A Numerical Method for Computing Border Curves of Bi-parametric Real Polynomial Systems and Applications

Changbo Chen and Wenyuan Wu*

Chongqing Key Laboratory of Automated Reasoning and Cognition, Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences
changbo.chen@hotmail.com, wuwenyuan@cigit.ac.cn

Abstract. For a bi-parametric real polynomial system with parameter values restricted to a finite rectangular region, under certain assumptions, we introduce the notion of border curve. We propose a numerical method to compute the border curve, and provide a numerical error estimation.
The border curve enables us to construct a so-called "solution map". For a given value u of the parameters inside the rectangle but not on the border, the solution map tells the subset that u belongs to together with a connected path from the corresponding sample point w to u. Consequently, all the real solutions of the system at u (which are isolated) can be obtained by tracking a real homotopy starting from all the real roots at w throughout the path. The effectiveness of the proposed method is illustrated by some examples.

1 Introduction

Parametric polynomial systems arise naturally in many applications, such as robotics [7], stability analysis of biological systems [25], model predictive control [8], etc. In these applications, often it is important to identify different regions in the real parametric space such that the system behaves the "same" in each region.

It is not a surprise that symbolic methods have been the dominant approaches for solving parametric systems due to their ability to describe exactly the structure of the solution sets. The symbolic methods can be classified into two categories, namely the approaches which are primarily interested in finding the solutions in an algebraically closed field (often the complex field) or a real closed field (often the real field). Methods belonging to the first category include the Gröbner basis method [2], the characteristic set or triangular decomposition method [29], the comprehensive Göbner bases method [26], the comprehensive triangular decomposition method [5, 4], etc. Methods belonging to the second one include the quantifier elimination method [24], the cylindrical algebraic decomposition method [6], the Sturm-Habicht sequence method [9], the parametric

[*] Corresponding author.

geometric resolution method [21], etc. Our method is directly motivated by the border polynomial method [30] and the discriminant variety method [12, 18].

Another motivation of our work comes from the recent advances from the numeric community. The homotopy continuation method [13, 22], which was initially used to compute all the complex solutions of zero-dimensional polynomial systems, has been developed to study the complex and real witness points of positive dimensional polynomial system [22, 10, 27], as well as describing the real algebraic curves [16] and surfaces [17]. The Cheater's homotopy [14, 22] provides a way to compute the complex solutions of a parametric polynomial system at a particular parameter value u by first computing the solutions of the system at a generic random parametric value u_0 and then using these solutions as starting points for constructing a homotopy. In [15], a real homotopy continuation method for computing the real solutions of zero-dimensional polynomial systems is introduced. In [19], the authors provide a local approach to detect the singularities of a parametric polynomial system along a solution path when sweeping though the parametric space.

At last, we would like to mention that this work is also motivated by an invited talk given by Hoon Hong [11], where the speaker suggested some possible ideas for doing quantifier elimination by a symbolic-numeric approach.

In this paper, we propose a numeric method for computing all the real solutions of a bi-parametric polynomial system F with parameters values restricted in a rectangular region R. Under certain assumptions (see the beginning of Section 2), there exists a border curve B in R which divides the rectangle R into finitely many connected components (called cells) such that in each cell the real zero set of F defines finitely many smooth functions with disjoint graphs. We provide a numeric algorithm to compute such a curve and analyze its numerical error. The key idea is to reduce the computation of the border curve to tracing all its corresponding connected components in a higher dimensional space with the help of the critical point techniques [20, 27].

To handle the numerical error, we define the notion of δ-connectedness and make use of the connected component of a graph G to represent the cells separated by the border curve. Moreover, in each cell, we choose a sample point far from the border curve and compute all the real solutions of F at this point. All in all, the border curve B, the connectivity graph G, the set of sample points W and the set Z of solutions at these sample points together form a so-called "solution map". Now if one wants to compute the real solutions of F at a given value u of parameters, instead of directly solving $F(u)$, one could easily make use of the solution map to choose the sample point w sharing the same cell with u and construct real homotopies from known results $V_{\mathbb{R}}(F(w))$ to unknown $V_{\mathbb{R}}(F(u))$.

Obviously, provided that the solution map has been computed during an offline phase, it will be very efficient to solve $V_R(F(u))$ online by the real homotopy approach when the number of the real solutions at u is much smaller than the number of complex ones. However, if one's interest is only on computing the real solutions of a zero-dimensional system, it is not recommended to go though the above costly offline computation.

The notion of border curve is inspired by the notions of border polynomial and discriminant variety. However, it applies directly to characterizing the real zero set of a parametric system while the later two also characterize the distinct complex solutions. We use the following example to illustrate the difference of border curve with them.

Example 1 *Consider a parametric system consisting of a single polynomial:*

$$f := (X_1 - U_1)(X_1 - U_2 - 1)(X_1^2 - 2U_1 + U_1^2 + 5)(X_1^2 - 2U_2 + U_2^2 + 5).$$

The border polynomial or discriminant variety of f is defined by the discriminant of f with respect to X_1, which has 9 irreducible factors: $U_1 - U_2, U_1 - 2 + U_2, U_1 - 1 - U_2, U_2^2 + 3, U_1^2 - 2U_1 + 5, 2U_1^2 - 2U_1 + 5, U_2^2 - 2U_2 + 5, U_1^2 + U_2^2 - 2U_2 + 5, U_1^2 + U_2^2 - 2U_1 + 2U_2 + 6$. Among them, only the zero sets of the first three factors are nonempty. They define the real counterpart of the border polynomial or discriminant variety, which characterizes when f has multiple complex roots. In contrast, the border curve is defined only by the polynomial $U_1 - 1 - U_2$, which characterizes when f has a multiple real root with respect to X_1. Indeed, the last two factors of f have no real points.

