

# Analysis of parametric biological models with non-linear dynamics

Romain Testylier

VERIMAG, Joseph Fourier University \*  
Grenoble, France

Romain.Testylier@imag.fr

Thao Dang

CNRS/VERIMAG  
France

Thao.Dang@imag.fr

In this paper we present recent results on parametric analysis of biological models. The underlying method is based on the algorithms for computing trajectory sets of hybrid systems with polynomial dynamics. The method is then applied to two case studies of biological systems: one is a cardiac cell model for studying the conditions for cardiac abnormalities, and the second is a model of insect nest-site choice.

## 1 Introduction

Computer aided modelling and analysis have been intensively used to study biological systems. Different models have been combined in a unified framework to capture the complex behavior of biological phenomena. Recently, the hybrid systems paradigm has been applied to various biochemical networks with switching behavior. On the other hand, the main difficulty in biological model validation is the precision of predictions provided by the model, in comparison with experimental data. Quantitative information on kinetic parameters is thus hard to obtain and, in addition, experimental data are often of qualitative nature, which poses a major problem to traditional numerical methods. In this work, we try to address this problem by extending the numerical approach to systems with uncertain parameters.

More concretely, we exploit the ability of handling uncertainty of the reachability computation techniques for hybrid systems, a mathematical model for describing the interaction between a discrete and a continuous process. Roughly speaking, the goal of reachability analysis is to study the set of all possible trajectories of a dynamical system. Well-established results for piecewise linear systems are available and they have been successfully applied to numerous biological systems (for example [3, 11, 8, 14]). Nevertheless, nonlinear systems still remain a challenge.

In particular we focus the following computation problem: given a set of initial states in  $\mathbb{R}^n$ , compute the trajectory set of a parametric discrete-time dynamical system described by the following difference equation:  $\mathbf{x}(k+1) = \pi(\mathbf{x}(k), \mathbf{p})$  where  $\pi : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$  is a multivariate polynomial. This problem can be seen as an extension of numerical integration, that is, solving the equations of the dynamics with sets, that is  $x(k)$  and  $x(k+1)$  in the equation are subsets of  $\mathbb{R}^n$  (while they are points if we only need a single solution, as in traditional numerical integration).

Our interest in polynomial systems is motivated by their applicability in modeling a variety of phenomena in bio-chemical networks. To handle continuous-time models, a time discretization can be needed and conservativeness of approximation needs to be guaranteed (which is the topic of our current research). It is however important to note that discrete-time systems can also be directly used to model

---

\*This work is supported by the ANR project VEDECY.

many biological systems, since experimental data are often obtained by sampling the biochemical reaction outputs, and in addition, such models can be readily used for computer aided analysis and numerical simulation.

This problem for polynomial systems was previously considered in the work [23, 9], which was inspired by the techniques from Computer Aided Geometric Design (CADG). In this paper, we pursue the direction by exploiting further the special properties of a technique from CADG, namely the Bézier simplex and Bernstein expansion, we only need to solve linear programming (LP) problems instead of polynomial optimization problems. In this paper, with a view to biological models, we propose an extension of these techniques to systems with uncertain parameters. We also describe a new techniques specialized for multi-affine systems

The paper is organized as follows. In Section 2 we introduce basic definitions of set integration. We then focus on the problem of computing the image of a set by a parametric polynomial and propose two methods: one is an extension to parametric systems of the method using the Bernstein expansion, and the other method is based on a property of multi-affine functions. Some experimental results on two biological systems are reported in Section 5: one is a model of cardiac abnormalities, the other is a model of insect nest-site choice.

## 2 Set integration

Let  $\mathbb{R}$  denote the set of reals. Throughout the paper, vectors are often written using bold letters. Exceptionally, scalar elements of multi-indices, introduced later, are written using bold letters. Given a vector  $\mathbf{x}$ ,  $x_i$  denotes its  $i^{\text{th}}$  component.

We use  $\mathcal{B}$  to denote the unit box anchored at the origin, that is  $\mathcal{B} = [0, 1]^n$ . We use  $\pi$  to denote a vector of  $n$  functions such that for all  $i \in \{1, \dots, n\}$ ,  $\pi_i$  is an  $n$ -variate polynomial of the form  $\pi_i : \mathbb{R}^n \rightarrow \mathbb{R}$ .

We consider a parametric discrete-time dynamical system:

$$\mathbf{x}(k+1) = \pi(\mathbf{x}(k), \mathbf{p}) \quad (1)$$

where  $\mathbf{x} \in \mathbb{R}^n$  is the state variables,  $\mathbf{p} \in P \subseteq \mathbb{R}^m$  is the vector of uncertain parameters. The set  $P$  is called the parameter set. We assume that  $P$  is a convex polyhedron. The initial state  $\mathbf{x}(0)$  is inside some set  $X_0 \subset \mathbb{R}^n$ , and  $X_0$  is called the initial set.

Given a set  $X \subset \mathbb{R}^n$ , the image of  $X$  by  $\pi$  with all possible values of the parameters, denoted by  $\pi(X)$ , is defined as follows:  $\pi(X) = \{(\pi_1(\mathbf{x}, \mathbf{p}), \dots, \pi_n(\mathbf{x}, \mathbf{p})) \mid \mathbf{x} \in X, \mathbf{p} \in P\}$ .

When starting from a set of initial states, the dynamical system (1) generates a set of solutions, and at each step the set of all visited states is the image by  $\pi$  of the previous set. We thus focus on the problem of computing the image of a polyhedron  $X$  by the polynomial  $\pi$ .

One way to characterize this set of solutions is to use special convex polyhedra with fixed geometric form, called template polyhedra [22, 6]. Indeed, by choosing the templates one can control both the precision and the geometric complexity of the approximations. A template is a set of linear functions defined by an  $l \times n$  matrix  $H$  and a real-valued vector  $\mathbf{c} \in \mathbb{R}^l$ . A template polyhedron is then defined by considering the conjunction of the linear inequalities of the form  $\langle H, \mathbf{c} \rangle = \{\mathbf{x} \mid \bigwedge_{i=1, \dots, l} H^i \mathbf{x} \leq c_i\}$ . The vector  $\mathbf{c}$  is called a polyhedral coefficient vector.

