition of the boundary into inward and outward pointing regions.

```

1 foreach connected graph  $G_i$  in the output of Algorithm 5.4 do
2   foreach node  $(\{u_0, \dots, u_k\}, v)$  of the graph  $G_i$  do
3     Set  $u = \sum_{i=0}^k \lambda_j u_j$  where  $\lambda_j$  are nonnegative unknowns satisfying
4      $\sum_{j=0}^k \lambda_j = 1$ .
5     Compute the  $(k - 1)$ -dimensional hypersurface in  $\Delta_k$  defined by
6      $\phi(u) \cdot v = 0$  and identify inward and outward pointing regions.
7   end
8 end

```

Example 7.2 ($n = 3$). We partition the boundary of the convex body in Figure 6. Its edge surface has two irreducible components, of degrees 3 and 16. Each contributes two patches. The cubic is $f_2 = z - 4x^3 + 3x$ in (6.1). The degree 16 polynomial g is displayed in [19, §1]. On the cubic patches, the equation $\nabla f_2 \cdot \phi = 0$ holds identically, so these patches are not partitioned. Hence, every trajectory that

starts on a cubic patch remains in that patch. Two patches of degree 16 are partitioned by a curve of degree 262, obtained by intersecting the patches with the surface defined by $\nabla g \cdot \phi = 0$. The two triangle facets lie in the planes $z = \pm 1$. They are partitioned by the lines $y = 0$ and $x = \pm 1/2$. Figure 6 shows the trajectory that starts at a red point in the outward pointing region of the top triangle.

In the next section we apply our methods to partition the boundary of convex trajectories of dynamical systems that arise from chemical reaction networks. Figure 9 shows our partition for a convex trajectory arising in an application.

8. CHEMICAL REACTION NETWORKS

Our interest in convex trajectories and attainable regions is motivated by dynamical systems for chemical reaction networks. The aim of *attainable region theory* [18] is to design chemical reactors that are optimal for chemical reactions of interest. This research topic was pioneered by Feinberg and Hildebrand in [9]. They argued that optimal reactors are often found at the extreme points of the attainable region and showed that these extreme points are realizable by parallel operations of elementary reactor types. We refer to [14] for a recent study from the perspective we take here. Our setting is that of convex algebraic geometry [3, 19]. The notion of *protrusions* in [9, §2.6] is dual to our notion of *patches* in Section 5, in the sense that a k -patch on C corresponds to an $(n - k - 1)$ -dimensional protrusion on C^\vee under the self-duality of the normal cycle $N(C)$ in (5.1). Patches and protrusions are interesting for further research.

To explain the connection to chemical reactions, we work in the setting of *mass action kinetics*. Let G be a directed graph with m vertices, each labeled by a monomial $x^{\mathbf{a}_i}$ in n unknowns $x = (x_1, \dots, x_n)$. These unknowns are the concentrations of n chemical species. The m monomials are the chemical complexes. Each $x_j = x_j(t)$ is a function of time t . With each edge (i, j) of G , connecting two monomials $x^{\mathbf{a}_i}$ and $x^{\mathbf{a}_j}$, we associate a parameter κ_{ij} , which is the rate constant for that reaction. The associated dynamical system is given by

$$\phi(x) = (x^{\mathbf{a}_1}, x^{\mathbf{a}_2}, \dots, x^{\mathbf{a}_m}) \cdot \Lambda_G(\kappa) \cdot (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m)^T. \quad (8.1)$$

This is a row vector of length n , written as a product of three matrices, of formats $1 \times m$, $m \times m$, and $m \times n$. The middle matrix $\Lambda_G(\kappa)$ is the *Laplacian* of the graph G , with entry κ_{ij} for each edge (i, j) , all other off-diagonal entries set to zero, and diagonal entries inferred so that the row sums of $\Lambda_G(\kappa)$ are zero. An important special case of (8.1) are the toric dynamical systems [6]. For details see the forthcoming book by Dickenstein and Feliu [7] and Shiu's dissertation [20, §1.3].

