Identifying the Parametric Occurrence of Multiple Steady States for some Biological Networks

Bradford, R., Davenport, J. H., England, M., Errami, H., Gerdt, V., Grigoriev, D., Hoyt, C., Kosta, M., Radulescu, O., Sturm, T. & Weber, A.

Accepted Manuscript

Identifying the parametric occurrence of multiple steady states for some biological networks

Russell Bradford, James H. Davenport, Matthew England, Hassan Errami, Vladimir Gerdt et al.
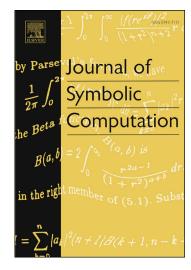
Identifying the Parametric Occurrence of Multiple Steady States for some Biological Networks

Russell Bradford[a], James H. Davenport[a], Matthew England[b,*], Hassan Errami[c], Vladimir Gerdt[d], Dima Grigoriev[e], Charles Hoyt[f], Marek Košta[g], Ovidiu Radulescu[h], Thomas Sturm[i,j], Andreas Weber[c]

[a]*Department of Computer Science, University of Bath, UK*
[b]*Faculty of Engineering, Environment and Computing, Coventry University, UK*
[c]*Institute for Informatics, University of Bonn, Germany*
[d]*Joint Institute for Nuclear Research (JINR), Dubna, Russian Federation*
and Friendship University of Russia (RUDN University), Moscow, Russian Federation
[e]*CNRS & University of Lille, France*
[f]*Department of Life Science Informatics, B-IT, University of Bonn, Germany*
[g]*Slovak Academy of Sciences, Slovakia*
[h]*DIMNP, University of Montpellier, France*
[i]*CNRS, Inria, and the University of Lorraine, Nancy, France*
[j]*MPI Informatics and Saarland University, Saarbrücken, Germany*

Abstract

We consider a problem from biological network analysis of determining regions in a parameter space over which there are multiple steady states for positive real values of variables and parameters. We describe multiple approaches to address the problem using tools from Symbolic Computation. We describe how progress was made to achieve semi-algebraic descriptions of the multistationarity regions of parameter space, and compare symbolic and numerical methods.

The biological networks studied are models of the mitogen-activated protein kinases (MAPK) network which has already consumed considerable effort using special insights into its structure of corresponding models. Our main example is a model with 11 equations in 11 variables and 19 parameters, 3 of which are of interest for symbolic treatment. The model also imposes positivity conditions on all variables and parameters.

We apply combinations of symbolic computation methods designed for mixed equality / inequality systems, specifically virtual substitution, lazy real triangularization and cylindrical algebraic decomposition, as well as a simplification technique adapted from Gaussian elimination and graph theory. We are able to determine multistationarity of our main example over a 2-dimensional parameter space. We also study a second MAPK model and a symbolic grid sampling technique which can locate such regions in 3-dimensional parameter space.

Keywords: Mixed Equation / Inequality Solving, Real Quantifier Elimination, Biological Networks, Signaling Pathways, MAPK

*Corresponding author
Email addresses: R.J.Bradford@bath.ac.uk (Russell Bradford), J.H.Davenport@bath.ac.uk (James H. Davenport), Matthew.England@coventry.ac.uk (Matthew England), errami@cs.uni-bonn.de (Hassan

1. Introduction

In this work we describe the application of combinations of symbolic computation methods in various computer algebra systems to a key problem from computational biology. The work serves to demonstrate how recent advances in such algorithms, and crucially their effective combination, allows for their application on problem instances previously thought beyond reach. In this introduction we start by describing the biological networks that are our topic of study, and highlight previous relevant work. We then outline the remainder of the paper and clarify the relationship of this article to prior work.

1.1. Multistationarity

The mathematical modelling of intra-cellular biological processes has been using nonlinear ordinary differential equations since the early ages of mathematical biophysics in the 1940s and 50s (Rashevsky, 1960). A standard modelling choice for cellular circuitry is to use chemical reactions with mass action law kinetics, leading to polynomial differential equations. Rational functions kinetics, for instance the Michaelis-Menten kinetics, can generally be decomposed into several mass action steps.

An important property of biological systems is their *multistationarity* by which we mean their having multiple stable steady states. It is instrumental to cellular memory and cell differentiation during development or regeneration of multicellular organisms and is also used by micro-organisms in survival strategies.

It is thus important to determine the parameter values for which a biochemical model is multistationary. As demonstrated in the next section, with mass action reactions, testing for multiple steady states boils down to counting real positive solutions of algebraic systems and so is suitable for study with Symbolic Computation and Computer Algebra Systems.

The models studied in this paper concern intracellular signaling pathways. These pathways transmit information about the cell environment by inducing cascades of protein modifications (phosphorylation) all the way from the plasma membrane via the cytosol to genes in the cell nucleus. Multistationarity of signaling usually occurs as a result of activation of upstream signaling proteins by downstream components (Bhalla and Iyengar, 1999). A different mechanism for producing multistationarity in signaling pathways was proposed by Markevich et al. (2004). In this mechanism the cause of multistationarity are multiple phosphorylation/ dephosphorylation cycles that share enzymes. A simple, two steps phosphorylation/dephosphorylation cycle is capable of ultrasensitivity, a form of all or nothing response with no multiple steady states (the Goldbeter–Koshland mechanism). In multiple phosphorylation/dephosphorylation cycles, enzyme sharing provides competitive interactions and positive feedback that ultimately leads to multistationarity (Markevich et al., 2004; Legewie et al., 2007).

1.2. Bistability

Multistationarity has important consequences on the capacity of signaling pathways to process biological signals, even in its elementary form of two stable steady states. This is known as *bistability* and is present in our case study problems. Bistable switches can act as memory

Errami), gerdt@jinr.ru (Vladimir Gerdt), dmitry.grigoryev@univ-lille.fr (Dima Grigoriev),
cthoyt@gmail.com (Charles Hoyt), marek.kosta@savba.sk (Marek Košta),
ovidiu.radulescu@umontpellier.fr (Ovidiu Radulescu), thomas@thomas-sturm.de (Thomas Sturm),
weber@cs.uni-bonn.de (Andreas Weber)

2

circuits storing the information needed for later stages of processing (Weng et al., 1999). The response of bistable signaling pathways shows hysteresis, namely dynamic and static lags between input and output. Because of hysteresis one can have, at the same time, a sharp binary response and protection against chatter noise.

1.3. Prior Symbolic Work

Our study is complementary to works applying numerical methods to ordinary differential equations models used for biology applications. Gross et al. (2016a) used polynomial homotopy continuation methods for global parameter estimation of mass action models. Bifurcations and multistationarity of signaling cascades was studied with numerical methods based on the Jacobian matrix by Zumsande and Gross (2010).

Algorithmically the task will be to count the positive real solutions of a parameterised system of polynomial or rational systems, making symbolic methods a possible tool. Due to the high computational complexity of this task (Grigoriev and Vorobjov, 1988) considerable work has been done to use specific properties of networks and to investigate the potential of multistationarity of a biological network out of the network structure.

This only determines whether or not there exist rate constants allowing multiple steady states, instead of coming up with a semi-algebraic description of the range of parameters yielding this property. These approaches can be traced back to the origins of Feinberg's *Chemical Reaction Network Theory* (CRNT) whose main result is that networks of deficiency 0 have a unique positive steady state for all rate constants (Feinberg, 1987; Craciun et al., 2009). We refer to Conradi et al. (2008); Millán and Turjanski (2015); Johnston (2014), and Conradi et al. (2017) for the use of CRNT and other graph theoretic methods to determine potential existence of multiple positive steady states, with Joshi and Shiu (2015) giving a survey.

Given a bistable mechanism it is also important to compute the bistability domains in parameter space: the parameter values for which there is more than one stable steady state. The size of bistability domains gives the spread of the hysteresis and quantifies the robustness of the switches. The work of Wang and Xia (2005) is relevant here: they used symbolic tools, including cylindrical algebraic decomposition as we do, to determine the number of steady states and their stability for several systems. They reported results up to a 5-dimensional system using specified parameter values, but their method is extensible to parametric questions. Higher-dimensional systems were studied using sign conditions on the coefficients of the characteristic polynomial of the Jacobian. In some cases these guarantee uniqueness of the steady state (Conradi and Mincheva, 2014).

1.4. Outline and New Contributions

In Section 2 we outline the particular biological model and symbolic problem that we aim to solve: BioModel 26 of the MAPK network, which can be found as Model 26 in the BioModels Database of (Li et al., 2010).

In Sections 3 and 4 we describe two independent symbolic attempts to solve the problem. The first in Section 3 is able to identify symbolically the multistationarity regions of a 1-dimensional parameter space with a combination of Virtual Substitution and Cylindrical Algebraic Decomposition in the Redlog package for Reduce. The second in Section 4 goes on to give full semi-algebraic solution formulae with a combination of Real Triangularization and Cylindrical Algebraic Decomposition using the Regular Chains Library for Maple. The solutions were obtained in different computer algebra systems using different fundamental algorithms, but all from the

3

family of methods for real quantifier elimination. We move on in Section 5 to describe a new pre-processing method for the problems inspired by graph theory and Gaussian elimination. Then in Section 6 we describe how a combination of ideas from all three preceding sections can be combined to provide solutions over a 2-dimensional parameter space.

In Section 7 we discuss testing the stability of fixed points. Then in Section 8 we consider an alternative larger model from the MAPK network (Model 28 in the BioModels Database of (Li et al., 2010)). In Section 9 we compare the models and detour to describe a symbolic grid sampling approach to this problem, including a comparison of this to a leading numerical solver. We consider how further progress could be achieved in Section 10, identifying a conjecture for determining where multistationarity for MAPK may occur without the costly calculations described. Finally we summarise and give final thoughts in Section 11.

This journal article follows conference papers at ISSAC 2017 (Bradford et al., 2017) and CASC 2017 (England et al., 2017). The present article reproduces this material clarifying, correcting and extending in places. In particular, Sections 3 and 4 were largely described in the ISSAC 2017 paper and Sections 5 and 9 in the CASC 2017 paper. The most notable new contributions are in Section 6, where we describe for the first time semi-algebraic solutions with two free parameters; and in Section 10, where we identify a promising conjecture for investigation.

2. Problem Outline

2.1. MAPK Bio-Model 26

The model of the MAPK cascade we investigate can be found in the BioModels Database (Li et al., 2010) as Model 26[1]. This is the first version of the models proposed by Markevich et al. (2004) corresponding to the so-called distributive ordered phosphorylation / dephosphorylation mechanism. Hereafter we will refer to it as Model 26.

It is given by the following set of differential equations. We have renamed the species names to $x_1, \ldots, x_{11}$ and the rate constants to $k_1, \ldots, k_{16}$ to facilitate reading. As usual $\dot{x}$ means the time derivative of x.

$$\dot{x}_1 = k_2 x_6 + k_{15} x_{11} - k_1 x_1 x_4 - k_{16} x_1 x_5$$
$$\dot{x}_2 = k_3 x_6 + k_5 x_7 + k_{10} x_9 + k_{13} x_{10} - x_2 x_5 (k_{11} + k_{12}) - k_4 x_2 x_4$$
$$\dot{x}_3 = k_6 x_7 + k_8 x_8 - k_7 x_3 x_5$$
$$\dot{x}_4 = x_6 (k_2 + k_3) + x_7 (k_5 + k_6) - k_1 x_1 x_4 - k_4 x_2 x_4$$
$$\dot{x}_5 = k_8 x_8 + k_{10} x_9 + k_{13} x_{10} + k_{15} x_{11} -$$
$$\qquad x_2 x_5 (k_{11} + k_{12}) - k_7 x_3 x_5 - k_{16} x_1 x_5$$
$$\dot{x}_6 = k_1 x_1 x_4 - x_6 (k_2 + k_3)$$
$$\dot{x}_7 = k_4 x_2 x_4 - x_7 (k_5 + k_6)$$
$$\dot{x}_8 = k_7 x_3 x_5 - x_8 (k_8 + k_9)$$
$$\dot{x}_9 = k_9 x_8 - k_{10} x_9 + k_{11} x_2 x_5$$
$$\dot{x}_{10} = k_{12} x_2 x_5 - x_{10} (k_{13} + k_{14})$$
$$\dot{x}_{11} = k_{14} x_{10} - k_{15} x_{11} + k_{16} x_1 x_5. \tag{1}$$

---

[1]www.ebi.ac.uk/biomodels-main/BIOMD0000000026

4

Later, we will use $\overline{(1)}$ to refer to (1) with all the left hand sides replaced by 0 in order to find fixed points of the system. The BioModels Database gives us meaningful values for the rate constants:

$$
\begin{aligned}
&k_1 = 0.02, &&k_2 = 1, &&k_3 = 0.01, &&k_4 = 0.032, \\
&k_5 = 1, &&k_6 = 15, &&k_7 = 0.045, &&k_8 = 1, \\
&k_9 = 0.092, &&k_{10} = 1, &&k_{11} = 0.01, &&k_{12} = 0.01, \\
&k_{13} = 1, &&k_{14} = 0.5, &&k_{15} = 0.086, &&k_{16} = 0.0011. &&(2)
\end{aligned}
$$

Some of these values are accurately measured and some are *well-educated guesses*. For the purpose of our study we assume they are all suitable.