The template matrix  $H$  is assumed to be given; to over-approximate the trajectory set, the polyhedral coefficient vector  $\mathbf{c} \in \mathbb{R}^l$  must satisfy

$$\pi(X) \subseteq \langle H, \mathbf{c} \rangle. \quad (2)$$

To determine the coefficients  $\mathbf{c} = (c_1, \dots, c_n)^T$  so that the template polyhedron  $\langle H, \mathbf{c} \rangle$  over-approximate  $\pi(X)$ , we can formulate the following optimization problems:

$$\forall i \in \{1, \dots, l\}, c_i = \max(\sum_{k=1}^n H_k^i \pi_k(\mathbf{x})) \quad (3)$$

$$\text{subj. to } \mathbf{x} \in X. \quad (4)$$

where  $H^i$  is the  $i^{\text{th}}$  row of the matrix  $H$  and  $H_k^i$  is its  $k^{\text{th}}$  element. Note that the above functions to optimize are polynomials. This polynomial optimization problem is computationally difficult (see for example [12] and references therein). We propose two alternative solutions:

- Use a linear relaxation of the above optimization problems, so that we can take advantage of well-developed linear programming techniques [21]. Indeed, the Bernstein expansion can be used to compute affine bound functions of polynomials, as shown in the next section.
- If  $\pi$  is multi-affine, we can exploit its particular properties that allows using corner analysis to reconstruct the set of all trajectories.

In the following section, we describe two methods for approximating the image of a box domain by a polynomial, and we then show how to extend these methods to polyhedral domains.

### 3 Approximating the image of a box domain

#### 3.1 Using the Bernstein expansion

The essence of this method, proposed in [13], for computing an affine bound function is to find a hyperplane that is close to *all* the control points, using linear least squares approximation. In this work, we extend this method to parametric polynomial functions. Before presenting the Bernstein expansion, we remark that computing convex lower bound functions for polynomials is a problem of great interest, especially in global optimization. The reader is also referred to [19] for more detail on the approach based on the Bernstein expansion.

To discuss the Bernstein expansion of polynomials, we use multi-indices of the form  $\mathbf{i} = (\mathbf{i}_1, \mathbf{i}_2, \dots, \mathbf{i}_n)$  where each  $\mathbf{i}_j$  is a non-negative integer. Given two multi-indices  $\mathbf{i}$  and  $\mathbf{d}$ , we write  $\mathbf{i} \leq \mathbf{d}$  if for all  $j \in \{1, \dots, n\}$ ,  $\mathbf{i}_j \leq \mathbf{d}_j$ . Also, we write  $\frac{\mathbf{i}}{\mathbf{d}}$  for  $(\mathbf{i}_1/\mathbf{d}_1, \mathbf{i}_2/\mathbf{d}_2, \dots, \mathbf{i}_n/\mathbf{d}_n)$  and  $\binom{\mathbf{i}}{\mathbf{d}}$  for the product of binomial coefficients  $\binom{\mathbf{i}_1}{\mathbf{d}_1} \binom{\mathbf{i}_2}{\mathbf{d}_2} \dots \binom{\mathbf{i}_n}{\mathbf{d}_n}$ .

The parametric polynomial  $\pi : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$  can be represented using the power base as follows:

$$\pi(\mathbf{x}, \mathbf{p}) = \sum_{\mathbf{i} \in I_{\mathbf{d}}} \mathbf{a}_{\mathbf{i}}(\mathbf{p}) \mathbf{x}^{\mathbf{i}}$$

where  $\mathbf{a}_{\mathbf{i}}$  is a function  $\mathbb{R}^m \rightarrow \mathbb{R}$ ;  $\mathbf{i}$  and  $\mathbf{d}$  are two multi-indices of size  $n$  such that  $\mathbf{i} \leq \mathbf{d}$ ;  $I_{\mathbf{d}}$  is the set of *all* multi-indices  $\mathbf{i} \leq \mathbf{d}$ , that is  $I_{\mathbf{d}} = \{\mathbf{i} \mid \mathbf{i} \leq \mathbf{d}\}$ ;  $\mathbf{x}^{\mathbf{i}}$  represents the monomial  $x_1^{\mathbf{i}_1} x_2^{\mathbf{i}_2} \dots x_n^{\mathbf{i}_n}$ . The multi-index  $\mathbf{d}$  is called the *degree* of  $\pi$ .

The polynomial  $\pi$  can also be represented using the Bernstein expansion. For  $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$ , the  $i^{\text{th}}$  Bernstein polynomial of degree  $\mathbf{d}$  is defined as follows:

$$\mathcal{B}_{\mathbf{d}, \mathbf{i}}(\mathbf{x}) = \beta_{\mathbf{d}_1, \mathbf{i}_1}(x_1) \dots \beta_{\mathbf{d}_n, \mathbf{i}_n}(x_n)$$

where for a real number  $y$ ,  $\beta_{\mathbf{d}, \mathbf{i}_j}(y) = \binom{\mathbf{d}_j}{\mathbf{i}_j} y^{\mathbf{i}_j} (1 - y^{\mathbf{d}_j - \mathbf{i}_j})$ .

Then, for all  $\mathbf{x} \in \mathcal{B} = [0, 1]^n$ , the polynomial  $\pi$  can be written using the Bernstein expansion as follows:

$$\pi(\mathbf{x}) = \sum_{\mathbf{i} \in I_{\mathbf{d}}} \mathbf{b}_{\mathbf{i}}(\mathbf{p}) \mathcal{B}_{\mathbf{d}, \mathbf{i}}(\mathbf{x})$$

where for each  $\mathbf{i} \in I_{\mathbf{d}}$  the Bernstein coefficient  $\mathbf{b}_{\mathbf{i}}(\mathbf{p})$  is defined as:

$$\mathbf{b}_{\mathbf{i}}(\mathbf{p}) = \sum_{\mathbf{j} \leq \mathbf{i}} \frac{\binom{\mathbf{i}}{\mathbf{j}}}{\binom{\mathbf{d}}{\mathbf{j}}} \mathbf{a}_{\mathbf{j}}(\mathbf{p}). \quad (5)$$

With respect to our problem of computing the image of a set by a polynomial, the Bernstein expansion is of particular interest since their coefficients allow to derive useful geometric properties. We now show that these coefficients can be used to efficiently compute an affine approximation of the polynomial  $\pi$ .

We denote by  $I_{\mathbf{d}} = \{\mathbf{i}^j \mid 1 \leq j \leq n_b\}$  be the set of all the multi-indices,  $n_b$  is thus their number. The set of all control points are denoted similarly. Let  $A$  be a matrix of size  $n_b \times (n + 1)$  ( $n$  is the number of state variables of the dynamical systems in question) such that its elements are defined as follows. For all  $1 \leq j \leq n_b$  and  $1 \leq k \leq n$ ,

$$A_k^j = \frac{\mathbf{i}_k^j}{\mathbf{d}_k}$$

and  $A_{n+1}^j = 1$ .