2 Border curve

In this section, we introduce the concept of border curve for a bi-parametric polynomial system and present a numeric method to compute it.

Throughout this paper, let $F(X_1, \ldots, X_m, U_1, U_2) = 0$ be a bi-parametric polynomial system consisting of m polynomials with real coefficients. Let $\pi : \mathbb{R}^{m+2} \to \mathbb{R}^2$ be the projection defined by $\pi(x_1, \ldots, x_m, u_1, u_2) = (u_1, u_2)$. Let R be a rectangle of the parametric space (U_1, U_2). We make the following assumptions for F and R:

(A_1) The set $V_{\mathbb{R}}(F) \cap \pi^{-1}(R)$ is compact.

(A_2) Let $F' := \{F, \det(\mathcal{J}_F)\}$, where $\det(\mathcal{J}_F)$ is the determinant of the Jacobian of F with respect to $(X_1, \ldots, X_m)$. We have $\dim(V_{\mathbb{R}}(F')) = 1$.

(A_3) At each regular point of $V_{\mathbb{R}}(F')$, the Jacobian of F' has full rank.

Definition 1 *Given a bi-parametric polynomial system $F(X_1, \ldots, X_m, U_1, U_2) = 0$. Assume that it satisfies the first two assumptions. Then the border curve B of F restricted to the rectangle R is defined as $\pi(V_{\mathbb{R}}(F, \det(\mathcal{J}_F))) \cap R$.*

Remark 1 *Assumption (A_3) is not needed for this definition. It is required by the subroutine* RealWitnessPoint *of Algorithm* BorderCurve *for numerically computing the border. Assumption (A_1) can be checked by symbolic methods [12].*

Proposition 1 *Let B be the border curve of F restricted to the rectangle R. Then $R \setminus B$ is divided into finitely many connected components, called cells, such that in each cell, the real zero set of F defines finitely many smooth functions, whose graphs are disjoint.*

Proof. By Assumption (A_2), the set $R \setminus B$ is non-empty. Moreover it is a semi-algebraic set and thus has finitely many connected components. Let $\mathcal{C}$ be any component and let u be any point of $\mathcal{C}$. Since $u \notin B$, by Definition 1, if the set $\pi^{-1}(u) \cap V_{\mathbb{R}}(F)$ is not empty, the Jacobian $\mathcal{J}_F$ is non-singular at each point of it. Thus the set $\pi^{-1}(u) \cap V_{\mathbb{R}}(F)$ must be finite. On the other hand, by the implicit function theorem, around each neighborhood of these points, $V_{\mathbb{R}}(F)$ uniquely defines a smooth function of (U_1, U_2). By the compactness assumption (A_1) and the connectivity of $\mathcal{C}$, the domain of these functions can be extended to the whole $\mathcal{C}$. For similar reasons, if the set $\pi^{-1}(u) \cap V_{\mathbb{R}}(F)$ is empty, the set $\pi^{-1}(\mathcal{C}) \cap V_{\mathbb{R}}(F)$ must also be empty. Thus the proposition holds.

Next we present a numeric algorithm for computing the border curve. Let RealWitnessPoint be the routine introduced in [27] for computing the witness points of a real variety $V_{\mathbb{R}}(F')$ satisfying Assumption (A_3). The basic idea of this routine is to introduce a random hyperplane H. Then "roughly speaking" the witness points of $V_{\mathbb{R}}(F')$ either belong to $V_{\mathbb{R}}(F') \cap H$ or are the critical points of the distance from the connected components of $V_{\mathbb{R}}(F')$ to H.

```
Algorithm BorderCurve
Input: a bi-parametric polynomial system $F = 0$; a rectangle $R$.
Output: an approximation of the border curve of $F$ restricted to $R$.
Steps:
  1. Let $F'$ be the new polynomial system $F \cup \{\det(\mathcal{J}_F)\}$.
  2. Set $W := \emptyset$.
  3. Compute the intersection of $V_{\mathbb{R}}(F')$ with the fibers of the four edges of
     $R$ by a homotopy continuation method and add the points into $W$.
  4. Compute RealWitnessPoint($F'$) and add the points into $W$.
  5. For each point $p$ in $W$, starting from $p$, follow both directions of the
     tangent line of $V_{\mathbb{R}}(F')$ at $p$, trace the curve $F'$ by a prediction-projection
     method (see Lemma 2), until a closed curve is found or the projections
     of the traced points onto $(U_1, U_2)$ hit the boundary of $R$.
  6. Return the projections of the traced points in $R$.
```

Remark 2 *In the above algorithm, it is possible that the determinant* $\det(\mathcal{J}_F)$ *is a polynomial of large degree or with large coefficients. For numerical stability, we can instead set* $F' := F \cup \mathcal{J}_F \cdot \mathbf{v} \cup \{\mathbf{n} \cdot \mathbf{v} - 1\}$ *with additional variables* $\mathbf{v} = \{v_1, \ldots, v_m\}$*, where* $\mathbf{n}$ *is a random vector of* $\mathbb{R}^m$ *and* $\mathbf{n} \cdot \mathbf{v} = 1$ *is a random real hyperplane.*

Example 2 *Consider a rectangle* $R := [-3, 3] \times [-3, 3]$ *and a parametric system*

```
F := [X_1^3+(11/6)*X_1^2-(8/15)*X_1*X_2-(8/3)*X_1*U_1-(4/3)*X_1*U_2
-(1/6)*U_2^2-77/30, X_2^3+(7/30)*X_1*U_1+(41/30)*X_1*U_2+(37/15)*X_2*U_2]
```

