HAL open science

Tropical geometry and interval arithmetic methods for the analysis of biochemical networks : homeostasis research and model reduction in the presence of conservation laws

Aurélien Desoeuvres

To cite this version:

Aurélien Desoeuvres. Tropical geometry and interval arithmetic methods for the analysis of biochemical networks : homeostasis research and model reduction in the presence of conservation laws. Biochemistry [q-bio.BM]. Université Montpellier, 2021. English. NNT : 2021MONTS107 . tel-03620247

HAL Id: tel-03620247

https://theses.hal.science/tel-03620247v1

Submitted on 25 Mar 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE POUR OBTENIR LE GRADE DE DOCTEUR DE L'UNIVERSITÉ DE MONTPELLIER

En Mathématiques et Modélisation

École doctorale I2S — Information, Structures, Systèmes

Unité de recherche UMR 5235 — Laboratory of Pathogen Host Interactions

Méthodes de géométrie tropicale et d'arithmétique d'intervalles pour l'analyse des réseaux biochimiques : recherche d'homéostasie et réduction de modèles en présence de lois de conservation

Présentée par Aurélien DESOEUVRES

Le 9 décembre 2021

Sous la direction de Ovidiu RADULESCU

Devant le jury composé de

Ovidiu RADULESCU, Professeur, Université de Montpellier	Directeur
Werner SEILER, Professor, Université de Kassel	Rapporteur
Peter SZMOLYAN, Professeur, Technische Universität Wien	Rapporteur
Jorge RAMIREZ ALFONSIN, Professeur, Université de Montpellier	Examinateur
Anne SIEGEL, Directrice de recherche, Université de Rennes 1	Examinatrice
Marianne AKIAN, Directrice de recherche, INRIA Saclay	Examinatrice

UNIVERSITÉ DE MONTPELLIER

Sommaire

1 Résumé en français **3**

2 Overview of thesis and models **7**

2.1 Introduction . . . 7

2.2 Biochemical models . . . 9

3 Interval methods for Homeostasis and multistationarity **15**

3.1 Introduction . . . 15

3.2 Settings and definitions . . . 16

3.3 Constraint methods for interval arithmetic . . . 18

3.4 Using interval methods to find homeostasis . . . 26

3.5 IbexHomeo, a multiple bi-optimization process . . . 26

3.6 Multistationarity . . . 29

3.7 Experimental Results . . . 31

3.7.1 Multistationarity . . . 31

3.7.2 Homeostasis . . . 33

3.8 Some models presenting homeostasis . . . 34

3.9 IbexBoxHull, firsts experimentation . . . 37

4 Tropical geometry for chemical reaction networks **39**

4.1 Introduction . . . 39

4.2 Background . . . 40

4.3 Tropical equilibration . . . 43

4.4 Tropical equilibration of the Tyson cell cycle model . . . 45

4.5 Algorithms for tropical equilibration . . . 51

5 Model reduction **57**

5.1 Introduction . . . 57

5.2 Singular perturbation . . . 58

5.2.1 Fenichel theorems . . . 59

5.2.2 Extension to the multi-scale systems . . . 61

5.3 A tropical Scaling Procedure . . . 63

6 Approximated conservation laws for model reduction **69**
6.1 Exact and approximate conservation laws 69
6.2 Conservation laws as slow variables 75
6.2.1 Reduction of the Michaelis-Menten Model under Quasi-equilibrium Conditions. 80

7 The Model Reduction Algorithms **83**
7.1 The Slowest Timescale Reduction 84
7.2 Nested intermediate timescale reductions 85
7.3 Hyperbolically attractive chains 86
7.4 Algorithmic solutions for eliminating the approximate conservation laws . . . 92
7.5 Case study: TGFβ 95
7.5.1 Model and its scaling 95
7.5.2 Elimination of the conservation laws 97
7.5.3 Reduced models 101
7.6 Conclusions 106

8 Computing Conservation laws **111**
8.1 Computing Linear Conservation Laws 111
8.2 Computing monomial conservation laws 113
8.3 Computing polynomial conservation laws 118

Chapter 1

Résumé en français

> Tous les modèles sont faux, mais certains sont utiles. George E. P. Box

> L'art de la modélisation mathématique, c'est un équilibre entre choisir un bon niveau de complexité et garder un certain degré de compréhension. Maximilian Strobl.

Ces deux citations résument parfaitement les difficultés à trouver un bon modèle, qui puisse à la fois décrire précisément la réalité, avoir un pouvoir prédictif, et en même temps être suffisamment simple pour être étudiable. C'est le problème que rencontre actuellement la biologie.

En effet, grâce aux avancées techniques des dernières décennies, la biologie profite de nouvelles données toujours plus nombreuses, permettant une meilleure compréhension des systèmes vivants, et donnant naissance à de nouveaux champs de recherches actifs, comme la multi-omique. Cependant, cette avalanche de données doit être analysée et comprise pour permettre la création de modèles toujours plus performant. Or des modèles aussi détaillés, aussi larges, peuvent admettre des centaines, voir des milliers d'interactions biochimiques, induisant des systèmes dynamiques tout aussi larges. Cela en rend l'analyse difficile, et la vérification expérimentale impossible.

Une stratégie pour contourner ce problème est de réduire le modèle en question, c'est-à-dire d'obtenir un modèle plus petit (par exemple, contenant moins d'équations, variables et paramètres) qui capture l'information essentielle du modèle d'origine, ou, en d'autres termes, qui décrive avec une précision suffisante le comportement du modèle d'origine. Cependant, la réduction de modèle est un problème difficile, souvent obtenue au cas par cas, avec l'utilisation de diverses méthodes. Le but est aujourd'hui de trouver des méthodes de réductions systématiques et prenant en compte l'un des problèmes les plus important : l'incertitude paramétrique caractérisant les systèmes biologiques.

En effet, le monde du vivant est, pour ainsi dire, vivant, en mouvement perpétuel. Or, pour déterminer les paramètres d'une interaction il faut l'isoler de son contexte, qui en biologie est extrêmement variable, ou il faut avoir accès à suffisament de données pour caractériser quantitativement le système complet d'interactions, contexte inclu. En absence de données précises, des nombreux paramètres sont incertains. En outre, des différences sont souvent observées entre les valeurs d'un paramètre obtenues par des expériences in-vitro et in-

vivo [Richey et al., 1987,Song et al., 2015]. Cela montre, d'une part, la difficulté d'obtenir un modèle biologique complet, de nombreuses interactions pouvant être manquantes, et d'autre part, la difficulté, de ce fait, à obtenir des paramètres précis pour un modèle permettant de décrire le comportement de la réalité. C'est pourquoi il est important de développer des méthodes mathématiques capables de gérer l'incertitude paramétrique lors de l'analyse des modèles biologiques. En effet, si les résultats d'une analyse de modèle étaient très sensibles aux changements de paramètres, il serait difficile d'en tirer des prédictions valables pour des applications, dans la mesure où deux ensembles de paramètres relativement proches mèneraient à des résultats différents, et donc à des prédictions différentes.

Dans cette thèse, nous appliquons à l'étude des sytèmes biologiques deux classes de méthodes mathématiques capables de gérer l'incertitude paramétrique : l'arithmétique des intervalles et la géométrie tropicale. Nous utilisons la première pour étudier deux propriétés importantes des modèles biologiques, l'homéostasie et la multistationnarité ; et la deuxième pour obtenir des réductions robustes de modèles biologiques.

La première partie de la thèse est consacrée aux méthodes d'arithmétique d'intervalles appliquées à la recherche d'homéostasie et de multistationnarité dans les modèles biologiques.

L'homéostasie est un concept important en biologie, introduit par Claude Bernard au dix-neuvième siècle [Bernard, 1879], et considérée comme élément essentiel pour la définition de la vie [Bartlett and Wong, 2020]. Le principe d'homéostasie peut être vu comme une résistance aux changements. L'exemple typique d'homéostasie concerne la température corporelle, qui reste stable malgré des changements importants de température extérieure. En biochimie, les exemples d'homéostasie concernent les concentrations d'espèces chimiques fortement régulées, comme le glucose, le fer, le calcium, etc.

L'homéostasie étant un concept large, il englobe plusieurs définitions plus spéciales et se retrouve couplé à d'autre principes similaires, comme la résilience. Notre approche visant à trouver l'homéostasie dans un modèle d'équations différentielles ordinaires (EDO) s'inspire de [Golubitsky and Stewart, 2017]. Nous reprenons ainsi son principe de fonction d'entrée-sortie sur lequel repose sa définition d'homéostasie. Nos deux approches diffèrent cependant sur les outils utilisés. En effet, leur approche est basée sur la théorie des singularités quand la notre repose sur l'arithmétique des intervalles.

Nous considérons en effet qu'il y a homéostasie pour une espèce donnée par rapport à des changements donnés lorsque, à l'état d'équilibre, la concentration de l'espèce chimique est encadrée par un petit intervalle quand les paramètres varient.

Dans cette partie de la thèse on décrit un algorithme capable de trouver les espèces homéostatiques d'un système, basé sur des schémas d'optimisations globales. L'idée derrière est que trouver de l'homéostasie chez une espèce revient en fait à résoudre un couple d'optimisations globales. Ainsi, cet algorithme cherche en principe à trouver une boîte minimale contenant les solutions d'un problème algébrique lorsque des paramètres varient.

Au cours du développement de cet algorithme, on s'est aperçu que l'on pouvait utiliser les mêmes outils pour un autre problème tout aussi important : celui de la multistationarité d'un système. Car en effet, si l'on fixe les paramètres, la mono-stationarité implique comme résultat un point. En revanche, si le système est multistationaire, alors la plus petite boîte contenant toutes les solutions au problème ne peut être réduite à un point.

Il est donc intéressant de noter que les deux problèmes peuvent être vus comme duaux, l'un cherchant une sortie constante lorsque les paramètres varient, l'autre cherchant une sortie variable lorsque les paramètres sont constants.

On a ensuite fait plusieurs benchmark, basés sur des modèles réalistes, dans le but de tester l'homéostasie et la multistationarité.

La deuxième partie de la thèse concerne la réduction des modèles de réseaux biochimiques.

Notre approche pour la réduction de modèle s'appuie, d'une part, sur une mise à l'échelle basée sur des méthodes de géométrie tropicale, et d'autre part, sur des résultats de perturbations singulières. La théorie des perturbations singulières s'intéresse à des problèmes dépendant d'un petit paramètre ϵ et pour lesquels le problème non perturbé peut avoir un comportement très différent. En d'autre termes, on s'intéresse à un problème ayant pour solution $x(t, \epsilon)$, et si, pour une norme appropriée, la différence $\|x(t, \epsilon) - x(t, 0)\|$ ne tend pas vers zéro lorsque ϵ tend vers zéro uniformément en t, alors nous faisons face à un problème de perturbation singulière [Verhulst, 2005b]. Cela concerne en particulier les problèmes de couche limite pour lesquels les dérivées d'ordres supérieurs sont multipliées par ϵ. Les problèmes qui nous intéressent dans notre cas sont des problèmes de couche limite dans le temps pour des systèmes d'EDOs, qui peuvent se trouver sous la forme $\dot{x} = f(x, y),\ \epsilon\dot{y} = g(x, y)$.

En effet, les modèles de réseaux biochimiques font régulièrement intervenir plusieurs échelles de temps, certaines réactions biochimiques s'effectuant en quelques secondes lorsque d'autres prennent des heures, voir des jours à s'effectuer. Cela peut être dû à des différences de concentrations importantes entre les espèces, à la présence d'espèces inhibantes ou activantes, à des constantes cinétiques différentes.

Dans le cadre de cette thèse, on travaille majoritairement sur des systèmes d'équations différentielles ordinaires polynomiales représentant la dynamique d'un réseau de réactions chimiques. Chaque monôme, représentant la vitesse d'une réaction, se compose d'une constante cinétique et d'un produit de concentrations d'espèces chimiques. Pour obtenir la mise à l'échelle nécessaire à la réduction des modèles, on représente chaque valeur par une puissance (entière ou rationnelle) d'un unique petit paramètre positif ϵ. Ainsi un paramètre k_i est représenté par $\overline{k_i}\epsilon^{\gamma_i}$ où $\overline{k_i}$ est très proche de 1. De même, une concentration x_i se réécrit $y_i\epsilon^{a_i}$.

Cette façon de procéder permet d'être peu sensible aux incertitudes paramétriques. En effet, si on prend $\epsilon = 1/10$, on obtient l'ordre décimal, et il faut globalement multiplier ou diviser par 10 la valeur du paramètre ou de la concentration pour obtenir un ordre de grandeur différent. En outre, cela nous permet d'obtenir une valuation, et donc d'utiliser la géométrie tropicale.

La géométrie tropicale, dénommée ainsi en l'honneur du mathématicien Brésilien Imre Simon qui est un des pionniers de la théorie, est une branche des mathématiques où les opérations considérées sont le minimum et l'addition en lieu et place de l'addition et de la multiplication. En premier lieu fortement étudié en lien avec la géométrie algébrique, depuis les années 2000/2010 les applications du domaine visant le monde réel commencent à émerger [Krivulin, 2014]. En biologie, celle-ci permet de retrouver les conditions de quasi-equilibrium (QE) et de quasi-steady state (QSS) couramment utilisées en réduction de modèle [Radulescu et al., 2012].

Pour ce faire, nous utilisons le principe d'équilibration tropicale. L'idée derrière est que si on a deux forces opposées d'ordre similaire, toutes deux dominantes (plus importantes que les autres forces), alors l'objet soumis à ces deux forces se déplacera lentement. En fait, le problème d'équilibration tropicale consiste, pour chaque équation, à chercher un monôme négatif et un monôme positif, tout deux du même ordre minimal (pour l'équation). On obtient ainsi deux monômes de signes opposés, représentant deux forces dominantes

opposées et similaires.

Dans cette partie de la thèse, on s'intéresse aussi aux équilibrations tropicales partielles, où seules les espèces rapides sont équilibrées. On décrit un algorithme capable d'obtenir les équilibrations totales d'un système et comment les modifier pour obtenir un algorithme capable d'obtenir les équilibrations partielles. On étudie ainsi les équilibrations d'un modèle du cycle cellulaire de Tyson, dans un cadre symbolique et numérique. On trouve que la structure des équilibrations totales pour ce modèle ne peut prendre que trois formes : vide, un point, ou un segment et une demi-droite. Les équilibrations partielles sont quant à elles l'intersection de deux problèmes : un problème d'équilibration pour certaines équations seulement, et un problème de séparation d'échelles de temps. Cela rend plus complexe l'analyse purement symbolique du problème, où il devient plus difficile de réduire le problème par élimination de variables.

On se sert ensuite de la mise à l'échelle produite et des équilibrations partielles pour travailler sur la réduction de modèle. On s'intéresse plus précisément au cas critique où des lois de conservation sont présentes et rendent la matrice jacobienne singulière. Or, l'une des conditions imposée par les théorèmes classiques de perturbation singulière est l'hyperbolicité normale du système rapide, c'est-à-dire que toutes les valeurs propres de la matrice jacobienne pour le système rapide doivent avoir une partie réelle non-nulle. On s'intéresse donc aux lois de conservation approchées du modèle, c'est-à-dire les lois de conservation pour le système rapide, qui ne sont pas nécessairement des lois de conservation pour le système complet.

Nous montrons que ces lois de conservation peuvent être vus comme des espèces plus lentes que toutes les espèces impliquées dans celles-ci et utilisons ce résultat pour éliminer du système rapide les espèces rendant la matrice jacobienne singulière en les remplaçant par les espèces "loi de conservation", plus lentes. Ainsi, la méthode de réduction algorithmique proposée dans [Kruff et al., 2020], basée sur les résultats de Cardin et Teixeira [Cardin and Teixeira, 2017], peut de nouveau être utilisée. Cela nécessite cependant de modifier l'algorithme en question, car cette élimination doit se faire à chaque étape de la réduction (*ie.* pour chaque échelle de temps).

En effet, on travaille sur un système multi-échelle, et la réduction suit un schéma de réductions imbriquées. En commençant par considérer uniquement l'échelle de temps la plus rapide, on obtient une première réduction, on regarde ensuite l'échelle de temps suivante et ainsi de suite jusqu'à considérer l'échelle de temps la plus lente. En revanche, cela nécessite que l'hypothèse de hyperbolicité normale soit valide pour chaque étape de la réduction, et la vérification des lois de conservation approchées doit alors se faire à chaque étape.

Pour illustrer cette réduction, on présente une réduction du modèle TGFβ, qui possède des lois de conservation exactes et approchées pour des multiples échelles de temps.

Chapter 2

Overview of thesis and models

2.1 Introduction

Mathematics has been initially used in biology to describe population growth [Malthus, 1872, Verhulst, 1838, Lotka, 1925, Volterra, 1926] as a dynamical system (continuous or discrete). The last century, the field of mathematical biology has grown rapidly due to several reasons including:

- the development of mathematics,
- the rapid growth of data due to the genomics revolution,
- the growth of computational power.

In 1990 the human genome project start [Collins et al., 2003], leads to 13 years of data accumulations, and the conclusion that, if it was useful to understand life, it wasn't enough. Then was born the systems biology [Westerhoff and Palsson, 2004], a field dedicated to the analysis and modeling of complex biochemical systems. In systems biology, a problem is generally approached with a bottom-up or a top-down approach. The top-down approach begins with a big picture that is a simplified view of reality coming generally from experimental results and try to decompose it in smaller, more detailed, segments. The bottom-up approach is the reversed situation when small detailed systems are assembled together to form more complex systems. But in 2003, Denis Noble proposes a middle-out approach [Noble, 2003], that tries to capture just enough details to render the essence of the overall system organisation. However, if this approach seems to be the more reasonable, it is difficult to apply it today, essentially due to a lack of general mathematical methods adapted to the biological context. In this thesis, we build mathematical tools that are well adapted to middle-out approaches, namely they can be used to capture the "just enough details" of the biochemical systems.

Today, models can have hundreds of variables and their analysis becomes difficult if not impossible. Moreover, these models can have parameters uncertainty and some mechanism could be still missing to understand the reality. Thus, model reduction becomes an important problem in biology. There exist several methods to reduce dynamic systems coming from chemical reaction networks. Methods based on conservation laws, exact lumping [Feret et al.,

2009], or symmetry [Rowley and Marsden, 2000], that use only the formal representation of the system. Methods based on timescale separation, such as computational singular perturbation [Lam and Goussis, 1994], or intrinsic low dimensional manifold [Maas and Pope, 1992]. However, the last two methods, numeric, perform a reduction locally and are based on simulation. So, they are dependant of the parameters and possibly not robust, and they do not provide all the possible reductions.

The most known reduction of systems biology is the Michaelis-Menten model [Johnson and Goody, 2011,Briggs and Haldane, 1925], with two different processes depending on the values concentration of the species. These reduction are known as quasi-equilibrium (QE), and quasi-steady state (QSS). Recently, new ways to detect QE reactions and QSS species has been proposed in [Radulescu et al., 2012], based on singular perturbation or on tropical geometry.

Using tropical geometry, we are able to detect species order concentration and time scale reactions, based on the tropical equilibration problem, to reduce the system using singular perturbation results. This reduction is robust w.r.t the parameters and the species concentrations, and offer several reduction, each reduction resulting from a branch of the tropical prevariety computed for the tropical equilibration problem.

My PhD is about formal methods for analysing biochemical reaction network models. My focus is on methods coping with parametric uncertainty, whose predictions are valid for parameters contained within intervals or whose orders of magnitudes are known. The thesis splits in two parts: one deals with interval methods, the other is about tropical methods.

In the first part of my thesis, I introduce a new algorithm based on interval arithmetics, constraint methods, and optimization, that allows to test homeostasis, defined as dependence of steady states on the parameters. This concept includes absolute concentration robustness that has been introduced elsewhere. I also show how to use the same kind of methods in order to test if biochemical network models have a single or multiple steady states.

In the second part of my thesis, I present two novel contributions to model reduction methods.

The first contribution concerns the concept of approximated conservation laws. A method for model reduction combining tropical geometry and singular perturbation results have been recently proposed by our team and others, but there are some cases when this method fails. One cause of failure is when the fast subsystem defined by the tropical method has conservation laws, that are not conserved by the full system. This case covers the "quasi-equilibrium" situation, well known in biochemistry. Then, we use these approximated conservation laws to reduce the model. We have proven that approximated conservation laws are slower than species involved in it and can then be used for model reduction. We also provide algorithmic results for finding these approximated conservation laws (linear, monomial or polynomials) and for testing hyperbolicity conditions needed for the application of singular perturbations techniques.

Another direction is to generalise the full tropical equilibrations, previously introduced by our team and others, into partial tropical equilibrations. The concept of partial equilibrations allows to identify reduced models valid in regions of the phase and parameter space where there are no full equilibrations. Heuristically, the concept is justified by the fact that slow species of multiple time scales models don't need to be equilibrated in the model reduction process. I provide algorithmic methods for computing the polyhedral complex of partial equilibrations and discuss how these can be used for rescaling and reducing biochemical

reaction models.

2.2 Biochemical models

In this chapter, we introduce the chemical reaction network models used throughout the thesis, as well as and their links with mathematics.

Definition 2.1. A chemical reaction network (CRN) is a triple $(\mathcal{X}, \mathcal{C}, \mathcal{R})$ of sets such that:

- $\mathcal{X} = \{A_1, ..., A_n\}$ is a set of n species, where A_i $(i = 1, ..., n)$ is identified with the ith unit vector.
- $\mathcal{C} = \{C_1, ..., C_m\}$ is a set of m complexes, such that $C_i \in \mathbb{N}^n$ $(i = 1, ..., m)$ is a non-negative linear combination of species,
- $\mathcal{R} \subset \mathcal{C} \times \mathcal{C}$ is a set of reactions.

Usually, reactions are denoted by $C_i \to C_j$ and the set of reaction is enough to describe a CRN since complexes and species can be extracted from them.

Remark 2.2. Generally, we will denote $x_1, ..., x_n$ the species concentration of $A_1, ..., A_n$

To a CRN we can associate a dynamical system. For example, we can associate an ordinary differential equations (ODE) system, where the variable are concentrations of the species and each reaction plays a role in the speed of variation of the concentrations. In order to associate a dynamical system to a CRN, one needs a set of rate laws that compute the reaction rates as functions of the concentrations of the reactants. The usual choice is the mass action law, that was formulated in the context of chemical equilibria over the period 1864–79 by the Norwegian scientists Cato M. Guldberg and Peter Waage [Lund, 1965].

Definition 2.3. A CRN follows the mass action law if for each reaction, the reaction rate is proportional to the product of the concentrations of the reactant molecules.

One should keep in mind that the mass action law applies only to elementary reactions, i.e. to reactions that can not be decomposed into simpler ones.

Example 2.4. The following model, designed by Michaelis and Menten in 1913 [Johnson and Goody, 2011], is a CRN following the mass action network:

$$
\begin{aligned}
S + E \underset{k_2}{\overset{k_1}{\rightleftharpoons}} ES \xrightarrow{k_3} E + P \\
\dot{x}_1 &= -k_1 x_1 x_3 + k_2 x_2 \\
\dot{x}_2 &= k_1 x_1 x_3 - k_2 x_2 - k_3 x_2 \\
\dot{x}_3 &= -k_1 x_1 x_3 + k_2 x_2 + k_3 x_2 \\
\dot{x}_4 &= k_3 x_2
\end{aligned}
\tag{2.1}
$$

where $x_1 = [S],\ x_2 = [ES],\ x_3 = [E],\ x_4 = [P]$.

This model describe the kinetics of an enzyme acting on a substrate and is one of the most common model. It is also a simple example of model reduction, as we will see later.

In this thesis, unless specified otherwise, we use a system of polynomial ODEs to describe the dynamics of a CRN with n species $A_1, \ldots, A_n$ whose concentrations are $\boldsymbol{x} = (x_1, \ldots, x_n)$, as following:

$$\dot{x}_1 = f_1(\boldsymbol{k}, \boldsymbol{x}), \ \ldots, \ \dot{x}_n = f_n(\boldsymbol{k}, \boldsymbol{x}) \tag{2.2}$$

where

$$f_i(\boldsymbol{k}, \boldsymbol{x}) = \sum_{j=1}^{r} S_{ij} k_j \boldsymbol{x}^{\boldsymbol{\alpha}_j} \in \mathbb{Z}[\boldsymbol{k}, \boldsymbol{x}] = \mathbb{Z}[k_1, \ldots, k_r, x_1, \ldots, x_n], \tag{2.3}$$

$\boldsymbol{x}^{\boldsymbol{\alpha}_j} = x_1^{\alpha_{j1}} \ldots x_n^{\alpha_{jn}}$, k_j represents a rate constant, and r is the number of reactions (which is also the number of distinct $k_j \boldsymbol{x}^{\boldsymbol{\alpha}_j}$). The variables $\boldsymbol{k} = (k_1, \ldots, k_r)$ take values in $\mathbb{R}_+^*$ and the integer coefficients S_{ij} form a matrix $\boldsymbol{S} = (S_{ij}) \in \mathbb{Z}^{n \times r}$ which is called the stoichiometric matrix. We denote the vector of right hand sides of (2.2) by

$$\boldsymbol{F}(\boldsymbol{k}, \boldsymbol{x}) = (f_1(\boldsymbol{k}, \boldsymbol{x}), f_2(\boldsymbol{k}, \boldsymbol{x}), \ldots, f_n(\boldsymbol{k}, \boldsymbol{x}))^{\mathsf{T}}.$$

One can see that in this system, coefficient are splitted into two parts, the integer coefficients S_{ij} provided by the stoichiometric matrix, and the rate constant k_j depending of the reaction implied. In fact, all polynomial systems can be described as in (2.3). Indeed, if the system is given as a symbolic one, it is already in this form. And in the case where polynomial systems are given with numeric values, hiding the values of the stoichoimetric matrix and the parameters, then we can use the following algorithm 1 to describe it as in (2.3). The algorithm exploits that some of these numeric values are dependent over the integers, and can then be interpreted as resulting from the same reaction, but with a different integer stoichiometric coefficient. Then, we can extract from this the stoichiometric matrix $\boldsymbol{S} = (S_{ij})$ using algorithm 2.

Theorem 2.5. *Any system of polynomial ODEs*

$$\dot{x}_1 = f_1(\boldsymbol{k}, \boldsymbol{x}), \ \ldots, \ \dot{x}_n = f_n(\boldsymbol{k}, \boldsymbol{x})$$

where $f_i(\boldsymbol{k}, \boldsymbol{x}) \in \mathbb{Z}[\boldsymbol{k}, \boldsymbol{x}]$ is polynomial in $\boldsymbol{k}$ and $\boldsymbol{x}$, homogeneous of degree one in $\boldsymbol{k} = (k_1, \ldots, k_r)$, can be written as (2.2) for an appropriate matrix $\boldsymbol{S}$.

Proof. A constructive proof is given by Algorithm 2. □

If the stoichiometric matrix doesn't have full rank, then we can find a conservation law, which is another important property in dynamical systems.

Definition 2.6. A conservation law of an ODE system is a quantity that is constant over time, we call this constant a total amount. More precisely, a function $\phi(\boldsymbol{k}, \boldsymbol{x})$ is a conservation law if it is a first integral of the system (2.2), i.e. if

$$\sum_{i=1}^{n} \frac{\partial \phi}{\partial x_i}(\boldsymbol{k}, \boldsymbol{x}) f_i(\boldsymbol{k}, \boldsymbol{x}) = 0$$

for all $\boldsymbol{x}$.

```
Algorithm CompareAndSplitCoefficients(ẋ = Σ_{h=1}^{m} B_h x^{α_h})
    for h=1 to m do
        for i=1 to n do
            for j=1 to i do
                if B_ih/B_jh ∈ ℤ then
                    s_ih := sign(B_ih)|B_ih/B_jh|, k_ih := B_ih/s_ih, s_jh := sign(B_jh),
                      k_jh := B_jh/s_jh
                else if B_jh/B_ih ∈ ℤ then
                    s_jh := sign(B_jh)|B_jh/B_ih|, k_jh := B_jh/s_jh, s_ih := sign(B_ih),
                      k_ih := B_ih/s_ih
                else
                    s_jh := sign(B_jh), k_jh := B_jh/s_jh, s_ih := sign(B_ih), k_ih := B_ih/s_ih
    return ẋ as in (2.2)
```

Algorithm 1: This algorithm allow to transform a numeric system into a symbolic one as in (2.3). For each $\boldsymbol{x}^{\boldsymbol{\alpha}_h}$, coefficients are compared and split such that the coefficients $k_{ih} \in \mathbb{R}_+^*$ of this $\boldsymbol{x}^{\boldsymbol{\alpha}_h}$ are equals or independent over the integers. Here, m represents the number of distinct $\boldsymbol{\alpha}_h$, n is the number of equations, $\boldsymbol{B}_h = (B_{1h}, ..., B_{nh})^\mathsf{T}$, $\boldsymbol{\alpha}_h = (\alpha_{h1}, ..., \alpha_{hn})$. One can note that r in (2.3) is greater than m, and s_{ih} are in fact coefficients of $\boldsymbol{S}$ for a suitable index.

It is generally requested that ϕ does not depend on $\boldsymbol{k}$, allowing to consider it despite the lack of precision on the model parameters. In most of the cases, conservation laws are linear and can be found as the left kernel of the stoichiometric matrix. Indeed, any vector in the left kernel of $\boldsymbol{S}$ represents a linear conservation law, independent of $\boldsymbol{k}$.

Example 2.7. The stoichiometric matrix of the Michaelis-Menten model is the following :

$$\begin{pmatrix} -1 & 1 & 0 \\ 1 & -1 & -1 \\ -1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

This allow us to find two conservation laws:

$$c_1 = x_2 + x_3, \text{ and } c_2 = x_1 + x_2 + x_4 \tag{2.4}$$

In biochemical systems, linear conservation laws with positive coefficients can be interpreted as cycles of transformations of a molecule, induced for example by multiple phosphorylations and dephosphorylations. When a linear conservation law have both negative and positive coefficients, the species with negative coefficients increase or decrease at the same time as the species with positive coefficients. It could be understand as two cycles of transformations using the same set of reactions as in the following example

$$\emptyset \to x_1 + x_2 \to x_1 + x_3 \to \emptyset$$

which has one linear conservation laws, $x_1 - x_2 - x_3$. However, these conservation laws are often the results of a difference between positive conservation laws as in the following

Input: An ODE system whose r.h.s. are sums of rational monomials in $\boldsymbol{k}$ and $\boldsymbol{x}$, homogeneous of degree one in $\boldsymbol{k}$, with integer coefficients:

$$\dot{x}_i = f_i(\boldsymbol{k}, \boldsymbol{x}) = \sum_{j=1}^{r_i} z_{ij} k_{ij} \boldsymbol{x}^{\boldsymbol{\alpha}_{ij}}$$

where $z_{ij} \in \mathbb{Z}$, $\boldsymbol{\alpha}_{ij} \in \mathbb{Z}^n$, $k_{ij} \in \{k_1, \ldots, k_r\}$, for $1 \leq i \leq n$, $1 \leq j \leq r_i$.

Output: An integer coefficient matrix $\boldsymbol{S}$.

1: Compute a list of distinct monomials

$$\{M_j = k_j \boldsymbol{x}^{\boldsymbol{\alpha}_j}, 1 \leq j \leq r\} = \bigcup_{i=1}^{n} \bigcup_{j=1}^{r_i} \{k_{ij} \boldsymbol{x}^{\boldsymbol{\alpha}_{ij}}\}.$$

Two monomials $k_{ij}\boldsymbol{x}^{\boldsymbol{\alpha}_{ij}}$, $k_{i'j'}\boldsymbol{x}^{\boldsymbol{\alpha}_{i'j'}}$ are distinct if $\boldsymbol{\alpha}_{ij} \neq \boldsymbol{\alpha}_{i'j'}$ or $k_{ij} \neq k_{i'j'}$.

```
2: for i := 1 to n do
3:    for j := 1 to r do
4:       S_ij = 0
5:       for l := 1 to r_i do
6:          if k_il x^α_il = M_j then
7:             S_ij = S_ij + z_il
8:          end if
9:       end for
10:   end for
11: end for
```

Algorithm 2: Smatrix

example, which is a modification of the previous one

$$x_1 \to x_2 + x_3 \to x_2 + x_4 \to x_5$$

which has two linear conservation laws, $x_2 - x_3 - x_4$ and $2x_1 + x_2 + x_3 + x_4 + 2x_5$. But the conservation law $x_2 - x_3 - x_4$ is the difference between the conservation law $2x_1 + x_2 + x_3 + x_4 + 2x_5$ and the conservation law $2x_1 + 2x_2 + 2x_5$.

Conservation laws play an important role in modelling as they indicate a constraint on the dynamics and have often a interpretation. For example, the Noether's theorem [Noether, 1918] associates conservation laws to symmetries. Also, a model with a linear conservation law doesn't have a Jacobian matrix of full rank, and adding the conservation law to the system completes it. Finally, conservation laws can help to find steady states of a model by reducing the search space.

In order to analyse a model, it is important to know what kind of behavior it has. An angle of attack to do this is to study the steady states of the model.

Definition 2.8. A steady state of an ODE system $E : \dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{k}, \boldsymbol{x})$ is a solution of the equation $\dot{\boldsymbol{x}} = 0$.

A system is said monostationary if this equation admits only one solution. If this equation admit more than one solution, the system is said multistationary.

One can note that these properties depend on the parameters, as the following example shows [Conradi et al., 2017]:

Example 2.9. The following mass action CRN

$$\begin{gathered} HK_{00} \xrightarrow{\kappa_1} HK_{p0} \xrightarrow{\kappa_2} HK_{0p} \xrightarrow{\kappa_3} HK_{pp} \\ HK_{0p} + RR \xrightarrow{\kappa_4} HK_{00} + RR_p \\ HK_{pp} + RR \xrightarrow{\kappa_5} HK_{p0} + RR_p \\ RR_p \xrightarrow{\kappa_6} RR \end{gathered}$$

is multistationarity when $\kappa_3 > \kappa_1$ and monostationarity when $\kappa_3 \leq \kappa_1$.

When $\kappa_3 = \kappa_1$, we have a bifurcation: the behavior of the system changes. A common bifurcation is the saddle-node bifurcation that the we can found in the hysteresis loop, illustrated by the following figure 2.1.

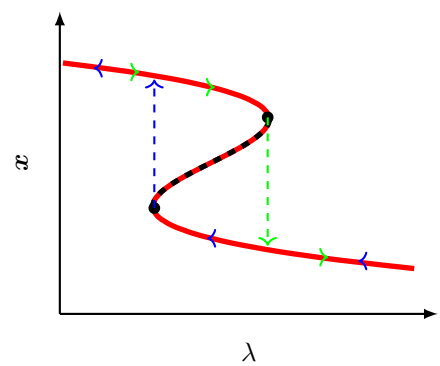


Figure 2.1: The red line correspond to the steady states w.r.t the parameter λ. The two black dots are the saddle-node bifurcation. The dashed part of the red curve denotes the unstable steady-states. Blue and green arrows denote the steady-state that we obtain when we slowly move the parameter.

As finding multistationarity is a important problem in mathematical biology, several results exist in this direction, such as the deficiency zero theorem [Feinberg, 1987], deficiency one theorem [Feinberg, 1987,Feinberg, 1988], homotopy continuation methods [Sommese and Wampler, 2005], and real triangularization and cylindrical algebraic decomposition methods [Bradford et al., 2020].

One can note that some models do not admit any steady state. In this case, we are generally searching for a limit cycle if there is any.

In the chemical reaction networks theory, the deficiency is an important concept, widely used, that made links between dynamic properties of a system and structure of the model. Deficiency is used in the deficiency theorems to prove the existence and uniqueness of steady states, but also to test absolute concentration robustness [Shinar and Feinberg, 2010]. So, we give here an overview of the concept.

Definition 2.10. A linkage class is a connected component of the CRN. A strong linkage class is a strongly connected component (*ie.* there exist a pathway between each couple of complexes in the strong linkage class). A terminal strong linkage class T is a strong linkage class in which there is no reaction starting from a node (*i.e.* a complex) of T and finishing in a node in another strong linkage class. Nodes belonging to terminal strong linkage classes are called terminal, whereas the other are called non-terminal.

The deficiency of a CRN is $\delta = m - l - s$, where

- m is the number of complexes,
- l is the number linkage classes,
- s is the dimension of the stoichiometric space ($s = rank(\boldsymbol{S})$).

Example 2.11. The deficiency of the Michaelis-Menten model is $3 - 1 - 2 = 0$. Strong linkage classes of the Michaelis-Menten models are $S + E \underset{k_2}{\overset{k_1}{\rightleftharpoons}} ES$ and $E + P$. $E + P$ is terminal.
The deficiency of the model in example 2.9 is $10 - 4 - 4 = 2$. All complexes are a strong linkage class.
It should be noted that an isolated strong linkage class is terminal, so we have a model $x_1 \rightleftharpoons x_2$, the whole model is a terminal strong linkage class.

Definition 2.12. We say that two nodes differ only in species S if the differences between the two nodes is a nonzero multiple of a single species S.

The complexes $A + B$ and B differ only in species A.

Definition 2.13. A biological system shows absolute concentration robustness (ACR) for an active molecular species if the concentration of that species is identical in every positive steady state the system might admit.

Theorem 2.14 ([Shinar and Feinberg, 2010])**.** *Consider a mass-action system that admits a positive steady state and suppose that the deficiency of the underlying network is one. If, in the network, there are two non-terminal nodes that differ only in species S, then the system has absolute concentration robustness in S.*

Chapter 3

Interval methods for Homeostasis and multistationarity

3.1 Introduction

In the 19th century, Claude Bernard introduced the concept of homeostasis that plays a crucial role in understanding the functioning of living organisms. He distinguished the "milieu intérieur" from the "milieu extérieur" [Bernard, 1879] and formulated four conditions that need to be roughly constant in the "milieu intérieur": humidity, aeration, heat, and chemical constitution. One of the most classical example of homeostasis is the body temperature that remains roughly constant despite large changes of external temperature. In chemical reaction networks, constancy of calcium or iron concentrations are other examples of homeostasis.

More recently, Martin Golubitsky and Ian Stewart define in their paper [Golubitsky and Stewart, 2017] the concept of homeostasis as follows: *"Homeostasis occurs in a biological or chemical system when some output variable remains approximately constant as one or several input parameters λ varies over some intervals"*. Using input-output functions, they propose the definition of infinitesimal homeostasis, based on the singularity theory. In their definition, the zero derivative of the input-output function is a locus of infinitesimal homeostasis and can be used to find homeostasis in gene regulatory networks [Antoneli et al., 2018].

Several other concepts such as robustness, resilience or viability are closely related to homeostasis and sometimes used with overlapping meaning. Robustness refers to the lack of sensitivity of temporal and static properties of systems with respect to parameters and/or initial conditions variation, thus encompassing homeostasis [Gorban and Radulescu, 2007, Rizk et al., 2009, Barr et al., 2019]. Resilience or viability has a more global, dynamical significance, meaning the capacity of systems to recover from perturbations via transient states that stay within bounds [Aubin, 2009].

The absolute concentration robustness (ACR) [Shinar and Feinberg, 2010] is another way to study homeostasis. It is defined by the following statement: *"A biological system shows ACR for an active molecular species if the concentration of that species is identical in every*

positive steady state the system might admit." This means that the initial conditions of the system do not impact the studied concentration species when the steady state is reached.

Using input-output function as well, we will define the k_{hom}-homeostasis, an interval point of view of homeostasis that include ACR, and present an algorithm dedicated to this problem. Modern interval arithmetic have been developed by Moore [Moore, 1966] and is used to cope with error analysis and parameter uncertainty [Tucker et al., 2007, Markov, 2010]. Eldon Hansen provided a contribution to global optimisation [Hansen and Walster, 2003, Hansen, 1979, Hansen, 1980] using interval arithmetic. Then, the Branch and Bound algorithm, used for discrete and combinatorial optimisation problems, have been generalised using interval arithmetic to deal with continuous values. Today, there are several efficient solvers and optimizers based on interval arithmetic [Araya et al., 2014, Tawarmalani and Sahinidis, 2005b].

Another important problem in systems biology is multistationarity, which means that the system has multiple steady states and possible outputs, at constant parameters. Considerable effort has been devoted to its study, with a variety of methods: numerical, such as homotopy continuation [Sommese and Wampler, 2005], symbolic, such as real triangularization and cylindrical algebraic decomposition [Bradford et al., 2020], or topological [Feinberg, 1987, Feinberg, 1988, Conradi et al., 2017], such as deficiency zero theorem or deficiency one theorem that are more closely related to monostationarity. However, as discussed in [Bradford et al., 2020], numerical errors in homotopy based methods may lead to failure in the identification of the correct number of steady states, whereas symbolic methods have a double exponential complexity in the number of variables and parameters.

Using the same methodology as for homeostasis, we provide a novel method for testing the multistationarity of CRN. Interestingly, our approach emphasizes a duality relationship between homeostasis and multistationarity; the former means constant output at variable input, whereas the latter means multiple output at constant input.

Our interval point point of view of these two problems lead us to find an algorithmic approach using the Ibex library (Interval Based EXplorer), an open source C++ library including a solver `IbexSolve` and an optimizer `IbexOpt`. Using this library, we have developed an algorithm to find homeostatic species as well as multistationarity. However, if the same algorithm can solve both of these problems, it is generally more efficient to use directly the solver `IbexSolve` when testing multistationarity.

Based on the idea that finding k_{hom}-homeostasis is linked to finding the minimal box that contains all the solutions of the constrained problem, our algorithm `IbexHomeo` applies successive optimization problems and compares the bounds obtained. In parallel, it is the first (portfolio) distributed variant of the `IbexOpt` Branch and Bound optimizer, where several variants of the solver are run on different threads and exchange information.

As a future improvement to studying homeostasis, a dedicated algorithm that finds the minimal box containing all the solutions of the constrained problem is currently developed.

3.2 Settings and definitions

We consider a set of species variables $x_1, ..., x_n$, representing species concentrations, a set of parameters $k_1, ..., k_r$, representing kinetic constants, and a set of differential equations :

$$\frac{dx_1}{dt} = f_1(x_1, ..., x_n, k_1, ..., k_r), \ldots, \frac{dx_n}{dt} = f_n(x_1, ..., x_n, k_1, ..., k_r). \tag{3.1}$$

We are interested in systems that have steady states, i.e. such that the system

$$f_1(x_1, ..., x_n, k_1, ..., k_r) = 0, \ldots, f_n(x_1, ..., x_n, k_1, ..., k_r) = 0 \tag{3.2}$$

admits real solutions for fixed parameters $k_1, \ldots, k_r$. For biochemical models, $x_1, ..., x_n$ represent concentrations, and in this case we constrain our study to real positive solutions. Generally, it is possible to have one or several steady states, or no steady state at all. The number of steady states can change at bifurcations. For practically all biochemical models, the functions $f_1, \ldots, f_n$ are rational, and at fixed parameters (3.2) defines an algebraic variety. The local dimension of this variety is given by the rank defect of the Jacobian matrix $\boldsymbol{J}$, of elements $J_{i,j} = \frac{\partial f_i}{\partial x_j}, 1 \leq i, j \leq n$.When $\boldsymbol{J}$ has full rank, then by the implicit function theorem, the steady states are isolated points (zero dimensional variety) and all the species are locally expressible as functions of the parameters:

$$x_1 = \Phi_{x_1}(\boldsymbol{k}), \ldots, x_n = \Phi_{x_n}(\boldsymbol{k}). \tag{3.3}$$

The functions Φ_y were called input-output functions in [Golubitsky and Stewart, 2017], where the input is the parameters k_i and the output variable y is any of the variables $x_i, 1 \leq i \leq n$. Also in the full rank case, a system is called multistationary when, for fixed parameters there are multiple solutions of (3.2), i.e. multiple steady states.

$\boldsymbol{J}$ has not full rank in two cases. The first case is at bifurcations, when the system output changes qualitatively and there is no homeostasis. The second case is when (3.1) has $l \leq n$ independent first integrals (conservation laws), i.e. functions of x that are constant on any solution of the ODEs (3.1). In this case the Jacobian matrix has rank defect l everywhere and steady states form a l-dimensional variety. For instance, for many biochemical models, there is a full rank constant matrix C such that $\sum_{j=1}^{n} C_{ij} f_j(x_1, ..., x_n, k_1, ..., k_r) = 0$, for all $x_1, ..., x_n, k_1, ..., k_r, 1 \leq i \leq l$. In this case there are l linear conservation laws, i.e. $\sum_{i=1}^{n} C_{ij} x_j = c_i, 1 \leq i \leq l$ are constant on any solution. Here c_i depends only on the initial conditions, $c_i = \sum_{i=1}^{n} C_{ij} x_j(0)$. In biochemistry, linear conservation laws occur typically when certain molecules are only modified, or complexified, or translocated from one compartment to another one, but neither synthesized, nor degraded. The constant quantities c_i correspond to total amounts of such molecules, in various locations, in various complexes or with various modifications.

A biological system is characterized not only by its parameters but also by the initial conditions. For instance, in cellular biology, linear conservation laws represent total amounts of proteins of a given type and of their modifications, that are constant within a cell type, but may vary from one cell type to another. Therefore we are interested in the dependence of steady states on initial conditions,represented as values of linear conservation laws. Because conservation laws can couple many species, steady states are generically very sensitive to their values. ACR represents a remarkable exception when steady states do not depend on conservation laws. In order to compute steady states at fixed initial conditions, we solve the extended system

$$\begin{gathered} f_1(x_1, ..., x_n, k_1, ..., k_r) = 0, \ldots, f_{n-l}(x_1, ..., x_n, k_1, ..., k_r) = 0, \\ C_{11}x_1 + \ldots + C_{1n}x_n = c_1, \ldots, C_{l1}x_1 + \ldots + C_{ln}x_n = c_l, \end{gathered} \tag{3.4}$$

where c_i are considered as extra parameters, and $f_1, \ldots, f_{n-l}$ are linearly independent functions. In this case, excepting the degenerate steady states with zero concentrations discussed

at the end of this section, the Jacobian of the extended system has full rank and one can define again input-output functions as unique solutions of (3.4). A system is multistationary if at fixed parameters there are multiple solutions of (3.4).

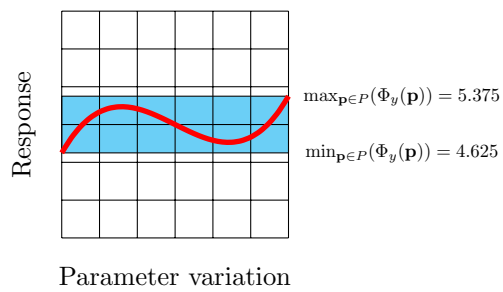
Homeostasis is defined using the input-output functions.

Definition 3.2.1. We say that y is a k_{hom}-homeostatic variable if in the path of steady states given by Φ_y we get :

$$\frac{\max_{\mathbf{p}\in P}(\Phi_y(\mathbf{p}))}{\min_{\mathbf{p}\in P}(\Phi_y(\mathbf{p}))} \leq k_{hom},$$

where $k_{hom} \geq 1$. We take $k_{hom} = 2$ by default, but we can choose a smaller k_{hom} for small parameter variation. P represents the space of parameters (we take a P compact for our examples), and $\mathbf{p}$ is a point inside P. So, we consider homeostasis of a variable y for any change of parameters in a domain P.

This is an illustration of homeostasis using interval arithmetic when $k_{hom} \geq 1.163$.



We exclude from our definition trivial solutions $x_i = 0$ obtained when

$$f_i(x_1, ..., x_n, k_1, ..., k_r) = x_i^{n_i} g_i(x_1, ..., x_n, k_1, ..., k_r),$$

where n_i are strictly positive integers, g_i are smooth functions with non-zero derivatives $\partial f_i/\partial x_i$ for $x_i = 0$. These solutions persist for all values of the parameters and are thus trivially robust. In this case we replace the problem $f_i = 0$ by the problem $g_i = 0$ that has only non-trivial solutions $x_i \neq 0$.

3.3 Constraint methods for interval arithmetic

Contrary to standard numerical analysis methods that work with single values, interval methods can manage sets of values enclosed in intervals. By these methods one can handle exhaustively the set of possible constraint systems solutions, with guarantees on the answer. Interval methods are therefore particularly useful for handling nonlinear, non-convex constraint systems.

Definition 3.3.1. An interval $[x_i] = [\underline{x_i}, \overline{x_i}]$ defines the set of reals x_i such that $\underline{x_i} \leq x_i \leq \overline{x_i}$. $\mathbb{IR}$ denotes the set of all intervals. A **box** $[x]$ denotes a Cartesian product of intervals $[x] = [x_1] \times ... \times [x_n]$. The size or width of a box $[x]$ is given by $w[x] = \max_i(w([x_i]))$ where $w([x_i]) = \overline{x_i} - \underline{x_i}$.

Interval arithmetic [Moore, 1966] has been defined to extend to $\mathbb{IR}$ the usual mathematical operators over $\mathbb{R}$ (such as $+$, $\cdot$, $/$, power, sqrt, exp, log, sin). For instance, the interval sum is defined by $[x_1]+[x_2] = [\underline{x_1}+\underline{x_2}, \overline{x_1}+\overline{x_2}]$. When a function f is a composition of elementary functions, an *extension* of f to intervals must be defined to ensure a conservative image computation.