We pick the centroid point  $\mathbf{p}_c$  in the parameter set  $P$ . Let  $\zeta$  be the solution of the following linear least squares approximation problem:

$$A^T A \zeta = A^T \mathbf{b}(\mathbf{p}_c).$$

Then, the affine function

$$\tilde{l}(\mathbf{x}) = \sum_{k=1}^n \zeta_k \mathbf{x}_k + \zeta_{n+1}$$

corresponds to the "median" axis of the convex hull of all the control points. It thus suffices to shift it downward by the amount:

$$\delta = \max\{\tilde{l}\left(\frac{\mathbf{i}^j}{\mathbf{d}}\right) - \mathbf{b}^j(\mathbf{p}) \mid 0 \leq j \leq n_b \wedge \mathbf{p} \in P\}. \quad (6)$$

If the function  $\pi$  is linear in the parameters  $\mathbf{p}$ , then the above optimization problem is a set of linear programs and can thus be solved efficiently. This results in a lower bound function

$$l(\mathbf{x}) = \tilde{l}(\mathbf{x}) - \delta, \text{ for all } \mathbf{x} \in \mathcal{B}.$$

We now show how the above affine bound functions can be used to solve the optimization problems (3) in order to determine the coefficients of a template polyhedron over-approximating the reachable set. The functions to optimize in (3) can be seen as the compositions of polynomials  $\pi_k$ . Since every coefficient  $H_k^i$  is constant, each

$$\sigma^i(\mathbf{x}) = \sum_{k=1}^n H_k^i \pi_k(\mathbf{x})$$

is simply a polynomial and we can compute its bound functions. The template polyhedral coefficients can hence be computed by solving the following optimization problems:

$$\forall i \in \{1, \dots, l\}, c_i = \max(\sigma^i(\mathbf{x})) \text{ subj. to } \mathbf{x} \in X. \quad (7)$$

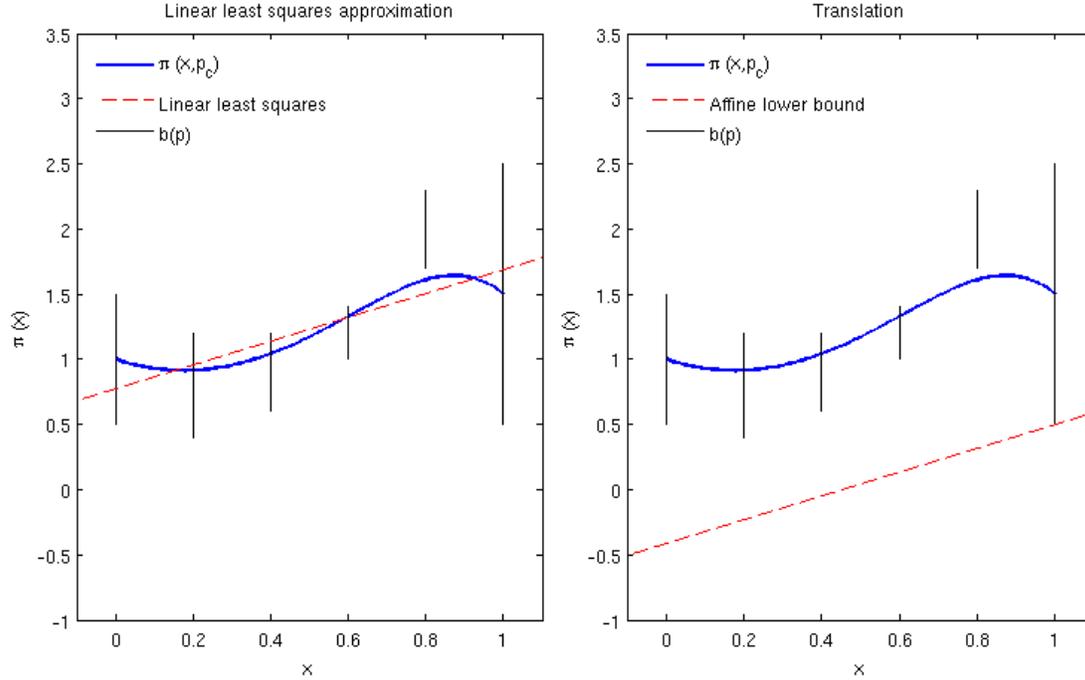


Figure 1: Computation of the affine lower bound function using linear least square approximation; the vertical line segments represent the intervals of the Bernstein coefficient values.

**Example.** We illustrate the method for computing the bound functions with an example. We consider a one dimensional polynomial of degree 5:

$$\pi(\mathbf{x}) = \mathbf{p}(1 - \mathbf{x} + 2\mathbf{x}^4) + 3\mathbf{x}^2 - \mathbf{x}^3 - 2.5\mathbf{x}^5$$

where  $\mathbf{p}$  is a scalar parameter included in  $[0.5, 1.5]$ . Using the equation (5), Bernstein coefficients are the following:

$$\mathbf{b}(\mathbf{p}) = (\mathbf{p}, 0.8\mathbf{p}, 0.6\mathbf{p} + 0.3, 0.4\mathbf{p} + 0.8, 0.6\mathbf{p} + 1.4, 2\mathbf{p} - 0.5).$$

Next we compute the matrix  $A$  and the vector  $\mathbf{b}(\mathbf{p}_c)$  where the centroid  $\mathbf{p}_c$  is equal to 1:

$$A = \begin{pmatrix} 0 & 0.2 & 0.4 & 0.6 & 0.8 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix},$$

$$\mathbf{b}(\mathbf{p}_c) = \mathbf{b}(1) = (1, 0.8, 0.9, 1.2, 2, 1.5),$$

which gives us the following linear system:

$$\begin{pmatrix} 2.2 & 3 \\ 3 & 6 \end{pmatrix} \begin{pmatrix} \zeta_1 \\ \zeta_2 \end{pmatrix} = \begin{pmatrix} 4.34 \\ 7.40 \end{pmatrix}.$$

By solving this linear equality system, we obtain following least square approximation

$$\tilde{l}(\mathbf{x}) = \zeta_1 \mathbf{x} + \zeta_2 = 0.9143\mathbf{x} + 0.7762.$$

Then it follows from (6) that  $\delta = 1.1905$  and we get the affine lower bound

$$l(\mathbf{x}) = \tilde{l}(\mathbf{x}) - \delta = 0.9143\mathbf{x} - 0.4143.$$

The left plot in Figure 1 shows the linear least square approximation computed for the polynomial in question. The other one represents the final affine function computed after shifting downward the previous affine function.

### 3.2 Using the property of multi-affine functions

In this section, we propose a method specialized for multi-affine systems, a particular case of polynomial systems. Multi-affine systems are systems composed by polynomials which are affine by each of their variables, i.e. if  $\mathbf{d}$  is the degree of an affine system  $\pi$ ,  $\mathbf{d}_k \leq 1$  for all  $k \in \{1, \dots, n\}$ . We will exploit the following property of such systems to achieve better computational efficiency.