We may add three linear conservation constraints to this system, which in turn introduce three further constant parameters k_{17}, k_{18}, k_{19}:

$$
\begin{aligned}
x_5 + x_8 + x_9 + x_{10} + x_{11} &= k_{17} \\
x_4 + x_6 + x_7 &= k_{18} \\
x_1 + x_2 + x_3 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} &= k_{19}.
\end{aligned}
\qquad (3)
$$

Computations to produce these, for example in MathWorks SimBiology, use the left-null space of the stoichiometric matrix under positivity conditions. For details see for example Schuster and Höfer (1991).

The constants k_{17}, k_{18}, and k_{19} represent total initial concentrations of cell substances, and meaningful values are harder to obtain than for (2). The following are some realistic value estimates, used by Markevich et al. (2004):

$$
k_{17} = 100, \qquad k_{18} = 50, \qquad k_{19} \in [200, 500]. \qquad (4)
$$

These should be considered significantly less reliable than those in (2). Indeed, the long-term goal of our research is to treat all three of these together parametrically, although in the present work we produce results only with $0 - 2$ of these parameters free.

Our computational biology problem is to identify regions in (k_{17}, k_{18}, k_{19}) parameter space over which the system formed by the unions of constraints in (1) and (3) under estimates (2) exhibits multistationarity.

The system has several special structure properties, e.g. it is a so called MESSI system (Millán and Dickenstein, 2018). However, in the following we will not directly use this structure property. The non-linearities occurring in the system are at most quadratic. As by introducing new variables the general polynomial case can be reduced to such a case and from a dynamical systems perspective point of view already quadratic systems are capable to generate all kinds of structurally stable dynamics including chaos (Vakulenko et al., 2015) this property is not restrictive.

2.2. Real Algebraic Problem

To identify fixed points we formulate a real algebraic problem by first replacing the left hand sides of all equations in (1) with 0, which as noted above we denote $\overline{(1)}$. This, together with the equations in (3), yields an algebraic system with polynomials in

$$
F \subset \mathbb{Z}[k_1, \ldots, k_{19}][x_1, \ldots, x_{11}].
$$

However, ideal theory is not sufficient, as we are concerned only with real valued solutions. Further, we have the additional inequality restrictions that all entities in our model are strictly positive. This yields an additional system

$$P = \{k_1, \ldots, k_{19}, x_1, \ldots, x_{11}\} \subset \mathbb{Z}[k_1, \ldots, k_{19}][x_1, \ldots, x_{11}]$$

establishing a side condition on the solutions of F that all variables x_i and parameters k_i of P be positive. In terms of first-order logic our specification of F and P yields a quantifier-free Tarski formula,

$$\varphi = \bigwedge_{f \in F} f = 0 \wedge \bigwedge_{v \in P} v > 0. \tag{5}$$

The estimations for the rate constants in (2) formally establish a substitution rule

$$\sigma = [0.02/k_1, \ldots, 0.0011/k_{16}]$$

in postfix notation, which can be applied to F, P, or φ. Applying this to φ; converting the floats from (2) into rational numbers; and multiplying over common denominators, gives us the quantifier-free Tarski formula ψ below.

$$
\begin{aligned}
\psi = {}& -200x_1x_4 - 11x_1x_5 + 860x_{11} + 10000x_6 = 0 \\
& \wedge -16x_2x_4 - 10x_2x_5 + 500x_{10} + 5x_6 + 500x_7 + 500x_9 = 0 \\
& \wedge -9x_3x_5 + 3000x_7 + 200x_8 = 0 \\
& \wedge -10x_1x_4 - 16x_2x_4 + 505x_6 + 8000x_7 = 0 \\
& \wedge -11x_1x_5 - 200x_2x_5 - 450x_3x_5 + 10000(x_8 + x_9 + x_{10}) + 860x_{11} = 0 \\
& \wedge 2x_1x_4 - 101x_6 = 0 \\
& \wedge 4x_2x_4 - 2000x_7 = 0 \\
& \wedge 45x_3x_5 - 1092x_8 = 0 \\
& \wedge 5x_2x_5 + 46x_8 - 500x_9 = 0 \\
& \wedge x_2x_5 - 150x_{10} = 0 \\
& \wedge 11x_1x_5 + 5000x_{10} - 860x_{11} = 0 \\
& \wedge -k_{17} + x_{10} + x_{11} + x_5 + x_8 + x_9 = 0 \\
& \wedge -k_{18} + x_4 + x_6 + x_7 = 0 \\
& \wedge -k_{19} + x_1 + x_{10} + x_{11} + x_2 + x_3 + x_6 + x_7 + x_8 + x_9 = 0 \\
& \wedge x_1 > 0 \wedge x_2 > 0 \wedge x_3 > 0 \wedge x_4 > 0 \wedge x_5 > 0 \\
& \wedge x_6 > 0 \wedge x_7 > 0 \wedge x_8 > 0 \wedge x_9 > 0 \wedge x_{10} > 0 \wedge x_{11} > 0 \\
& \wedge k_{17} > 0 \wedge k_{18} > 0 \wedge k_{19} > 0. \tag{6}
\end{aligned}
$$

Our problem in real algebra is to obtain a semi-algebraic description of those regions in the (k_{17}, k_{18}, k_{19}) parameter-space where there are multiple solutions of (6). The multistationarity problem also requires us to know about the stability of these solutions, as discussed in Section 7.

2.3. Suitable Symbolic Technology

This real algebraic problem is amenable to technology developed for real quantifier elimination. Note that the number of indeterminates (variables and parameters) is high compared to

those usually tackled by such technology. However, the degrees involved are low, with every monomial at most degree 2, which helps make it tractable.

As we will not include a priori information about the stability of the fixed points, we must not only consider the existence of (at least) two stable fixed points but also unstable fixed points. Hence we simply investigate where in parameter space there exist multiple different roots $\mathbf{x} \in (0, \infty)^{11}$ of F.

In theory, any *Real Quantifier Elimination* (QE) technology can directly handle the parametric existence of steady states, taking as input $\exists x_1 \ldots \exists x_{11}\varphi$ and producing as output a quantifier free formula in the parameters describing where solutions exists. However, this is not sufficient to solve our problem as we are not only interested in the existence but also in the number of solutions. We can use a specific QE tool to do this: Cylindrical Algebraic Decomposition.

2.3.1. Cylindrical algebraic decomposition and its terminology

Cylindrical Algebraic Decomposition (CAD) was first proposed by Collins in the 1970s. This original algorithm[2] took as input a set of polynomials in $\mathbb{Z}[x_1, \ldots, x_N]$, producing as output a set of *cells* which together give a decomposition of $\mathbb{R}^n$ which is *sign-invariant*, meaning each input polynomial has constant sign over each cell. The sign-invariance means that the polynomials may be studied over an an infinite domain by querying a finite number of sample points: one per cell.

The cells are all semi-algebraic, meaning they can be described by a polynomial system, and they are arranged *cylindrically*, meaning their projections with respect to a stated variable ordering are either equal or disjoint. The cylindricity means the semi-algebraic descriptions are triangular and the cells form cylinders over another (induced) CAD of $\mathbb{R}^{n-1}$ given by the projection of the *n*-dimensional cells. All cells are either *sections*, defined by a polynomial vanishing; or a *sector*, defined as the space between two sections, or possibly extending infinitely.

Collins' algorithm proceeded with a system of: projection, which identified key polynomials in fewer variables; and lifting, where the induced CADs are incrementally constructed via substitution of sample points and univariate root isolation. The act of projection must be defined so that working at a sample point may be concluded representative for the entire cell.

There has been numerous extensions and improvements to CAD since Collins' original method. The collection edited by Caviness and Johnson (1998) is a key resource; in particular the survey paper within by Collins (1998). A more recent survey was given in the Introduction section of the work by Bradford et al. (2016). A key choice for CAD is the variable ordering which defines the cylindricity property and controls the order steps are taken by the algorithm. For use in quantifier elimination CAD must project variables in the order they are quantified. Our problem (6) is not quantified but our desire to understand the problem over parameter space means that we must project variables before parameters. However, besides this the choice is free for us. We define the *main variable* of a polynomial / constraint to be the highest one present (first to be projected) in the ordering.

The worst-case time complexity of CAD is doubly exponential. Traditionally, this is doubly exponential in the number of indeterminates, which would include our symbolically treated parameters. However recent progress on CAD in the presence of equational constraints (see for example the work of England et al. (2015)), of which there are many in (6), allows us to conclude it is actually doubly-exponential in the number of variables minus the number of equational

---

[2]see for example the work of Arnon et al. (1984).

constraints at different levels of the projection (England and Davenport, 2016). Despite this, the number of variables present in (6) is too large for contemporary CAD implementations to tackle alone.

2.3.2. Combing with other symbolic tools

We are able to make progress by combining CAD with additional symbolic methods. Two independent investigations were undertaken. The first, described in Section 3, uses the Redlog package in Reduce and combines CAD with virtual substitution. The second, described in Section 4, uses the Regular Chains Library in Maple and combines CAD with real triangularization. In both cases we have combined the corresponding methods by hand, but automation is clearly possible.

3. Using Real Quantifier Elimination Technology in Redlog

In this section we are going to combine *Virtual Substitution* (VS) with CAD. The former smoothly eliminates the majority of the quantifiers while the latter allows us to count numbers of solutions via decomposition of the remaining low-dimensional spaces. That combination of methods requires the solution of several QE runs with each problem and some combinatorial arguments. Throughout this section we are performing computations using the Redlog Package (Dolzmann and Sturm, 1997a) for Reduce revision r3606. Timings are reported for a 2.4 GHz Intel Core i7 with 3 GB RAM or cores on a compute server with similar speed and memory limitations.

3.1. Virtual Substitution

Substitution methods for quantifier elimination date back to an article from Weispfenning (1988), which treated the special case with only linear occurrences of the quantified variables. Originally motivated by the proof of tight complexity bounds for the real decision problem, that approach turned out to be applicable to practical problems, especially with many parameters. Consequently, the method was systematically generalized by Weispfenning and his students to arbitrary but bounded degrees (Weispfenning, 1997, 1994; Košta, 2016).

Quantifier elimination proceeds from the inside to the outside of a prenex quantifier block. An innermost existential quantifier is eliminated by equivalently replacing it with a finite disjunction:

$$\mathrm{VS}(\exists x_n \varphi) := \bigvee_{t \in E} \varphi[t /\!\!/ x_n],$$

where E is a finite *elimination set* containing abstract *test points* $t = (\gamma, z)$. The terms z are derived from symbolic representations of formal zeros of parametric univariate polynomials from $\mathbb{Z}[x_1, \ldots, x_{n-1}][x_n]$ occurring in φ with possibly adding infinitesimals $\pm \varepsilon$. They are guarded by quantifier-free formulas $\gamma(x_1, \ldots, x_{n-1})$ that guarantee the existence of the zeros in terms of the parameters. Recall that regular term substitution maps terms to terms, which naturally generalizes to corresponding maps on quantifier-free formulas. *Virtual substitution* $[t /\!\!/ x_n]$, in contrast, maps atomic formulas to quantifier-free formulas. This allows to express the substitution of the terms z without using any non-standard symbols. Furthermore, virtual substitution adds the guarding conditions γ in a suitable way. For examples and surveys of the virtual substitution method see the work of Sturm (2017, 2018).

8

3.2. Parameter Free Computations

We start by considering the case where all parameters in (5) are substituted for their estimates in (2) and (4) (interpreted as rational numbers):

$$\varphi_{500} = \varphi\sigma[100/k_{17}, 50/k_{18}, 500/k_{19}].$$

The closed formula $\bar\varphi_{500} = \exists x_1 \ldots \exists x_{11}\varphi_{500}$ states the existence of a suitable real solution. In a first step, we solve for $i \in \{1, \ldots, 11\}$ the following eleven QE problems using VS:

$$\varphi_{500}^{(i)} = \mathrm{VS}(\exists x_1 \ldots \exists x_{i-1}\exists x_{i+1} \ldots \exists x_{11}\varphi_{500}).$$

Each $\varphi_{500}^{(i)}$ is a univariate quantifier-free formula describing all possible real choices for x_i for which there exist real choices for all other variables such that φ_{500} holds. CAD can easily decompose the corresponding one-dimensional spaces. It happens that for each x_i there are exactly three zero-dimensional cells $a_i, b_i, c_i \in \mathbb{R}$ where $\varphi_{500}^{(i)}$ holds. We extract all a_i, b_i, and c_i as *real algebraic numbers*, i.e., as the unique root of a univariate defining polynomials with integer coefficients within an isolating interval. By combinatorial arguments it is not hard to see that the following holds for the set S_{500} of real solutions of φ_{500}:

$$3 \leq |S_{500}| \quad \text{and} \quad S_{500} \subseteq \prod_{i=1}^{11}\{a_i, b_i, c_i\}.$$

Notice that at this point we have proven the existence of multiple fixed points of the system for $k_{19} = 500$. We can furthermore compute S_{500} by plugging the 3^{11} candidates from the Cartesian product into φ_{500}. A straightforward approach requires arithmetic with real algebraic numbers followed by the determination of the signs of the results, which is quite inefficient in practice. However, it turns out that interval arithmetic starting with refinements of the isolating intervals of the real algebraic numbers excludes $3^{11} - 3$ of the candidate solutions. Even the three remaining candidates then require no further checking with algebraic numbers since we already know that $|S_{500}| \geq 3$. The overall CPU time is 71.3 seconds for 11 runs of VS plus 11 runs of CAD, followed by 16 hours for checking candidates. Our checking procedure is a file-based prototype starting a Reduce process for every single of the 3^{11} candidates; there is considerable room for optimization.

For $k_{19} = 200$ instead of 500 all eleven univariate CAD computations yield unique solutions which can be straightforwardly combined to one unique solution for the corresponding φ_{200}. The overall CPU time here is 66.4 seconds for 11 runs of VS plus 11 runs of CAD. Machine float approximations of all our solutions are given in Table 1.

