

RESEARCH

Open Access



Polynomial superlevel set representation of the multistationarity region of chemical reaction networks

AmirHosein Sadeghimanesh* and Matthew England*

*Correspondence:

Amirhossein.

Sadeghimanesh@coventry.ac.uk;

Matthew.England@coventry.ac.uk

Research Centre
for Computational Sciences
and Mathematical Modelling,
Coventry University, Coventry, UK

Abstract

In this paper we introduce a new representation for the multistationarity region of a reaction network, using polynomial superlevel sets. The advantages of using this polynomial superlevel set representation over the already existing representations (cylindrical algebraic decompositions, numeric sampling, rectangular divisions) is discussed, and algorithms to compute this new representation are provided. The results are given for the general mathematical formalism of a parametric system of equations and so may be applied to other application domains.

Keywords: Polynomial superlevel set, Steady states, Multistationarity, Parameter analysis

Introduction

Many problems in applied sciences can be modelled by a parametric polynomial system, and therefore to solve such problems we must be able to explore the properties of these systems. In particular, we often seek to identify areas of the parameter space where a property holds. The contribution of this paper is a new methodology for exploring these.

Motivation: multistationarity regions of CRNs

We are motivated by the problem of understanding the multistationarity behavior of a Chemical Reaction Network (CRN). In a CRN, variables represent the concentrations of the species. These change as time passes and are studied as part of the field of dynamical systems. This is of polynomial type when the kinetics is assumed to follow the mass action rules. The equilibria of such a dynamical system are therefore the solutions to a system of polynomial equations. However, the coefficients of the terms on the polynomials may involve some parameters. These parameters are usually the rates under which a reaction occurs and the total amounts (thought of as a dependency on the initial concentration of the species).

Both the variables and parameters can only attain non-negative real values. A network is called multistationary if there exists a choice of parameters for which the network has



more than one equilibrium. There are already many algorithms developed for answering the binary question of whether a system can exhibit multistationarity [1–9]. The input of these algorithms is a reaction network and the output is the confirmation or rejection of the possibility of exhibiting multistationary behavior.

In the case where multistationarity can exist it then becomes important to determine the parameters where the network has this behavior. There has been less progress in this direction in the literature to date: the present paper offers a promising new development for this problem.

Prior work

Reviewing the state of the art in the literature, we see one vein of work focused on specific reaction networks, with success following heuristic or manual calculations to find a suitable parameter which may not work in generality [10, 11]. Then, in another vein of work the system of equations for finding equilibria are solved for many random points from the parameter space to approximate the region where the network is multistationary [12–15].

Recently in [16] a new approach to get a description of the multistationarity region is proposed. In this method one does not need to solve the system of equations to count the number of equilibriums. Instead one computes an integral to get the expected number of equilibriums when the parameters are following a random distribution. This method partitions the parameter region into subsets that are a Cartesian product of intervals, called hyperrectangles. By choosing the uniform distribution and computing the average number of equilibriums on these hyperrectangles, one can approximate the multistationarity region as a union of sub-hyperrectangles. While efficient and widely applicable, this list of hyperrectangles does not allow the reader much information or intuition about the geometry of this region, such as connectedness or convexity.

Contribution

In this work, we propose using polynomial superlevel sets to approximate the union of the hyperrectangles from [16] as a set that can be described by one polynomial. Polynomial superlevel sets are already employed to approximate semi-algebraic sets and have been used in control and robust filtering contexts, see [17, 18]. The polynomial superlevel set representation we propose is a more compact representation of the region compared to a list of many hyperrectangles each described as a Cartesian product of intervals. Further, to check if a point belongs to the region given in this representation one can easily just evaluate the polynomial in this point. Further benefits of the polynomial superlevel set description of the region will be explored later.

Organization of the paper

The organization of this paper is as follows. The mathematical framework of reaction networks and the definition of the multistationarity region is given first followed by a section containing the notations regarding parametric functions and definitions of the sampling and the rectangular representations of the multistationarity region from [19, Section 2.4].

We then define polynomial superlevel sets formally and describe how one can algorithmically find a polynomial superlevel set representation of a set using the sampling and the rectangular representations. We demonstrate how to use it to find the polynomial superlevel set representation of the multistationarity region of a reaction network. In the final section we discuss methods that sometimes can speed up computation of the polynomial superlevel set representation by the help of bisecting algorithms and where possible, algorithms for computing the expected number of solutions independently of solving the system itself.

Notations

The cardinality of a set A is denoted by $\#(A)$. Let $x \in \mathbb{Z}$ and $n \in \mathbb{Z} \setminus \{0\}$. In this paper we define x modulo n to be n instead of 0 whenever x is a multiple of n . For a function $f : A_1 \rightarrow A_2$ and a point $u \in A_2$, the level set of f is denoted by $L_u(f)$ and is defined as $\{x \in A_1 \mid f(x) = u\}$. For two points $a = (a_1, \dots, a_n)$ and $b = (b_1, \dots, b_n)$ in \mathbb{R}^n , the notation $[a, b]$ is used to show the hyperrectangle $\prod_{i=1}^n [a_i, b_i]$. For a subset S of a hyperrectangle $B \subseteq \mathbb{R}^n$, let $\overline{\text{Vol}}(S)$ denote the normalized volume of S with respect to B , i.e.

$$\overline{\text{Vol}}(S) = \frac{\text{Vol}(S)}{\text{Vol}(B)}.$$

When a random vector $X = (X_1, \dots, X_n)$ is distributed by a uniform distribution on a set $S \subseteq \mathbb{R}^n$, we write $X \sim U(S)$. If X is distributed by a normal distribution with mean $\mu \in \mathbb{R}^n$ and variance $\sigma^2 \in \mathbb{R}_{>0}$, then we write $X \sim N(\mu, \sigma^2)$ and we mean that X_1, \dots, X_n are identically and independently distributed by $N(\mu_i, \sigma^2)$. The expectation of $g(X)$ when X is distributed by a probability distribution q is denoted by $\mathbb{E}(g(X) \mid X \sim q)$.

Computer information

All computations for the examples of this paper were done on a computer with the following information. Processor: Intel(R) Core(TM) i7-10850H CPU @2.70GHz 2.71 GHz. Installed memory (RAM): 64.0 GB (63.6 GB usable). System type: 64-bit Operating System, x64-based processor.

The software and programming languages used for the computations reported in this paper had the following version numbers: Maple 2021, Matlab R2021a, YALMIP, SeDuMi 1.3, Julia 1.6.2, MCKR 1.0.

Multistationarity region of chemical reaction networks

In this section, we introduce the concepts of reaction network theory that are needed throughout the rest of the paper, with the help of a simple gene regulatory network example.

One can think of a gene as a unit encoding information for the synthesis of a product such as a protein. First, a group of DNA binding proteins called transcription factors bind a region of the gene called promoter. Now an enzyme called RNA polymerase starts reading the gene and produces an RNA until it arrives in the terminator region of the gene. The process until here is called the transcription step. After transcription is completed, the resulting RNA leaves the nucleus (in eukaryotes) and reaches ribosomes. In ribosomes, the second step, called translation, gets started. Ribosomes assemble a

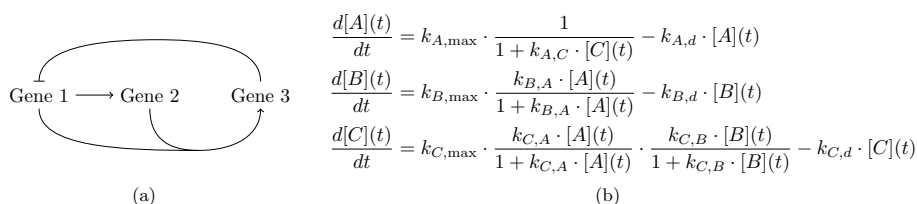


Fig. 1 A regulatory network of 3 genes [20, Figure 2]. **a** This graph shows the relations between expressions of the genes. We denote by \dashv an inhibitory relation and by \rightarrow a positive relation. **b** The system of ordinary differential equations for the network

protein from amino acids using the manual guide written in the RNA. A gene encoding of a protein recipe is said to be expressed when it gets transcribed to an RNA, and the RNA translated to the protein. A gene is not always expressed in a constant rate. There might be proteins that bind the transcription factors or the promoter region and, as a result, inhibit the RNA polymerase starting the transcription process. On the other hand, there might be other proteins in which their binding to the transcription factors or the promoter region enhances the transcription.

Consider a simple example from [20, Figure 2], depicted here in Fig. 1a. There are three genes with proteins A , B , and C as their final products. Denote their concentrations at time t by $[A](t)$, $[B](t)$ and $[C](t)$ respectively. The concentration of these proteins will not remain constant all the time, and we have an Ordinary Differential Equation (ODE) system describing the variation of the concentrations as time passes, see Fig. 1b. Each protein is degraded with a first-order kinetics with the reaction rate constants $k_{A,d}$, $k_{B,d}$ and $k_{C,d}$ correspondingly. Protein A activates the expression of the second gene with Michaelis-Menten kinetics with the maximum rate $k_{B,\max}$ and the Michaelis constant $k_{B,A}^{-1}$. The third gene gets activated by both proteins A and B together with the product of two Michaelis-Menten kinetics, with maximum rate $k_{C,\max}$ and Michaelis constants $k_{C,A}^{-1}$ and $k_{C,B}^{-1}$. The first gene gets expressed by the rate $k_{A,\max}$ in the absence of protein C , and protein C has an inhibitory effect on the expression of the first gene, captured by the denominator $(1 + k_{A,C}[C](t))$ in the rate expression.

A solution to the system $\frac{d[X_i](t)}{dt} = 0$ (where the X_i s are A , B and C) is called an equilibrium of the ODE system. Since the concentration of the proteins can only be non-negative real numbers, the complex or negative real solutions are not relevant. Sometimes we may only consider the positive solutions, for example, if a total consumption of a protein is not possible or of no interest. Therefore by steady states we mean positive solutions to the system of equations $\frac{d[X_i](t)}{dt} = 0$. The equations in this system are called the steady state equations.

Now we are ready to define a reaction network formally. A reaction network, or a network for short, is an ordered pair, $\mathcal{N} = (\mathcal{S}, \mathcal{R})$ where \mathcal{S} and \mathcal{R} are two finite sets called the set of species and the set of reactions. In our example, $\mathcal{S} = \{A, B, C\}$ and \mathcal{R} contains six reactions: three gene expressions and three protein degradations. To each network, an ODE is attached with concentration of the species as its variables and the constants of the reaction rate expressions as its parameters. In our example, we have 3 such variables and 10 parameters. To fix the notation assume $S = \{X_1, \dots, X_n\}$ and that there are r constants involved in the reaction rate expressions. Then we use x_i instead of $[X_i](t)$

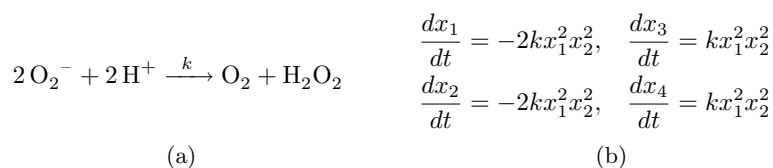


Fig. 2 A simple example of a closed network consisting of one reaction. **a** Two molecules of superoxide and two hydrogen atoms react to each other and produce one molecule of dioxygen and a molecule of hydrogen peroxide. The reaction rate here follows the mass-action kinetics with the reaction rate constant k . **b** The system of ODE equations of the network. The concentrations of O_2^- , H^+ , O_2 and H_2O_2 are denoted by x_1 , x_2 , x_3 and x_4 respectively

and k_i for the i -th parameter. Denote by $f_{i,k}(x)$ the i -th steady state equation where $x = (x_1, \dots, x_n)$ and $k = (k_1, \dots, k_r)$.