A feature of many chemical reaction systems (8.1) is the existence of *conservation relations*. These arise when the entries of the polynomial vector $\phi(x)$ are linearly dependent over \mathbb{R} . If this happens then all trajectories lie in certain lower-dimensional subspaces of \mathbb{R}^n . In such cases, the ambient dimension n can be reduced. Namely, we always transform our dynamical systems so that each trajectory affinely spans \mathbb{R}^n . We examined some important classes with $n \leq 4$. Of

particular interest are chemical reaction networks that admit multistationarity. The smallest such networks were characterized by Joshi and Shiu [13].

Given an arbitrary polynomial dynamical system (1.1), it is natural to ask whether it arises from some chemical reaction network G . The solution to this inverse problem was given by Hárs and Tóth [11]. They showed that $\phi = (\phi_1, \dots, \phi_n)$ is realized by a graph G as above if and only if each monomial with negative coefficient in ϕ_i is divisible by x_i , for $i = 1, 2, \dots, n$.

We discussed Hamiltonian systems in Section 2. The following result characterizes chemical reaction dynamics in \mathbb{R}^2 that is Hamiltonian. It would be interesting to study such reaction networks, along with the higher-dimensional versions arising from Proposition 6.1.

Proposition 8.1. *Let $h(x, y)$ be a polynomial. The Hamiltonian system (2.3) can be realized as a mass action system (8.1) if and only if the coefficients of all powers of y in $h(x, y)$ are nonnegative and those of all pure powers of x are nonpositive, i.e.*

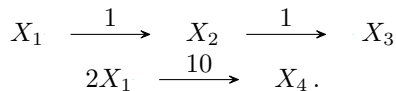
$$h(x, y) = xy \cdot a(x, y) - b(x) + c(y),$$

where b and c have nonnegative coefficients.

Proof. This is immediate from Theorem 3.2 in [11]. □

The convex trajectory we compute as an answer to Problem (i) in the Introduction is a first approximation to the attainable region and its representation in [9]. In Section 7 we presented an algorithm for partitioning the approximated boundary of the convex trajectory. We next apply that algorithm to two interesting chemical reaction networks.

Example 8.2 ($n = 4, m = 5$). We revisit the *Van de Vusse reaction*. This is studied extensively in the chemistry literature (cf. [18, Chapter 6]). The network equals



The rate constants κ_{ij} are written over the edges. The mass action system (8.1) equals

$$\phi(x) = [x_1 \quad x_2 \quad x_3 \quad x_1^2 \quad x_4] \cdot \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -10 & 10 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (8.2)$$

where x_i is the concentration of species X_i . Explicitly, this is the system (1.1) with

$$\phi(x_1, x_2, x_3, x_4) = [-x_1 - 20x_1^2, x_1 - x_2, x_2, 10x_1^2].$$

Computations of the critical reactors of this system are found in [18, Section 5.3].

Fix the starting point $y = (1, 0, 0, 0)$. The trajectory starting at y converges to the steady state $y^* = (0, 0, 0.1522, 0.4238)$. The dynamics takes place in \mathbb{R}^4 , but the stoichiometry space has dimension 3. In our analysis we use the projection onto

the first three coordinates. With this, the trajectory is an arc in a 3-dimensional space, shown in blue in Figure 9.

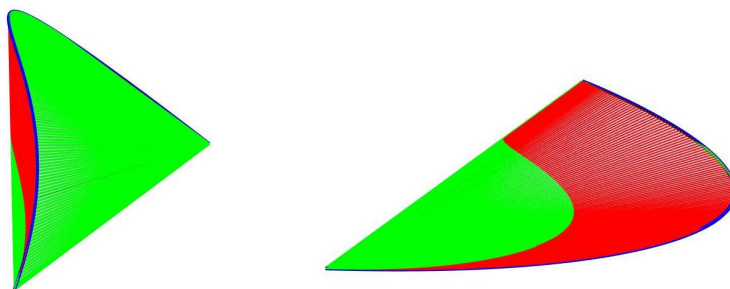


FIGURE 9. Convex trajectory of the Van de Vusse reaction and the partition of its boundary.