3.3. Parametric Analysis for k_{19}

We next consider the case where k_{19} is left as a free parameter:

$$\varphi_{k_{19}} = \varphi\sigma[100/k_{17}, 50/k_{18}]. \tag{7}$$

Again, we solve for $i \in \{1, \ldots, 11\}$ eleven QE problems using VS:

$$\varphi_{k_{19}}^{(i)} = \mathrm{VS}(\exists x_1 \ldots \exists x_{i-1}\exists x_{i+1} \ldots \exists x_{11}\varphi_{k_{19}}).$$

This time each $\varphi_{k_{19}}^{(i)}$ is a bivariate quantifier-free formula in k_{19} and the corresponding x_i. Hence we must now construct a two-dimensional CAD for each $\varphi_{k_{19}}^{(i)}$. The projection order is important:

9

Table 1: The unique solution $x^{(200)}$ for $k_{19} = 200$ and the three solutions $x_1^{(500)}$, $x_2^{(500)}$, $x_3^{(500)}$ for $k_{19} = 500$. Note that we have actually computed real algebraic numbers, which are pairs of univariate polynomials and isolated intervals. For convenience we are giving machine float approximations here, which can be made arbitrarily precise.

	$x^{(200)}$	$x_1^{(500)}$	$x_2^{(500)}$	$x_3^{(500)}$
x_1	90.6512	17.6392	122.034	323.761
x_2	2.67311	6.97675	14.6721	9.49621
x_3	10.4996	367.57	234.974	37.1013
x_4	17.8545	36.6772	14.5102	6.72938
x_5	35.9695	5.50874	7.16952	13.6295
x_6	32.0501	12.811	35.064	43.1428
x_7	0.0954536	0.511775	0.42579	0.127807
x_8	15.5631	83.4416	69.4223	20.8381
x_9	2.39331	8.06095	7.43877	3.21139
x_{10}	0.641001	0.25622	0.70128	0.862856
x_{11}	45.4331	2.73253	15.2681	61.4581

we first project x_i, then the CAD base phase decomposes the k_{19}-axis, followed by an extension phase that decomposes the x_i-space over the k_{19}-cells obtained in the base phase. This is feasible if we make one limitation: not to extend over zero-dimensional k_{19}-cells. In other words, we accept finitely many blind spots in parameter space, which we can explicitly read off from the CAD so that in the end we know exactly what we are missing.

Figure 1 shows our CAD tree for $\varphi_{k_{19}}^{(2)}$. The first layer from the root shows the decomposition of the k_{19}-axis. The five zero-dimensional (rectangular) cells are the previously mentioned blind spots, among which the smallest one is not relevant, as it has negative value of k_{19}. Those zero-dimensional cells also establish the limits of the full dimensional (oval) cells in between. The cylinders over those one-dimensional k_{19}-cells each contain either one or three zero-dimensional x_2-cells where $\varphi_{k_{19}}^{(2)}$ holds. We have deleted from the tree all x_2-cells where $\varphi_{k_{19}}^{(2)}$ does not hold.

We make two observations, important for a qualitative analysis of our system:

(i) For all positive choices of k_{19}, extending to infinity, there is at least one positive solution for x_2.

(ii) There is a break point around $k_{19} = 409.253$ where the system changes from having a unique solution to exactly three solutions.

Recall that for all floating point numbers given here as approximations we in fact know exact real algebraic numbers. For instance, the exact break point is the only real zero in the open interval $(409, 410)$ of an irreducible defining polynomial

$$\sum_{i=0}^{10} c_i k_{19}^i \text{ with integer coefficients } c_i \text{ as in Appendix A.} \tag{8}$$

Figure 2 depicts all eleven CAD trees for $\psi_{k_{19}}^{(1)}, \ldots, \psi_{k_{19}}^{(11)}$. They are quite similar to the one just discussed. Even the break point from one to three solutions for x_i is identical for all $i \in \{1, \ldots, 11\}$ so that we can generalize our observations from earlier:
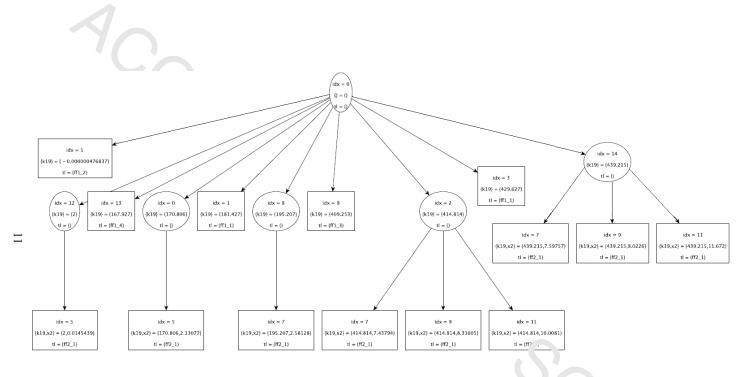
10

11

Figure 1: The pruned CAD tree for x_2. Ellipses and rectangles are full-dimensional and zero-dimensional cells, respectively. We have removed cells where k_{19} is negative or where the input formula is false.

(i) For all positive choices of k_{19}, extending to infinity, there is at least one positive solution for $(x_1, \ldots, x_{11})$.

(ii) There is a break point β around $k_{19} = 409.253$ where the system changes its qualitative behaviour. We have exactly given β as a real algebraic number in Equation (8). For $k_{19} < \beta$ there is exactly one positive solution for $(x_1, \ldots, x_{11})$. For $k_{19} > \beta$ there are at least 3 and at most 3^{11} positive solutions for $(x_1, \ldots, x_{11})$.

The overall computation time for our parametric analysis is 4.3 minutes. It is strongly dominated by 2.8 minutes for the computation of one particular CAD tree, for $\varphi_{k_{19}}^{(11)}$. It turns out that the suitable projection order with x_i eliminated first is computationally considerably harder than projecting the other way round. As a preprocessing step we apply CAD-based simplification of the $\varphi_{k_{19}}^{(i)}$ with the opposite, faster, projection order. Here we use QEPCAD-B (v1.69), which performs better than Redlog at simple solution formula construction (Brown, 2003).

4. Using Triangular Decomposition Tools in the Regular Chains Library for Maple

In this section we are going to apply triangular decomposition methods, including CAD. We find that a triangular decomposition can derive solution formulae for many variables in terms of a smaller subset for which we must apply CAD to count solutions. Throughout this section we are performing computations in Maple 2016, but using an updated version of the Regular Chains Library[3]. Timings are reported for a Windows 7 64 bit Desktop PC with Intel i5.

4.1. Parametric Analysis for k_{19}

Regular chains are the triangular decompositions of systems of polynomial equations, where triangular means decreasing subsets of variables occurring in each polynomial. Highly efficient methods for working in complex space have been developed based on these; see the work of Wang (2000) and Aubry et al. (1999) for a survey.

Recent work by Chen et al. (2013) proposes adaptations of these tools to the real analogue: semi-algebraic systems. They describe two algorithms to decompose any real polynomial system into finitely many regular semi-algebraic systems. The first, *Real Triangularize* (RT), does so directly while the second, *Lazy Real Triangularize* (LRT), produces the highest (complex) dimension solution component and unevaluated function calls, which if all evaluated would combine to give the full solution. These algorithms are implemented in the Regular Chains Library for Maple.

We will apply LRT on the quantifier-free formula (5) evaluated with the parameter estimates for $k_1, \ldots, k_{18}$, i.e. the system (7) as studied with Redlog in Section 3.3.

We need to choose a variable ordering: our analysis requires that k_{19} be the indeterminate considered alone. We place the remaining variables in lexicographical order since the in-built heuristics to make the choice could suggest nothing better. The solutions must hence contain constraints in k_{19}, constraints in (x_1, k_{19}), in (x_2, x_1, k_{19}) and so on.

Applying LRT this way produces one solution component and 6 unevaluated function calls in around 15 seconds.

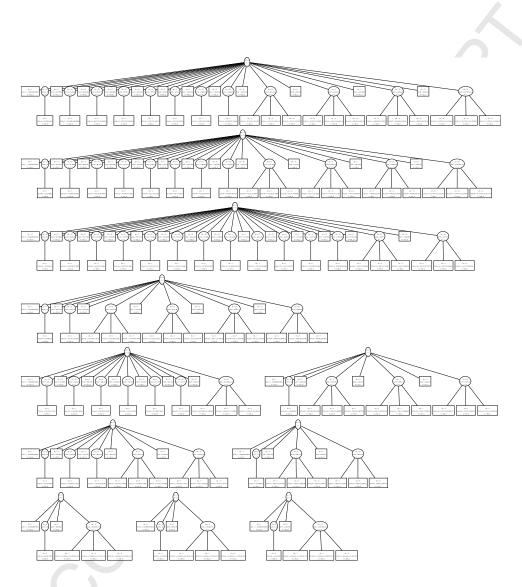[3] www.regularchains.org

Figure 2: All CAD trees for $\psi_{k_{19}}^{(1)}, \ldots, \psi_{k_{19}}^{(11)}$. In the second but last row on the left hand side there is the tree for $\psi_{k_{19}}^{(1)}$, which is displayed in detail in Figure 1. Note that in the digital version of this article readers can *zoom into* these trees to see the details (as are visible in the printed version of Figure 1).

13

4.1.1. The main solution component from LRT

In the evaluated component: for each of $x_2, \ldots, x_{11}$ there is a single equation which has this as the main variable. Further, these are all linear in their main variable meaning they can be easily rearranged into the solution formulae given below.

$$x_{11} = -\frac{1}{60}x_2^2 + \frac{1}{600}(10k_{19} - 10x_1 - 37x_3 + 10x_4 - 2100)x_2 - \frac{9}{200}x_3^2$$
$$+ \frac{1}{600}(-27x_1 + 27x_4 + 27k_{19} - 4650)x_3 - x_1 + x_4 + k_{19} - 50 \tag{9}$$

$$x_{10} = \frac{1}{150}x_2(x_2 + x_3 - x_4 - k_{19} + x_1 + 150) \tag{10}$$

$$x_9 = \frac{1}{18200}(69x_3 + 182x_2)(x_2 + x_3 - x_4 - k_{19} + x_1 + 150) \tag{11}$$

$$x_8 = \frac{15}{364}(x_2 + x_3 - x_4 - k_{19} + x_1 + 150)x_3 \tag{12}$$

$$x_7 = 50 - \frac{2}{101}x_4 x_1 - x_4 \tag{13}$$

$$x_6 = \frac{2}{101}x_4 x_1 \tag{14}$$

$$x_5 = x_2 + x_3 - x_4 - k_{19} + x_1 + 150 \tag{15}$$

$$x_4 = 2525000/(101x_2 + 1000x_1 + 50500) \tag{16}$$

$$x_3 = n_3/d_3 \quad \text{where} \tag{17}$$

$$n_3 = -101x_2^3 - (-101k_{19} + 1101x_1 + 65650)x_2^2 - (1000x_1^2$$
$$+ (-1000k_{19} + 200500)x_1 - 50500k_{19} + 5050000)x_2 + 150000x_1)$$

$$d_3 = 101x_2^2 + (1000x_1 + 50500)x_2$$

$$x_2 = n_2/d_2 \quad \text{where} \tag{18}$$

$$n_2 = 306258330647900095489914199920x_1^5$$
$$+ (-43795148662369306906962603840k_{19}$$
$$+ 377499792254877318052736865046632200)x_1^4$$
$$+ (148712106477824620536932359920k_{19}^2$$
$$- 1696333629369275091915491069067240 0k_{19}$$
$$+ 6815925407229297763234036009365120000)x_1^3$$
$$+ (15383254482229832299305300492000 00k_{19}^2$$
$$- 86270216410420829103135799600002000 0k_{19}$$
$$+ 279241219028720368578809336249748000000)x_1^2$$
$$+ (293703416949546481010850990000 00k_{19}^2$$
$$- 12995812279808313524592161760000000 0k_{19}$$
$$+ 370596028211752324288676921370000000 0000)x_1$$
$$- 12623587451027839577736900000000 00000000k_{19}$$

$$d_2 = 2327636637521132379740294044200 89x_1^5$$

14

$$+ (-332853615301041845577671639990228k_{19}$$
$$+ 886463032152050753763081470296677220)x_1^4$$
$$+ (113024761399450186949390623074789k_{19}^2$$
$$- 808439080283314981399545277761762740k_{19}$$
$$+ 1168246506839176979663298692907276500)x_1^3$$
$$+ (1145523230964903430559704879147902 0k_{19}^2$$
$$- 554725102606043356664062052802387700 0k_{19}$$
$$+ 619147207587597001268026254404647600000)x_1^2$$
$$+ (29024599706300155013019802645852500 0k_{19}^2$$
$$- 1413482867583527623234895486743985000 00k_{19}$$
$$+ 145472885295813822525870715414946000 00000)x_1$$
$$- 124749850181857994662675693177500000 0000(k_{19} - 100)$$

Note that these solution formula: are guaranteed valid for all positive k_{19} excluding three isolated points which are provided as part of the output from LRT and described below; are triangular, with each x_k is expressed in variables $\{x_i, i < k\}$; and are provided for all but variable x_1.

The output of LRT also requires that x_1 be both positive and satisfy:

$$f(x_1, k_{19}) = \sum_{i=0}^{6} d_i x_1^i = 0 \tag{19}$$

where the coefficients d_i are univariate polynomials in k_{19} of maximum degree 2 as given in Appendix B. Hence there are at most six solutions for x_1, with the exact number depending on whether solutions of (19) are real and positive.