A network with an inflow (injection) or an outflow (extraction or degradation) for at least one of its species is called an *open network*. The network in Fig. 1a is an open network because of the presence of the degradation reactions. A network can also be fully or partially conserved.

Consider the simple single reaction network depicted in Fig. 2a. The system of its ODE equations is given in Fig. 2b. Because $\dot{x}_1 + 2\dot{x}_3 = 0$, the linear combination $x_1 + 2x_3$ should be constant with respect to the time. Therefore there exists a positive constant T_1 such that the relation $x_1 + 2x_3 = T_1$ holds. Similarly there exist two other positive constants T_2 and T_3 such that $x_1 + 2x_4 = T_2$ and $x_2 + 2x_3 = T_3$. The values of T_1 , T_2 and T_3 can be determined by the initial conditions of the ODE system. These linear invariants imply that three of the steady state equations are linearly redundant and can be replaced by these three linear invariants which are called *conservation laws* in CRN theory. The linear subspace determined by the conservation laws is called the *stoichiometric compatibility class*. For a more detailed definition of conservation laws see Definition 1 in [19, Chapter 2]. One should note that the trajectories of the ODE system are confined to stoichiometric compatibility classes. In this case, one only cares about the steady states in one stoichiometric compatibility class.

Now we are ready to define the main concept of interest, multistationarity.

Definition 2.1 Consider a network with n species and replace redundant steady state equations by conservation laws if there exist any. Let k stands for the vector of constants of both the reaction rates and conservation laws and be of the size r . A network is called *multistationary* over $B \subseteq \mathbb{R}^r$ if there exists a $k \in B$ such that $f_k(x) = 0$ has more than one solution in $\mathbb{R}_{>0}^n$.

Remark 2.2

- i) One may also consider non-linear invariants such as first integrals as defined in [21, Definition 11].
- ii) Note that we are not concerned with the choice of the kinetics such as mass-action, Michaelis-Menten, Hill function, power-law kinetics and S-systems, or the form of

the steady state equations such as polynomial or rational functions. Therefore the results of this paper will remain valid and practical for a general reaction network.

From here on the word parameters also includes the constants of the conservation laws in addition to the reaction rate constants. To answer the question of whether a network is multistationary or not one can use one of many algorithms available in the literature, see [1–5, 7, 9] for a few examples. However, to partition the parameter space into two subsets, one consisting of the choices of parameters for which $f_k(x)$ has more than one solution and the other comprising those parameter choices for which $f_k(x)$ has at most one solution, is a more laborious task which we tackle in this paper.

Definition 2.3 Consider a reaction network with the setting and notation of Definition 2.1. The set

$$\{k \in B \mid \#(f_k^{-1}(0) \cap \mathbb{R}_{>0}^n) \geq 2\}$$

is called the multistationarity region of the network.

The region B in Definition 2.3 represents the regions of scientific interest. It will usually be a hyperrectangle made by the inequality restrictions of the form $k_{i,\min} < k_i < k_{i,\max}$ for the parameters. This is because, for example the rate of expression of a gene can not be any arbitrary positive number but must be limited; or the constant of conservation laws may be limited from above due to the limitation of the materials in the lab.

Prior state-of-the-art for parametric systems of equations

Let $f_k : \mathbb{R}^n \rightarrow \mathbb{R}^m$ be a parametric function with $B \subseteq \mathbb{R}^r$ as its parameter region and u a point in \mathbb{R}^m . For each choice of the parameters $k^* \in B$, the system $f_{k^*}(x) = u$ is a non-parametric system of equations. One can solve this system and look at the cardinality of the solution set. For different choices of k^* , this number can be different. Therefore we define a new function $\Phi_f^u : B \rightarrow \mathbb{Z}_{\geq 0} \cup \{\infty\}$ sending $k \in B$ to $\#(L_u(f))$, i.e. the size of the level set of f_k (the set of points in \mathbb{R}^n which f_k maps to u). Now one can partition B into the union of level sets of the map Φ_f^u . For a general form of $f_k(x)$, finding $L_i(\Phi_f^u)$ is a hard question.

CAD with respect to discriminant variety

In the case where $f_k(x) \in (\mathbb{R}(k)[x])^m$ and A and B are semi-algebraic sets there are a variety of tools which can be employed, see for example [22]. In the literature, the approach used most commonly (e.g. [10, 23, 24]) is a Cylindrical Algebraic Decomposition (CAD) computed with respect to the discriminant variety. For a full description of this technique we refer the reader to [25, 26] or the short sketch of the main idea in [24, section 3]. Briefly: the discriminant variety of the system $f_k(x)$ with the domain and codomain restrictions on the semi-algebraic sets A and B is the solution set to a new set of (non-parametric) polynomial equations with k as its indeterminants. This new set of polynomials can be computed algorithmically for example using Gröbner bases and elimination theory. Then CAD with respect to the discriminant variety decomposes B into a finite number of connected semi-algebraic sets

called cells. Each cell has intersection with only one $L_i(\Phi_f^u)$ and therefore $L_i(\Phi_f^u)$ can be expressed as union of a finite number of cells with an exact description of their boundaries.

As we see later in this section, in many cases one is only interested in open cells (i.e. only those cells which have full dimension [27]). A Maple package, `RootFinding[Parametric]` has implemented an algorithm to compute the open CAD with respect to the discriminant variety of a system of parametric polynomial equations and inequalities [28]. From here on in this paper by *CAD* we mean such an open CAD with respect to the discriminant variety.

Both the computation of the discriminant variety and the subsequent decomposition involve the use of algorithms with doubly exponential complexity which can cause problems. The number of cells in the decomposition will grow doubly exponentially in the number of parameters of $f_k(x)$ [29]; and even computation of the discriminant variety itself before any decomposition can be infeasible for moderate examples, see e.g. [24]. This makes CAD impractical for studying parametric systems of polynomial equations with more than a few variables and parameters.

Approximation by sampling

Another approach adopted by scientists is to solve the system $f_k(x) = u$ for many different choices of $k \in B$ [12–15].

Mathematically speaking, this means that B is replaced by a finite set. Then each $L_i(\Phi_f^u)$ is expressed as a subset of this finite set. This approach hereafter is referred as the *sampling representation* approach. In contrast with the CAD approach which provides an exact description of $L_i(\Phi_f^u)$, the sampling representation approach provides an approximation. Note that there are different ways to choose the sample parameter points for the sampling representation. One way is to arrange all points equally distanced like a grid, and another way is to randomly sample from a distribution such as the uniform distribution on B , which is the one used in this paper. For an example of a case where a sampling representation with grid-like parameter sampling is used see [10, Figures 7–12].

Since we are motivated from the application, we should note that in a lab, it is usually not possible to design the experiment so that the parameter values are exactly the numbers that we decide. Therefore when the experiment is designed to have $k = k^*$, what happens is that k is a point in a neighborhood of k^* and not necessarily k^* itself. This can happen for example because of errors coming from the measurement tools or the noise from the environment. In such cases picking a point close to the boundaries of $L_i(\Phi_f^u)$ could lead to a different result than what the experimentalist expects, if errors or noise push it over the boundary.

Rectangular representation

A different discretization can be done using a rectangular division of B . For example if B is a hyperrectangle $[a, b]$ then a grid on B is achieved by dividing B along each axis to equal parts. Then for each sub-hyperrectangle of B in this rectangular division we assign the average of the number of solutions of $f_k(x) = u$ for several choices of k coming from

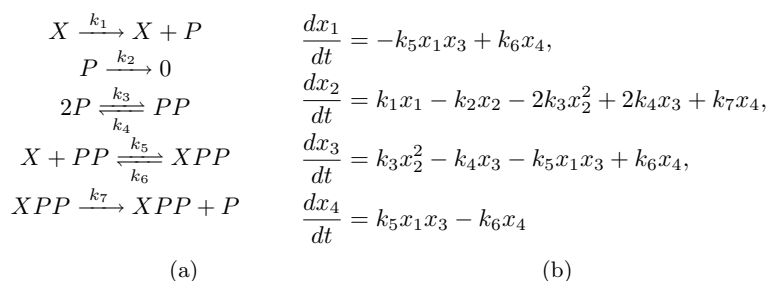


Fig. 3 A bistable autoregulatory motif presented in [30, Figure 3B]. **a** X is a gene, P is a protein that can form a dimer PP and then binding to X . The gene X will get expressed and produce P in both forms X and XPP . Finally there is a degradation of P . **b** The ODE system of the gene regulatory network in **a**. The variables x_1, x_2, x_3 and x_4 are standing for the concentration of the species X, P, PP and XPP respectively

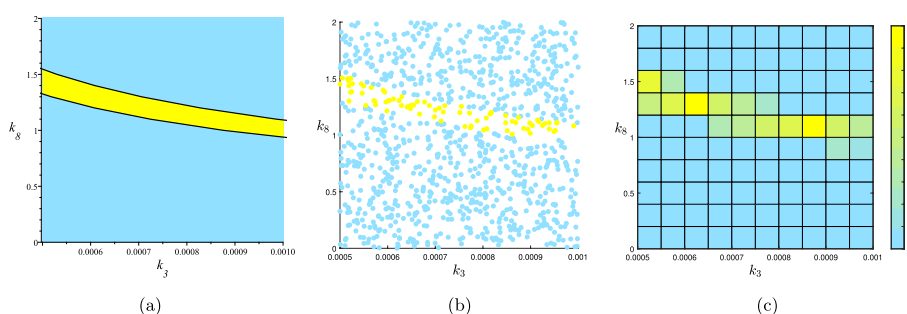


Fig. 4 Three representations of the multistationarity region of the network in Fig. 3a after fixing all parameter values other than k_3 and k_8 to the values in (1). **a** CAD gives the exact boundary of $L_1(\Phi_f^0)$ and $L_3(\Phi_f^0)$. The first one is colored by sky blue and the later one with yellow. **b** A sampling representation of the parameter region $B = [(0.0005, 0), (0.001, 2)]$ by 1000 random points sampled from uniform distribution on B . 78 of these points belong to $L_3(\Phi_f^0)$ and are colored yellow. The other 922 points belong to $L_1(\Phi_f^0)$ and are colored by sky blue. **c** A rectangular representation of B . Each subrectangle is colored with respect to the average number of solutions for 10 random points sampled from the uniform distribution on the subrectangle. The color bar of the figure is in the right side

the sub-hyperrectangle. This approach hereafter is referred as the *rectangular representation* approach. See Fig. 4 to compare the three approaches visually.