Definition 3.3.2. (Extension of a function to $\mathbb{IR}$)
Consider a function $f : \mathbb{R}^n \to \mathbb{R}$.
$[f] : \mathbb{IR}^n \to \mathbb{IR}$ is said to be an **extension** of f to intervals iff:

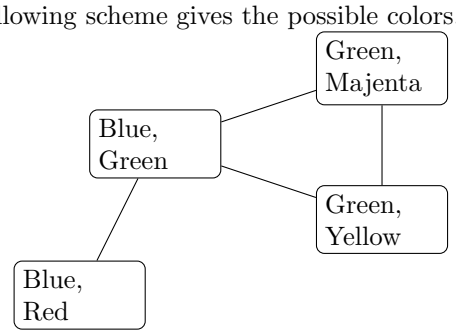
$$\begin{array}{ll} \forall [x] \in \mathbb{IR}^n & [f]([x]) \supseteq \{f(y),\ y \in [x]\} \\ \forall x \quad \in \mathbb{R}^n & f(x) = [f]([x,x]) \end{array}$$

The *natural extension* of a real function f corresponds to the mapping of f to intervals using interval arithmetic. More sophisticated interval extensions have been defined, based on interval Taylor forms or exploiting function monotonicity [Jaulin et al., 2001].

Example 3.1. It exists different extensions to function, for example we can think about the extension of the function x^2. If we define it as the product of the same interval $[x] \times [x]$, we will get a large enclosure of the real result if negative numbers are implied, since the result of the product is considered as $[\min(a \times b), \max(a \times b)], \forall a, b \in [x]$, then the result of this extension for the interval $[-2, 2]$ is $[-4, 4]$. Then, the arithmetic function $[x]^2$ currently used, that truncate the result below 0 or consider $[\min(a \times a), \max(a \times a)], \forall a \in [x]$, and gives as result $[0, 4]$ differs from $[x] \times [x]$.

A typical constraint scheme in combinatorial problems is the backtracking algorithm, the interval version of this algorithm is the bisect and evaluate algorithm. A combinatorial problem can be view as the tree of possibilities, each stage of the tree correspond to a variable, and the nodes of this stage is the possible values of the variable. A leaf of the tree represents a combination that can be proposed to solve the problem. For example, if there is four variables, each having three different possibilities, we obtain $3^4 = 81$ leafs. The backtracking algorithm explore the tree, descending the tree and checking if the combination is satisfying the constraints. If not, it back to the previous node, suppress the branch, and check another descent. If yes, it continues.

Example 3.2. Consider the following problem of coloration: France, Spain, Swiss, and Italy have to be colored such that two adjacent countries does not have the same color. The following scheme gives the possible colors.



This lead to the following tree in figure 3.1.

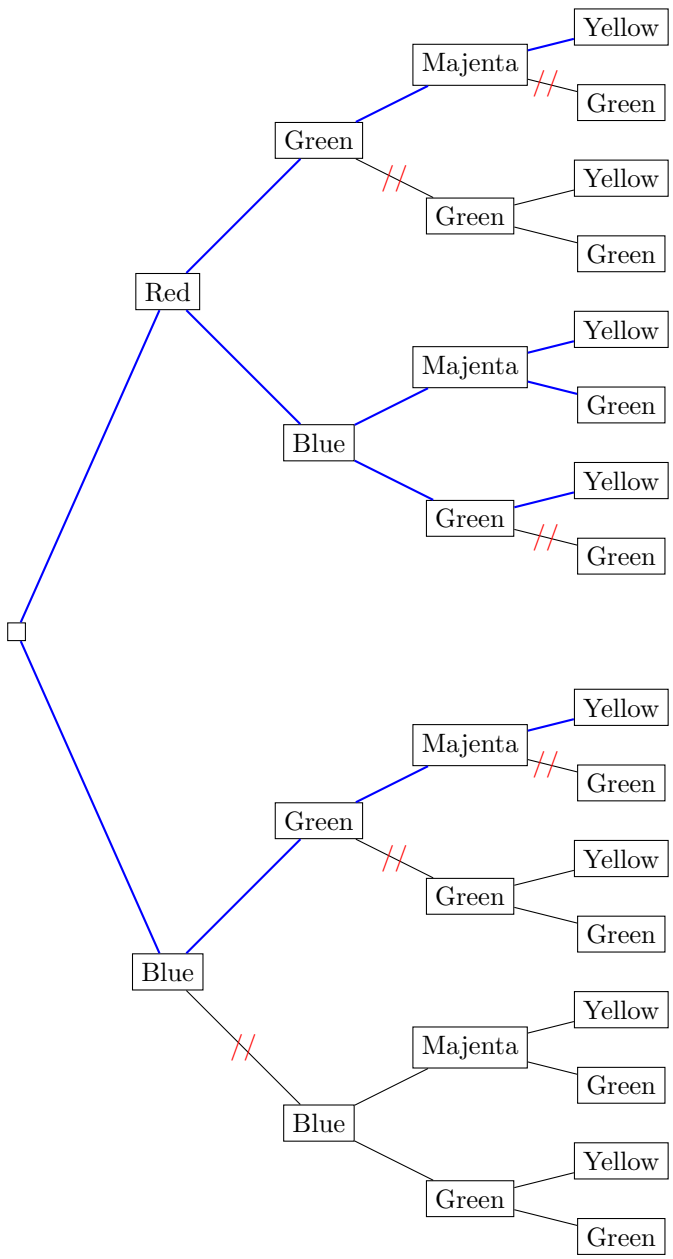


Figure 3.1: From the root: Spain, France, Swiss, Italy. Each pathway that doesn't encounter a // is a solution of the problem and is denoted with thick blue arrows. The symbol // represents a branch cutting by the algorithm.

The bisect and evaluate algorithm starts from an initial box and splits it before checking the constraints by evaluation. The initial box can be considered as the root of the search tree, and the nodes of the tree correspond to sub-boxes. If an evaluation of a sub-box leads to an impossibility to check the constraints (*e.g.* if we are searching for the roots of a polynomial and the evaluation is strictly positive like $[5, 10]$) then the box is removed from the search space. It stops the bisection when a precision is reached, and the box is considered as a possible solution.

Example 3.3. Consider that we are searching for $(xy-1, x-y) = (0, 0)$ in the box $[0.25, 4] \times [0.25, 4]$, then we get as a possible tree the following figure 3.2. In the initial box, the two intervals for the constraints are $[x].[y] - 1 = [0.25, 4].[0.25, 4] - 1 = [0.0625, 16] - 1 = [-0.9375, 15] \ni 0$ and $[0.25, 4] - [0.25, 4] = [-3.75, 16] \ni 0$. As the box is too big to be considered as solution, and as each constraint is satisfied, the box is split into two new boxes. Then one of the box is evaluate with the new values of $[x]$ and $[y]$. If a constraints is not satisfied, the box is removed from the list of boxes stored, and if the box is small enough it is saved as a solution.

Following the same kind of scheme, constraint algorithms are built to cut branches, involving algorithms that propagate the constraints [Bessiere, 2006], such arc consistency [Bessiere, 1991]. In interval constraint algorithm, these constraint propagators work as contractors [Chabert and Jaulin, 2009]. The `Ibex` library [Ninin, 2015, Chabert, 2020] (for Interval Based EXplorer) is a C++ library including a solver `IbexSolve` and an optimizer `IbexOpt`. The solver follows a Branch and Contract process that achieve two main operations:

- **Bisection**: The current box is split into two sub-boxes along one variable interval.
- **Contraction**: Both sub-boxes are handled by *contraction* algorithms that can remove sub-intervals without solution at the bounds of the boxes.

At the end of this tree search, the "small" boxes of size less than a user-given precision ϵ contain all the solutions to the equation system. The process is combinatorial, but the contraction methods are polynomial-time acceleration algorithms that make generally the approach tractable for small or medium-sized systems. Contraction methods are built upon interval arithmetic and can be divided into constraint programming (CP) [Van Hentenryck et al., 1997, Neveu et al., 2015, Benhamou et al., 1999a] and convexification [Tawarmalani and Sahinidis, 2005a, Misener and Floudas, 2014] algorithms.

We present two contractors, `HC4` [Benhamou et al., 1999b] and `ACID` [Neveu et al., 2015].

We say that a box B is hull-consistent for a constraint c if the box-hull of the box B intersected with the set ρ_c given by the constraint c is the box B itself, *ie.* B is hull-consistent if $B = hull_{\square}(\rho_c \cap B)$.

The goal of `HC4` is to obtain a box hull-consistent from the given box, for each constraint of the system. It is described by algorithms 3 and 4, and the figure 3.3.

`ACID` means Adaptive Constructive Interval Disjunction, this algorithm is a version of `3BCID` that is the state of the art in numerical constraint programming. `var3BCID`, the main procedure in `3BCID`, can be described as the following and is illustrated in figure 3.4:

- `var3B`: It shaves the box with respect to one variable x_i, by taking iteratively sub-boxes of width $w(x_i)/s_{3B}$ on the bounds and contract them, and stop when the sub-boxes (left and right) are not excluded.

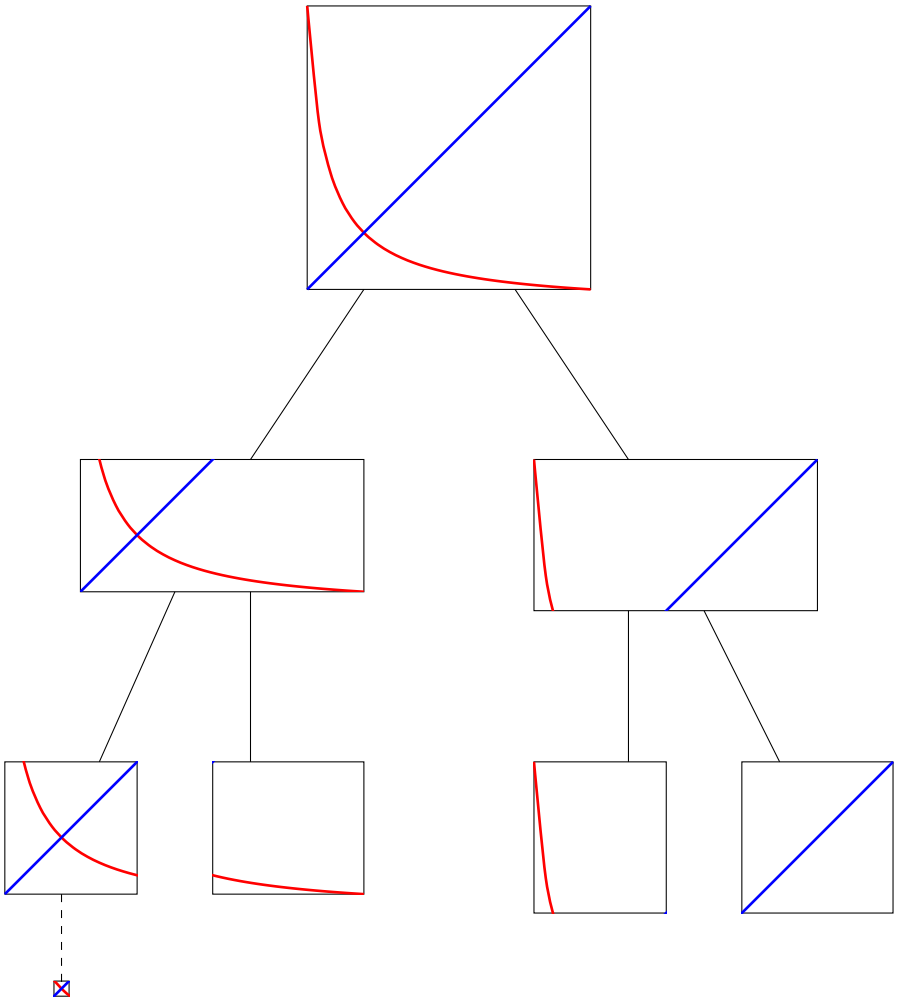


Figure 3.2: The tree search. The dashed line correspond to the right pathway to get the solution. Each other terminal node violates a constraint, as illustrated by the absence of one of the colored curve, and are then removed from the list of possible solutions.

```
Algorithm HC4(c_1, ..., c_m, B = I_1 × ... × I_n)
    S ← {c_1, ..., c_m}
    while S ≠ 0 and B ≠ 0 do
        c ← c_i ∈ S /*we select one constraint*/
        B' ← HC4Revise(c, B)
        if B' ≠ B then
            S ← S ∪ {c_j | ∃x_k ∈ var(c_j) and I'_k ≠ I_k}
            B ← B'
        else
            S ← S\{c}
    return B
```

Algorithm 3: `HC4` contract the box B w.r.t each constraint. `HC4Revise` is the contraction of B for the constraint c. If the contraction is inefficient, we remove c from the set of constraint considered. If the contraction is efficient, we retrieve in S each constraint for which a variable contracted is involved (this imply c itself), since an inefficient contraction at a moment can be efficient later.

```
Algorithm HC4Revise(c, B = I_1 × ... × I_n)
    D ← B
    ForwardEvaluation(c, D)
    BackwardPropagation(c, D)
    B ← hull_□(D)
    return B
```

Algorithm 4: `HC4Revise` follows a forward backward scheme. The constraint c is viewed as a tree t_c, each leaf representing a variable or constant, and other nodes representing operations, the root being the relation. The forward evaluation follows the natural extension of each side of c, and gives two intervals I_l resulting from the l.h.s. of c, and I_r resulting from the r.h.s. of c. The backward propagation consider the intersection of I_r and I_l (if equality, else it considers the intersection of these intervals extended to infinity on one border). Then, descending the tree, for an operation $op(x_{old}, y_{old})$, x_{new} (resp. y_{new}) is computed as the minimal interval leading to $I_{new} = op(x_{new}, y_{old})$ (resp. $I_{new} = op(x_{old}, y_{new})$).

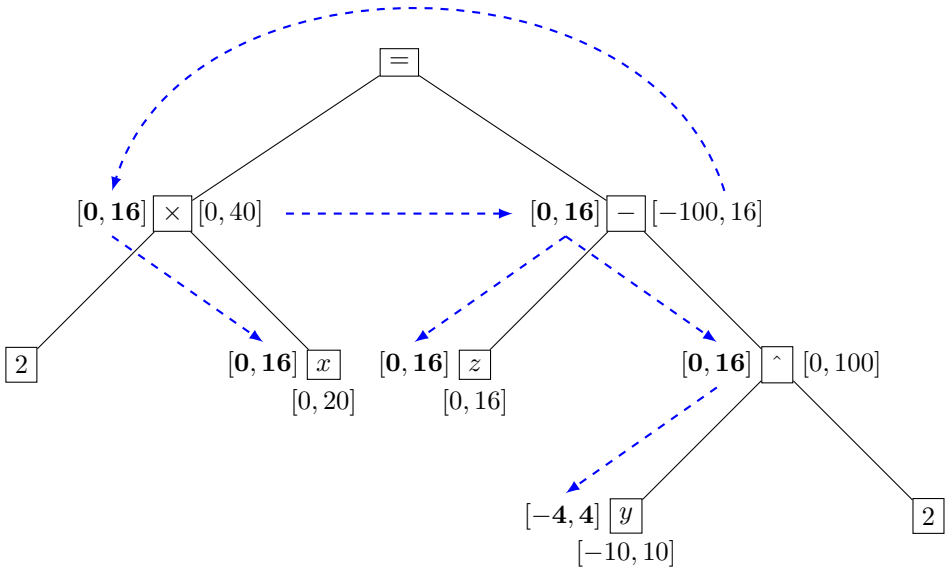


Figure 3.3: An example of `HC4Revise` for the constraint $2x = z - y^2$. Intervals in bold are results of the backward phase. The other ones are the result of the forward (natural extension) phase.

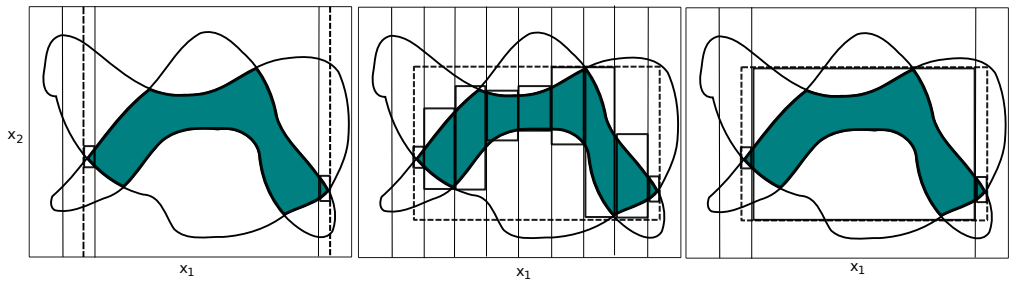


Figure 3.4: The three steps of `var3BCID`.

- `varCID`: The middle box is then subdivised in s_{cid} boxes by subdivising the interval $[x_i]$ s_{cid} times, then each sub-box is contracted.
- `var3BCID`: At the end, we take the hull of each remaining boxes.

In experiments, $s_{3B} \in \{5, ..., 20\}$ and $s_{cid} \in \{1, 2\}$, meaning that the shaving is more used than the global contraction given the CID procedure, which has an important cost but can contract other variables.

In `3BCID`, `var3BCID` is called in two nested loops. An inner loop that calls `var3BCID` for each variable x_i, and an outer loop that calls the inner loop until we reach a kind of fixed point, meaning that no interval is contracted. The main improvement of `ACID` is to avoid these expensive loops thanks to a learning phase, avoiding dynamically the contraction of some variables, in a single loop "while", by measuring a contraction ratio.

Returning to a general overview, the optimizer `IbexOpt` follows a Branch and Bound strategy that generalises the Branch and Contract process. Constrained global optimization consists in finding a vector in the domain that satisfies the constraints while minimizing an *objective function*.

Definition 3.3.3. Let $x = (x_1, ..., x_n)$ varying in a box $[x]$, and functions $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^n \to \mathbb{R}^m$, $h : \mathbb{R}^n \to \mathbb{R}^p$.

Given the system $S = (f, g, h, x, [x])$, the constrained global optimization problem consists in finding f^* :

$$f^* \equiv \min_{x \in [x]} f(x) \text{ subject to } g(x) \leq 0 \text{ and } h(x) = 0.$$

f denotes the objective function, f^* being the objective function value (or best "cost"), g and h are inequality and equality constraints respectively. x is said to be feasible if it satisfies the constraints.

Example 3.4. In the case of homeostasis, the box corresponds to $[\boldsymbol{k}, \boldsymbol{c}, \boldsymbol{x}]$, the constraints correspond to $h_1(\boldsymbol{k}, \boldsymbol{c}, \boldsymbol{x}) = \dot{\boldsymbol{x}} = 0$ to which we add the conservation laws $h_2(\boldsymbol{k}, \boldsymbol{c}, \boldsymbol{x}) = (c_j - \phi_j(\boldsymbol{k}, \boldsymbol{x}))_j = 0$, then $h(\boldsymbol{k}, \boldsymbol{c}, \boldsymbol{x}) = (h_1(\boldsymbol{k}, \boldsymbol{c}, \boldsymbol{x}), h_2(\boldsymbol{k}, \boldsymbol{c}, \boldsymbol{x}))$ and $g(\boldsymbol{k}, \boldsymbol{c}, \boldsymbol{x}) = 0$. The objective function is to minimize x_i or $-x_i$. Then we compare the two objective functions to get the result.

The Branch and Bound solver maintains two bounds lb and ub of f^*. The upper bound ub of f^* is the best (lowest) value of $f(x)$ satisfying the constraints found so far, and the lower bound lb of f^* is the highest value under which it does not exist any solution (feasible point). The strategy terminates when $ub - lb$ (or a relative distance) reaches a user-defined precision ϵ_f. To do so, a variable x_{obj} representing the objective function value and a constraint $x_{obj} = f(x)$ are first added to the system. Then a tree search is run that calls at each node a bisection procedure, a contraction procedure, but also an additional bounding procedure that aims at decreasing ub and increasing lb. Improving lb can be performed by the contraction procedure: it is given by the minimum value of $\underline{x_{obj}}$ over all the nodes in the search tree. Improving the upper bound is generally achieved by local numerical methods. Like any other Branch and Bound method, improving the upper bound ub allows the strategy to eliminate nodes of the tree for which $ub < \underline{x_{obj}}$.

Remark 3.5. Interval Branch and Bound codes can solve the optimization problem defined in Def. 3.3.3, but they sometimes require a significant CPU time because of the guarantee on the equality constraints. A way to better tackle the problem in practice is to relax equalities $h(\boldsymbol{k}, \boldsymbol{c}, \boldsymbol{x}) = 0$ by pairs of inequalities $-\epsilon_h \leq h(\boldsymbol{k}, \boldsymbol{c}, \boldsymbol{x})$ and $h(\boldsymbol{k}, \boldsymbol{c}, \boldsymbol{x}) \leq +\epsilon_h$, where ϵ_h is a user-defined positive parameter. Therefore, in practice, interval Branch and Bound codes generally compute a feasible vector $\boldsymbol{v} = (\boldsymbol{k}\ \boldsymbol{c}\ \boldsymbol{x})$ satisfying the constraints $g(\boldsymbol{k}, \boldsymbol{c}, \boldsymbol{x}) \leq 0$ and $-\epsilon_h \leq h(\boldsymbol{k}, \boldsymbol{c}, \boldsymbol{x}) \leq +\epsilon_h$ such that $|f^* - f(\boldsymbol{k}, \boldsymbol{c}, \boldsymbol{x})| \leq \epsilon_f$.

3.4 Using interval methods to find homeostasis

Using the definition 3.2.1, we can think of the homeostasis problem as to finding the minimal box containing all the steady-states of the system, or in constraint language, all the feasible solutions of the constrained system. Now, we can easily think of three ways to solve the problem.

The first one is to use the solver `IbexSolve`, get all the steady-states and make the hull of these solutions. However, this solution have two main problems. The first one is that we need to find, and store, each solution, that can easily lead to a grow in memory. The second one is that the solver doesn't cope very well with high complexity since there is no major cut in the searching tree. It will not be developed furthermore here.

The second solution is to use the optimizer `IbexOpt` to find each bound of the box. This solution has been tested with some modifications and seems to be useful for small and medium size models.

The third solution is to make a dedicated algorithm for this problem, which has several advantages. First, an algorithm dedicated to the problem of finding a minimal box containing all the solutions of a problem has a broader interest and can be used for for others problems than homeostasis. Second, such an algorithm will probably be more performing than the previous solutions, at least if we use the same kind of modifications.

3.5 IbexHomeo, a multiple bi-optimization process

Since we want to compute the minimum and the maximum value of $x_i = \Phi_{x_i}(\mathbf{p})$, the homeostasis detection amounts to two optimization problems, one minimizing the simple objective function x_i, and one maximizing x_i, i.e. minimizing $-x_i$. The two values returned

are finally compared to decide the x_i homeostasis. It is useful to consider that minimizing and maximizing x_i are somehow symmetric, allowing the strategy to transmit bounds of x_i from one optimization process to the dual one. These bounds can also be compared during optimization to stop both optimizations if they give enough information about homeostasis. Indeed, an optimizer minimizing x_i computes $[ln, un] \ni \min(x_i)$, where ln and un are lb and ub of the objective function x_i. An optimizer maximizing x_i computes $[ux, lx] \ni \max(x_i)$, where ux and lx are $-ub$ and $-lb$ of the objective function $-x_i$. As lx and ln are bounds for which there is no solution above and below respectively, lx/ln is an overestimate of the "distance" between any two feasible values of x_i, and a small value states that the species is homeostatic (see Def. 3.2.1). Conversely, ux/un is an underestimate of any two feasible values distance, and $ux/un > k_{hom}$ asserts that the species is not homeostatic. This `TestHom` decision procedure is implemented by Algorithm 5.

```
Algorithm TestHom(un, ln, ux, lx, k_hom)
    if lx/ln ≤ k_hom then
        return 2 /* homeostatic variable */
    if ux/un > k_hom then
        return 1 /* non homeostatic variable */
    else
        return 0 /* not enough information */
```

Algorithm 5: The `TestHom` decision procedure.

The bi-optimization described above runs on the system S corresponding to the system (3.4), where the equations $f_j(\boldsymbol{k}, \boldsymbol{c}, \boldsymbol{x}) = 0$ are relaxed by inequalities $-\epsilon_h \leq f_j(\boldsymbol{k}, \boldsymbol{c}, \boldsymbol{x}) \leq +\epsilon_h$; the parameters $\boldsymbol{k}$ and $\boldsymbol{c}$ can vary in a box $[\boldsymbol{p}]$ and are added to the set of processed variables. In our problem, it is important to notice that a steady state is expected for *every* parameter vector $\boldsymbol{p} \in [\boldsymbol{p}]$ (this is not valid, for instance, in the neighborhood of a saddle-node bifurcation, which should be avoided by re-defining $[\boldsymbol{p}]$). We exploit this key point by also running minimization and maximization of x_i on a system S', corresponding to the system S where the parameters have been fixed to a random value $\boldsymbol{p} \in [\boldsymbol{p}]$, with the hope that reducing the parameter space allows a faster optimization. Indeed, if the size of a problem is often considered by the number of variables, but the number of parameters is generally more important, and our problem need to cope with this dimensional explosion, and the computed values for fixed parameters constitute feasible points for the initial problem (i.e., with parameters that can vary) and can fasten the bi-optimization algorithm described above. Recall indeed that finding feasible points enables to improve the upper bound ub of f^* and to remove from the search tree the nodes with a greater cost.

Overall, homeostasis detection of species x_i is performed by Algorithm 6.

All the optimization processes are run in parallel and exchange newly found feasible points stored in FP. Every call to `Minimize` on S can start with an initial upper bound initialized with the best feasible point found so far ($min_{x_i}(FP)$ or $max_{x_i}(FP)$).

The minimization processes on S' are generally fast so that several ones can be called in a loop (with different parameters fixed to random values) until the end of the main minimization processes on S.

It is important to understand that `IbexSolve` and `IbexOpt` are generic strategies. That

Algorithm `Bi-Optimize`$(x_i, t, S = (\boldsymbol{x} \times \boldsymbol{p}, [\boldsymbol{x}] \times [\boldsymbol{p}],$ *system* (3.4)$), \epsilon_h),$
$P = (\epsilon_f, t)$,$FP)$
  Execute in parallel until timeout t:
  $(un, ln, FP) \leftarrow$ `Minimize`$(x_i, S, \min_{x_i}(FP), P)$
  $(ux, lx, FP) \leftarrow$ `Minimize`$(-x_i, S, \max_{x_i}(FP), P)$
  while *true* **do**
    $S' \leftarrow$ `FixRandomParameters`(S)
    $FP \leftarrow$ `Minimize`$(x_i, S', +\infty, P)$
  return (un, ln, ux, lx, FP)

Algorithm 6: The double optimization process on a given species x_i. P is the set of solver parameters: ϵ_f is the user-defined precision on the objective function value, t is the timeout required.

is, different procedures can be selected for carrying out the choice of the next variable interval to bisect (called branching heuristic) or for selecting the next node to handle in the search tree. It is known that some heuristics in general useful can be sometimes bad for some specific problems, and it was observed in our preliminaries experiments that it was regularly the case in our tested models. Therefore we propose a *portfolio* parallelization strategy where different processes (threads) run Branch and Bound algorithms using different branching heuristics (called *cutters* hereafter) or node selection heuristics (called *nodeSel*). This choice is probably not CPU efficient, but can lead to a human gain time, when some problems are locked for hours in a set of heuristics and only a couple of second for another one. These threads can communicate their bounds to each other, reducing the risks of an ineffective strategy.

The branching heuristics used in the different threads are all the variants of the *smear* branching strategy described in [Trombettoni et al., 2011a] and [Araya and Neveu, 2018]. As explained before, the natural extension of a function f corresponds to the mapping of f to intervals. We can define the *smear* function, that reflects an impact of x_i on f_j as:

$$smear(x_i, f_j) = \left| \left[\frac{\partial f_j}{\partial x_i} \right]_N ([\boldsymbol{x}]) \right| w([x_i]).$$

Then, we can define:

$$smearSum(x_i) = \sum_{f_j} smear(x_i, f_j),$$

$$smearMax(x_i) = \max_{f_j}(smear(x_i, f_j)).$$

To get a relative impact, a variant of *smear* is used, called *smearRel*:

$$smearRel(x_i, f_j) = \frac{smear(x_i, f_j)}{\sum_{x_k \in x} smear(x_k, f_j)}.$$

Then, we can define:

$$smearSumRel(x_i) = \sum_{f_j} smearRel(x_i, f_j),$$

$$smearMaxRel(x_i) = \max_{f_j}(smearRel(x_i, f_j)).$$

The *lsmear* [Araya and Neveu, 2018] is a more complex strategy that use the Lagrangian of the linearized constraints if an optimum (for the linearized constraints) is found using a simplex method, else it use the *smearSum* strategy.

Strategies used to select the next node to be handled are described in [Neveu et al., 2016]. The cutting strategy *lsmear* is generally more efficient than the others, and will be more often used.

In practice, we should modify a call to `Minimize` as follows:

$$\texttt{Minimize}(x_i, S, P, cutters, nodeSel)$$

where *cutters* denotes a set of branching heuristics and *nodeSel* denotes a set of node selection heuristics. This routine calls $|cutters| \times |nodeSel|$ threads, each of them corresponding to one Branch and Bound using one branching heuristic in *cutters* and one node selection heuristic in *nodeSel*. These threads work in the same time on the same problem, but they build different search trees. Therefore one optimizer can compute an *lb* value better (greater) than the others. In this case, it sends it to the other threads.

Finally, because we want to determine all the homeostatic species, we run the double optimization n times, for every species x_i, as shown in Algorithm 7. After a first call to a `FirstContraction` procedure that contracts the domain $[\boldsymbol{x}] \times [\boldsymbol{p}]$, `IbexHomeo` calls two successive similar loops of different performance. The first loop iterates on every species x_i and calls on it the double optimization function `Bi-Optimize`. The optimization threads are all run using the *lsmear* branching heuristic and have a "short" timeout in order to not be blocked by a given species computation. If a bi-Optimization call on x_i reaches the timeout t without enough information about homeostasis, x_i is stored in L and the computation on subsequent species continues and can learn (and store in FP) new feasible points than can be exploited by other optimization processes. Indeed, the feasible region defined by S is the same for each optimization. Therefore the second loop is similar to the first one, but with a greater timeout and more threads in parallel running the optimization with more various branching heuristics.

To summarize, the `IbexHomeo` algorithm creates communicating threads for:

- exploiting the duality min/max of the bi-optimization related to a given species homeostasis detection,
- finding feasible points more easily using the existence of a solution for all parameters,
- running a portfolio of similar Branch and Bound algorithms using different heuristics.

3.6 Multistationarity

`IbexSolve` can find all the solutions of (3.4) with fixed parameters in a straightforward way.

This method is useful for small and medium systems, and sometimes for large systems, depending on the nature of the constraints and the efficiency of the contractors. Also, it provides as output each solution box. This output is easy to read, because (3.4) has always a finite set of solutions.

Algorithm `IbexHomeo`$(S = (\boldsymbol{x} \times \boldsymbol{p},\ [\boldsymbol{x}] \times [\boldsymbol{p}],\ system\ (3.4),\ \epsilon_h),\ P = (\epsilon_f, t, k_{hom}))$

```
cutters ← {lsmear}
nodeSel ← {double_heap, cell_beam_search}
[x] ← FirstContraction([x], S)
FP ← ∅, L ← ∅
foreach x_i ∈ x do
    (un, ln, ux, lx, FP) ← Bi-Optimize(x_i, t, S, P, FP, cutters, nodeSel)
    [x_i] ← [un, ux]
    if timeout(t) and TestHom(un, ln, ux, lx, k_hom)=0 then
        L ← L ∪ x_i
t ← 10 t
cutters ← {lsmear, smearSum, smearSumRel, smearMax, smearMaxRel}
foreach x_i ∈ L do
    (un, ln, ux, lx, FP) ← Bi-Optimize(x_i, t, S, P, FP, cutters, nodeSel)
    [x_i] ← [un, ux]
return HomeostaticSpecies([x], x, k_hom)
```

Algorithm 7: Main frame of IbexHomeo. $k_{hom} \in [1, 2]$ is defined in Def. 3.2.1. Via the procedure `HomeostaticSpecies`, the algorithm returns the set of homeostatic variables.

In the case of large systems, it can be easier to answer the question: do we have zero, one, or several steady states? In this case, we can use the strategy used for homeostasis, with fixed parameters, where the problem is reformulated in terms of $2n$ constrained global optimization problems: for every variable x_i, we call twice an optimization code that searches for the minimum and the maximum value of x_i while respecting the system (3.4).

- If the system admits at least two distinct solutions, the criterion used in Definition 3.2.1 (using k_{hom} close to 1) will fail for at least one species, *i.e.* we will find a species x_i whose minimum and maximum values are not close to each other.
- If the system admits no solution, the first call to the optimizer (*i.e.*, minimizing x_1) will assert it.
- And if we have only one solution, every species will respect the criterion.

Let us give a simple example given by the model 233 in the Biomodels database [Le Novere et al., 2006]. In this model we have two species x and y together with seven parameters (one for the volume of the compartment, four for kinetic rates, and two for assumed fixed species). The system of ODEs is given by:

$$\frac{dx}{dt} = \frac{2k_2k_6y - k_3x^2 - k_4xy - k_5x}{k_1}, \qquad \frac{dy}{dt} = \frac{-k_2k_6y + k_3x^2}{k_1}. \tag{3.5}$$

After replacing the symbolic parameters by their given values, the steady state equations read:

$$16y - x^2 - xy - \frac{3}{2}x = 0, \qquad -8y + x^2 = 0. \tag{3.6}$$

The system (3.6) has two non-zero solutions, given by (6,4.5) and (2,0.5). When the system (3.6) is tested by `IbexHomeo` (the dedicated strategy for homeostasis) on a strictly positive box (to avoid the trivial solution (0,0)), we find $x \in [2, 6]$ and $y \in [0.5, 4.5]$. The homeostasy criterion fails at fixed parameters and we know that we have multistationarity.

3.7 Experimental Results

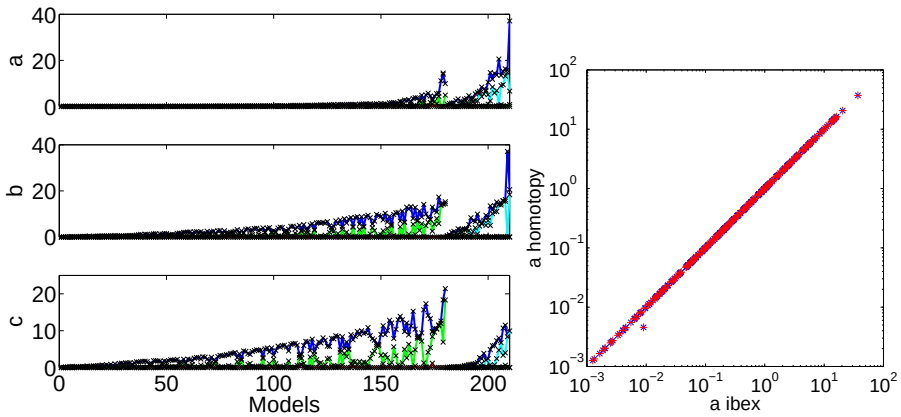
3.7.1 Multistationarity

For benchmarking the multistationarity test we have used DOCSS (Database of Chemical Stability Space, `http://docss.ncbs.res.in`), a repository of multistationary biochemical circuits. DOCSS contains biochemical circuits with up to four species and up to five catalytic reactions. The catalytic reactions are decomposed into several mass action laws, elementary steps. In DOCSS, the models are specified as short strings of symbols coding for the catalytic reactions and as lists of numeric parameters. These specifications were first parsed to SBML files, then to systems of differential equations and conservation laws using tools developed in [Lüders et al., 2020], and transformed into an input file for our algorithms. For the benchmarking we have selected all the 210 DOCSS circuits with 3 species (denoted a,b,c) and 3 catalytic reactions. They correspond to 13 different symbolic systems of equations (the remaining differences concern numerical parameters). The mass action models have up to 6 variables (*i.e.*, the species a,b,c, and several complexes resulting from the decomposition of catalytic reactions into mass action steps). The steady states of all models in DOCSS were numerically computed in [Ramakrishnan and Bhalla, 2008] using a homotopy continuation method [Sommese and Wampler, 2005]. For all the 3×3 models both homotopy and interval `IbexSolve` methods find 3 or 4 steady states. Although the positions of most of the solutions are almost identical using the two methods (see Figure 3.5), there are a few exceptions where the two solutions diverge. We have investigated each of these exceptions. The result is presented in Table 3.1.

The main reason of discrepancy is a different number of solutions computed by the two methods. Then we have also computed symbolic steady state solutions using the Symbolic Math Toolbox of Matlab R2013b (MathWorks, Natick, USA). The symbolic solver did not find explicit solutions for 1 of these symbolic systems, reduced 3 other models to 4th degree equations in 3.5 to 47 s, and solved the 9 others in times from 2 to 20 s. The comparison to `IbexSolve` and homotopy solutions shows that `IbexSolve` always finds the right number of solutions in a fraction of a second and computes their positions with better precision than the homotopy method. We conclude that discrepancies result from the failure of the homotopy method to identify the right number of solutions.

We have also tested multistationarity on the database Biomodel (`https://www.ebi.ac.uk/biomodels/`), a repository of mathematical models of biological and biomedical systems, with `IbexSolve`. Parsed from SBML files to systems of differential equations and conservation laws using tools developed in [Lüders et al., 2020], and transformed into a minibex file for our algorithms. On 491 models tested with a timeout on 3600 seconds, 225 passed the test (with 199 models in less than 1 second), 191 have a timeout, and the 75 other models have syntax problems that need to be fixed or are piece-wise systems.

Among the models with a timeout, we can observe that some of them have unused variable, rendering the test inefficient. Indeed, in these models, solutions box are multiplied



a) b)

Figure 3.5: Comparison between homotopy and `IbexSolve` steady states. All the tested models are multistationary. a) Models were partitioned into two classes, with 3 (appearing first) and 4 homotopy solutions, then sorted by the average of the steady state concentrations in the homotopy solutions. Homotopy and `IbexSolve` solutions are represented as lines (red, green and blue for models with 3 steady states, cyan for the fourth) and crosses, respectively. b) Values of the steady state concentration a computed by homotopy and `IbexSolve`. Each `IbexSolve` steady state was related to the closest homotopy state (red +), in the Euclidean distance sense; reciprocally, each homotopy state was related to the closest `IbexSolve` state (blue crosses).

Table 3.1: Comparison of most divergent Ibex vs. homotopy solutions to symbolic solutions. n is the number of steady states. *dist* is the distance between sets of steady states solutions computed by homotopy or IbexSolve and the symbolic solutions, computed as $\frac{1}{2n}(\sum_{i=1}^{n} \min_j d_{i,j}) + \frac{1}{2n_s}(\sum_{j=1}^{n_s} \min_i d_{i,j})$, where $d_{i,j}$ is the Euclidean distance between the numerical solution i and the symbolic solution j, n_s is either the number of solutions n_h found by the homotopy method or the number n_i found by IbexSolve.

model	n **sym**	n_h **homo**	n_i **Ibex**	*dist* **homo**	*dist* **Ibex**
M338-2	3	4	3	0.020087	2.9101e-05
M464-1	3	4	3	0.012388	4.7049e-05
M464-2	3	4	3	0.029019	3.5893e-05
M488-1	3	4	3	0.011739	2.4117e-05
M488-2	3	4	3	0.010935	2.3473e-05
M488-3	3	4	3	0.017165	5.5298e-05
M506-1	4	3	4	0.10086	2.6487e-05
M506-2	4	3	4	0.0069613	4.3250e-06
M506-14	4	3	4	0.0092411	5.0559e-05
M95-1	4	3	4	0.0033133	1.1911e-05
M95-2	4	3	4	0.00063113	1.1230e-05

by the precision times the larger of the initial box for the unused variables. For example, the Biomodel 233 which is a two variables model, have four variables in the given input. It have only two solutions if we correct the input, but with the two unused variables in the original one, the solver is out of timeout since it need to split the space $[0, 1e14]^2$ in small boxes. It is then important to enhance my parser to take into account these unused variables.

With these observations, these results are very encouraging since multistationarity is a computationally hard problem with numerous applications to cell fate decision processes in development, cancer, tissue remodelling.

3.7.2 Homeostasis

The homeostasis tests were benchmarked using the database Biomodels (`https://www.ebi.ac.uk/biomodels/`). Of the 297 models initially considered (available at this moment), 72 were selected. These models have a unique steady state where every species has a non null concentration. To select them we have considered several tests described in Table 3.2.

The selected models were partitioned into three categories depending on the possible tests: kinetics rates, conservation laws, and volume compartments. We also tested for ACR the three models described in [Shinar and Feinberg, 2010] in which the parameters have been fixed to random values. These three models were previously tested for ACR by the Shinar/Feinberg topological criterion, therefore should remain so for any parameter set. As expected, the three models respect the ACR condition.

The initial boxes/domains for the conservation laws and parameters values were determined from the nominal initial conditions and parameter values found in the SBML files. The initial intervals bounds were obtained by dividing and multiplying these nominal values by a factor 10 for total amount of conservation laws and for volume compartments, and by a factor 100 for kinetics parameters, respectively. Homeostasis was tested using Definition 3.2.1. ACR was tested using Definition 3.2.1 with $k \to 1$, *i.e.* almost zero width intervals, and where the varying input parameters are the conservation laws values (*i.e.* kinetic rates and volume compartment fixed). The execution time statistics are given in Table 3.3.

Table 3.2: The methodology applied to select models to be tested for homeostasis from the initial set of models. A test using `IbexSolve` guarantees the existence and position of all steady states. Then, each model with a unique steady state having a non zero concentration is selected. Then a COPASI [Hoops et al., 2006] time course test starting from the steady state indicated by `IbexSolve` has been achieved in order to remove models that presents a cycling behavior.

Test	**# tested**	**# passed**
$\lvert$steady states$\rvert \geq 1$ (`IbexSolve`)	297	191
unique steady state > 0	191	107
non-oscillatory steady state (COPASI)	107	72

In homeostasis studies, interval methods perform well for small and medium size models in the Biomodels database. When the size of compartments change, 3 models have homeostasis, with 2 of them presenting species independent from parameter changes. For the kinetical parameters change, 4 models present homeostasis and two of them are of small

Table 3.3: Statistics on the homeostasis test using `IbexHomeo`. Selected biomodels have been classified for three tests. All of them have been tested w.r.t. the kinetics rates, and each model presenting at least one conservation law has been tested for ACR. Moreover, models with several compartments have been tested w.r.t. their volume. As we have many timeout (computed as 360s per species, with an average dimension of 51.92 (15.48 for species, and 36.4 for parameters)), the time columns consider only models that passed the test. The last two columns indicate the models with ACR in the first test, or with a 2-homeostatic species in other cases (because we get data during computation it may occur that a timeout model gives us a homeostasis).

Test `IbexHomeo`	**# models**	**# timeout**	**time (s) (min/median/max)**	**yes**	**no**
ACR only	33	17	0.19/4.32/10887	3	14
kinetics only	72	41	0.4/122/1923	4	29
compartments only	14	9	0.4/70/189	3	3

size: BIOMD614 is a univariate model when steady-state happens only with a concentration equal to one, and BIOMD629 presents a buffering mechanism (a buffer is a molecule occurring in much larger amounts than its interactors and whose concentration is nearly constant). The other two models are BIOMD048 and BIOMD093, where a timeout happens. For the initial conditions change, 3 models presents ACR, and two others (BIOMD041 and BIOMD622) homeostasis. Among them, BIOMD413 verifies the conditions of the Shinar-Feinberg theorem [Shinar and Feinberg, 2010], but the other two models (BIOMD489 and BIOMD738) do not since the deficiency of their reaction network is different from 1. This confirms that these conditions are sufficient, but not necessary. Our new examples could be the starting point of research on more general conditions for ACR.

The low proportion of homeostasis could be explained by the possible incompleteness of the biochemical pathways models. Not only these models are not representing full cells or organisms, but they may also miss regulatory mechanisms required for homeostasis. Negative feed-back interaction is known to be the main cause of homeostasis (although feed-forward loops can also produce homeostasis) [Cooper, 2008]. As well known in machine learning, it is notoriously difficult to infer feed-back interaction. For this reason, many of the models in the Biomodels database were built with interactions that are predominantly forward and have only few feed-back interactions. It is therefore not a surprise that models that were on purpose reinforced in negative feed-back to convey biological homeostasis, such as BIOMD041, a model of ATP homeostasis in the cardiac muscle, or BIOMD433, a model of MAPK signalling robustness, or BIOMD355, a model of calcium homeostasis, were tested positively for homeostatic species.

3.8 Some models presenting homeostasis

Biomd614

BIOMD614 is a one species model, with equation:

$$\dot{x} = k_1 + k_2 k_3 x - k_1 x - k_2 k_3 x^2 \tag{3.7}$$

At steady state, this leads to:

$$k_1 + k_2k_3x = (k_1 + k_2k_3x)x \tag{3.8}$$

If $k_1 \neq 0$, the only solution to (3.8) is $x = 1$, which is the answer given by `IbexHomeo`. The model describes the irreversible reaction kinetics of the conformational transition of a human hormone, where x is the fraction of molecules having undergone the transition, which is inevitably equal to one at the steady state [Kamihira et al., 2000].

Biomd629

BIOMD629 has 2 reactions and 5 species, provides a 2-homeostasis for kinetics parameters with conserved total amounts fixed, provided by the SBML file. This model does not provide ACR, and the homeostasis found can be explained by the conserved total amounts, that lock species to a small interval. But if we change these total amounts and try again an homeostasis test, it should fail. Indeed this model is given by the equations :

$$\begin{aligned}
\dot{x_1} &= -k_2x_1x_3 + k_3x_2 \\
\dot{x_2} &= k_2x_1x_3 - k_3x_2 - k_4x_2x_4 + k_5x_5 \\
\dot{x_3} &= -k_2x_1x_3 + k_3x_2 \\
\dot{x_4} &= -k_4x_2x_4 + k_5x_5 \\
\dot{x_5} &= k_4x_2x_4 - k_5x_5 \\
x_4 + x_5 &= k_6 \\
x_2 + x_5 + x_3 &= k_7 \\
x_2 + x_5 + x_1 &= k_8
\end{aligned} \tag{3.9}$$

Here x_3 (receptor), and x_4 (coactivator) have been found homeostatic w.r.t. variations of the kinetics parameters. The total amounts are $k_6 = 30$, $k_7 = \frac{7}{2000}$, $k_8 = \frac{1}{2000}$. With these values, we get $x_5 \in]0, \frac{1}{2000}[$, which implies $x_4 \in]29.9995, 30.0005[$. In the same way we have $x_2 + x_5 \in]0, \frac{1}{2000}[$, which implies $x_3 \in]\frac{6}{2000}, \frac{7}{2000}[$. If k_6, k_7, k_8 were closer to each other, there would be no reason for homeostasis. This example corresponds to the homeostasis mechanism known in biochemistry as buffering: a buffer is a molecule in much larger amounts than its interactors and whose concentration is practically constant.

Biomd413

BIOMD413 is a five species model given by equations:

$$\begin{aligned}
\dot{x_1} &= -k_2x_1x_2 + k_1x_3 + k_{10} - k_3x_1 \\
\dot{x_2} &= -k_2x_1x_2 + k_1x_3 \\
\dot{x_3} &= k_2x_1x_2 - k_1x_3 - k_6x_3x_5 + k_4x_4 + k_5x_4 \\
\dot{x_4} &= k_6x_3x_5 - k_4x_4 - k_5x_4 \\
\dot{x_5} &= -k_6x_3x_5 + k_4x_4 + k_7 - k_8x_5 \\
x_2 + x_3 + x_4 &= k_{12}
\end{aligned} \tag{3.10}$$

which have the following CRN:

$$x_1 + x_2 \rightleftharpoons x_3 \leftarrow x_4 \rightleftharpoons x_3 + x_5$$
$$x_1 \rightleftharpoons \emptyset \rightleftharpoons x_5$$

Computing the deficiency of the network, we obtain $7 - 2 - 4 = 1$. However, there is no non-terminal nodes that differs in a species S. So, the Shinar-Feinberg criterion doesn't apply. We found that the species x_1 have ACR where the others have not.

Biomd489

The biomodel 489 have 35 species and 59 reactions, with 4 conservation laws. The computation of the homeostasis is under a time out, but we found 4 ACR species, when the others seems to be out of homeostasis when we check the solutions found in the first loop (when the constraints are a little more relaxed, see remark 3.5). The deficiency of the model is 20 and can not therefore respect the Shinar-Feinberg criterion.