There are four constraints on free parameter k_{19} as given below, one of which is the non-vanishing of the polynomial in Appendix Appendix A whose root defined the break point found by Redlog in Section 3.3. Note that the coefficients break over lines within the final constraint.

$$k_{19} > 0 \tag{20}$$
$$\wedge \text{ polynomial in (8)} \neq 0 \tag{21}$$

$$\wedge\ 23197989433419579994929k_{19}^2 - 894074006154524094530988 00k_{19}$$
$$- 482241930341916652549114919 0000 \neq 0 \tag{22}$$

$$\wedge\ 505465566622475867655547880786544637953790406059982726185509k_{19}^4$$
$$-\ 1272578045696439189317856051518387368422217896986836692050$$
$$5134120k_{19}^3 + 1175510330915205241831243213231417517003037315 56$$
$$2884193657451445400k_{19}^2 - 2818673598836761598111920829785411 93$$
$$6002928043245969118783379725600 00k_{19} - 42434363570215587465$$
$$66842370156393218505106689274120793187930720000 0000 \neq 0 \tag{23}$$

Evaluating the real roots of the polynomials appearing in the above allows us to conclude that this solution component is valid for all positive values of k_{19} excluding three points. As with

Redlog, Maple can represent these as exact algebraic numbers but for brevity we give float approximations:

$$409.253, 16473.337, \text{ and } 25084.536. \tag{24}$$

Software Remark 1. In the authors' ISSAC 2017 paper (Bradford et al., 2017) the description of the evaluated solution component ended here. However, following the publication of that paper a bug was uncovered by one of the authors in the simplifier of the Regular Chains Library when working with a different MAPK model to the one considered presently. For that example the simplifier was incorrectly discarding certain positivity conditions. The bug was reported to the Regular Chains developers, and the current version of the simplifier[4] now excludes all such simplifications. So presently, the output from LRT includes also the positivity conditions

$$x_2 > 0, x_3 > 0, \ldots, x_{11} > 0.$$

Some of these can clearly be removed. For example, if we know $x_1 > 0$ and $x_2 > 0$ then (16) implies $x_4 > 0$ and this coupled with (14) implies $x_6 > 0$. However, it is not trivial to imply all such inequalities, and so any proposed solution in (k_{19}, x_1) should be checked to see if it implies a positive solution in all the remaining variables before being accepted. This is indeed the case for all solutions described in the ISSAC 2017 paper, and below.

4.1.2. The unevaluated function calls from LRT

The main solution component described in Section 4.1.1 is not the entire solution to the system. LRT produced also six unevaluated function calls which if evaluated and combined with the main component would give the full solution. LRT guarantees that the complex dimension of the solution components from these unevaluated calls is smaller that the main component. In fact, three of the six unevaluated calls define empty solution sets, with evaluating to discover this instantaneous.

With regards to the other three: we can infer from the arguments to these function calls that each defines the solution at one of the three points in (24) that were excluded from the main component. I.e. each of these three calls has as an argument the negation of one of the univariate inequations for k_{19} from (21)−(23). Actually evaluating these solution components is not possible in reasonable time. Thus, as with Redlog in Section 3, we proceed accepting a small number of blind spots.

The output of LRT has quickly given us the structure of the solution space valid at all but three isolated values of k_{19}. However, it does not identify where the number of real solutions change. Note that although the break point identified in Section 3 has been rediscovered in (24), there is not yet any information gathered by Maple from which we can infer its significance. We also note that there seems to be no significance for our application of the other two isolated points in (24).

4.1.3. Counting solutions with CAD

To finish the analysis we need to decompose (x_1, k_{19})-space according to the real roots of $f(x_1, k_{19})$; and also x_1 and k_{19} since the constraints $x_1 > 0$ and $k_{19} > 0$ were specified separately in the output. CAD is ideally suited for this task. We apply the Regular Chains based implementation in Maple first described by Chen et al. (2009). A CAD for $f(x_1, k_{19})$, with the

---

[4]http://www.arcnl.org/cchen/software.html

ordering chosen so that the k_{19}-axis is the one decomposed, divides the plane into 135 cells in a few seconds. This CAD decomposes the k_{19} axis into 11 cells, i.e. identifying five points, which approximate to:

$$-379.993, \ -87.776, \ 0, \ 409.253, \ \text{and} \ 25084.536.$$

We give these as floats for brevity but exact algebraic numbers are available[5].

On the cell where $0 < k_{19} < 409.253$, the cylinder above in the (x_1, k_{19})-plane is divided into 11 cells: three of which cover $x_1 > 0$ (two 2d sectors and a 1d section). We see that $f(x_1, k_{19})$ is zero on the section but not the sectors. This can be inferred by testing a sample point of the section (the invariance properties of the CAD mean that the signs of the input at this point are representative for the whole cell. In fact, with the CAD implementation we use the cells comes with a semi-algebraic description which for this section is the statement that $f(x_1, k_{19}) = 0$ (along with the bounds on k_{19}).

We can perform a similar analysis on the two cells for $409.253 < k_{19} < 25084.536$ and $25084.536 < k_{19} < \infty$. In each case the cylinders above are divided into 15 cells, seven of which cover $x_1 > 0$, with the three sections satisfying $f(x_1, k_{19}) = 0$.

So we can conclude that: (a) if $0 < k_{19} < 409.253$ then $f(x_1, k_{19})$ has a single positive real solution; and (b) if $k_{19} \in (409.253, \infty) \setminus \{25084.536\}$ then $f(x_1, k_{19})$ has three positive real solutions. We cannot conclude with certainty what happens at the points 409.253 and 25084.536.

At the end of this analysis we have rediscovered the break point identified in Section 3 where the system moves from a single positive real solution to three. We also have explicit solutions valid for all except three isolated k_{19} values. To obtain an actual numerical solution we need only: select the k_{19} value of interest (call it $\hat{k}_{19}$); perform univariate root isolation on $f(x_1, \hat{k}_{19})$, noting we know in advance how many to expect based on $\hat{k}_{19}$; then for each x_1 solution substitute recursively into equations (9)−(18), starting with (18) and working up, substituting the new variable solution from each formula into the next. The solutions in Table 1 may be easily rediscovered this way, for example.

We note that, as discussed in Software Remark 1, we have ensured that for each cell all the positive solutions in x_1 provided by the sample point do indeed lead to positive solutions for all other variables via the back substitution process.

4.2. Repeating for Other Choices

We have repeated the approach described in Section 4.1 for different choices of free parameter and different choices of fixed parameter values. For example:

- With k_{17} set to 95 instead of 100 we find that the break point between 1 and 3 real positive solutions moves to $k_{19} = 369.917$. With k_{17} set to 105 it moves to $k_{19} = 450.077$.

- Allowing k_{17} to be free and fixing $k_{19} = 200$ we find that there is only ever one positive real solution.

- Allowing k_{17} to be free and fixing $k_{19} = 500$ we find the number of positive real solutions moving from 1 to 3 to 1 breaking at $k_{17} = 85.988$ and $k_{17} = 110.869$.

- Similarly, allowing k_{18} to be free and fixing $k_{19} = 200$ we find there is only ever one positive real solution; but fixing $k_{19} = 500$ instead we find 3 real solutions between $k_{18} = 44.434$ and 58.329 and 1 otherwise.

---

[5]See the Research Data Statement at the end of the paper to access them.

This hints that there is a shape approximating a paraboloid within (k_{17}, k_{18}, k_{19})-space within which bistability may occur; with bistability available for any k_{17} and k_{18} value but bounded from below in the k_{19} coordinate.

We note that these conclusions are, as with the one described in detail, valid at all but a handful of isolated values of the free parameter.

5. A Graph Theory Guided Parametric Gaussian Elimination Preprocessing Method

As described above, the complexity of polynomial systems obtained with steady-state approximations of biological models is comparatively high for the application of symbolic methods, particularly in reference to the dimension (number of indeterminates). The two studies described in Sections 3 and 4 both used tools to effectively reduce the problem dimension before applying the costly CAD method.

More generally, it is highly relevant for the the success of general polynomial systems methods if we can first identify and exploit particular structural properties of the input. Here, the MAPK models have remarkably low total degrees with many linear monomials after some substitutions for rate constants. For example, the final equation of (1) suggests a simple polynomial expression for x_{11} in terms of the remaining variables of the system. This promoted the idea of pre-processing MAPK input with essentially Gaussian elimination: in the sense of solving single suitable equations with respect to some variable and substituting the corresponding solution into the system.

5.1. Parametric Gaussian Elimination

Generalizing this idea to situations where linear variables have parametric coefficients in the other variables requires, in general, a parametric variant of Gaussian elimination, which replaces the input system with a finite case distinction with respect to the vanishing of certain coefficients and one reduced system for each case. Further, for our problem the positivity conditions establish a further apparent obstacle, because we are formally not dealing with a parametric system of linear equations but with a parametric linear programming problem.

The theory of real quantifier elimination by virtual substitution tells us that it is sufficient for the inequality constraints to play a passive role in the sense that their polynomials do not contribute to the elimination set E discussed in Section 3.1. This key idea occurred first for the linear case in Theorem 3.11 of the work by Loos and Weispfenning (1993); while the current state-of-the-art is described in the thesis of Košta (2016). The crucial observation is that our entire formula is (and remains during the considered elimination) a single Gauss Prime Constituent in the sense of (Košta, 2016, Section 3.1.1). Further, for the considered MAPK model, it turns out that those positivity assumptions on the variables are actually strong enough to guarantee the non-vanishing of all relevant coefficients, so case-distinctions are never necessary! We do not claim such an approach will always be so lucky, but it may be this result generalises for the MAPK hierarchy. It was the case also for the second larger MAPK model we describe in Section 8.

5.2. An Optimal Strategy

Parametric Gaussian elimination can increase the degrees of variables in the parametric coefficient, in particular destroying their linearity and suitability to be used for further reductions.
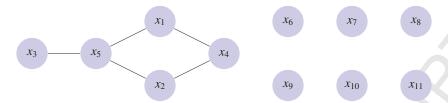
18

Figure 3: The graph for $\overline{(1)}$ is loosely connected. Its minimum vertex cover $\{x_4, x_5\}$ is small. All other variables form a maximum independent set, which can be eliminated with linear methods.

For example, solving the last equation of $\overline{(1)}$ and substituting into the first equation would destroy any linearity present in that first equation.

The natural question is whether there is an optimal strategy to Gauss-eliminate a maximal number of variables? This has been answered positively only recently by Grigoriev et al. (2015): draw a graph, where vertices are variables and edges indicate multiplication between variables within some monomial. Then one can Gauss-eliminate a *maximum independent set*, which is the complement of a *minimum vertex cover*. Figure 3 shows that graph for $\overline{(1)}$, where $\{x_4, x_5\}$ is a minimal vertex cover, and all other variables can be linearly eliminated.

Recall that minimum vertex cover is one of 21 classical NP-complete problems described by Karp (1972). However, our instances considered here and instances to be expected from other biological models are so small that the use of existing approximation algorithms (Grandoni et al., 2008) appears unnecessary. We have used real quantifier elimination, which did not consume measurable CPU time; alternatively one could use integer linear programming or SAT-solving.

It is a most remarkable fact that a significant number of biological models in the databases have that property of loosely connected variables. This phenomenon resembles the well-known *community structure* of propositional satisfiability problems, which has been identified as one of the key structural reasons for the impressive success of state-of-the-art CDCL-based SAT solvers by Girvan and Newman (2002).

5.3. Reduced System for Model 26

We conclude this section with the reduced system computed with an implementation of this pre-processing in Redlog (Dolzmann and Sturm, 1997a). From (6) we obtain

$$
\begin{aligned}
\overline{\psi} = \; & x_5 > 0 \,\wedge\, x_4 > 0 \,\wedge\, k_{19} > 0 \,\wedge\, k_{18} > 0 \,\wedge\, k_{17} > 0 \\
& \wedge\, 1062444 k_{18} x_4^2 x_5 + 23478000 k_{18} x_4^2 + 1153450 k_{18} x_4 x_5^2 + 2967000 k_{18} x_4 x_5 \\
& \quad + 638825 k_{18} x_5^3 + 49944500 k_{18} x_5^2 - 5934 k_{19} x_4^2 x_5 - 989000 k_{19} x_4 x_5^2 \\
& \quad - 1062444 x_4^3 x_5 - 23478000 x_4^3 - 1153450 x_4^2 x_5^2 - 2967000 x_4^2 x_5 \\
& \quad - 638825 x_4 x_5^3 - 49944500 x_4 x_5^2 = 0 \\
& \wedge\, 1062444 k_{17} x_4^2 x_5 + 23478000 k_{17} x_4^2 + 1153450 k_{17} x_4 x_5^2 + 2967000 k_{17} x_4 x_5 \\
& \quad + 638825 k_{17} x_5^3 + 49944500 k_{17} x_5^2 - 1056510 k_{19} x_4^2 x_5 - 164450 k_{19} x_4 x_5^2 \\
& \quad - 638825 k_{19} x_5^3 - 1062444 x_4^2 x_5^2 - 23478000 x_4^2 x_5 - 1153450 x_4 x_5^3 \\
& \quad - 2967000 x_4 x_5^2 - 638825 x_5^4 - 49944500 x_5^3 = 0.
\end{aligned}
\tag{25}
$$

19

We now have a system of just two equalities in 5 indeterminates together with positivity conditions on those indeterminates. Notice that no complicated positivity constraints come into existence from this method. All corresponding substitution results are entailed by the other constraints, which is implicitly discovered by using the standard simplifier of Dolzmann and Sturm (1997b) during preprocessing.