Example

Consider the gene regulatory network in [30, Figure 3B], depicted here in Fig. 3a with the ODE system in Fig. 3b. This network has one conservation law, $x_1 + x_4 = k_8$. Therefore we consider the system of equations obtained by the first three steady state equations in the ODE system and the conservation law to study the multistationarity of this network. For illustration purpose we fix values of all parameters other than two so we can plot the multistationarity region in 2 dimensions. In [30, Figure 4] the reaction rate constants other than k_3 were fixed to the values listed below.

$$k_1 = 2.81, k_2 = 1, k_4 = 0.98, k_5 = 2.76, k_6 = 1.55, k_7 = 46.9. \tag{1}$$

We fix the values of all parameters other than k_3 and k_8 to these values also.

Let B be the rectangle made by the constraints $0.0005 < k_3 < 0.001$ and $0 < k_8 < 2$. Using the `RootFinding[Parametric]` package of Maple we get the exact description of the multistationarity region of the network, in 0.12 s, depicted in Fig. 4a.

A sampling representation of the multistationarity region is found by solving the system of the equations for 1000 points (k_3, k_8) sampled from the uniform distribution on $[(0.0005, 0), (0.001, 2)]$. We used the `vpasolve` command from Matlab to solve the system numerically. The Matlab code to generate this sampling representation took 154 s to run, with the output visualised in Fig. 4b.

A rectangular representation is given by dividing $[(0.0005, 0), (0.001, 2)]$ to 100 equal sub-rectangles and then solving the system for 10 points (k_3, k_8) sampled from the uniform distribution on each sub-rectangles. The sub-rectangles are colored with respect to the average number of solutions. This computation also was done by Matlab and took 166 s, with the output visualised in Fig. 4c.

Polynomial superlevel set representation

Superlevel sets

Definition 4.1 Consider $f : \mathbb{R}^n \rightarrow \mathbb{R}$, an arbitrary function. For a given $u \in \mathbb{R}$ a *superlevel set* of f is the set of the form

$$U_u(f) = \{x \in \mathbb{R}^n \mid f(x) \geq u\}.$$

When $u = 1$ we drop the index and write only $U(f)$. Naturally, a *polynomial superlevel set* is a superlevel set of a polynomial.

Polynomial sublevel sets are defined similarly as in Definition 4.1 with the only difference the direction of the inequality. However, in this paper, we only focus on superlevel sets. For $d \in \mathbb{Z}_{\geq 0}$ let P_d denote the set of polynomials of total degree at most d . A sum of squares (SOS) polynomial of degree $2d$ is a polynomial $p \in P_{2d}$ such that there exist $p_1, \dots, p_m \in P_d$ so that $p = \sum_{i=1}^m p_i^2$. We denote the set of SOS polynomials of degree at most $2d$ by Σ_{2d} .

Theorem 4.2 ([17, Theorem 2]) *Let $B \subseteq \mathbb{R}^n$ be a compact set and K a closed subset of B . For $d \in \mathbb{N}$ define*

$$S_d = \{p \in P_d \mid p \geq 0 \text{ on } B, p \geq 1 \text{ on } K\}.$$

Then there exists a polynomial $p_d \in S_d$ such that

$$\int_B p_d(x) dx = \inf \left\{ \int_B p(x) dx \mid p \in S_d \right\}.$$

Furthermore $\lim_{d \rightarrow \infty} \text{Vol}(U(p_d) - K) = 0$.

Given a pair (B, K) where $B \subseteq \mathbb{R}^n$ is a compact set and $K \subseteq B$ a closed set, and given $d \in \mathbb{N}$; we call the polynomial superlevel set $U(p)$ (with p being the polynomial $p_d \in S_d$

found in Theorem 4.2) the PSS representation of $K \subseteq B$ of degree d . When K is a semi-algebraic set, one can find p_d numerically using a minimization problem subject to some positivity constraints [18, Equation 13].

Let $B = [a_B, b_B]$ and $K_i = [a_{K_i}, b_{K_i}], i = 1, \dots, m$ be some hyperrectangles in \mathbb{R}^n such that $K := \cup_{i=1}^m K_i \subseteq B$. By solving a similar optimization problem it is possible to find the PSS representation of $K \subseteq B$. Let $d \in \mathbb{N}$. The goal is to find the coefficients of a polynomial of degree d such that $\int_B p(x) dx$ becomes minimum subject to some conditions. Before presenting the constraints, let us look at the target function. A polynomial $p(x)$ of degree d can be written as $\sum_{\alpha \in \mathbb{N}_d^n} c_\alpha x^\alpha$. Here \mathbb{N}_d^n is the set of $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{Z}_{\geq 0}^n$ such that $\sum_{i=1}^n \alpha_i \leq d$. Now the integral can be simplified as below.

$$\begin{aligned} \int_B p(x) dx &= \int_B \left(\sum_{\alpha \in \mathbb{N}_d^n} c_\alpha x^\alpha \right) dx \\ &= \sum_{\alpha \in \mathbb{N}_d^n} c_\alpha \int_B x^\alpha dx \\ &= \sum_{\alpha \in \mathbb{N}_d^n} \left(\int_B x^\alpha dx \right) c_\alpha. \end{aligned}$$

Since $\int_B x^\alpha dx$ are constant real numbers independent of the coefficients of the polynomial, the target function is a linear function on the coefficients of $p(x)$ which are the variables of the optimization problem.

Now let us look at the constraints. First of all $p(x)$ has to be nonnegative on B . This can be enforced by letting

$$p(x) - \sum_{j=1}^n s_{B,j}(x)(x_j - a_{B,j})(b_{B,j} - x_j) \in \Sigma_{2r}, \quad s_{B,j} \in \Sigma_{2r-2}, j = 1, \dots, n, \tag{2}$$

where $r = \lfloor \frac{d}{2} \rfloor$ the largest integer less than or equal to $\frac{d}{2}$. Secondly we need $p(x) \geq 1$ on K or in other words $p(x) - 1 \geq 0$ on K . This holds if and only if $p(x) - 1 \geq 0$ on each K_i . Therefore for every $i = 1, \dots, m$ one more constraint of the shape (2) has to be added:

$$p(x) - 1 - \sum_{j=1}^n s_{K_i,j}(x)(x_j - a_{K_i,j})(b_{K_i,j} - x_j) \in \Sigma_{2r}, \quad s_{K_i,j} \in \Sigma_{2r-2}, j = 1, \dots, n.$$

Recall Definition 2.3: the multistationarity region of a network is in fact a superlevel set, $U_2(\Phi_f^0)$. The goal is to find a PSS representation of the set $U_2(\Phi_f^0)$. One way to accomplish this goal is to find a rectangular representation of the multistationarity region and then solve the above mentioned SOS optimization problem. The next example illustrates this idea. To tackle it we use a Matlab toolbox called YALMIP [31, 32] which can receive an SOS optimization problem, process it and use other solvers to solve it. For the solver to be used by YALMIP, we chose SeDuMi [33].

Examples

We continue with the example from Fig. 3. Consider the rectangular representation of the multistationarity region of that example given in Fig. 4c. To find the PSS representation of

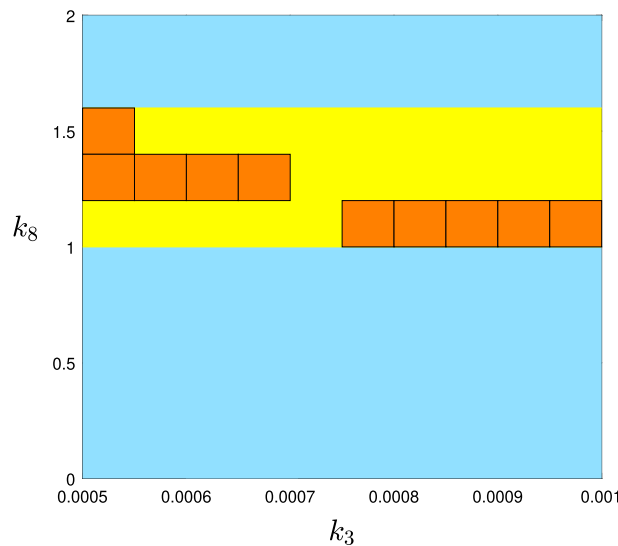


Fig. 5 The PSS approximation of the multistationarity region for the network in Fig. 3a of degree 2 obtained by the information of Fig. 4c. The union of orange colored subrectangles is considered as the initial approximation of the multistationarity region obtained by the rectangular representation and chosen as the set K . The yellow colored area is the difference of $U(p) - K$. Remember that the PSS representation of the multistationarity region is the yellow region which contains the orange rectangles as well

this set, we let $B = [(0.0005, 0), (0.0010, 2)]$ and K be the union of rectangles K_i s such that their associated number is greater than or equal to 2. From the 100 sub-rectangles of B , 9 of them satisfy this condition. These sub-rectangles are colored orange in Fig. 5. We use the YALMIP and SeDuMi packages of Matlab to solve the SOS optimization discussed before this example. To report the computation time we add the two times reported in the output of YALMIP: the “yalmiptime” and “solvertime”. It takes about 2 s to get the coefficients of the polynomial p of the PSS representation of degree 2. Figure 5 shows the plot of $U(p)$.

Unfortunately the same code does not produce a better approximation when we increase d , the degree of p , from 2 to 4, 8 or 16. The output from Matlab gives similar figures in these cases as Fig. 5.