**Theorem 1** *Given a hyper-rectangle  $X \subseteq \mathbb{R}^n$ , let  $V_X$  be the set of its vertices. If  $\pi$  is a multi-affine function, then*

$$\pi(X) \subseteq \text{conv}\{\pi(\mathbf{v}) \mid \mathbf{v} \in V_X\}$$

To prove the theorem, we first prove an intermediate result.

We denote the line segment connecting two points  $\mathbf{x}$  and  $\mathbf{x}'$  by  $s(\mathbf{x}, \mathbf{x}')$ .

**Lemma 1** *If two points  $\mathbf{x}$  and  $\mathbf{x}'$  that differ only in one coordinate (that is there exists only one axis  $i \in \{1, \dots, n\}$  such that  $x_i \neq x'_i$  and for all other axes  $j \neq i$   $x_j = x'_j$ ), then for any point  $\mathbf{y}$  in the line segment  $s(\mathbf{x}, \mathbf{x}')$ , its image  $\pi(\mathbf{y})$  can be written as a linear combination of  $\pi(\mathbf{x}')$  and  $\pi(\mathbf{x})$ .*

*Proof.* It is easy to see that any two points in  $s(\mathbf{x}, \mathbf{x}')$  also differ only in the axis  $i$ , (that is  $s(\mathbf{x}, \mathbf{x}')$  is parallel to the axis  $x_i$ ). The  $i^{\text{th}}$  coordinate of any point  $\mathbf{y} \in s(\mathbf{x}, \mathbf{x}')$  can be expressed as  $y_i = \lambda_1 x_i + \lambda_2 x'_i$  where  $\lambda_1, \lambda_2 \geq 0$  and  $\lambda_1 + \lambda_2 = 1$ . In addition, if the domain of the function  $\pi$  is restricted to the line segment  $s(\mathbf{x}, \mathbf{x}')$ , only the variable  $x_i$  varies while all the variables remain unchanged and the function  $\pi$  hence becomes linear in  $x_i$ . It then follows that  $\pi(\mathbf{y})$  can be written as a linear combination of  $\pi(\mathbf{x}')$  and  $\pi(\mathbf{x})$ . ■

We proceed with Theorem 1. Let  $\mathbf{x}$  is a point inside  $X$ . Let  $\mathbf{y}$  and  $\mathbf{z}$  be the projections of  $\mathbf{x}$  on two opposite faces of  $X$  perpendicular to the axis  $x_n$ . It follows from Lemma 1 that  $\pi(\mathbf{x})$  can be written as a linear combination of  $\pi(\mathbf{y})$  and  $\pi(\mathbf{z})$ . Furthermore,  $\mathbf{y}$  can now be treated as a point inside a hyper-rectangle in  $(n-1)$  dimensions  $(x_1, \dots, x_{n-1})$ , and by successive projections until dimension  $x_1$  we can prove that  $\pi(\mathbf{x})$  can be written as a linear combination of the vertices of  $X$ . ■

Another proof of this well-known property of multi-affine functions can be found in [4]

Note that the above theorem is only applicable to the sets which are axis-aligned hyper-rectangles. Hence, even if the initial set satisfies this condition, after the application of  $\pi$ , the resulting set is a general polytope and we need to approximate it by an axis-aligned hyper-rectangle. Using the theorem, to compute  $\pi(X)$  it suffices to compute the images of the vertices of  $X$  and then take the convex hull of the resulting points.

Before continuing, we remark that a number of different methods for the multi-affine systems have also been developed [1, ?].

## 4 Approximating the image of a polyhedral domain

As mentioned earlier, the above described methods can be applied only when the set  $X$  is inside the unit box  $\mathcal{B}$  anchored at the origin. To extend it to polyhedral domains, we transform the polyhedra to the unit box by two methods: (1) via an (oriented) box approximation, and (2) by rewriting the polynomials using a change of variables.

If we over-approximate  $X$  with a box  $B$ , it is then possible to derive a formula expressing the Bernstein coefficients of  $\pi$  over  $B$ . However, this formula is complex and its representation and evaluation can become expensive.

We alternatively consider the composition of the polynomial  $\pi$  with an affine transformation  $\tau$  that maps the unit box to  $B$ . The functions resulting from this composition are still polynomials, for which we can compute their bound functions over the unit box, using the formula (5) of the Bernstein expansion. This is explained more formally in the following.

Let  $B$  be the bounding box of the polyhedron  $X$ , that is, the smallest box that includes  $X$ . The affine function  $\tau$  that maps the unit box  $\mathcal{B}$  to  $B$  can be easily defined as:  $\tau(\mathbf{x}) = \text{diag}(\lambda)\mathbf{x} + \mathbf{g}$  where  $\mathbf{g} \in \mathbb{R}^n$  such that  $g_i = l_i$ , and  $\text{diag}(\lambda)$  is a  $n \times n$  diagonal matrix with the elements on the diagonal defined as follows: for each  $i \in \{1, \dots, n\}$ ,  $\lambda_i = h_i - l_i$ . The composition  $\gamma = (\pi \circ \tau)$  is defined as  $\gamma(\mathbf{x}) = \pi(\tau(\mathbf{x}))$ . Note that  $\gamma = \pi \circ \tau$ . Then,  $\pi(X) \subseteq \gamma(\mathcal{B})$ .

Since multi-affine functions are not closed under the above composition. They can become quadratic. In this case, we use the following blossoming principle to transform a polynomial to an affine function.

**Theorem 2 (Blossoming principle)** *For any polynomial  $\pi : \mathbb{R}^n \rightarrow \mathbb{R}^m$  of degree  $d$ , there is a unique symmetric multi-affine map  $w : \mathbb{R}^n \times \dots \times \mathbb{R}^n \rightarrow \mathbb{R}^m$  such that for all  $\mathbf{x} \in \mathbb{R}^n$*

$$w(\mathbf{x}, \dots, \mathbf{x}) = \pi(\mathbf{x}).$$

The proof of the above theorem is standard and can be found in [15]. We can thus compute the image  $\pi(B)$  by the convex hull of all the images of the vertices of  $X$  by the multi-affine function  $w$ . Nevertheless, it is important to note that when  $\mathbf{x}$  inside an rectangle, the arguments of the blossom varies only on the diagonal of the corresponding rectangle in the augmented space  $\mathbb{R}^{nd}$ . Thus the convex hull is an over-approximation.