Note that, with ψ defined in (6), we have a formal equivalence here, from the theory of quantifier elimination via virtual substitution:

$$\exists x_1 \exists x_2 \ldots \exists x_{11} \, \psi = \exists x_4 \exists x_5 \, \overline{\psi}.$$

So if we can determine the region of parameter space where solutions to $\overline{\psi}$ exist we are guaranteed to also find solutions to ψ there. However, our problem concerns not just the existence of solutions but the number, and so on the surface this may seems inadequate. However, because the only technology used in this reduction is linear substitution we can also conclude that the number of solutions found for $\overline{\psi}$ will lead to the same number of solution of ψ.

Hence it is sufficient to study $\overline{\psi}$. This pre-processing allows us to derive solutions with two free-parameters in the next section. We also give some indication of the performance improvements of various methods offered by the pre-processing later in Section 9.

6. Combined Approach for a Solution over 2-parameter space

In this section we describe a new derivation of a solution to the real algebraic problem with two free parameters, produced after the publication of the authors' ISSAC 2017 and CASC 2017 conference papers (Bradford et al., 2017; England et al., 2017). The progress is made by combining ideas from all three of the preceding sections. We describe in detail below but broadly we: start with the reduced system from the pre-processing of Section 5 with two free-parameters; apply the LRT method of Section 4 to reduce the problem by an indeterminate; build part of a CAD, an idea used in Section 3, sufficient to identify the regions of parameter space of interest. Timings are reported for the same hardware and software as Section 4.

6.1. Applying LRT and Preparing for CAD

We start with the reduced system (25) derived in Section 5 above. We set k_{18} to 50 and leave k_{17} and k_{19} free. Hence we seek the regions of the (k_{17}, k_{19})-plane where there exist multiple solutions. We first run the LRT algorithm introduced in Section 4, with variable ordering $(x_4, x_5, k_{17}, k_{19})$. We needed the parameters to come after the variables so we work over the parameter space, but within the pairs the orders could have been reversed. In around 5 seconds LRT outputs one solution component and 4 unevaluated function calls.

The evaluated component consists of the four positivity conditions from the input and the two equations, which may be seen in Appendix C where they are labelled (C.1) and (C.2). Of course these equations are triangular: (C.1) involves $\{x_4, x_5, k_{17}, k_{19}\}$ while (C.2) does not depend on x_4. Note that (C.1) is linear in x_4 and so we can easily rearrange to give a solution formula for x_4 in terms of (x_5, k_{17}, k_{19}). (C.2) is of degree 6 in x_5 but of course not all its solutions need be real and positive. If we can determine where (C.2) has multiple positive real solutions then all that remains is to back substitute and to get real solutions for the other variables and check these are also positive. We will determine this using CAD.

Before that, we examine the 4 unevaluated functions calls from LRT: two instantly evaluate to empty solution sets while the other two cannot be evaluated in reasonable time. We infer

from the arguments to the calls that these define solutions on the graphs of two polynomials in (k_{17}, k_{19})-space:

$$p_1 = \sum_{i=0}^{5} e_{1,i} \, k_{17}^i, \qquad\qquad\qquad (26)$$

$$p_2 = \sum_{i=0}^{14} e_{2,i} \, k_{17}^i \qquad\qquad\qquad (27)$$

where the $e_{1,i}$ and $e_{2,i}$ are univariate polynomials in k_{19}. The polynomials are too large to reproduce here[6]. The smaller one is (26): degree 4 in k_{19}, total degree 5, 20 terms when expanded with average coefficient length of 51 digits. The larger (26) has degree 10 in k_{19}, total degree 14, 110 terms when expanded with average coefficient length of 79 digits[7].

We proceed on the understanding that any results are valid everywhere in (k_{17}, k_{19})-space except on these graphs. We may compare this to Sections 3.3 and 4.1 which accepted a finite number of isolated blind spots in a one-dimensional parameter space.

6.2. Solution via an Open CAD

A CAD sign-invariant for the polynomial defining (*C*.2) (and x_5, k_{17}, k_{19} to allow for positivity checks) would be sufficient. However, the size of the polynomial puts this beyond CAD currently. Instead, we proceed as follows:

Step 1: Calculate the projection set for CAD input consisting of polynomial defining (C.2) and polynomial x_5 (to allow for positivity check).

This is a set of 19 polynomials in (k_{17}, k_{19}) the greatest of which has degree 34, and so it is not reasonable to print them all here.

Step 2: Build an Open CAD of (k_{17}, k_{19})-space for these polynomials, along with polynomials k_{17} and k_{19} (to allow for positivity checks).

An *Open CAD* means the full dimensional cells only. The boundaries may be determined by algebraic numbers but because we do not lift over the boundaries there no costly algebraic number calculations. The idea has been much discussed by McCallum (1993); Strzeboński (2000); Wilson et al. (2014), and other names used for it include generic CAD and 1-layered Sub-CAD. It was partly applied by the approach in Redlog in Section 3. It is sufficient to solve problems which are only in strict inequalities, but of course, that is not the case here. By making this restriction we are accepting that our solutions and conclusions are not necessarily valid on cell boundaries: a finite number of curve segments in the (k_{17}, k_{19})-plane. However, we have already made such an acceptance, in the use of LRT above.

We perform the above steps with the ProjectionCAD package of England et al. (2014) in Maple[8] in 17 seconds. The resulting CAD has 533 cells.

Step 3: Identify those cells in the upper quadrant of the (k_{17}, k_{19})-plane.

---

[6]The polynomials may be found online in a machine readable form − see the Research Data Statement at the end of the paper.

[7]But as described later in Section 10.3 the larger polynomial is the one of interest: the boundary of the multistationarity region is actually defined by part of the graph of (27), although there is no reason to conclude that at this stage of the analysis.

[8]http://computing.coventry.ac.uk/~mengland/ProjectionCAD.html

We only care about solutions in this upper quadrant. We can easily identify 139 such cells by querying sample points (note that no cell can straddle the boundary of the quadrant since the CAD produced was also produced sign-invariant for k_{17} and k_{19} as polynomials). Since in Step 1 we ensured that this CAD was built for the projection of the polynomial defining (C.2) we may conclude that for this polynomial we can work at a sample point of the cell but draw conclusions for the whole cell, as we do next.

Step 4: Identify the number of positive real roots the polynomial defining (C.2) has over each of these cells.

We do this by substituting for the sample point and applying Maple's default real root isolation algorithm. We identify 35 of the 139 cells where there are three positive real roots for x_5, with the other 104 all having one.

Step 5: Check that these solutions provide a positive solution for x_4 via back substitution into (C.1).

We first checked that the 104 cells with one positive real solution for x_5 all lead to one positive real solution for x_4 as expected. We then analyse the 35 cells and each of their three positive real solutions for x_5 in turn. For 28 of these cells each solution gives a corresponding positive real solution for x_4. For the other 7 cells, only one of the three solutions does, so these join the other 104 as representing the parameter space with one solution.

The semi-algebraic descriptions of these 28 cells provide the exact description of the regions in (k_{17}, k_{19})-space where multistationarity can occur. We use these descriptions to produce the 4 plots of the multistationarity region in Figures 4 and 5. The 4 images are all produced from the data in the 28 cells, but with different plotting regions. In each case, the coloured regions represent the cells with multistationarity, with the only purpose of the different colours to show the separation of the cells[9].

The left plot in Figure 4 is for the original range of k_{19} values considered and has the region of multistationarity described by 4 full dimensional CAD cells. The right plot shows that this region grows as k_{19} increases: at this range 9 cells are in view including the 4 from the left plot which are at the bottom of the region.

The left plot of Figure 5 expands the ranges considerably. There are 24 cells in view of the range but the original 9 described above are now too small to see. The right plot of Figure 5 expands the range further to include all 28 cells; with all 24 from the previous image now too small to see. In this final image the two cells at the top actually extend infinitely in the k_{19} direction while always being bounded on both sides in the k_{17} direction.

7. Stability of Fixed Points

The work described in Section 3−6 was dedicated to identifying where multiple fixed points occur. This alone does not prove multistationarity as we must also check the stability properties of these fixed points.

We may use the three linear conservation constraint equations (3) to eliminate x_1, x_7, and x_{11} from system (1) and symbolically compute the Jacobian $\tilde{J}$ of the obtained reduced system. We

---

[9]Because we produced an Open CAD above we cannot formally conclude what happens on these cell boundaries.
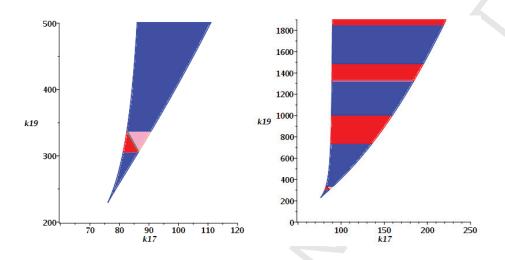
Figure 4: Visualisations of the Open CAD cells describing the multistationarity region derived in Section 6 for smaller values of k_{17} and k_{19}.
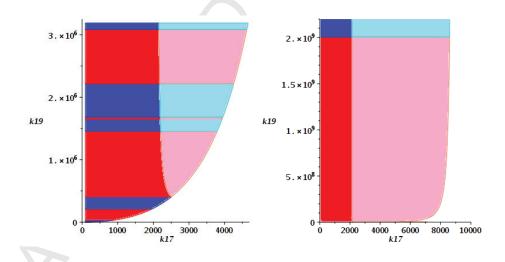

Figure 5: Visualisations of the Open CAD cells describing the multistationarity region derived in Section 6 for larger values of k_{17} and k_{19}.

23

can then numerically compute the eigenvalues of $\tilde{J}$ for the instances arising from the substitution of the parameter values and the different positive fixed points for the variables.

We have used the float approximations for the unique solution $x^{(200)}$ with $k_{19} = 200$ and the three solutions $x_1^{(500)}, \ldots, x_3^{(500)}$ for $k_{19} = 500$ in Table 1. For the single positive fixed point $x^{(200)}$ the Jacobian $\tilde{J}(x^{(200)})$ has eigenvalues with negative real part only and hence can be shown to be stable. For $k_{19} = 500$ one of the three positive fixed points $x_2^{(500)}$ can be shown to be unstable, as $\tilde{J}(x_2^{(500)})$ has one eigenvalue with positive real part; the other seven had negative real parts. In contrast $x_1^{(500)}$ and $x_3^{(500)}$ can be shown to be stable. Hence for $k_{19} = 500$ the system is indeed bistable.

A verification of the stability of the fixed points using exact real algebraic numbers by the well-known Routh–Hurwitz criterion is possible algorithmically (Hong et al., 1997), but seems to be out of range of current methods for this example. Notice that in other studies on multistationarity of signaling pathways, such as those of Conradi et al. (2008) and Gross et al. (2016b), the question of stability has also been left to one side.

8. Another MAPK Model

We describe a second MAPK model, which we will use alongside the first from Section 2 in the remaining sections, to broaden the conclusions drawn.

8.1. MAPK Bio-Model 28

The system with number 28 in the BioModels Database is given by the following set of differential equations. This model is the distributive fully random kinetics version of the models proposed by Markevich et al. (2004). Hereafter we refer to it as Model 28. Again, we have renamed the species to $x_1, \ldots, x_{16}$ and the rate constants to $k_1, \ldots, k_{27}$ to facilitate reading:

$$
\begin{aligned}
\dot{x}_1 &= k_2 x_9 + k_8 x_{10} + k_{21} x_{15} + k_{26} x_{16} \\
&\quad -k_1 x_1 x_5 - k_7 x_1 x_5 - k_{22} x_1 x_6 - k_{27} x_1 x_6 \\
\dot{x}_2 &= k_3 x_9 + k_5 x_7 + k_{24} x_{12} - k_4 x_2 x_5 - k_{23} x_2 x_6 \\
\dot{x}_3 &= k_9 x_{10} + k_{11} x_8 + k_{16} x_{13} + k_{19} x_{14} - k_{10} x_3 x_5 - k_{17} x_3 x_6 - k_{18} x_3 x_6 \\
\dot{x}_4 &= k_6 x_7 + k_{12} x_8 + k_{14} x_{11} - k_{13} x_4 x_6 \\
\dot{x}_5 &= k_2 x_9 + k_3 x_9 + k_5 x_7 + k_6 x_7 + k_8 x_{10} + k_9 x_{10} + k_{11} x_8 + k_{12} x_8 - \\
&\quad k_1 x_1 x_5 - k_4 x_2 x_5 - k_7 x_1 x_5 - k_{10} x_3 x_5 \\
\dot{x}_6 &= k_{14} x_{11} + k_{16} x_{13} + k_{19} x_{14} + k_{21} x_{15} + k_{24} x_{12} + k_{26} x_{16} - \\
&\quad k_{13} x_4 x_6 - k_{17} x_3 x_6 - k_{18} x_3 x_6 - k_{22} x_1 x_6 - k_{23} x_2 x_6 - k_{27} x_1 x_6 \\
\dot{x}_7 &= k_4 x_2 x_5 - k_6 x_7 - k_5 x_7 \\
\dot{x}_8 &= k_{10} x_3 x_5 - k_{12} x_8 - k_{11} x_8 \\
\dot{x}_9 &= k_1 x_1 x_5 - k_3 x_9 - k_2 x_9 \\
\dot{x}_{10} &= k_7 x_1 x_5 - k_9 x_{10} - k_8 x_{10} \\
\dot{x}_{11} &= k_{13} x_4 x_6 - k_{15} x_{11} - k_{14} x_{11} \\
\dot{x}_{12} &= k_{23} x_2 x_6 - k_{25} x_{12} - k_{24} x_{12} \\
\dot{x}_{13} &= k_{15} x_{11} - k_{16} x_{13} + k_{17} x_3 x_6
\end{aligned}
$$

24

$$\dot{x}_{14} = k_{18}x_3x_6 - k_{20}x_{14} - k_{19}x_{14}$$
$$\dot{x}_{15} = k_{20}x_{14} - k_{21}x_{15} + k_{22}x_1x_6$$
$$\dot{x}_{16} = k_{25}x_{12} - k_{26}x_{16} + k_{27}x_1x_6 \tag{28}$$

We denote by $\overline{(28)}$ the system formed by replacing all left hand sides of (28) by 0.