The discriminant variety of F *is the zero set of an irreducible polynomial of degree* 24 *with* 301 *terms.*

```
-3278867339608044797952000000000000000000*U_1^12*U_2^12
-...+1587241704052253597928493467628800000 U_1
+5802432050835806172320572531891200000 U_2
-109907933560147956473543161255680000
```

Its zero set is the red curve plotted in Figure 3. Algorithm BorderCurve *computes a space curve in $\mathbb{R}^4$, whose projection onto (U_1, U_2, X_1) is illustrated in Figure 1. The projection of the space curve onto (U_1, U_2) (namely the border curve B), when drawn in the rectangle R, is "the same" (cannot tell the difference by eyes) as the red curve shown in Figure 3.*

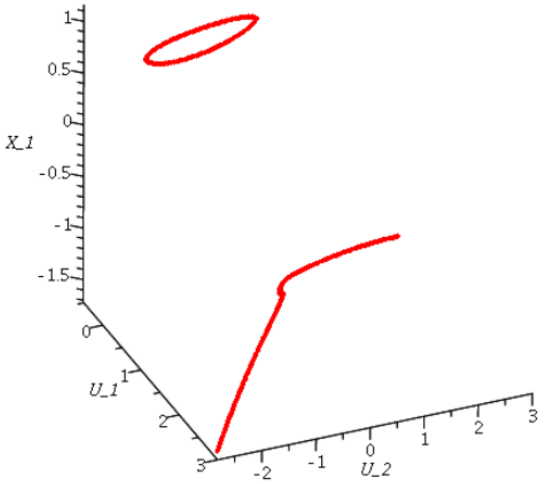

Fig. 1. The projection of the space curve onto (U_1, U_2, X_1).

3 Numerical error estimation

In last section, we introduced the concept of border curve and provided an algorithm to compute a numerical approximation of it. In this section, we estimate the numerical error of such an approximated border curve. We first recall several results from [28].

Lemma 1 *Let $P(X_1, \ldots, X_n) = \{P_1, \ldots, P_{n-1}\}$ be a set of $n-1$ polynomials with n variables. Let $\mathcal{J}$ be the Jacobian of P. Let $\mathcal{J}_{ij} = \partial P_i / \partial X_j$, $i = 1, \ldots, n-1$, $j = 1, \ldots, n$. Let $K(P) := \max(\{\|\nabla \mathcal{J}_{ij}(z)\|_2 \mid z \in V_{\mathbb{R}}(\sum_{i=1}^{n} X_i^2 - 1\})$. Let $\mu = \sqrt{(n-1)n}$. Assume that $K(P) \leq 1$ holds. Let z_0 and z_1 be two points of $V_{\mathbb{R}}(\sum_{i=1}^{n} X_i^2 - 1)$. Then we have*

$$\|\mathcal{J}(z_1) - \mathcal{J}(z_0)\|_2 \leq \mu \|z_1 - z_0\|_2.$$

Remark 3 *This lemma was proved in [28] as Equation* (21).

Lemma 2 (Theorem 3.9 in [28]) *Let $P(X_1,\dots,X_n)=\{P_1,\dots,P_{n-1}\}$ be a set of $n-1$ polynomials with n variables such that $V_{\mathbb{R}}(P)\subset V_{\mathbb{R}}(\sum_{i=1}^{n}X_i^2-1)$ and $K(P)\leq 1$ hold. Let z_0 be a point of $V_{\mathbb{R}}(P)$. Let σ be the smallest singular value of $\mathcal{J}(z_0)$. Let $\mu=\sqrt{(n-1)n}$. Let $\omega=\sqrt{2\,(2\,\rho-1)\left(2\,\rho-2\,\sqrt{\rho\,(\rho-1)}-1\right)}$, for some $\rho\geq 1$. Assume that $2\rho>3\omega$ holds (which is true for any $\rho\geq 1.6$). Let $s=\frac{\sigma}{2\mu\rho}$. Let L be a hyperplane which is perpendicular to the tangent line of $V_{\mathbb{R}}(P)$ at z_0 of distance s to z_0.*

We move z_0 in the tangent direction in distance s and apply the Newton Iteration to the zero dimensional system $P\cup\{L\}$. Assume that the Newton Iteration converges to z_1. Then z_1 is on the same component with z_0 if and only if

$$\|z_1-z_0\|<\omega\cdot s. \tag{1}$$

Remark 4 *Note that one can always find a $\rho\geq 1.6$ (by increasing ρ and thus decreasing the step size s) such that $\|z_1-z_0\|<\omega\cdot s$ holds.*

We have the following numerical error estimation of the border curve.

Theorem 1 *Let $F(X_1,...,X_m,U_1,U_2)=0$ be a bi-parametric polynomial system satisfying the assumptions A_1 and A_2. Let B be the border curve of F restricted to a rectangle R. Let $F':=F\cup\{\det(\mathcal{J}_F)\}$. We consider*

$$P=\{\bar{F'}_1,...,\bar{F'}_{m+1},\sum_{k=0}^{m}X_k^2/2-1/2=0\}, \tag{2}$$

where $\bar{F'}_i, 1\leq i\leq m+1$ is homogenized from F'_i in variables $\{X_1,...,X_m\}$ by an additional variable X_0.

Since we consider the solutions of P in a rectangular region, without loss of generality, we can assume $K(P)\leq 1$. Otherwise the polynomials can be rescaled by that upper bound of $K(P)$.