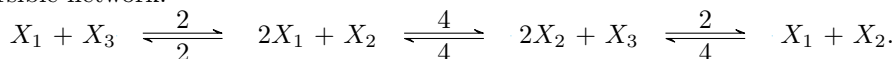
We computed the convex trajectory starting at y for (8.2) using Algorithm 5.4, and we then partitioned its boundary using Algorithm 7.1. The result is shown in Figure 9. The convex body has two 1-patches, obtained by joining each of the two endpoints with each point on the curve. One of the patches is entirely green. This means that the vector field is pointing inward on that patch. The other patch is partitioned into a green region and a red region, as shown on the right in Figure 9. Red color indicates that the vector field points outward. In particular, the convex trajectory is strictly contained in the attainable region.

The mass action system (8.1) is called *weakly reversible* if every connected component of the underlying directed graph is strongly connected in G , i.e. there is a directed path from any node in the component graph to any other node. It was conjectured in [14] that convex trajectories of weakly reversible systems are forward closed. We here resolve that conjecture.

Proposition 8.3. *Not all convex trajectories of weakly reversible systems are forward closed.*

The proof is by computation using our algorithms. Here is the counterexample:

Example 8.4 (Weakly Reversible System). Consider the following weakly reversible network:



The three coordinates for (1.1) are explicitly given by

$$\begin{aligned} \phi_1 &= -10x_1^2x_2 + 10x_2^2x_3 - 4x_1x_2 + 2x_1x_3, \\ \phi_2 &= 2x_1^2x_2 - 6x_2^2x_3 + 4x_1x_2 + 2x_1x_3, \\ \phi_3 &= 6x_1^2x_2 - 6x_2^2x_3 + 4x_1x_2 - 2x_1x_3. \end{aligned}$$

This system has deficiency zero, and it is a toric dynamical system [6]. There are no conservation relations. The trajectories are curves that span the ambient space \mathbb{R}^3 .

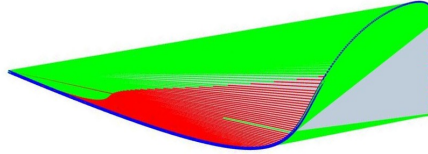


FIGURE 10. Convex trajectory of a weakly reversible system that is not forward closed.

Let $y = (4, 4, 2)$. The convex body $C = \text{convtraj}(y)$ was computed using Algorithm 5.4 and is shown in Figure 10. The triangle shown in gray is a 2-patch of C . The vector field given by (ϕ_1, ϕ_2, ϕ_3) points inward at all points on that triangle facet. We also show the partition of the 1-patches of C , as computed by Algorithm 7.1. One of the patches is partitioned into a green region and a red region. As before, the vector field points outward at each red point. We conclude that the convex trajectory C of y is not forward closed.