Consider another gene regulatory example from [19, Chapter 2]. To avoid lengthening the text, we only reproduce the system of equations needed to study the multistationarity of the network:

$$\begin{aligned}
 k_1x_7x_5 - k_5x_1 &= 0 & k_2x_8x_6 - k_6x_2 &= 0 \\
 k_3x_1 - k_7x_3 &= 0 & k_4x_2 - k_8x_4 &= 0 \\
 k_9x_7x_4 - k_{11}x_9 &= 0 & k_{10}x_8x_3 - k_{12}x_{10} &= 0 \\
 k_{13}x_9x_4 - k_{15}x_{11} &= 0 & k_{14}x_{10}x_3 - k_{16}x_{12} &= 0 \\
 x_5 &= k_{17} & x_6 &= k_{18} \\
 x_7 + x_9 + x_{11} &= k_{19} & x_8 + x_{10} + x_{12} &= k_{20}.
 \end{aligned}
 \tag{3}$$

We fix all parameters other than k_7 and k_8 to the following values coming from Equation (2.10) of [19, Chapter 2]:

$$\begin{aligned}
 k_1 = k_2 = k_3 = k_4 &= 1, & k_5 &= 0.0082, & k_6 &= 0.0149, \\
 k_9 = k_{10} &= 0.01, & k_{11} = k_{12} &= 10000, & k_{13} &= 2, \\
 k_{14} &= 25, & k_{15} &= 1, & k_{16} &= 9, & k_{17} = k_{18} = k_{19} &= 1, \\
 k_{20} &= 4.
 \end{aligned}
 \tag{4}$$

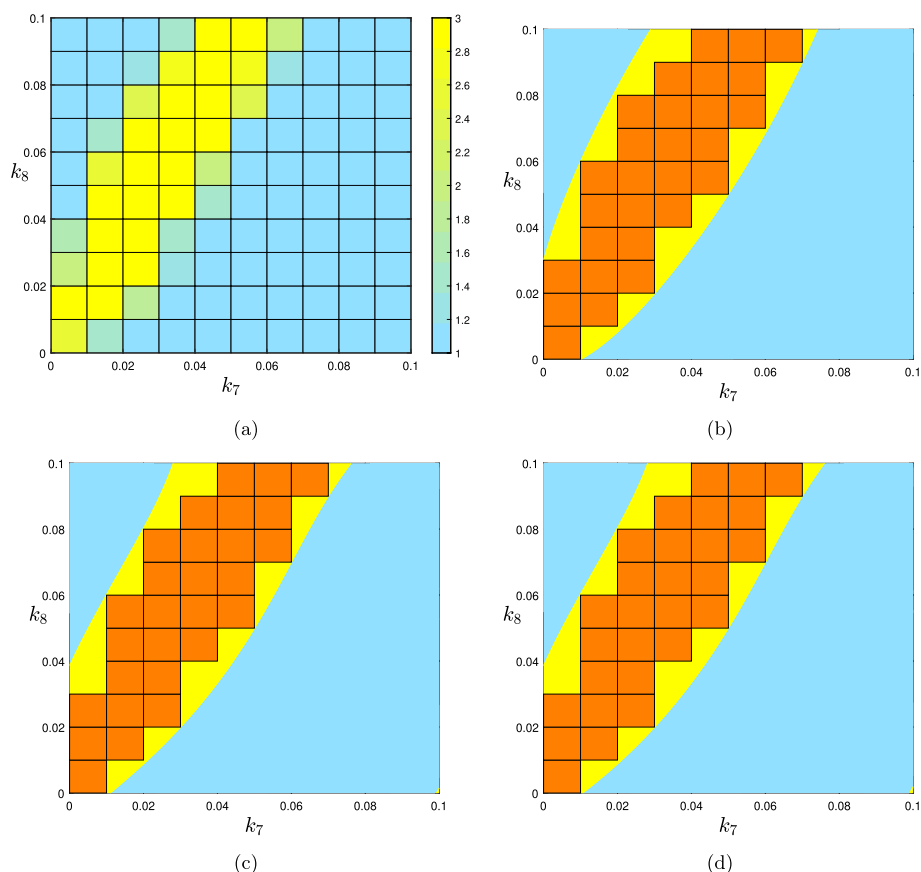


Fig. 6 PSS representations of different degrees for the multistationarity region of the LacI-TetR gene regulatory network using the information we got from the rectangular representation. The union of orange colored subrectangles is considered as the approximation of the multistationarity region obtained by the rectangular representation and chosen as the set K . The yellow colored area is the difference of $U(p) - K$. **a** The rectangular representation of multistationarity region of the network with the system of equations given in (3) and some parameters being fixed by the values in (4). **b–d** PSS representations of the multistationarity region of degrees 2, 4 and 8 respectively obtained by the information of Fig. 6a

We reproduced the rectangular representation of the multistationarity region of this network by Matlab, as shown in Fig. 6a. From the 100 sub-rectangles in total, for 28 of them the average number of steady states is greater than or equal to 2. Using YALMIP and SeDuMi it took between 1 and 2 s to get the polynomials of the PSS representations of degrees 2, 4 and 8 represented in Fig. 6b–d respectively. For this example, the PSS approximation of degree 4 looks different than that of degree 2, but for degree 8 the plot looks similar to degree 4.

Advantages of PSS representation over rectangular representation

It is natural to ask why one should find a PSS representation of the multistationarity region using the rectangular representation given one already has the rectangular representation? Let $B \subseteq \mathbb{R}^r$ be the parameter region of the form of a hyperrectangle, and $K \subseteq B$ be the multistationarity region. In the rectangular representation we have $K \simeq \cup_{i=1}^m K_i$ where $K_i = [a_{K_i}, b_{K_i}]$ are hyperrectangles. In the PSS representation we have $K \simeq U(p)$ where p is a polynomial of degree d .

- 1- When $r \geq 4$, plotting K is impossible. In order to save or show the rectangular representation one needs to use a matrix of size $(m) \times (2r)$, where each row stands for one K_i and the first r columns have the coordinates of the point a_{K_i} and the second r columns correspond to the coordinates of the point b_{K_i} . However for the PSS representation one needs to use only a vector of size $\binom{r+d}{r} = \sum_{i=0}^d \binom{r-1+i}{r-1}$, where $\binom{r-1+i}{r-1}$ entries are coefficients of the terms of degree i . The terms are ordered from smaller total degree to larger and for the terms of the same total degree we use the lexicographic order.
- 2- To test if a point $k^* \in B$ belongs to K using the rectangular representation one should check m conditions of the form $k^* \in K_i$ which means verifying an inequality on each coordinate of the point, i.e. $a_{K_{i,j}} \leq k_j^* \leq b_{K_{i,j}}$. If one of the conditions $k^* \in K_i$ is positive, then there is no need to check the rest, otherwise all should fail to conclude that $k^* \notin K$. However, using the PSS representation one needs to check only one condition of an evaluation form, i.e. $p(k^*) \geq 1$.
- 3- Recall from the last paragraph of the “[Approximation by sampling](#)” Section explaining that parameters near the boundary of the multistationarity region are not suitable choices for an experimentalist. To check the distance of a point $k^* \in B \setminus K$ to the boundaries of K using the rectangular representation one should find distance of k^* from boundaries of each K_i and then taking the minimum. However, using the PSS representation, in both cases of $k^* \in B \setminus K$ or $k^* \in K$, one just needs to find the distance of k^* from the algebraic set defined by $p(k) - 1 = 0$, for example by Lagrange multipliers, as in the next section.

To conclude, if $\binom{r+d}{r}$ is considerably smaller than $2mr$, then storing the PSS representation instead of the initial rectangular representation will save memory without losing information about the multistationarity region.

Approximating the distance of parameter point from the boundary

To illustrate how to approximate distance of a parameter point from the boundaries of the multistationarity region using a PSS representation we continue with the example from Fig. 3.

Let p be the polynomial of degree 2 in two variables k_7 and k_8 corresponding to $U(p)$ in Fig. 6b. We will approximate distance of the point $k^* = (0.08, 0.02)$ from the boundary of the multistationarity region by the distance of k^* from the algebraic set defined by $p(k_7, k_8) - 1 = 0$. This question is equivalent to minimizing the Euclidean distance function of a point (k_7, k_8) from the point k^* subject to the constraint $(k_7, k_8) \in L_1(p)$. The target function is $\sqrt{(k_7 - 0.08)^2 + (k_8 - 0.02)^2}$ which gets minimized if and only if $(k_7 - 0.08)^2 + (k_8 - 0.02)^2$ gets minimized. An elementary way to solve this minimization problem is to use the method of Lagrangian multipliers [34, Chapter 7, Theorem 1.13]. Define

$$F(k_7, k_8, \lambda) = (k_7 - 0.08)^2 + (k_8 - 0.02)^2 + \lambda(p(k_7, k_8) - 1).$$

Now we must find the critical points of $F(k_7, k_8, \lambda)$. So we should solve the system of equations obtained by $\frac{\partial F}{\partial k_7} = \frac{\partial F}{\partial k_8} = \frac{\partial F}{\partial \lambda} = 0$. It takes 0.167 s to solve this system of equations by the `solve` command in Maple. It has 4 solutions, from which 2 belong to the rectangle $B = [(0, 0), (0.1, 0.1)]$ and the minimum of the target function is obtained at the point (0.04499222669, 0.04161251428). The distance of this point from k^* is 0.04114176669.

Constructing a PSS representation from a sampling representation

It is not necessary to have a rectangular representation to get the PSS representation. Let $B = [a_B, b_B]$ be a hyperrectangle and $K = \{a^{(1)}, \dots, a^{(m)}\}$ a finite set. Let $d \in \mathbb{N}$, and the goal be to find the coefficients of a polynomial of degree d such that $\int_B p(x) dx$ becomes minimum subject to some conditions. We already saw that the target function is linear. The constraint $p \geq 1$ on K can be enforced by $p(a^{(i)}) \geq 1$ for every i , which are linear constraints. The positivity of p on B can be enforced by Eq. (2) or by adding a large enough number of random points from B and putting the constraint $p(a) > 0$. The later idea makes the problem solvable by any common linear programming tool. However, here we still use Eq. (2).

Let us illustrate this with our ongoing example. Consider the sampling representation of the multistationarity region of the network of Example 3.4 given at Fig. 4b. To find the PSS representation of this set, we let $B = [(0.0005, 0), (0.001, 2)]$ and K to be the set of points for which the system $f_k(x) = 0$ had more than one positive solution. There are 1000 points from which 78 of them are parameter choices where the network has three steady states. Using the YALMIP package of Matlab, it takes less than a second to get the coefficients of each of the polynomials p of the PSS representation of degrees 2, 6 and 10. Figure 7a–c show the plots of $U(p)$ for degrees 2, 6 and 10 respectively. The plot for degree 6 actually looks worse than the plot for degree 2 (further away from the actual result in Fig. 4a), although the one for degree 10 looks a little better. In all these cases YALMIP finished the computations with a message ‘Numerical problems (SeDuMi)’ indicating that the solver found the problem to be numerically ill-posed. Rescaling the parameter region of interest, B , to $[(0, 0), (1, 1)]$ and then transforming the PSS polynomial back to the original B allows a better PSS approximations via YALMIP. For degrees 2 and 6 the numerical problem message is avoided but for degree 10 it remains. The results are shown in Fig. 7d–f.