Let z_0 and z_1 be two points of $V_{\mathbb{R}}(P)$ satisfying Equation (1). Let $\mathcal{C}_{z_0z_1}$ be the curve segment between z_0 and z_1 in $V_{\mathbb{R}}(P)$. Let $\tilde{z}_0$ and $\tilde{z}_1$ be computed points caused by numerical error within distance ϵ to z_0 and z_1 respectively.

Let $\mu=\sqrt{(m+2)(m+3)}$. Let $\tilde{\sigma_0}$ and $\tilde{\sigma_1}$ be respectively the smallest singular value of $\mathcal{J}_P(\tilde{z}_0)$ and $\mathcal{J}_P(\tilde{z}_1)$. Let $\tilde{\sigma}:=\max(\tilde{\sigma_0},\tilde{\sigma_1})$. Let ρ and ω be as defined in in Lemma 2. Let u_0 and u_1 be respectively the projection of $\tilde{z}_0$ and $\tilde{z}_1$. Let $B_{z_0z_1}\subset B$ be the projection of $\mathcal{C}_{z_0z_1}$. Then the distance from $B_{z_0z_1}$ to the segment $\overline{u_0u_1}$ is at most

$$\tan\left(2\arccos(\frac{1}{\omega})\right)\frac{\omega}{4\mu\rho}(\mu\epsilon+\tilde{\sigma})+\epsilon.$$

or $1.082\epsilon+0.082\frac{\tilde{\sigma}}{\mu}$ if we choose $\rho=1.6$.

Proof. Let σ_0 (resp. σ_1) be the smallest singular value of $\mathcal{J}_P(z_0)$ (resp. $\mathcal{J}_P(z_1)$). Let $s_i=\frac{\sigma_i}{2\mu\rho}, i=1,2$. Let L_0 (resp. L_1) be hyperplane which is perpendicular to the tangent line of $V_{\mathbb{R}}(P)$ at z_0 (resp. z_1) of distance s_0 (resp. s_1) to z_0 (resp. z_1). Let $h := \|z_0 - z_1\|$.

We define a cone with z_0 as the apex, the tangent line at z_0 as the axis, and the angle deviating from the axis being $\theta := \arccos(\frac{s_0}{\omega s_0})$. By Lemma 2, the curve from z_0 to z_1 must be in this cone when the step size is small. Similarly, we can construct another cone with z_1 as the apex, the tangent line at z_1 as the axis, and the angle deviating from the axis being $\theta := \arccos(\frac{s_1}{\omega s_1})$, such that it contains the curve from z_1 back to z_0. The projection of the intersection of these cones onto the parametric space is a triangle or quadrilateral. Figure 2 illustrates the two cones and the curve contained in them. From Figure 2, we know that $|CE| < |AE|\tan(2\theta)$ and $|CE| < |EB|\tan(2\theta)$ hold. Since $|AE| + |EB| = h$, we deduce that $|CE| < \frac{h}{2}\tan(2\theta) = \frac{h}{2}\tan(2\arccos(1/\omega))$.

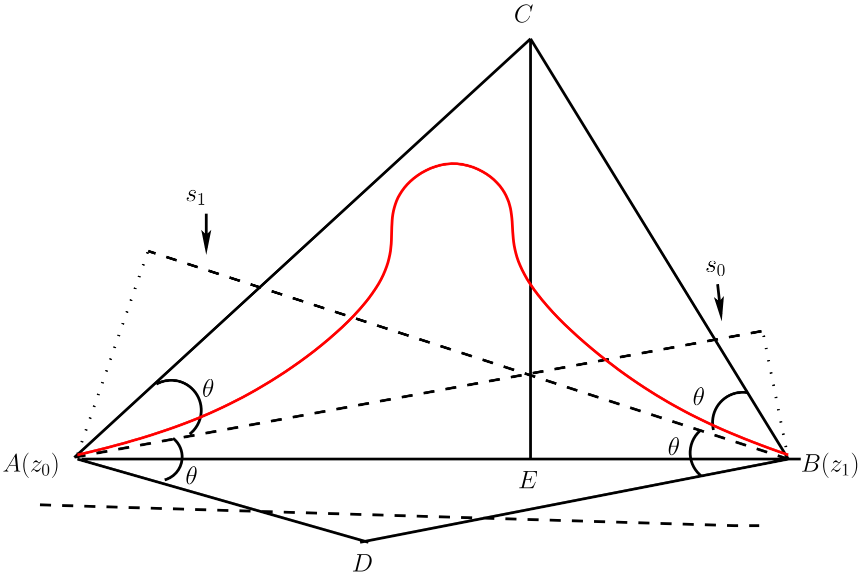

Fig. 2. A 2D image illustrating the intersection of two cones.

Next we estimate s_0 and s_1. By Weyl's theorem [23], $|\tilde{\sigma_i} - \sigma_i| \leq \|\mathcal{J}_P(\tilde{z_i}) - \mathcal{J}_P(z_i)\|_2$ holds. By Lemma 1, we have $\|\mathcal{J}_P(\tilde{z_i}) - \mathcal{J}_P(z_i)\|_2 \leq \mu\epsilon$. Thus $\sigma_i \leq \mu\epsilon + \tilde{\sigma}_i$, $i = 1, 2$, holds. Therefore we have $s_i \leq \frac{\mu\epsilon + \tilde{\sigma}}{2\mu\rho}$, $i = 1, 2$.