The estimates of the rate constants given in the BioModels Database are:

$$
\begin{array}{llll}
k_1 = 0.005, & k_2 = 1, & k_3 = 1.08, & k_4 = 0.025, \\
k_5 = 1, & k_6 = 0.007, & k_7 = 0.05, & k_8 = 1, \\
k_9 = 0.008, & k_{10} = 0.005, & k_{11} = 1, & k_{12} = 0.45, \\
k_{13} = 0.045, & k_{14} = 1, & k_{15} = 0.092, & k_{16} = 1, \\
k_{17} = 0.01, & k_{18} = 0.01, & k_{19} = 1, & k_{20} = 0.5, \\
k_{21} = 0.086, & k_{22} = 0.0011, & k_{23} = 0.01, & k_{24} = 1, \\
k_{25} = 0.47, & k_{26} = 0.14, & k_{27} = 0.0018. & \tag{29}
\end{array}
$$

Again, using the left-null space of the stoichiometric matrix under positive conditions as a conservation constraint (Famili and Palsson, 2003) we obtain the following three linear conservation constraints:

$$
\begin{aligned}
x_6 + x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} &= k_{28}, \\
x_5 + x_7 + x_8 + x_9 + x_{10} &= k_{29}, \\
x_1 + x_2 + x_3 + x_4 + x_7 + x_8 + x_9 + x_{10} + x_{11} + \quad\quad & \\
x_{12} + x_{13} + x_{14} + x_{15} + x_{16} &= k_{30}, \tag{30}
\end{aligned}
$$

where k_{28}, k_{29}, k_{30} are new constants. Meaningful values for these three are harder to obtain than the constants in (2). The following are some realistic value estimates:

$$k_{28} = 100, \qquad k_{29} = 180, \qquad k_{30} = 800. \tag{31}$$

Ideally we would treat all three symbolically and identify multistationarity within (k_{28}, k_{29}, k_{30}) parameter space.

8.2. Preprocessing

We may apply the preprocessing procedure outlined in Section 5 to $\overline{(28)}$ and the positivity constrains similarly to as described in Section 5 for Model 26. The connection graph is given in Figure 6 showing that $\{x_5, x_6\}$ as a minimum vertex cover. We obtain the simplified system:

$$
\begin{aligned}
& 3796549898085k_{29}x_5^3x_6 + 71063292573000k_{29}x_5^3 \\
& +106615407090630k_{29}x_5^2x_6^2 + 479383905861000k_{29}x_5^2x_6 \\
& +299076127852260k_{29}x_5x_6^3 + 3505609439955600k_{29}x_5x_6^2 \\
& +91244417457024k_{29}x_6^4 + 3557586742819200k_{29}x_6^3 \\
& -598701732300k_{30}x_5^3x_6 - 83232870778950k_{30}x_5^2x_6^2 \\
& -185019487578700k_{30}x_5x_6^3 - 3796549898085x_5^4x_6 \\
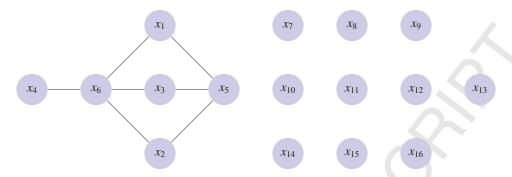& -71063292573000x_5^4 - 106615407090630x_5^3x_6^2
\end{aligned}
$$

25

Figure 6: The graph for $\overline{(28)}$ produced according to the techniques setout in Section 5. Despite being a larger system the minimum vertex cover $\{x_5, x_6\}$ is still small. All other variables form a maximum independent set, which can be eliminated with linear methods.

$$- 479383905861000x_5^3x_6 - 299076127852260x_5^2x_6^3$$
$$-3505609439955600x_5^2x_6^2 - 91244417457024x_5x_6^4$$
$$-3557586742819200x_5x_6^3 = 0,$$

$$3796549898085k_{28}x_5^3x_6 + 71063292573000k_{28}x_5^3$$
$$+106615407090630k_{28}x_5^2x_6^2 + 479383905861000k_{28}x_5^2x_6$$
$$+299076127852260k_{28}x_5x_6^3 + 3505609439955600k_{28}x_5x_6^2$$
$$+91244417457024k_{28}x_6^4 + 3557586742819200k_{28}x_6^3$$
$$-3197848165785k_{30}x_5^3x_6 - 23382536311680k_{30}x_5^2x_6^2$$
$$-114056640273560k_{30}x_5x_6^3 - 91244417457024k_{30}x_6^4$$
$$-3796549898085x_5^3x_6^2 - 71063292573000x_5^3x_6$$
$$- 106615407090630x_5^2x_6^3 - 479383905861000x_5^2x_6^2$$
$$-299076127852260x_5x_6^4 - 3505609439955600x_5$$
$$x_6^3 - 91244417457024x_6^5 - 3557586742819200x_6^4 = 0.$$

along with positivity constraints $x_6 > 0$, $x_5 > 0$, $k_{30} > 0$, $k_{29} > 0$, and $k_{28} > 0$.

9. Grid Sampling: Symbolic vs Numeric

In this section we summarise work that was first presented in CASC 2017 (England et al., 2017) which compared the use of symbolic and numeric techniques to identify multistationary regions via grid sampling.

9.1. Algorithms and Software

In this section we will use Symbolic Grid Sampling: so we have results only for a set of numerical sample points, but each sample point will undergo a symbolic computation. The result will still be an approximate identification of the region, since the sampling will be finite, but the results at those sample points will be guaranteed free of numerical errors. The symbolic

26

computations follow exactly the strategy introduced in Section 4 except each sample point will set all parameters (rather than leaving one free) meaning a simpler symbolic computation than in Section 4 performed multiple times. In particular, with no free parameters the Lazy variant of Real Triangularization (LRT) used in Section 4 gives the full solution (no laziness) as we would get from Real Triangularization (RT) and so we just use the latter.

We will compare this symbolic grid sampling with a fully numerical gird sampling approach using the homotopy solver Bertini developed by Bates et al. (2013), in its standard configuration to compute complex roots. Alternatives to Bertini include PHCpack by Verschelde (2011) and the Numerical Algebraic Geometry package for Macaulay2 by Leykin (2011). Reasons for choosing Bertini include that it is the most cited homotopy solver for the past 8 years and that it allows adaptive and very high-precision arithmetic (whereas PHCpack only allows double-double)[10]. We parsed the output of Bertini using Python, and determined numerically which of the complex roots are real and positive using a threshold of 10^{-6} for positivity.

Bertini computations (v1.5.1) were carried out on a Linux 64 bit Desktop PC with Intel i7. Maple computations (v2016 with April 2017 Regular Chains) were carried out on a Windows 7 64 bit Desktop PC with Intel i5.

Software Remark 2. For the reduced system of Model 28 Bertini (incorrectly) could not find any roots, not even complex ones, for any of the parameter settings. The situation did not change when going from adaptive precision to a very high fixed precision. However, we have not attempted more sophisticated techniques like providing user homotopies. It seems a bug in Bertini has been triggered by this problem instance. It has been reported to the developers.

9.2. Sample Ranges and Plots

For Model 26 we will use a sampling range for k_{19} from 200 to 1000 by 50; for k_{17} from 80 to 200 by 10; and for k_{18} from 5 to 75 by 5.

For Model 28 we will use a sampling range for k_{30} from 100 to 1600 by 100; for k_{28} from 40 to 160 by 10; and for k_{29} from 120 to 240 by 10.

We produce 2d plots in each case with the third parameter fixed to its values indicated in (4) and (31). In those plots we will colour sample points according to the number of fixed points observed: yellow discs indicate one fixed point and blue boxes three. Diamonds indicate numerical errors where zero (red) or two (green) fixed states were identified.

9.3. Results and Comparison

The plots produced by the grid sampling are presented in Figures 7−10; and the time taken to produce them is summarised in Table 2.

9.3.1. Comparison of models

Model 28 forms a larger real algebraic problem than Model 26, 16 variables and equations rather than 11, so it unsurprising that it takes longer to perform computations.

Regarding the symbolic computations: Model 28 requires an actual CAD of a plane to be produced for each sample point while Model 26 only real root isolation (decomposition of a line).

---

[10]We note that a recent development for Bertini published after this article was in press could be applicable to this problem: Paramotopy by Bates et al. (2018) allows for parallelism and computation reuse, well suited for such grid sampling.

This was the case regardless of whether the original or reduced system was used as the starting point, since the RT preprocessing also reduced the number of variables that needed analysis by CAD. We note that even with the reduced system it was still beneficial to pre-process CAD with RT: the average time per sample point with pre-processing (and including time taken to pre-process) was 0.485 seconds while without it was 3.577 seconds. It is not clear if this is because of a genuine simplification or because the CAD algorithm from the Regular Chains Library that we used it particularly tuned for triangular systems.

9.3.2. *Effects of the pre-processing in Section 5*

Figure 7 and Figure 8 both refer to Model 26. The latter is produced by Maple's symbolic calculations and so guaranteed free of numerical error. The former, Figure 7, represents the output of Bertini on the original system. We see that there are numerous numerical errors present: the rouge red and green diamonds in Figure 7. We find that when computing with the reduced system rather than the original system Bertini was able to to avoid all these errors, producing the same plots as Maple in Figure 8.

With Model 28 we see similar numerical errors from Bertini in Figure 9 when compared with Maple in Figure 10. However, in the case of Model 28 the reduction led to catastrophic effects for Bertini: built-in heuristics quickly (and incorrectly) concluded that there are no zero dimensional solutions for the system, and when switching to a positive dimensional run also no solutions could be found.

From the timing data in Table 2 we see that both Bertini and Maple benefited from the reduced system: For Model 26 Bertini took a third of the original time while Maple took a tenth of the original. For Model 28 the speed-up enjoyed by the symbolic method from the pre-processing was even greater: almost 100 fold!

9.3.3. *Symbolic vs Numerical*

As described above, we have observed numerous numerical errors when using Bertini which may avoided with the symbolic computations of Maple. However, they can also be avoided (at least for Model 26) by using the pre-processing technique described in Section 5.

However, and surprisingly, for Model 26 the symbolic methods were actually quicker than the numerical ones. The symbolic methods used are well known for their doubly exponential computational complexity (in the number of variables) so it is not necessary surprising that as the system size increases the results of the comparison would change. For Model 28 we have the expected outcome of the numerical calculations being quicker.

Table 2: Timing data (in seconds) of the grid samplings described in Section 9. Numerical is using Bertini and Symbolic the Regular Chains Library for Maple.

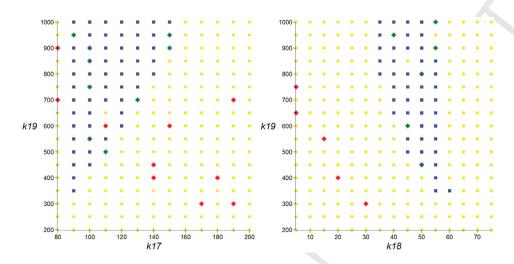| Model	**Numerical** Mean	**Symbolic** Mean	Median	StdDev	Maximum
26 – Original	2.4	0.568	0.530	0.107	0.905
26 – Reduced	0.85	0.053	0.047	0.036	0.343
28 – Original	16.57	42.430	40.529	8.632	84.116
28 – Reduced	$\perp$	0.485	0.468	0.119	0.796

28

Figure 7: Plots illustrating the result of Bertini's grid sampling on the original version of Model 26.
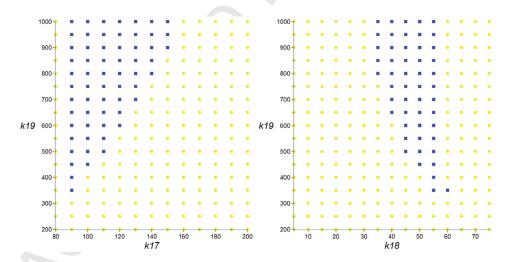

Figure 8: Plots illustrating the result of Bertini's numerical grid sampling on the reduced version of Model 26. These are also identical to those plots produced by Maple's symbolic grid sampling of Model 26 (both original and reduced versions).
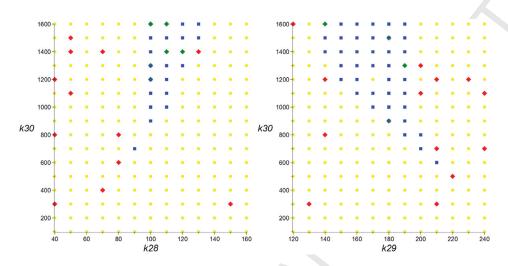
29

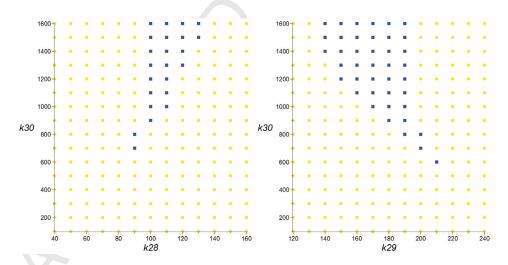Figure 9: Plots illustrating the result of Bertini's grid sampling on the original version of Model 28.