Biomd738

This model presents homeostasis with respect to the size of the different compartments, and also ACR for 7 of the 11 species involved in the system. The deficiency of the system is 8, and doesn't follow the mass action law. The models is given by the following equations:

$$
\begin{aligned}
\dot{x}_1 &= \Big(k_{31}k_{25} - \frac{k_9k_{31}x_1}{(k_6+x_1)(1+\frac{x_6}{k_7})} - k_{13}k_{31}x_1 + k_2k_{35}(2x_7+x_8)\Big)/k_{31}\\
\dot{x}_2 &= \Big(k_4k_{37}x_2 - k_{14}k_{32}x_2\Big)/k_{32}\\
\dot{x}_3 &= \Big(k_{14}k_{32}x_2 - \frac{k_{11}k_{33}x_3}{(k_6+x_3)(1+\frac{x_6}{k_7})} + k_{20}k_{37}x_{12}\Big)/k_{33}\\
\dot{x}_4 &= \Big(-\frac{k_{10}k_{34}x_4}{(k_6+x_4)(1+\frac{x_6}{k_7})} + k_3k_{35}(2x_7+x_8)\Big)k_{34}\\
\dot{x}_5 &= \Big(k_{19}k_{35}(x_7+x_8) - k_1k_{35}x_9x_5 + k_3k_{35}(x_7+x_8) + k_5k_{35}(x_7+x_8) + k_2k_{35}(x_7+x_8)\Big)/k_{35}\\
\dot{x}_6 &= \Big(k_{35}k_{38} - k_{35}k_{39}x_6\Big)/k_{35}\\
\dot{x}_7 &= \Big(k_8k_{35}x_8x_9 - (k_{19}+k_2+k_3+k_5)k_{35}x_7\Big)/k_35\\
\dot{x}_8 &= \Big(k_1k_{35}x_5x_9 - k_8k_{35}x_8x_9 - (k_{19}+k_2+k_3+k_5)k_{35}x_8\Big)/k_{35}\\
\dot{x}_9 &= \Big(\frac{k_9k_{31}x_1}{(k_6+x_1)(1+\frac{x_6}{k_7})} + \frac{k_{11}k_{33}x_3}{(k_6+x_3)(1+\frac{x_6}{k_7})} + \frac{k_{10}k_{34}x_4}{(k_6+x_4)(1+\frac{x_6}{k_7})} + \frac{k_{12}k_{36}x_{10}}{(k_6+x_{10})(1+\frac{x_6}{k_7})}\\
&\quad - k_1k_{35}x_5x_9 - k_8k_{35}x_8x_9\Big)/k_{35}\\
\dot{x}_{10} &= \Big(k_5k_{35}(2x_7+x_8) - k_{15}k_{36}x_{10} - \frac{k_{12}k_{36}x_{10}}{(k_6+x_{10})(1+\frac{x_6}{k_7})}\Big)/k_{36}\\
\dot{x}_{11} &= 0\\
\dot{x}_{12} &= \Big(k_{19}k_{35}(2x_7+x_8) - (k_4+k_{20})k_{37}x_{12}\Big)/k_{37}\\
k_{41} &= k_{35}x_5 + k_{35}x_7 + k_{35}x_8
\end{aligned}
\tag{3.11}
$$

In this model, x_6 is quite robust to volume compartment and initial concentrations. This is in fact explained by the equation $\dot{x}_6 = 0 = k_{38} - k_{39}x_6$ which have for only solution $x_6 = k_{38}/k_{39}$.

Except x_6, we found that, $x_1, x_2, x_3, x_4, x_{10}$ and x_{12} present ACR. All these species are outside the compartment indicated by the volume k_{35}, but the process leading to this ACR is not clear. However, there is a structure that could be relevant, x_9 play probably an important role, when x_7 and x_8 play together outside the cycle.

3.9 IbexBoxHull, firsts experimentation

As explained before, searching k_{hom}-homeostasis is based on the idea that we are searching for the minimal box containing all the steady-states. Then, it was natural to develop an algorithm dedicated to this problem, that can be used for more general problems.

The scheme of the new algorithm is the following. We begin with an initial box that is immediately contracted and becomes the root of our tree of search. Then we consider two

boxes: the outer box and the inner box. The outer box begins as the initial box and will decrease over time, the inner box begins as an empty box and will grow over time. Outside the outer box, there is no solution. The inner box grows only when new feasible points are found, and is then the smallest box found that contains solutions. When the inner and outer boxes are equal, then we have found the smallest box containing all the feasible points of the system that are contained in the initial box. It should be noted that we consider two kinds of variables: the species and the parameters. We are searching for a convergence of the two boxes for the species only.

The algorithm follows a Branch and Bound scheme w.r.t the two boxes. The outer box is split (w.r.t species or parameters, currently using a largest first criterion), and then we follow a breadth first search in the sense that we contract each boxes in the bound (w.r.t species) of the outer box (that represent the hull of all boxes in the tree), before continuing.

We are testing a new contractor, based on `3BCID`. In this contractor, the shaving occurs for each species on the bounds, and only to shave the bounds. Then, if a box B shares a bound with the outer box, and this bound is given by $\overline{x_i}$, then the shaving will be done for the species x_i, and only in the direction that can decrease $\overline{x_i}$ (in figure 3.4 left, the procedure `var3B` is called only for the right side of the box). The rest remains the same.

After contraction of the current box $B = ([x_1], ..., [x_n], [p_1], ..., [p_r])$, we search for feasible points in this box. To do this, we fix the parameter variables to a random value $(p'_1, ..., p'_r) \in [p_1] \times ... \times [p_r]$, and make a call to `ibexsolve` with, for initial box, $B' = ([x_1], ..., [x_n], p'_1, ..., p'_r)$. If solutions are found, we double-check the solution with the loupfinder procedure of `ibexopt` (described in [Araya et al., 2014, Trombettoni et al., 2011b], and then we grow the inner box IB by taking the hull of IB and the new solutions.

Remark 3.6. It can be noted that, as our models come from ODE, we could use a simulation (for fixed parameters) to rapidly find a box where feasible points could be found. The difficulty comes from the fact that the simulation should be interfaced with the rest of the code. But, more importantly, if a model has a long time scale, the simulation should be adapted to this timescale. For example, if a model stabilizes after days, the simulation should cover theses days and not stop after some minutes.

Chapter 4

Tropical geometry for chemical reaction networks

4.1 Introduction

In 1920, Niels Bohr introduce the correspondence principle [Bohr, 1920], which states that the behavior of systems described by the theory of quantum mechanics can be reduced to a behavior of classical physics when the quantum numbers are very large or when the quantity of action represented by Planck's constant can be neglected in front of the action implemented in the system.

On another side, George M. Bergman study the limit of the logarithmic deformation of algebraic varieties [Bergman, 1971] and can be today considered as the first work about tropical geometry in our knowledge. These logarithmic deformations are referred as amoebas and Oleg Viro discover that, in the limit, they tend to a piece-wise linear set [Viro, 2001]. In fact, they consider deformations of the classical arithmetic given by

$$x \oplus_h y = h \log\big(\exp\Big(\frac{x}{h}\Big) + \exp\Big(\frac{y}{h}\Big)\big) \quad \text{and} \quad x \odot_h y = x + y.$$

When is not zero, these spaces are isomorphic to the classical $\mathbb{R}$ space, but when h tends to 0, we refer to the Maslov dequantization [Litvinov and Maslov, 1998] and the space obtained at the limit is not isomorphic to $\mathbb{R}$, it is the tropical semi-ring. This introduce an analogue of the Bohr's correspondence principle, when h corresponds to the Plank's constant, called the idempotent correspondence principle.

In the tropical arithmetic, we replace addition by minimum (which is idempotent), and multiplication by addition [Akian et al., 2006]. Then, in tropical arithmetic, $1 \oplus 1 = 1$ and $1 \odot 2 = 3$. This gives an interesting arithmetic that is strongly connected to the polyhedral geometry. Indeed, polynomials in tropical arithmetic corresponds to piece-wise linear convex functions with rational slopes. And we can link the classical algebraic geometry to the tropical algebraic geometry using the concept of valuation, which is the pillar of the idempotent correspondence principle.

Indeed, the valuation is the algebraic tool necessary to understand tropical geometry. With this, we retrieve in the tropical geometry the deformation of the algebraic variety.

The Kapranov theorem, and then the fundamental theorem of tropical algebraic geometry [Maclagan and Sturmfels, 2015] make links between classical varieties and tropical varieties. And as in classical geometry the Gröbner basis plays an important role, there is an analogous tropical basis that plays an important role as the tropicalization is not commutative with the intersection as it is shown in example 4.7.

Whenever the quantities measured in experimental sciences are widely distributed, in other words they span several orders of magnitude, the use of a logarithmic scale is appropriate. This remark is particularly important for the understanding of our application of tropical geometry to biology. Tropical geometry has also a direct relation with models of reaction networks as it can be used to approximate steady-states. The useful concept here is the Puiseux series [Maclagan and Sturmfels, 2015]. Indeed, by viewing each kinetic parameter as a power of ϵ, we can build a Puiseux series that represent the species concentrations at the steady state. Furthermore, tropical geometry can be applied in order to define and compute dominant terms in these series. To do this, it is useful to view the tropical geometry as the result of a valuation, rendering the justification of the theory easier and more general than using amoebas. Indeed, a tropical space is the result of a valuation on a field. In the case of Puiseux series, the valuation of a Puiseux series $c(t)$ is the minimal power of t. In our context, valuations can be interpreted as orders of magnitude.

4.2 Background

This section introduces the elements of tropical geometry needed in our study. The proofs can be found in [Maclagan and Sturmfels, 2015].

Definition 4.1. Let K be a field, and K^* the non-zero elements of K. A valuation on K is a function $v : K \to \mathbb{R} \cup \{\infty\}$ satisfying the following three axioms :

- $v(a) = \infty$ IFF $a = 0$,
- $\forall a, b \in K,\ v(ab) = v(a) + v(b)$,
- $\forall a, b \in K^*,\ v(a + b) \geq \min(v(a), v(b))$.

Moreover, if $v(a) \neq v(b)$ then $v(a + b) = \min(v(a), v(b))$.

The image of the valuation is denoted Γ_v.

Example 4.2. Let $\mathbb{C}\{\{t\}\}$ be the field of Puiseux series with coefficiens in $\mathbb{C}$. The scalars in this field are of the formal power series $c(t) = c_1 t^{a_1} + c_2 t^{a_2} + ...$, where the c_i are non-zero complex numbers for all i, and $a_1 < a_2 < ...$ are rational numbers that have a common denominator. The field of Puiseux series can be viewed as the union of the fields of Laurent series: $\mathbb{C}\{\{t\}\} = \bigcup_{n\geq 1} \mathbb{C}((t^{1/n}))$.

This field has the natural valuation $v : \mathbb{C}\{\{t\}\} \to \mathbb{R}$. For a given $c(t) \in \mathbb{C}\{\{t\}\}^*$, the valuation $v(c(t))$ is the lowest exponent a_1 appearing in the series expansion of $c(t)$.

Our strategy is to use scaling (see section 5.3) in order to transform each polynomial of an ODE system associated to the chemical reaction network into a Puiseux series in a scaling parameter ϵ. That allows then to use the valuation on them.

Definition 4.3. Let $f = \sum_{\boldsymbol{u}\in\mathbb{Z}^n} c_{\boldsymbol{u}}\boldsymbol{x}^{\boldsymbol{u}}$ a Laurent Polynomial, we define its tropicalization $trop(f)$ as:

$$trop(f)(\boldsymbol{w}) = \min_{\boldsymbol{u}\in\mathbb{Z}^n}(v(c_{\boldsymbol{u}}) + <\boldsymbol{w},\boldsymbol{u}>)$$

where $w \in \mathbb{R}^n$, and $<.,.>$ denote the scalar product.

We define the tropical hypersurface $trop(V(f))$ by the set

$$\{\boldsymbol{w} \in \mathbb{R}^n \ s.t. \text{ the minimum in } trop(f) \text{ is achieved at least twice}\}.$$

We note $V(f) = \{\boldsymbol{y} \in \mathbb{R}^n,\ f(\boldsymbol{y}) = 0\}$, and, if F is a tropical polynomial, we note $V(F) = \{\boldsymbol{w} \in \mathbb{R}^n \ s.t. \text{ the minimum in } F \text{ is achieved at least twice}\}$. With these notations we have, $trop(V(f)) = V(trop(f))$.

Definition 4.4. We say that the valuation splits if there is a group homomorphism $\phi : \Gamma_v \to K^*$ s.t. $v(\phi(w)) = w$. This is always the case if the field K is algebraically closed. We will denote the splitting $\phi(w) = t^w$. If Γ_v is also dense in $\mathbb{R}$, we can then define the initial form of f with respect to $\boldsymbol{w}$ by:

$$in_{\boldsymbol{w}}(f) = \overline{t^{-trop(f)(\boldsymbol{w})}f(t^{\boldsymbol{w}}\boldsymbol{x})} = \sum_{\boldsymbol{u}\in\mathbb{Z}^n, v(c_{\boldsymbol{u}})+<\boldsymbol{w},\boldsymbol{u}>=trop(f)(\boldsymbol{w})} \overline{c_{\boldsymbol{u}}t^{-v(c_{\boldsymbol{u}})}}\boldsymbol{x}^{\boldsymbol{u}}$$

The initial form is a element of $\Bbbk[\boldsymbol{x}^{\pm}]$, where $\Bbbk$ is the residue field of K, *ie.* $\Bbbk = \{c \in K,\ v(c) \geq 0\}/\{c \in K,\ v(c) > 0\}$, and $\overline{a}$ is the image of a in the residue field.

The initial form represents a way to get only the dominant terms of the polynomial considered, but ignoring the coefficient in a certain way (as we are in $\Bbbk$). For example, if we consider $f = (t + t^2)x_0 + 2t^2x_1 + 3t^4x_2 \in \mathbb{C}\{\{t\}\}[x_0^{\pm1}, x_1^{\pm1}, x_2^{\pm1}]$, and $\boldsymbol{w} = (0, 0, 0)$, then $trop(f)(\boldsymbol{w}) = 1$ and $in_{\boldsymbol{w}}(f) = x_0$. If $\boldsymbol{w} = (4, 2, 0)$, then $trop(f)(\boldsymbol{w}) = 4$ and $in_{\boldsymbol{w}}(f) = 2x_1 + 3x_2$. In our application we use slightly different notion to capture the dominant monomials, since we want to conserve the dynamic. Indeed, when we consider the truncated system of a system E, we consider the whole dominant monomial. In the previous example, for $\boldsymbol{w} = 1$, we consider $(t + t^2)x_0$ instead of x_0. However, it is easy to pass from our consideration to the initial form.

Theorem 4.5 (Kapranov's Theorem)**.** *Fix a Laurent polynomial* $f = \sum_{\boldsymbol{u}\in\mathbb{Z}^n} c_{\boldsymbol{u}}\boldsymbol{x}^{\boldsymbol{u}} \in \mathbb{K}[x_1^{\pm1}, ..., x_n^{\pm1}]$, *with* $\mathbb{K}$ *an algebraic closed field with a non-trivial valuation. The following three sets coincide:*

- *the tropical hypersurface* $trop(V(f))$ *in* $\mathbb{K}^n$,
- *the closure in* $\mathbb{K}^n$ *of the set* $\{\boldsymbol{w} \in \Gamma_v^n \ s.t.\ in_{\boldsymbol{w}}(f) \text{ is not a monomial}\}$,
- *the closure in* $\mathbb{K}^n$ *of* $\{(v(y_1), ..., v(y_n)),\ (y_1, ..., y_n) \in V(f)\}$.

In addition, if $\boldsymbol{w} = v(\boldsymbol{y})$ *for* $\boldsymbol{y} \in (\mathbb{K}^*)^n$ *with* $f(\boldsymbol{y}) = 0$ *and* $n > 1$ *then* $\mathcal{U}_{\boldsymbol{w}} = \{\boldsymbol{y}' \in V(f),\ v(\boldsymbol{y}') = \boldsymbol{w}\}$ *is an infinite subset of the hypersurface* $V(f)$.

This means that for a given polynomial, the set of non-linearity of the associated tropical polynomial corresponds to the valuation of the zero of the polynomial. However, tropicalization does not commute with intersection, rendering this theorem more complex in the case of several polynomials. This can be illustrated by the following example.

Definition 4.6. Let I be an ideal in the Laurent polynomial ring $\mathbb{K}[x^{\pm}] = \mathbb{K}[x_1^{\pm 1}, ..., x_n^{\pm n}]$ and let $X = V(I)$ be its variety in the algebraic torus $T^n = (\mathbb{K}^*)^n$. The tropicalization $trop(X)$ of the variety X is the intersection of all tropical hypersurfaces defined by Laurent polynomials in the ideal I:

$$trop(X) = \bigcap_{f \in I} trop(V(f)) \subset \mathbb{R}^n. \tag{4.1}$$

Example 4.7. Let $n = 2$, $\mathbb{K} = \mathbb{C}\{\{t\}\}$, and $I =< x + y + 1, x + 2y >$. Then $X = V(I) = \{(-2, 1)\}$ and hence $trop(X) = \{(0, 0)\}$. However, the intersection of the two tropical lines given by the ideal generators equals

$$trop(V(x + y + z)) \cap trop(V(x + 2y)) = \{(w_1, w_2) \in \mathbb{R}^2,\ w_1 = w_2 \leq 0\}.$$

Definition 4.8. A tropical variety is the tropicalization of a variety X. A finite intersection of tropical hypersurfaces is called a tropical prevariety.

Despite this fact, we have some interesting results:

Definition 4.9. We define the initial form of an ideal I as:

$$in_w(I) =< \{in_w(f),\ f \in I\} > .$$

Definition 4.10. Let I be an ideal in the Laurent polynomial ring $\mathbb{K}[x^{\pm}]$ over an algebraic closed field $\mathbb{K}$ with a non-trivial valuation. We call a unit an element f of $\mathbb{K}[x^{\pm}]$ such that $\overline{x} = 1$ in $\Bbbk[x^{\pm}]$. A finite generating set $\mathcal{T}$ of I is said to be a tropical basis if, for all weight vectors $\boldsymbol{w} \in \Gamma_v^n$, the initial ideal $in_{\boldsymbol{w}}(I)$ contains a unit if and only if $in_{\boldsymbol{w}}(\mathcal{T}) = \{in_{\boldsymbol{w}}(f),\ f \in \mathcal{T}\}$ contains a unit.

Theorem 4.11. *Every ideal I in $\mathbb{K}[x^{\pm}]$ has a finite tropical basis $\mathcal{T}$.*

Proposition 4.12. *Every tropical variety is a finite intersection of tropical hypersurfaces. More precisely, if $\mathcal{T}$ is a tropical basis of the ideal I then $trop(X) = \bigcap_{f \in \mathcal{T}} trop(V(f))$.*

Corollary 4.13. *If X is a subvariety of $(K^*)^n$, then its tropicalization $trop(X)$ is the support of a Γ_v-rational polyhedral complex.*

Theorem 4.14. *Let I be an ideal in $\mathbb{K}[x^{\pm}]$ and $X = V(I)$ its variety in $(\mathbb{K}^*)^n$. Then the following three subsets of $\mathbb{R}^n$ coincide:*

- *the tropical variety $trop(X)$ as defined in equation (4.1)*
- *the closure in $\mathbb{R}^n$ of the set of all vectors $\boldsymbol{w} \in \Gamma_v^n$ with $in_{\boldsymbol{w}}(I) \neq < 1 >$*
- *the closure of the set of coordinatewise valuations of points in X:*

$$v(X) = \{(v(u_1), ..., v(u_n)),\ (u_1, ..., u_n) \in X\}.$$

Using this theorem, if the steady states of a model form a discrete set of n points, the tropicalization of this variety (the n points) is at most n points in $\mathbb{R}^n$. But, as mentioned in example 4.7, if we consider only the intersection of the tropicalization of the zero of the polynomials involved in the system, we can have more points. This can happen when the polynomials in the system don't form a tropical basis after tropicalization.

The next section introduces the concept of tropical equilibration. In a tropical equilibration we consider two polynomials, one positive and one negative, that get the same valuation. This can interpreted as two forces that cancel each other out.

4.3 Tropical equilibration

To a polynomial, one can associate a polynomial in the Puiseux series using a scaling procedure. For example, the polynomial $k_1x_3 - k_2x_1 + k_3x_2$ is associated to $f := t^{\gamma_1}x_3 - t^{\gamma_2}x_1 + t^{\gamma_3}x_2 \in \mathbb{C}\{\{t\}\}[x_1^{\pm 1}, x_2^{\pm 1}, x_3^{\pm 1}]$, when the γ_i values depend on the scaling chosen depending of ϵ, if we take $k_i = \overline{k_i}\epsilon^{\gamma_i}$ with $\overline{k_i}$ near to 1. Then, we can compute the tropicalization of this new polynomial as in Defintion 4.3: $trop(f)(\boldsymbol{w}) = \min(\gamma_1 + w_3, \gamma_2 + w_1, \gamma_3 + w_2)$.

In practical applications, one can choose $\gamma_i = round(\log_\epsilon(|k_i|)$ if we are searching for integer exponents, and $\gamma_i = round(d \log_\epsilon(|k_i|)/d$ if we are searching for rational exponents. The values of w_i correspond to the same formula applied to the species concentration x_i.

With $0 < \epsilon < 1$, the scaling can be considered as defining orders of magnitude and if $\epsilon = 1/10$, we have the common (reversed) decimal order. Then the notion of tropicalization of the associated polynomial can be viewed as taking only the dominant monomial of the original system.

Definition 4.15. Let S be an ODE system as in (2.2). We split $f_i(\boldsymbol{k}, \boldsymbol{x}) = f_i^+(\boldsymbol{k}, \boldsymbol{x}) + f_i^-(\boldsymbol{k}, \boldsymbol{x})$, where f_i^+ is the sum of monomial with positive S_{ij} and f_i^- the sum of monomial wih negative S_{ij}. A full tropical equilibration is a subset of the tropical prevariety defined by the following set of equations:

$$trop(f_i^+)(\boldsymbol{w}) = trop(f_i^-)(\boldsymbol{w}), \ \forall i \in \{1, ..., n\}. \tag{4.2}$$

Moreover, if the system admits linear conservation laws

$$g_i = \sum_{j=1}^{n} C_{ij}x_j = k_i', \quad 1 \leq i \leq n_c, \tag{4.3}$$

where $C_{ij} \geq 0$, $k_i' > 0$, n_c the number of conservation laws, we add the following equations:

$$trop(g_i)(\boldsymbol{w}) = \gamma_i' \tag{4.4}$$

where $k_i' = \overline{k_i'}\epsilon^{\gamma_i'}$.

The full tropical equilibration problem comes from the idea that if we have two similar forces of opposite directions, then the object does move slowly, depending on the other forces and the slight differences between the two opposite forces. This corresponds in biology to a metastable state (a region of very slow dynamic in the phase space), or to a steady state.

As $\mathbb{C}$ is algebraically closed, this is the case for the field of Puiseux series $\mathbb{C}\{\{t\}\}$. Then each polynomial in the Puiseux series has solutions in the Puiseux series. Suppose now, that we have a Polynomial in the Puiseux series $\mathbb{C}\{\{\epsilon\}\}$ in the form $P(x, \epsilon) = \sum_j S_j\epsilon^{\gamma_j}x^{\alpha_j}$ with solution $x(\epsilon) = c_1\epsilon^{a_1} + c_2\epsilon a_2 + ... = c_1\epsilon^{a_1}(1 + x_1(\epsilon))$. Then, by replacing the solution in the polynomial, we get $P(x, \epsilon) = \sum_j S_j c_1^{\alpha_j}\epsilon^{\gamma_j + a_1\alpha_j} + r_1(\epsilon) = 0$. A necessary condition for $P(x, \epsilon) = 0$ read at lowest order $\sum_{j, \gamma_j + a_1\alpha_j = m}$ with $m = \min_j(\gamma_j + a_1\alpha_j)$. It is then necessary for the minimum m to be attained at least twice. Moreover, if one looks for real solutions, then we retrieve the tropical equilibration problem. Then the tropical equilibration can be viewed as a first approximation to an attractive slow manifold.

About the conservation laws in the tropical equilibration, the idea is that if we move slowly, we should stay around the same concentrations for each species, in particular in

a cycle of reactions. Moreover, adding the conservation law constraints on the tropical equilibration problem allow to simplify the problem, rendering the tropical equilibration easier to understand.

We note that the tropical equilibrations depend on the ϵ value since we use a rounding. Without the rounding, we have $\epsilon' = \epsilon^{\alpha}$, and the minimums for ϵ' correspond to the minimums of ϵ plus α and we obtain a translation of the solutions in the $\mathbb{R}^n$ space. But with the rounding, some equilibrations disappear or appear when we change the value of ϵ. When ϵ stays around one, the rounding doesn't have a real influence, but as ϵ tends to 0, the rounding will crush everything, meaning that all values will tend to be a zero order of ϵ, rendering the multiscaleness of the system unusable.

The tropical equilibration problem is important for the model reduction part. Indeed, using the scaling, we are able to get a multiscale system, but this scaling depends on the species concentrations, and the tropical equilibration offers different regions of interest, since, as explained in [Radulescu et al., 2015a], the tropical equilibration is necessary to obtain the zero of a real polynomial (considering nonzero concentration of species), which is a condition on a the fast species in the reduced model. However, a same branch (*ie.* a maximal polyhedron of the polyhedral complex that represents solutions of the tropical equilibration problem) of tropical equilibrations can lead to several reductions. Indeed, a species time scale can change on a branch, as it will be illustrated with the Tyson cell cycle model.

However, it can happen that some models can not satisfy the tropical equilibration problem, or that the condition is too strong. For example, the original Michaelis-Menten system has a term with only one monomial and then can not satisfy the tropical equilibration problem. Fortunately, this term is involved in conservation laws and can then be replaced formally, leading to work with a simplified systems that no longer has this problem. But this is not always the case, and as the slow species do not need to be equilibrated in the theorems justifying the model reduction (see section 5.2), it leads to the formulation of a weaker problem: the partial tropical equilibration problem.

The partial equilibration problem can be decomposed in two parts, the first part is a choice of slow species, and the second part acts as the full tropical equilibration problem for the other species. The space given by the partial equilibration is then an over-space of the full equilibration truncated by the slow/fast decomposition.

Let us reformulate the full tropical equilibration problem and compare it with the partial equilibration problem.

Definition 4.16.

- (*Full equilibration*) Let S be a polynomial ODE system as in equation (2.3) with eventually some conservation laws as in (4.3). Then the full tropical equilibration is expressed as finding a vector $\boldsymbol{a}$ such that:

$$\min_{j, S_{ij}>0}(\gamma_{ij} + \langle \boldsymbol{a}, \boldsymbol{\alpha}_j \rangle) = \min_{j, S_{ij}<0}(\gamma_{ij} + \langle \boldsymbol{a}, \boldsymbol{\alpha}_j \rangle),\ 1 \leq i \leq n$$

$$\min_{j, C_{lj}\neq 0}(a_j) = \gamma'_l,\ 1 \leq l \leq n_c.$$

- (*Partial equilibration*) Let S be a polynomial ODE system as in equation (2.3) with eventually some conservations laws as in (4.3). Let $T \subset \{1, ..., n\}$ be a set representing the fast species. Then the partial equilibration for T consists in finding a vector $\boldsymbol{a}$ such that:

$$\min_{j, S_{ij}>0}(\gamma_{ij} + \langle \boldsymbol{a}, \boldsymbol{\alpha}_j \rangle) = \min_{j, S_{ij}<0}(\gamma_{ij} + \langle \boldsymbol{a}, \boldsymbol{\alpha}_j \rangle),\ i \in T$$

$$\min_{j, C_{lj} \neq 0}(a_j) = \gamma'_l, \ 1 \leq l \leq n_c$$

$$\min_j(\gamma_{ij} + \langle \boldsymbol{a}, \boldsymbol{\alpha}_j \rangle) - a_i < \min_j(\gamma_{i'j} + \langle \boldsymbol{a}, \boldsymbol{\alpha}_j \rangle) - a_{i'}, \ i \in T, \ i' \in \{1, ..., n\} \backslash T.$$

The idea of considering partial equilibrations has been first mentioned in [Soliman et al., 2014] without a criterion on which variables to equilibrate and which not. Timescale criteria have been proposed for the first time in [Samal et al., 2016].

As an example to illustrate the differences between these two similar concepts and how they works, we show the tropical equilibrations for the Tyson cell cycle model.

4.4 Tropical equilibration of the Tyson cell cycle model

The following model is stored in the Biomodels database [Le Novere et al., 2006] as the BIOMD0000000005. The original model has 9 species and 9 chemical reactions. However, three species are on boundary and among those one satisfies an assignment rule. We have used a Matlab parser to generate the ODE system. Because the Matlab parser doesn't cope with rules, the model has been modified to take this into account. We call this new model biomd005c.

$$\begin{aligned}
\dot{x}_1 &= k_1 x_3 - k_2 x_1 + k_3 x_2, \\
\dot{x}_2 &= k_2 x_1 - k_3 x_2 - k_4 x_2 x_5, \\
\dot{x}_3 &= k_{10} x_4 - k_1 x_3 + k_9 x_3^2 x_4, \\
\dot{x}_4 &= k_4 x_2 x_5 - k_{10} x_4 - k_9 x_3^2 x_4, \\
\dot{x}_5 &= k_6 - k_4 x_2 x_5, \\
\dot{x}_6 &= k_1 x_3 - k_8 x_6, \\
k_{14} &= x_1 + x_2 + x_3 + x_4.
\end{aligned} \tag{4.5}$$

The values of the parameters are $k_1 = 1$, $k_2 = 1000000$, $k_3 = 1000$, $k_4 = 200$, $k_6 = 3/200$, $k_8 = 3/5$, $k_9 = 180$, $k_{10} = 9/500$, $k_{14} = 1$.

Using that $k_i = \overline{k_i}\epsilon^{\gamma_i}$ and $x_i = \overline{x_i}\epsilon^{a_i}$, we obtain the following full tropical equilibration problem:

$$\begin{aligned}
\min(a_3 + \gamma_1, a_2 + \gamma_3) &= a_1 + \gamma_2, \\
a_1 + \gamma_2 &= \min(a_2 + \gamma_3, a_2 + a_5 + \gamma_4), \\
\min(a_4 + \gamma_{10}, 2a_3 + a_4 + \gamma_9) &= a_3 + \gamma_1, \\
a_2 + a_5 + \gamma_4 &= \min(a_4 + \gamma_{10}, 2a_3 + a_4 + \gamma_9), \\
\gamma_6 &= a_2 + a_5 + \gamma_4, \\
a_3 + \gamma_1 &= a_6 + \gamma_8, \\
\min(a_1, a_2, a_3, a_4) &= \gamma_{14}.
\end{aligned} \tag{4.6}$$

We will solve the problem parametrically, i.e. the parameter valuations will be treated as symbols, rather than use numerical instances.

We can easily eliminate a_3 and a_5 to get the system:

$$\begin{aligned}
\min(a_6+\gamma_8, a_2+\gamma_3) &= a_1+\gamma_2,\\
a_1+\gamma_2 &= \min(a_2+\gamma_3,\gamma_6),\\
\min(a_4+\gamma_{10}, 2(a_6+\gamma_8-\gamma_1)+a_4+\gamma_9) &= a_6+\gamma_8,\\
\gamma_6 &= \min(a_4+\gamma_{10}, 2(a_6+\gamma_8-\gamma_1)+a_4+\gamma_9),\\
a_5 &= \gamma_6-\gamma_4-a_2,\\
a_3 &= a_6+\gamma_8-\gamma_1,\\
\min(a_1,a_2,a_3,a_4) &= \gamma_{14}.
\end{aligned} \tag{4.7}$$

Using lines 3 and 4, we can also eliminate a_6 to get:

$$\begin{aligned}
\min(\gamma_6, a_2+\gamma_3) &= a_1+\gamma_2,\\
a_1+\gamma_2 &= \min(a_2+\gamma_3,\gamma_6),\\
a_6 &= \gamma_6-\gamma_8\\
\gamma_6 &= a_4+\min(\gamma_{10}, 2(\gamma_6-\gamma_1)+\gamma_9),\\
a_5 &= \gamma_6-\gamma_4-a_2,\\
a_3 &= \gamma_6-\gamma_1,\\
\min(a_1,a_2,a_3,a_4) &= \gamma_{14}.
\end{aligned} \tag{4.8}$$

Then, we can make a less conventional elimination of a_1 and a_4:

$$\begin{aligned}
a_1 &= \min(a_2+\gamma_3,\gamma_6)-\gamma_2,\\
a_6 &= \gamma_6-\gamma_8\\
a_4 &= \gamma_6-\min(\gamma_{10}, 2(\gamma_6-\gamma_1)+\gamma_9),\\
a_5 &= \gamma_6-\gamma_4-a_2,\\
a_3 &= \gamma_6-\gamma_1,\\
\min(a_1,a_2,a_3,a_4) &= \gamma_{14}.
\end{aligned} \tag{4.9}$$

Then, we can rewrite the system in the following manner:

$$\begin{aligned}
\gamma_{14} &= \min(a_1,a_2,a_3,a_4),\\
a_1 &= \min(a_2+\gamma_3-\gamma_2, \gamma_6-\gamma_2),\\
a_3 &= \gamma_6-\gamma_1,\\
a_4 &= \gamma_6-\min(\gamma_{10}, 2\gamma_6-2\gamma_1+\gamma_9),\\
a_5 &= \gamma_6-\gamma_4-a_2,\\
a_6 &= \gamma_6-\gamma_8,
\end{aligned} \tag{4.10}$$

Now, consider that the parameter $k_{14}=1$, then $\gamma_{14}=0$ (whatever ϵ is). In this case we can define $\Gamma=\min(\gamma_6-\gamma_2,a_3,a_4)=\min(\gamma_6-\gamma_2,\gamma_6-\gamma_1,\gamma_6-\min(\gamma_{10},2\gamma_6-2\gamma_1+\gamma_9))$, $\Gamma'=\gamma_3-\gamma_2$, $\Gamma''=\gamma_6-\gamma_3$, $\Theta=2\gamma_6-2\gamma_1+\gamma_9$, and the system rewrite:

$$\begin{aligned}
0 &= \min(a_2 + \Gamma', a_2, \Gamma), \\
a_1 &= \min(a_2, \Gamma'') + \Gamma', \\
a_3 &= \gamma_6 - \gamma_1, \\
a_4 &= \gamma_6 - \min(\gamma_{10}, \Theta), \\
a_5 &= \gamma_6 - \gamma_4 - a_2, \\
a_6 &= \gamma_6 - \gamma_8,
\end{aligned} \tag{4.11}$$

The signs of $\Gamma, \Gamma', \Gamma''$ determine the dimension and the number of tropical branches.

1. $\Gamma < 0 (= \gamma_{14})$. There is no solution.

2. $\Gamma > 0$. Then the equation is equivalent to

$$\min(a_2 + \Gamma', a_2) = 0,$$

which has a unique solution (one point)

   (a) If $\Gamma' \geq 0$, then $a_2 = 0$. Then, if $\Gamma'' \geq 0$, $a_1 = \Gamma'$, else $a_1 = \gamma_6 - \gamma_2$, and if $\Theta \geq \gamma_{10}$, $a_4 = \gamma_6 - \gamma_{10}$, else $a_4 = 2\gamma_1 - \gamma_6 - \gamma_9$.

   (b) If $\Gamma' \leq 0$, then $a_2 + \Gamma' = 0$. Then, if $\Gamma'' \geq 0$, $a_1 = 0$, else $a_1 = \gamma_6 - \gamma_2$, and if $\Theta \geq \gamma_{10}$, $a_4 = \gamma_6 - \gamma_{10}$, else $a_4 = 2\gamma_1 - \gamma_6 - \gamma_9$.

3. $\Gamma = 0$. Then the equation is equivalent to

$$\min(a_2 + \Gamma', a_2) \geq 0,$$

which has one or two one dimensional branches: a half line and possibly an interval.

   (a) If $\Gamma' \geq 0$, then the two branches are
   - $\Gamma'' \geq a_2 \geq 0$, $a_1 = a_2 + \Gamma'$ (the interval). One can note that this branch exists only if $\Gamma'' \geq 0$;
   - $a_2 \geq \Gamma''$, $a_1 = \gamma_6 - \gamma_2$. If $\Gamma'' \leq 0$, this branch turns into $a_2 \geq 0$, $a_1 = \gamma_6 - \gamma_2$.

   (b) If $\Gamma' \leq 0$, then the two branches are
   - $\Gamma'' \geq a_2 \geq -\Gamma'$, $a_1 = a_2 + \Gamma'$ (the interval). One can note that this branch exist only if $\Gamma'' \geq -\Gamma'$;
   - $a_2 \geq \Gamma''$, $a_1 = \gamma_6 - \gamma_2$. If $\Gamma'' \leq -\Gamma'$, this branch turns into $a_2 \geq -\Gamma'$, $a_1 = \gamma_6 - \gamma_2$.

For both of these case, the comparison between Θ and γ_{10} will determine the value of a_4, but does not have an impact on the topology of the set of solutions.

Using equations (4.5), (4.6) and (4.11), we can compute the timescale orders of the species in the full tropical equilibration:

$$
\begin{aligned}
b_1 &= \gamma_2 \\
b_2 &= \min(\gamma_3, \gamma_6 - a_2) = \gamma_3 + \min(0, \Gamma'' - a_2), \\
b_3 &= \gamma_1, \\
b_4 &= \min(\gamma_{10}, 2\gamma_6 - 2\gamma_1 + \gamma_9), \\
b_5 &= \gamma_4 + a_2, \\
b_6 &= \gamma_8.
\end{aligned}
\tag{4.12}
$$

We can see that, for fixed parameters, each timescale is fixed, except b_2 and b_5, that both depend on a_2.

Now, consider that x_4 is a slow species. This is possible with our numerical parameters, so we will consider a formal partial equilibration as above, and then turn out to numerical computation for both total and partial equilibrations.

We consider x_4 as the slowest species, so we need to add in the constraints that $b_4 > b_i$ for $i \in \{1, 2, 3, 5, 6\} = T$, but we don't need that x_4 is equilibrated, so we get the following system:

$$
\begin{aligned}
\min(a_3 + \gamma_1, a_2 + \gamma_3) &= a_1 + \gamma_2, \\
a_1 + \gamma_2 &= \min(a_2 + \gamma_3, a_2 + a_5 + \gamma_4), \\
\min(a_4 + \gamma_{10}, 2a_3 + a_4 + \gamma_9) &= a_3 + \gamma_1, \\
\gamma_6 &= a_2 + a_5 + \gamma_4, \\
a_3 + \gamma_1 &= a_6 + \gamma_8, \\
\min(a_1, a_2, a_3, a_4) &= \gamma_{14}, \\
\gamma_2 &< \min(a_2 + a_5 - a_4 + \gamma_4, \gamma_{10}, 2a_3 + \gamma_9), \\
\min(\gamma_3, \gamma_6 - a_2) &< \min(a_2 + a_5 - a_4 + \gamma_4, \gamma_{10}, 2a_3 + \gamma_9), \\
\gamma_1 &< \min(a_2 + a_5 - a_4 + \gamma_4, \gamma_{10}, 2a_3 + \gamma_9), \\
a_2 + \gamma_4 &< \min(a_2 + a_5 - a_4 + \gamma_4, \gamma_{10}, 2a_3 + \gamma_9), \\
\gamma_8 &< \min(a_2 + a_5 - a_4 + \gamma_4, \gamma_{10}, 2a_3 + \gamma_9),
\end{aligned}
\tag{4.13}
$$

As for the full equilibration, we can eliminate some species, here they are a_6 and a_5, then a_1 and a_4.

$$
\begin{aligned}
\min(a_3+\gamma_1, a_2+\gamma_3) &= \min(a_2+\gamma_3, \gamma_6),\\
a_1 &= \min(a_2+\gamma_3, \gamma_6) - \gamma_2,\\
a_4 &= a_3+\gamma_1 - \min(\gamma_{10}, 2a_3+\gamma_9),\\
a_5 &= -a_2+\gamma_6-\gamma_4,\\
a_6 &= a_3+\gamma_1-\gamma_8,\\
\min(a_1,a_2,a_3,a_4) &= \gamma_{14},\\
\gamma_2 &< \min(-a_3-\gamma_1+\min(\gamma_{10},2a_3+\gamma_9)+\gamma_6, \gamma_{10}, 2a_3+\gamma_9),\\
\min(\gamma_3, \gamma_6-a_2) &< \min(-a_3-\gamma_1+\min(\gamma_{10},2a_3+\gamma_9)+\gamma_6, \gamma_{10}, 2a_3+\gamma_9),\\
\gamma_1 &< \min(-a_3-\gamma_1+\min(\gamma_{10},2a_3+\gamma_9)+\gamma_6, \gamma_{10}, 2a_3+\gamma_9),\\
a_2+\gamma_4 &< \min(-a_3-\gamma_1+\min(\gamma_{10},2a_3+\gamma_9)+\gamma_6, \gamma_{10}, 2a_3+\gamma_9),\\
\gamma_8 &< \min(-a_3-\gamma_1+\min(\gamma_{10},2a_3+\gamma_9)+\gamma_6, \gamma_{10}, 2a_3+\gamma_9),
\end{aligned}
\tag{4.14}
$$

At this stage, by just considering equalities, we obtain 32 cases, and adding inequalities, we have 128 cases (they can lead to the same polyhedron). With this example, we can see that the partial equilibration can be more complex to compute than the total one. But lets check the numerical values.

First of all, we can see that the values of $\Gamma, \Gamma', \Gamma''$ can depends on the ϵ value. Indeed, computed for $\epsilon = 1/11$, we get $\gamma_1 = 0$, $\gamma_2 = -6$, $\gamma_3 = -3$, $\gamma_4 = -2$, $\gamma_6 = 2$, $\gamma_8 = 0$, $\gamma_9 = -2$, $\gamma_{10} = 2$, $\gamma_{14} = 0$, and lead to $\Gamma = 0$, $\Gamma' = 3$, and $\Gamma'' = 5$. However, computed for $1/29$, we get $\Gamma = 1$, $\Gamma' = 2$, and $\Gamma'' = 3$. This show that the value of ϵ is important, and also that the convergence to zero (we remember that when $\epsilon \to 0$, $\gamma_i \to 0$ for all i) is not monotonic for these critical orders, since they involve several γ_i. For the rest of the comparison, we will consider $\epsilon = 1/11$.

For this experience, the full tropical equilibration solutions correspond to one segment and one half line. We have compared the tropical equilibration with some trajectory of the system to obtain that, if we begin a trajectory inside the tropical equilibration, the trajectory can leave the tropical equilibration but remains close to it and finishes around the limit point of the tropical equilibration. This is shown in figure 4.1. This shows that, even if the idea of compensated terms bears semblance to the idea of attractive region, mathematically there is no complete equivalence of the two notions. Indeed, outside the tropical equilibration, uncompensated terms will lead the trajectory to the tropical equilibration. On the tropical equilibration the equality of orders is not a sufficient condition for compensation: two polynomials term of same order can have a ratio different from one, and the other monomials, which are not dominant, can have a small impact on the trajectory. As a matter of fact, tropical geometry ideas must be combined with the geometric singular perturbations theory in order to justify existence and stability of the compensation. As discussed in the chapter 7, this implies several other conditions, such as the hyperbolicity of the equilibration.

Considering, x_4 as the slowest spacies again, we obtain, with the numeric values, the following system.

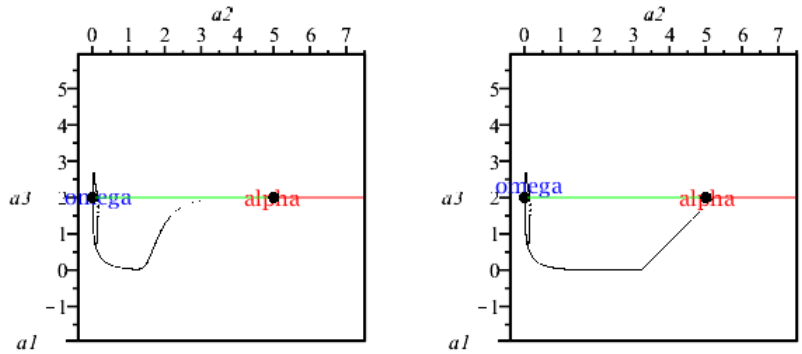


Figure 4.1: Trajectories (in log scale) of the models with different initial values of x_5 and x_6, compared to the tropical equilibration.

$$
\begin{aligned}
\min(a_3, a_2 - 3) &= \min(a_2 - 3, 2),\\
a_1 &= \min(a_2 - 3, 2) + 6\\
a_4 &= a_3 - \min(2, 2a_3 - 2),\\
a_5 &= -a_2 + 4,\\
a_6 &= a_3,\\
0 &= \min(\min(a_2 - 3, 2) + 6, a_2, a_3, a_3 - \min(2, 2a_3 - 2)),\\
-6 &< \min(-a_3 + 2 + \min(2, 2a_3 - 2), 2, 2a_3 - 2),\\
\min(-3, 2 - a_2) &< \min(-a_3 + 2 + \min(2, 2a_3 - 2), 2, 2a_3 - 2),\\
0 &< \min(-a_3 + 2 + \min(2, 2a_3 - 2), 2, 2a_3 - 2),\\
a_2 - 2 &< \min(-a_3 + 2 + \min(2, 2a_3 - 2), 2, 2a_3 - 2),\\
0 &< \min(-a_3 + 2 + \min(2, 2a_3 - 2), 2, 2a_3 - 2).
\end{aligned}
\tag{4.15}
$$

This lead to the following system

$$
\begin{aligned}
\min(a_3, a_2 - 3) &= \min(a_2 - 3, 2),\\
a_1 &= \min(a_2 - 3, 2) + 6\\
a_4 &= a_3 - \min(2, 2a_3 - 2),\\
a_5 &= -a_2 + 4,\\
a_6 &= a_3,\\
0 &= \min(\min(a_2 - 3, 2) + 6, a_2, a_3, a_3 - \min(2, 2a_3 - 2)),\\
\max(0, a_2 - 2, \cancel{\min(-3, 2 - a_2)}) &< \min(-a_3 + 2 + \min(2, 2a_3 - 2), 2, 2a_3 - 2),
\end{aligned}
\tag{4.16}
$$

Forgetting the last line, coming from the time scale inequalities, we obtain 10 different cases leading to the following polyhedra (in blue the total equilibration):

$$\begin{array}{cccc}
a_1 = 8 & a_1 = 6 & a_1 = 8 & a_1 = 8 \\
a_2 = 5 & a_2 = 3 & a_2 = 0 & a_2 = 5 \\
a_3 = 2 & a_3 = 0 & a_3 = 2 & a_3 = 0 \\
a_4 = 0 & a_4 = 2 & a_4 = 0 & a_4 = 2 \\
a_5 = -1 & a_5 = 1 & a_5 = 4 & a_5 = -1 \\
a_6 = 2 & a_6 = 0 & a_6 = 2 & a_6 = 0
\end{array}$$

$$\begin{array}{ccc}
a_1 = 8 & a_1 = 8 & 3 < a_1 = a_2 + 3 < 6 \\
a_2 > 5 & a_2 > 5 & 0 < a_2 < 3 \\
a_3 = 2 & a_3 = 0 & a_3 = 0 \\
a_4 = 0 & a_4 = 0 & a_4 = 2 \\
a_5 = -a_2 + 4 < -1 & a_5 = -a_2 + 4 < -1 & 1 < a_5 = -a_2 + 4 < 4 \\
a_6 = 2 & a_6 = 0 & a_6 = 0
\end{array}$$

$$\begin{array}{ccc}
a_1 = 3 & 3 < a_1 = a_2 + 3 < 8 & a_1 = 3 \\
a_2 = 0 & 0 < a_2 < 5 & a_2 = 0 \\
a_3 > 2 & a_3 = 2 & 0 < a_3 < 2 \\
a_4 = a_3 - 2 > 0 & a_4 = 0 & 0 < a_4 = -a_3 + 2 < 2 \\
a_5 = 4 & -1 < a_5 = -a_2 + 4 < 4 & a_5 = 4 \\
a_6 = a_3 > 2 & a_6 = 2 & 0 < a_6 = a_3 < 2
\end{array}$$

Adding the time scale constraint and projecting the polyhedral complex on the space (a_2, a_3), we obtain the following figure 4.2.

If we consider also x_3 as slow with x_4, we get two two-dimensional branches that extend the partial equilibration for x_4 slow, and one one-dimensional branch that is a truncation of the branch for x_4 slow coming from the total equilibration. This is illustrated in figure 4.3.

4.5 Algorithms for tropical equilibration

Several methods were used to computing tropical equilibration, starting with constraint programming [Soliman et al., 2014] and Newton polytopes [Samal et al., 2015]. In this section we present the method combining polyhedral geometry and Satisfiability Modulo Theories (SMT) introduced in [Lüders, 2020].

A polyhedron is defined as finite intersection of half-spaces. The full tropical equilibration is defined by an intersection of union of polyhedra.

In definition 4.16, let $\rho_i = argmin_{j,S_{ij}>0}(\gamma_{ij} + \langle \boldsymbol{a}, \boldsymbol{\alpha}_j \rangle)$, $\eta_i = argmin_{j,S_{ij}<0}(\gamma_{ij} + \langle \boldsymbol{a}, \boldsymbol{\alpha}_j \rangle)$, and $\sigma_l = argmin_{j,C_{lj}\neq 0}(a_j)$. For each $1 \leq i \leq n$, by fixing ρ_i and η_i, we get a polyhedron consisting in the equation and inequations

$$\begin{aligned}
&\gamma_{i\rho_i} + \langle \boldsymbol{a}, \boldsymbol{\alpha}_{\rho_i} \rangle = \gamma_{i\eta_i} + \langle \boldsymbol{a}, \boldsymbol{\alpha}_{\eta_i} \rangle \\
&\gamma_{i\rho_i} + \langle \boldsymbol{a}, \boldsymbol{\alpha}_{\rho_i} \rangle \leq \gamma_{ij} + \langle \boldsymbol{a}, \boldsymbol{\alpha}_j \rangle, \quad 1 \leq j \leq r_i
\end{aligned} \tag{4.17}$$

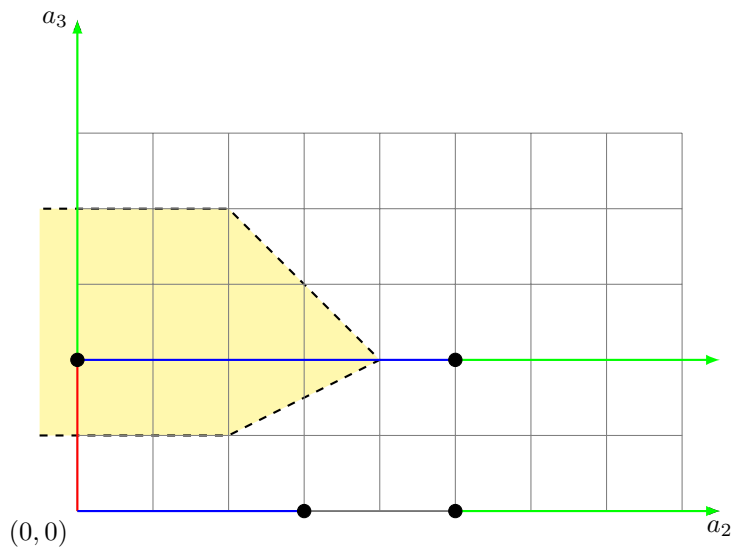


Figure 4.2: the yellow open space represent the constraint $b_4 > b_i$ for $i \in T$. Points and colored lines represent each a polyhedron defined by the other constraints and described above. The two lines defined with $a_3 = 2$ are the full tropical equilibration.

when r_i denote the number of distinct monomials in $f_i(\boldsymbol{k}, \boldsymbol{x})$. We have also a polyhedron for each $1 \leq l \leq n_c$ by fixing σ_l, given by the equation and inequations

$$\begin{aligned} &a_{\sigma_l} = \gamma_l' \\ &a_{\sigma_l} \leq a_j, \quad 1 \leq j \leq n,\ C_{lj} \neq 0. \end{aligned} \tag{4.18}$$

By rolling the choices, for each equation, we get a union of polyhedra, called bag. The tropical equilibration problem consists to take the intersection of these bags.