Since the distance from any point on the curve to the segment $\overline{z_0 z_1}$ is no greater than $|CD|$ and the distance from $\overline{z_0 z_1}$ to $\overline{\tilde{z}_0 \tilde{z}_1}$ is no greater than ϵ, we obtain the final estimation

$$\tan\left(2\arccos(\frac{1}{\omega})\right)\frac{\omega}{4\mu\rho}(\mu\epsilon + \tilde{\sigma}) + \epsilon.$$

Example 3 *For the polynomial system F, the rectangle R given in Example 2, the theoretical error estimation given by Theorem 1 is about 7.08×10^{-3} while the actual computed error is about 9.91×10^{-4}. Both errors can be reduced if a smaller step size is chosen.*

4 Constructing the solution map

As an extension of Proposition 1, in this section, we introduce a numerical version of the connected components of $R \setminus B$ and the notion of solution map to describe the real solutions of the parametric system F restricted to the rectangle R.

In Section 2, we provided an algorithm to compute a numerical approximation of the border curve. In Section 3, we estimated the distance between the approximated border curve and the true border curve. Since the border curve and its approximation usually do not overlap, two points are connected with respect to the approximated curve does not imply that they are connected with respect to the border curve. To handle this problem, we introduce the notion of "δ-connected", where δ should be no less than the error estimation provided by Theorem 1, to divide the interior of a rectangle into "numerically connected areas" with respect to an approximation of the border curve.

Definition 2 (δ-stripe) *Let $p \in \mathbb{R}^2$. Let $D_r(p)$ be the closed disk of center p and radius r. Let Γ be a path of $\mathbb{R}^2$. We define $\Gamma_\delta := \cup_{p \in \Gamma} D_{\delta/2}(p)$ as the* associated δ-stripe *of Γ.*

Definition 3 (δ-connected) *Let R be a rectangle of $\mathbb{R}^2$ and let S be a set of points in R. Let $\delta \geq 0$. Let p_1, p_2 be two points of $R \setminus S$. We say p_1 and p_2 are* δ-connected with respect to S *if there is a path Γ in R connecting p_1 with p_2 such that there are no points of S belonging to the associated δ-stripe of Γ connecting p_1 and p_2. A* δ-connected set *of R is a subset of R such that every two points of it is δ-connected.*

It is clear that the two notions connectedness and δ-connectedness coincide when $\delta = 0$ holds. The following two propositions establish the relations between connectedness and δ-connectedness when $\delta > 0$ holds.

Proposition 2 *Let B be the border curve of a bi-parametric polynomial system F restricted to a rectangle R. By Theorem 1, there exists a polyline $\tilde{B}$ and a $\delta \geq 0$ such that $B \subseteq \tilde{B}_\delta$. If two points p_1 and p_2 are δ-connected with respect to $\tilde{B}$, then they are connected with respect to B.*

Proof. Since p_1 and p_2 are δ-connected with respect to $\tilde{B}$, there exists a path Γ connecting p_1 and p_2 such that $\Gamma_\delta \cap \tilde{B} = \emptyset$ holds. To prove the proposition, it is enough to show that B has no intersection with Γ. Now assume that B intersects with Γ at a point p. Since Γ_δ is the associated δ-stripe of Γ, the distance between p and $\tilde{B}$ is greater than $\delta/2$, which is a contradiction to the fact that $B \subseteq \tilde{B}_\delta$.

Next we associate a rectangle with a grid graph.

Definition 4 *Let $\delta > 0$. Let R be a rectangle with width $m\delta$ and with length $n\delta$, where $m, n \in \mathbb{N}$. It can be naturally divided into a $m \times n$ grid of mn squares. The grid itself also defines an undirected graph, whose vertices and edges are exactly those of squares (overlapped vertices and edges are treated as one) in the grid. Such a grid (together with the graph) is called the* associated δ-grid *of R.*

Let G be a subgraph of the associated δ-grid of R. The set defined by G *is the set of points on G and the set of points inside the squares of G. Let p be a point of R. Then p belongs to at least one square in the δ-grid of R. We pick one of them according to some fixed rule and call it the* associated square *of p, denoted by A_p.*

Remark 5 *Here, for simplicity, we use a grid where each square has the same size. It is also possible to define a grid with different sizes of squares.*

We have the following key observation.

Proposition 3 *Let R be a rectangle and G be its associated δ-grid graph. Let S be a set of interior points of R. For every $p \in S$, we remove from G all the vertices of the associated square A_p of p and name the resulting graph still by G. Let $Z(G)$ be the set defined G. Then we have*

- *The distance between p and any point of $Z(G)$ at least δ.*
- *Let $q \in R \setminus Z(G)$. Then there exists a p, $1 \leq i \leq s$, such that the distance between p and q is at most $2\sqrt{2}\delta$.*

Proof. For a given A_p, let N_p be the set of all its neighboring squares (at most 8). When we delete the vertices (and the edges connected to them) of A_p, these vertices and edges are also removed from N_p. As a result, the distance between p and those undeleted vertices and edges of N_p is at least δ and at most $2\sqrt{2}\delta$. Thus the proposition holds.

Remark 6 *Informally speaking, this proposition tells us that the connected components in G is at least δ, but at most $2\sqrt{2}\delta$ far from the points in S.*

Definition 5 (Connectivity Graph) *Let $\delta > 0$. Let R be a rectangle of $\mathbb{R}^2$ of size $m\delta \times n\delta$. Let B be a set of sequences of points in R. The distance between two successive points in a sequence is at most δ. A* connectivity graph *of (R, B) is a subgraph G of the δ-grid of R such that*

- *Each connected component of G defines a δ-connected subset of R with respect to B.*
- *There exists $s \in \mathbb{N}$ such that every interior point of R, which is $s\delta$ far from points in S belongs to at least one of subsets defined by the connected components of G.*

The following algorithm computes a connectivity graph.