Figure 10: Plots illustrating the result of Maple's symbolic grid sampling on Model 28 (both original and reduced versions).

30

We can see some other statistical data for the timings in Maple: the standard deviation for the timings is fairly modest but in each row there are large outliers and so the median is always a little less than the mean average.

9.4. Higher Sampling Rates

Of course, the grid sampling described in this section scales directly with the number of sample points, so we can easily produce plots with higher sampling rates such as those shown later in Figure 11.

Figure 11: Higher sampling rate for symbolic grid sampling of Model 26.

10. Going Further

The work presented is a substantial step forward but there is a wide range of directions for future work.

10.1. Solution in 3-parameter Space

The complexity of the fully symbolic approaches puts a complete analysis over this space out of reach (for now). However, the grid-sampling method of Section 9 can already be extended into 3 parameters with relative ease: at a cost linearly proportional to the increased number of sample points. This was completed for Model 26, where the multistationarity region is bounded on both sides in the k_{17} and k_{18} directions but extends infinitely above in k_{19}. For example, with the k_{19} range bound at 1000 the region is bounded by extending k_{17} to 800 and k_{18} to 600. With a sample rate of 20 for k_{17} and k_{18} and 50 for k_{19} we have produced a Maple point plot of 20,400 points in 18 minutes. Figure 12 shows 2D captures of the 3D plot of the bistable points only. Figure 13 gives two views of the convex hull of the bistable points in Figure 12. This was produced using the convex package[11]. We note the lens shape seen in the orientation in the left plot is comparable with the image in the original paper of Markevich et al. (2004) (Fig. S7).

---

[11]http://www.math.uwo.ca/~mfranz/convex/
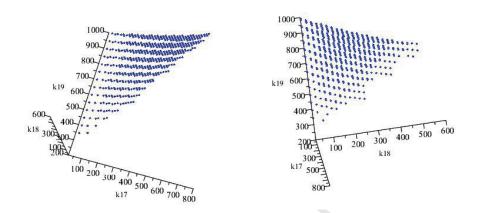
Figure 12: 3D Maple Point Plot produced grid sampling on Model 26.
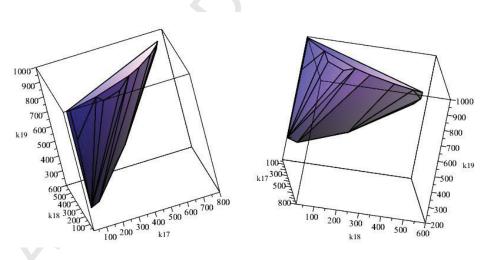

Figure 13: Convex Hull of the bistable points in Figure 12 for Model 26

32

10.2. Effect of Other Parameters

Our work has focussed on understanding the behaviour of the system in the 3-parameter space (k_{17}, k_{18}, k_{19}) but as described in Section 2 there are many other parameters for which we simply took the values from the BioModels Database. While there is confidence in the accuracy of these values, an important question for future work is the stability of the approaches we present to small perturbations in these values.

10.3. Conjecture for Semi-algebraic Solutions without CAD

All our semi-algebraic calculations used CAD as the backend to produce solutions, although after considerable simplification of the input. CAD is the most expensive technology employed by a significant margin. Its doubly exponential theoretical complexity is felt clearly in practice and so will be a barrier to studying larger parameter spaces or models. However, the results of Sections 4 and 6 hint that the solution could be available without CAD.

Recall from Section 4 that with one free-parameter the key break point in parameter space between 1 and 3 fixed points was determined by a real root of (8), one of the univariate polynomials whose roots were excluded from the validity of the LRT solution component. Similarly, studying the 28 cells where multistationarity could occur identified in Section 6 shows that the key region was also identified by the polynomial defining one of the graphs where LRT's solution component was not valid. Figures 14 and 15 give numerical plots of the polynomial (27), the former on smaller ranges and the latter on larger. The images on the right focus on the upper quadrant of interest and should be compared with Figures 4 and 5 of the exact multistationarity region. It is clear that (27) provides the boundary of this region. However, as the images on the left show, it is only one segment of the graph of this polynomial that is of interest.

Of course, this is just an observation. We have yet to derive a proof that this would always be identified by LRT. Even if it were there would still be things to clarify:

- Which polynomial from the several that LRT uses to define excluded regions is the one of interest? Recall from Section 4 that as well as (8) LRT identified two further polynomials in (22) and (23); while in Section LRT identified not only (27) but also (26).

- Which portion of the graph forms the boundary? The graph of (27) is a superset of the boundary. Even, when restricting our view to the positive quadrant (plot on the right of Figure 14) there is a second curve segment that does not have relevance to the application.

Nevertheless, we have identified a promising conjecture for continued study. At the least it gives useful insight on where to look for multistationarity without employing CAD. For example, it could direct future application of detailed grid sampling.

11. Summary and Final Thoughts

11.1. Summary

We have considered the problem of identifying regions of multistationarity in models of biological networks, an important problem with potentially clinical applications. We have investigated a variety of symbolic approaches encompassing multiple algorithms and computer algebra systems. We have derived semi-algebraic solution formulae and region descriptions for a classic MAPK model; as well as demonstrating the utility of symbolic-numeric grid sampling. We have drawn together the work first presented at conferences in 2017 (Bradford et al., 2017; England et al., 2017) and extended it to give solutions over a 2-parameter space not previously published and a conjecture on where future progress may come from.
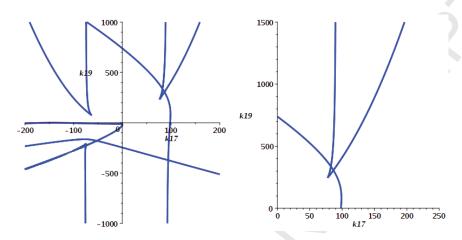
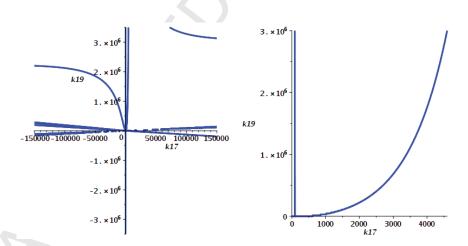Figure 14: Numerical plot of the graph of polynomial (27) on smaller ranges.

Figure 15: Numerical plot of the graph of polynomial (27) on larger ranges.

34

11.2. Final Thoughts

We hope this work will inspire further study on the application of symbolic tools to biological network analysis, from both communities. Indeed, work on developing Mathematica tools for such problems has now been undertaken by Lichtblau (2017), inspired by Bradford et al. (2017) but based on tools for discriminant varieties not considered there. The study of such real world problems is of great benefit not only to the application domains but also to the software developers: these MAPK studies uncovered bugs in both Regular Chains (see Software Remark 1 in Section 4.1) and Bertini (see Software Remark 2 in Section 9.1) which had escaped the numerous other tests and applications of those algorithms.

Key areas of future study include the sensitivity of the analysis to variations in the other parameters (Section 10.2) and the conjecture described in Section 10.3. Additional areas to investigate could include the various degrees of freedom with the algorithms used. For example, we have a free choice of variable ordering: Model 26 has 11 variables corresponding to $39\,916\,800$ possible orderings while Model 28 has 16 variables corresponding to more than 10^{13} orderings! Heuristics that exist to help with this choice, such as those of Dolzmann et al. (2004); Bradford et al. (2013), could not discriminate between the orderings on offer, even though the orderings do make a difference to the computation. Recent work on using machine learning to make such choice by Huang et al. (2014, 2016, 2019) may be applicable. Also, since MAPK problems contain many equational constraints an approach as described by England et al. (2015) may be applicable for the higher dimensional CADs required to study more parameters.

Semi-algebraic solutions over 3-parameter space is out of reach at the time of writing. We note however that instances like MAPK were until recently thought out of reach of symbolic computation altogether, and while writing the ISSAC 2017 contribution we thought the 2-parameter case of Section 6 out of reach. So further progress will surely follow.

Research Data Statement:

Data supporting the research in this paper is freely available in a Zenodo repository: https://doi.org/10.5281/zenodo.2560661.

Appendix A. Defining Polynomial of the Section 3 Break Point

In Section 3.3 a break point where the system moved from 1 to 3 positive real solutions was discovered at around $k_{19} = 409.253$. The exact point is an algebraic number defined as the only real zero of a polynomial $\sum_{i=0}^{10} c_i k_{19}^i$ with coefficients as below. Note that the coefficients are too large to fit on a single line: the line breaks between digits should be read as a continuation of the single coefficient description rather than anything else.

$c_{10} = 35159093450274029093689503326701715873606031394069307665
0155371250411$

$c_9 = -21369907285215767428399752774639558327303398317042608057
4800781989093156$

$c_8 = 25374851641220554774259605635053469432582109883965015804077119110958034090$

$c_7 = 12972493018300022707027639267804259251235991618029852880330004508564391594000$

$c_6 = -84689459636928024142264272497261234934483724397783490293556363169296870206600000$

$c_5 = 22310982703374064506703016631726643334214408338758486214236832656638465330796000000$

$c_4 = -376265008904112258290319173193792052014899485528994925965885895511831873444245100000$

$c_3 = 392621015487908694070579949853201565009689583613961789081800268428066437667831040000000$

$c_2 = -249262399074302923497435408127029610630960346245151705777987759684244828277993376000000000$

$c_1 = 709788507358874734591769971861759784258732672467600232129406169246431718684780800000000000$

$c_0 = -10628711928389858769480771149238982049904341389014953948347496131846703628103680000000000000$

Appendix B. Polynomial $f(x_1, k_{19})$ from Section 4.1.1

In Section 4 we described the application of LRT to (7). The main solution component provided the formulae 9–18 and required that $f(x_1, k_{19}) = \sum_{i=0}^{6} d_i x_1^i = 0$ where the coefficients d_i are as given below.

$d_6 = 16838105723097694257603469$

$d_5 = -24078605201553273505077988 k_{19} + 7723967969644977896148686580$

$d_4 = 8176202638735769127032169 k_{19}^2 - 772341166546354447701499460 k_{19}$

36

$$+ 123215435794133887615660681290$$

$$d_3 = 146540875744058984180345238 0 k_{19}^2$$
$$- 7981695575868055828424813098 00 k_{19}$$
$$+ 83152655240002767729550477640000$$

$$d_2 = 854625249012768461072516694 00 k_{19}^2$$
$$- 3526641140142765683457209514 0000 k_{19}$$
$$+ 255680535485331833219748963600 0000$$

$$d_1 = 16316856497197026722825055000 00 k_{19}^2$$
$$- 7219895711004618624773423200 00000 k_{19}$$
$$+ 2884375593831878082321840000 0000000$$

$$d_0 = -7013104139459910876520500 00000000000 k_{19}.$$

Appendix C. Evaluated LRT Solution Component from Section 6

In Section 6.1 we applied LRT to (25) to simplify that reduced system further before applying CAD. The evaluated solution component consisted of the positivity conditions $x_4 > 0, x_5 > 0, k_{17} > 0, k_{19} > 0$ and the two following equations.

$$(333770827232 x_5^4 + (3404343829252 k_{17} - 6863249873129 k_{19}$$
$$- 106111961633240) x_5^3 + (-3738114656484 k_{17}^2 + 7455351062094 k_{17} k_{19}$$
$$- 3717236405610 k_{19}^2 + 271801037104280 k_{17} - 114254579857600 k_{19}$$
$$- 831673402560000) x_5^2 + (-165689075471040 k_{17}^2$$
$$+ 165225032754600 k_{17} k_{19} + 2667668498040000 k_{17} - 129311541450000 k_{19}$$
$$- 2873589810000000) x_5 - 1835995095480000 k_{17}^2$$
$$+ 2873589810000000 k_{17}) x_4 + 2261223222841 x_5^5 + (-2274797538607 k_{17}$$
$$+ 2274721722856 k_{19} + 174844014037860) x_5^4 + (13574315766 k_{17}^2$$
$$- 27072815781 k_{17} k_{19} + 13498500015 k_{19}^2 - 176205245392020 k_{17}$$
$$- 883400777350 k_{19} + 6648403506290000) x_5^3 + (1361231354160 k_{17}^2$$
$$- 1355303940900 k_{17} k_{19} - 6671855445710000 k_{17} + 6724440511425000 k_{19}$$
$$+ 149432011365000000) x_5^2 + (23451939420000 k_{17}^2$$
$$- 149432011365000000 k_{17}) x_5 = 0 \tag{C.1}$$

$$487656080889027413 x_5^6 + (-1352408212353388839 k_{17}$$
$$+ 2227511326365959821 k_{19} + 97141513552593345960) x_5^5$$
$$+ (1810515745366146214 k_{17}^2 - 4490852292185431392 k_{17} k_{19}$$
$$+ 2680336546819285178 k_{19}^2 - 2206768034543466691680 k_{17}$$

$$+ 166893970054477098860k_{19} + 681914283986632293080 0)x_5^4$$

$$+ (-945763613901784788k_{17}^3 + 2832008529145922346k_{17}^2 k_{19}$$

$$- 2826726216586490328k_{17}k_{19}^2 + 940481301342352770k_{19}^3$$

$$+ 23939821125017070948 0 k_{17}^2 - 397099010517367066520k_{17}k_{19}$$

$$+ 89401058522195274400k_{19}^2 - 14716205773190097360400k_{17}$$

$$+ 83131286964761843470 00k_{19} + 3083305127820397418000 00)x_5^3$$

$$+ (-11586292134841736376 0 k_{17}^3 + 231195450091661030160k_{17}^2 k_{19}$$

$$- 115332528743243666400k_{17}k_{19}^2 + 116390967562785368984 00k_{17}^2$$

$$- 8542395106508656744000k_{17}k_{19} + 52336162668920130000 0 k_{19}^2$$

$$- 42066056463140319020000 0 k_{17} + 15948686720945888000000k_{19}$$

$$+ 51596772977068956000000 00)x_5^2 + (-37420338229547624688 00 k_{17}^3$$

$$+ 373285435455817357200 0 k_{17}^2 k_{19} + 148648818114128214000000k_{17}^2$$

$$- 262355559415638780000 00 k_{17}k_{19} - 5484239465944512000000000k_{17}$$

$$+ 51014470691381242500000 00 k_{19} + 1133654904252916500000000 00)x_5$$

$$- 3631876626476476560000 0 k_{17}^3 + 3245621682376164000000 00 k_{17}^2$$

$$- 11336549042529165000000000 0 k_{17} = 0 \qquad \text{(C.2)}$$

References

Arnon, D. S., Collins, G. E., McCallum, S., 1984. Cylindrical algebraic decomposition I: The basic algorithm. SIAM J. Comput. 13 (4), 865–877.