## 5 Experimental results

We implemented the above described algorithms, and their time efficiency was also evaluated by considering a number of randomly generated polynomials, which shows that the algorithms can handle general polynomial systems in up to 11 dimensions with reasonable computation time. In the following, we demonstrate their application to two biological systems.

### 5.1 A model of insect nest-site choice

We study a model of a decision making mechanism used by a swarm of honeybees to select one among many different nest-sites [5]. This is built upon classical mathematical models of the spread of diseases, information and beliefs. The bee population is divided into 5 groups:

- $X$ : neutral bees that have not chosen a site

- $Y_1$ : evangelic bees dancing for the site 1
- $Y_2$ : evangelic bees dancing for the site 2
- $Z_1$ : non-evangelical bees which have been converted to the site 1
- $Z_2$ : non-evangelical bees which have been converted to the site 2

The bees dance not only to advertise the quality of a site but also to transmit to the other bees information about the direction and distance to the site. Non-evangelical bees stay idle but may take up dancing when they are stimulated by a dancing bee. For simplicity, we also use the names of the groups to denote the respective numbers of individuals in each group.

The equations describing the evolutions of the variables are as follows:

$$\begin{aligned}
\dot{X}(t) &= -\beta_1 X(t)Y_1(t) - \beta_2 X(t)Y_2(t), \\
\dot{Y}_1(t) &= \beta_1 X(t)Y_1(t) - \gamma Y_1(t) + \delta \beta_1 Y_1(t)Z_1(t) + \alpha \beta_1 Y_1(t)Z_2(t), \\
\dot{Y}_2(t) &= \beta_2 X(t)Y_2(t) - \gamma Y_2(t) + \delta \beta_2 Y_2(t)Z_2(t) + \alpha \beta_2 Y_2(t)Z_1(t), \\
\dot{Z}_1(t) &= \gamma Y_1(t) - \delta \beta_1 Y_1(t)Z_1(t) - \alpha \beta_2 Y_2(t)Z_1(t), \\
\dot{Z}_2(t) &= \gamma Y_2(t) - \delta \beta_1 Y_2(t)Z_2(t) - \alpha \beta_2 Y_1(t)Z_2(t),
\end{aligned}$$

where  $\beta_1$  and  $\beta_2$  are scalar parameters representing a measure of how vigorously the bees dance for the sites 1 and 2 respectively;  $\alpha$  is the parameter representing the proportionality of switching back to the neutral state, and  $\gamma$  is the proportionality of conversion from the dancing state to the idle state. The proportionality of conversion from the neutral state to any site  $Y_i$  is 1 and from the idle state to  $Y_i$  is  $\delta$ .

In this case study, we want to study the influence of the parameter  $\beta_2$  on the possibility of achieving a consensus on nest-site choice.

Using the Euler discretization method we obtain the following discrete-time dynamics.

$$\begin{aligned}
X(k+1) &= X(k) + h(-\beta_1 X(k)Y_1(k) - \beta_2 X(k)Y_2(k)), \\
Y_1(k+1) &= Y_1(k) + h(\beta_1 X(k)Y_1(k) - \gamma Y_1(k) + \delta \beta_1 Y_1(k)Z_1(k) + \alpha \beta_1 Y_1(k)Z_2(k)), \\
Y_2(k+1) &= Y_2(k) + h(\beta_2 X(k)Y_2(k) - \gamma Y_2(k) + \delta \beta_2 Y_2(k)Z_2(k) + \alpha \beta_2 Y_2(k)Z_1(k)), \\
Z_1(k+1) &= Z_1(k) + h(\gamma Y_1(k) - \delta \beta_1 Y_1(k)Z_1(k) - \alpha \beta_2 Y_2(k)Z_1(k)), \\
Z_2(k+1) &= Z_2(k) + h(\gamma Y_2(k) - \delta \beta_1 Y_2(k)Z_2(k) - \alpha \beta_2 Y_1(k)Z_2(k)),
\end{aligned}$$

We choose a discretization time  $h = 0.01$ . In the following,  $\beta_1 N = 1.0$ , the initial set is a small box centered at  $(N, 0, 0, 0, 0)$ , where  $N$  is the total number of honeybees arbitrary fixed to 1000.

Figure 2 and Figure 3 show the evolution of the proportions of converted honeybees  $((Y_i + Z_i)/N)$  during 6000 iterations, with different interval values of the parameter  $\beta_2$ . The blue sets (in lighter color) correspond to the proportions of bees supporting the site 1 and the red sets (in darker color) correspond to those supporting the site 2. These results were obtained using the method for multi-affine systems which took about 7.5s

First we consider that at the beginning no honeybees dance for the site 2, in other words  $\beta_2 = 0$ . After 300 iterations the site 2 is discovered and its propaganda begins. We can observe two main compartments: a consensus and the absence of consensus. We consider that there is a consensus if the distance between the proportions of honeybees supporting the site 1 or 2 is large and tends to increase.

Figure 2 shows the analysis using the parameters  $\alpha = 0.7$ ,  $\gamma = 0.3$  and  $\beta_2 \in [1, 1.2]$ . We observe that  $\beta_2$  is marginally superior to  $\beta_1$ , however we can see that there is no consensus between the two evangelic groups.

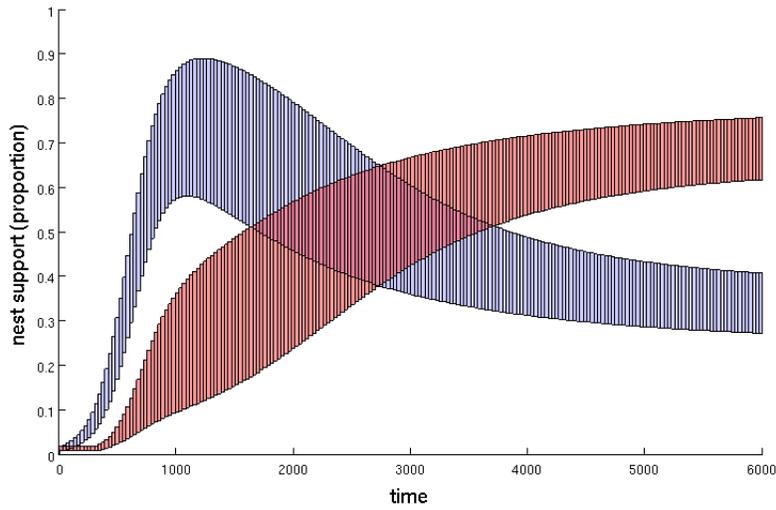


Figure 2: No consensus is observed with  $\alpha = 0.7, \gamma = 0.3$  and  $\beta_2 \in [1, 1.2]$ .