Algorithm ConnectivityGraph

Input: A rectangle R of size $m\delta \times n\delta$, a set B of sequence of points belonging to R. The distance between two successive points in a sequence is at most δ.

Output: A connectivity graph of (R, B).
Steps:

```
1. Let $G$ be the $\delta$-grid of $R$.
2. For each point $p$ of $B$, delete the four vertices (and edges connected to
   them) of the associated square $A_p$ from $G$.
3. Return $G$.
```

Proposition 4 *Algorithm* ConnectivityGraph *is correct.*

Proof. By Proposition 3, for any vertex v of G and any point b of B, the distance between them is at least δ. Thus, by Definition 3, the zero set defined by each connected component of G is δ-connected. By Proposition 3, if the minimal distance between an interior point p of R and points in B is greater than $2\sqrt{2}\delta$, then p must belong to the set defined by G. Thus, the algorithm is correct.

Finally we are able to define the solution map of a bi-parametric polynomial system restricted to a rectangle R.

Definition 6 (Solution map) *Given a rectangle R and a bi-parametric polynomial system $F(X_1, \ldots, X_m, U_1, U_2) = 0$. Assume that F satisfies the assumptions A_1, A_2. A* solution map *of F restricted to R, denoted by M, is a quadruple (B, G, W, Z) where*

- *B is a set of sequences of points approximating the border curve of F.*
- *G is a connectivity graph of (R, B).*
- *W is a set of points in R s.t. each point of W is a vertex of a connected component of G and each connected component has exactly one vertex in W.*
- *Z is a correction of sets of points in $\mathbb{R}^m$ such that each element of Z is a solution set of F at a point of W.*

Example 4 *Consider the system in Example 2. Its connectivity graph and solution map are shown in Figure 3. The set of sample points is $W = \{(U_1 = \frac{1}{2}, U_2 = \frac{-6}{5}), (U_1 = \frac{1}{2}, U_2 = \frac{6}{5}), (U_1 = \frac{-12}{5}, U_2 = \frac{-3}{5})\}$. The corresponding set of solution sets Z is*

$$\{\{(X_1 = 1.077556426, X_2 = 1.497479963), (X_1 = .9908314677, X_2 = .4355629194),\\ (X_1 = .8136158335, X_2 = -1.930040765)\}, \{(X_1 = -.8199351413, X_2 = -.6397859006),\\ (X_1 = -2.611117469, X_2 = .3233313936), (X_1 = 1.354310493, X_2 = -1.291254606)\},\\ \{X_1 = .2814014780, X_2 = -2.885892092\}\}.$$

5 Real homotopy continuation

As an application of the solution map, in this section, we present how to construct a real homotopy to compute all the real zeros of a parametric polynomial system at a given value of parameters.

Definition 7 *Let $H(X, t) \subset \mathbb{R}[X_1, \ldots, X_n, t]$. We call $H(X, t)$ a* real homotopy *if it satisfies the smoothness property: over the interval $[0, 1]$, the real zero set of $H(X, t)$ defines a finite number of smooth functions of t with disjoint graphs.*

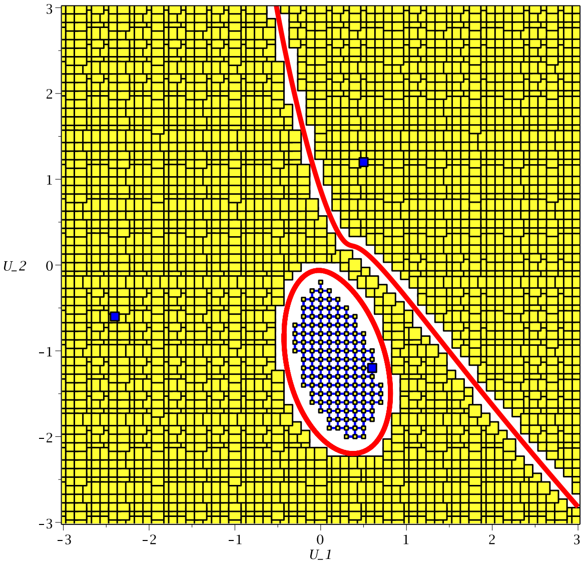

Fig. 3. The solution map (solution sets not shown).

Given a real homotopy and the solutions of H at $t = 0$, one can use a standard prediction-projection method with the adaptive step size strategy in Lemma 2 to trace the solution curves of $H(X, t)$ to get the solutions of H at $t = 1$. We denote such an algorithm by RealHomotopy.

For a bi-parametric polynomial system F, in previous section, we have shown how to construct a solution map of F. To compute the real solutions of F at a given u of the parameters, RealHomotopy is called to trace the real homotopy starting from a known solution of F stored in M. More precisely, we have the following algorithm OnlineSolve.

Algorithm OnlineSolve

Input: a bi-parametric system $F \subset \mathbb{Q}[X, U]$, a rectangle R, an interior point u of R, a solution map M of F, and a point u of R.

Output: if u is not close to the border curve of F, return the real solutions of F at u, that is $V_{\mathbb{R}}(F(u))$; otherwise throw an exception.