This is encoded in `SMTcut` [Lüders, 2020] and follow the algorithm 8.

Algorithm `smtcut`*(*(2.3)*,*(4.3)*)*
- $bb \leftarrow$ `makePolyhedraForFTE`((2.3), (4.3))
- $rr \leftarrow$ `computePolyhedronDnf`(bb)
- **return** rr

Algorithm 8: The algorithm used to compute the full tropical equilibration of a system.

In definition 4.16, let $\zeta_i = argmin_j(\gamma_{ij} + \langle \boldsymbol{a}, \boldsymbol{\alpha}_j \rangle - a_i)$, $\omega_{i'} = argmin_j(\gamma_{i'j} + \langle \boldsymbol{a}, \boldsymbol{\alpha}_j \rangle - a_{i'})$, and $\sigma_l = argmin_{j, C_{lj} \neq 0}(a_j)$. For each $1 \leq i, i' \leq n$, by fixing ζ_i and $\omega_{i'}$, we get a polyhedron

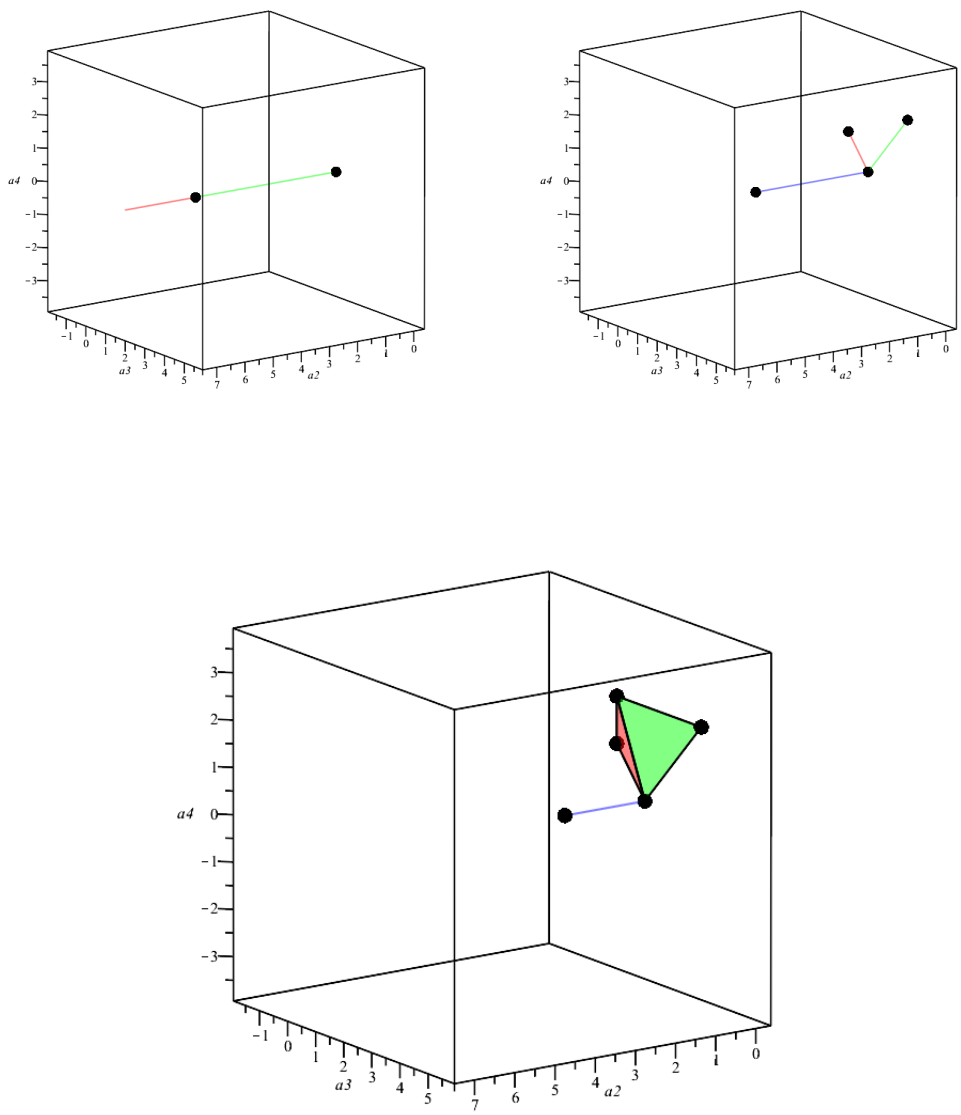


Figure 4.3: Left: the total equilibration. Right: the partial equilibration for $T = \{1, 2, 3, 5, 6\}$ (x_4 slow). Bottom: the partial equilibration for $T = \{1, 2, 5, 6\}$ (x_3, x_4 slows). All pictures are a projection in the space generated by a_2, a_3, a_4.

```
Algorithm makePolyhedraForFTE((2.3),(4.3))
    bb ← ∅
    foreach 1 ≤ i ≤ n do
        bb ← bb ∪ equilibrate(ẋ_i)
    return bb
```

Algorithm 9: This algorithm makes the list of bags, each bag representing a list of polyhedra, such that each polyhedron is linked to an equilibration for a species or a conservation law.

```
Algorithm equilibrate(ẋ_i)
    pp, np, b ← ∅
    foreach 1 ≤ j ≤ r_i do
        t = trop(k_j x^{α_j})
        if S_ij < 0 then
            np ← np ∪ t
        else
            pp ← pp ∪ t
    foreach (a, c) ∈ pp × np do
        p = makePolyhedron(a, c, pp, np)
        b ← b ∪ p
    return b
```

Algorithm 10: This algorithm computes the bag b linked to an equation of the system, that represent each possible equilibration. It splits negative and positive monomials, computes their tropicalization and makes the bag.

consisting in the equation and inequations

$$\begin{aligned} \gamma_{i\zeta_i} + \langle \boldsymbol{a}, \boldsymbol{\alpha}_{\zeta_i} \rangle - a_i &\leq \gamma_{i'\omega_{i'}} + \langle \boldsymbol{a}, \boldsymbol{\alpha}_{\eta_{i'}} \rangle - a_{i'} \\ \gamma_{i\zeta_i} + \langle \boldsymbol{a}, \boldsymbol{\alpha}_{\zeta_i} \rangle - a_i &\leq \gamma_{ij} + \langle \boldsymbol{a}, \boldsymbol{\alpha}_j \rangle, \quad 1 \leq j \leq r_i \\ \gamma_{i'\omega_{i'}} + \langle \boldsymbol{a}, \boldsymbol{\alpha}_{\omega_{i'}} \rangle - a_{i'} &\leq \gamma_{i'j} + \langle \boldsymbol{a}, \boldsymbol{\alpha}_j \rangle - a_{i'}, \quad 1 \leq j \leq r_{i'} \end{aligned} \tag{4.19}$$

when r_i denote the number of distinct monomials in $f_i(\boldsymbol{k}, \boldsymbol{x})$. We have also a polyhedron for each $1 \leq l \leq n_c$ by fixing σ_l, given by the equation and inequations as in the full tropical equilibration:

$$\begin{aligned} a_{\sigma_l} &= \gamma'_l \\ a_{\sigma_l} &\leq a_j, \quad 1 \leq j \leq n, \ C_{lj} \neq 0. \end{aligned} \tag{4.20}$$

To find the partial equilibration for T, a modified version of the algorithms used to find the full tropical equilibration has been developed. The idea behind is that we just need to modify the sets of polyhedra we use, this is encoded by the algorithm13.

```
Algorithm makePolyhedron(a, c, pp, np)
    eq ← a = c
    ieq ← ∅
    foreach i ∈ pp ∪ np do
        ieq ← ieq ∪ a ≤ i
    p ← make(eq, ieq)
    return p
```

Algorithm 11: p is a polyhedron defined by equation (4.17) or (4.18).

```
Algorithm computePolyhedronDnf(bb)
    solver ← getSolveur(incremental = true)
    f ← convertToSMTFormula(bb)
    rr ← ∅, bool ← true
    while bool do
        solver.addAssertion(f)
        (x, bool) = solver.solve(f) /*x is a point that satisfy the constraints, bool is
          false if no x*/
        if Not(bool) then
            Break
        else
            R = ∅
            foreach b ∈ bb do
                foreach P ∈ b do
                    if x ∈ P then
                        R ← R ∪ P.constraints()
                        Break
            f ← Not(R)
            rr ← rr ∪ R
    return rr
```

Algorithm 12: This algorithm computes the intersection of a set of bags bb. Each polyhedron is transformed to a logical constraint. x represents a point that satisfy the set of constraints, then, if a such x is found, it is contained by a polyhedron P common to each equilibration. We remove this polyhedron from search by adding a constraint and continue the search until there is no feasible point. At the end we get a list of polyhedra rr, that is the full tropical equilibration.

```
Algorithm smtcutpartial((2.3),(4.3),T)
    bb ← makePolyhedraForPTE((2.3), (4.3), T)
    rr ← computePolyhedronDnf(bb)
    return rr
```

Algorithm 13: The algorithm used to find the partial tropical equilibration for T.

```
Algorithm makePolyhedraForPTE((2.3),(4.3), T)
    bb ← ∅
    foreach i ∈ T do
        bb ← bb ∪ equilibrate(ẋ_i)
        foreach j ∈ {1, ..., n}\T do
            bb ← bb ∪ slowFastPol(ẋ_i, ẋ_j)
    return bb
```

Algorithm 14: This algorithm makes the list of bags, each bag representing a list of polyhedra, such that each polyhedron is linked to an equilibration for a fast species or a conservation law, or a slow fast decomposition.

```
Algorithm slowFastPol(ẋ_i, ẋ_j)
    sp, fp, b ← ∅
    foreach 1 ≤ m ≤ r_i do
        t = trop(k_m x^{α_m − a_i})
        fp ← fp ∪ t
    foreach 1 ≤ m ≤ r_j do
        t = trop(k_m x^{α_m − a_j})
        sp ← sp ∪ t
    foreach (a, c) ∈ sp × fp do
        p = makePolyhedronSF(a, c, sp, fp)
        b ← b ∪ p
    return b
```

Algorithm 15: This algorithm computes the bag b linked to a slow fast decomposition between two species x_i (fast) and x_j (slow). As the order the species has an impact on the slow fast decomposition, we multiply each monomial in $\dot{x}_q$ by $\frac{1}{x_q}$. Then we split each monomials, compute their tropicalization and make the bag.

```
Algorithm makePolyhedronSF(a, c, sp, fp)
    eq ← ∅
    ieq ← a ≥ c
    foreach i ∈ sp do
        ieq ← ieq ∪ a ≤ i
    foreach i ∈ fp do
        ieq ← ieq ∪ c ≤ i
    p ← make(eq, ieq)
    return p
```

Algorithm 16: p is a polyhedron defined by equation (4.19).

Chapter 5

Model reduction

5.1 Introduction

Two methods are traditionally used to reduce non-linear chemical reaction network models with multiple timescales: the quasi-equilibrium (QE) [Gorban et al., 2001] and the quasi-steady state (QSS) approximations [Segel and Slemrod, 1989, Boulier et al., 2011, Radulescu et al., 2012].

In order to introduce these methods let us consider again the Michaelis-Menten model introduced in the Section 2.2.

For the QSS approximation, we consider that the total concentration of enzyme $c_1 = x_2 + x_3 = [E] + [ES]$ is much lower than the total concentration of substrate $x_1 + x_2$. Then, the complex ES is a low concentration, fast species. Its concentration is slaved by the concentration of S, meaning that the value of $[ES]$ almost instantly relaxes to a value depending on S. The simplified mechanism corresponds to pooling the two reactions of the mechanism into a unique irreversible reaction $S \xrightarrow{R(x_1,c_1)} P$, which means that $\dot{x}_4 = -\dot{x}_1 = k_3 x_{2(QSS)}$. The QSS value of x_2 results from the equation $k_1 x_1 (c_1 - x_{2(QSS)}) = (k_2 + k_3) x_{2(QSS)}$. From this it follows that

$$R(x_1, c_1) = \frac{k_3 c_1 x_1}{\frac{k_2+k_3}{k_1} + x_1}.$$

We note here that the equation inducing the QSS value of x_2 is a polynomial equation for the fast species concentration, akin to the one leading to tropical equilibration.

For the QE approximation, we consider that the first reaction of the mechanism is a fast, reversible reaction. The simplified mechanism corresponds to a pooling of species. The total concentration of enzymes, c_1, and the total concentration of substrate, $S_{tot} = x_1 + x_2$, are conserved by the fast reversible reaction, but only one, c_1, is conserved by the two reaction of the mechanism. The total concentration of substrate is slowly consumed by the second reaction and represent the slow variable of the system. The single step approximation reads $S_{tot} \xrightarrow{R(S_{tot},c_1)} P$ and we have $\dot{x_4} = -\dot{S}_{tot} = k_3 x_{2(QE)}$. The QE value of x_2 is the unique positive solution of the quadratic equation $k_1(S_{tot} - x_{2(QE)}(c_1 - x_{2(QE)}) = k_2 x_{2(QE)}$, representing the compensation of forward and backward fluxes of the fast reaction. From

this it follows that

$$R(S_{tot}, c_1) = \frac{2k_3 c_1 S_{tot}}{c_1 + S_{tot} + \frac{k_2}{k_1}} \frac{1}{1 + \sqrt{1 - \frac{4c_1 S_{tot}}{(c_1 + S_{tot} + \frac{k_2}{k_1})^2}}}.$$

When the concentration of enzyme is small, $c_1 << S_{tot}$, we obtain the original equation of Michaelis and Menten, $R(S_{tot}, c_1) \approx k_3 \frac{c_1 S_{tot}}{\frac{k_2}{k_1} + S_{tot}}$.

The main problem of these reductions is that the QE reaction and QSS species should be detected. This is possible by rescaling parameters and variables. The rescaling can be found by intuition for simple models such as the Michaelis-Menten mechanism, but this does not work for large models. Like in [Radulescu et al., 2012,Kruff et al., 2021], we propose to use a general scaling algorithm based on tropical geometry before applying singular perturbation results that reduce the model.

5.2 Singular perturbation

In 1952, Tikhonov is interested in the asymptotic analysis for differential equations containing small parameters in the derivatives and predicts the behavior of a such systems supposing several conditions [Tikhonov, 1952a]. Hoppensteadt then corrects the proof, adding some more conditions [Hoppensteadt, 1967,Hoppensteadt, 1969a]. The actual formulation of the Tikhonov theorem have the following form [Verhulst, 2005a]:

Theorem 5.1. *Consider the initial value problem*

$$\begin{aligned} \dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{y}, t) + \epsilon..., \ \boldsymbol{x}(0) = \boldsymbol{x}_0, \ \boldsymbol{x} \in D \subset \mathbb{R}^n, \ t \geq 0 \\ \epsilon \dot{\boldsymbol{y}} = g(\boldsymbol{x}, \boldsymbol{y}, t) + \epsilon..., \ \boldsymbol{y}(0) = \boldsymbol{y}_0, \ \boldsymbol{y} \in G \subset \mathbb{R}^m. \end{aligned} \tag{5.1}$$

For f and g, we take sufficiently smooth vector functions in $\boldsymbol{x}, \boldsymbol{y}$ and t; the dots represents (smooth) high-order terms in ϵ.

a) We assume that a unique solution of the initial value problem exists and suppose this holds also for the reduced problem

$$\begin{aligned} \dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{y}, t), \ \boldsymbol{x}(0) = \boldsymbol{x}_0, \\ 0 = g(\boldsymbol{x}, \boldsymbol{y}, t), \end{aligned} \tag{5.2}$$

with solutions $\bar{\boldsymbol{x}}(t), \bar{\boldsymbol{y}}(t)$.

b) Suppose that $0 = g(\boldsymbol{x}, \boldsymbol{y}, t)$ is solved by $\bar{\boldsymbol{y}} = \phi(\boldsymbol{x}, t)$, where $\phi(\boldsymbol{x}, t)$ is a continuous function and an isolated root. Also suppose that $\bar{\boldsymbol{y}} = \phi(\boldsymbol{x}, t)$ is an asymptotically stable solution of the equation

$$\frac{d\boldsymbol{y}}{d\tau} = g(\boldsymbol{x}, \boldsymbol{y}, t) \tag{5.3}$$

that is uniform in the parameters $\boldsymbol{x} \in D$ and $t \in \mathbb{R}_+$, where $\tau = \frac{t}{\epsilon}$.

c) $\boldsymbol{y}(0)$ is contained in an interior subset of the domain of attraction of $\bar{\boldsymbol{y}} = \phi(\boldsymbol{x}, t)$ in the case of the parameter values $\boldsymbol{x} = \boldsymbol{x}(0), t = 0$.

Then we have

$$\begin{aligned} \lim_{\epsilon\to 0} \boldsymbol{x}_\epsilon(t) &= \bar{\boldsymbol{x}}(t),\ 0 \le t \le L, \\ \lim_{\epsilon\to 0} \boldsymbol{y}_\epsilon(t) &= \bar{\boldsymbol{y}}(t),\ 0 < d \le t \le L \end{aligned} \tag{5.4}$$

with d and L constants independent of ϵ.

It can be noted that Hoppensteadt has given a version of the theorem for infinite time interval [Hoppensteadt, 1966].

Later, in 1979, Fenichel approached the problem with a geometric point of view [Fenichel, 1979]. Then Cardin and Teixeira [Cardin and Teixeira, 2017] extended the Fenichel theory for multiple time scales. Combining tropical geometry and the the results of Cardin and Teixera, a model reduction method, using tropical geometry ideas to detect concentration time scales and singular perturbations to justify the reduction, has been proposed very recently [Kruff et al., 2021].

The methods proposed in these papers rely on a hyperbolicity condition: the eigenvalues of the Jacobian matrix of the fast subsystem should be non-zero. However, in applications, it happens that some eigenvalues are zero. These critical cases were studied by Vasil'eva in [Vasil'eva and Butuzov, 1980] to construct regular and boundary layer series for the asymptotic expansions of the solutions. This is also the goal of this chapter, when the these critical cases result from conservation laws.

We give here the principal results of the geometric singular perturbations theory as they are a the starting point for our model reduction method based on approximate conservation laws exposed in the next chapters. The two following sections are mainly derived from [Cardin and Teixeira, 2017].

5.2.1 Fenichel theorems

We consider the system

$$\begin{aligned} \dot{\boldsymbol{z}}_1 &= f_1(\boldsymbol{z}, \boldsymbol{\epsilon}), \\ \dot{\boldsymbol{z}}_i &= \Big(\prod_{i=1}^{j-1} \epsilon_i\Big) f_j(\boldsymbol{z}, \boldsymbol{\epsilon}),\ j \in \{2, ..., n\} \end{aligned} \tag{5.5}$$

where $\boldsymbol{z} = (\boldsymbol{z}_1, ..., \boldsymbol{z}_i) \in U \subset \mathbb{R}^{m_1} \times ... \times \mathbb{R}^{m_n}$, U is open with compact closure, $(0, ..., 0) < \boldsymbol{\epsilon} = (\epsilon_1, ..., \epsilon_n) << (1, ..., 1)$, and f_j is supposed sufficiently smooth.

If $n = 2$ we obtain the fast system

$$\dot{\boldsymbol{z}}_1 = f(\boldsymbol{z}, \epsilon_1), \quad \dot{\boldsymbol{z}}_2 = \epsilon_1 f_2(\boldsymbol{z}, \epsilon_1) \tag{5.6}$$

After time rescaling we obtain the slow system

$$\epsilon_1 \dot{\boldsymbol{z}}_1 = f(\boldsymbol{z}, \epsilon_1), \quad \dot{\boldsymbol{z}}_2 = f_2(\boldsymbol{z}, \epsilon_1) \tag{5.7}$$

One can note that these two systems are equivalent for $\epsilon_1 \neq 0$. But at the limit $\epsilon_1 \to 0$, they give two different systems. The layer problem

$$\dot{\boldsymbol{z}}_1 = f_1(\boldsymbol{z}, 0), \quad \dot{\boldsymbol{z}}_2 = 0, \tag{5.8}$$

and the reduced problem

$$0 = f_1(\boldsymbol{z}, 0), \quad \dot{\boldsymbol{z}}_2 = f_2(\boldsymbol{z}, 0). \tag{5.9}$$

We note $\mathcal{S} = \{\boldsymbol{z} : f_1(\boldsymbol{z}, 0) = 0\}$ the critical manifold.

Definition 5.2. We say that $\mathcal{S}_0 \subset \mathcal{S}$ is normally hyperbolic if $\forall \boldsymbol{z} \in \mathcal{S}_0, \forall \lambda$ eigenvalue of the jacobian matrix $D_{\boldsymbol{z}_1} f_1(\boldsymbol{z}, 0)$, $\mathcal{R}e(\lambda) \neq 0$.

We recall that the Hausdorff distance between two non empty and compact subsets X and Y of a metric space (E, d) is given by:

$$d_H(X, Y) = \max\{\sup_{y \in Y} d(X, y), \ \sup_{x \in X} d(x, Y)\} = \max\{\sup_{y \in Y} \inf_{x \in X} d(x, y), \ \sup_{x \in X} \inf_{y \in Y} d(x, y)\}.$$

Theorem 5.3. *Consider a C^∞ family like (5.6). Let $\mathcal{S}_0 \subset \mathcal{S}$ be a normally hyperbolic compact manifold, possibly with a boundary. Then, for ϵ_1 sufficiently small, there exists a manifold $\mathcal{S}_{\epsilon_1}$ locally invariant under the flow of system (5.6). The manifold $\mathcal{S}_{\epsilon_1}$ has a Hausdorff distance $\mathcal{O}(\epsilon_1)$ (as $\epsilon_1 \to 0$) from $\mathcal{S}_0$, and it is diffeomorphic to $\mathcal{S}_0$. Moreover, $\mathcal{S}_{\epsilon_1}$ is C^r smooth for any $r < \infty$, and the flow on $\mathcal{S}_{\epsilon_1}$ converges to the slow flow as $\epsilon_1 \to 0$.*

$\mathcal{S}_{\epsilon_1}$ is called a slow manifold. Now, suppose that $\mathcal{S}_0 = \{(\boldsymbol{z}_1, \boldsymbol{z}_2) : \boldsymbol{z}_1 = h_0(\boldsymbol{z}_2)\}$ and $\mathcal{S}_{\epsilon_1} = \{(\boldsymbol{z}_1, \boldsymbol{z}_2) : \boldsymbol{z}_1 = h_{\epsilon_1}(\boldsymbol{z}_2)\}$, then we have $\dot{\boldsymbol{z}}_2 = f_2(h_{\epsilon_1}(\boldsymbol{z}_2), \boldsymbol{z}_2, \epsilon_1) \to \dot{\boldsymbol{z}}_2 = f_2(h_0(\boldsymbol{z}_2), \boldsymbol{z}_2, 0)$. If $\mathcal{S}_0 \subset \mathcal{S}$ is normally hyperbolic, then each point $\boldsymbol{z}_0 \in \mathcal{S}_0$ is a hyperbolic equilibrium of the layer problem (5.8), so that $\boldsymbol{z}_0$ possesses local stable and unstable manifolds $W^s(\boldsymbol{z}_0)$ and $W^u(\boldsymbol{z}_0)$, respectively. We define the local stable and unstable manifolds of $\mathcal{S}_0$ by

$$W^s(\mathcal{S}_0) = \bigcup_{\boldsymbol{z}_0 \in \mathcal{S}_0} W^s(\boldsymbol{z}_0) \quad \text{and} \quad W^u(\mathcal{S}_0) = \bigcup_{\boldsymbol{z}_0 \in \mathcal{S}_0} W^u(\boldsymbol{z}_0) \tag{5.10}$$

Theorem 5.4. *Under the hypotheses of Theorem 5.3, for ϵ_1 sufficiently small, there exist local stable and unstable manifolds $W^s(\mathcal{S}_{\epsilon_1})$ and $W^u(\mathcal{S}_{\epsilon_1})$ diffeomorphic to $W^s(\mathcal{S}_0)$ and $W^u(\mathcal{S}_0)$, respectively. They lie within $\mathcal{O}(\epsilon_1)$ of $W^s(\mathcal{S}_0$ and $W^u(\mathcal{S}_0)$, respectively. Moreover, they are C^r smooth for any $r < \infty$ and locally invariant under the flow of the system (5.6).*

We note $p.t$ the trajectory through p evolved after time t, and $V.t$ the application of the flow after time t.

Theorem 5.5. *Under the hypotheses of theorem 5.3, for every $\boldsymbol{z}_{\epsilon_1} \in \mathcal{S}_{\epsilon_1}$ with ϵ_1 sufficiently small, there are manifolds $W^s(\boldsymbol{z}_{\epsilon_1}) \subset W^s(\mathcal{S}_{\epsilon_1})$ and $W^u(\boldsymbol{z}_{\epsilon_1}) \subset W^u(\mathcal{S}_{\epsilon_1})$ diffeomorphic to $W^s(\boldsymbol{z}_0)$ and $W^u(\boldsymbol{z}_0)$, respectively. They lie within $\mathcal{O}(\epsilon_1)$ of $W^s(\boldsymbol{z}_0)$ and $W^u(\boldsymbol{z}_0)$, respectively. Moreover, They are C^r smooth for any $r < \infty$, and the families $\{W^s(\boldsymbol{z}_{\epsilon_1}) : \boldsymbol{z}_{\epsilon_1} \in \mathcal{S}_{\epsilon_1}\}$ and $\{W^u(\boldsymbol{z}_{\epsilon_1}) : \boldsymbol{z}_{\epsilon_1} \in \mathcal{S}_{\epsilon_1}\}$ are invariant in the sense that*

$$W^s(\boldsymbol{z}_{\epsilon_1}).t \subset W^s(\boldsymbol{z}_{\epsilon_1}.t) \quad \textit{and} \quad W^u(\boldsymbol{z}_{\epsilon_1}).t \subset W^u(\boldsymbol{z}_{\epsilon_1}.t)$$

for all $t \geq 0$ and $t \leq 0$, respectively.

We suppose that the manifolds $W^s(\mathcal{S}_0)$ and $W^u(\mathcal{S}_0)$ have dimensions $j_1^s + m_2$ and $j_1^u + m_2$ with $j_1^s + j_1^u = m_1$.

Theorem 5.6. *Consider a C^∞ family like 5.3, and let $\mathcal{S}_0 \subset \mathcal{S}$ be a j-dimensional compact normally hyperbolic invariant manifold of the reduced problem (5.9) with a $(j + j_2^s)$-dimensional local stable manifold W^s and a $(j + j_2^u)$-dimensional local unstable manifold W^u. Suppose that $D_{x_1} f_1(x_0, 0)$ has j_1^s and j_1^u eigenvalues with negative and positive real parts, respectively, for all $x_0 \in \mathcal{S}$. Then there exists $\delta_1 > 0$ such that the following hold:*

- *There exists a family of smooth manifolds $\{\mathcal{S}_{\epsilon_1} : \epsilon_1 \in (0, \epsilon_1)\}$ such that $\mathcal{S}_{\epsilon_1} \to \mathcal{S}_0$ when $\epsilon_1 \to 0$, according to Hausdorff distance, and $\mathcal{S}_{\epsilon_1}$ is a hyperbolic invariant manifold of (5.6).*

- *There are families of $(j + j_1^s + j_2^s)$-dimensional and $(j + j_1^u + j_2^u)$-dimensional smooth manifolds $\{W^s_{\epsilon_1} : \epsilon_1 \in (0, \delta_1)\}$ and $\{W^u_{\epsilon_1} : \epsilon_1 \in (0, \delta_1)\}$ such that the manifolds $W^s_{\epsilon_1}$ and $W^u_{\epsilon_1}$ are local stable and unstable manifolds of $\mathcal{S}_{\epsilon_1}$.*

5.2.2 Extension to the multi-scale systems

From system (5.5), we can extract n different time scales:

$$\tau_1, \quad \tau_k = \Big(\prod_{i=1}^{k-1} \epsilon_i\Big)\tau_1 = \epsilon_{k-1}\tau_{k-1} \quad \text{for } k \in \{2, ..., n\}.$$

τ_1 is the fastest time scale, τ_n the slowest one, τ_k the k-intermediate time scale ($k \in \{2, ..., n-1\}$). We note that system (5.5) is called the fastest system, when

$$\begin{aligned} \Big(\prod_{i=j}^{k-1} \epsilon_i\Big)\dot{\boldsymbol{z}}_j &= f_j(\boldsymbol{z}, \boldsymbol{\epsilon}), \quad j \in \{1, ..., k-1\} \\ \dot{\boldsymbol{z}}_k &= f_k(\boldsymbol{z}, \boldsymbol{\epsilon}) \\ \dot{\boldsymbol{z}}_l &= \Big(\prod_{i=k}^{l-1} \epsilon_i\Big) f_l(\boldsymbol{z}, \boldsymbol{\epsilon}), \quad l \in \{k+1, ..., n\} \end{aligned} \tag{5.11}$$

is the k-intermediate system, and

$$\begin{aligned} \Big(\prod_{i=j}^{n} \epsilon_i\Big)\dot{\boldsymbol{z}}_j &= f_j(\boldsymbol{z}, \boldsymbol{\epsilon}), \quad j \in \{1, ..., n-1\} \\ \dot{\boldsymbol{z}}_n &= f_n(\boldsymbol{z}, \boldsymbol{\epsilon}) \end{aligned} \tag{5.12}$$

is the slowest system.

For each time scale, we will note the derivation by $\dot{\boldsymbol{z}}$ instead of $\frac{d\boldsymbol{z}}{d\tau_k}$ if it is clear from the context. We note that when for each $i \in \{1, ..., n-1\}$, $\epsilon_i \neq 0$, then the three systems are equivalent. Now, if $\boldsymbol{\epsilon} \to 0$, we obtain three different systems. The layer problem

$$\begin{aligned} \dot{\boldsymbol{z}}_1 &= f_1(\boldsymbol{z}, 0) \\ \dot{\boldsymbol{z}}_j = 0, \quad & j \in \{2, ..., n\}, \end{aligned} \tag{5.13}$$

the k-intermediate problem ($k \in \{2, ..., n-1\}$)

$$\begin{aligned} 0 = f_j(\boldsymbol{z}, 0), \quad j \in \{1, ..., k-1\} \\ \dot{\boldsymbol{z}}_k = f_k(\boldsymbol{z}, 0) \\ \dot{\boldsymbol{z}}_l = 0, \quad l \in \{k+1, ..., n\}, \end{aligned} \tag{5.14}$$

and the reduced problem

$$\begin{aligned} 0 = f_j(\boldsymbol{z}, 0), \quad j \in \{1, ..., n-1\} \\ \dot{\boldsymbol{z}}_n = f_n(\boldsymbol{z}, 0). \end{aligned} \tag{5.15}$$

For each $k \in \{1, ..., n-1\}$, let $\mathcal{M}_k = \{\boldsymbol{z} \in U : f_j(\boldsymbol{z}, 0) = 0 \text{ for } j = 1, ..., k\}$. We note that $\mathcal{M}_{n-1} \subset ... \subset \mathcal{M}_1$. $\mathcal{M}_1$ is the critical manifold and $\mathcal{M}_k$ is called the k-critical manifold. We call the auxiliary system the following system

$$\begin{aligned} f_j(\boldsymbol{z}, 0, ..., 0, \epsilon_k, ..., \epsilon_{n-1}) = 0 \quad \text{for } j \in \{1, ..., k-1\} \\ \dot{\boldsymbol{z}}_k = f_k(\boldsymbol{z}, 0, ..., 0, \epsilon_k, ..., \epsilon_{n-1}) \\ \dot{\boldsymbol{z}}_l = \big(\prod_{i=k}^{l-1} \epsilon_i\big) f_l(\boldsymbol{z}, 0, ..., 0, \epsilon_k, ..., \epsilon_{n-1}) \quad \text{for } l \in \{k+1, ..., n\}. \end{aligned} \tag{5.16}$$

These dynamical systems are defined on the manifold

$$\mathcal{M}_k^{(\epsilon_k, ..., \epsilon_{n-1})} = \{\boldsymbol{z} \in U : f_j(\boldsymbol{z}, 0, ..., 0, \epsilon_k, ..., \epsilon_{n-1}) = 0 \text{ for } j \in \{1, ..., k-1\}\}.$$

For the following theorems, we consider the following assumption:

(H) For each $k \in \{1, ..., n-1\}$, if $\mathcal{L} \subset \mathcal{M}_k$ is a compact manifold, we assume that the matrix $D_{\boldsymbol{z}_k} f_k(\boldsymbol{z}_0, 0)$ has no eigenvalue with zero real part for all $\boldsymbol{z}_0 \in \mathcal{L}$.

Theorem 5.7. *Consider a C^∞ family like (5.5), with $n \geq 2$. Let $\mathcal{N} \subset \mathcal{M}_{n-1}$ be a compact manifold, possibly with a boundary, and assume that hypothesis (H) is fulfilled. Then there are small positive real numbers $\delta_1, ..., \delta_{n-1}$ and a family $\{\mathcal{N}_{\boldsymbol{\epsilon}} = \mathcal{N}_{(\epsilon_1, ..., \epsilon_{n-1})} : \epsilon_i \in (0, \delta_i), i \in \{1, ..., n-1\}\}$ of locally invariant manifolds under the flow of systems (5.5). Moreover, $\mathcal{N}_{\boldsymbol{\epsilon}}$ has Hausdorff distance $\mathcal{O}(\epsilon_1 + ... + \epsilon_{n-1})$ from $\mathcal{N}$, and it is diffeomorphic to $\mathcal{N}$. Also, $\mathcal{N}_{\boldsymbol{\epsilon}}$ is C^r smooth for any $r < \infty$, and the flow on $\mathcal{N}_{\boldsymbol{\epsilon}}$ converges to the slow flow (the reduced problem) (5.15) as $\boldsymbol{\epsilon} \to 0$.*

Corollary 5.8. *Consider a C^∞ family like (5.5), with $n \geq 3$. For each $k \in \{2, ..., n-1\}$, assume that $\epsilon_k, ..., \epsilon_{n-1}$ are fixed and nonzero, and let $\mathcal{L} \subset \mathcal{M}_l^{(\epsilon_k, ..., \epsilon_{n-1})}$ be compact manifold, possibly with boundary, and assume that hypothesis (H) for the cases $j \in \{1, ..., k-1\}$ is fulfilled. Then, for $\epsilon_1, ..., \epsilon_{k-1}$ positive and sufficiently small, there exists a manifold $\mathcal{L}_{(\epsilon_1, ..., \epsilon_{n-1})}$ locally invariant under the flow of system (5.5). Moreover, $\mathcal{L}_{(\epsilon_1, ..., \epsilon_{n-1})}$ has Hausdorff distance $\mathcal{O}(\epsilon_1, ..., \epsilon_{k-1})$ from $\mathcal{L}$, and it is diffeomorphic to $\mathcal{L}$. Also, the flow on $\mathcal{L}_{(\epsilon_1, ..., \epsilon_{n-1})}$ converges to the flow of system (5.11) as $\epsilon_j \to 0$, $j \in \{1, ..., k-1\}$.*

For $k = 1$, hypothesis (H) implies that each point $\boldsymbol{z}_0 \in \mathcal{N} \subset \mathcal{M}_{n-1} \subset \mathcal{M}_1$ is a hyperbolic fixed point of the layer problem (5.13). Let $W^s(\boldsymbol{z}_0)$ and $W^u(\boldsymbol{z}_0)$ be the local stable and unstable manifolds of $\boldsymbol{z}_0 \in \mathcal{N}$, respectively. The local stable and unstable manifolds of $\mathcal{N}$ are given by

$$W^s(\mathcal{N}) = \bigcup_{\boldsymbol{z}_0 \in \mathcal{N}} W^s(\boldsymbol{z}_0) \quad \text{and} \quad W^u(\mathcal{N}) = \bigcup_{\boldsymbol{z}_0 \in \mathcal{N}} W^u(\boldsymbol{z}_0) \tag{5.17}$$

Theorem 5.9. *Under the hypotheses of theorem 5.7, for $\epsilon_1, ..., \epsilon_{n-1}$ positive and sufficiently small, there exist local stable and unstable manifolds $W^s(\mathcal{N}_{\boldsymbol{\epsilon}})$ and $W^u(\mathcal{N}_{\boldsymbol{\epsilon}})$ which are locally invariant under the flow of system (5.5). They are C^r smooth for any $r < \infty$, and they are diffeomorphic to $W^s(\mathcal{N})$ and $W^u(\mathcal{N})$, respectively. Moreover, $W^s(\mathcal{N}_{\boldsymbol{\epsilon}})$ and $W^u(\mathcal{N}_{\boldsymbol{\epsilon}})$ have Hausdorff distance $\mathcal{O}(\epsilon_1 + ... + \epsilon_{n-1})$ from $W^s(\mathcal{N})$ and $W^u(\mathcal{N})$, respectively.*

Supposing that $W^s(\mathcal{N})$ and $W^u(\mathcal{N})$ have dimensions $j_1^s + m_2 + ... + m_n$ and $j_1^u + m_2 + ... + m_n$, respectively, with $j_1^s + j_1^u = m_1$.

Theorem 5.10. *Under the hypotheses of theorem 5.7, for every $\boldsymbol{z}_{\boldsymbol{\epsilon}} \in \mathcal{N}_{\boldsymbol{\epsilon}}$, with $\epsilon_1, ..., \epsilon_{n-1}$ positive and sufficiently small, there are j_1^s-dimensional and j_1^u-dimensional manifolds $W^s(\boldsymbol{z}_{\boldsymbol{\epsilon}}) \subset W^s(\mathcal{N}_{\boldsymbol{\epsilon}})$ and $W^u(\boldsymbol{z}_{\boldsymbol{\epsilon}}) \subset W^u(\mathcal{N}_{\boldsymbol{\epsilon}})$ that have Hausdorff distance $\mathcal{O}(\epsilon_1 + ... + \epsilon_{n-1})$ from $W^s(\boldsymbol{z}_0)$ and $W^u(\boldsymbol{z}_0)$, respectively. Moreover, they are C^r smooth for any $r < \infty$, and the families $\{W^s(\boldsymbol{z}_{\boldsymbol{\epsilon}}) : \boldsymbol{z}_{\boldsymbol{\epsilon}} \in \mathcal{N}_{\boldsymbol{\epsilon}}\}$ and $\{W^u(\boldsymbol{z}_{\boldsymbol{\epsilon}}) : \boldsymbol{z}_{\boldsymbol{\epsilon}} \in \mathcal{N}_{\boldsymbol{\epsilon}}\}$ are invariant in the sense that*

$$W^s(\boldsymbol{z}_{\boldsymbol{\epsilon}}).t \subset W^s(\boldsymbol{z}_{\boldsymbol{\epsilon}}.t) \quad \text{and} \quad W^u(\boldsymbol{z}_{\boldsymbol{\epsilon}}).t \subset W^u(\boldsymbol{z}_{\boldsymbol{\epsilon}}.t)$$

for all $t \geq 0$ and $t \leq 0$, respectively.

Theorem 5.11. *Consider a C^∞ family like (5.5), with $n \geq 2$. Let $\mathcal{N} \subset \mathcal{M}_{n-1}$ be a j-dimensional compact invariant manifold of the reduced problem (5.15) with a $(j + j_n^s)$-dimensional local stable manifold W^s and a $(j + j_n^u)$-dimensional local unstable manifold W^u. Assume that hypothesis (H) is fulfilled. Then there are positive real numbers $\delta_1, ..., \delta_{n-1}$ such that the following hold:*

- *There exists a family of smooth manifolds $\{\mathcal{N}_{(\epsilon_1,...,\epsilon_{n-1})} : \epsilon_i \in (0, \delta_i), i \in \{1, ..., n-1\}\}$ such that $\mathcal{N}_{(\epsilon_1,...,\epsilon_{n-1})} \to \mathcal{N}$ when $\boldsymbol{\epsilon} \to 0$, according to the Hausdorff distance, and $\mathcal{N}_{(\epsilon_1,...,\epsilon_{n-1})}$ is a hyperbolic invariant manifold of (5.5).*
- *There are families of $(j + j_1^s + ... + j_n^s)$-dimensional and $(j + j_1^u + ... + j_n^u)$-dimensional smooth manifolds $\{\mathcal{W}^s_{(\epsilon_1,...,\epsilon_{n-1})} : \epsilon_i \in (0, \delta_i), i \in \{1, ..., n-1\}\}$ and $\{\mathcal{W}^u_{(\epsilon_1,...,\epsilon_{n-1})} : \epsilon_i \in (0, \delta_i), i \in \{1, ..., n-1\}\}$ such that the manifolds $\mathcal{W}^s_{(\epsilon_1,...,\epsilon_{n-1})}$ and $\mathcal{W}^u_{(\epsilon_1,...,\epsilon_{n-1})}$ are local stable and unstable manifolds of $\mathcal{N}_{(\epsilon_1,...,\epsilon_{n-1})}$, respectively.*

5.3 A tropical Scaling Procedure

We define here a method to get the scaling of a system in order to apply the previous theorems. We consider CRN models described by the system of ODEs (2.2). The goal is to obtain from this system, the system (5.5) that is the starting point of the reduction.

Supposing that the parameters $\boldsymbol{k}$ have a numeric value $\boldsymbol{k}^*$, we can choose a new parameter $\epsilon_* \in]0, 1[$, and rescale the model parameters $\boldsymbol{k}^*$ by powers of ϵ_*:

$$k_i^* = \bar{k}_i^* \epsilon_*^{\gamma_i}, \tag{5.18}$$

where the exponents $\gamma_i \in \mathbb{Q}$. Furthermore, the prefactors $\bar{k}_i^*$ have the order $\mathcal{O}(\epsilon_*^0)$, more precisely

$$(\epsilon_*)^{-\eta} \leq \bar{k}_i^* < (\epsilon_*)^{\eta}, \tag{5.19}$$

for $1 \leq i \leq r$, where η is a positive parameter much smaller than one.

A possible choice of the exponents is

$$\gamma_i = \frac{\text{round}(p \log_{\epsilon_*}(k_i^*))}{p} \in \mathbb{Q}, \tag{5.20}$$

where p is a strictly positive integer controlling the precision of the rounding step. This choice leads to prefactors satisfying (5.19) with $\eta = 1/(2p)$.

We further rescale the variables $x_k = y_k \epsilon_*^{d_k}$, where $d_k \in \mathbb{Q}$, and transform (2.2) into the rescaled system

$$S_{\epsilon_*} \quad := \quad \{\dot{y}_i = \sum_{j=1}^{r} \epsilon_*^{\gamma_j + \langle \boldsymbol{d}, \boldsymbol{\alpha}_j \rangle - d_i} S_{ij} \bar{k}_j^* \boldsymbol{y}^{\boldsymbol{\alpha}_j} \mid 1 \leq i \leq n\} \tag{5.21}$$

with $\boldsymbol{d} = (d_1, d_2, \ldots, d_n)$.

The parameter and concentration orders γ_i and d_k should be understood as orders of magnitude. For instance, if $\epsilon = 1/10$, a parameter or concentration of order $d = 2$ equals roughly 10^{-2}. Thus, small orders mean large parameters or concentration values.

From now on, we will transform the numerical parameters $\epsilon_*, \boldsymbol{k}^*$ into variables $\epsilon, \boldsymbol{k} = (k_1, \ldots, k_r)$ such that $k_i = \epsilon^{\gamma_i} \bar{k}_i$ and consider the family of ODE systems indexed by ϵ

$$S_\epsilon := \{\dot{y}_i = \sum_{j=1}^{r} \epsilon^{\gamma_j + \langle \boldsymbol{d}, \boldsymbol{\alpha}_j \rangle - d_i} S_{ij} \bar{k}_j \boldsymbol{y}^{\alpha_j}, 1 \leq i \leq n\} \tag{5.22}$$

Our initial model S_{ϵ_*} is a member of this family, obtained for $\epsilon = \epsilon_*, \boldsymbol{k} = \boldsymbol{k}^*$. We are interested in characterizing the behaviour of the solutions of this family of ODEs, when $\epsilon \to 0$. This limit will correspond to a reduced model that is a good approximation to the initial model if ϵ^* is close to zero for $\boldsymbol{k} = \boldsymbol{k}^*$ and if several conditions ensuring the convergence of the solutions of S_ϵ are satisfied.

As we require $\epsilon_1, ..., \epsilon_{n-1} \in]0, 1[$ in the theorems, we need that the powers of ϵ (that factorise the function) are positives. Then, we call $\psi_{ij} = \gamma_j + \langle \boldsymbol{d}, \boldsymbol{\alpha}_j \rangle - d_i$ and $\mu = \min\{\psi_{ij} \mid 1 \leq i \leq n, 1 \leq j \leq r,\ S_{ij} \neq 0\}$. By performing a time scaling $\tau = \epsilon^\mu t$, we obtain the following equations for $i \in \{1, ..., n\}$

$$y_i' = \sum_{j=1}^{r} \epsilon^{\psi_{ij} - \mu} S_{ij} \bar{k}_j \boldsymbol{y}^{\boldsymbol{\alpha}_j}.$$

Factorising each equation by their minimal epsilon power, we obtain

$$y_i' = \epsilon^{\psi_i - \mu} (\sum_{\psi_{ij} = \psi_i} S_{ij} \bar{k}_j \boldsymbol{y}^{\boldsymbol{\alpha}_j} + \sum_{\psi_{ij} \neq \psi_i} S_{ij} \bar{k}_j \epsilon^{\psi_{ij} - \psi_i} \boldsymbol{y}^{\boldsymbol{\alpha}_j}), \tag{5.23}$$

when $\psi_i = \min\{\psi_{ij} \mid 1 \leq j \leq r,\ S_{ij} \neq 0\}$ and y' denote the derivative w.r.t. τ, and all power of epsilon present in the equations are positives.

To obtain positives and integers powers, we define $\delta = \epsilon^{1/o}$ where $o \in \mathbb{N}$ is the smallest common multiple of the denominators of $\psi_{ij} - \mu$ (for all i and j), this is possible since we have a finite number of monomials. Then, (5.23) becomes

$$y_i' = \delta^{b_i} (\bar{f}_i^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{y}) + \delta^{b_i'} \bar{f}_i^{(2)}(\bar{\boldsymbol{k}}, \boldsymbol{y}, \delta)), \tag{5.24}$$

where

$$\bar{f}_i^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{y}) = \sum_{\psi_{ij}=\psi_i} S_{ij}\bar{k}_j \boldsymbol{y}^{\boldsymbol{\alpha}_j},$$

$$\bar{f}_i^{(2)}(\bar{\boldsymbol{k}}, \boldsymbol{y}, \delta) = \sum_{\psi_{ij}\neq\psi_i} S_{ij}\bar{k}_j \delta^{b'_{ij}} \boldsymbol{y}^{\boldsymbol{\alpha}_j},$$

and all the powers of δ are positive integers, as follows: $b_i = o(\psi_i - \mu)$, $b'_i = o\min\{\psi_{ij} - \psi_i \mid 1 \leq j \leq r, S_{ij} \neq 0, \psi_{ij} \neq \psi_i\} > 0$, $b'_{ij} = o(\psi_{ij} - \psi_i) - b'_i > 0$, $1 \leq i \leq n$.

We call **truncated system**, the system obtained by retaining only the minimal order, dominant terms in (5.24)

$$y'_i = \delta^{b_i} \bar{f}_i^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{y}),\ 1 \leq i \leq n. \tag{5.25}$$

Let us define the **truncated stoichiometric matrix $\boldsymbol{S}^{(1)}$** whose entries are

$$S_{ij}^{(1)} = \begin{cases} S_{ij} & \text{if } \psi_{ij} = \psi_i \\ 0 & \text{if not} \end{cases} \tag{5.26}$$

Then, it follows

$$\bar{f}_i^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{y}) = \sum_{j=1}^{r} S_{ij}^{(1)} \bar{k}_j \boldsymbol{y}^{\boldsymbol{\alpha}_j}.$$

For several calculations it is convenient to return to the variables $(\boldsymbol{x}, t)$. In these variables, the truncated system reads

$$\dot{x}_i = f_i^{(1)}(\boldsymbol{k}, \boldsymbol{x}),\ 1 \leq i \leq n, \tag{5.27}$$

where

$$f_i^{(1)}(\boldsymbol{k}, \boldsymbol{x}) = \delta^{od_i + b_i + o\mu} \bar{f}_i^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{y}) = \sum_{j=1}^{r} S_{ij}^{(1)} k_j \boldsymbol{x}^{\alpha_j}.$$

By definition, $\min\{\psi_i \mid 1 \leq i \leq n\} - \mu = 0$. Therefore $\min\{b_i \mid 1 \leq i \leq n\} = 0$ and up to a relabelling of the variables y_i one can consider that $b_1 = 0 \leq b_2 \leq \ldots \leq b_n$.