To demonstrate that the number of free parameters need not be 2 to be able to compute the PSS representation, we repeated the above process with all 8 parameters of the system being free in the following hyperrectangle:

$$B = [(1, 0, 0.0005, 0, 1, 1, 40, 0), (4, 2, 0.001, 2, 4, 3, 50, 2)].$$

Solving the system at 1000 random points uniformly chosen from B takes the same amount of time as solving the system at 1000 random points with only 2 of their

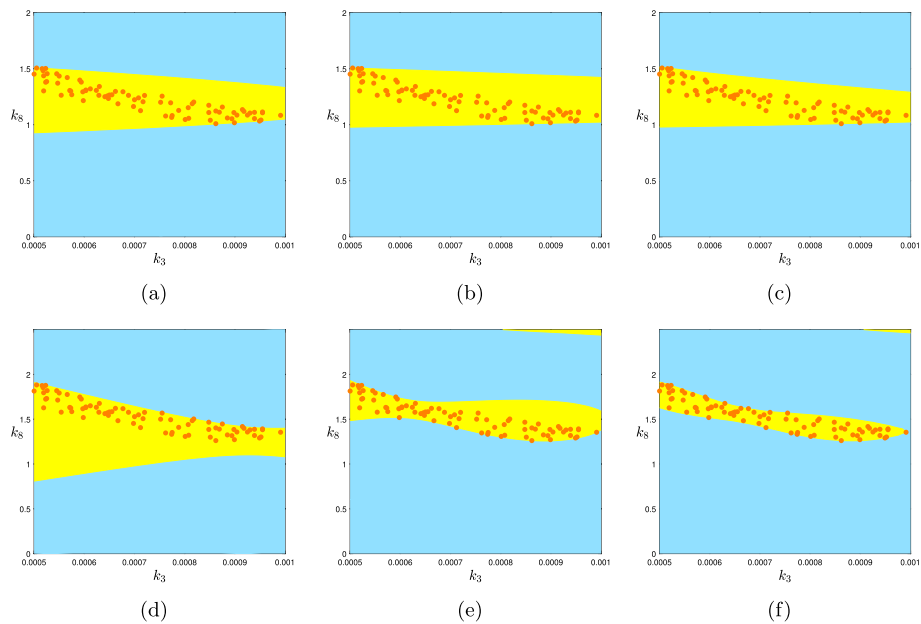


Fig. 7 PSS representation of different degrees of the multistationarity region of the network of Example 3.4 inside the hyperrectangle $B = [(0.0005, 0), (0.001, 2)]$ using the information we got from the sampling representation of the multistationarity region. The orange colored points are the points with three steady states and their union is considered as approximation of K . The yellow colored area is the difference of $U(p) - K$. One expects to see that this difference is getting smaller as the degree increases. However, the Matlab code that we wrote using YALMIP and SeDuMi does not behave as expected. **a–c** gives the PSS representation of the original problem of degrees 2, 6 and 10 respectively. **d–f** gives the PSS representation of those degrees for the problem after after rescaling the parameters for better numerical behavior via YALMIP and SeDuMi

coordinates varying took time in the previous case. It took about 1.5 s to compute the 45 coefficients of the polynomial of the PSS representation of degree 2. The polynomial found is the following.

$$\begin{aligned}
 p = & 0.0232593410k_1^2 - 0.2965863774k_2^2 + 559.9760177000k_3^2 \\
 & - 0.0618678914k_4^2 + 0.0098179292k_5^2 - 0.0061375489k_6^2 \\
 & - 0.0036871825k_7^2 - 0.0716829116k_8^2 - 0.0110701085k_1k_2 \\
 & - 5.3373104650k_1k_3 + 0.0186292240k_1k_4 + 12.5539028600k_2k_3 \\
 & + 0.0051935911k_1k_5 - 0.0820767495k_2k_4 + 0.0250525965k_1k_6 \\
 & - 0.0649618436k_2k_5 - 8.2165668850k_3k_4 + 0.0092355627k_1k_7 \\
 & - 0.0310047257k_2k_6 - 4.2926137530k_3k_5 - 0.0527315151k_1k_8 \\
 & + 0.0017547696k_2k_7 - 2.7647028150k_3k_6 + 0.0288291263k_4k_5 \\
 & + 0.0785940967k_2k_8 - 0.4049844472k_3k_7 + 0.0536360086k_4k_6 \\
 & + 12.6014929300k_3k_8 - 0.0091147734k_4k_7 + 0.0198800552k_5k_6 \\
 & - 0.0567174819k_4k_8 + 0.0009448114k_5k_7 - 0.0190068615k_5k_8 \\
 & + 0.0102920949k_6k_7 + 0.0316830973k_6k_8 + 0.0022468043k_7k_8 \\
 & - 0.6043381476k_1 + 0.7152344469k_2 + 0.3287484744k_4 \\
 & + 36.3811699100k_3 - 0.0617318336k_5 - 0.5940226892k_6 \\
 & + 0.2987521874k_7 + 0.2558883329k_8 - 5.0441247030.
 \end{aligned}$$

Advantages of PSS representation over the sampling representation

Let us address why one should find a PSS representation of the multistationarity region using the sampling representation if one already has a sampling representation. Let $B \subseteq \mathbb{R}^r$ be the parameter region of the form of a hyperrectangle, $K \subseteq B$ be the multistationarity region. In the sampling representation we have $\{a^{(1)}, \dots, a^{(m)}\} \subseteq K$. In the PSS representation we have $K \simeq U(p)$ where p is a polynomial of degree d .

- 1- When $r \geq 4$, plotting K is impossible. In order to save or show the sampling representation one needs to use a matrix of the size $m \times r$, where each row stands for one point $a^{(i)}$ and the columns correspond to the coordinates of the points. However, for the PSS representation one needs to use a vector of the size $\binom{r+d}{r}$, as explained in the first item of the “Advantages of PSS representation over rectangular representation” section1.
- 2- To test if a point $k^* \in B$ belongs to K using the sampling representation is not a straightforward task. However, using the PSS representation one needs to verify only one condition of the evaluation form, $p(k^*) \geq 1$.
- 3- To compute the distance of a point $k^* \in B$ to the boundaries of K , using the sampling representation, if $k^* \notin K$, one should compute the distance of k^* from each point in the sampling representation of K and then take the minimum. Using the PSS representation, whether $k \notin K$ or not, one just needs to find the distance of k^* from the algebraic set defined by $p(k) - 1 = 0$.

In a typical example from CRN theory, r is usually much higher than 2, and therefore item 1 is really important. When $\binom{r+d}{r}$ is lower than rm , one can use less memory by saving the PSS representation instead of keeping all the points of the sampling representation in the memory. Further, as items 2 and 3 show, this will not cause a loss of information about the multistationarity region.

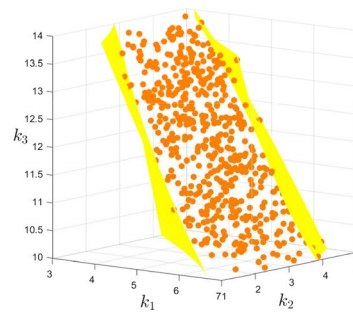
More involved example

We showed earlier that the PSS representation can be generated for examples with a higher number of parameters than two. There, we let all 8 parameters of the network in Fig. 3a to be free and found the PSS representation of degree two in 8 indeterminants. Now we bring another such example which also serves to emphasize Remark 2.2 item (ii): that to have a PSS representation of the multistationarity region, one does not need to have the right hand side functions of the ODE system to be of polynomial or even rational functions.

Consider a gene expression system with 4 species $X_i, i = 1, \dots, 4$ where these species can be m-RNA or protein molecules or other relevant factors, with the ODE system as in Fig. 8a which was introduced in [35, Figure 2]. As one can see the right hand side functions involve at least a square root, and as a result this system is not polynomial, or

$$\begin{aligned} \frac{dx_1}{dt} &= \alpha_1 \left(v_1 + \frac{v_2 x_4^n}{v_3^n + x_4^n} \right) \cdot x_3^{-0.5} - \beta_1 x_1^{h_{1,1}}, \\ \frac{dx_2}{dt} &= \alpha_2 x_1^{g_{2,1}} - \beta_2 x_2^{h_{2,2}}, \\ \frac{dx_3}{dt} &= \alpha_3 x_2^{g_{3,2}} - \beta_3 x_3^{h_{3,3}}, \\ \frac{dx_4}{dt} &= \alpha_4 x_3^{g_{4,3}} - \beta_4 x_4^{h_{4,4}}. \end{aligned}$$

(a)



(b)

Fig. 8 A gene expression network with an S-system kinetics that allows Hill terms and exhibits multistationary behavior, taken from [35, Figure 2]. **a** The ODE system of the network. The set of species is $\{X_1, X_2, X_3, X_4\}$. Each equation dx_i/dt consists of two terms responsible for production and degradation of the species X_i . The right hand side functions are not necessarily of a polynomial type, or even a rational function form. **b** The sampling and PSS representation of the multistationarity region of this network when all the parameters other than $\beta_i, i = 2, 3, 4$ are fixed to the values in (5) and the remaining three parameters renamed $k_i, i = 1, 2, 3$ vary in the cube $B = [(3, 1, 10), (7, 5, 14)]$. The sampling representation consists of the orange color points and the PSS representation is the area between the two yellow colored surfaces

even defined by rational functions. Let us fix the parameter values other than the three degradation rates $\beta_i, i = 2, 3, 4$ to the following values, chosen the same as in [35]:

$$\begin{aligned} \alpha_1 &= 1, \quad n = 4, \quad v_1 = 4, \quad v_2 = 8, \quad v_3 = 4, \\ h_{1,1} &= h_{2,2} = 0.5, \quad h_{3,3} = 1, \quad h_{4,4} = 0.75, \\ g_{2,1} &= g_{3,2} = g_{4,3} = 1, \quad \beta_1 = 0.5, \\ \alpha_2 &= 1, \quad \alpha_3 = 2, \quad \alpha_4 = 3. \end{aligned} \tag{5}$$

In [35] it is shown that for $(\beta_2, \beta_3, \beta_4) = (5, 3, 12)$, the network is bistable, with three steady states. Here we rename these three parameters by $k_i, i = 1, 2, 3$ respectively, let them vary in the 3-dimensional cube $B = [(3, 1, 10), (7, 5, 14)]$, and look for the multistationary region. I.e. we seek the relation between these three parameters so that the number of steady states of the network remains the same. Substituting the values into Fig. 8a and letting $dx_i/dt = 0, i = 1, \dots, 4$ gives the following system of equations.

$$\begin{aligned} 4 + \frac{8x_4^4}{256+x_4^4} \frac{1}{\sqrt{x_1}} - \frac{1}{2} \sqrt{x_1} &= 0, \\ x_1 - k_1 \sqrt{x_2} &= 0, \\ 2x_2 - k_2 x_3 &= 0, \\ 3x_3 - k_3 \sqrt[4]{x_4^3} &= 0. \end{aligned} \tag{6}$$

With a simple calculation one can see that the set of positive solutions to the system of Eq. (6) is in one to one correspondence with the the set of positive roots of the following univariate polynomial:

$$f(y) = \sqrt[4]{24k_1^2 k_2 k_3^3} y^{73} - 144y^{64} + 256 \sqrt[4]{24k_1^2 k_2 k_3^3} y^9 - 12288. \tag{7}$$

For any positive root of the polynomial in (7), a positive solution for (6) is

$$(x_1, x_2, x_3, x_4) = \left(k_1 \sqrt{\frac{k_2 k_3}{6}} y^6, \frac{k_2 k_3}{6} y^{12}, \frac{k_3}{3} y^{12}, y^{16} \right).$$

By Descartes' rule of signs it is clear that the polynomial in (7) has 1 or 3 positive roots counted by multiplicity for any choice of k_i s. To find the sampling representation of the multistationarity region of our system we solved the equation $f(y) = 0$ for 1000 random choices of the parameters using Matlab, which took about 8 min. At 532 of these choices the polynomial had 3 positive roots. The sampling representation is shown in Fig. 8b, with the points where the system has 3 steady states shown as orange spheres. We then used the information from the sampling representation to compute the PSS representation of the multistationarity region of degree two, which took less than 2 s. The resulting polynomial is below, with coefficients given to 6 decimal places.

$$\begin{aligned} p(k_1, k_2, k_3) = & -0.043653k_1^2 - 0.029919k_2^2 \\ & -0.013650k_3^2 - 0.082681k_1k_2 - 0.045005k_1k_3 \\ & - 0.055895k_2k_3 + 1.226648k_1 + 1.271835k_2 \\ & + 0.710556k_3 - 8.112402. \end{aligned}$$

Therefore the PSS representation of the multistationarity region is the set of the points $(k_1, k_2, k_3) \in B$ that satisfy $p(k_1, k_2, k_3) \geq 1$, which is the region between the two yellow surfaces in Fig. 8b. Note that the point (5, 3, 12) corresponding to the value examined in [35] is in the middle of this region.