Steps:

1. Let G be the connectivity graph in M.
2. Let A_u be the associated square of u. Let C_u be a connected component of G such that A_u is a subgraph of C_u. If C_u does not exist, throw an exception. If C_u exists, let v_u be one of the vertices of A_u.
3. Let w_u be the sample point of C_u in M.
4. Let $w_u \rightsquigarrow v_u$ be the shortest path between w_u and v_u computed for instance by Dijkstra's algorithm. Connecting v_u and u with a segment and denote the path $w_u \rightsquigarrow v_u \to u$ by Γ.

```
5. Let $p_0, p_1, \ldots, p_s$ be the sequence of successive vertices of $\Gamma$.
6. Let $S$ be the solution set of $F(p_0)$ in $M$.
7. For i from 0 to $s-1$ do
   (a) let $(\phi_i(t), \psi_i(t))$ be a parametrization of the segment $\overline{p_i p_{i+1}}$, $t \in [0,1]$;
   (b) let $H_i(X,t) := F(X, U_1 = \phi_i(t), U_2 = \psi_i(t))$;
   (c) let $S := \mathsf{RealHomotopy}(H_i, S)$;
8. Return $S$.
```

Proposition 5 *If an exception is not thrown, Algorithm* OnlineSolve *correctly computes the real solutions of F at u.*

Remark 7 *The correctness of* OnlineSolve *can be easily verified from its description. When the exception is thrown, it means that u is close to the border curve of F, which indicates a numerically difficult region with ill-conditioned Jacobians.*

A connected path between a sample point $(-5/2, -3/5)$ and a chosen point $(1, -1)$ is shown in Figure 4. This path is a guide for building real homotopies.

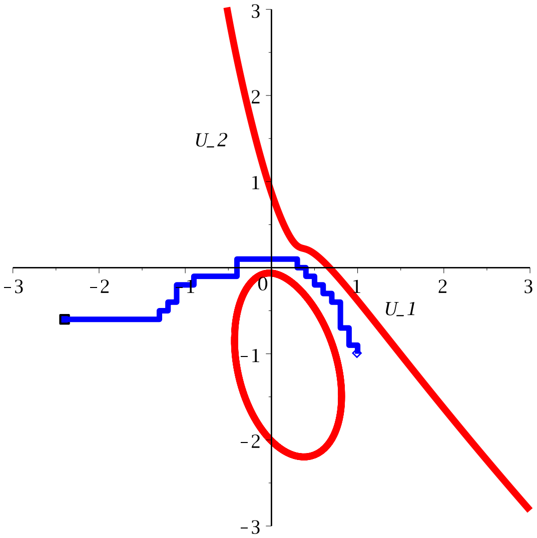

Fig. 4. A connected path for constructing real homotopies.

6 Experimentation

In this section, we illustrate the effectiveness of our method by some examples. Our method was implemented in Maple, which will make external calls to Hom4Ps2 (by exchanging input and output files) when it needs to compute the solutions of a zero-dimensional polynomial system. The experimentation was conducted on a Ubuntu Laptop (Intel i7-4700MQ CPU @ 2.40GHz, 8.0GB memory). The memory usage was restricted to 60% of the total memory. The timeout (represented by $-$ in Table 1) was set to be 1800 seconds. The experimentation results are summarized in Table 1, where BP denotes the command RegularChains[ParametricSystemTools][BorderPolynomial] and DV denotes the command RootFinding[Parametric][DiscriminantVariety] in the computer algebra system Maple 18, and BC denotes BorderCurve.

	Symbolic methods				Numeric method	
	BP		DV		BC	
Sys	time (s)	deg	time (s)	deg	time (s)	#(points)
1-2	1.340	1	0.562	1	3.798	171
1-3	0.575	4	0.019	4	2.147	307
1-4	0.433	9	0.021	9	1.097	252
1-5	0.385	16	0.024	16	0.668	153
1-6	0.575	25	0.031	25	1.940	313
1-7	0.396	30	0.053	30	1.579	127
1-8	0.389	48	2.724	48	3.399	668
2-2	0.552	4	0.028	4	3.641	839
2-3	0.800	24	0.372	24	14.748	2694
2-4	4.329	70	90.661	69	41.572	4084
2-5	68.930	173	-		10.695	266
2-6	-		-		110.657	4190
3-2	0.726	14	0.070	12	1.073	306
3-3	-		-		32.550	2301
3-4	-		-		286.638	9672
4-2	16.309	48	6.947	32	11.654	1073
4-3	-		-		188.058	5452
4-4	-		-		1415.822	11342
5-2	-		991.215	72	53.230	185
5-3	-		-		1054.712	11640
6-2	-		-		790.768	1486

Table 1. Experimentation results

The first column denotes the tested random sparse systems, each of which has a label i-j, where i denotes the number of equations (same as number of unknowns, or number of total variables minus 2) and j denotes the total degree. More precisely, the system i-j has the form $\{X_k^j +$ low degree terms on X, U $\mid k = 1 \cdots i\}$. The third and fifth column denotes respectively the degree of the border polynomial and the degree of the polynomial representing the discriminant variety. The seventh column denotes the number of points obtained by the numeric method for representing the border curves. The rest columns denote the execution time for three methods. From the table, it is clear that the numeric method (BC) outperforms the symbolic counterparts (BP and DV) on computing the border curves of systems of larger size.

7 Conclusion and future work

In this paper, under some assumptions, we introduced the concept of border curve for a bi-parametric real polynomial system and proposed a numerical method to compute it. The border curve was applied to describing the real solutions of a parametric polynomial system through the construction of a solution map and computing the real solutions of the parametric system at a particular value of parameters through constructing a real homotopy.

We tested a preliminary implementation of our method in Maple on computing the border curves of a set of randomly chosen sparse polynomial systems and compared our implementation with similar symbolic solvers on these examples. The experimentation shows that the numerical one is much more efficient than the symbolic ones on examples with more than 2 unknowns (or 4 variables).

In a future work, we will consider how to relax the assumptions introduced in Section 2. To make this approach more practical, structures of the polynomial system F' in Remark 2 must be exploited, for example using ideas in [28]. An efficient (and parallel) implementation of the method in C like languages is also important for applications. Extending the method to the multi-parametric case with possibly the help of the roadmap method [3, 1] and the numerical cell decomposition method [17] will be investigated.