Figure 3 shows that a consensus can be observed if the measure of how vigorously the bees dance for sites 2 is increased. In this analysis the parameters stay the same except for  $\beta_2 \in [1.5, 2.0]$ . We observe a clear consensus for the site 2 despite its lateness.

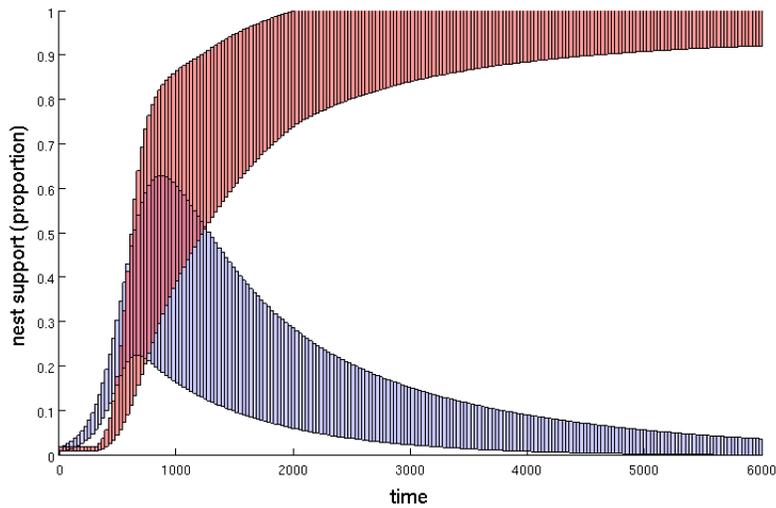


Figure 3: A consensus for the site 2 is observed with  $\alpha = 0.7, \gamma = 0.3$  and  $\beta_2 \in [1.5, 2]$ .

Another way to obtain a consensus without modifying  $\beta_2$  is to reduce  $\alpha$  which corresponds to the proportionality of honeybees reconversion to another site. When  $\alpha = 0.2$ , the late propaganda of the site 2 leads to a consensus to choose the site 1 as shown in Figure 4.

Finally, to compare the two proposed methods, Figure 5 shows the projection of the reachable sets on the variables  $Y_1$  and  $Y_2$  during the first 150 iterations, with  $\alpha = 0.7, \gamma = 0.3$  and  $\beta_2 = 1.2$ . We plot

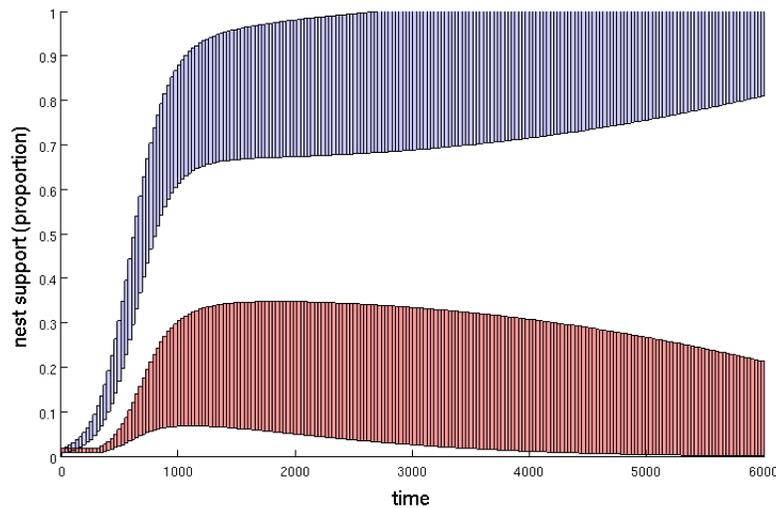


Figure 4: A consensus for the site 1 is observed with  $\alpha = 0.2$ ,  $\gamma = 0.3$  and  $\beta_2 \in [1, 1.2]$ .

the results obtained by the two methods and we can see that the sets computed using the method for multi-affine systems are larger than those computed using the Bernstein expansion. We observe a gain of precision with the Bernstein method, however the computation time of this method is 25.06s and is more than 250 times superior than the computation time of the other method which is 0.09s.

## 5.2 A model of cardiac abnormalities

The second case study comes from a simplified model of cardiac cells [14] which describes the conditions under which cardiac abnormalities (such as, ventricular tachycardia and fibrillation) can arise. The behavior of cardiac cells can be modelled using a Detailed Ionic Model (DIM) which are differential equation models of reaction-diffusion type. Such models are complex with a lot of parameters. An alternative model is called Minimal Resistor Model (MRM), the parameters of which can be identified from either experimental data or from DIM-based simulation results. The main advantage of the MRM model is that it exhibits more accurate mesoscopic behavior and additionally allows faster simulations. From the MRM model, the authors of [14] derived an equivalent Minimal Conductance Model (MCM) which is exactly in form of a genetic regulatory network model (GRM). By approximating slow sigmoidal switches in the MCM model with a succession of ramps, a Hybrid Automaton ([16]) is obtained.

In this case study, we consider the Hybrid Automaton model and the following problem: find the parameter ranges which correspond to the behaviors where the cardiac cells lose their excitability. These ranges can then be used to infer the corresponding parameter ranges in the DIM model. The hybrid automaton has two modes, denoted by *loc1* and *loc2*.

- *loc1*:

$$\begin{aligned} \frac{du}{dt} &= e - g_1 u \\ \frac{dv}{dt} &= (1 - v)g_1 \end{aligned}$$

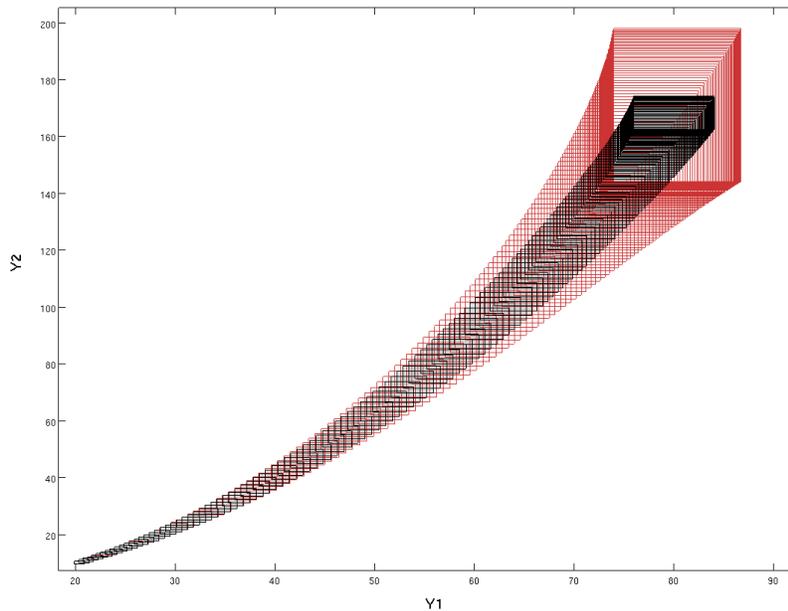


Figure 5: Reachability analysis during 150 iterations using the two reachability methods, the black outlines set are computed with the Bernstein expansion method, the red outlines set correspond to the result using the method for multi-affine systems.