REFERENCES

- [1] F. Acquistapace, F. Broglia and J. F. Fernando: On globally defined semianalytic sets, *Math. Ann.* **366** (2016), no. 1-2, 613–654. MR 3552251.
- [2] C. Bajaj and M. S. Kim: Convex hulls of objects bounded by algebraic curves, *Algorithmica* **6** (1991), no. 4, 533–553. MR 1107814.
- [3] G. Blekherman, P. Parrilo and R. Thomas (eds.): *Semidefinite Optimization and Convex Algebraic Geometry*, MOS-SIAM Series on Optimization 13, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA; Mathematical Optimization Society, Philadelphia, PA, 2013. MR 3075433.
- [4] E. M. Bronstein: Approximation of convex sets by polytopes, *J. Math. Sci. (N.Y.)* **153** (2008), no. 6, 727–762. MR 2336506.
- [5] D. Ciripoi, A. Löhne and B. Weißing: *Bensolve tools - Calculus of convex polyhedra, calculus of polyhedral convex functions, global optimization, vector linear programming for Octave and Matlab*, Version 1.2, <http://tools.bensolve.org>.
- [6] G. Craciun, A. Dickenstein, A. Shiu and B. Sturmfels: Toric dynamical systems, *J. Symbolic Comput.* **44** (2009), no. 11, 1551–1565. MR 2561288.
- [7] A. Dickenstein and E. Feliu: *Algebraic Methods for Biochemical Reaction Networks*, textbook in preparation.
- [8] M. Ehrgott, A. Löhne and L. Shao: A dual variant of Benson’s “outer approximation algorithm” for multiple objective linear programming, *J. Global Optim.* **52** (2012), no. 4, 757–778. MR 2897457.
- [9] M. Feinberg and D. Hildebrandt: Optimal reactor design from a geometric viewpoint—I. Universal properties of the attainable region, *Chemical Engin. Sci.* **52** (1997), no. 10, 1637–1665.
- [10] J. Fu: Algebraic integral geometry, in: *Integral Geometry and Valuations*, 47–112, Adv. Courses Math. CRM Barcelona, Birkhäuser/Springer, Basel, 2014. MR 3330875.
- [11] V. Hárs and J. Tóth: On the inverse problem of reaction kinetics, in: *Qualitative Theory of Differential Equations (Szeged, 1979)*, 363–379, Colloquia Mathematica Societatis János Bolyai, 30, Amsterdam-New York, North-Holland, 1981.

- [12] J. van der Hoeven: Certifying trajectories of dynamical systems, In: I. S. Kotsireas, S. M. Rump and C. K. Yap (eds.), *Mathematical Aspects of Computer and Information Sciences (MACIS 2015)*, 520–532. Lecture Notes in Computer Science, 9582, Springer, [Cham], 2016. MR 3517059.
- [13] B. Joshi and A. Shiu: Which small reaction networks are multistationary?, *SIAM J. Appl. Dyn. Syst.* **16** (2017), no. 2, 802–833. MR 3633777.
- [14] N. Kaihnsa: *Attainable regions of dynamical systems*, presented at MEGA 2019 (Effective Methods in Algebraic Geometry), Madrid, June 2019, arXiv:1802.07298 [math.DS].
- [15] Y. J. Kim, J. Lee, M. S. Kim and G. Elber: Efficient convex hull computation for planar freeform curves, *Computers & Graphics* **35** (2011), no. 3, 698–705.
- [16] A. Löhne and B. Weißing: Equivalence between polyhedral projection, multiple objective linear programming and vector linear programming, *Math. Methods Oper. Res.* **84** (2016), no. 2, 411–426. MR 3566932.
- [17] A. Löhne and B. Weißing: The vector linear program solver Bensolve—notes on theoretical background, *European J. Oper. Res.* **260** (2017), no. 3, 807–813. MR 3626170.
- [18] D. Ming, D. Glasser, D. Hildebrandt, B. Glasser and M. Metzger: *Attainable Region Theory: An Introduction to Choosing an Optimal Reactor*, Wiley, 2016.
- [19] K. Ranestad and B. Sturmfels: On the convex hull of a space curve, *Adv. Geom.* **12** (2012), no. 1, 157–178. MR 2911165.
- [20] A. Shiu: *Algebraic methods for biochemical reaction network theory*, Ph.D. thesis, University of California, Berkeley, 2010. MR 2941539.
- [21] Z. Smilansky: Convex hulls of generalized moment curves, *Israel J. Math.* **52** (1985), no. 1-2, 115–128. MR 0815607.
- [22] G. Ziegler: *Lectures on Polytopes*, Graduate Texts in Mathematics, 152, Springer, New York, 1995. MR 1311028.

D. Ciripoi

Universität Jena

daniel.ciripoi@uni-jena.de

N. Kaihnsa

MPI-MiS Leipzig

kaihnsa@mis.mpg.de

A. Löhne

Universität Jena

andreas.loehne@uni-jena.de

B. Sturmfels [✉]

MPI-MiS Leipzig and UC Berkeley

bernd@mis.mpg.de

Received: October 9, 2018

Accepted: May 28, 2019