Acknowledgements. This work is partially supported by NSFC (11301524, 11471307, 61572024) and CSTC (cstc2015jcyjys40001).

References

1. Basu, S., Roy, M.F., Safey El Din, M., Schost, É.: A baby step–giant step roadmap algorithm for general algebraic sets. Foundations of Computational Mathematics **14**(6), 1117–1172 (2014)
2. Buchberger, B.: An algorithm for finding a basis for the residue class ring of a zero-dimensional polynomial ideal. Ph.D. thesis, University of Innsbruck (1965)
3. Canny, J.: The Complexity of Robot Motion Planning. MIT Press (1987)
4. Chen, C., Golubitsky, O., Lemaire, F., Moreno Maza, Pan, W.: Comprehensive triangular decomposition. In: Ganzha, V.G., Mayr, E.W., Vorozhtsov, E.V. (eds.) CASC 2007. LNCS, vol. 4770, pp. 73–101. Springer, Heidelberg (2007)
5. Chou, S.C., Gao, X.S.: Computations with parametric equations. In: ISSAC 1991. pp. 122–127. ACM (1991)
6. Collins, G.E.: Quantifier elimination for real closed fields by cylindrical algebraic decompostion. In: Brakhage, H. (ed.) Automata Theory and Formal Languages 2nd GI Conference. LNCS, vol. 33, pp. 134–183. Springer, Heidelberg (1975)
7. Corvez, S., Rouillier, F.: Using computer algebra tools to classify serial manipulators. In: Winkler, F. (ed.) ADG 2002. LNCS, vol. 2930, pp. 31–43. Springer, Heidelberg (2004)

8. Fotiou, I.A., Rostalski, P., Parrilo, P.A., Morari, M.: Parametric optimization and optimal control using algebraic geometry. International Journal of Control **79**(11), 1340–1358 (2006)
9. González Vega, L., Lombardi, H., Recio, T., Roy, M.F.: Sturm-Habicht sequence. In: ISSAC 1989. pp. 136–146. ACM (1989)
10. Hauenstein, J.D.: Numerically computing real points on algebraic sets. Acta Applicandae Mathematicae **125**(1), 105–119 (2012)
11. Hong, H.: Overview on real quantifier elimination. In: MACIS 2013. pp. 1–1 (2013)
12. Lazard, D., Rouillier, F.: Solving parametric polynomial systems. J. Symb. Comput. **42**(6), 636–667 (2007)
13. Li, T.Y.: Numerical solution of multivariate polynomial systems by homotopy continuation methods. Acta Numerica **6**, 399–436 (1997)
14. Li, T.Y., Sauer, T., Yorke, J.A.: The Cheater's homotopy: An efficient procedure for solving systems of polynomial equations. SIAM Journal on Numerical Analysis **26**(5), 1241–1251 (1989)
15. Li, T.Y., Wang, X.S.: Solving real polynomial systems with real homotopies. Mathematics of Computation **60**, 669–680 (1993)
16. Lu, Y., Bates, D., Sommese, A., Wampler, C.: Finding all real points of a complex curve. Contemp. Math. **448**, 183 – 206 (2007)
17. Mario Besana, G., Di Rocco, S., Hauenstein, J.D., Sommese, A.J., Wampler, C.W.: Cell decomposition of almost smooth real algebraic surfaces. Numerical Algorithms **63**(4), 645–678 (2012)
18. Moroz, G.: Complexity of the resolution of parametric systems of polynomial equations and inequations. In: ISSAC 2006. pp. 246–253. ACM (2006)
19. Piret, K., Verschelde, J.: Sweeping algebraic curves for singular solutions. Journal of Computational and Applied Mathematics **234**(4), 1228 – 1237 (2010)
20. Rouillier, F., Roy, M.F., Safey El Din, M.: Finding at least one point in each connected component of a real algebraic set defined by a single equation. Journal of Complexity **16**(4), 716 – 750 (2000)
21. Schost, E.: Computing parametric geometric resolutions. Applicable Algebra in Engineering, Communication and Computing **13**(5), 349–393 (2003)
22. Sommese, A., Wampler, C.: The Numerical solution of systems of polynomials arising in engineering and science. World Scientific Press (2005)
23. Stewart, G.W.: Perturbation theory for the singular value decomposition. In: SVD and Signal Processing, II: Algorithms, Analysis and Applications. pp. 99–109. Elsevier (1990)
24. Tarski, A.: A decision method for elementary algebra and geometry. Fund. Math **17**, 210–239 (1931)
25. Wang, D.M., Xia, B.: Stability analysis of biological systems with real solution classification. In: Kauers, M. (ed.) ISSAC 2005. pp. 354–361. ACM (2005)
26. Weispfenning, V.: Comprehensive Gröbner bases. J. Symb. Comp. **14**, 1–29 (1992)
27. Wu, W., Reid, G.: Finding points on real solution components and applications to differential polynomial systems. In: ISSAC 2013. pp. 339–346. ACM (2013)
28. Wu, W., Reid, G., Feng, Y.: Computing real witness points of positive dimensional polynomial systems (2015), accepted for Theoretical computer science, `http://www.escience.cn/people/wenyuanwu`
29. Wu, W.T.: Basic principles of mechanical theorem proving in elementary geometries. J. Sys. Sci. and Math. Scis **4**(3), 207–235 (1984)
30. Yang, L., Xia, B.: Real solution classifications of a class of parametric semialgebraic systems. In: A3L 2005. pp. 281–289 (2005)