- *loc2*:

$$\begin{aligned} \frac{du}{dt} &= e - g_2 u \\ \frac{dv}{dt} &= -vg_2; \end{aligned}$$

The transition guard from *loc1* to *loc2* is  $u \geq 0.06$ . In this model, an *action potential* is a change in cell transmembrane potential  $u$  as a response to an external stimulus (current)  $e$  defined by  $e = 0.66$  if  $t < 0.25$  and  $e = 0$  if  $t \geq 0.25$ . The initial state corresponds to  $u = 0$ . In the above equations,  $g_1$  and  $g_2$  are parameters such that  $1 < g_1 < 180$  and  $1 < g_2 < 10$ . The safety property we are interested in is stated as follows: the system, when at location *loc2*, will never reach a state where  $u \geq 0.13$ . To this end, we add a fictitious location *loc3* and let the transition guard from *loc2* to *loc3* be  $u \geq 0.13$ . We also consider the two parameters as two additional variables with derivatives equal to 0. The problem is now to compute a set of possible values of  $g_1$  and  $g_2$  which guarantee that the system does not reach *loc3*.

The model hence becomes a Hybrid Automaton with 4 variables and their dynamics are described by multi-affine differential equations, which we discretize in time. Using backward reachability analysis starting from the unsafe set, we can over-approximate the set of  $(g_1, g_2)$  for which the system can reach the guard of the transition to *loc3*. The figure 6 shows the projection on the variables  $u$ ,  $g_1$ ,  $g_2$  of the reachable set, starting from the guard leading to the location *loc3*. Its intersection with the hyperplane defined by  $u = 0$  (in grey) corresponds to the set of unsafe values of  $g_1$  and  $g_2$ . This computation uses dynamical templates and static boxes, and the computation time is 1.52s. The complement of this set is the set of safe parameter values.

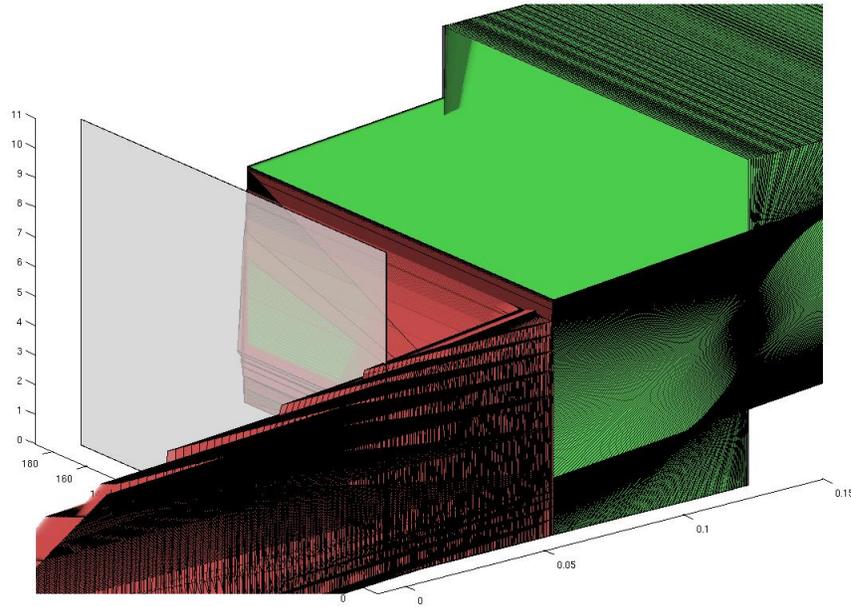


Figure 6: Backward reachability analysis, the change of color represents the change of location.

## 6 Related work

Our reachability analysis approach is similar to a number of existing ones for continuous and hybrid systems in the use of linear approximation. Its novelty resides in the efficient way of computing linear approximations. Indeed, a common method to approximate a non-linear function by a piecewise linear one, as in the hybridization approach [2, 10] for hybrid systems, requires non-linear optimization.

Besides constrained global optimization, other important applications of the Bernstein expansion include various control problems [18] (in particular, robust control). The approximation of the range of a multivariate polynomial over a box and a polyhedron is also used in program analysis and optimization (for example [17, 7]). In the hybrid systems verification, polynomial optimization is used to compute barrier certificates [20].

## 7 Conclusion

The reachability computation methods we proposed in this paper combine the ideas from optimization and the properties of polynomials. These results can be readily applicable to hybrid systems with polynomial continuous dynamics. We are currently working on the problem of parameter synthesis using the analysis techniques from robust control. The next step will be a more intensive evaluation of the methods on biological systems.

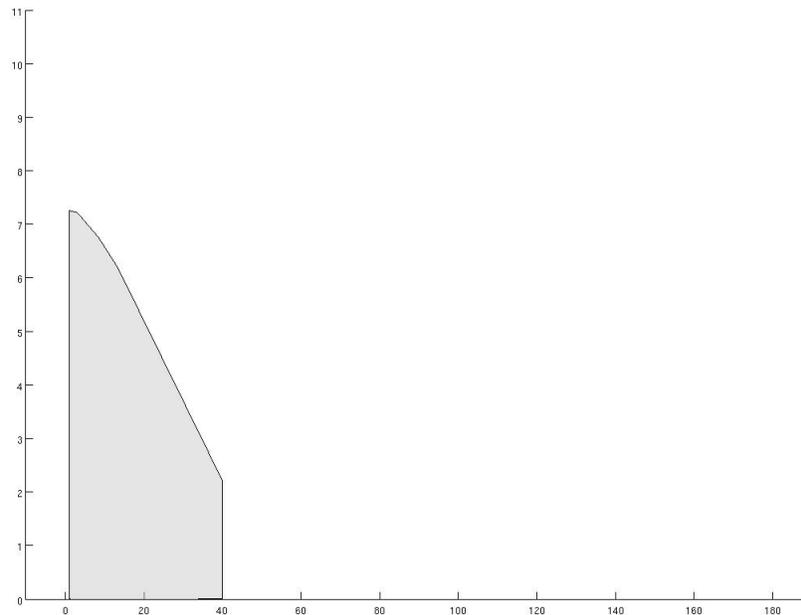


Figure 7: Over-approximation of the parameter values  $(g_1, g_2)$  that lead to the unsafe location.