From (5.24) it follows that each rescaled variable y_i have significant changes of order $\mathcal{O}(\delta^0)$ on a time scale of order $\mathcal{O}(\delta^{-b_i})$. Therefore, the powers δ^{b_i} indicate the time scales of the variables y_i: the most rapid is y_1 and the slowest is y_n. Of course, several variables can have the same time scale order, i.e. the same value of b_i. Let us regroup the variables y_i into vectors $\boldsymbol{z}_k$ where $\boldsymbol{z}_k = (y_{i_k}, y_{i_k+1}, \ldots, y_{i_k+n_k-1})$ regroups all variables such that $b_{i_k} = b_{i_k+1} = \ldots = b_{i_k+n_k-1} = b_k$, where $n_1 + n_2 + \ldots + n_m = n$. We then obtain

$$\boldsymbol{z}'_k = \delta^{b_k}(\bar{\boldsymbol{f}}_k^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}) + \delta^{b'_k} \bar{\boldsymbol{f}}_k^{(2)}(\bar{\boldsymbol{k}}, \boldsymbol{z}, \delta)), \tag{5.28}$$

where $b_1 = 0 < b_2 < \ldots < b_m$, $b'_k > 0$, $\bar{\boldsymbol{f}}_k^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}) \in \mathbb{Z}[\bar{\boldsymbol{k}}, \boldsymbol{z}]$, $\bar{\boldsymbol{f}}_k^{(2)}(\bar{\boldsymbol{k}}, \boldsymbol{z}, \delta) \in \mathbb{Z}[\bar{\boldsymbol{k}}, \boldsymbol{z}, \delta]$.

We call **truncated system at the fastest time** the system obtained by setting $\delta = 0$ in (5.28)

$$\begin{aligned} \boldsymbol{z}'_1 &= \bar{\boldsymbol{f}}_1^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}), \\ \boldsymbol{z}'_2 &= 0, \\ &\vdots \\ \boldsymbol{z}'_m &= 0, \end{aligned} \tag{5.29}$$

By redefining again the time to $\tau' = \tau\delta^{b_l}$, where $2 \leq l \leq m$ we get

$$\begin{aligned}
\delta^{q_1}\boldsymbol{z}_1' &= (\bar{\boldsymbol{f}}_1^{(1)}(\bar{\boldsymbol{k}},\boldsymbol{z}) + \delta^{b_1'}\bar{\boldsymbol{f}}_1^{(2)}(\bar{\boldsymbol{k}},\boldsymbol{z},\delta)),\\
&\vdots\\
\delta^{q_{l-1}}\boldsymbol{z}_{l-1}' &= (\bar{\boldsymbol{f}}_{l-1}^{(1)}(\bar{\boldsymbol{k}},\boldsymbol{z}) + \delta^{b_{l-1}'}\bar{\boldsymbol{f}}_{l-1}^{(2)}(\bar{\boldsymbol{k}},\boldsymbol{z},\delta)),\\
\boldsymbol{z}_l' &= (\bar{\boldsymbol{f}}_l^{(1)}(\bar{\boldsymbol{k}},\boldsymbol{z}) + \delta^{b_l'}\bar{\boldsymbol{f}}_l^{(2)}(\bar{\boldsymbol{k}},\boldsymbol{z},\delta)),\\
\boldsymbol{z}_{l+1}' &= \delta^{q_{l+1}}(\bar{\boldsymbol{f}}_{l+1}^{(1)}(\bar{\boldsymbol{k}},\boldsymbol{z}) + \delta^{b_{l+1}'}\bar{\boldsymbol{f}}_{l+1}^{(2)}(\bar{\boldsymbol{k}},\boldsymbol{z},\delta)),\\
&\vdots\\
\boldsymbol{z}_m' &= \delta^{q_m}(\bar{\boldsymbol{f}}_m^{(1)}(\bar{\boldsymbol{k}},\boldsymbol{z}) + \delta^{b_m'}\bar{\boldsymbol{f}}_m^{(2)}(\bar{\boldsymbol{k}},\boldsymbol{z},\delta)),
\end{aligned} \tag{5.30}$$

where $0 < q_{l+1} = b_{l+1} - b_l < \ldots < q_m = b_m - b_l$, $q_1 = b_l - b_1 > q_2 = b_l - b_2 > \ldots > q_{l-1} = b_l - b_{l-1} > 0$.

We call **truncated system at the lth fastest time** the system obtained by setting $\delta = 0$ in (7.5)

$$\begin{aligned}
0 &= \bar{\boldsymbol{f}}_1^{(1)}(\bar{\boldsymbol{k}},\boldsymbol{z}),\\
&\vdots\\
0 &= \bar{\boldsymbol{f}}_{l-1}^{(1)}(\bar{\boldsymbol{k}},\boldsymbol{z}),\\
\boldsymbol{z}_l' &= \bar{\boldsymbol{f}}_l^{(1)}(\bar{\boldsymbol{k}},\boldsymbol{z}),\\
\boldsymbol{z}_{l+1}' &= 0,\\
&\vdots\\
\boldsymbol{z}_m' &= 0.
\end{aligned} \tag{5.31}$$

It should be noted that our scaling, to be used by the previous theorems, needs to satisfy some conditions. Indeed, we need that fast species satisfy QE or QSS conditions. So, to be used, our scaling needs to satisfy a partial or total tropical equilibration.

Importance of Concentration Valuations for the Scaling

In many studies, valuations are computed for, and scalings are applied only to parameters. This is because concentration are unknown and in this case it is handy to consider even concentration valuations. However, rather systematically, this leads to flaws because species concentrations have not even valuations in general.

To illustrate this common mistake, let us consider the following example, adapted from Schneider and Wilhelm, 2000.

Example 5.12. Let us consider the following mass action CRN

$$\emptyset \xrightarrow{1} A_2 \xrightarrow{1/\epsilon} A_1 \xrightarrow{1} \emptyset, \quad A_2 + A_2 \xrightarrow{1/\epsilon} \emptyset,$$

with explicit valuations for the parameters.

The corresponding ODEs are

$$\dot{x}_1 = x_2/\epsilon - x_1, \quad \dot{x}_2 = -x_2/\epsilon - x_2^2/\epsilon + 1.$$

The model does not have exact conservation laws.

By imposing total tropical equilibration to this model we get

$$d_2 - 1 = d_1, \quad \min(d_2 - 1, 2d_2 - 1) = 0.$$

These equations have a unique solution $d_1 = 0$, $d_2 = 1$, meaning that the valuations of x_1 and x_2 are different.

The corresponding scaling is $x_1 = y_1$, $x_2 = y_2\epsilon$ and the rescaled ODEs read

$$\dot{y}_1 = y_2 - y_1, \quad \dot{y}_2 = \epsilon^{-1}(-y_2 - \epsilon y_2^2 + 1).$$

This scaling shows that y_1 and y_2 are slow and fast variables, respectively. The truncated system at fastest time scale is

$$\dot{y}_2 = \epsilon^{-1}(-y_2 + 1),$$

and has a unique hyperbolic steady state $y_2 = 1$.

This model is not an example of approximate conservation and standard singular perturbation theory results (quasi-steady state approximation) can be applied for its reduction.

Notice that [Schneider and Wilhelm, 2000] used non-scaled concentrations. By doing so, the truncated system is $\dot{x}_1 = x_2/\epsilon, \quad \dot{x}_2 = -x_2/\epsilon - x_2^2/\epsilon$, where both variables x_1 and x_2 seem fast. Using the assumption of even concentration valuations also leads to the nonlinear approximate conservation law $\phi = x_1 + \log(1 + x_2)$ (see [Schneider and Wilhelm, 2000]).

Chapter 6

Approximated conservation laws for model reduction

6.1 Exact and approximate conservation laws

In order to introduce the notions of exact and approximate conservation laws we use an example.

Example 6.1. Let us consider again the irreversible Michaelis-Menten mechanism that is paradigmatic for enzymatic reactions. We choose rate constants corresponding to the so-called quasi-equilibrium, studied by Michaelis and Menten. The reaction network for this model is

$$S + E \underset{k_2}{\overset{k_1}{\rightleftharpoons}} ES \xrightarrow{k_3\delta} E + P$$

where S is a substrate, E is an enzyme, ES is an enzyme-substrate complex and k_1,k_2,k_3 are rate constants. Here $0 < \delta < 1$ is a small positive scaling parameter, indicating that the third rate constant is small.

According to the mass-action kinetics, the concentrations $x_1 = [S]$, $x_2 = [SE]$ and $x_3 = [E]$ satisfy the system of ODEs

$$\begin{aligned} \dot{x_1} &= -k_1x_1x_3 + k_2x_2 \\ \dot{x_2} &= k_1x_1x_3 - k_2x_2 - \delta k_3x_2 \\ \dot{x_3} &= -k_1x_1x_3 + k_2x_2 + \delta k_3x_2. \end{aligned} \tag{6.1}$$

We consider the truncated system of ODEs

$$\begin{aligned} \dot{x_1} &= -k_1x_1x_3 + k_2x_2 \\ \dot{x_2} &= k_1x_1x_3 - k_2x_2 \\ \dot{x_3} &= -k_1x_1x_3 + k_2x_2 \end{aligned} \tag{6.2}$$

which is obtained by setting $\delta = 0$ in (6.1). The truncated system (6.2) describes the dynamics of the model on fast time scales of order $O(\delta^0)$.

The steady state of the fast dynamics is obtained by equating to zero the r.h.s. of (6.2). The resulting condition is called quasi-equilibrium (QE) because it means that the complex formation rate $k_1x_1x_3$ is equal to the complex dissociation rate k_2x_2. In other words, the reversible reaction $S + E \underset{k_2}{\overset{k_1}{\rightleftharpoons}} ES$ functions at equilibrium. The QE condition is reached only at the end of the fast dynamics, and is satisfied with a precision of order $O(\delta)$ during the slow dynamics [Gorban et al., 2010]. Because of its approximate validity, QE is different from the similar concept of detailed balance [Boltzmann, 1964].

We introduce the linear combinations of variables $x_4 = x_1 + x_2$ and $x_5 = x_2 + x_3$, corresponding to the total substrate and total enzyme concentrations, respectively.

Addition of the last two equations of (6.1) leads to $\dot{x_5} = 0$ which means that for solutions of the full system x_5 is constant for all times. We will call such a quantity "exact conservation law".

Additions of the first two and the last two equations of (6.2) lead to $\dot{x_4} = 0$, and $\dot{x_5} = 0$. This means that x_4 is constant for solutions of the truncated dynamics, valid at short times $t = O(\delta^0)$, and is not constant at larger times $t = O(\delta^{-1})$. We call such a quantity "approximate conservation law". The quantity x_5 is both an exact and approximate conservation law.

Definition 6.2.

1. A function $\phi(\boldsymbol{x})$ is an **exact conservation law** unconditionally on the parameters, if it is a first integral of the full system (2.2), i.e. if

$$\sum_{i=1}^{n} \frac{\partial \phi}{\partial x_i}(\boldsymbol{x}) f_i(\boldsymbol{k}, \boldsymbol{x}) = 0$$

   for all $\boldsymbol{k}$ and $\boldsymbol{x}$.

2. A function $\phi(\boldsymbol{x})$ is an **approximate conservation law** unconditionally on the parameters, if it is a first integral of the truncated system (5.27), i.e. if

$$\sum_{i=1}^{n} \frac{\partial \phi}{\partial x_i}(\boldsymbol{x}) f_i^{(1)}(\boldsymbol{k}, \boldsymbol{x}) = 0$$

   for all $\boldsymbol{k}$, $\boldsymbol{x}$.

3. An exact (approximate) conservation law of the form $c_1x_1 + \cdots + c_nx_n$ with coefficients $c_i \in \mathbb{R}$ is called an **exact** (**approximate**) **linear conservation law**. If $c_i \geq 0$ for $1 \leq i \leq n$, the linear conservation law is called **semi-positive**.

4. An exact (approximate) conservation law of the form $x_1^{m_1} \cdots x_n^{m_n}$ with $m_i \in \mathbb{Z}$ is called an **exact** (**approximate**) **rational monomial conservation law**. If $m_i \in \mathbb{Z}_+$ for $1 \leq i \leq n$, the conservation law is called **monomial**. For simplicity, in this paper, we will call both types **monomial**.

5. An exact (approximate) conservation law of the form $\sum_{i=1}^{s} a_i x_1^{m_{1i}} \cdots x_n^{m_{ni}}$ with $m_{ji} \in \mathbb{Z}_+$ and $a_i \in \mathbb{R}$ is called an **exact** (**approximate**) **polynomial conservation law**.

Some variables may not appear in a conservation law $\phi(\boldsymbol{x})$. This means that in case of a linear conservation law some coefficients c_i may be zero, or in case of a nonlinear conservation law some partial derivatives $\frac{\partial \phi}{\partial x_i}(\boldsymbol{x})$ may vanish. If r is the number of all non-zero quantities $\frac{\partial \phi}{\partial x_i}(\boldsymbol{x})$, then we say that the conservation law **depends on** r **variables**. Sometimes, conservation laws depends on the parameter, in this case, we will note $\phi(\boldsymbol{k}, \boldsymbol{x})$ instead of $\phi(\boldsymbol{x})$, it is generally the case when we are searching for monomial or polynomial conservation laws.

Definition 6.3. An exact or approximated conservation law depending on r variables is called **reducible** if it can be split into the sum or the product of two conservation laws such that at least one of them depends on a number of variables r' with $1 \leq r' < r$. Otherwise it is called **irreducible**.

Definition 6.4. For $\boldsymbol{k} \in \mathbb{R}_+^r$ a steady state $\boldsymbol{x}$ is a solution of $\boldsymbol{F}(\boldsymbol{k}, \boldsymbol{x}) = 0$ and we denote the **steady state variety** by $\mathcal{S}_{\boldsymbol{k}}$. A steady state is **degenerate** or **non-degenerate** if the Jacobian $D_{\boldsymbol{x}}\boldsymbol{F}(\boldsymbol{k}, \boldsymbol{x})$ is singular, or non-singular, respectively.

Degeneracy of steady states usually results from the continuity of $\mathcal{S}_{\boldsymbol{k}}$.

Theorem 6.5. *Assume that for $\boldsymbol{k} \in \mathbb{R}_+^r$, $\mathcal{S}_{\boldsymbol{k}}$ is a manifold. If the local dimension at a point $\boldsymbol{x}_0 \in \mathcal{S}_{\boldsymbol{k}}$ is strictly positive, then $\boldsymbol{x}_0$ is degenerate.*

Proof. There is a neighborhood U of zero and a smooth function $U \to \mathcal{S}_{\boldsymbol{k}}$, $\alpha \mapsto \boldsymbol{x}(\alpha)$ with $\boldsymbol{x}(0) = \boldsymbol{x}_0$. By differentiating in $\boldsymbol{F}(\boldsymbol{k}, \boldsymbol{x}(\alpha)) = 0$ with respect to α at 0 we get $D_{\boldsymbol{x}}\boldsymbol{F}(\boldsymbol{k}, \boldsymbol{x}_0)\frac{d\boldsymbol{x}(0)}{d\alpha} = 0$, and thus $D_{\boldsymbol{x}}\boldsymbol{F}(\boldsymbol{k}, \boldsymbol{x}_0)$ is singular. □

Definition 6.6. A set

$$\boldsymbol{\Phi}(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}), \ldots, \phi_s(\boldsymbol{x}))^{\mathsf{T}}$$

of exact conservation laws is called **complete** if the Jacobian matrix

$$\boldsymbol{J}_{\boldsymbol{F},\boldsymbol{\Phi}}(\boldsymbol{k}, \boldsymbol{x}) = D_{\boldsymbol{x}}(\boldsymbol{F}(\boldsymbol{k}, \boldsymbol{x}), \boldsymbol{\Phi}(\boldsymbol{x}))^{\mathsf{T}}$$

has rank n for any $\boldsymbol{k} \in \mathbb{R}_+^r, \boldsymbol{x} \in \mathbb{R}_+^n$ satisfying $\boldsymbol{F}(\boldsymbol{k}, \boldsymbol{x}) = 0$. The set is called **independent** if the Jacobian matrix of $\boldsymbol{\Phi}(\boldsymbol{x})^{\mathsf{T}}$ with respect to $\boldsymbol{x}$ has rank s for any $\boldsymbol{k} \in \mathbb{R}_+^r, \boldsymbol{x} \in \mathbb{R}_+^n$ such that $\boldsymbol{F}(\boldsymbol{k}, \boldsymbol{x}) = 0$. In the case of a set $\boldsymbol{\Phi}(\boldsymbol{x})$ of approximate conservation laws completeness is defined with $\boldsymbol{F}(\boldsymbol{k}, \boldsymbol{x})$ replaced by

$$\boldsymbol{F}^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{x}) = (\delta^{b_1}\bar{f}_1^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{x}), \ldots, \delta^{b_n}\bar{f}_n^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{x}))^{\mathsf{T}}$$

where δ is treated as a parameter.

Proposition 6.7. *If a system has a complete set of conservation laws, then the intersection of $\mathcal{S}_{\boldsymbol{k}}$ with $\{\boldsymbol{\Phi}(\boldsymbol{x}) = \boldsymbol{c}_0\} \cap \mathbb{R}_+^n$, where $\boldsymbol{c}_0 \in \mathbb{R}^r$ is finite.*

Proof. Suppose that the intersection is non-empty and contains $\boldsymbol{x}$. Because the rank of $\boldsymbol{J}_{\boldsymbol{F},\boldsymbol{\phi}}$ in $\boldsymbol{x}$ is n, the implicit function theorem implies that, for $\boldsymbol{k}$ and $\boldsymbol{c}_0$, $\boldsymbol{x}$ is isolated from other solutions of $\boldsymbol{F}(\boldsymbol{k}, \boldsymbol{x}) = 0$, $\boldsymbol{\Phi}(\boldsymbol{x}) = \boldsymbol{c}_0$. As $\boldsymbol{F}$ is polynomial, the intersection is finite. □

Remark 6.8. When $\boldsymbol{\Phi}(\boldsymbol{x})$ is linear in $\boldsymbol{x}$ and results from a stoichiometric matrix, the set $\{\boldsymbol{\Phi}(\boldsymbol{x}) = \boldsymbol{c}_0\} \cap \mathbb{R}^n_+$ is called **stoichiometric compatibility class**, or **reaction simplex** [Wei and Prater, 1962, Feinberg and Horn, 1974]. Stoichiometric compatibility classes of systems with complete sets of linear conservation laws contain a finite number of steady states. We note that some authors call completeness of linear conservation laws "non-degeneracy" [Feliu and Wiuf, 2012].

Remark 6.9. Because our concern is the number of positive solutions of $\boldsymbol{F}(\boldsymbol{k}, \boldsymbol{x}) = 0$, $\boldsymbol{\Phi}(\boldsymbol{x}) = \boldsymbol{c}_0$, in Definition 6.6 it would be more natural to consider the rank of $\boldsymbol{J}_{\boldsymbol{F},\boldsymbol{\phi}}(\boldsymbol{k}, \boldsymbol{x})$ on $\mathcal{S}_{\boldsymbol{k}} \cap \{\boldsymbol{\Phi}(\boldsymbol{x}) = \boldsymbol{c}_0\} \cap \mathbb{R}^n_+$. In fact, as this rank does not depend on $\boldsymbol{c}_0$, it is simpler and equivalent to impose its value on $\mathcal{S}_{\boldsymbol{k}} \cap \mathbb{R}^n_+$.

Remark 6.10. The independent linear conservation laws

$$\boldsymbol{\Phi}(\boldsymbol{x}) = (x_1 + x_2, x_2 + x_3)$$

of Example 6.1 are complete. More precisely, the Jacobian of $(\boldsymbol{F}(\boldsymbol{x}), \boldsymbol{\Phi}(\boldsymbol{x}))^\mathsf{T}$, where $\boldsymbol{F}(\boldsymbol{x})$ is the vector of right hand sides of (6.2), has a 3×3 minor

$$M := \det(D_{\boldsymbol{x}}(-k_1 x_1 x_3 + k_2 x_2, \boldsymbol{\Phi}(\boldsymbol{x}))^\mathsf{T}) = -k_2 - k_1 x_1 - k_1 x_3.$$

This minor can not be zero for positive $\boldsymbol{x}$, $\boldsymbol{k}$, on the steady state variety defined by the equation $-k_1 x_1 x_3 + k_2 x_2 = 0$, therefore the rank of $\boldsymbol{J}_{\boldsymbol{F},\boldsymbol{\phi}}$ is three.

Furthermore, all stoichiometric compatibility classes defined by $x_1 + x_2 = c_{01}, x_2 + x_3 = c_{02}, \boldsymbol{x} > 0$ contain a unique steady state

$$\begin{aligned} x_1 &= \big(k_1(c_{01} - c_{02}) - k_2 + \sqrt{\Delta}\big)/(2k_1), \\ x_2 &= \big(k_1(c_{01} + c_{02}) + k_2 - \sqrt{\Delta}\big)/(2k_1), \\ x_3 &= \big(k_1(c_{02} - c_{01}) - k_2 + \sqrt{\Delta}\big)/(2k_1), \end{aligned}$$

where $\Delta = (c_{01} - c_{02})^2 k_1^2 + k_2^2 + 2k_1 k_2 (c_{01} + c_{02})$.

The notions of completeness and independence in Definition 6.6 are effective. Algorithm 17 tests for completeness. It uses in l.1 a parametric rank computation, which yields for a matrix A with parametric entries a set $R = (\Gamma_i, r_i)_{i=1,\ldots,n}$ of pairs. For each $i = 1, \ldots, n$, Γ_i is a conjunction of constraints of the form $p = 0$ or $p \neq 0$ where p is a polynomial in the parameters, and $r_i \in \mathbb{N}$. Let $S_1, \ldots, S_n$ be the semialgebraic sets defined by $\Gamma_1, \ldots, \Gamma_n$, respectively. Then the following holds:

1. $S_1, \ldots, S_n$ form a finite partitioning of real parameter space.
2. Over each S_i, we have invariantly $\operatorname{rank}(A) = r_i$.

On these grounds, we construct in l.2–3 an equivalent logic condition ϱ_n for $\operatorname{rank}(\boldsymbol{J}_{\boldsymbol{F},\boldsymbol{\phi}}(\boldsymbol{k}, \boldsymbol{x})) = n$. In l.4 we construct γ as a direct formalization of the definition of completeness. In l.5 we finally test validity of γ over the reals. Technically we use a combination of various effective quantifier elimination procedures for the theory of real closed fields [Dolzmann et al., 1998, Sturm, 2017, Sturm, 2018, and the references there] combined with heuristic simplification techniques [Dolzmann and Sturm, 1997b] in the Redlog system [Dolzmann and Sturm, 1997a, Košta, 2016, Seidl, 2006]. Alternatively, one could use Satisfiability Modulo Theories solving over the logic `QF_NRA` [Nieuwenhuis et al., 2006, Barrett et al., 2017, Ábrahám et al., 2016]. Algorithm 18 proceeds analogously to Algorithm 17 and tests for independence.

Input: 1. $\boldsymbol{F}(\boldsymbol{k},\boldsymbol{x}) = (f_1(\boldsymbol{k},\boldsymbol{x})), \ldots, f_n(\boldsymbol{k},\boldsymbol{x}))$; 2. $\boldsymbol{\Phi}(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}), \ldots \phi_s(\boldsymbol{x}))$;
3. $\boldsymbol{k} = (k_1, \ldots, k_r)$; 4. $\boldsymbol{x} = (x_1, \ldots, x_n)$
Output: "yes" if Φ is complete according to Definition 6.6, "no" otherwise

```
R := ParametricRank(J_{F,φ}(k, x))
R_n := { (Γ, r) ∈ R | r = n }
ϱ_n := ⋁_{(Γ,n)∈R_n} Γ
γ := ∀k∀x(k > 0 ∧ x > 0 ∧ F(x) = 0 ⟶ ϱ_n)
if ℝ ⊨ γ then
    return "yes"
else
    return "no"
end if
```

Algorithm 17: IsComplete

Input: 1. $\boldsymbol{F}(\boldsymbol{k},\boldsymbol{x}) = (f_1(\boldsymbol{k},\boldsymbol{x})), \ldots, f_n(\boldsymbol{k},\boldsymbol{x}))$; 2. $\boldsymbol{\Phi}(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}), \ldots \phi_s(\boldsymbol{x}))$;
3. $\boldsymbol{k} = (k_1, \ldots, k_r)$; 4. $\boldsymbol{x} = (x_1, \ldots, x_n)$
Output: "yes" if Φ is independent according to Definition 6.6, "no" otherwise

```
R := ParametricRank(D_x Φ(x)^T)
R_s := { (Γ, r) ∈ R | r = s }
ϱ_s := ⋁_{(Γ,s)∈R_s} Γ
ι := ∀k∀x(k > 0 ∧ x > 0 ∧ F(x) = 0 ⟶ ϱ_s)
if ℝ ⊨ ι then
    return "yes"
else
    return "no"
end if
```

Algorithm 18: IsIndependent

Example 6.11. We automatically process Example 6.1 with our Algorithm 17 IsComplete. Our input is

$$\boldsymbol{F}(\boldsymbol{k},\boldsymbol{x}) = (-k_1x_1x_3 + k_2x_2, k_1x_1x_3 - k_2x_2, -k_1x_1x_3 + k_2x_2),$$

$\boldsymbol{\Phi}(\boldsymbol{x}) = (x_1 + x_2, x_2 + x_3)$, $\boldsymbol{k} = (k_1, k_2)$, and $\boldsymbol{x} = (x_1, x_2, x_3)$. We obtain the parametric Jacobian

$$\boldsymbol{J}_{\boldsymbol{F},\boldsymbol{\phi}}(\boldsymbol{k},\boldsymbol{x}) = \begin{pmatrix} -k_1x_3 & k_2 & -k_1x_1 \\ k_1x_3 & -k_2 & k_1x_1 \\ -k_1x_3 & k_2 & -k_2x_1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

and compute in l.1 the parametric rank

$$R = \{(k_1x_1 + k_1x_3 + k_2 \neq 0, 3), (k_1x_1 + k_1x_3 + k_2 = 0, 2)\},$$

from which we select $R_3 = \{(k_1x_1 + k_1x_3 + k_2 \neq 0, 3)\}$ in l.2 and $\varrho_3 = (k_1x_1 + k_1x_3 + k_2 \neq 0)$ in l.3. Completeness is straightforwardly formalized in l.4 by

$$\gamma = \forall \boldsymbol{k} \forall \boldsymbol{x} (\boldsymbol{k} > 0 \wedge \boldsymbol{x} > 0 \wedge k_1x_1x_3 - k_2x_2 = 0 \longrightarrow k_1x_1 + k_1x_3 + k_2 \neq 0),$$

where some redundant equations are automatically removed from $\boldsymbol{F}(\boldsymbol{x}) = 0$ via the standard simplifier described in [Dolzmann and Sturm, 1997b]. Real quantifier elimination on γ in l.5 equivalently yields "true", which confirms completeness, and we return "yes" in l.6.

Similarly, our Algorithm 18 IsIndependent computes

$$D_{\boldsymbol{x}}\boldsymbol{\Phi}(\boldsymbol{x})^T = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}, \quad R = R_2 = \{(\text{true}, 2)\}, \quad \varrho_2 = \text{true}\,.$$

This yields $\iota = \forall \boldsymbol{k} \forall \boldsymbol{x} (\boldsymbol{k} > 0 \wedge \boldsymbol{x} > 0 \wedge k_1x_1x_3 - k_2x_2 = 0 \longrightarrow \text{true})$, which by quantifier elimination is equivalent to "true", and we return "yes". Recall that any implication with "true" on the right hand side holds already due to Boolean logic [Seidl and Sturm, 2003]. In our framework here, this corresponds to the observation that completeness and independence hold whenever we encounter full rank of the corresponding Jacobian for all choices of parameters.

The automatic computations described here take less than 0.01 s altogether.

The following example shows that linear conservation laws are not always complete.

Example 6.12. One checks easily that the system

$$\dot{x}_1 = 1 - x_1 - x_2, \quad \dot{x}_2 = x_1 + x_2 - 1$$

has the linear conservation law $\Phi(\boldsymbol{x}) = x_1 + x_2$ and that the Jacobian of $(\boldsymbol{F}(\boldsymbol{x}), \Phi(\boldsymbol{x}))^\mathsf{T}$ is constant and has rank 1. Trivially the Jacobian has everywhere rank 1 and so $\Phi(\boldsymbol{x})$ is not complete. The explicit solutions of the ODE system are $x_1(t) = (1 - c_0)t + c_1$, $x_2(t) = (c_0 - 1)t + c_0 - c_1$. One can easily show that all invariant curves are of the form $x_1 + x_2 = c_0$. We conclude that there are no further first integrals and so the system has no complete set of conservation laws. We can also note that the intersection of the steady state variety

$x_1 + x_2 = 1$ with a stoichiometric compatibility class $x_1 + x_2 = c_0$ is either empty or continuous.

In the absence of parameters $\boldsymbol{k}$ our Algorithm 17 IsComplete computes parametric rank $R = \{(\text{true}, 1)\}$. However, $n = 2$, which yields $R_2 = \emptyset$ and $\varrho_2 =$ false. This gives us $\gamma = \forall \boldsymbol{k} \forall \boldsymbol{x} (\text{true} \wedge \boldsymbol{x} > 0 \wedge x_1 + x_2 - 1 = 0 \longrightarrow \text{false})$, which quantifier elimination identifies to be "false", and the algorithm returns "no". Notice that γ is equivalent to $\forall \boldsymbol{x} (\boldsymbol{x} > 0 \longrightarrow x_1 + x_2 - 1 \neq 0)$, which illustrates that with deficient rank, completeness can only hold formally when there is no steady state in the positive orthant.

Parametric Rank Computation In the following we present in more general setting algorithms which will allow us to check for completeness and independence of a set of exact and approximate conservation laws. For this purpose let P, N, $M \in \mathbb{N}$ and let $\mathbb{R}[\boldsymbol{v}]$ be a polynomial ring in P indeterminates $\boldsymbol{v} = (v_1, \ldots, v_P)$. Further let $A \in \mathbb{R}[\boldsymbol{v}]^{M \times N}$ be a matrix whose entries are polynomials in $\boldsymbol{v}$. We are interested in the rank of A in dependence of different values $\bar{\boldsymbol{v}} \in \mathbb{R}^P$ for the variables $\boldsymbol{v}$. Algorithm (19) will provide a decomposition of $\mathbb{R}^P$ into disjoint sets such that for each set the rank of A is constant for all point $\bar{\boldsymbol{v}}$ of this set.

By a guard Γ we mean a conjunction of polynomial constraints in the indeterminates $\boldsymbol{v}$, i.e. polynomial equations and inequations in $\boldsymbol{v}$ with coefficients in $\mathbb{R}$.

6.2 Conservation laws as slow variables

As an exact conservation law is constant over time, one can think that an approximate conservation law is a slow variable, this is the case for (at least) any polynomial CRN model of the type (2.2) and for all linear, monomial, or polynomial approximated linear conservation laws, as the following theorems show.

Theorem 6.13. *Let $q = \sum_{i=1}^{n} c_i x_i$, where $c_i \in \mathbb{N}$, be an approximate linear conservation law, that is conserved by the truncated ODE system (5.27). Let $I = \{i | c_i \neq 0\}$, and consider that for all $i \in I$, $c_i = O(1)$. If q is irreducible, then the variable q is slower than the variables $\{x_i | i \in I\}$ of the system (2.2). Furthermore, if the timescales of x_i, $i \in I$ have the same order, then the concentrations of these variables have the same orders.*

Before starting the proof, let use state the following

Lemma 6.14. *Consider that $q = \sum_{i=1}^{n} c_i x_i$ is an irreducible linear conservation law. Then all the polynomials $f_i^{(1)}(\boldsymbol{k}, \boldsymbol{x})$ in the truncated system (5.27) have the same order in δ, i.e. $od_i + b_i = od_{i'} + b_{i'}$ for all $i, i' \in I$. If furthermore, the time scales of the variables x_i have the same order in δ for all $i \in I$, i.e. $b_i = b_{i'}$ for all $i, i' \in I$, then the concentrations x_i have also the same order in δ, i.e. $d_i = d_{i'}$ for all $i, i' \in I$.*

Proof of Lemma 6.14. In the $\boldsymbol{x}$ variables the truncated system reads $\dot{x}_i = \delta^{od_i} \dot{y}_i = f_i^{(1)}(\boldsymbol{k}, \boldsymbol{x})$, where $f_i^{(1)}(\boldsymbol{k}, \boldsymbol{x}) = \delta^{od_i + b_i + o\mu} \bar{f}_i^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{y})$ (see (5.27)). Since q is conserved by the ODE system (5.27) one has $\sum_{i=1}^{r} c_i f_i^{(1)}(\boldsymbol{k}, \boldsymbol{x}) = 0$, for all $\boldsymbol{k}$, $\boldsymbol{x}$. This can only be satisfied if for all $i \in I$ there is at least one $j \in I$ such that $f_i^{(1)}(\boldsymbol{k}, \boldsymbol{x})$ and $f_j^{(1)}(\boldsymbol{k}, \boldsymbol{x})$ have a common monomial. Since q is irreducible, for all $i, j \in I$, $i \neq j$, either $f_i^{(1)}(\boldsymbol{k}, \boldsymbol{x})$ and $f_j^{(1)}(\boldsymbol{k}, \boldsymbol{x})$ share a monomial,

Input: An $M \times N$-matrix $A(\boldsymbol{v})$ with polynomial entries in parameters $\boldsymbol{v}$.
Output: A set of pairs $\{(\Gamma_1(\boldsymbol{v}), r_1), \ldots, (\Gamma_I(\boldsymbol{v}), r_I)\}$, where each $\Gamma_i(\boldsymbol{v})$ is a conjunction of polynomial equations and inequations and $r_i \in \{1, \ldots, N\}$. For any real choice $\bar{\boldsymbol{v}}$ of parameters $\boldsymbol{v}$ there is one and only one $i \in \{1, \ldots, I\}$ such that $\mathbb{R} \models \Gamma_i(\bar{\boldsymbol{v}})$. For this i we have $\operatorname{rank}(A(\bar{\boldsymbol{v}})) = r_i$.

$I := 0$
create an empty stack
push $(\text{true}, A, 1)$
while stack is not empty **do**
  $(\Gamma, A, p) :=$ pop
  if $\Gamma \nvdash$ false **then**
    if there is $m \in \{p, \ldots, M\}$, $n \in \{p, \ldots, N\}$ s.t. $\Gamma \vdash A_{mn} \neq 0$ **then**
      in A, swap rows p with m and columns p with n
      in A, use row p to obtain $A_{p+1,p} = \cdots = A_{m,p} = 0$
      push $(\Gamma, A, p+1)$
    else if there is $m \in \{p, \ldots, M\}$, $n \in \{p, \ldots, N\}$ s.t. $\Gamma \nvdash A_{mn} = 0$ **then**
      $a := A_{mn}$
      push $(\Gamma \wedge a \neq 0, A, p)$
      in A, set $A_{mn} := 0$ *optional optimisation*
      push $(\Gamma \wedge a = 0, A, p)$
    else *A is in row echelon form modulo Γ*
      $I := I + 1$
      $(\Gamma_I, r_I) := (\Gamma, p)$
    end if
  end if
end while
return $\{(\Gamma_1, r_1), \ldots, (\Gamma_I, r_I)\}$

Algorithm 19: ParametricRank

or there is finite sequence $i = i_0, i_1, \ldots, i_k = j$ such that $f^{(1)}_{i_l}(\boldsymbol{k}, \boldsymbol{x})$ and $f^{(1)}_{i_{l+1}}(\boldsymbol{k}, \boldsymbol{x})$ share a monomial, for $0 \leq l \leq k$. Since by definition of the truncated system, all the monomials in $f^{(1)}_i(\boldsymbol{k}, \boldsymbol{x})$ have the same order, it follows that all the polynomials $f^{(1)}_i(\boldsymbol{k}, \boldsymbol{x})$ for some $i \in I$ have the same order $\nu = od_i + b_i + o\mu$. The time scale order μ_i of x_i is the order of $\frac{\dot{x}_i}{x_i} = \frac{f^{(1)}_i(\boldsymbol{k}, \boldsymbol{x})}{x_i}$, namely $\mu_i = \nu - od_i = b_i + o\mu$. Thus, the evenness of b_i implies that of d_i. The second part of the Theorem follows from the Lemma 6.14.

□

Proof of Theorem 6.13 . Let us note that $\dot{q} = q \sum_{i \in I} c_i \dot{x}_i$. Since q is conserved by the truncated system (5.27) we have

$$\sum_{i=1}^{n} c_i \delta^{b_i + od_i + o\mu} \bar{f}^{(1)}_i(\bar{\boldsymbol{k}}, \boldsymbol{y}) = 0,$$

for all $\bar{\boldsymbol{k}}$,$\boldsymbol{y}$, δ. Furthermore

$$\dot{q} = \sum_{i=1}^{n} c_i \delta^{b_i + od_i + o\mu} \left(\bar{f}_i^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{y}) + \delta^{b_i'} \bar{f}_i^{(2)}(\bar{\boldsymbol{k}}, \boldsymbol{y}, \delta) \right) = \sum_{i=1}^{n} c_i \delta^{b_i + b_i' + od_i + o\mu} \bar{f}_i^{(2)}(\bar{\boldsymbol{k}}, \boldsymbol{y}, \delta).$$

Let us denote by μ_q the order of the time scale of q, i.e. the order of $\frac{\dot{q}}{q}$. The order of q is $\min\{od_i \mid i \in I\}$. Thus,

$$\mu_q = \min\{b_i + b_i' + od_i + o\mu \mid i \in I\} - \min\{od_i \mid i \in I\}$$

We prove now that q is a slow variable, that is, we need to check that $b_i + o\mu < \mu_q$, for all $i \in I$. These conditions are equivalent to

$$b_i < \min\{b_j + b_j' + od_j \mid j \in I\} - \min\{od_j \mid j \in I\}, \ \forall i \in I.$$

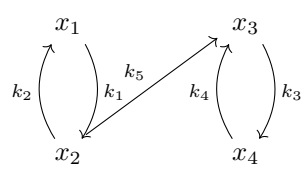
According to the Lemma 6.14, $b_i = \nu - o\mu - od_i$, for all $i \in I$. Obviously, it is enough to prove that

$$\nu - o\mu - \min\{od_j \mid j \in I\} < \min\{b_j + b_j' + od_j \mid j \in I\} - \min\{od_j \mid j \in I\},$$

which leads us to the inequation $\nu - o\mu < \min\{b_j + b_j' + od_j \mid j \in I\}$. Since $\nu = b_j + od_j + o\mu$ and $b_j' > 0$ for all $j \in I$, all the above inequalities are satisfied.

□

Example 6.15. Consider the following chemical reaction network



If the dynamics of this reaction network is of mass-action form, then it is given by the following set of ODEs:

$$\begin{aligned}
\dot{x}_1 &= k_2 x_2 - k_1 x_1, \\
\dot{x}_2 &= k_1 x_1 - (k_2 + k_5) x_2, \\
\dot{x}_3 &= k_5 x_2 + k_4 x_4 - k_3 x_3, \\
\dot{x}_4 &= k_3 x_3 - k_4 x_4.
\end{aligned}$$

We consider the case where $e_1 = e_2 = e_3 = e_4 = 0$, $e_5 = 1$. The total equilibrations are solutions of the system $d_1 = d_2, \min(d_4, d_2+1) = d_3, d_3 = d_4$ that read $d_1 = d_2 \geq d_4 - 1, d_3 = d_4$. Let us consider total equilibration $d_1 = d_2 = -1$, $d_3 = d_4 = -2$. The rescaled system corresponding to this situations is

$$\begin{aligned}
\dot{y}_1 &= \bar{k}_2 y_2 - \bar{k}_1 y_1, \\
\dot{y}_2 &= \bar{k}_1 y_1 - (\bar{k}_2 + \epsilon \bar{k}_5) y_2, \\
\dot{y}_3 &= \epsilon^2 \bar{k}_5 y_2 + \bar{k}_4 y_4 - \bar{k}_3 y_3, \\
\dot{y}_4 &= \bar{k}_3 y_3 - \bar{k}_4 y_4.
\end{aligned}$$

Since ϵ occurs with only integer powers we can choose $\delta = \epsilon$.

All species $x_i, 1 \leq i \leq 4$ have the same time scale orders $b_i = 0$.

The truncated system is

$$\begin{aligned}\dot{y}_1 &= \bar{k}_2 y_2 - \bar{k}_1 y_1,\\ \dot{y}_2 &= \bar{k}_1 y_1 - \bar{k}_2 y_2,\\ \dot{y}_3 &= \bar{k}_4 y_4 - \bar{k}_3 y_3,\\ \dot{y}_4 &= \bar{k}_3 y_3 - \bar{k}_4 y_4,\end{aligned}$$

and the truncated stoichiometric matrix reads

$$\boldsymbol{S}^{(1)} = \begin{pmatrix} -1 & 1 & 0 & 0\\ 1 & -1 & 0 & 0\\ 0 & 0 & -1 & 1\\ 0 & 0 & 1 & -1\end{pmatrix}.$$

The system has two irreducible conservation laws $y_1 + y_2$ and $y_3 + y_4$. These correspond to the species pools $q_1 = x_1 + x_2$ and $q_2 = x_3 + x_4$ that have time scale orders $\mu_{q_1} = 1$ and $\mu_{q_2} = 2$, respectively. Thus, q_1 and q_2 are slower than the species $x_i, 1 \leq i \leq 4$. To be noticed that the species concentrations in these pools have equal orders $d_1 = d_2$ and $d_3 = d_4$, consistent with the fact that in irreducible pools, species with the same time scales have the same concentration orders (Lemma 6.14).

Theorem 6.16. *Let $q = \prod_{i=1}^{n} x_i^{m_i}$ be an approximate, irreducible monomial conservation law conserved by the truncated system (5.27). Let $I = \{i \mid 1 \leq i \leq n, m_i \neq 0\}$. If q is irreducible, then the variable q is slower than all the variables $\{x_i \mid i \in I\}$ of the system (2.2). Furthermore, x_i, $i \in I$ have the same time scale orders.*

Proof. Let us note that $\dot{q} = q \sum_{i \in I} m_i \frac{\dot{x}_i}{x_i}$. Since q is conserved by the truncated system (5.27) we have

$$\sum_{i \in I} \delta^{b_i + o\mu} \frac{m_i}{y_i} \bar{f}_i^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{y}) = 0,$$

for all $\bar{\boldsymbol{k}}$,$\boldsymbol{y}$, δ. As $\frac{\bar{f}_i^{(1)}}{y_i}$ is a sum of rational monomials, this is possible only if for any $i \in I$, there is $j \in I$ such that $\frac{\bar{f}_i^{(1)}}{y_i}$ and $\frac{\bar{f}_j^{(1)}}{y_j}$ share a common monomial. Since q is irreducible, for all $i, j \in I$, either $\frac{\bar{f}_i^{(1)}}{y_i}$ and $\frac{\bar{f}_j^{(1)}}{y_j}$ share a common monomial, or there is a sequence $i = i_0, i_1, \ldots, i_k = j$ such that $\frac{\bar{f}_{i_l}^{(1)}}{y_{i_l}}$ and $\frac{\bar{f}_{i_{l+1}}^{(1)}}{y_{i_{l+1}}}$ share a common monomial, for $0 \leq l \leq k-1$. It follows that $\frac{f_i^{(1)}(\boldsymbol{k},\boldsymbol{x})}{x_i}$ have the same order, i.e. $b_i + o\mu$ is the same and for all $i \in I$.

Let μ_i be the time scale order of x_i. As shown in the previous section

$$\mu_i = b_i + o\mu.$$

It follows that x_i have the same time scale orders for $i \in I$.

In order to compute the time scale of q we use

$$\frac{\dot{q}}{q} = \sum_{i \in I} m_i \frac{\dot{x}_i}{x_i} = \sum_{i \in I} m_i \delta^{b_i + o\mu} \frac{\bar{f}_i^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{y}) + \delta^{b'_i} \bar{f}_i^{(2)}(\bar{\boldsymbol{k}}, \boldsymbol{y}, \delta)}{y_i} = \sum_{i \in I} m_i \delta^{b_i + b'_i + o\mu} \frac{f_i^{(2)}(\bar{\boldsymbol{k}}, \boldsymbol{y}, \delta)}{y_i},$$

where the last equality follows from the fact that q is conserved by (5.27).

Let us denote by μ_q the order of the time scale of q, i.e. the order of $\frac{\dot{q}}{q}$. Considering that m_i are small integers of order $O(\delta^0)$, it follows that

$$\mu_q = \min_{i\in I}\{b_i + b'_i + o\mu\}.$$

To prove that q is a slow variable with respect to the variables $\{x_i, i \in I\}$, we need to show that $\forall i \in I, \mu_i < \mu_q$. So we need to prove that $\forall i \in I, b_i < \min_{j\in I}\{b_j + b'_j\}$

As $\forall i, j \in I, b_i = b_j$ and $b'_j > 0$, we have $\forall i \in I, b_i = \min_{j\in I} b_j < \min_{j\in I}\{b_j + b'_j\}$.

□

Theorem 6.17. *Let $q = \sum_{j=1}^{r} c_j \boldsymbol{x}^{\boldsymbol{m}_j}$, where $c_j \in \mathbb{R}^*$ and $\boldsymbol{m}_j = (m_{1j}, \dots, m_{nj}) \in \mathbb{N}^n$, be an approximated conservation law, conserved by (5.27). Let $I = \{i \mid m_{ij} \neq 0$ for some j, $1 \leq j \leq r\}$. If q is irreducible, then q is slower than all the variables x_i, $i \in I$ of the system (2.2). If furthermore, the timescales of the of the variables x_i have the same order in δ, i.e. $b_i = b_{i'}$ for all $i, i' \in I$, then the monomials in q have also the same order in δ, i.e. $\langle \boldsymbol{d}, \boldsymbol{m}_j\rangle = \langle \boldsymbol{d}, \boldsymbol{m}_{j'}\rangle$ for all $j, j' \in \{1, \dots, r\}$ such that $m_{i,j} \neq 0$, $m_{i',j'} \neq 0$ for some $i, i' \in I$.*

Proof. We note that $\dot{q} = \sum_{j=1}^{r} c_j x^{\boldsymbol{m}_j}(\sum_{i=1}^{r} m_{ij}\frac{\dot{x}_i}{x_i}) = \sum_{i=1}^{n}\frac{\dot{x}_i}{x_i}(\sum_{m_{i,j}\neq 0} m_{ij}c_j\boldsymbol{x}^{\boldsymbol{m}_j})$.

Let us define the sum of rational monomials

$$E_{i,j}(\bar{\boldsymbol{k}}, \boldsymbol{y}, \delta) = \delta^{b_i+o\mu}\frac{\bar{f}_i^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{y}, \delta)}{y_i} m_{ij}c_j\delta^{o\langle \boldsymbol{d}, \boldsymbol{m}_j\rangle}\boldsymbol{y}^{\boldsymbol{m}_j}.$$

As q is conserved by (5.27) it follows

$$\sum_{1\leq i\leq n,\, 1\leq j\leq r,\, m_{i,j}\neq 0} E_{i,j}(\bar{\boldsymbol{k}}, \boldsymbol{y}, \delta) = 0,$$

for all $\bar{\boldsymbol{k}}$, $\boldsymbol{y}$, δ.

This is only possible if for any i, j such that $m_{i,j} \neq 0$, there is i', j' such that $m_{i',j'} \neq 0$, and such that $E_{i,j}$ and $E_{i',j'}$ share a common monomial. Because q is irreducible for all i, j, i', j' such that $m_{i,j} \neq 0$ and $m_{i',j'} \neq 0$, either $E_{i,j}$ and $E_{i',j'}$ share a common monomial, or there is a sequence $(i, j) = (i_0, j_0), (i_1, j_1), \dots, (i_k, j_k) = (i', j')$, such that E_{i_l,j_l} and $E_{i_{l+1},j_{l+1}}$ share a common monomial, for $0 \leq l \leq k-1$. Thus, the expressions $\frac{f_i^{(1)}(\boldsymbol{k},\boldsymbol{x})}{x_i} m_{ij}c_j\boldsymbol{x}^{\boldsymbol{m}_j}$ have the same order for all i, j such that $m_{i,j} \neq 0$, i.e.

$$b_i + o\mu + o\langle \boldsymbol{d}, \boldsymbol{m}_j\rangle = b_{i'} + o\mu + o\langle \boldsymbol{d}, \boldsymbol{m}_{j'}\rangle,$$

for all i, j, i', j' such that $m_{i,j} \neq 0$ and $m_{i',j'} \neq 0$.

In particular, if all the variables x_i have the same timescale order b_i for all $i \in I$, it follows that $\langle \boldsymbol{d}, \boldsymbol{m}_j\rangle$ are the same for all $1 \leq j < r$, such that $m_{i,j} \neq 0$ for some $i \in I$, which proves the second part of the theorem.