Bisection algorithms for rectangular representation

As shown in the previous section, it is possible to get the PSS representation both from the sampling and the rectangular representations. If one needs to solve the system at random points and take an average in order to get the rectangular representation, then using the rectangular representation does not have much advantage over using the sampling representation for the purpose of finding the PSS representation.

But in some cases it is possible to compute the average number of the solutions without solving the system. One such case is introduced in [16]. Instead of solving the system for many points, it is enough to compute one integral called the Kac-Rice integral. In this situation, if the computation of the integral is possible and faster than solving the system for many random points, then the rectangular representation can be preferred to the sampling representation. However, one still needs a computation per each sub-hyperrectangle of the rectangular representation and this number can grow by the number of parameters. If $B \subseteq \mathbb{R}^r$ and we divide it along each axis to m equal parts, then the number of sub-hyperrectangles in the rectangular representation becomes m^r . A different approach to build a rectangular decomposition is to use bisection algorithms instead of dividing B to equal sub-hyperrectangles. When using a bisection algorithm,

the rectangular decomposition usually contains fewer number of sub-hyperrectangles (not necessarily of equal volume).

Two possible number of solutions for a given parameter point

Let us simplify the question. There is a hyperrectangle $B \subseteq \mathbb{R}^r$ and a function $g : B \rightarrow \mathbb{Z}_{\geq 0} \cup \{\infty\}$ which in our case is Φ_f^0 associated to a parametric function $f_k(x)$. Let \mathcal{B} be the set containing all sub-hyperrectangles of B . The goal is to express $L_i(g)$ or $U_2(g)$ as a union of sub-hyperrectangles of B . One of the shapes of multistationary networks, observed often for realistic models, are bistable networks with a folding type of bifurcation, such as in dual phosphorylation cycle [36].

In our settings these networks have one steady state for some choices of parameters and three¹ steady states for some other choices of the parameters and for a zero measure set of parameters in the boundary of the two regions it has two steady states. This was the case for the networks studied earlier: the LacI-TetR gene regulatory, and auto-regulatory motif networks. In such cases Φ_f^0 is almost always either 1 or 3. Going back to our question, motivated from application, assume $\text{Im}(g) = \{n_1, n_2\}$ where $n_1 \not\approx n_2$. In this case for each $S \in \mathcal{B}$ one of the followings occurs.

(i) $\mathbb{E}(g(k) \mid k \sim U(S)) = n_1$.

This can happen if and only if for almost every $k \in S$, $g(k) = n_1$.

(ii) $\mathbb{E}(g(k) \mid k \sim U(S)) = n_2$

This can happen if and only if for almost every $k \in S$, $g(k) = n_2$.

(iii) $\mathbb{E}(g(k) \mid k \sim U(S)) = \alpha$, $n_1 \not\approx \alpha \not\approx n_2$.

This can happen if and only if $S \cap L_{n_1}(g)$ and $S \cap L_{n_2}(g)$ both are nonempty and of non-zero measure.

The proof is straightforward by noting that

$$\mathbb{E}(g(k) \mid k \sim U(K)) = n_1 \overline{\text{Vol}}(K \cap L_{n_1}(g)) + n_2 \overline{\text{Vol}}(K \cap L_{n_2}(g)).$$

Therefore one can compute $\mathbb{E}(g(k) \mid k \sim U(B))$. Then if the answer is n_1 conclude that almost the whole B is a subset of $L_{n_1}(g)$ and if the answer is n_2 , conclude that almost the whole B is a subset of $L_{n_2}(g)$. Otherwise we proceed by dividing B along only one axis into two equal sub-hyperrectangles. We continue in this fashion until each sub-hyperrectangle is inside $L_{n_1}(g)$ or $L_{n_2}(g)$ or a termination condition on the length of the edges of the sub-hyperrectangles is met. When the termination condition on the edges is obtained, we put the sub-hyperrectangle in $L_{n_i}(g)$ if $\mathbb{E}(g(k))$ on this sub-hyperrectangle is closer to n_i . We refer to this approach as the *two-value bisection search* hereafter. This algorithm is formally written in Algorithm 1.

¹ Two stable and one unstable, however we do not study stability of the steady states in this paper.

```

Input :  $B = [a, b] \subseteq \mathbb{R}^r$ ,  $g: B \rightarrow \{n_1, n_2\} \subseteq \mathbb{Z}_{\geq 0}$ ,
           $n_1 \not\leq n_2$ ,  $\epsilon \in \mathbb{R}_{>0}$ .
Output:  $L_{n_1}(g) \simeq \cup_{i=1}^{m_1} S_{n_1,i}$ ,  $L_{n_2}(g) \simeq \cup_{i=1}^{m_2} S_{n_2,i}$ ,
           $\forall i, j: S_{n_i,j} \in \mathcal{B}$  and the minimum of the
          length of the edges of  $S_{n_i,j} > \epsilon$ .

1 initialization:  $L_1 = \{\}$ ,  $L_2 = \{\}$ ,  $T = \{(B, 1)\}$ ;
2 while  $T \neq \emptyset$  do
3   choose the first element of  $T$  and call its first
   element by  $S$  and its second element by  $i$  and
   remove  $(S, i)$  from  $T$ ;
4   compute  $\alpha = \mathbb{E}(g(k) \mid k \sim U(S))$ ;
5   switch the value of  $\alpha$  do
6     case  $n_1$  do
7       | add  $S$  to  $L_1$ ;
8     end
9     case  $n_2$  do
10      | add  $S$  to  $L_2$ ;
11    end
12    otherwise do
13      define  $\beta$  to be the minimum of the length of
      the edges of  $S$ ;
14      if  $\beta \leq \epsilon$  then
15        | if  $\alpha \leq \frac{n_1+n_2}{2}$  then
16          | add  $S$  to  $L_1$ ;
17        else
18          | add  $S$  to  $L_2$ .
19        end
20      else
21        define  $S_1$  and  $S_2$  by dividing  $S$  along
        the  $i$ -th axis to two equal
        sub-hyperrectangles. Replace  $i$  by
         $(i + 1) \bmod n$  and add  $(S_1, i)$  and
         $(S_2, i)$  to  $T$ ;
22      end
23    end
24  end
25 end

```

Algorithm 1: A bisection algorithm for building a rectangular representation of a parameter region when there are only two possible number of solutions for the system of equations, ignoring a zero measure set of parameter values.

A similar algorithm was presented in [16, Section 2]. The first difference is that the input to the bisection algorithm in [16] is not necessarily a system with two general number of solutions. The second difference is that the output, there, is not only the two lists $L_{n_1}(g)$ and $L_{n_2}(g)$ (compare Fig. 9d–f with [16, Figure 1c]).

If the length of the edges of B are of different scales, then it is better to replace the termination condition of Algorithm 1 with the following:

$$\min \left\{ \frac{b_{S,j} - a_{S,j}}{b_{B,j} - a_{B,j}} \mid 1 \leq j \leq r \right\} \leq \epsilon,$$

where $S = [a_S, b_S], B = [a_B, b_B]$ and $0 < \epsilon < 1$. The motivation is that one usually is interested in knowing the parameter values with some number of digits of accuracy, after writing the number in scientific notation.

We are not going to explain how to use the Kac-Rice integral in the CRN framework as it is not the topic of this paper. All algorithms in this paper are independent from the choice of the algorithm to compute the expected number of solutions of a parametric system on a given parameter region. A simple algorithm to achieve this is to solve the system for several random points from the given hyperrectangle according to a random distribution of interest and take the average. Thus we may assume the existence of a method capable of computing $\mathbb{E}(\Phi_f^0(k) \mid k \sim q)$ where q is a distribution on S and then find a rectangular representation using two-valued bisection search and afterwards a PSS representation.

Example

To illustrate this method we use Example 2.1 of [16], shown here in Fig. 9a, for which the Kac-Rice integral is already derived. The system of equations for studying multistationarity is given in Fig. 9b. Fixing all values of parameters other than k_7 and k_8 to the following values (similar to the values in [16]), the goal is to find the multistationarity region of the network in the rectangle $[(0, 0), (5, 5)]$:

$$k_1 = 0.7329, k_2 = 100, k_3 = 73.29, k_4 = 50, k_5 = 100, k_6 = 5.$$

Figure 9c shows the exact region computed by CAD. Using Algorithm 1 we get the approximation of this region represented in Figs. 9d–f. We see that by decreasing the ϵ of the termination condition, the approximation is improved. Furthermore using the Kac-Rice integral given in [16] it takes 1.22, 3.64 and 7.96 s for our code written in Julia to compute the approximations in Fig. 9d–f respectively, while solving the system in 1000 points to get the sampling² representation in Matlab takes 17.63 s. We used Julia for the Kac-Rice integral because it is suggested by [16] as the fastest platform for this computation. Figure 9g, h show the PSS approximation of degrees 2 and 4 achieved from Fig. 9f. One can also generate random points from the rectangular representation and find the PSS representation from this sampling approximation. Adding the times for using the Kac-Rice integral, two-valued bisection search, generating random points, and computing PSS representation; all together for this example it took 8.53 s which is less than finding a rectangular representation by solving the system at 140 points. That result is shown in Fig. 9i.

Remark 5.1

Note that having fewer hyperrectangles in the rectangular representation obtained by the two-valued bisection search does not guarantee a faster speed than the simple approach for finding a rectangular representation discussed in the previous section. Consider the

² To get the rectangular representation by 100 equal subrectangles and solving for 10 points in each subrectangle, it is again necessary to solve the system for 1000 points.