## References

- [1] Grgory Batt A, Boyan Yordanov B, Ron Weiss C, Calin Belta A & Calin Belta A (2007): *Robustness Analysis and Tuning of Synthetic Gene Networks*. doi:10.1093/bioinformatics/btm362.
- [2] E. Asarin, T. Dang & A. Girard (2007): *Hybridization methods for the analysis of nonlinear systems*. *Acta Informatica*. 43(7), pp. 451–476, doi:10.1007/s00236-006-0035-7.
- [3] G. Batt, H. de Jong, J. Geiselman, M. Page, D. Ropers & D. Schneider (2005): *Qualitative analysis and verification of hybrid models of genetic regulatory networks : Nutritional stress response in Escherichia coli*. In M. Morari & L. Thiele, editors: *Hybrid Systems: Computation and Control*, LNCS 3414, Springer-Verlag, pp. 134–150, doi:10.1007/978-3-540-31954-2\_9.
- [4] Calin Belta & Luc C. G. J. M. Habets (2006): *Controlling a Class of Nonlinear Systems on Rectangles*. *IEEE Transactions on Automatic Control* 51, pp. 1749–1759, doi:10.1109/TAC.2006.884957.
- [5] N.F. Britton, N. R. Franks, S. C. Pratt & T. D. Seeley (2002): *Deciding on a new home: how do honeybees agree?* In: *Proceedings of the Royal Society of London Series B - Biological Sciences*, 269(1498), pp. 1383–1388, doi:10.1098/rspb.2002.2001.
- [6] Liqian Chen, Antoine Miné, Ji Wang & Patrick Cousot (2009): *Interval Polyhedra: An Abstract Domain to Infer Interval Linear Relationships*. In: *SAS, Lecture Notes in Computer Science 5673*, Springer, pp. 309–325, doi:10.1007/978-3-642-03237-0\_21.
- [7] F. Clauss & I.Yu. Chupaeva (2004): *Application of symbolic approach to the bernstein expansion for program analysis and optimization*. *Program. Comput. Softw.* 30(3), pp. 164–172.
- [8] Thao Dang, Colas Le Guernic & Oded Maler (2011): *Computing reachable states for nonlinear biological models*. *Theoretical Computer Science*, doi:10.1016/j.tcs.2011.01.014.

- [9] Thao Dang & David Salinas (2009): *Image computation for polynomial dynamical systems using the Bernstein expansion*. In Ahmed Bouajjani & Oded Maler, editors: *Computer Aided Verification, CAV 2009, Lecture Notes in Computer Science 5643*, Springer, pp. 219–232, doi:10.1007/978-3-642-02658-4\_19.
- [10] Thao Dang & Romain Testylier (2011): *Hybridization Domain Construction using Curvature Estimation*. In: *Proceedings HSCC 2011*, ACM, doi:10.1145/1967701.1967721.
- [11] S. Drulhe, G. Ferrari-Trecate & H. de Jong (2008): *The switching threshold reconstruction problem for piecewise affine models of genetic regulatory networks*. *IEEE Transactions on Automatic Control and IEEE Transactions on Circuits and Systems* 53(1), pp. 153–165, doi:10.1109/TAC.2007.911326.
- [12] I.A. Fotiou, P. Rostalski, P.A. Parrilo & M. Morari (2006): *Parametric Optimization and Optimal Control using Algebraic Geometriy Methods*. *International Journal of Control* 79(11), pp. 1340–1358, doi:10.1080/00207170600726592.
- [13] J. Garloff & A.P. Smith (2008): *Rigorous Affine Lower Bound Functions for Multivariate Polynomials and their Use in Global Optimisation*. In: *Proceedings of the 1st International Conference on Applied Operational Research, Tadbir Institute for Operational Research, Systems Design and Financial Services, Lecture Notes in Management Science 1*, pp. 199–211.
- [14] Radu Grosu, Grégory Batt, Flavio H. Fenton, James Glimm, Colas Le Guernic, Scott A. Smolka & Ezio Bartocci (2011): *From Cardiac Cells to Genetic Regulatory Networks*. In: *CAV, Lecture Notes in Computer Science 6806*, Springer, pp. 396–411, doi:10.1007/978-3-642-22110-1\_31.
- [15] H. Prautzsch and W. Boehm (2002): *Box splines*. In G. Farin, J. Hoschek, M.S. Kim, editor: *Handbook of Computer Aided Geometric Design*, Elsevier Science Publishers B.V., North-Holland, Amsterdam, pp. 255–282, doi:10.1016/B978-044451104-1/50011-3.
- [16] T. A. Henzinger (1996): *The theory of hybrid automata*. In: *LICS '96: Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science*, p. 278, doi:10.1109/LICS.1996.561342.
- [17] I. Tchoupaeva (2004): *A symbolic approach to bernstein expansion for program analysis and optimization*. In: *13th International Conference on Compiler Construction, CC 2004*, Springer, pp. 120–133, doi:10.1007/978-3-540-24723-4\_9.
- [18] J. Garloff (2004): *Application of Bernstein Expansion to the Solution of Control Problems*. In J. Vehi and M. A. Sainz, editor: *Workshop on Applications of Interval Analysis to Systems and Control*, pp. 421–430.
- [19] J. Garloff and A.P. Smith (2005): *A comparison of methods for the computation of affine lower bound functions for polynomials*. In C. Jermann, A. Neumaier & D. Sam, editors: *Global Optimization and Constraint Satisfaction*, LNCS, Springer, pp. 71–85, doi:10.1007/11425076\_6.
- [20] Stephen Prajna & Ali Jadbabaie (2004): *Safety Verification of Hybrid Systems Using Barrier Certificates*. In Rajeev Alur & George J. Pappas, editors: *Hybrid Systems: Computation and Control, Lecture Notes in Computer Science 2993*, Springer, pp. 477–492, doi:10.1007/978-3-540-24743-2\_32.
- [21] S. Boyd and S. Vandenberghe (2004): *Convex optimization*. Cambridge Uni. Press.
- [22] S. Sankaranarayanan, H. Sipma & Z. Manna (2005): *Scalable analysis of linear systems using mathematical programming*. In: *Verification, Model-Checking and Abstract-Interpretation (VMCAI 2005)*, LNCS 3385, Springer, doi:10.1007/978-3-540-30579-8\_2.
- [23] Thao Dang (2006): *Approximate Reachability Computation for Polynomial Systems*. In João P. Hespanha and Ashish Tiwari, editor: *Hybrid Systems: Computation and Control HSCC, Lecture Notes in Computer Science 3927*, Springer, pp. 138–152, doi:10.1007/11730637\_13.