As q is conserved by (5.27) it follows

$$\dot{q} = \sum_{i,j,m_{i,j}\neq 0} \delta^{b_i+b'_i+o\mu}\frac{\bar{f}_i^{(2)}(\bar{\boldsymbol{k}}, \boldsymbol{y}, \delta)}{y_i} m_{ij}c_j\delta^{o\langle \boldsymbol{d}, \boldsymbol{m}_j\rangle}\boldsymbol{y}^{\boldsymbol{m}_j}.$$

The time scale order of q is

$$\mu_q = \min_{m_{ij}\neq 0}(b_i + b_i' + o\mu + o\langle \boldsymbol{d}, \boldsymbol{m}_j\rangle) - \min_{1\leq j\leq r} o\langle \boldsymbol{d}, \boldsymbol{m}_j\rangle.$$

Let $i \in I$ and j such that $m_{i,j} \neq 0$. Using $b_i + o\mu + o\langle \boldsymbol{d}, \boldsymbol{m}_j\rangle = \min_{m_{i,j}\neq 0}(b_i + o\mu + o < \boldsymbol{d}, \boldsymbol{m}_j\rangle)$ and $b_i' > 0$ it follows

$$b_i + o\mu = \min_{m_{i,j}\neq 0}(b_i + o\mu + o\langle \boldsymbol{d}, \boldsymbol{m}_j\rangle) - o\langle \boldsymbol{d}, \boldsymbol{m}_j\rangle < \mu_q,$$

meaning that q is slower than all variables x_i, $i \in I$.

□

6.2.1 Reduction of the Michaelis-Menten Model under Quasi-equilibrium Conditions.

The Michaelis-Menten model has been used as a paradigmatic example as it allows to introduce the main concepts of model reduction. Both QSS and QE reductions were discussed in [Noel et al., 2014,Radulescu et al., 2015b,Samal et al., 2015], for a two variable Michaelis-Menten model obtained from the three variable one by exact reduction, using one exact linear conservation laws. In this subsection we illustrate a different approach that starts with the three variable model introduced in Example6.1.

The scaling in (6.1) is based on the total tropical equilibration solution $d_1 = d_2 = d_3 = 0$ and $\delta = \epsilon$. According to this scaling all the three variables x_1,x_2 and x_3 have the same time scale. The new variables $x_4 = x_1 + x_2$ and $x_5 = x_2 + x_3$ are conservation laws (approximate and exact, respectively).

We use the approximate and exact conservation laws to eliminate two out of the three variables x_1, x_2, x_3 and obtain

$$\begin{aligned} x_2 &= x_4 - x_1, \\ x_3 &= x_5 - x_4 + x_1. \end{aligned} \tag{6.3}$$

The remaining variables satisfy

$$\begin{aligned} \dot{x_1} &= -k_1x_1(x_5 - x_4 + x_1) + k_2(x_4 - x_1) \\ \dot{x_4} &= -\delta k_3(x_4 - x_1) \\ \dot{x_5} &= 0 \end{aligned} \tag{6.4}$$

The system (6.4) shows that x_4 is a slow variable and x_5 is a conserved, constant variable.

The constant variable can be turned into a parameter $x_5 = k_4$ that leads to

$$\dot{x_1} = -k_1x_1(k_4 - x_4 + x_1) + k_2(x_4 - x_1) \tag{6.5}$$

$$\dot{x_4} = \delta k_3(x_4 - x_1) \tag{6.6}$$

The system (6.5),(6.6) is typically a slow/fast system with x_1 fast and x_4 slow. The fast dynamics is described by

$$\dot{x_1} = -k_1x_1(k_4 - x_4 + x_1) + k_2(x_4 - x_1), \tag{6.7}$$

and has two hyperbolic steady states, among which only one is positive and stable[1]:

$$x_1^* = \frac{-(k_1(k_4 - x_4) + k_2) + \sqrt{(k_1(k_4 - x_4) + k_2)^2 + 4k_1k_2x_4}}{2k_1}. \tag{6.8}$$

By singular perturbation results [Tikhonov, 1952b, Hoppensteadt, 1969b, Fenichel, 1979] the solutions of (6.5),(6.6) with appropriate initial conditions converge, when $\delta \to 0$, to the solutions of the differential-algebraic system

$$\begin{aligned} 0 &= -k_1x_1(k_4 - x_4 + x_1) + k_2(x_4 - x_1) \\ x_4' &= k_3(x_4 - x_1), \end{aligned} \tag{6.9}$$

where the derivative is with respect to the time $\tau = t\delta$.

Using the solution (6.8), the semi-explicit differential-algebraic system (6.9) can be transformed into the reduced ODE

$$x_4' = k_3\left(x_4 - \frac{-(k_1(k_4 - x_4) + k_2) + \sqrt{(k_1(k_4 - x_4) + k_2)^2 + 4k_1k_2x_4}}{2k_1}\right). \tag{6.10}$$

The Figure 6.1 illustrates the accuracy of this reduction.

[1] all the eigenvalues of the Jacobian matrix computed in this state lie in the complex half-plane

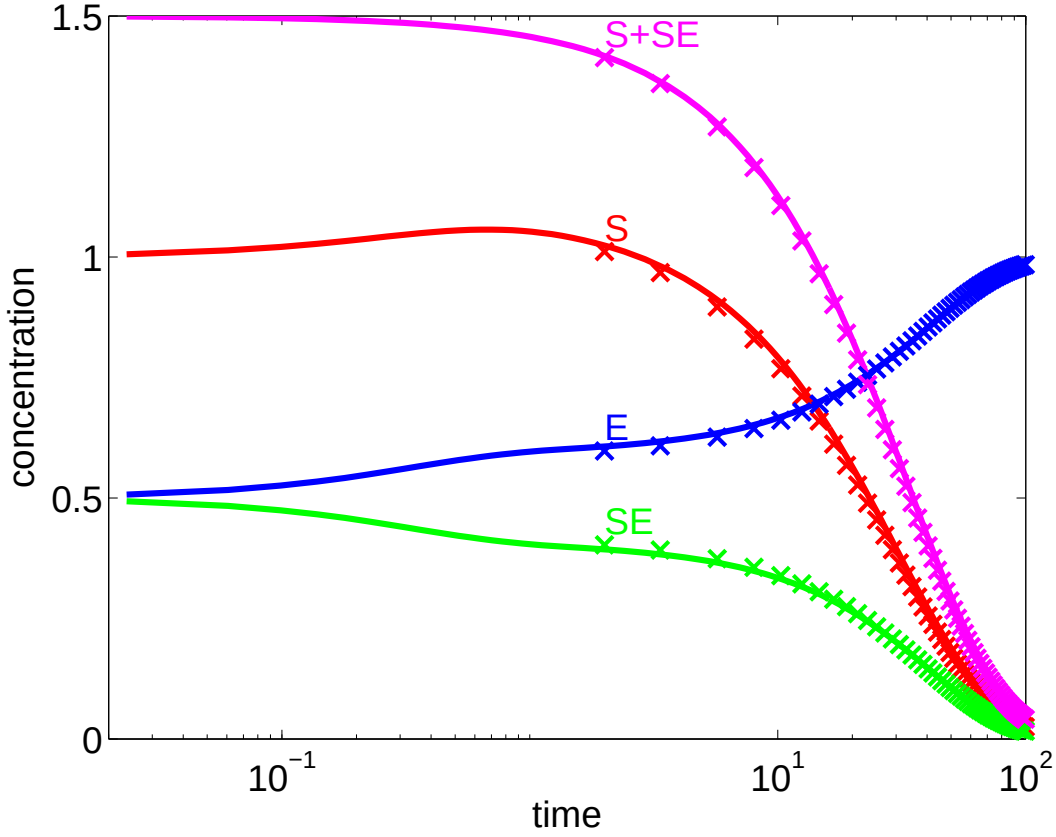


Figure 6.1: Comparison of numerical solutions obtained with the full Michaelis-Menten model (continuous lines) and with the reduced model (crosses). The variables S, E and SE are fast and slaved (in the reduced model, their values are obtained as solutions of algebraic equations) and the approximated conservation law $S + SE$ is slow. The initial values are such that E and S have comparable concentrations; this is not compatible with the quasi-steady state approximation but is compatible with the quasi-equilibrium approximation that has been used here.

Chapter 7

The Model Reduction Algorithms

In this section, to wrap up all the above developed concepts, we propose several model reduction algorithms. These take into account approximate conservation laws and are applicable to CRN models with multiple timescales and polynomial rate functions.

We consider two types of reductions:

i) Reduction at the slowest timescale, and

ii) Nested reductions at intermediate timescales.

In the case i), all the variables except the slowest one are eliminated during the reduction procedure. The reduced model is an ODE for the slowest variable. The elimination of fast variables proceeds hierarchically, the fastest variables being eliminated first.

In the case ii), all the fast variables up to the (l-1)-th fastest one satisfy polynomial quasi-steady state equations and can be eliminated. The remaining variables satisfy a reduced system of ODEs. The reduced dynamics takes place on the normally hyperbolic invariate manifold that is close to the critical manifold defined by the quasi-steady state equations. Changing l from $l = 1$ to $l = m$ one obtains m nested, attractive, normally hyperbolic invariant manifolds along which the reduced dynamics evolves at successively slower timescales. Of course, the case i) results from ii) with $l = m$.

For both i) and ii) types of reduction, the elimination of the fast variables is possible only if the truncated system at the k-th timescale has hyperbolically stable steady states (with $1 \leq k \leq m$ in the case i) and $1 \leq k \leq l$ in the case ii)). When there are approximate conservation laws conserved by the truncated system, the hyperbolicity condition is not fulfilled and the standard reduction algorithm proposed in [Kruff et al., 2020] does not apply. Our solution to this problem is to add approximate conservation laws to the set of variables, eliminate some of the fast variables and obtain a modified system that has no approximate conservation laws and satisfies the hyperbolicity condition.

7.1 The Slowest Timescale Reduction

Like in [Kruff et al., 2020] we introduce the small parameters $\delta_{l-1} = \delta^{b_l - b_{l-1}}$, for $2 \le l \le m$ and the vector $\bar{\delta} = (\delta_1, \ldots, \delta_m)$.

Let us change the time variable to $\tau' = \tau\delta_1\delta_2\ldots\delta_m$, the slowest timescale of the model. The system (5.28) becomes

$$\begin{aligned}
\delta_1\delta_2\ldots\delta_{m-1}\boldsymbol{z}_1' &= \bar{\boldsymbol{f}}_1^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}) + \bar{\boldsymbol{g}}_1(\bar{\boldsymbol{k}}, \boldsymbol{z}, \bar{\delta}), \\
&\vdots \\
\delta_{m-1}\boldsymbol{z}_{m-1}' &= \bar{\boldsymbol{f}}_{m-1}^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}) + \bar{\boldsymbol{g}}_{m-1}(\bar{\boldsymbol{k}}, \boldsymbol{z}, \delta), \\
\boldsymbol{z}_m' &= \bar{\boldsymbol{f}}_m^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}) + \bar{\boldsymbol{g}}_m(\bar{\boldsymbol{k}}, \boldsymbol{z}, \delta)
\end{aligned} \tag{7.1}$$

where $\bar{\boldsymbol{g}}_k(\bar{\boldsymbol{k}}, \boldsymbol{z}, \bar{\delta}) = \delta^{b_k'}\bar{\boldsymbol{f}}_k^{(2)}(\bar{\boldsymbol{k}}, \boldsymbol{z}, \delta)$ satisfy $\bar{\boldsymbol{g}}_k(\bar{\boldsymbol{k}}, \boldsymbol{z}, 0) = 0$ for $1 \le k \le m$.

We also assume that the functions $\bar{\boldsymbol{g}}_k$ are smooth in all their variables. The smoothness in $\bar{\delta}$ can be tested algorithmically with methods introduced in [Kruff et al., 2020].

By setting $\bar{\delta} = 0$ in (7.1) we obtain the *slowest timescale reduced system*

$$\begin{aligned}
0 &= \bar{\boldsymbol{f}}_1^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}), \\
&\vdots \\
0 &= \bar{\boldsymbol{f}}_{m-1}^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}), \\
\boldsymbol{z}_m' &= \bar{\boldsymbol{f}}_m^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z})
\end{aligned} \tag{7.2}$$

For $\boldsymbol{k} \in \mathbb{R}_+^r$, $\boldsymbol{z}_m \in \mathbb{R}_+^{n_m}$, a state $(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{m-1})$ satisfying the system of equations

$$\bar{\boldsymbol{f}}_1^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_{m-1}, \boldsymbol{z}_m) = 0, \ldots, \bar{\boldsymbol{f}}_{m-1}^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_{m-1}, \boldsymbol{z}_m) = 0 \tag{7.3}$$

is called *quasi-steady state*.

Consider that the system (7.3) can be solved for $(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{m-1})$ in the following, hierarchical way. First, there is a differentiable function $\tilde{\boldsymbol{f}}_1(\bar{\boldsymbol{k}}, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_m)$ such that

$$\bar{\boldsymbol{f}}_1^{(1)}(\bar{\boldsymbol{k}}, \tilde{\boldsymbol{f}}_1(\bar{\boldsymbol{k}}, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_m), \boldsymbol{z}_2, \ldots, \boldsymbol{z}_m) = 0.$$

Next, consider that there is a differentiable function $\tilde{\boldsymbol{f}}_2(\bar{\boldsymbol{k}}, \boldsymbol{z}_3, \ldots, \boldsymbol{z}_m)$ such that

$$\bar{\boldsymbol{f}}_2^{(1)}(\bar{\boldsymbol{k}}, \tilde{\boldsymbol{f}}_1(\bar{\boldsymbol{k}}, \tilde{\boldsymbol{f}}_2(\bar{\boldsymbol{k}}, \boldsymbol{z}_3, \ldots, \boldsymbol{z}_m), \ldots, \boldsymbol{z}_m), \tilde{\boldsymbol{f}}_2(\bar{\boldsymbol{k}}, \boldsymbol{z}_3, \ldots, \boldsymbol{z}_m), \ldots, \boldsymbol{z}_m) = 0.$$

Assuming that the procedure can go on, consider finally that there is a function $\tilde{\boldsymbol{f}}_{m-1}(\bar{\boldsymbol{k}}, \boldsymbol{z}_m)$ such that

$$\bar{\boldsymbol{f}}_{m-1}^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_{m-1}, \boldsymbol{z}_m) = 0,$$

where $\boldsymbol{z}_1$, $\boldsymbol{z}_2$, …, $\boldsymbol{z}_{m-1}$ are recursively replaced by $\tilde{\boldsymbol{f}}_1(\bar{\boldsymbol{k}}, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_m)$, $\tilde{\boldsymbol{f}}_2(\bar{\boldsymbol{k}}, \boldsymbol{z}_3, \ldots, \boldsymbol{z}_m)$, …, $\tilde{\boldsymbol{f}}_{m-1}(\bar{\boldsymbol{k}}, \boldsymbol{z}_m)$, respectively.

Consider the reduced system

$$\boldsymbol{z}_m' = \boldsymbol{f}_m^*(\bar{\boldsymbol{k}}, \boldsymbol{z}_m) \tag{7.4}$$

where $\boldsymbol{f}_m^*(\bar{\boldsymbol{k}}, \boldsymbol{z}_m)$ is obtained from $\bar{\boldsymbol{f}}_m^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_m)$ by substituting $\boldsymbol{z}_1$, $\boldsymbol{z}_2$, ..., $\boldsymbol{z}_{m-1}$ as above.

Solutions of the system (7.1) in the limit $\bar{\delta} \to 0$ were studied by Tikhonov [Tikhonov, 1952b], Hoppensteadt [Hoppensteadt, 1969b] and O'Malley [O'Malley, 1971]. They showed that under appropriate conditions (roughly non-degeneracy and hyperbolicity, see Section 7.3), the solutions of the system (7.1) with initial conditions $\boldsymbol{z}_i(0) = g_i(\delta)$, where $g_i(\delta)$ are differentiable functions, converge to the solutions of the system (7.4) with initial conditions $\boldsymbol{z}_m(0) = g_m(0)$ when $\bar{\delta} \to 0$. By this reduction, called *quasi-steady state approximation*, all the variables faster than the slowest one are eliminated and the reduced model describes the dynamics at the slowest timescale. This type of reduction is the most popular in applications, for instance in physical chemistry (where it is known as the Semenov-Bodenstein quasi-steady state approximation) and in computational systems biology [Radulescu et al., 2012].

7.2 Nested intermediate timescale reductions

This type of reduction was proposed by Cardin and Teixera [Cardin and Teixeira, 2017], and algorithmically formalized in Kruff et al. [Kruff et al., 2020]. We call it nested because the reduced dynamics at intermediate timescales are embedded in normally hyperbolic invariant manifolds that form a nested family (manifolds of slower variables are included in manifolds of faster variables). In [Cardin and Teixeira, 2017] both stable and unstable manifolds were considered. Here, like in [Kruff et al., 2020], we consider only the stable case. The stable, normally hyperbolic invariant manifolds attract and confine the dynamics at various timescales and are used to obtain reductions valid at intermediate timescales.

We provide here the formal description of the reduction at an intermediate timescale of order $\delta^{b_l} = \delta^{b_1}\delta_1\delta_2\ldots\delta_{l-1}$.

By redefining the time to $\tau' = \tau\delta^{b_l}$, where $1 \le l \le m$ we get

$$\begin{aligned}
\delta_1\delta_2\ldots\delta_{l-1}\boldsymbol{z}_1' &= (\bar{\boldsymbol{f}}_1^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}) + \bar{\boldsymbol{g}}_1(\bar{\boldsymbol{k}}, \boldsymbol{z}, \bar{\delta})),\\
&\vdots\\
\delta_{l-1}\boldsymbol{z}_{l-1}' &= (\bar{\boldsymbol{f}}_{l-1}^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}) + \bar{\boldsymbol{g}}_{l-1}(\bar{\boldsymbol{k}}, \boldsymbol{z}, \bar{\delta})),\\
\boldsymbol{z}_l' &= (\bar{\boldsymbol{f}}_l^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}) + \bar{\boldsymbol{g}}_l(\bar{\boldsymbol{k}}, \boldsymbol{z}, \bar{\delta})),\\
\boldsymbol{z}_{l+1}' &= \delta_l(\bar{\boldsymbol{f}}_{l+1}^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}) + \bar{\boldsymbol{g}}_{l+1}(\bar{\boldsymbol{k}}, \boldsymbol{z}, \bar{\delta})),\\
&\vdots\\
\boldsymbol{z}_m' &= \delta_l\delta_{l+1}\ldots\delta_{m-1}(\bar{\boldsymbol{f}}_m^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}) + \bar{\boldsymbol{g}}_m(\bar{\boldsymbol{k}}, \boldsymbol{z}, \bar{\delta})),
\end{aligned} \tag{7.5}$$

where $\bar{\boldsymbol{g}}_k(\bar{\boldsymbol{k}}, \boldsymbol{z}, \bar{\delta})$ are smooth functions such that $\bar{\boldsymbol{g}}_k(\bar{\boldsymbol{k}}, \boldsymbol{z}, 0) = 0$ for $1 \le k \le m$.

We call *reduced system at the lth fastest time or slower* the system obtained by setting

$\delta_1 = \delta_2 = \ldots = \delta_{l-1} = 0$ in (7.5)

$$\begin{aligned}
0 &= \bar{\boldsymbol{f}}_1^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}), \\
&\vdots \\
0 &= \bar{\boldsymbol{f}}_{l-1}^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}), \\
\boldsymbol{z}_l' &= \bar{\boldsymbol{f}}_l^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}), \\
\boldsymbol{z}_{l+1}' &= \delta_l(\bar{\boldsymbol{f}}_{l+1}^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}) + \bar{\boldsymbol{g}}_{l+1}(\bar{\boldsymbol{k}}, \boldsymbol{z}, \bar{\delta})), \\
&\vdots \\
\boldsymbol{z}_m' &= \delta_l \delta_{l+1} \ldots \delta_{m-1}(\bar{\boldsymbol{f}}_m^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}) + \bar{\boldsymbol{g}}_m(\bar{\boldsymbol{k}}, \boldsymbol{z}, \bar{\delta})),
\end{aligned} \tag{7.6}$$

with $\bar{\delta} = (0, \ldots, 0, \delta_l, \ldots, \delta_{m-1})$.

In [Kruff et al., 2020] we have also defined the simpler reduced system

$$\begin{aligned}
0 &= \bar{\boldsymbol{f}}_1^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}), \\
&\vdots \\
0 &= \bar{\boldsymbol{f}}_{l-1}^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}), \\
\boldsymbol{z}_l' &= \bar{\boldsymbol{f}}_l^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}), \\
\boldsymbol{z}_{l+1}' &= 0, \\
&\vdots \\
\boldsymbol{z}_m' &= 0,
\end{aligned} \tag{7.7}$$

that emphasizes three groups of variables: slaved (faster than δ^{b_l}), driving (timescale δ^{b_l}), and quenched (slower than δ^{b_l}).

If regularity and hyperbolicity conditions are satisfied (see [Cardin and Teixeira, 2017] and Section 7.3) then the solutions of the system (7.5) converge to the solutions of the system (7.6) when $\epsilon_k \to 0$ for $1 \leq k \leq l-1$ (Corrolary of Theorem A in [Cardin and Teixeira, 2017]).

The limit $(\delta_l, \ldots, \delta_{m-1}) \to 0$ leading from (7.6) to (7.7) can be treated in the simpler framework of regular perturbations. Using the same regularity and hyperbolicity conditions one can show that there is a time $T > 0$ such that the solutions of (7.5) converge towards solutions of (7.7) when $\bar{\delta} \to 0$, uniformly on any close subinterval $(0, T]$ (See Theorem 1 of [Kruff et al., 2020]). This result implies that the reduction (7.7) is valid on a time interval $[t_1\delta^{-b_l}, t_2\delta^{-b_l}]$, $[t_1, t_2] \subset (0, T]$, whereas the reduction (7.6) has a broader validity including times longer than δ^{-b_l}.

7.3 Hyperbolically attractive chains

The two types of reductions are based on hierarchical elimination of fast variables, previously discussed in [Kruff et al., 2020]. We revisit here this construction, using the concept of Schur complement.

Let us define $\boldsymbol{Z}_k = \begin{pmatrix} \boldsymbol{z}_1 \\ \vdots \\ \boldsymbol{z}_k \end{pmatrix}$ and $\bar{\boldsymbol{F}}_k^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}) = \begin{pmatrix} \bar{\boldsymbol{f}}_1^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}) \\ \vdots \\ \bar{\boldsymbol{f}}_k^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}) \end{pmatrix}$.

For any $k, 1 \leq k \leq m$, the set of positive solutions of $\bar{\boldsymbol{F}}_k^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}) = 0$ is denoted $\mathcal{M}_k$ and represents the intersection of the k-th quasi-steady state variety and the first orthant. For any l, $1 \leq l \leq m$, we call l-chain the chain of nested varieties intersected with the first orthant $\mathcal{M}_0 = \mathbb{R}_+^n \supseteq \mathcal{M}_1 \supseteq \ldots \supseteq \mathcal{M}_l$.

Let us solve $\bar{\boldsymbol{F}}_k^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}) = 0$ by successive elimination of variables, starting with $\boldsymbol{z}_1$ and ending with $\boldsymbol{z}_k$. During the elimination, the intermediary functions $\boldsymbol{f}_k^*(\bar{\boldsymbol{k}}, \boldsymbol{z}_k, \ldots, \boldsymbol{z}_m)$, $\tilde{\boldsymbol{f}}_k(\bar{\boldsymbol{k}}, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_m)$, $\tilde{\boldsymbol{F}}_k(\bar{\boldsymbol{k}}, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_m)$ are defined in the following, recursive way:

$$\boldsymbol{z}_k = \tilde{\boldsymbol{f}}_k(\bar{\boldsymbol{k}}, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_m) \text{ is a solution of } \boldsymbol{f}_k^*(\bar{\boldsymbol{k}}, \boldsymbol{z}_k, \ldots, \boldsymbol{z}_m) = 0, \tag{7.8}$$

$$\tilde{\boldsymbol{F}}_k(\bar{\boldsymbol{k}}, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_m) = \begin{pmatrix} \tilde{\boldsymbol{F}}_{k-1}(\bar{\boldsymbol{k}}, \tilde{\boldsymbol{f}}_k(\bar{\boldsymbol{k}}, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_m), \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_m) \\ \tilde{\boldsymbol{f}}_k(\bar{\boldsymbol{k}}, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_m) \end{pmatrix}, \tag{7.9}$$

$$\boldsymbol{f}_{k+1}^*(\bar{\boldsymbol{k}}, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_m) = \bar{\boldsymbol{f}}_{k+1}^{(1)}(\bar{\boldsymbol{k}}, \tilde{\boldsymbol{F}}_k(\bar{\boldsymbol{k}}, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_m), \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_m), \text{ if } k < m, \tag{7.10}$$

for $1 \leq k \leq m$ and with $\boldsymbol{f}_1^*(\bar{\boldsymbol{k}}, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_m) = \bar{\boldsymbol{f}}_1^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_m)$.

From the construction, it is straightforward that

Proposition 7.1. *$\boldsymbol{z} = (\boldsymbol{Z}_k, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_m)$, where $\boldsymbol{Z}_k = \tilde{\boldsymbol{F}}_k(\bar{\boldsymbol{k}}, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_m)$, is a solution of $\bar{\boldsymbol{F}}_k^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}) = 0$.*

From (7.8), the existence of the implicit functions $\tilde{\boldsymbol{f}}_k$, $\tilde{\boldsymbol{F}}_k$ needs the following non-degeneracy condition

Condition 7.2. *The solution of the equation $\boldsymbol{f}_k^*(\bar{\boldsymbol{k}}, \boldsymbol{z}_k, \ldots, \boldsymbol{z}_m) = 0$ is unique (non-degenerate), which is equivalent to $|D_{\boldsymbol{z}_k} \boldsymbol{f}_k^*| \neq 0$.*

This condition can be more conveniently written as follows

Theorem 7.3. *The implicit functions $\tilde{\boldsymbol{f}}_k(\bar{\boldsymbol{k}}, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_m)$, $\tilde{\boldsymbol{F}}_k(\bar{\boldsymbol{k}}, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_m)$ exist and are differentiable for $k \in \{1, \ldots, l\}$, if and only if*

$$|D_{\boldsymbol{Z}_k} \bar{\boldsymbol{F}}_k^{(1)}| \neq 0 \text{ for } \boldsymbol{z} \in \mathcal{M}_k \text{ and } k \in \{1, \ldots, l\}. \tag{7.11}$$

In order to prove this theorem we need the notion of Schur complement, that occurs naturally during the Gaussian elimination of variables (see, for instance, [Zhang, 2006]).

Definition 7.4. Let $M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$ be a block matrix such that A is invertible. The matrix $M/A = D - CA^{-1}B$ is called Schur complement of the block A of M. It satisfied several elementary properties [Zhang, 2006]:

$$|M| = |M/A||A|, \text{ Schur formula,} \tag{7.12}$$

$$rk(M) = rk(M/A) + rk(A), \text{ Guttman rank additivity formula.} \tag{7.13}$$

Returning to our problem, we have the following

Lemma 7.5. *The matrix $D_{\boldsymbol{z}_{k+1}}\boldsymbol{f}^*_{k+1}$ is a Schur complement. More precisely,*

$$D_{\boldsymbol{z}_{k+1}}\boldsymbol{f}^*_{k+1} = D_{\boldsymbol{Z}_{k+1}}\bar{\boldsymbol{F}}^{(1)}_{k+1}/D_{\boldsymbol{Z}_k}\bar{\boldsymbol{F}}^{(1)}_k, \tag{7.14}$$

for all $k \in \{1, \dots, m-1\}$ and $\boldsymbol{z} \in \mathcal{M}_k$ such that $D_{\boldsymbol{Z}_k}\bar{\boldsymbol{F}}^{(1)}_k$ is invertible.

Proof. Differentiating (7.10) with respect to $\boldsymbol{z}_{k+1}$ it follows

$$D_{\boldsymbol{z}_{k+1}}\boldsymbol{f}^*_{k+1} = (D_{\boldsymbol{Z}_k}\bar{\boldsymbol{f}}^{(1)}_{k+1})D_{\boldsymbol{z}_{k+1}}\tilde{\boldsymbol{F}}_k + D_{\boldsymbol{z}_{k+1}}\bar{\boldsymbol{f}}^{(1)}_{k+1}.$$

From the Proposition 7.1 it follows that

$$\bar{\boldsymbol{F}}^{(1)}_k(\bar{\boldsymbol{k}}, \tilde{\boldsymbol{F}}_k(\bar{\boldsymbol{k}}, \boldsymbol{z}_{k+1}, \dots, \boldsymbol{z}_m), \boldsymbol{z}_{k+1}, \dots, \boldsymbol{z}_m) = 0.$$

Differentiating with respect to $\boldsymbol{z}_{k+1}$ it follows

$$(D_{\boldsymbol{Z}_k}\bar{\boldsymbol{F}}^{(1)}_k)D_{\boldsymbol{z}_{k+1}}\tilde{\boldsymbol{F}}_k + D_{\boldsymbol{z}_{k+1}}\bar{\boldsymbol{F}}^{(1)}_k = 0,$$

for $\boldsymbol{z} \in \mathcal{M}_k$. Finally,

$$D_{\boldsymbol{z}_{k+1}}\boldsymbol{f}^*_{k+1} = -(D_{\boldsymbol{Z}_k}\bar{\boldsymbol{f}}^{(1)}_{k+1})(D_{\boldsymbol{Z}_k}\bar{\boldsymbol{F}}^{(1)}_k)^{-1}D_{\boldsymbol{z}_{k+1}}\bar{\boldsymbol{F}}^{(1)}_k + D_{\boldsymbol{z}_{k+1}}\bar{\boldsymbol{f}}^{(1)}_{k+1} = D_{\boldsymbol{Z}_{k+1}}\bar{\boldsymbol{F}}^{(1)}_{k+1}/D_{\boldsymbol{Z}_k}\bar{\boldsymbol{F}}^{(1)}_k.$$

□

Now we can prove the Theorem 7.3.

Proof. Using the Schur formula (7.12),

$$|D_{\boldsymbol{z}_k}\boldsymbol{f}^*_k||D_{\boldsymbol{Z}_{k-1}}\bar{\boldsymbol{F}}^{(1)}_{k-1}| = |D_{\boldsymbol{Z}_k}\bar{\boldsymbol{F}}^{(1)}_k|, \text{ for } k \in \{2, \dots, l\},$$

$$|D_{\boldsymbol{z}_1}\boldsymbol{f}^*_1| = |D_{\boldsymbol{Z}_1}\bar{\boldsymbol{F}}^{(1)}_1|.$$

This implies that $|D_{\boldsymbol{z}_k}\boldsymbol{f}^*_k| \neq 0$ for $k \in \{1, \dots, l\}$ is equivalent to $|D_{\boldsymbol{Z}_k}\bar{\boldsymbol{F}}^{(1)}_k| \neq 0$ for $k \in \{1, \dots, l\}$. □

As discussed in [Kruff et al., 2020] and [O'Malley, 1971], the validity of the quasi-steady state approximation depends also on the following, hyperbolicity condition:

Condition 7.6 (Hyperbolicity)**.** *$\boldsymbol{z}_l = \tilde{\boldsymbol{f}}_l(\bar{\boldsymbol{k}}, \boldsymbol{z}_{l+1}, \dots, \boldsymbol{z}_m)$ is a hyperbolically stable steady state of the ODE*

$$\boldsymbol{z}'_l = \boldsymbol{f}^*_l(\bar{\boldsymbol{k}}, \boldsymbol{z}_l, \boldsymbol{z}_{l+1}, \dots, \boldsymbol{z}_m),$$

for all $1 \leq l \leq m-1$, where $\boldsymbol{f}^*_l$ are defined like in subsection 7.3.

By hyperbolically stable it is meant that all the eigenvalues of the Jacobian matrix computed at the steady state have strictly negative real values.

As discussed in [Kruff et al., 2020], an important concept for the geometric theory of singular perturbations is the hyperbolically attractive chain.

Definition 7.7. The chain of nested quasi-steady state varieties intersected with the first orthant form a hyperbolically attractive l-chain if all eigenvalues of $D_{\boldsymbol{z}_k}\boldsymbol{f}_k^*$ have negative real parts for all $k \in \{1, \ldots, l\}$. In this case we write $\mathcal{M}_0 \rhd \mathcal{M}_1 \rhd \ldots \rhd \mathcal{M}_l$.

From the Lemma 7.5 it follows

Proposition 7.8. *A l-chain is hyperbolically attractive if and only if all eigenvalues of $D_{\boldsymbol{z}_1}\bar{\boldsymbol{f}}_1^{(1)}$ have negative real parts for $\boldsymbol{z} \in \mathcal{M}_1$ and all eigenvalues of $D_{\boldsymbol{Z}_k}\bar{\boldsymbol{F}}_k^{(1)}/D_{\boldsymbol{Z}_{k-1}}\bar{\boldsymbol{F}}_{k-1}^{(1)}$ have negative real parts for $\boldsymbol{z} \in \mathcal{M}_{k-1}$ and $k \in \{2 \ldots, l\}$.*

Remark 7.9. The nested reduction (7.6) is valid up to the l-th timescale if the l-th chain $\mathcal{M}_0 \rhd \mathcal{M}_1 \rhd \ldots \rhd \mathcal{M}_l$ is hyperbolically attractive (see [Cardin and Teixeira, 2017,Kruff et al., 2020]). The slowest timescale reduction discussed in the section is valid if $\mathcal{M}_0 \rhd \mathcal{M}_1 \rhd \ldots \rhd \mathcal{M}_m$ is hyperbolically attractive.

The existence of approximate conservation laws corresponds to the so-called *quasi-equilibrium* condition [Gorban et al., 2010,Radulescu et al., 2012]. Let $\phi_l(\boldsymbol{x})$ be a linear, monomial, or polynomial irreducible approximate conservation law depending only on variables having time-scales orders equal to or faster than b_l. As shown in section (see Lemma 34, Theorems 33,39), in this case all the monomial terms at the r.h.s of ODEs obeyed by the variables involved in $\phi_l(\boldsymbol{x})$ have the same order in δ, therefore

$$f_i^{(1)}(\boldsymbol{k}, \boldsymbol{x}) = \delta^{\mu_l} \bar{f}_i^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}), \text{ for all } i \text{ such that } \frac{\partial \phi_l(\boldsymbol{x})}{\partial x_i} \neq 0. \tag{7.15}$$

From the definition of approximate conservation laws we have $D_{\boldsymbol{x}}\phi_l(\boldsymbol{x})\boldsymbol{F}_l^{(1)}(\boldsymbol{k}, \boldsymbol{x}) = 0$. Because ϕ_l depends only on $\boldsymbol{X}_l$, where $\boldsymbol{X}_l$ contains all the variables x_i faster or of the timescale b_l, we got

$$D_{\boldsymbol{X}_l}\phi_l(\boldsymbol{x})\boldsymbol{F}_l^{(1)}(\boldsymbol{k}, \boldsymbol{x}) = 0. \tag{7.16}$$

Using (7.15) and (7.18) we find

$$D_{\boldsymbol{Z}_l}\phi_l(\boldsymbol{z})\bar{\boldsymbol{F}}_l^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}) = 0, \tag{7.17}$$

meaning that ϕ_l is also a conservation laws of the vector field $\bar{\boldsymbol{F}}_l^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z})$.

The existence of such approximated conservation laws implies failure of the condition (7.11) in the Theorem 7.3 as in the following

Proposition 7.10. *Let us consider that there is an approximate conservation law $\phi_l(\boldsymbol{X}_l)$, where $1 \leq l \leq m-1$. Then $|D_{\boldsymbol{Z}_l}\bar{\boldsymbol{F}}_l^{(1)}| = 0$ if $\boldsymbol{z} \in \mathcal{M}_l$.*

Proof. From the definition of approximate conservation laws we have

$$D_{\boldsymbol{X}_l}\phi_l(\boldsymbol{X}_l)\boldsymbol{F}_l^{(1)}(\boldsymbol{k}, \boldsymbol{x}) = 0. \tag{7.18}$$

Because the polynomials $\boldsymbol{F}_l^{(1)}$ and $\bar{\boldsymbol{F}}_l^{(1)}$ are identical we find

$$D_{\boldsymbol{Z}_l}\phi_l(\boldsymbol{Z}_l)\bar{\boldsymbol{F}}_l^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}) = 0. \tag{7.19}$$

By differentiating (7.19) we get $\frac{d^2\phi_l}{d\boldsymbol{Z}_l{}^2}\bar{\boldsymbol{F}}_l^{(1)} + \frac{d\phi_l}{d\boldsymbol{Z}_l}\frac{d\bar{\boldsymbol{F}}_l^{(1)}}{d\boldsymbol{Z}_l} = 0$, where $\frac{d^2\phi_l}{d\boldsymbol{Z}_l{}^2}$ is the second derivative of ϕ_l with respect to $\boldsymbol{Z}_l$. If $\boldsymbol{z} \in \mathcal{M}_l$, then $\bar{\boldsymbol{F}}_l^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{z}) = 0$ and $\frac{d\phi_l}{d\boldsymbol{Z}_l}\frac{d\bar{\boldsymbol{F}}_l^{(1)}}{d\boldsymbol{Z}_l} = 0$. The left kernel of the matrix $\frac{d\bar{\boldsymbol{F}}_l^{(1)}}{d\boldsymbol{Z}_l}$ contains the covector $\frac{d\phi_l}{d\boldsymbol{Z}_l} \neq 0$, therefore $|D_{\boldsymbol{Z}_l}\bar{\boldsymbol{F}}_l^{(1)}| = 0$. □

In this case, the variables $\boldsymbol{X}_l$ will be eliminated from the extended set of equations

$$\boldsymbol{F}_l^{(1)}(\boldsymbol{X}_l, \boldsymbol{x}_{l+1}, \ldots, \boldsymbol{x}_m) = 0,\ \Phi_l(\boldsymbol{X}_l) = \boldsymbol{x}_l^c, \tag{7.20}$$

where Φ_l is a complete set of approximate conservation laws. We call (7.20) the quasi-equilibrium conditions.

More precisely, let us consider that

Condition 7.11. *The following conditions are satisfied:*

1. *There exists $\boldsymbol{x} \in \mathbb{R}^n_{>0}$ such that $\boldsymbol{F}_l^{(1)}(\boldsymbol{x}) = 0$. For all $\boldsymbol{x} \in \mathbb{R}^n_{>0}$ such that $\boldsymbol{F}_l^{(1)}(\boldsymbol{x}) = 0$, $|D_{\boldsymbol{X}_k}\boldsymbol{F}_k^{(1)}| \neq 0$ for all $k \in \{1, \ldots, l-1\}$, and $|D_{\boldsymbol{X}_l}\boldsymbol{F}_l^{(1)}| = 0$.*
2. *There is a set of s_l, where $0 < s_l \leq n_l$, irreducible approximate conservation laws $\boldsymbol{\Phi}_l(\boldsymbol{x}) = (\phi_{1l}(\boldsymbol{x}), \ldots, \phi_{s_l l}(\boldsymbol{x}))^\mathsf{T}$ depending only on $\boldsymbol{X}_l$, such that*

$$(D_{\boldsymbol{X}_l}\boldsymbol{\Phi}_l)\boldsymbol{F}_l^{(1)} = 0.$$

   These conservation laws are complete, namely

$$rk\left(D_{\boldsymbol{X}_l}\begin{pmatrix}\boldsymbol{F}_l^{(1)}\\ \boldsymbol{\Phi}_l\end{pmatrix}\right) = n_1 + \ldots + n_l, \tag{7.21}$$

   for $\boldsymbol{x} \in \mathbb{R}^n_{>0}$ such that $\boldsymbol{F}_l^{(1)}(\boldsymbol{x}) = 0$.
3. *These conservation laws are independent as functions of $\boldsymbol{x}_l$, namely*

$$rk(D_{\boldsymbol{x}_l}\boldsymbol{\Phi}_l) = s_l. \tag{7.22}$$

4. *Furthermore, $\begin{pmatrix}\boldsymbol{F}_k^{(1)}\\ \boldsymbol{\Phi}_l\end{pmatrix}$ are independent as functions of $(\boldsymbol{X}_k, \boldsymbol{x}_l)$, namely*

$$rk\left(D_{(\boldsymbol{X}_k, \boldsymbol{x}_l)}\begin{pmatrix}\boldsymbol{F}_k^{(1)}\\ \boldsymbol{\Phi}_l\end{pmatrix}\right) = n_1 + \ldots + n_k + s_l, \tag{7.23}$$

   for all $1 \leq k \leq l-1$ and $\boldsymbol{x} \in \mathbb{R}^n_{>0}$ such that $\boldsymbol{F}_l^{(1)}(\boldsymbol{x}) = 0$.

In this case, the CRN (2.2) can be modified in such a way that the condition (7.11) is fulfilled for $k \in \{1, \ldots, l\}$. This modification is detailed below.

First, consider the equation

$$\boldsymbol{x}_l^c = \boldsymbol{\Phi}_l(\boldsymbol{X}_l). \tag{7.24}$$

From (7.22) it follows that, up to a relabelling of the coordinates of $\boldsymbol{X}_l$, we have the following splitting $\boldsymbol{X}_l = (\boldsymbol{X}_{l-1}, \hat{\boldsymbol{x}}_l, \check{\boldsymbol{x}}_l)$, where $\boldsymbol{X}_{l-1} \in \mathbb{R}^{n_1+\ldots+n_{l-1}}$, $\hat{\boldsymbol{x}}_l \in \mathbb{R}^{n_l - s_l}$, $\check{\boldsymbol{x}}_l \in$

$\mathbb{R}^{s_l}$ and $|D_{\check{\boldsymbol{x}}_l}\boldsymbol{\Phi}_l| \neq 0$. Hence, (7.24) defines the implicit function $\check{\boldsymbol{x}}_l = \Psi_l(\boldsymbol{X}_{l-1}, \hat{\boldsymbol{x}}_l, \boldsymbol{x}_l^c)$, allowing to eliminate the variables $\check{\boldsymbol{x}}_l$. The above splitting of $\boldsymbol{X}_l$ induces the splitting $\boldsymbol{F}_l^{(i)} = (\boldsymbol{F}_{l-1}^{(i)}, \hat{\boldsymbol{f}}_l^{(i)}, \check{\boldsymbol{f}}_l^{(i)})$, for $i \in \{1, 2\}$.

Let us define the functions

$$\boldsymbol{F}_k^{red,i}(\boldsymbol{k}, \boldsymbol{X}_{l-1}, \hat{\boldsymbol{x}}_l, \boldsymbol{x}_l^c, \boldsymbol{x}_{l+1}, \boldsymbol{x}_{l+2}, \ldots, \boldsymbol{x}_m) = \\ \hat{\boldsymbol{F}}_k^{(i)}(\boldsymbol{k}, \boldsymbol{X}_{l-1}, \hat{\boldsymbol{x}}_l, \Psi_l(\boldsymbol{X}_{l-1}, \hat{\boldsymbol{x}}_l, \boldsymbol{x}_l^c), \boldsymbol{x}_{l+1}, \boldsymbol{x}_{l+2}, \ldots, \boldsymbol{x}_m), \tag{7.25}$$

where $\hat{\boldsymbol{F}}_k^{(i)} = \boldsymbol{F}_k^{(i)}$, for $k \in \{1, \ldots, l-1\}$, and $\hat{\boldsymbol{F}}_l^{(i)} = (\boldsymbol{F}_{l-1}^{(i)}, \hat{\boldsymbol{f}}_l^{(i)})$, for $i \in \{1, 2\}$.

Theorem 7.12. *If the Conditions 7.11 are fulfilled and* $\boldsymbol{F}_l^{red,1}(\boldsymbol{k}, \boldsymbol{X}_{l-1}, \hat{\boldsymbol{x}}_l, \boldsymbol{x}_l^c, \boldsymbol{x}_{l+1}, \boldsymbol{x}_{l+2}, \ldots, \boldsymbol{x}_m) = 0$*, then* $|D_{\hat{\boldsymbol{X}}_k}\boldsymbol{F}_k^{(red,1)}| \neq 0$*, where* $\hat{\boldsymbol{X}}_k = \boldsymbol{X}_k$*, for* $1 \leq k \leq l-1$*, and* $\hat{\boldsymbol{X}}_l = (\boldsymbol{X}_{l-1}, \hat{\boldsymbol{x}}_l)$*.*

The proof is based on the following lemmas

Lemma 7.13. **i)** $D_{\hat{\boldsymbol{X}}_l}\boldsymbol{F}_l^{red,1} = D_{\boldsymbol{X}_l}\begin{pmatrix}\hat{\boldsymbol{F}}_l^{(1)} \\ \boldsymbol{\Phi}_l\end{pmatrix} \Big/ D_{\check{\boldsymbol{x}}_l}\boldsymbol{\Phi}_l$ *and*

ii) $D_{\boldsymbol{X}_k}\boldsymbol{F}_l^{red,1} = D_{(\boldsymbol{X}_k, \check{\boldsymbol{x}}_l)}\begin{pmatrix}\hat{\boldsymbol{F}}_l^{(1)} \\ \boldsymbol{\Phi}_l\end{pmatrix} \Big/ D_{\check{\boldsymbol{x}}_l}\boldsymbol{\Phi}_l$*, for* $1 \leq k \leq l-1$*.*

Proof. Differentiating (7.24) with respect to $\hat{\boldsymbol{X}}_l$ we get $D_{\hat{\boldsymbol{X}}_l}\boldsymbol{\Phi}_l + D_{\check{\boldsymbol{x}}_l}\boldsymbol{\Phi}_l D_{\hat{\boldsymbol{X}}_l}\boldsymbol{\Psi}_l = 0$. From the definition of $\hat{\boldsymbol{F}}_l^{red,1}$ it follows $D_{\hat{\boldsymbol{X}}_l}\hat{\boldsymbol{F}}_l^{red,1} = D_{\hat{\boldsymbol{X}}_l}\hat{\boldsymbol{F}}_l^{(1)} + D_{\check{\boldsymbol{x}}_l}\hat{\boldsymbol{F}}_l^{(1)} D_{\hat{\boldsymbol{X}}_l}\boldsymbol{\Psi}_l = D_{\hat{\boldsymbol{X}}_l}\hat{\boldsymbol{F}}_l^{(1)} - D_{\check{\boldsymbol{x}}_l}\hat{\boldsymbol{F}}_l^{(1)}(D_{\check{\boldsymbol{x}}_l}\boldsymbol{\Phi}_l)^{-1} D_{\hat{\boldsymbol{X}}_l}\boldsymbol{\Phi}_l$ which implies i). ii) is proved similarly. □

Lemma 7.14. *For all* $\boldsymbol{x}$ *such that* $\boldsymbol{F}_l^{(1)}(\boldsymbol{x}) = 0$ *we have*

i) $rk\left(D_{\boldsymbol{X}_l}\begin{pmatrix}\boldsymbol{F}_l^{(1)} \\ \boldsymbol{\Phi}_l\end{pmatrix}\right) = rk\left(D_{\boldsymbol{X}_l}\begin{pmatrix}\hat{\boldsymbol{F}}_l^{(1)} \\ \boldsymbol{\Phi}_l\end{pmatrix}\right)$ *and*

ii) $rk\left(D_{(\boldsymbol{X}_k, \boldsymbol{x}_l)}\begin{pmatrix}\boldsymbol{F}_k^{(1)} \\ \boldsymbol{\Phi}_l\end{pmatrix}\right) = rk\left(D_{(\boldsymbol{X}_k, \check{\boldsymbol{x}}_l)}\begin{pmatrix}\hat{\boldsymbol{F}}_l^{(1)} \\ \boldsymbol{\Phi}_l\end{pmatrix}\right)$.

Proof. Differentiating $(D_{\boldsymbol{X}_l}\boldsymbol{\Phi}_l)\boldsymbol{F}_l^{(1)} = 0$ with respect to $\hat{\boldsymbol{X}}_l$ we get $(D_{\boldsymbol{X}_l}\boldsymbol{\Phi}_l)(D_{\boldsymbol{X}_l}\boldsymbol{F}_l^{(1)}) = 0$. In block matrix form this reads

$$\begin{pmatrix} D_{\hat{\boldsymbol{X}}_l}\boldsymbol{\Phi}_l & D_{\check{\boldsymbol{x}}_l}\boldsymbol{\Phi}_l \end{pmatrix} \begin{pmatrix} D_{\hat{\boldsymbol{X}}_l}\hat{\boldsymbol{F}}_l^{(1)} & D_{\check{\boldsymbol{x}}_l}\hat{\boldsymbol{F}}_l^{(1)} \\ D_{\hat{\boldsymbol{X}}_l}\check{\boldsymbol{f}}_l^{(1)} & D_{\check{\boldsymbol{x}}_l}\check{\boldsymbol{f}}_l^{(1)} \end{pmatrix} = \begin{pmatrix} 0 & 0 \end{pmatrix},$$

that leads to $(D_{\hat{\boldsymbol{X}}_l}\boldsymbol{\Phi}_l)(D_{\hat{\boldsymbol{X}}_l}\hat{\boldsymbol{F}}_l^{(1)}) + (D_{\check{\boldsymbol{x}}_l}\boldsymbol{\Phi}_l)(D_{\hat{\boldsymbol{X}}_l}\check{\boldsymbol{f}}_l^{(1)}) = 0$ and $(D_{\hat{\boldsymbol{X}}_l}\boldsymbol{\Phi}_l)(D_{\check{\boldsymbol{x}}_l}\hat{\boldsymbol{F}}_l^{(1)}) + (D_{\check{\boldsymbol{x}}_l}\Phi_l)(D_{\check{\boldsymbol{x}}_l}\check{\boldsymbol{F}}_l^{(1)}) = 0$. It follows that $D\hat{\boldsymbol{X}}_l\check{\boldsymbol{f}}_l^{(1)} = \Lambda D_{\hat{\boldsymbol{X}}_l}\hat{\boldsymbol{F}}_l^{(1)}$ and $D\check{\boldsymbol{x}}_l\check{\boldsymbol{f}}_l^{(1)} = \Lambda D_{\check{\boldsymbol{x}}_l}\hat{\boldsymbol{F}}_l^{(1)}$ where $\Lambda = (D_{\check{\boldsymbol{x}}_l}\boldsymbol{\Phi}_l)^{-1}(D_{\hat{\boldsymbol{X}}_l}\boldsymbol{\Phi}_l)$ which proves i). ii) is proven similarly. □

The proof of the Theorem 7.12 follows.