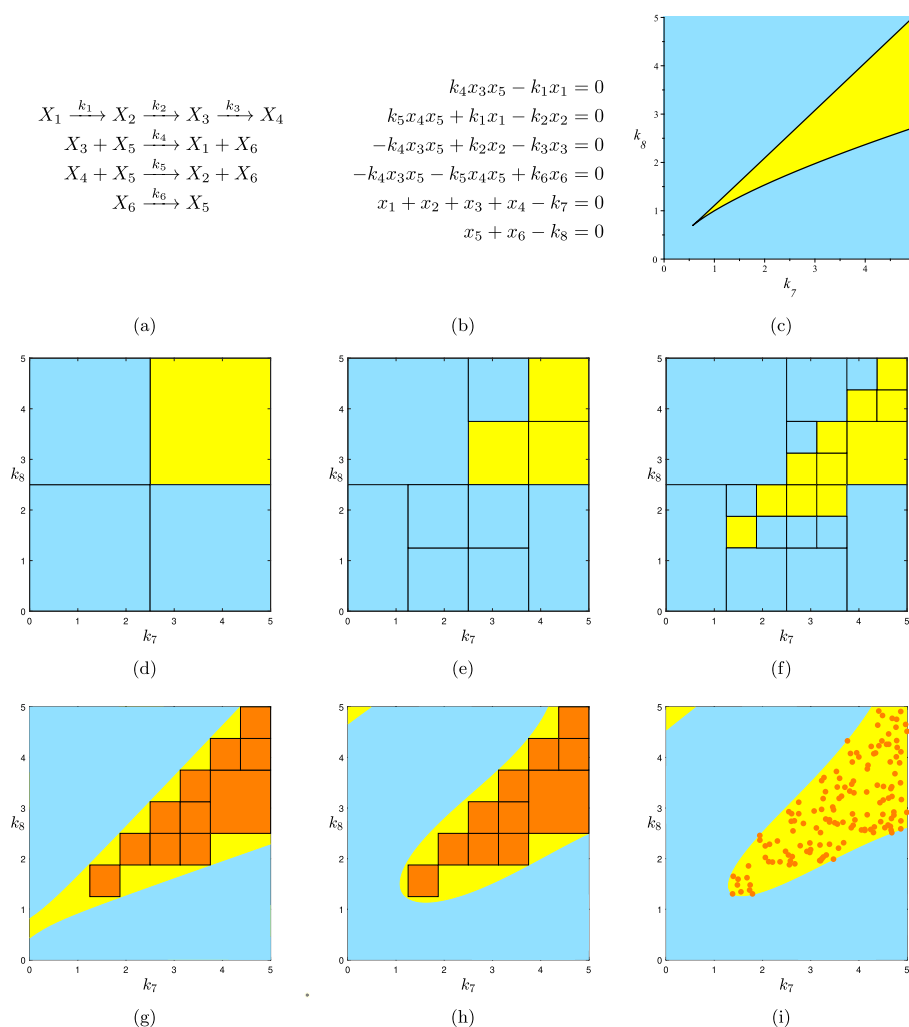


Fig. 9 Using two-valued bisection search to get the PSS representation of the multistationarity region. **a** The reaction network under study. This network has two conservation laws, thus two of the six steady state equations may be replaced by linear invariants. **b** The system of equations for studying multistationarity. **c** The CAD representation of the multistationarity region. **d–f** Approximations of the multistationarity region computed by Algorithm 1. **g–h** PSS representations of degrees 2 and 4 computed from the union of yellow rectangles in **f** as K . **i** PSS representations of degree 4 computed from 140 sample points from the union of yellow rectangles in **f** as approximation of K

setting in Algorithm 1. Assume length of all edges of $B \subseteq \mathbb{R}^r$ are the same and equal to 2^m . Let $\epsilon = 1$ and assume $\mathbb{E}(g(k) \mid k \sim U(S))$ is not getting close enough to n_1 or n_2 for any S in the process of this algorithm. Then the total number of expectations that are needed to be computed until the termination of this algorithm is equal to $\sum_{i=0}^{mr} 2^i$. On the other hand in the simple approach, by dividing B along each axis to 2^m equal parts, the number of needed expectations to be computed is 2^{mr} .

More general setting

Now consider a more general case where $\text{Im}(g) = \{n_1, \dots, n_s\} \subseteq \mathbb{Z}_{\geq 0}$. In this case we can not judge about $S \cap L_{n_i}(g)$ just by looking at $\mathbb{E}(g(k) \mid k \sim U(S))$. For example if $\text{Im}(g) = \{1, 3, 5\}$ and we receive $\mathbb{E}(g(k) \mid k \sim U(S)) = 3$, it is not clear that S is an almost subset of $L_3(g)$ or if almost half of it is inside $L_1(g)$ and the other half in $L_5(g)$. So now the goal is to find a way to decide when to add S to L_i when $\mathbb{E}(g(k) \mid k \sim U(S)) = i$ and when to not and instead bisect it into two sub-hyperrectangles, as in Algorithm 1.

Note that

$$\mathbb{E}(g(k) \mid k \sim U(S)) = n_1 \overline{\text{Vol}}(S \cap L_{n_1}(g)) + \dots + n_s \overline{\text{Vol}}(S \cap L_{n_s}(g)).$$

Assume $\{n_{\alpha_1}, \dots, n_{\alpha_t}\} \subseteq \{n_1, \dots, n_s\}$ such that $\overline{\text{Vol}}(S \cap L_i(g)) \neq 0$ only for $i \in \{n_{\alpha_1}, \dots, n_{\alpha_t}\}$. In that case for any distribution on S which has the same zero measure sets as Lebesgue measure's we have $\int_{S \cap L_i(g)} q(x) dx = 0$ if and only if $i \notin \{n_{\alpha_1}, \dots, n_{\alpha_t}\}$.

Returning to our goal assume $\mathbb{E}(g(k) \mid k \sim U(S)) = n_i$ for some $i \in \{1, \dots, s\}$. If for every $j \neq i$, $\overline{\text{Vol}}(S \cap L_{n_j}(g)) = 0$, then for any other distribution q on S we have $\int_{S \cap L_j(g)} q(x) dx = \delta_{i,j}$, where $\delta_{i,j}$ is 1 if $i = j$, and 0 for $j \neq i$. Therefore $\mathbb{E}(g(k) \mid k \sim q) = n_i$. Now again assume that $\{n_{\alpha_1}, \dots, n_{\alpha_t}\} \subseteq \{n_1, \dots, n_s\}$ such that $\overline{\text{Vol}}(S \cap L_j(g)) \neq 0$ only for $j \in \{n_{\alpha_1}, \dots, n_{\alpha_t}\}$. This time let $t \geq 2$. Define the following two sets:

$$T_1 = \{(x_1, \dots, x_t) \in (0, 1)^t \mid x_1 + \dots + x_t = 1\},$$

$$T_2 = \{(x_1, \dots, x_t) \in (0, 1)^t \mid x_1 + \dots + x_t = 1, n_{\alpha_1} x_1 + \dots + n_{\alpha_t} x_t = n_i\}.$$

Note that T_2 is a set of one dimension lower than dimension of T_1 . By varying q one can attain any point in T_1 by $(\int_{S \cap L_{n_1}(g)} q(x) dx, \dots, \int_{S \cap L_{n_t}(g)} q(x) dx)$ and $\mathbb{E}(g(k) \mid k \sim q) = n_i$ if and only if this point belongs to T_2 . Therefore by probability one for randomly chosen distribution q , we will not get $\mathbb{E}(g(k) \mid k \sim q) = n_i$. Hence we proved the following lemma.

Lemma 5.2 *Let $B \subseteq \mathbb{R}^r$ be a hyperrectangle and $g : B \rightarrow \{n_1, \dots, n_s\} \subseteq \mathbb{Z}_{\geq 0}$. Assume that $\mathbb{E}(g(k) \mid k \sim U(B)) = n_i$ for some $i \in \{1, \dots, s\}$. Then with probability one we have that B is almost subset of $L_{n_i}(g)$ if and only if $\mathbb{E}(g(k) \mid k \sim q) = n_i$ for a randomly chosen distribution q on B with the same zero measure sets as Lebesgue measure's.*

Note that in [16] it is mentioned that the Kac-Rice integral can also be used to compute the expected number of steady states when the parameters are equipped by normal distributions. Even without the Kac-Rice integral, one can solve the system of equations for random sample parameter points chosen from a (truncated) normal distribution and then take the average of the number of solutions. Thus we get Algorithm 2 which we call *two-step bisection search*.

```

Input :  $B = [a, b] \subseteq \mathbb{R}^r$ ,  $g: B \rightarrow \{n_1, \dots, n_s\} \subseteq \mathbb{Z}_{\geq 0}$ ,
 $n_1 \leq \dots \leq n_s$ ,  $\epsilon \in \mathbb{R}_{>0}$ .
Output:  $L_{n_1}(g) \simeq \cup_{i=1}^{m_1} S_{n_1, i}, \dots,$ 
 $L_{n_s}(g) \simeq \cup_{i=1}^{m_s} S_{n_s, i}, \forall i, j: S_{n_i, j} \in \mathcal{B}$  and the
minimum of the length of the edges of
 $S_{n_i, j} > \epsilon$ .
1 initialization:  $L_1 = \{\}, \dots, L_s = \{\}, T = \{(B, 1)\}$ ;
2 while  $T \neq \emptyset$  do
3   choose the first element of  $T$  and denote its first
   element by  $S$  and its second element by  $i$ . Remove
    $(S, i)$  from  $T$ ;
4   compute  $\alpha = \mathbb{E}(g(k) \mid k \sim U(S))$ ;
5   define  $\beta$  to be the minimum of the length of the
   edges of  $S$ ;
6   if  $\beta \leq \epsilon$  then
7     add  $S$  to  $L_j$  if  $\alpha$  is closer to  $n_j$ ;
8   else
9     if  $\alpha \notin \{n_1, \dots, n_s\}$  then
10      define  $S_1$  and  $S_2$  by dividing  $S$  along the
       $i$ -th axis to two equal sub-hyperrectangles.
      Replace  $i$  with  $(i + 1) \bmod n$ . Add  $(S_1, i)$ 
      and  $(S_2, i)$  to  $T$ ;
11     else
12      choose a distribution on  $S$  randomly and call
      it  $q$ , for example choose a random point
      from  $S$  and call it  $k^*$ , then let  $q$  be the
      truncated normal distribution on  $S$  with
      mean being  $k^*$  and let the variance to be a
      number not too small and not too large
      comparing to the length of the edges of  $S$ ;
13      compute  $\gamma = \mathbb{E}(g(k) \mid k \sim q)$ ;
14      if  $\gamma = n_j$  then
15        add  $S$  to  $L_j$ ;
16      else
17        define  $S_1$  and  $S_2$  by dividing  $S$  along
        the  $i$ -th axis to two equal
        sub-hyperrectangles. Replace  $i$  with
         $(i + 1) \bmod n$ . Add  $(S_1, i)$  and
         $(S_2, i)$  to  $T$ ;
18      end
19    end
20  end
21 end

```

Algorithm 2: Bisection algorithm without restriction on the possible number of solutions for the system of equations.