Aubry, P., Lazard, D., Moreno Maza, M., 1999. On the theories of triangular sets. J. Symb. Comput. 28 (1-2), 105–124.

Bates, D., Brake, D., Niemerg, M., 2018. Paramotopy: Parameter homotopies in parallel. In: Davenport, J., Kauers, M., Labahn, G., Urban, J. (Eds.), Mathematical Software – Proc. ICMS 2018. Vol. 10931 of Lecture Notes in Computer Science. Springer International Publishing, pp. 28–35.

Bates, D. J., Hauenstein, J. D., Sommese, A. J., Wampler, C. W., 2013. Bertini: Software for numerical algebraic geometry. doi:10.7274/R0H41PB5.

Bhalla, U. S., Iyengar, R., 1999. Emergent properties of networks of biological signaling pathways. Science 283 (5400), 381–387.

Bradford, R., Davenport, J., England, M., Errami, H., Gerdt, V., Grigoriev, D., Hoyt, C., Kosta, M., Radulescu, O., Sturm, T., Weber, A., 2017. A case study on the parametric occurrence of multiple steady states. In: Proc. ISSAC '17. ACM, pp. 45–52.

Bradford, R., Davenport, J., England, M., McCallum, S., Wilson, D., 2016. Truth table invariant cylindrical algebraic decomposition. J. Symb. Comput. 76, 1–35.

Bradford, R., Davenport, J., England, M., Wilson, D., 2013. Optimising problem formulations for cylindrical algebraic decomposition. In: Carette, J., Aspinall, D., Lange, C., Sojka, P., Windsteiger, W. (Eds.), Intelligent Computer Mathematics. Vol. 7961 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 19–34.

Brown, C., 2003. QEPCAD B: A program for computing with semi-algebraic sets using CADs. ACM SIGSAM Bulletin 37 (4), 97–108.

Caviness, B., Johnson, J., 1998. Quantifier Elimination and Cylindrical Algebraic Decomposition. Texts & Monographs in Symbolic Computation. Springer-Verlag.

Chen, C., Davenport, J., May, J., Moreno Maza, M., Xia, B., Xiao, R., 2013. Triangular decomposition of semi-algebraic systems. J. Symb. Comput. 49, 3–26.

Chen, C., Moreno Maza, M., Xia, B., Yang, L., 2009. Computing cylindrical algebraic decomposition via triangular decomposition. In: Proc. ISSAC '09. ACM, pp. 95–102.

Collins, G., 1998. Quantifier elimination by cylindrical algebraic decomposition – 20 years of progress. In: Caviness, B., Johnson, J. (Eds.), Quantifier Elimination and Cylindrical Algebraic Decomposition. Texts & Monographs in Symbolic Computation. Springer-Verlag, pp. 8–23.

Conradi, C., Feliu, E., Mincheva, M., Wiuf, C., 2017. Identifying parameter regions for multistationarity. PLoS Comput. Biol. 13 (10), Article e1005751.

Conradi, C., Flockerzi, D., Raisch, J., 2008. Multistationarity in the activation of a MAPK: parametrizing the relevant region in parameter space. Math. Biosci. 211 (1), 105–31.

Conradi, C., Mincheva, M., 2014. Catalytic constants enable the emergence of bistability in dual phosphorylation. Journal of The Royal Society Interface 11 (95).

Craciun, G., Dickenstein, A., Shiu, A., Sturmfels, B., 2009. Toric dynamical systems. J. Symb. Comput. 44 (11), 1551–1565.

Dolzmann, A., Seidl, A., Sturm, T., 2004. Efficient projection orders for CAD. In: Proc. ISSAC '04. ACM, pp. 111–118.

Dolzmann, A., Sturm, T., 1997a. Redlog: Computer algebra meets computer logic. ACM SIGSAM Bulletin 31 (2), 2–9.

Dolzmann, A., Sturm, T., Aug. 1997b. Simplification of quantifier-free formulae over ordered fields. J. Symb. Comput. 24 (2), 209–231.

England, M., Bradford, R., Davenport, J., 2015. Improving the use of equational constraints in cylindrical algebraic decomposition. In: Proc. ISSAC '15. ACM, pp. 165–172.

England, M., Davenport, J., 2016. The complexity of cylindrical algebraic decomposition with respect to polynomial degree. In: Proceedings of the CASC 2016. Vol. 9890 of LNCS. Springer, pp. 172–192.

England, M., Errami, H., Grigoriev, D., Radulescu, O., Sturm, T., Weber, A., 2017. Symbolic versus numerical computation and visualization of parameter regions for multistationarity of biological networks. In: Computer Algebra in Scientific Computing (Proc. CASC '17). Vol. 10490 of Lecture Notes in Computer Science. Springer, pp. 93–108.

England, M., Wilson, D., Bradford, R., Davenport, J., 2014. Using the Regular Chains Library to build cylindrical algebraic decompositions by projecting and lifting. In: Hong, H., Yap, C. (Eds.), Mathematical Software – ICMS 2014. Vol. 8592 of Lecture Notes in Computer Science. Springer Heidelberg, pp. 458–465.

Famili, I., Palsson, B. Ø., 2003. The convex basis of the left null space of the stoichiometric matrix leads to the definition of metabolically meaningful pools. Biophys. J. 85 (1), 16–26.

Feinberg, M., 1987. Stability of complex isothermal reactors–I. the deficiency zero and deficiency one theorems. Chem. Eng. Sci. 42 (10), 2229–2268.

Girvan, M., Newman, M. E. J., 2002. Community structure in social and biological networks. Proc. Natl. Acad. Sci. USA 99 (12), 7821–7826.

Grandoni, F., Könemann, J., Panconesi, A., 2008. Distributed weighted vertex cover via maximal matchings. ACM Trans. Algorithms 5 (1), 1–12.

Grigoriev, D., Samal, S. S., Vakulenko, S., Weber, A., 2015. Algorithms to study large metabolic network dynamics. Math. Model. Nat. Phenom. 10 (5), 100–118.

Grigoriev, D., Vorobjov, N. N., 1988. Solving systems of polynomial inequalities in subexponential time. J. Symb. Comput. 5, 37–64.

Gross, E., Davis, B., Ho, K. L., Bates, D. J., Harrington, H. A., 2016a. Numerical algebraic geometry for model selection and its application to the life sciences. Journal of The Royal Society Interface 13 (123).

Gross, E., Harrington, H. A., Rosen, Z., Sturmfels, B., 2016b. Algebraic systems biology: A case study for the Wnt pathway. Bull. Math. Biol. 78 (1), 21–51.

Hong, H., Liska, R., Steinberg, S., 1997. Testing stability by quantifier elimination. J. Symb. Comput. 24 (2), 161–187.

Huang, Z., England, M., Davenport, J., Paulson, L., 2016. Using machine learning to decide when to precondition cylindrical algebraic decomposition with Groebner bases. In: 18th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC '16). IEEE, pp. 45–52.

Huang, Z., England, M., Wilson, D., Bridge, J., Davenport, J., Paulson, L., 2019. Using machine learning to improve cylindrical algebraic decomposition. Mathematics in Computer Science Volume to be assigned, 28 pages.
URL https://doi.org/10.1007/s11786-019-00394-8

Huang, Z., England, M., Wilson, D., Davenport, J., Paulson, L., Bridge, J., 2014. Applying machine learning to the problem of choosing a heuristic to select the variable ordering for cylindrical algebraic decomposition. In: Intelligent Computer Mathematics. Vol. 8543 of LNAI. Springer, pp. 92–107.

Johnston, M. D., 2014. A note on "MAPK networks and their capacity for multistationarity due to toric steady states". arXiv:1407.5651.

Joshi, B., Shiu, A., 2015. A survey of methods for deciding whether a reaction network is multistationary. Math. Model. Nat. Phenom. 10 (5), 47–67.

Karp, R. M., 1972. Reducibility among combinatorial problems. In: Complexity of Computer Computations. Plenum Press, New York, pp. 85–103.

Košta, M., December 2016. New concepts for real quantifier elimination by virtual substitution. Doctoral dissertation. available from http://dx.doi.org/10.22028/D291-26679, Saarland University, Germany.

Legewie, S., Schoeberl, B., Blüthgen, N., Herzel, H., 2007. Competing docking interactions can bring about bistability in the MAPK cascade. Biophys. J. 93 (7), 2279–2288.

Leykin, A., 2011. Numerical algebraic geometry. Journal of Software for Algebra and Geometry 3, 5–10.

Li, C., Donizelli, M., Rodriguez, N., Dharuri, H., Endler, L., Chelliah, V., Li, L., He, E., Henry, A., Stefan, M. I., Snoep, J. L., Hucka, M., Le Novère, N., Laibe, C., 2010. BioModels database: An enhanced, curated and annotated resource for published quantitative kinetic models. BMC Systems Biology 4, 92.

Lichtblau, D., 2017. Symbolic analysis of multiple steady states in a MAPK chemical reaction network. Under Preparation − received in personal communication.

Loos, R., Weispfenning, V., 1993. Applying linear quantifier elimination. THE Computer Journal 36 (5), 450–462.

Markevich, N. I., Hoek, J. B., Kholodenko, B. N., 2004. Signaling switches and bistability arising from multisite phosphorylation in protein kinase cascades. J. Cell Biol. 164 (3), 353–359.

McCallum, S., 1993. Solving polynomial strict inequalities using cylindrical algebraic decomposition. The Computer Journal 36 (5), 432–438.

Millán, M. P., Dickenstein, A., 2018. The structure of MESSI biological systems. SIAM Journal on Applied Dynamical Systems 17 (2), 1650–1682.

Millán, M. P., Turjanski, A. G., 2015. MAPK's networks and their capacity for multistationarity due to toric steady states. Math. Biosci. 262, 125–37.

Rashevsky, N., 1960. Mathematical Biophysics: Physico-Mathematical Foundations of Biology. Dover, New York.

Schuster, S., Höfer, T., 1991. Determining all extreme semi-positive conservation relations in chemical reaction systems: a test criterion for conservativity. J. Chem. Soc. Faraday T. 87 (16), 2561–2566.

Strzeboński, A., 2000. Solving systems of strict polynomial inequalities. Journal of Symbolic Computation 29 (3), 471–480.

Sturm, T., December 2017. A survey of some methods for real quantifier elimination, decision, and satisfiability and their applications. Math. Comput. Sci. 11 (3–4), 483–502.

Sturm, T., 2018. Thirty years of virtual substitution. In: Proc. ISSAC 2018. ACM, pp. 11–16.

Vakulenko, S., Grigoriev, D., Weber, A., 2015. Reduction methods and chaos for quadratic systems of differential equations. Stud. Appl. Math. 135 (3), 225–247.

Verschelde, J., 2011. Polynomial homotopy continuation with PHCpack. ACM Communications in Computer Algebra 44 (3/4), 217–220.

Wang, D., 2000. Elimination Methods. Springer.

Wang, D., Xia, B., 2005. Stability analysis of biological systems with real solution classification. In: Proceedings of the ISSAC 2005. ACM, pp. 354–361.

Weispfenning, V., Feb.–Apr. 1988. The complexity of linear problems in fields. J. Symb. Comput. 5 (1&2), 3–27.

Weispfenning, V., 1994. Quantifier elimination for real algebra—the cubic case. In: Proceedings of the international symposium on Symbolic and algebraic computation. ACM Press, pp. 258–263.

Weispfenning, V., Feb. 1997. Quantifier elimination for real algebra—the quadratic case and beyond. Appl. Algebr. Eng. Comm. 8 (2), 85–101.

Weng, G., Bhalla, U. S., Iyengar, R., 1999. Complexity in biological signaling systems. Science 284 (5411), 92–6.

Wilson, D., Bradford, R., Davenport, J., England, M., 2014. Cylindrical algebraic sub-decompositions. Mathematics in Computer Science 8, 263–288.

Zumsande, M., Gross, T., 2010. Bifurcations and chaos in the MAPK signaling cascade. J. Theor. Biol. 265 (3), 481–491.