Proof. Let us show that $D_{\hat{\boldsymbol{X}}_l}\boldsymbol{F}_l^{red,1}$ and $D_{\hat{\boldsymbol{X}}_k}\boldsymbol{F}_k^{red,1}$, $1 \leq k \leq l-1$, are invertible. Using the completeness of $\boldsymbol{\Phi}_l$, the Guttman rank additivity formula (7.13), Lemma 7.13, and Lemma 7.14 we find that $rk\left(D_{\hat{\boldsymbol{X}}_l}\boldsymbol{F}_l^{red,1}\right) = n_1 + n_2 + \ldots + n_l - s_l$, hence $D_{\hat{\boldsymbol{X}}_l}\hat{\boldsymbol{F}}_l^{red}$ is invertible. Similarly, $rk\left(D_{\hat{\boldsymbol{X}}_k}\boldsymbol{F}_k^{red,1}\right) = n_1 + n_2 + \ldots + n_k$, hence $D_{\hat{\boldsymbol{X}}_k}\hat{\boldsymbol{F}}_k^{red}$ is invertible, for $1 \leq k \leq l-1$.

□

Remark 7.15. The 4th part of Condition 7.11 is automatically satisfied if the functions $\boldsymbol{F}_k^{(1)}$ do not depend on $\boldsymbol{x}_l$, for $1 \leq k \leq l-1$.

Remark 7.16. The Theorem 7.12 allows defining $l-$chains in the case of quasi-equilibrium. The set of positive solutions of

$$\boldsymbol{F}_k^{red,1}(\boldsymbol{k}, \boldsymbol{X}_{l-1}, \hat{\boldsymbol{x}}_l, \boldsymbol{x}_l^c, \boldsymbol{x}_{l+1}, \ldots, \boldsymbol{x}_m) = 0$$

is denoted $\mathcal{M}_k^{QE}$ and represents the k-th *quasi-equilibrium variety* (intersected with the first orthant). These sets satisfy $\mathcal{M}_0^{QE} = \mathbb{R}^n_{>0} \supset \mathcal{M}_1^{QE} \supset \ldots \supset \mathcal{M}_l^{QE}$. The concept of hyperbolically attractive $l-$chains is applicable to quasi-equilibrium varieties as well. It is obtained by replacing $\boldsymbol{Z}_k$ with $\hat{\boldsymbol{X}}_k$ and $\bar{\boldsymbol{F}}_k^{(1)}$ with $\boldsymbol{F}_k^{red,1}$ in Proposition 7.8.

The transformed functions (7.25) allow to define the transformed system of differential equations

$$\dot{\hat{\boldsymbol{X}}}_l = \boldsymbol{F}_l^{red,1} + \boldsymbol{F}_l^{red,2}, \tag{7.26}$$

$$\dot{\boldsymbol{x}}_l^c = (D_{\boldsymbol{X}_l}\boldsymbol{\Phi}_l)\boldsymbol{F}_l^{red,2}, \tag{7.27}$$

$$\dot{\boldsymbol{x}}_k = \boldsymbol{f}_k(\boldsymbol{k}, \boldsymbol{X}_{l-1}, \hat{\boldsymbol{x}}_l, \Psi_l(\boldsymbol{X}_{l-1}, \hat{\boldsymbol{x}}_l, \boldsymbol{x}_l^c), \boldsymbol{x}_{l+1}, \boldsymbol{x}_{l+2}, \ldots, \boldsymbol{x}_m),\ k \in \{l+1, \ldots, m\} \tag{7.28}$$

By the results of the Section **??**, the new variables $\boldsymbol{x}_l^c$ are slower than $\hat{\boldsymbol{X}}_l$, and by Theorem 7.12, the transformed system satisfies the non-degeneracy conditions $|D_{\hat{\boldsymbol{X}}_k}F_k^{(red,1)}| \neq 0$ up to order l. If the approximate conservation laws are also exact, the new variables $\boldsymbol{x}_l^c$ are constant and stand for new parameters. Then, the new equations (7.27) are not added to the transformed ones (7.26),(7.28). In this case as well, the transformed system (7.26),(7.28) satisfies the non-degeneracy conditions up to order l.

7.4 Algorithmic solutions for eliminating the approximate conservation laws

The previous section allows us to define an algorithm transforming the CRN (2.2) into an equivalent CRN that does not have approximate conservation laws and that can be further reduced using the method introduced in [Kruff et al., 2020]. According to this method, some old variables are substituted by new ones, representing approximate conservation laws.

Algorithm 20 transforms the CRN into another CRN that satisfies the condition $|D_{\boldsymbol{X}_l}F_k| \neq 0$ for $k \in \{1, \ldots, l\}$ up to the l-th timescale. It further iterates the procedure for increasing l, computes a rescaled and truncated version of the CRN at each step by implementing the algorithm ScaleAndTruncate introduced in [Kruff et al., 2020].

Input: A CRN given by a polynomial vector field $\boldsymbol{F}(\boldsymbol{k}, \boldsymbol{x})$.

Output: A transformed CRN given by a modified polynomial vector field.

1: ScaleAndTruncate.
2: l:=0
3: **while** $l < m$ **do**
4: **while** $|D_{\boldsymbol{X}_l} F_l^{(1)}| \neq 0$ **do**
5: l:=l+1
6: **end while**
7: Find a complete set $\boldsymbol{\Phi}_l$ of independent conservation laws for $\boldsymbol{F}_l^{(1)}$ satisfying the conditions (7.21), (7.22).
8: Compute $\psi_{il}(\boldsymbol{X}_{l-1}, \hat{\boldsymbol{x}}_l, \boldsymbol{x}_l^c)$ the solution of the equation $\boldsymbol{x}_l^c = \boldsymbol{\Phi_l}(\boldsymbol{x})$.
9: **for** $i := 1$ **to** s_l **do**
10: **if** Φ_{il} is not total **then**
11: Replace the ODE satisfied by $\check{x}_{il}$ by $\dot{x}_{il}^c = D_{\boldsymbol{x}} \Phi_{il} \boldsymbol{F}(\boldsymbol{k}, \boldsymbol{x})$
12: Substitute $\check{x}_{il} \leftarrow \psi_{il}(\boldsymbol{X}_{l-1}, \hat{\boldsymbol{x}}_l, \boldsymbol{x}_l^c)$
13: **else**
14: Delete the ODE satisfied by $\check{x}_{il}$
15: Define new constants k_{il}^c and concatenate them to $\boldsymbol{k}$.
16: Substitute $\check{x}_{il} \leftarrow \psi_{il}(\boldsymbol{X}_{l-1}, \hat{\boldsymbol{x}}_l, \boldsymbol{k}_l^c)$
17: **end if**
18: **end for**
19: ScaleAndTruncate.
20: **end while**

Algorithm 20: TransformCRNexplicit

If none of the approximate conservation laws used in the transformation are exact, the resulting CRN has the same number of variables, ODEs and parameters as the initial one. Any exact conservation law used in the transformation reduces both numbers of variables and ODEs by one and increases the number of parameters by one.

Because at each step the total number of variables can only decrease, the total number of variables having slower timescales and remaining to be treated is strictly decreasing. This shows that the algorithm terminates in a finite number of steps. Furthermore, at each step l of the algorithm the set of variables $\boldsymbol{X}_k$ having timescales of order δ^{b_k}, remains unchanged for all $1 \leq k \leq l-1$. This is a consequence of the fact (see Theorems 6.13,6.16,**??**) that approximate conservation laws are slower than the variables $\boldsymbol{X}_l$ and need to be treated at later steps. This property guarantees that the transformed system satisfies $D_{\boldsymbol{X}_l} \boldsymbol{F}_l^{(1)} = 0$ for $1 \leq l \leq m$ and therefore has no approximate conservations.

The applicability of the Algorithm 20 is limited by the possibility of solving the equation $\boldsymbol{x}_l = \boldsymbol{\Phi}_l(\boldsymbol{x})$ symbolically (step 8 of the algorithm). This is always possible when the approximate conservation laws $\boldsymbol{\Phi}_l$ are linear, but may not be easy when $\boldsymbol{\Phi}_l$ are polynomial. However, for most biochemical CRN models used in computational biology, linear conservation laws are enough for obtaining completeness (condition (7.21)).

If one wants to avoid the elimination step (for instance, for polynomial conservation laws) there is another possible algorithmic solution whose output is a differential algebraic

Input: A CRN given by a polynomial vector field $\boldsymbol{F}(\boldsymbol{k}, \boldsymbol{x})$.

Output: A differential algebraic CRN given by a modified polynomial vector field and a set of algebraic constraints.

```
ScaleAndTruncate.
l:=0
while l < m do
    while |D_{X_l} F_l^{(1)}| ≠ 0 do
        l:=l+1
    end while
    Find a complete set Φ_l of independent conservation laws for F_l^{(1)} satisfying the
    conditions (7.21), (7.22).
    for i := 1 to s_l do
        if Φ_{il} is not total then
            Replace the ODE satisfied by x̌_{il} by ẋ^c_{il} = D_x Φ_{il} F(k, x)
            Add x^c_l = Φ_{il}(x) to the set of algebraic constraints.
        else
            Delete the ODE satisfied by x̌_{il}
            Define new constants k^c_{il} and concatenate them to k.
            Add k^c_{il} = Φ_{il}(x) to the set of algebraic constraints.
        end if
    end for
    ScaleAndTruncate.
end while
```

Algorithm 21: TransformCRNimplicit

transformed system. More precisely, at each step l one concatenates the truncated vector field $\bar{\boldsymbol{F}}_l^{(l)}(\boldsymbol{k}, \boldsymbol{z})$ and the conservation law $\Phi_l(\boldsymbol{z})$. The first one is used for the ODE part of the transformed system and the second defines the algebraic constraint (7.24). This choice is implemented in Algorithm 21. Using the Lemma 7.13 and the Proposition 7.8 we can show that the hyperbolicity test justifying this reduction should be performed on the eigenvalues of the Schur complement

$$\left(D_{\boldsymbol{Z}_l} \begin{pmatrix} \hat{\boldsymbol{F}}_l^{(1)} \\ \boldsymbol{\Phi}_l \end{pmatrix} \Big/ D_{\check{\boldsymbol{z}}_l} \boldsymbol{\Phi}_l \right) \Big/ \left(D_{\boldsymbol{Z}_{l-1}} \begin{pmatrix} \hat{\boldsymbol{F}}_{l-1}^{(1)} \\ \boldsymbol{\Phi}_{l-1} \end{pmatrix} \Big/ D_{\check{\boldsymbol{z}}_{l-1}} \boldsymbol{\Phi}_{l-1} \right).$$

7.5 Case study: TGFβ

7.5.1 Model and its scaling

The TGFβ signaling model including including transcriptional repression of SMAD by TIF1γ is described by 21 ODEs [Andrieux et al., 2012]:

$$
\begin{aligned}
\dot{x}_1 &= k_2x_2 - k_1x_1 - k_{16}x_1x_{11} \\
\dot{x}_2 &= k_1x_1 - k_2x_2 + k_{17}k_{36}x_6 \\
\dot{x}_3 &= k_3x_4 - k_3x_3 + k_7x_7 + k_{33}k_{38}x_{20} - k_6x_3x_5 \\
\dot{x}_4 &= k_3x_3 - k_3x_4 + k_9x_8 - k_8x_4x_6 \\
\dot{x}_5 &= k_5x_6 - k_4x_5 + k_7x_7 + 2k_{11}x_9 - 2k_{10}x_5^2 - k_6x_3x_5 + k_{16}x_1x_{11} \\
\dot{x}_6 &= k_4x_5 - k_5x_6 + k_9x_8 + 2k_{13}x_{10} + k_{35}x_{21} - 2k_{12}x_6^2 - k_{17}k_{36}x_6 - k_8x_4x_6 \\
\dot{x}_7 &= k_6x_3x_5 - x_7(k_7 + k_{14}) \\
\dot{x}_8 &= k_{14}x_7 - k_9x_8 + k_8x_4x_6 - k_{31}x_8x_{17} \\
\dot{x}_9 &= k_{10}x_5^2 - x_9(k_{11} + k_{15}) \\
\dot{x}_{10} &= k_{15}x_9 - k_{13}x_{10} + k_{12}x_6^2 \\
\dot{x}_{11} &= k_{23}x_{14} - k_{30}x_{11} \\
\dot{x}_{12} &= k_{18} - x_{12}(k_{20} + k_{26}) + k_{30}x_{11} + k_{27}x_{15} - k_{22}k_{37}x_{12}x_{13} \\
\dot{x}_{13} &= k_{19} - x_{13}(k_{21} + k_{28}) + k_{30}x_{11} + k_{29}x_{16} - k_{22}k_{37}x_{12}x_{13} \\
\dot{x}_{14} &= k_{22}k_{37}x_{12}x_{13} - x_{14}(k_{23} + k_{24} + k_{25}) \\
\dot{x}_{15} &= k_{26}x_{12} - k_{27}x_{15} \\
\dot{x}_{16} &= k_{28}x_{13} - k_{29}x_{16} \\
\dot{x}_{17} &= k_{35}x_{21} - k_{31}x_8x_{17} \\
\dot{x}_{18} &= k_{31}x_8x_{17} - k_{34}x_{18} \\
\dot{x}_{19} &= k_{34}x_{18} - k_{32}x_{19} \\
\dot{x}_{20} &= k_{32}x_{19} - k_{33}k_{38}x_{20} \\
\dot{x}_{21} &= k_{34}x_{18} - k_{35}x_{21}
\end{aligned}
\tag{7.29}
$$

This model is particularly interesting because it contains multiple exact and approximate conservation laws, at various timescales. The model has three exact linear conservations laws $x_{17}+x_{18}+x_{21}$, $x_1+x_2+x_5+x_6+x_7+x_8+2x_9+2x_{10}+x_{18}+x_{21}$, $x_3+x_4+x_7+x_8+x_{18}+x_{19}+x_{20}$, that can be interpreted as the total amounts of TIF1γ, SMAD2, and SMAD4, respectivelly.

We propose a reduction based on the full tropical equilibration

$$\boldsymbol{a} = (-2, -1, -2, -2, 0, 0, -1, -1, 1, 1, 1, 1, 1, 4, 2, 0, 3, -1, -1, -1),$$

computed for $\epsilon = 1/11$. This equilibration is the closest, in logarithmic coordinates, to the steady state of the TGFb model.

The rescaled set of ODEs is

$$
\begin{aligned}
\dot{y}_1 &= \epsilon^2(\bar{k}_2 y_2 - \bar{k}_1 y_1 - \epsilon^2 \bar{k}_{16} y_1 y_{11}) \\
\dot{y}_2 &= \epsilon^1(\bar{k}_1 y_1 + \epsilon^2 \bar{k}_{17} \bar{k}_{36} y_6 - \bar{k}_2 y_2) \\
\dot{y}_3 &= \epsilon^2(\bar{k}_3 y_4 + \epsilon \bar{k}_7 y_7 + \epsilon \bar{k}_{33} \bar{k}_{38} y_{20} - \bar{k}_3 y_3 - \epsilon \bar{k}_6 y_3 y_5) \\
\dot{y}_4 &= \epsilon^2(\bar{k}_3 y_3 + \epsilon \bar{k}_9 y_8 - \bar{k}_3 y_4 - \epsilon \bar{k}_8 y_4 y_6) \\
\dot{y}_5 &= \epsilon^1(\bar{k}_5 y_6 + \bar{k}_7 y_7 + 2\epsilon^2 \bar{k}_{11} y_9 + \epsilon \bar{k}_{16} y_1 y_{11} - 2\epsilon^2 \bar{k}_{10} y_5^2 - \epsilon \bar{k}_4 y_5 - \bar{k}_6 y_3 y_5) \\
\dot{y}_6 &= \epsilon^1(\bar{k}_9 y_8 + \bar{k}_{35} y_{21} + 2\epsilon^2 \bar{k}_{13} y_{10} + \epsilon \bar{k}_4 y_5 - \bar{k}_5 y_6 - 2\epsilon^2 \bar{k}_{12} y_6^2 - \bar{k}_8 y_4 y_6 - \epsilon \bar{k}_{17} \bar{k}_{36} y_6) \\
\dot{y}_7 &= \epsilon^2(\bar{k}_6 y_3 y_5 - \bar{k}_7 y_7 - \bar{k}_{14} y_7) \\
\dot{y}_8 &= \epsilon^2(\bar{k}_{14} y_7 + \bar{k}_8 y_4 y_6 - \bar{k}_9 y_8 - \bar{k}_{31} y_8 y_{17}) \\
\dot{y}_9 &= \epsilon^2(\bar{k}_{10} y_5^2 - \bar{k}_{11} y_9 - \bar{k}_{15} y_9) \\
\dot{y}_{10} &= \epsilon^2(\bar{k}_{15} y_9 + \bar{k}_{12} y_6^2 - \bar{k}_{13} y_{10}) \\
\dot{y}_{11} &= \epsilon^3(\bar{k}_{23} y_{14} - \bar{k}_{30} y_{11}) \\
\dot{y}_{12} &= \epsilon^2(\epsilon \bar{k}_{18} + \bar{k}_{27} y_{15} + \epsilon \bar{k}_{30} y_{11} - \bar{k}_{26} y_{12} - \epsilon \bar{k}_{20} y_{12} - \epsilon \bar{k}_{22} \bar{k}_{37} y_{12} y_{13}) \\
\dot{y}_{13} &= \epsilon^0(\bar{k}_{19} + \bar{k}_{30} y_{11} + \epsilon^2 \bar{k}_{29} y_{16} - \epsilon^3 \bar{k}_{21} y_{13} - \epsilon^2 \bar{k}_{28} y_{13} - \bar{k}_{22} \bar{k}_{37} y_{12} y_{13}) \\
\dot{y}_{14} &= \epsilon^2(\bar{k}_{22} \bar{k}_{37} y_{12} y_{13} - \bar{k}_{23} y_{14} - \bar{k}_{25} y_{14} - \epsilon \bar{k}_{24} y_{14}) \\
\dot{y}_{15} &= \epsilon^3(\bar{k}_{26} y_{12} - \bar{k}_{27} y_{15}) \\
\dot{y}_{16} &= \epsilon^3(\bar{k}_{28} y_{13} - \bar{k}_{29} y_{16}) \\
\dot{y}_{17} &= \epsilon^2(\bar{k}_{35} y_{21} - \bar{k}_{31} y_8 y_{17}) \\
\dot{y}_{18} &= \epsilon^2(\bar{k}_{31} y_8 y_{17} - \bar{k}_{34} y_{18}) \\
\dot{y}_{19} &= \epsilon^2(\bar{k}_{34} y_{18} - \bar{k}_{32} y_{19}) \\
\dot{y}_{20} &= \epsilon^1(\bar{k}_{32} y_{19} - \bar{k}_{33} \bar{k}_{38} y_{20}) \\
\dot{y}_{21} &= \epsilon^2(\bar{k}_{34} y_{18} - \bar{k}_{35} y_{21}),
\end{aligned}
\tag{7.30}
$$

and the truncated set is

$$
\begin{aligned}
\dot{y}_1 &= \epsilon^2(k_2 y_2 - \bar{k}_1 y_1) \\
\dot{y}_2 &= \epsilon^1(\bar{k}_1 y_1 - \bar{k}_2 y_2) \\
\dot{y}_3 &= \epsilon^2(\bar{k}_3 y_4 - \bar{k}_3 y_3) \\
\dot{y}_4 &= \epsilon^2(\bar{k}_3 y_3 - \bar{k}_3 y_4) \\
\dot{y}_5 &= \epsilon^1(\bar{k}_5 y_6 + \bar{k}_7 y_7 - \bar{k}_6 y_3 y_5) \\
\dot{y}_6 &= \epsilon^1(\bar{k}_9 y_8 + \bar{k}_{35} y_{21} - \bar{k}_8 y_4 y_6) \\
\dot{y}_7 &= \epsilon^2(\bar{k}_6 y_3 y_5 - \bar{k}_7 y_7 - \bar{k}_{14} y_7) \\
\dot{y}_8 &= \epsilon^2(\bar{k}_{14} y_7 + \bar{k}_8 y_4 y_6 - \bar{k}_9 y_8 - \bar{k}_{31} y_8 y_{17}) \\
\dot{y}_9 &= \epsilon^2(\bar{k}_{10} y_5^2 - \bar{k}_{11} y_9 - \bar{k}_{15} y_9) \\
\dot{y}_{10} &= \epsilon^2(\bar{k}_{15} y_9 + \bar{k}_{12} y_6^2 - \bar{k}_{13} y_{10}) \\
\dot{y}_{11} &= \epsilon^3(\bar{k}_{23} y_{14} - \bar{k}_{30} y_{11}) \\
\dot{y}_{12} &= \epsilon^2(\bar{k}_{27} y_{15} - \bar{k}_{26} y_{12}) \\
\dot{y}_{13} &= \epsilon^0(\bar{k}_{19} + \bar{k}_{30} y_{11} - \bar{k}_{22} \bar{k}_{37} y_{12} y_{13}) \\
\dot{y}_{14} &= \epsilon^2(\bar{k}_{22} \bar{k}_{37} y_{12} y_{13} - \bar{k}_{23} y_{14} - \bar{k}_{25} y_{14}) \\
\dot{y}_{15} &= \epsilon^3(\bar{k}_{26} y_{12} - \bar{k}_{27} y_{15}) \\
\dot{y}_{16} &= \epsilon^3(\bar{k}_{28} y_{13} - \bar{k}_{29} y_{16}) \\
\dot{y}_{17} &= \epsilon^2(\bar{k}_{35} y_{21} - \bar{k}_{31} y_8 y_{17}) \\
\dot{y}_{18} &= \epsilon^2(\bar{k}_{31} y_8 y_{17} - \bar{k}_{34} y_{18}) \\
\dot{y}_{19} &= \epsilon^2(\bar{k}_{34} y_{18} - \bar{k}_{32} y_{19}) \\
\dot{y}_{20} &= \epsilon^1(\bar{k}_{32} y_{19} - \bar{k}_{33} \bar{k}_{38} y_{20}) \\
\dot{y}_{21} &= \epsilon^2(\bar{k}_{34} y_{18} - \bar{k}_{35} y_{21}),
\end{aligned}
\tag{7.31}
$$

After this scaling four timescales are apparent, represented by the reciprocal orders, in order from the fastest to the slowest: ϵ^0, ϵ^1, ϵ^2, ϵ^3. The corresponding groups of variables are from the fastest to the slowest: $\boldsymbol{z}_1 = y_{13}$, $\boldsymbol{z}_2 = (y_2, y_5, y_6, y_{20})$, $\boldsymbol{z}_3 = (y_1, y_3, y_4, y_7, y_8, y_9, y_{10}, y_{12}, y_{14}, y_{17}, y_{18}, y_{19}, y_{21})$, $\boldsymbol{z}_4 = (y_{11}, y_{15}, y_{16})$.

7.5.2 Elimination of the conservation laws

We now transform the model into an equivalent one that has no conservation laws.

At the first steps of the transformation algorithm $|D_{\boldsymbol{Z}_1}\bar{\boldsymbol{F}}_1^{(1)}| = -k_{22}k_{37}y_{12} \neq 0$, $|D_{\boldsymbol{Z}_2}\bar{\boldsymbol{F}}_2^{(1)}| = -k_2k_6k_{22}k_{33}k_{37}k_{38}y_3y_{12}(k_5 + k_8y_4) \neq 0$, but $|D_{\boldsymbol{Z}_3}\bar{\boldsymbol{F}}_3^{(1)}| = 0$. Building a stoichiometric matrix $\boldsymbol{S}_3^{(1)}$ from $\bar{\boldsymbol{F}}_3^{(1)}$ we find four linear, independent, approximate conservation laws that satisfy the conditions: $y_1 + y_2$, $y_3 + y_4$, $y_5 + y_6 + y_7 + y_8 + y_{18} + y_{21}$, $y_{17} + y_{18} + y_{21}$. The last one is an exact conservation law that we have already interpreted. The first three are approximate conservation laws and can be interpreted as: the total free (non complexified) SMAD2 in cytosol and nucleus, the total free (non modified, non complexified) SMAD4 in cytosol and nucleus, and the total SMAD2 phosphorylated or forming heteromers (excluding pS22, that is a homodimer), respectively. At this step $\check{\boldsymbol{z}}_3 = (y_1, y_3, y_8, y_{21})$ are substituted as

$y_1 \leftarrow y_1 - y_2$, $y_3 \leftarrow y_3 - y_4$, $y_8 \leftarrow -y_5 - y_6 - y_7 + y_8 - y_{18} - y_{21}$ $y_{21} \leftarrow k_{39} - (y_{17} + y_{18})$. The new variables must obey $y_1 \geq y_2$, $y_3 \geq y_4$, $y_8 \geq y_5 + y_6 + y_7 + y_8 + y_{18} + y_{21}$ and $k_{39} \geq y_{17} + y_{18}$. After this substitution, a fifth timescale occurs of order ϵ^4 (all the approximate conservation laws are slower variables, of timescales ϵ^3 and ϵ^4).

The resulting system satisfies $|D_{\boldsymbol{Z}_k} \bar{\boldsymbol{F}}_k^{(1)}| \neq 0, k \in \{1, 3\}$ and $|D_{\boldsymbol{Z}_4} \bar{\boldsymbol{F}}_4^{(1)}| \neq 0$. We find two approximate conservation laws $y_3 - y_5 - y_6 + y_{17} + y_{18} + y_{19} + y_{20}$, $y_{12} + y_{15}$. In old variables, the first one corresponds to $x_3 + x_4 - x_5 - x_6 + x_{17} + x_{18} + x_{19} + x_{20}$. At this step $\check{\boldsymbol{z}}_4 = (y_3, y_{15})$ are substituted as $y_3 \leftarrow y_3 + y_5 + y_6 - y_{17} - y_{18} - y_{19} - y_{20}$, $y_{15} \leftarrow -y_{12} + y_{15}$. All the new variables are of timescales ϵ^4.

At the next step, $|D_{\boldsymbol{Z}_k} \bar{\boldsymbol{F}}_k^{(1)}| \neq 0, k \in \{1, 4\}$ but $|D_{\boldsymbol{Z}_5} \bar{\boldsymbol{F}}_5^{(1)}| = 0$. We find two new approximate conservation laws $y_3 + y_8$ and $y_3 - y_1$. The first one is an exact conservation as in old variables is $x_3 + x_4 + x_7 + x_8 + x_{18} + x_{19} + x_{20} + x_{17} + x_{18} + x_{21} = k_{39} + k_{41}$. At this step $\check{\boldsymbol{z}}_5 = (y_1, y_3)$ are substituted as $y_1 \leftarrow y_3 - y_1$, $y_3 \leftarrow k_{39} + k_{41} - y_8$. After this step, a sixth timescale occurs, of order ϵ^5 for the variable y_1.

At the last step we identify one more, exact, conservation law $2y_{10} + 2y_9 - y_1$ that in initial variables represents $x_1 + x_2 + x_5 + x_6 + x_7 + x_8 + 2x_9 + 2x_{10} + x_{18} + x_{21} = k_{40} - k_{39} - k_{41}$. The variable y_1 is eliminated and the timescale of order ϵ^5 disappears. The substitution is $y_1 \leftarrow 2y_{10} + 2y_9 - k_{40} + k_{39} + k_{41}$. After this step the full Jacobian matrix is non-singular and there is no more conservation law, approximate or exact. Five timescales remain, of orders ϵ^0,ϵ^1,ϵ^2,ϵ^3,ϵ^4.

To summarize, 6 approximate and 3 exact conservation laws were used in this reduction. The 3 exact conservation laws were used to eliminate 3 of the initial model variables, see Table 7.1. Among the 6 approximate conservation laws, 2 were kept as variables in the final transformed model, the other being eliminated at different steps of the procedure. The final transformed model has a reduced dimensionality (18 variables) and no conservation laws.

The transformed model reads:

$$
\begin{aligned}
\dot{x}_2 &= k_{17}k_{36}x_6 - k_1(x_2 - k_{40} + x_8 + 2x_9 + 2x_{10}) - k_2x_2, \\
\dot{x}_4 &= -k_3(x_4 - k_{41} - k_{39} - x_5 - x_6 + x_8 + x_{17} + x_{18} + x_{19} + x_{20}) - k_3x_4 - \\
&\quad -k_9(k_{39} + x_5 + x_6 + x_7 - x_8 - x_{17}) - k_8x_4x_6, \\
\dot{x}_5 &= k_5x_6 - k_4x_5 + k_7x_7 + 2k_{11}x_9 - 2k_{10}x_5^2 - k_{16}x_{11}(x_2 - k_{40} + x_8 + 2x_9 + 2x_{10}) + \\
&\quad +k_6x_5(x_4 - k_{41} - k_{39} - x_5 - x_6 + x_8 + x_{17} + x_{18} + x_{19} + x_{20}), \\
\dot{x}_6 &= k_4x_5 - k_5x_6 + 2k_{13}x_{10} - 2k_{12}x_6^2 - k_9(k_{39} + x_5 + x_6 + x_7 - x_8 - x_{17}) - k_{35}(x_{17} - k_{39} + x_{18}) - \\
&\quad -k_{17}k_{36}x_6 - k_8x_4x_6, \\
\dot{x}_7 &= -x_7(k_7 + k_{14}) - k_6x_5(x_4 - k_{41} - k_{39} - x_5 - x_6 + x_8 + x_{17} + x_{18} + x_{19} + x_{20}), \\
\dot{x}_8 &= k_7x_7 - x_7(k_7 + k_{14}) + 2k_{11}x_9 + k_{14}x_7 + 2k_{13}x_{10} - 2k_{10}x_5^2 - 2k_{12}x_6^2 - \\
&\quad -k_{16}x_{11}(x_2 - k_{40} + x_8 + 2x_9 + 2x_{10}) - k_{17}k_{36}x_6, \\
\dot{x}_9 &= k_{10}x_5^2 - x_9(k_{11} + k_{15}), \\
\dot{x}_{10} &= k_{15}x_9 - k_{13}x_{10} + k_{12}x_6^2, \\
\dot{x}_{11} &= k_{23}x_{14} - k_{30}x_{11}, \\
\dot{x}_{12} &= k_{18} - x_{12}(k_{20} + k_{26}) + k_{30}x_{11} - k_{27}(x_{12} - x_{15}) - k_{22}k_{37}x_{12}x_{13}, \\
\dot{x}_{13} &= k_{19} - x_{13}(k_{21} + k_{28}) + k_{30}x_{11} + k_{29}x_{16} - k_{22}k_{37}x_{12}x_{13}, \\
\dot{x}_{14} &= k_{22}k_{37}x_{12}x_{13} - x_{14}(k_{23} + k_{24} + k_{25}), \\
\dot{x}_{15} &= k_{18} - x_{12}(k_{20} + k_{26}) + k_{26}x_{12} + k_{30}x_{11} - k_{22}k_{37}x_{12}x_{13}, \\
\dot{x}_{16} &= k_{28}x_{13} - k_{29}x_{16}, \\
\dot{x}_{17} &= k_{31}x_{17}(k_{39} + x_5 + x_6 + x_7 - x_8 - x_{17}) - k_{35}(x_{17} - k_{39} + x_{18}), \\
\dot{x}_{18} &= -k_{34}x_{18} - k_{31}x_{17}(k_{39} + x_5 + x_6 + x_7 - x_8 - x_{17}), \\
\dot{x}_{19} &= k_{34}x_{18} - k_{32}x_{19}, \\
\dot{x}_{20} &= k_{32}x_{19} - k_{33}k_{38}x_{20},
\end{aligned}
\tag{7.32}
$$

and the truncated rescaled transformed model reads

$$
\begin{aligned}
\dot{y}_2 &= \epsilon^1(k_1k_{40} - k_2y_2),\\
\dot{y}_4 &= \epsilon^2(k_3k_{41} - 2k_3y_4),\\
\dot{y}_5 &= \epsilon^1(k_5y_6 + k_7y_7 + k_6y_4y_5 - k_6k_{41}y_5),\\
\dot{y}_6 &= \epsilon^1(k_{35}k_{39} + k_9y_8 + k_9y_{17} - k_9k_{39} - k_5y_6 - k_9y_7 - k_{35}y_{17} - k_{35}y_{18} - k_8y_4y_6),\\
\dot{y}_7 &= \epsilon^2(k_6k_{41}y_5 - k_7y_7 - k_14y_7 - k_6y_4y_5),\\
\dot{y}_8 &= \epsilon^3(k_{16}k_{40}y_{11} - k_{17}k_{36}y_6),\\
\dot{y}_9 &= \epsilon^2(k_{10}y_5^2 - k_{11}y_9 - k_{15}y_9),\\
\dot{y}_{10} &= \epsilon^2(k_{15}y_9 + k_{12}y_6^2 - k_{13}y_{10}),\\
\dot{y}_{11} &= \epsilon^3(k_{23}y_{14} - k_{30}y_{11}),\\
\dot{y}_{12} &= \epsilon^2(k_{27}y_{15} - k_{26}y_{12}),\\
\dot{y}_{13} &= \epsilon^0(k_{19} + k_{30}y_{11} - k_{22}k_{37}y_{12}y_{13}),\\
\dot{y}_{14} &= \epsilon^2(k_{22}k_{37}y_{12}y_{13} - k_{23}y_{14} - k_{25}y_{14}),\\
\dot{y}_{15} &= \epsilon^4(k_{18} + k_{30}y_{11} - k_{20}y_{12} - k_{22}k_{37}y_{12}y_{13}),\\
\dot{y}_{16} &= \epsilon^3(k_{28}y_{13} - k_{29}y_{16}),\\
\dot{y}_{17} &= \epsilon^2(k_{35}k_{39} + k_{31}k_{39}y_{17} + k_{31}y_7y_{17} - k_{35}y_{17} - k_{35}y_{18} - k_{31}y_{17}^2 - k_{31}y_8y_{17}),\\
\dot{y}_{18} &= \epsilon^2(k_{31}y_{17}^2 + k_{31}y_8y_{17} - k_{34}y_{18} - k_{31}k_{39}y_{17} - k_{31}y_7y_{17}),\\
\dot{y}_{19} &= \epsilon^2(k_{34}y_{18} - k_{32}y_{19}),\\
\dot{y}_{20} &= \epsilon^1(k_{32}y_{19} - k_{33}k_{38}y_{20}).
\end{aligned} \tag{7.33}
$$

As can be seen from (7.33), this method unravels one new timescale that was not apparent in the initial rescaled model (7.31).

The new variables of the transformed model can be expressed in the old variables of the initial model as shown in Table 7.2. In order to find to inverse transformation, from new variable x_i to old variables x_i^o, we need to add the three exact conservation laws that were used to eliminate three old variables. More precisely, we have to solve

$$
\begin{aligned}
x_8 &= x_5 + x_6 + x_7 + x_8^o + x_{18} + x_{21}^o\\
x_{15} &= x_{12} + x_{15}^o\\
k_{39} &= x_{17} + x_{18} + x_{21}^o\\
k_{40} &= x_1^o + x_2 + x_5 + x_6 + x_7 + x_8^o + 2x_9 + 2x_{10} + x_{18} + x_{21}^o\\
k_{41} &= x_3^o + x_4 + x_7 + x_8^o + x_{18} + x_{19} + x_{20},
\end{aligned} \tag{7.34}
$$

leading to

$$
\begin{aligned}
x_1^o &= k_{40} - x_2 - x_8 - 2x_9 - 2x_{10}\\
x_3^o &= k_{39} + k_{41} - x_4 + x_5 + x_6 - x_8 - x_{17} - x_{18} - x_{19} - x_{20}\\
x_8^o &= x_8 + x_{17} - x_5 - x_6 - x_7 - k_{39}\\
x_{15}^o &= x_{15} - x_{12}\\
x_{21}^o &= k_{39} - x_{17} - x_{18}.
\end{aligned} \tag{7.35}
$$

Groups of variables	**ToV**
$\boldsymbol{z}_1 = y_{13}$	ϵ^0
$\boldsymbol{z}_2 = (y_2, y_5, y_6, y_{20})$	ϵ^1
$\boldsymbol{z}_3 = (y_1, y_3, y_4, y_7, y_8, y_9, y_{10}, y_{14}, y_{17}, y_{18}, y_{19}, y_{21})$	ϵ^2
$\boldsymbol{z}_4 = (y_{11}, y_{15}, y_{16}, y_3 + y_4, y_5 + y_6 + y_7 + y_8 + y_{18} + y_{21})$	ϵ^3
$\boldsymbol{z}_5 = (y_1 + y_2, y_3 + y_4 - y_5 - y_6 + y_{17} + y_{18} + y_{19} + y_{20}, y_{12} + y_{15})$	ϵ^4
$\boldsymbol{z}_6 = y_3 + y_4 + y_{17} + y_{19} + y_{20} - (y_1 + y2 + y_5 + y_6)$	ϵ^5
Conservation laws	**ToC**
$y_1 + y_2$	ϵ^4
$y_3 + y_4$	ϵ^3
$y_5 + y_6 + y_7 + y_8 + y_{18} + y_{21}$	ϵ^3
$y_{17} + y_{18} + y_{21}$	ϵ^∞
$y_{12} + y_{15}$	ϵ^4
$y_3 + y_4 - y_5 - y_6 + y_{17} + y_{18} + y_{19} + y_{20}$	ϵ^4
$y_3 + y_4 + y_{17} + y_{19} + y_{20} - (y_1 + y2 + y_5 + y_6)$	ϵ^5
$y_3 + y_4 + y_7 + +y_8 + y_{17} + 2y_{18} + y_{19} + y_{20} + y_{21}$	ϵ^∞
$y_1 + y_2 + y_5 + y_6 + 2y_9 + 2y_{10} - (y_3 + y_4 + y_{17} + y_{18} + y_{19} + y_{20})$	ϵ^∞

Table 7.1: Transformed model: various steps and conservation laws. The ToV column contains the timescale order of the variable, i.e. the power of ϵ multiplying the r.h.s. of the rescaled ODEs of the slowest variable included in the conservation laws (the characteristic times scale like the reciprocal of these quantities, so slower means higher order) and the ToC column the same thing for conservation laws. Exact conservations have infinite timescale orders.

Because the old variables are all positive this leads to the following constraints for the new variables:

$$\begin{aligned} k_{40} - x_2 - x_8 - 2x_9 - 2x_{10} &\geq 0 \\ k_{39} + k_{41} - x_4 + x_5 + x_6 - x_8 - x_{17} - x_{18} - x_{19} - x_{20} &\geq 0 \\ x_8 + x_{17} - x_5 - x_6 - x_7 - k_{39} &\geq 0 \\ x_{15} - x_{12} &\geq 0 \\ k_{39} - x_{17} - x_{18} &\geq 0. \end{aligned} \tag{7.36}$$

7.5.3 Reduced models

The non-degeneracy condition being satisfied, the transformed model can be now reduced by successive elimination of the fast variables, obtained nested or slowest timescale reductions. The hyperbolicity condition can be tested with methods exposed in [Kruff et al., 2020]. The reduced models at various last slow timescales are summarized in the Tables 7.3, 7.4, and 7.5.

In order to test numerically the accuracy of the reduction we have eliminated the fast variables up to timescale order ϵ^{q_k} by symbolically solving the algebraic truncated system $\bar{\boldsymbol{F}}_k^{(1)}(\boldsymbol{k}, \boldsymbol{x}) = 0$, eliminating the variables $\boldsymbol{X}_k$, and numerically solving the system of ODEs for the remaining slow variables (nested reduction (7.6)). The result is represented in the

Variable	**Definition**	**Timescale**	**Interpretation**
x_2	x_2	ϵ^1	SMAD2n
x_4	x_4	ϵ^2	SMAD4n
x_5	x_5	ϵ^1	pSMAD2c
x_6	x_6	ϵ^1	pSMAD2n
x_7	x_7	ϵ^2	pSMAD24c
x_8	$x_5 + x_6 + x_7 + x_8 + x_{18} + x_{21}$	ϵ^3	total pSMAD2 without pSMAD22
x_9	x_9	ϵ^2	pSMAD22c
x_{10}	x_{10}	ϵ^2	pSMAD22n
x_{11}	x_{11}	ϵ^3	LRe
x_{12}	x_{12}	ϵ^2	RI
x_{13}	x_{13}	ϵ^0	RII
x_{14}	x_{14}	ϵ^2	LR
x_{15}	$x_{12} + x_{15}$	ϵ^4	total free RI
x_{16}	x_{16}	ϵ^3	RIIe
x_{17}	x_{17}	ϵ^2	TIF
x_{18}	x_{18}	ϵ^2	pSMAD24nTIF
x_{19}	x_{19}	ϵ^2	SMAD4ubn
x_{20}	x_{20}	ϵ^1	SMAD4ubc

Table 7.2: Transformed model: variables and their interpretation. The timescale column contains the power of ϵ multiplying the r.h.s. of the rescaled ODEs (the characteristic times scale like the reciprocal of these quantities, so slower means higher order). The variables of the transformed model are all positive and must also satisfy $k_{40} - x_2 - x_8 - 2x_9 - 2x_{10} \geq 0$, $k_{39} + k_{41} - x_4 + x_5 + x_6 - x_8 - x_{17} - x_{18} - x_{19} - x_{20} \geq 0$, $x_8 + x_{17} - x_5 - x_6 - x_7 - k_{39} \geq 0$, $x_{15} - x_{12} \geq 0$, $k_{39} - x_{17} - x_{18} \geq 0$.

ODEs	
$\dot{y}_4$	$= -k_3(y_4 - k_{41} - k_{39} - y_5 - y_6 + y_8 + y_{17} + y_{18} + y_{19} + y_{20})$
	$-k_3 y_4 - k_9(k_{39} + y_5 + y_6 + y_7 - y_8 - y_{17}) - k_8 y_4 y_6,$
$\dot{y}_7$	$= -y_7(k_7 + k_{14}) - k_6 y_5(y_4 - k_{41} - k_{39} - y_5 - y_6 + y_8 + y_{17} + y_{18} + y_{19} + y_{20}),$
$\dot{y}_8$	$= 2(k_{11} y_9 + k_{13} y_{10} - k_{10} y_5^2 - k_{12} y_6^2) - k_{16} y_{11}(y_2 - k_{40} + y_8 + 2y_9 + 2y_{10}) - k_{17} k_{36} y_6,$
$\dot{y}_9$	$= k_{10} y_5^2 - y_9(k_{11} + k_{15}),$
$\dot{y}_{10}$	$= k_{15} y_9 - k_{13} y_{10} + k_{12} y_6^2,$
$\dot{y}_{11}$	$= k_{23} y_{14} - k_{30} y_{11},$
$\dot{y}_{12}$	$= k_{18} - y_{12}(k_{20} + k_{26}) + k_{30} y_{11} - k_{27}(y_{12} - y_{15}) - k_{22} k_{37} y_{12} y_{13},$
$\dot{y}_{14}$	$= k_{22} k_{37} y_{12} y_{13} - y_{14}(k_{23} + k_{24} + k_{25}),$
$\dot{y}_{15}$	$= k_{18} - y_{12}(k_{20} + k_{26}) + k_{26} y_{12} + k_{30} y_{11} - k_{22} k_{37} y_{12} y_{13},$
$\dot{y}_{16}$	$= k_{28} y_{13} - k_{29} y_{16},$
$\dot{y}_{17}$	$= k_{31} y_{17}(k_{39} + y_5 + y_6 + y_7 - y_8 - y_{17}) - k_{35}(y_{17} - k_{39} + y_{18}),$
$\dot{y}_{18}$	$= -k_{34} y_{18} - k_{31} y_{17}(k_{39} + y_5 + y_6 + y_7 - y_8 - y_{17}),$
$\dot{y}_{19}$	$= k_{34} y_{18} - k_{32} y_{19}.$
Fast variables	
y_2	$= \frac{k_1 k_{40}}{k_2},$
y_5	$= \frac{k_5(k_{35}-k_9)k_{39}+k_5(k_7-k_9)y_7+k_5 k_9(y_8+y_{17})-k_5 k_{35}(y_{17}+y_{18})+k_7 k_8 y_4 y_7}{k_6(k_5+k_8 y_4)(k_{41}-y_4)},$
y_6	$= -\frac{(k_9-k_{35})k_{39}+k_9(y_7-y_8)+(k_{35}-k_9)y_{17}+k_{35} y_{18}}{k5+k8y4},$
y_{13}	$= \frac{k_{19}+k_{30} y_{11}}{k_{22} k_{37} y_{12}},$
y_{20}	$= \frac{k_{32} y_{19}}{k_{33} k_{38}}.$

Table 7.3: Description of a first reduced model. The timescale order of slow variables satisfying ODEs are at least ϵ^2. The fast variable column expresses the concentrations of fast variables as functions of the slow ones.

ODEs	
$\dot{y}_8$	$= -2k_{10}y_5^2 - 2k_{12}y_6^2 - k_{16}y_{11}(y_2 - k_{40} + y_8 + 2y_9 + 2y_{10}) + 2k_{13}y_{10} + 2k_{11}y_9 - k_{17}k_{36}y_6,$
$\dot{y}_{11}$	$= k_{23}y_{14} - k_{30}y_{11},$
$\dot{y}_{15}$	$= k_{18} - y_{12}(k_{20} + k_{26}) + k_{26}y_{12} + k_{30}y_{11} - k_{22}k_{37}y_{12}y_{13},$
$\dot{y}_{16}$	$= k_{28}y_{13} - k_{29}y_{16}.$
Fast variables	
y_2	$= \frac{k_1k_{40}}{k_2},$
y_4	$= \frac{k_{41}}{2},$
y_{17}	$= (-b + \sqrt{b^2 - 4ac})/(2a),$
y_5	$= \frac{4k_5(k_9+k_{31}y_{17})(k_7+k_{14})(y_8-k_{39}+y_{17})}{k_6k_{41}(2k_5k_9+2k_5k_{14}+k_8k_{14}k_{41}+2k_5k_{31}y_{17})},$
y_6	$= \frac{2k_{14}(k_9+k_{31}y_{17})(y_8-k_{39}+y_{17})}{2k_5k_9+2k_5k_{14}+k_8k_{14}k_{41}+2k_5k_{31}y_{17}},$
y_7	$= \frac{2k_5(k_9+k_{31}y_{17})(y_8-k_{39}+y_{17})}{2k_5k_9+2k_5k_{14}+k_8k_{14}k_{41}+2k_5k_{31}y_{17}},$
y_9	$= \frac{16k_5^2k_{10}(k_9+k_{31}y_{17})^2(k_7+k_{14})^2(y_8-k_{39}+y_{17})^2}{k_6^2k_{41}^2(k_{11}+k_{15})(2k_5k_9+2k_5k_{14}+k_8k_{14}k_{41}+2k_5k_{31}y_{17})^2},$
y_{18}	$= \frac{k_{35}k_{39}+(k_{31}k_{39}-k_{35})y_{17}+k_{31}(y_7-y_8)y_{17}-k_{31}y_{17}^2}{k_{35}},$
y_{19}	$= \frac{k_{34}y_{18}}{k_{32}},$
y_{20}	$= \frac{k_{34}y_{18}}{k_{33}k_{38}}.$
c	$= -k_{34}k_{35}k_{39}(2k_5k_9 + 2k_5k_{14} + k_8k_{14}k_{41}),$
a	$= k_{31}(2k_5k_{14}k_{34} + 2k_5k_{14}k_{35} + 2k_5k_{34}k_{35} + k_8k_{14}k_{34}k_{41} + k_8k_{14}k_{35}k_{41}),$
b	$= (a - 2k_5k_{31}k_{34}k_{35})y_8 + (c - a)k_{39} - c(k_{39}^2 + 1)/k_{39}.$

Table 7.4: Description of a second reduced model. The timescale order ofslow variables satisfying ODEs are at least ϵ^3. The fast variable column expresses the concentrations of fast variables as functions of the slow ones.