Example

Consider the following univariate polynomial of degree 5 with two parameters from [16, Section 2.3].

$$\begin{aligned}
 f(x) = & x^5 - (k_1 + \frac{9}{2})x^4 + (\frac{9}{2}k_1 + \frac{21}{4})x^3 + (-\frac{23}{4}k_1 + \frac{3}{8})x^2 \\
 & + (\frac{15}{8}k_1 - \frac{23}{8})x + (\frac{1}{100}k_2 - \frac{1}{16})
 \end{aligned}$$

The equation $f(x) = 0$ can obtain any number of positive real solutions between 0 and 5 depending on the choice of the parameters. Using the MCKR application [37] which computes the Kac-Rice integral to give the expected number of solutions of a parametric system when the parameters are equipped with a random distribution, we have that the average number of positive solutions of $f(x) = 0$ on the two following rectangles is 2 with at least one decimal place accuracy:

$$B_1 = [(0.5, 8), (1, 9)], \quad B_2 = [(2, 2), (2.5, 3)].$$

However, only the first rectangle is inside the parameter region where the number of positive solutions to the system is invariant and equal to 2. Using MCKR, this time we compute the expected number of positive solutions of $f(x) = 0$ on B_2 when the parameters are equipped with the truncated normal distribution with mean $\mu = (2.25, 2.5)$ and variance $\sigma^2 = 0.1$. The result with one digit accuracy after the decimal point is 2.9. Thus by Lemma 5.2 we can infer the fact that the number of solutions of the system is not invariant in B_2 and this set has a non-zero measure subset where the system has something other than 2 positive solutions.

Acknowledgements

Thanks to Hamed Baghal Ghafari for reading and discussing the earlier drafts of the paper and helpful comments about Matlab and LATEX.

Author contributions

A.S. developed the PSS representation theory and the implementation. M.E. helped organise, clarifying and explain the contributions. Both authors read and approved the final manuscript.

Funding

The authors acknowledge the support of EPSRC Grant EP/T015748/1, "Pushing Back the Doubly-Exponential Wall of Cylindrical Algebraic Decomposition". AmirHosein Sadeghimanesh was also supported by NKFIH KKP 129877 before moving to Coventry University.

Availability of data and materials

All the code and data underpinning the results of this paper is openly available from this URL: <https://doi.org/10.5281/zenodo.6927946>.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

Not applicable.

Received: 31 January 2022 Accepted: 5 September 2022

Published online: 27 September 2022

References

1. Feinberg M. Chemical reaction network structure and the stability of complex isothermal reactors—I. The deficiency zero and deficiency one theorems. *Chem Eng Sci*. 1987;42(10):2229–68. [https://doi.org/10.1016/0009-2509\(87\)80099-4](https://doi.org/10.1016/0009-2509(87)80099-4).
2. Feinberg M. Chemical reaction network structure and the stability of complex isothermal reactors—II. Multiple steady states for networks of deficiency one. *Chem Eng Sci*. 1988;43(1):1–25. [https://doi.org/10.1016/0009-2509\(88\)87122-7](https://doi.org/10.1016/0009-2509(88)87122-7).
3. Feinberg M. Multiple steady states for chemical reaction networks of deficiency one. *Arch Rational Mech Anal*. 1995;132(4):371–406. <https://doi.org/10.1007/BF00375615>.
4. Ellison PR. The advanced deficiency algorithm and its applications to mechanism discrimination. PhD thesis, The University of Rochester, Eastman School of Music. 1998. <https://doi.org/10.5555/927930>.

5. Müller S, Feliu E, Regensburger G, Conradi C, Shiu A, Dickenstein A. Sign conditions for injectivity of generalized polynomial maps with applications to chemical reaction networks and real algebraic geometry. *Found Comput Math*. 2016;16(1):69–97. <https://doi.org/10.1007/s10208-014-9239-3>.
6. Ji H, Ellison PR, Knight D, Feinberg M. CRNTToolBox Version 2-3—The Chemical Reaction Toolbox. <http://crnt.osu.edu/CRNTWin>. 2015.
7. Sadeghimanesh A, Feliu E. The multistationarity structure of networks with intermediates and a binomial core network. *Bull Math Biol*. 2019;81(7):2428–62. <https://doi.org/10.1007/s11538-019-00612-1>.
8. Donnell P, Banaji M, Marginean A, Pantea C. CoNtRol: an open source framework for the analysis of chemical reaction networks. *Bioinformatics*. 2014;30(11):1633–4. <https://doi.org/10.1093/bioinformatics/btu063>.
9. Joshi B. Complete characterization by multistationarity of fully open networks with one non-flow reaction. *Appl Math Comput*. 2013;219(12):6931–45. <https://doi.org/10.1016/j.amc.2013.01.027>.
10. Bradford R, Davenport JH, England M, Errami H, Gerdt V, Grigoriev D, Hoyt C, Košta M, Radulescu O, Sturm T, Weber A. Identifying the parametric occurrence of multiple steady states for some biological networks. *J Symb Comput*. 2020;98:84–119. <https://doi.org/10.1016/j.jsc.2019.07.008>. Special Issue on Symbolic and Algebraic Computation: ISSAC 2017
11. Flockerzi D, Holstein K, Conradi C. N-site phosphorylation systems with $2N-1$ steady states. *Bull Math Biol*. 2014;76(8):1892–916. <https://doi.org/10.1007/s11538-014-9984-0>.
12. Nam K-M, Gyori BM, Amethyst SV, Bates DJ, Gunawardena J. Robustness and parameter geography in post-translational modification systems. *PLOS Comput Biol*. 2020;16(5):1–50. <https://doi.org/10.1371/journal.pcbi.1007573>.
13. Pušnik Ž, Mraz M, Zimic N, Moškon M. Computational analysis of viable parameter regions in models of synthetic biological systems. *J Biol Eng*. 2019;13(1):75. <https://doi.org/10.1186/s13036-019-0205-0>.
14. Shah NA, Sarkar CA. Robust network topologies for generating switch-like cellular responses. *PLOS Comput Biol*. 2011;7(6):1–12. <https://doi.org/10.1371/journal.pcbi.1002085>.
15. Chau A, Walter J, Gerardin J, Tang C, Lim W. Designing synthetic regulatory networks capable of self-organizing cell polarization. *Cell*. 2012;151(2):320–32. <https://doi.org/10.1016/j.cell.2012.08.040>.
16. Feliu E, Sadeghimanesh A. Kac-rice formulas and the number of solutions of parametrized systems of polynomial equations. *Math Comput*. 2022;91:2739–2769. <https://doi.org/10.1090/mcom/3760>.
17. Dabbene F, Henrion D, Lagoa C. Simple approximations of semialgebraic sets and their applications to control. *Automatica*. 2017;78:110–8. <https://doi.org/10.1016/j.automatica.2016.11.021>.
18. Dabbene F, Henrion D. Set approximation via minimum-volume polynomial sublevel sets. In: European control conference (ECC), Zurich, Switzerland, 2013. p. 11. <https://hal.archives-ouvertes.fr/hal-00740794>
19. Sadeghimanesh A. Algebraic tools in the study of multistationarity of chemical reaction networks. PhD thesis, University of Copenhagen. 2018. <http://web.math.ku.dk/noter/filer/phd18ahs.pdf>.
20. Karlebach G, Shamir R. Modelling and analysis of gene regulatory networks. *Nat Rev Mol Cell Biol*. 2008;9(10):770–80. <https://doi.org/10.1038/nrm2503>.
21. Mahdi A, Ferragut A, Valls C, Wiuf C. Conservation laws in biochemical reaction networks. *SIAM J Appl Dyn Syst*. 2017;16(4):2213–32. <https://doi.org/10.1137/17M1138418>.
22. Basu S, Pollack R, Roy M-F. Algorithms in real algebraic geometry, 2nd edn. Springer. 2006. <https://doi.org/10.1007/3-540-33099-2>.
23. Lichtblau D. Symbolic analysis of multiple steady states in a MAPK chemical reaction network. *J Symb Comput*. 2021;105:118–44. <https://doi.org/10.1016/j.jsc.2020.06.004>.
24. Röst G, Sadeghimanesh A. Exotic bifurcations in three connected populations with Allee effect. *Int J Bifurc Chaos*. 2021;31(13):2150202. <https://doi.org/10.1142/S0218127421502023>.
25. Lazard D, Rouillier F. Solving parametric polynomial systems. *J Symb Comput*. 2007;42(6):636–67. <https://doi.org/10.1016/j.jsc.2007.01.007>.
26. Moroz G. Sur la décomposition réelle et algébrique des systèmes dépendant de paramètres. PhD thesis, Université Pierre et Marie Curie—Paris VI. 2008. https://tel.archives-ouvertes.fr/tel-00812436/file/these_moroz.pdf.
27. Wilson D, Bradford R, Davenport JH, England M. Cylindrical algebraic sub-decompositions. *Math Comput Sci*. 2014;8(2):263–88. <https://doi.org/10.1007/s11786-014-0191-z>.
28. Gerhard J, Jeffrey DJ, Moroz G. A package for solving parametric polynomial systems. *ACM Commun Comput Algebra Sigsam*. 2010;43(3/4):61–72. <https://doi.org/10.1145/1823931.1823933>.
29. Bradford R, Davenport JH, England M, McCallum S, Wilson D. Truth table invariant cylindrical algebraic decomposition. *J Symb Comput*. 2016;76:1–35. <https://doi.org/10.1016/j.jsc.2015.11.002>.
30. Siegal-Gaskins D, Grotewold E, Smith GD. The capacity for multistability in small gene regulatory networks. *BMC Syst Biol*. 2009;96(3):66. <https://doi.org/10.1186/1752-0509-3-96>.
31. Löfberg J. YALMIP: a toolbox for modeling and optimization in MATLAB. In: 2004 IEEE international conference on robotics and automation (IEEE Cat. No.04CH37508), 2004. p. 284–9. <https://doi.org/10.1109/CACSD.2004.1393890>,
32. Löfberg J. Pre- and post-processing sum-of-squares programs in practice. *IEEE Trans Autom Control*. 2009;54(5):1007–11. <https://doi.org/10.1109/TAC.2009.2017144>.
33. Sturm JF. Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones. *Optim Methods Softw*. 1999;11(1–4):625–53. <https://doi.org/10.1080/10556789908805766>.
34. de la Fuente A. Mathematical methods and models for economists. Cambridge: Cambridge University Press; 2000. <https://doi.org/10.1017/CBO9780511810756>.
35. Voit EO. Smooth bistable S-systems. *IEE Proc Syst Biol*. 2005;152(4):207–13. <https://doi.org/10.1049/ip-syb:20050063>.
36. Markevich NI, Hoek JB, Kholodenko BN. Signaling switches and bistability arising from multisite phosphorylation in protein kinase cascades. *J Cell Biol*. 2004;164(3):353–9. <https://doi.org/10.1083/jcb.200308060>.
37. Sadeghimanesh A, Feliu E. MCKR Version 1.0. 2020. <https://doi.org/10.5281/zenodo.4085165>.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.