ODEs	
$\dot{y}_{15}$	$= k_{18} - y_{12}(k_{20} + k_{26}) + k_{26}y_{12} + k_{30}y_{11} - k_{22}k_{37}y_{12}y_{13}$
Fast variables	
y_2	$= \frac{k_1 k_{40}}{k_2}$,
y_4	$= \frac{k_{41}}{2}$,
y_5	$= \frac{2k_5k_{16}k_{19}k_{23}k_{40}(k_7+k_{14})}{k_6k_{14}k_{17}k_{25}k_{30}k_{36}k_{41}}$,
y_6	$= \frac{k_{16}k_{19}k_{23}k_{40}}{k_{17}k_{25}k_{30}k_{36}}$,
y_7	$= \frac{k_5k_{16}k_{19}k_{23}k_{40}}{k_{14}k_{17}k_{25}k_{30}k_{36}}$,
y_8	$= \frac{(2k_5(k_9+k_{14})+k_8k_{14}k_{41})k_{16}k_{19}k_{23}k_{40}+k_{14}k_{17}k_{25}k_{30}k_{36}(2(k_9-k_{35})(k_{39}-y_{17})+2k_{35}y_{18})}{2k_9k_{14}k_{17}k_{25}k_{30}k_{36}}$,
y_{18}	$= \frac{-k_{34}((2k_5+k_8k_{41})k_{16}k_{19}k_{23}k_{31}k_{40}+2(k_9-k_{31}k_{39})k_{17}k_{25}k_{30}k_{35}k_{36})y_{17}+ay_{17}^2+c}{k_{34}(2k_{17}k_{25}k_{30}k_{35}k_{36}(k_9+k_{31}y_{17}))}$,
y_{19}	$= \frac{k_{34}y_{18}}{k_{32}}$,
y_{20}	$= \frac{k_{34}y_{18}}{k_{33}k_{38}}$,
y_9	$= \frac{4k_5^2k_{10}k_{16}^2k_{19}^2k_{23}^2k_{40}^2(k_7+k_{14})^2}{k_6^2k_{14}^2k_{17}^2k_{25}^2k_{30}^2k_{36}^2k_{41}^2(k_{11}+k_{15})}$,
y_{13}	$= \frac{k_{19}(k_{23}+k_{25})k_{26}}{k_{22}k_{25}k_{27}k_{37}y_{15}}$,
y_{14}	$= \frac{k_{19}}{k_{25}}$,
y_{16}	$= \frac{k_{19}k_{26}k_{28}(k_{23}+k_{25})}{k_{22}k_{25}k_{27}k_{29}k_{37}y_{15}}$
y_{10}	$= \frac{k_{16}^2k_{19}^2k_{23}^2k_{40}^2(4k_5^2k_{15}k_{10}(k_7+k_{14})^2+k_6^2k_{12}k_{14}^2(k_{15}+k_{11})k_{41}^2)}{k_6^2k_{13}k_{14}^2k_{17}^2k_{25}^2k_{30}^2k_{36}^2k_{41}^2(k_{11}+k_{15})}$,
y_{11}	$= \frac{k_{19}k_{23}}{k_{25}k_{30}}$,
y_{12}	$= \frac{k_{27}y_{15}}{k_{26}}$.
a	$= 2k_{17}k_{25}k_{30}k_{31}k_{34}k_{35}k_{36}$,
c	$= -2k_9k_{17}k_{25}k_{30}k_{34}k_{35}k_{36}k_{39}$,
b	$= k_5k_{16}k_{19}k_{23}k_{31}(2+k_8k_{41})(k_{34}+k_{35})k_{40} + k_{39}(c-a)$, $y_{17} = (-b+\sqrt{b^2-4ac})/(2a)$,

Table 7.5: Description of the last reduced model. The timescale order of the slow variable satisfying ODEs is ϵ^4. The fast variable column expresses the concentrations of fast variables as functions of the slow ones.

Figure 7.1, for various choices of k. As it can be noticed, especially at shorter timescales there are few species that are predicted with errors by the reduced model. There are two reasons to this phenomenon. The first reason is that the values of fast species are based on the truncated system of equations. Although all the terms neglected by truncation have orders larger than the dominant terms and therefore the reduction is justified in the limit $\epsilon \to 0$, for finite ϵ the quality of the approximation can be low if the number of the neglected terms is large. This source of error can be reduced by considering higher order terms in the approximation, for instance higher order Puiseux series to represent the fast variables. Another reason for bad approximation is the choice of the tropical equilibration used for the reduction. A tropical equilibration solution is valid in a domain in the space of concentrations but not for all species concentrations. Furthermore, several tropical equilibration solutions (a polytope in log scale) lead to the same reduced model, but again the corresponding polytope does not cover all the concentration. It is thus possible that the tropical equilibration solution and the reduced model has to change along a trajectory of the full model when this crosses polytopes corresponding to different reductions. This is the case for the transformed TGFβ model, see Figure 7.2. This source of error can be reduced by considering tropical equilibration solutions at the boundary between polytopes, leading to reductions valid for two or several polytopes of solutions.

7.6 Conclusions

We have provided an algorithmic solution allowing to transform systems of polynomial ODEs with approximate conservation laws into equivalent systems that do not have approximate conservation laws. This allowed us to reduce the transformed systems by using geometric singular perturbation theory.

Our reduction algorithms are based on calculation of complete sets of linear, monomial or polynomial conservation laws.

Linear conservation laws are easier to compute and one should always start by looking for complete sets of linear conservation laws. We have seen that important CRN case studies can be reduced down to the slowest timescale by using only linear conservation laws. The large propensity for linear conservation laws in biochemical models coming from real life may be explained by the modularity of these networks: pools of species form cycles whose internal dynamics have well defined timescales and these cycles are nested hierarchically. Interestingly, the introduction of new variables corresponding to the linear conservation laws correspond to pooling the species taking part in a cycle into a single variable, whereas fast species elimination corresponds to pooling several reactions and pruning the internal variables [Radulescu et al., 2012]. It is thus possible, in certain cases, to accompany the reduction of the ODE system by reduction of the graph of chemical reactions using graph rewriting, in order to obtain a CRN structure for the reduced model. We will discuss this possibility in detail elsewhere.

The model rescaling and reduction is based on the choice of a tropical equilibration solution. Changing this solution usually leads to a change of the reduction, which is not surprising because a reduction have limited validity in concentration space and in time. However, some reductions are better than other in the sense that they have wider validity in the concentration space and in time. It would be therefore useful to define extra geometric criteria that lead to effective reduction. For instance, some tropical equilibration solutions

may lead to approximate conservation laws that are not equilibrated (there are only positive or only negative dominant terms in the r.h.s. of the corresponding ODEs). Finding tropical equilibration solutions that lead to equilibrated approximate conservation laws is an interesting problem, although this may be very hard algorithmically.

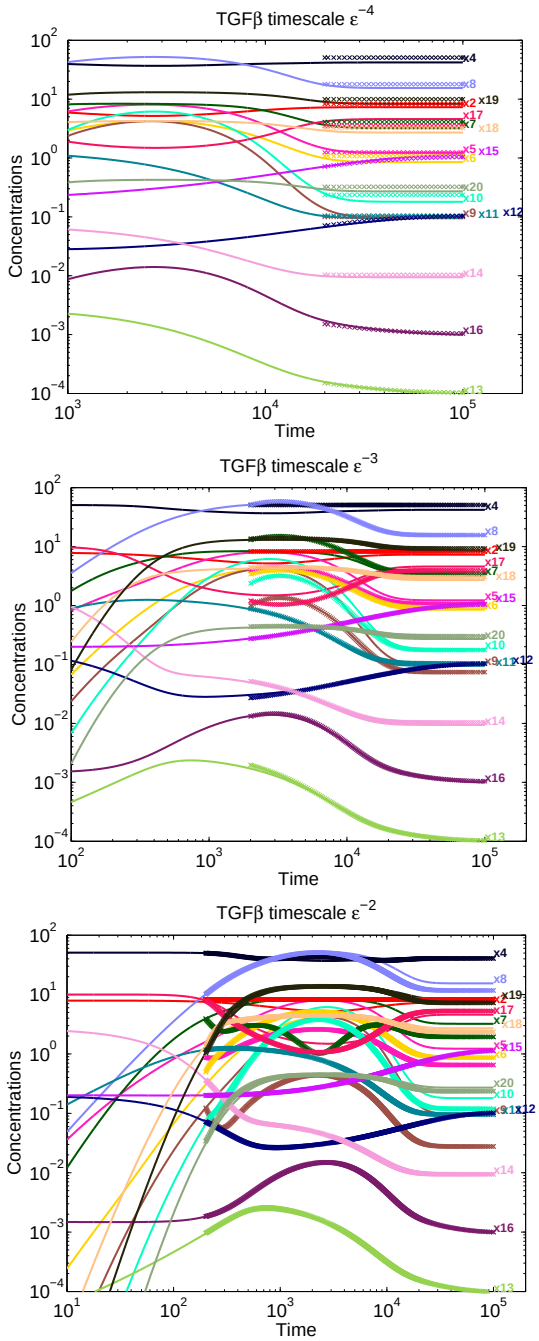


Figure 7.1: Comparison of numerical solutions obtained with the transformed TGFβ model (continuous lines) and with slowest timescale reduced models (crosses). For each reduced model, a small number of variables (slow) follow ODEs. The initial values of these were chosen the same as the values computed with the full transformed model at a large enough time. The remaining fast variables were computed as functions of the slow variables. The large errors for a few species at times shorther than 10^4 could be explained by lack of validity of the tropical equilibration used for the reduction at these shorter timescales, see also Figure 7.2.

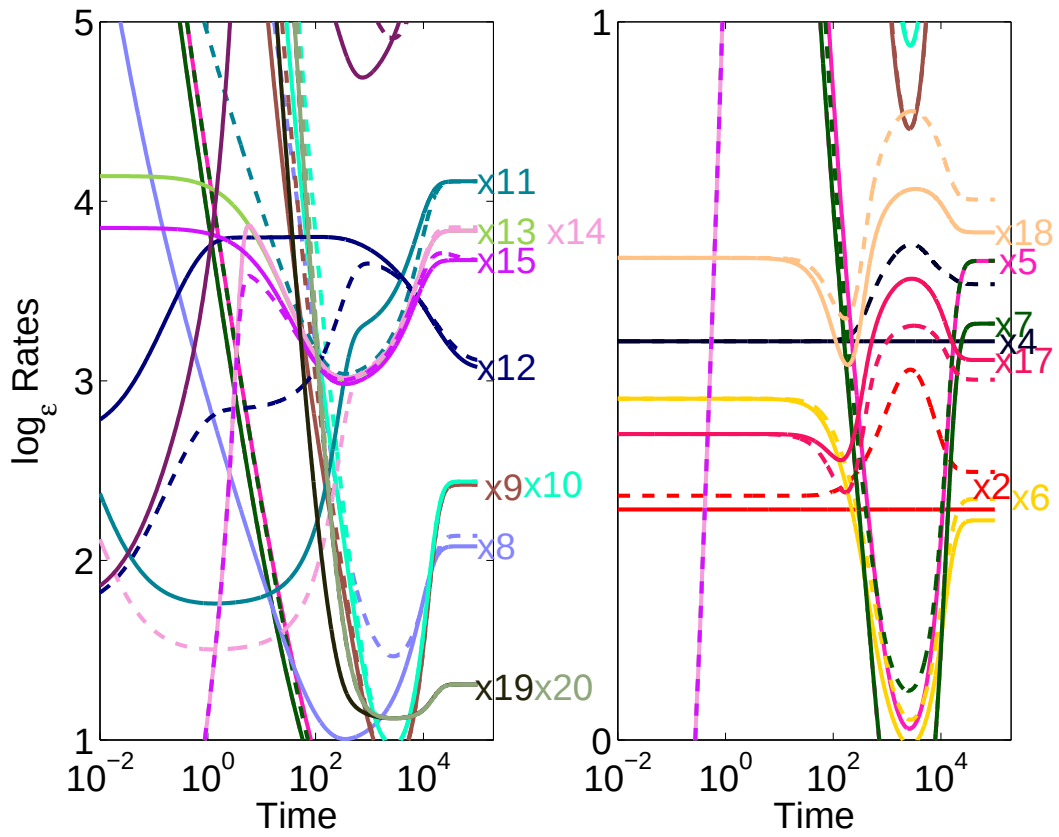


Figure 7.2: Testing tropical equilibration for various species of the transformed TGFβ model. For each species we have plotted the $\log_\epsilon$ of positive (continuous line) and negative (dotted lines) rates producing and consuming these species, respectively. For tropically equilibrated species the two rates must have the same order (the $\log_\epsilon$ values should round up to the same integer for continuous and dotted curves of the same color). For times larger than 10^4 this condition is valid for all species. For shorter times, a few species are not equilibrated. Furthermore, some rates change abruptly at these timescales, suggesting that different tropical equilibration solutions should be considered at shorter timescales.

Chapter 8

Computing Conservation laws

The model reduction algorithm presented in the previous chapter uses complete exact and approximate conservation laws. In this chapter we present algorithmic methods for computing these conservation laws.

8.1 Computing Linear Conservation Laws

Let us consider again model (2.2). The truncated vector field of this model has been obtained by a formal scaling procedure as described in Section 5.3. The right hand side of the truncated system is of the form

$$\boldsymbol{F}^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{x}, \delta) = \left(\delta^{b_1} \bar{f}_1^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{x}), \ldots, \delta^{b_n} \bar{f}_n^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{x})\right)^{\mathsf{T}},$$

where

$$\bar{f}_i^{(1)}(\bar{\boldsymbol{k}}, \boldsymbol{x}) = \sum_{j=1}^{r} S_{ij}^{(1)} \bar{k}_j \boldsymbol{x}^{\boldsymbol{\alpha}_j}.$$

The matrix $\boldsymbol{S}^{(1)} = (S_{ij}^{(1)}) \in \mathbb{Z}^{n \times r}$ is called **truncated stoichiometric matrix**. Since the truncated system contains less monomials than the full system, $\boldsymbol{S}^{(1)}$ is obtained from $\boldsymbol{S}$ by setting the respective entries in $\boldsymbol{S}$ to zero. The matrix $\boldsymbol{S}^{(1)}$ can be also computed by applying Algorithm 2 to the truncated vector field.

A vector $\boldsymbol{c} = (c_1, \ldots, c_n) \in \mathbb{R}^n$ defines an exact linear conservation law $\Phi(\boldsymbol{x}) = \sum_{i=1}^{n} c_i x_i$ of model (2.2), if

$$\sum_{i=1}^{n} c_i f_i(\boldsymbol{k}, \boldsymbol{x}) = \sum_{j=1}^{r} (\boldsymbol{c}\boldsymbol{S})_j k_j \boldsymbol{x}^{\boldsymbol{\alpha}_j} = 0,$$

for all $\boldsymbol{k}$, $\boldsymbol{x}$. Exact linear conservation laws of (2.2) can be obtained from the matrix $\boldsymbol{S}$ in the following way.

Theorem 8.1. *Assume that all monomial reaction rates with the same multi-index have different rate constants, i.e.* $\boldsymbol{\alpha}_j = \boldsymbol{\alpha}_{j'}$ *with* $j \neq j'$ *implies* $k_j \neq k_{j'}$. *Let* s *be the rank of the matrix* $\boldsymbol{S}$. *Then there is a full rank matrix* $\boldsymbol{C}$ *with* $n - s$ *rows such that* $\boldsymbol{C}\boldsymbol{S} = 0$.

Further $\boldsymbol{C}$ can be chosen such that its rows form a set of independent, irreducible exact linear conservation laws unconditional on $\boldsymbol{k}$. Furthermore, all $\boldsymbol{c} = (c_1, \ldots, c_n)$ defining exact linear conservation laws unconditional on $\boldsymbol{k}$ are linear combinations of the rows of $\boldsymbol{C}$.

Proof. As a direct consequence of the rank theorem, there is a matrix $\boldsymbol{C}$ with $n - s$ independent rows such that $\boldsymbol{CS} = 0$. If $\boldsymbol{c} = (c_1, \ldots, c_n)$ defines an exact linear conservation law unconditional on $\boldsymbol{k}$ it follows that $\sum_{j=1}^{r} (\boldsymbol{cS})_j k_j \boldsymbol{x}^{\boldsymbol{\alpha}_j} = 0$ for all $\boldsymbol{k}$, $\boldsymbol{x}$. If $i \neq j$ one has $\boldsymbol{\alpha}_i \neq \boldsymbol{\alpha}_j$ or $k_i \neq k_j$. It follows that $k_j \boldsymbol{x}^{\boldsymbol{\alpha}_j}$ are linearly independent in $\mathbb{R}[\boldsymbol{k}, \boldsymbol{x}]$ and $\boldsymbol{cS} = 0$. Therefore, $\boldsymbol{c}$ is a linear combination of rows of $\boldsymbol{C}$. Those rows of $\boldsymbol{C}$ that are not irreducible can be decomposed as a sum of irreducible linear conservation laws. The resulting conservation laws form a generating set for the vector space of linear conservation laws and therefore this set contains a basis. The $n - s$ elements of this basis can be chosen as the irreducible rows of $\boldsymbol{C}$. □

Remark 8.2. Because $S_{ij} \in \mathbb{Z}$, the coefficients c_i can be chosen to be integers. In most practical applications, c_i are small integers and can be considered of order $O(\delta^0)$. This property will be used in Section 5.3. For conservative CRNs linear conservation laws can be chosen semi-positive, that is all $c_i \geq 0$ (see [Schuster and Höfer, 1991,Soliman, 2012,Lemaire and Temperville, 2014]). Semi-positive linear conservation laws are important because their existence implies that concentrations of some species are bounded. Algorithms to compute irreducible, semi-positive linear conservation laws can be found in [Schuster and Höfer, 1991, Soliman, 2012,Lemaire and Temperville, 2014].

Remark 8.3. In Example 6.12 we have seen that $n - s$ independent, linear conservation laws may not form a complete system. The system in this example is of the type (2.2), although not a mass action network.

Remark 8.4. Approximate conservation laws are computed in the same way as the exact ones after replacing the matrix $\boldsymbol{S}$ by the truncated stoichiometric matrix $\boldsymbol{S}^{(1)}$.

Theorem 8.5. *Consider the Jacobian matrix $\boldsymbol{J}(\boldsymbol{k}, \boldsymbol{x}) = D_{\boldsymbol{x}} \boldsymbol{F}(\boldsymbol{k}, \boldsymbol{x})$ and the matrix $\boldsymbol{C}$ introduced in Theorem 8.1. The rows of $\boldsymbol{C}$ provide a complete set of linear conservation laws, if for all $\boldsymbol{k} \in \mathbb{R}_+^r$,$\boldsymbol{x} \in \mathbb{R}_+^n$ satisfying $\boldsymbol{F}(\boldsymbol{k}, \boldsymbol{x}) = \boldsymbol{0}$ we have*

i) *$rk(\boldsymbol{J}(\boldsymbol{k}, \boldsymbol{x})) = rk(\boldsymbol{S}) = s$, and*

ii) *none of the rows of the product $J(\boldsymbol{k}, \boldsymbol{x})\boldsymbol{S}$ can be zero.*

Proof. We check that the assumptions imply the conditions in Definition 6.6. From i) it follows that $\boldsymbol{J}(\boldsymbol{k}, \boldsymbol{x})$ has s independent rows. From ii) we obtain that these rows are not in the left kernel of $\boldsymbol{S}$ which is spanned by the $n - s$ rows of $\boldsymbol{C}$. Thus the matrix $\begin{pmatrix} \boldsymbol{J}(\boldsymbol{k}, \boldsymbol{x}) \\ \boldsymbol{C} \end{pmatrix}$ has rank n. □

The CRN in Example 6.12 has no complete set of linear conservation laws. This model fails to satisfy condition ii) of the theorem, but it fulfills i). The network is not a mass action CRN, so one may ask whether mass action CRNs automatically satisfy both i) and ii). The answer to this question is no, as it is shown by the following example.

Example 8.6. The CRN

$$\dot{x}_1 = -k_1x_1 - k_2x_1 - 2k_3x_1^2, \quad \dot{x}_2 = 2k_3x_1^2 + 2k_2x_1, \quad \dot{x}_3 = k_3x_1^2 + k_1x_1$$

is a mass action network described by the reactions

$$A_1 \xrightarrow{k_1} A_3, \quad A_1 \xrightarrow{k_2} 2A_2, \quad 2A_1 \xrightarrow{k_3} 2A_2 + A_3.$$

Its stoichiometric matrix is $\boldsymbol{S} = \begin{pmatrix} -1 & -1 & -2 \\ 0 & 2 & 2 \\ 1 & 0 & 1 \end{pmatrix}$ and one easily checks that it has rank two. There is a single linear conservation law, namely $\phi(\boldsymbol{x}) = 2x_1 + x_2 + 2x_3$. The Jacobian matrix of the system computes as

$$J(\boldsymbol{k}, \boldsymbol{x}) = \begin{pmatrix} -k_1 - k_2 - 4k_3x_1 & 0 & 0 \\ 2k_2 + 4k_3x_1 & 0 & 0 \\ k_1 + 2k_3x_1 & 0 & 0 \end{pmatrix}$$

and clearly has rank one. Therefore this model does not satisfy condition i) of the Theorem 8.5, whereas ii) is fulfilled. Furthermore the system of equations $\boldsymbol{F}(\boldsymbol{k}, \boldsymbol{x}) = 0$ and $\phi(\boldsymbol{x}) = c_0$ defining the intersection of $\mathcal{S}_{\boldsymbol{k}}$ with the stoichiometric compatibility class has degenerate solutions $x_1 = 0$, $x_2 = c_0 - 2x_3$, which means that the conservation law is not complete.

In [Feliu and Wiuf, 2012] sufficient conditions were found for mass action CRNs to be "injective", meaning to have a unique steady state in any stoichiometric compatibility class. The same conditions also imply completeness of any $n - s$ independent, linear conservation laws (also see [Feliu and Wiuf, 2012]).

8.2 Computing monomial conservation laws

Proposition 8.7. *Let E be a system of ODE given by:*

$$\dot{x}_i = f_i(\boldsymbol{k}, \boldsymbol{x}).$$

Let E' the system of ODE given by:

$$\dot{x}_i = \frac{f_i(\boldsymbol{k}, \boldsymbol{x})}{x_i}.$$

E admits a monomial conservation law if and only if E' admits a linear conservation law with integer coefficients. Moreover, if the linear conservation laws for E' reads $\sum_{m_i \neq 0} m_i x_i$ then the monomial conservation law for E reads $\prod_{m_i \neq 0} x_i^{m_i}$ ($m_i \in \mathbb{R}_+^*$). The linear conservation laws of E' are irreducible if and only if the corresponding monomial conservation laws of E are irreducible.

Remark 8.8. The system E can be any system of ODE, in particular it works for a system E of the form (2.2).

Proof. Let $M = \prod_{i=1}^{n} x_i^{m_i}$ be a monomial conservation law for E, then the derivative of M reads:

$$\dot{M} = (\sum_{i=1}^{n} m_i \frac{\dot{x_i}}{x_i}) M.$$

Since M is a conservation law, we can suppose that $\forall i \in \{1, ..., n\}$, such that $m_i \neq 0$, one has $x_i \neq 0$; this reduces to considering only initial conditions such that $M \neq 0$. Then:

$$\dot{M} = 0 \Longleftrightarrow \sum_{i=1}^{n} m_i \frac{\dot{x_i}}{x_i} = 0 \Longleftrightarrow \sum_{i=1}^{n} m_i \frac{f_i(\boldsymbol{k}, \boldsymbol{x})}{x_i} = 0$$

Let $\phi = \sum_{i=1}^{n} m_i x_i$ be the linear conservation corresponding to M. If M is reducible then $M = M_1 M_2$ with $\dot{M}_1 = 0$ and $\dot{M}_2 = 0$. Write $M_k = \prod_{i=1}^{n} x_i^{m_{k,i}}$, then $\dot{M}_k = 0 \Longleftrightarrow \sum_{i=1}^{n} m_{k,i} \frac{\dot{x_i}}{x_i} = 0$, for $k \in \{1, 2\}$. Thus, $\phi = \phi_1 + \phi_2$, where $\phi_k = \sum_{i=1}^{n} m_{k,i} x_i$ are linear conservation laws of E', for $k \in \{1, 2\}$. This means that the linear conservation for E' is also reducible. The reverse follows the same reasoning.

□

Let us consider that E is a rational system of ODE of the form

$$\dot{x_i} = x_i \Big(\sum_{j=1}^{m} B_{ij} \boldsymbol{x}^{\boldsymbol{H}_j} \Big) \tag{8.1}$$

or in vector form

$$\dot{\boldsymbol{x}} = \boldsymbol{x} \circ \Big(\sum_{j=1}^{m} \boldsymbol{B}_j \circ \boldsymbol{x}^{\boldsymbol{H}_j} \Big) \tag{8.2}$$

with $\boldsymbol{B}_j = (b_{1j}, ..., b_{nj})^\top \in \mathbb{R}^n$, $\boldsymbol{H}_j = (h_{j1}, ..., h_{jn}) \in \mathbb{Z}^n$, m denotes the number of distinct monomials $\boldsymbol{x}^{H_j}$, and $\circ$ denotes the Hadamard, element-wise product.

In this case E' reads

$$\dot{\boldsymbol{x}} = \sum_{j=1}^{m} \boldsymbol{B}_j \boldsymbol{x}^{\boldsymbol{H}_j}, \tag{8.3}$$

According to the Algorithms 2 and Algorithm 1, (8.3) can be written under the form (2.2). It can be used to compute linear conservation laws for E', independent of $\boldsymbol{k}$, and consequently, monomial conservation laws for E, independent of $\boldsymbol{k}$. Indeed, if the stoichiometric matrix $\boldsymbol{S}$ of E' have rank $s = rk(\boldsymbol{S})$, which is the number of independent columns of $\boldsymbol{S}$, then the Theorem (8.1) implies that there are $n - s$ independent linear conservation laws of (8.3), independant of $\boldsymbol{k}$, and consequently $n - s$ independent monomial conservation laws of (8.2), independent of $\boldsymbol{k}$. These linear conservation laws are the lines of a matrix $\boldsymbol{C}$ that satisfies $\boldsymbol{CS} = \boldsymbol{0}$. Thus, irreducible, independent monomial conservation laws of (8.2), independent of $\boldsymbol{k}$, can be computed using the corresponding algorithms for linear conservation laws. This procedure is summarized in the Algorithm 22.

However, existing examples of monomial conservation laws in the literature may depend on $\boldsymbol{k}$ [Mahdi et al., 2017]. To find them, it exists another method based on the following theorem [Goldman, 1987]:

Algorithm `StoechiometricMonomialConservation`*(system E as in equation (2.2))*
  for *i=1* **to** n **do**
    $\dot{x}_i := \frac{\dot{x}_i}{x_i}$ /* transforms E to E' */
  $E' :=$CompareAndSplitCoefficients($\dot{\boldsymbol{x}}$)
  $\boldsymbol{S'} :=$ Smatrix(E')
  return LeftKernel($\boldsymbol{S'}$)

Algorithm 22: Given a system of EDO as in (2.2), find monomial conservation laws of this system using the stoichiometric matrix of E'. This allows to find monomial conservation laws independent of the parameters.

Theorem 8.9 (Goldman, 1985)**.** *Let E as in (8.2), and let W be the subspace of $\mathbb{R}^n$ generated by $\boldsymbol{B}_1, ..., \boldsymbol{B}_m$. Then $\boldsymbol{x}^{\boldsymbol{\phi}}$ is a conservation law for E if and only if $\boldsymbol{\phi}$ is normal to W.*

Example 8.10. Let the E be the Volpert model:

$$A_1 + A_2 \xrightarrow{k_1} 2A_2, \quad A_2 + A_3 \xrightarrow{k_2} 2A_3, \quad A_3 + A_1 \xrightarrow{k_3} 2A_1$$

and the ODEs

$$\dot{x}_1 = x_1(k_2x_3 - k_1x_2), \quad \dot{x}_2 = x_2(k_1x_1 - k_2x_3), \quad \dot{x}_3 = x_3(k_2x_2 - k_3x_1).$$

In this example, the stoechiometric matrix for the system E' is

$$\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix}$$

which is of full rank. However, the subspace generated by the $\boldsymbol{B}_j$ is:

$$\begin{pmatrix} 0 & -k_1 & k_3 \\ k_1 & 0 & -k_2 \\ -k_3 & k_2 & 0 \end{pmatrix}$$

which is of rank 2. The left-kernel of this matrix is given by the vector (k_2, k_3, k_1). This means that $x_1^{k_2}x_2^{k_3}x_3^{k_1}$ is a monomial conservation law for E, depending on $\boldsymbol{k}$, but existing for all $\boldsymbol{k}$.

Algorithm `MonomialConservation`*(system E as in equation (8.2))*
  $\boldsymbol{B} := (\boldsymbol{B_1}...\boldsymbol{B_m})$
  return LeftKernel($\boldsymbol{B}$)

Algorithm 23: Given a system of EDO as in (8.2), find monomial conservation laws depending or not of the parameters.

Remark 8.11. One can see that algorithms 22 and 23 are very similar, the differences is that the stoechiometric matrix $\boldsymbol{S}$ depends on distincts monomials $k_j\boldsymbol{x}^{\boldsymbol{\alpha}_j}$ of E' and have coefficients in $\mathbb{Z}$, when the matrix $\boldsymbol{B}$ depends on distincts monomials $\boldsymbol{x}^{\boldsymbol{H}_j}$ and have coefficient in $\mathbb{R}$. These methods are equivalent in the case when in equation (2.2), $\boldsymbol{x}^{\boldsymbol{\alpha}_j} = \boldsymbol{x}^{\boldsymbol{\alpha}_h}$ implies $k_j = k_h$.

We can use the Theorem 8.9 to gain more insight into the type of networks that have monomial conservation laws. To this end, we prove the following:

Proposition 8.12. *Let E is a polynomial system of ODE given by:*

$$\dot{x_i} = P_i(\boldsymbol{k}, \boldsymbol{x}),\ (P_i \in \mathbb{R}[\boldsymbol{x}]).$$

E can have a monomial conservation law only if some of these ODE can be written as a product of the variable considered and a polynomial, ie. if

$$\exists A \subset \{1, ..., n\},\ A \neq \emptyset\ s.t.\ \forall j \in A,\ \dot{x_j} = x_j Q_j(\boldsymbol{k}, \boldsymbol{x}),\ (Q_j \in \mathbb{R}[\boldsymbol{x}]).$$

Or, in another terms, if it exist equations of E' that are polynomials.

Moreover, if E is given by a chemical reaction network induced by mass action law, then E can have a monomial conservation law only if:

- *It exists a partition $A \sqcup B$ of the set of species $\{x_1, ..., x_n\}$.*
- *For each reaction involving only species in A, the reaction is of the form:*

$$\sum \underbrace{k_{i,j}}_{\neq 0} x_i \longrightarrow \sum (k_{i,j} \pm l) x_i.$$

- *For each reaction involving species in A and species in B, the product belongs to B, meaning that species in A can only be reactants in these reactions.*

Proof. Suppose that E is a polynomial system of ODE, and write it as in equation (8.2):

$$\dot{\boldsymbol{x}} = \boldsymbol{x} \circ \Big(\sum_{j=1}^{m} \boldsymbol{B}_j \circ \boldsymbol{x}^{\boldsymbol{H}_j} \Big)$$

where $\boldsymbol{H}_j = (h_{j1}, ..., h_{jn}) \in \mathbb{Z}^n$, $\boldsymbol{x}^{\boldsymbol{H}_j} = x_1^{H_{j1}} ... x_n^{H_{jn}}$, $\boldsymbol{B}_j = (B_{1j}, ..., B_{nj})$, and m denote the number of distinct monomials $\boldsymbol{x}^{\boldsymbol{H}_j}$.

Since E is polynomial, we have for all i, j, $h_{ji} \in \mathbb{N} \cup \{-1\}$. Moreover, $h_{ji} = -1$ only if the monomial $x^{H_j} = x_1^{h_{j1}} ... x_i^{-1} ... x_n^{h_{jn}}$ and is involved in $\dot{x_i}$. In such case, the vector $\boldsymbol{B}_j$ associated to H_j have a non zero term B_{ij}, and, since E is polynomial, the monomial H_j is involved only in $\dot{x_i}$, so $\boldsymbol{B}_j$ is a one-component vector, non-zero on row i. If for all $i \in \{1, ..., n\}$ it exists a such $h_{ji} = -1$ (meaning that $\dot{x_i}$ is not factorisable by x_i), then it exists n one-component vectors $\boldsymbol{B}_j^{(i)}$ that generate the full space $\mathbb{R}^n$. Then, no normal vector can be found, and the system E does not admit monomial conservation law.

Now, suppose that the system E is following the mass action law. Then, each monomial with a negative sign of $\dot{x_i}$ is factorisable by x_i. For a monomial with positive sign, this is the case only if x_i is a reactant of the reaction associated to this monomial, and also a

product, appearing at least one more time (*eg.* $x_i \to 2x_i$). So, a reaction induce monomials factorisable by each species involved in this reaction if an only if the reaction is of the form :

$$\sum \underbrace{k_{i,j}}_{\neq 0} x_i \longrightarrow \sum (k_{i,j} \pm l) x_i.$$

□

Example 8.13. The model

$$\dot{x}_1 = x_1(x_2 - x_1) - \delta x_1,\ \dot{x}_2 = x_2(x_1 - x_2)$$

is a mass action network described by

$$A_1 + A_2 \xrightarrow{1} 2A_2, \quad A_1 + A_2 \xrightarrow{1} \emptyset, \quad A_2 + A_2 \xrightarrow{1} \emptyset, \quad A_1 \xrightarrow{\delta} \emptyset.$$

The truncated system

$$\dot{x}_1 = x_1(x_2 - x_1),\ \dot{x}_2 = x_2(x_1 - x_2)$$

has a continuous steady state variety $x_1 = x_2$ and singular Jacobian on it.

After truncation and elimination of the factor $\boldsymbol{x}$ one gets

$$\dot{x}_1 = x_2 - x_1,\ \dot{x}_2 = x_1 - x_2$$

that has the linear, irreducible conservation law $x_1 + x_2$.

Therefore the model has $\phi(x_1, x_2) = x_1 x_2$ as approximate, monomial, irreducible conservation law. The intersection of the steady state variety with the set $\phi = c_0$ is the point $x_1 = x_2 = c_0/2$. The monomial conservation law is complete because the 2×2 minor of the Jacobian $det(D(x_1(x_2 - x_1), x_1 x_2)^T) = -2x_1^2$ does not vanish identically.

Remark 8.14. The proposition 8.12 is a necessary condition, determining the class of models when a monomial conservation law can be found, but it is not a sufficient condition, as the following example prove it.

Example 8.15. The Lotka-Volterra network is defined by the mass action reactions

$$A_1 \xrightarrow{k_1} 2A_1, \quad A_1 + A_2 \xrightarrow{k_2} 2A_2, \quad A_2 \xrightarrow{k_3} \emptyset$$

and the ODEs

$$\dot{x}_1 = x_1(k_1 - k_2 x_2),\ \dot{x}_2 = x_2(k_2 x_1 - k_3).$$

Using algorithm 23, the matrix $\boldsymbol{B}$ is:

$$\begin{pmatrix} k_1 & 0 & k_2 \\ -k_3 & -k_2 & 0 \end{pmatrix}$$

which have a full rank. Thus, according to the Theorem 8.9, the model has not monomial conservation laws.

However, this model has a non-linear conservation law $\phi = x_1^{k_3} x_2^{k_1} e^{-k_2(x_1+x_2)}$. Furthermore, tested for $k_1 = k_2 = k_3 = 1$, the model has four discrete steady states, not a continuous variety of steady states. Two of these are non-degenerate, and two are degenerate, but the degeneracy does not follow from the continuity of the steady state variety. Therefore this example is outside the scope of the paper.

Example 8.16. A singularly perturbed Volpert model is described by the mass action network

$$A_1 + A_2 \xrightarrow{1} 2A_2, \quad A_2 + A_3 \xrightarrow{2} 2A_3, \quad A_3 + A_1 \xrightarrow{2} 2A_1, \quad A_1 \xrightarrow{\delta} \emptyset, \quad A_2 \xrightarrow{\delta} \emptyset, \quad A_3 \xrightarrow{\delta} \emptyset,$$

and the ODEs

$$\dot{x}_1 = x_1(x_3 - x_2) - \delta(x_1 + x_2 + x_3),\ \dot{x}_2 = x_2(x_1 - x_3),\ \dot{x}_3 = x_3(x_2 - x_1).$$

The truncated system $\dot{x}_1 = x_1(x_3 - x_2)$, $\dot{x}_2 = x_2(x_1 - x_3)$, $\dot{x}_3 = x_3(x_2 - x_1)$ has a continuous steady state variety $x_1 = x_2 = x_3$ and a singular Jacobian on this variety.

The stoichiometric matrix of the truncated system is $\boldsymbol{S}^{(1)} = \begin{pmatrix} -1 & 0 & 1 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \end{pmatrix}$, has rank $s = 2$ and its left kernel is generated by $\boldsymbol{c} = (1, 1, 1)$.

The stoichiometric matrix of the system obtained after truncation and elimination of the factor $\boldsymbol{x}$ is $\boldsymbol{S}'^{(1)} = \begin{pmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{pmatrix}$, has rank $s = 2$ and its left kernel is generated by the same $\boldsymbol{c} = (1, 1, 1)$.

Therefore, the model has two approximate conservation laws, one linear $\phi_1 = x_1 + x_2 + x_3$ and one monomial $\phi_2 = x_1 x_2 x_3$.

Each one of these conservation laws is complete. The Jacobian of $(F_1, F_2, F_3, \phi_2)^T$ has a non-vanishing 3×3 minor $det(D(x_1(x_3 - x_2), x_2(x_1 - x_3), x_1 x_2 x_3)^T) = x_1^2 x_2 x_3 + x_1 x_2^2 x_3 + x_1 x_2 x_3^2$ (this is equal to $3c_2^{4/3}$ on the intersection of the steady state variety with $\phi_2 = c_2$). The $(F_1, F_2, F_3, \phi_1)^T$ has a non-vanishing 3×3 minor $det(D(x_1(x_3 - x_2), x_2(x_1 - x_3), x_1 + x_2 + x_3)^T) = -(x_1 - x_2)^2 + 2x_1 x_3 + 2x_2 x_3 - x_3^2$ (this is equal to $C_1^2/3$ on the intersection of the steady state variety with $\phi_1 = c_1$).

The two conservation laws are not independent on the steady state variety. However, none of these intersections are hyperbolic steady states (see Section) and this model is outside the scope of the paper.

8.3 Computing polynomial conservation laws

We begin by showing some simple examples that introduce polynomial conservation laws.

Example 8.17. If we take as conservation law $x_1 x_2 + x_3$ then $\dot{x_3} = -\dot{x_1} x_2 - x_1 \dot{x_2}$. Reciprocally, the polynomial system:

$$\dot{x_1} = -k_1 x_1,\ \dot{x_2} = -k_2 x_2,\ \dot{x_3} = (k_1 + k_2) x_1 x_2$$

admits $x_1 x_2 + x_3$ as polynomial conservation law.

If we take as conservation law $x_1 x_2 + x_1 x_3$ then $\dot{x_3} = -\dot{x_2} - \frac{x_2}{x_1}\dot{x_1} - \frac{x_3}{x_1}\dot{x_1}$. Reciprocally, the polynomial system:

$$\dot{x_1} = -k_1 x_1,\ \dot{x_2} = -k_2 x_2,\ \dot{x_3} = k_2 x_2 + k_1 x_2 + k_1 x_3$$

admits $x_1 x_2 + x_1 x_3$ as polynomial conservation law.

In this section, we present two algorithms for computing polynomial conservation laws, that is, conservation laws $\Phi(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$. Our first algorithm uses syzygies, a well-known concept in computational commutative algebra and algebraic geometry. Computing syzygies can be expensive, as it is based on computing Gröbner bases. Our second algorithm avoids using syzygies; it uses educated guesses (Ansatzes) of polynomial conservation laws and obtains a linear system from which the coefficients of the conservation laws can be computed.

Definition 8.18. Let $R = \mathbb{K}[x_1, \ldots, x_n] = \mathbb{K}[\boldsymbol{x}]$ be the ring of polynomials over the field $\mathbb{K}$ and $I = (f_1, \ldots, f_m) \subseteq R$ be an ideal in R. A syzygy of I is an element $(g_1, \ldots, g_m) \in R^m$ such that

$$g_1 f_1 + \ldots + g_m f_m = 0.$$

Syzygies of an ideal form a submodule of R^m as an $R-$module.

Remark 8.19. Note that in the definition of a syzygy, some authors require $\{f_1, \ldots, f_m\}$ to be a minimal generating set of the ideal I. However, for the purpose of this paper, we do not require such a condition on the generators.

As an example, consider the ideal $I = (x, y) \subseteq \mathbb{R}[x, y]$. As $(-y)x + (x)y = 0$, $(-y, x) \in \mathbb{R}[x, y]^2$ is a syzygy of the ideal I. In general, for an ideal $I = (f_1, \ldots, f_m) \subseteq R$, a syzygy of the form $(0, \ldots, 0, f_j, 0 \ldots, 0, -f_i, 0, \ldots, 0)$, where f_j is the $i-$th coordinate and $-f_i$ is the $j-$th coordinate, is called a trivial syzygy of I.

Syzygies have been extensively studied in commutative algebra and algebraic geometry. Buchberger's algorithm for computing Gröbner bases [Buchberger, 1965, Bose, 1995] leads to an algorithm for computing a basis for a syzygy module. Several authors have worked on algorithms for computing a basis for a syzygy module, among which, Schreyer presented his algorithm in [Schreyer, 1991]. Many computer algebra systems such as Singular, CoCoA, etc. include implementations for computing syzygies. For more on syzygies, in particular on algorithms for computing syzygies, See, e.g., Weispfenning's book [Becker et al., 1993], or [Cox et al., 2006, Ch. 5.3].

Consider the ODE system (2.2) of a chemical reaction network, i.e.,

$$\dot{x}_1 = f_1(\boldsymbol{k}, \boldsymbol{x}), \ldots, \dot{x}_n = f_n(\boldsymbol{k}, \boldsymbol{x}) \in \mathbb{Z}[\boldsymbol{k}, \boldsymbol{x}]$$

and let $\phi(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$ be an exact polynomial conservation law of the CRN. By Definition 6.2, $\sum_{i=1}^{n} \frac{\partial \phi}{\partial x_i}(\boldsymbol{x}) f_i(\boldsymbol{k}, \boldsymbol{x}) = 0$, for all $\boldsymbol{k}$ and $\boldsymbol{x}$. Assuming unconditionality on $\boldsymbol{k}$ and $\boldsymbol{x}$ in this paper, one can see that $(\frac{\partial \phi}{\partial x_1}(\boldsymbol{x})), \ldots, \frac{\partial \phi}{\partial x_n}(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]^n \subseteq \mathbb{R}[\boldsymbol{k}, \boldsymbol{x}]^n$ is a syzygy of the steady state ideal $I = (f_1, \ldots, f_n) \subseteq \mathbb{R}[\boldsymbol{k}, \boldsymbol{x}]$, that is, the gradient of a polynomial conservation law ϕ, i.e., $\boldsymbol{\nabla} \phi(x) = (\frac{\partial \phi}{\partial x_1}(\boldsymbol{x}), \ldots, \frac{\partial \phi}{\partial x_n}(\boldsymbol{x}))$ is a syzygy of the steady state ideal.

On the other hand, if we are given a syzygy of the steady state ideal that contains variables $\boldsymbol{x}$ but does not contain rate constant $\boldsymbol{k}$, such that it is the gradient of a polynomial $\phi(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$, then $\phi(\boldsymbol{x})$ is a polynomial conservation law. Roughly speaking, this implies that for a syzygy $g = (g_1, \ldots, g_n)$, one can obtain a conservation law by integrating g. Below we clarify this idea.

Consider the differential form $dg = g_1 dx_1 + \cdots + g_n dx_n$. Take a piecewise smooth curve connecting the origin to $\boldsymbol{x}$. For instance

$$\mathcal{C} = \{(sx_1, 0, 0) \mid s \in [0, 1]\} \cup \{(x_1, sx_2, 0) \mid s \in [0, 1]\} \cup \{(x_1, x_2, sx_3) \mid s \in [0, 1]\}.$$

Integrate the differential form dg along $\mathcal{C}$, i.e.,

$$\int_0^{x_1} g_1(\cdot, 0, ..., 0)dx_1 + \int_0^{x_2} g_2(x_1, \cdot, 0, ..., 0)dx_2 + \cdots + \int_0^{x_n} g_n(x_1, ..., x_{n-1}, \cdot)dx_n.$$

The result of this integration is a polynomial $\phi(\boldsymbol{k}, \boldsymbol{x}) \in \mathbb{R}[\boldsymbol{k}, \boldsymbol{x}]$. The rotational of the vector field g defined as the tensor $[\boldsymbol{\nabla} \times g]_{ij} = \frac{\partial g_i}{\partial x_j} - \frac{\partial g_i}{\partial x_j}$ is important in this procedure. If $\boldsymbol{\nabla} \times g \neq 0$, the result of the integration depends on the path and there is no function ϕ satisfying $\boldsymbol{\nabla}\phi = g$ for all $\boldsymbol{x}$. ϕ is well defined in a simply connected domain $\mathcal{D} \subset \mathbb{R}^n$ only if $\boldsymbol{\nabla} \times g = 0$ everywhere in $\mathcal{D}$, in other words only when g defines a conservative vector field in $\mathcal{D}$. If this irrotationality condition is satisfied, than one has $\boldsymbol{\nabla}\phi = g$, $\boldsymbol{\nabla}\phi \cdot F = 0$ everywhere in $\mathcal{D}$.

Another method to obtain a conservation law out of a syzygy is by taking the symbolic integration of each coordinate g_i of a syzygy $(g_1, \ldots, g_n)$. However, some considerations should be taken into account. First of all, the integration of the polynomial in $i-$th coordinate of a syzygy vector is done with respect to the variable x_i, and therefore, it includes a polynomial in variables other than x_i, that is, $\phi(\boldsymbol{x}) = \int_{x_i} g_i + C^i(\hat{x}_i)$, where $C^i(\hat{x}_i)$ is a polynomial in all variables but x_i. The polynomial $C^i(\hat{x}_i)$ can be obtained by making an ansatz, that is, by taking its coefficients as indeterminates. Identifying monomials with the same degree, one can obtain a linear system of equations, the solutions of which will give us the coefficients of $C^i(\hat{x}_i)$. The caveat of this method is the large number of linear equations obtained. However, from the complexity point of view, given the fact that computing syzygies essentially requires computing Gröbner bases, and that computing Gröbner bases is more expensive than solving the linear system, this will not add a cost to the computations. In practice, obtaining and solving those linear equations can be expensive.

As noticed above, having obtained a syzygy, one has to check if the vector field defined by it is conservative. As a classical result, a vector field is conservative over a simply connected set if and only if it is irrotational, that is $\boldsymbol{\nabla} \times g(\boldsymbol{x}) = 0$. If this condition is not satisfied, the linear system in the second method is incompatible, as the equation $\boldsymbol{\nabla}\phi = g$ has no solution. As an example, consider the trivial syzygy $(-y, x)$ for the ideal (x, y). As the vector field arising from this particular syzygy is not irrotational, i.e., $\boldsymbol{\nabla} \times (-y, x) \neq 0$, one cannot obtain a polynomial conservation laws from this syzygy. There exist vector fields whose trivial syzygy is the gradient of a conservation law. For example, consider the vector field $(x, -y)$ with (y, x) as its trivial syzygy. Then $\boldsymbol{\nabla} \times (y, x) = 0$, which means that the trivial syzygy is irrotational, hence it is the gradient of a conservation law, which is $xy = c$, where c is a constant. In the context of CRNs, vector fields defining CRN dynamics (2.2) must satisfy the condition $\frac{\partial f_i}{\partial x_i} < 0$, for all $1 \leq i \leq n$ and at least for some $\boldsymbol{x} \in \mathbb{R}^n_+$, ensuring stability. This condition, implies that trivial syzygies of CRNs are not gradients.

Another issue to be considered on computing conservation laws via syzygies is that for the steady state ideal $I \subseteq \mathbb{R}[\boldsymbol{k}, \boldsymbol{x}]$, the syzygy module is a submodule of $\mathbb{R}[\boldsymbol{k}, \boldsymbol{x}]^n$. However, according to our definition, the conservation laws that we are interested in do not depend on the rate constants, i.e., they are polynomials in $\mathbb{R}[\boldsymbol{x}]$. Hence, in order to use syzygies for computing conservation laws, we are only interested in those syzygies that depend on the variables $x_1, \ldots, x_n$ and in which the rate constants $\boldsymbol{k}$ do not appear. One can obtain such conservation laws by first computing the ideal of conservation laws in $\mathbb{R}[\boldsymbol{k}, \boldsymbol{x}]^n$ and then intersecting it with the ring $\mathbb{R}[\boldsymbol{x}]$.

By definition of independence of conservation laws, if two polynomial conservation laws $\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x})$ are independent then the Jacobian matrix

$$\begin{pmatrix} \frac{\partial \phi_1(\boldsymbol{x})}{\partial x_1} & \cdots & \frac{\partial \phi_1(\boldsymbol{x})}{\partial x_n} \\ \frac{\partial \phi_2(\boldsymbol{x})}{\partial x_1} & \cdots & \frac{\partial \phi_2(\boldsymbol{x})}{\partial x_n} \end{pmatrix}$$

is full rank. This implies that the rows of the Jacobian are linearly independent, which means that the rows as syzygies are linearly independent. Conversely, if two linearly independent syzygies are the gradients of two conservation laws, then the conservation laws are independent.

Discussions above lead to the following algorithm, which uses syzygies in order to compute conservation laws.

Input: A polynomial ODE system $F(\boldsymbol{k}, \boldsymbol{x}) = (f_1(\boldsymbol{k}, \boldsymbol{x}), \ldots, f_n(\boldsymbol{k}, \boldsymbol{x}))$ whose r.h.s. are polynomials in $\mathbb{Z}[\boldsymbol{k}, \boldsymbol{x}]$, homogeneous of degree one in $\boldsymbol{k}$:

$$\dot{x}_i = f_i(\boldsymbol{k}, \boldsymbol{x}) = \sum_{j=1}^{r_i} z_{ij} k_{ij} \boldsymbol{x}^{\boldsymbol{\alpha}_{ij}}$$

where $z_{ij} \in \mathbb{Z}$, $\boldsymbol{\alpha}_{ij} \in \mathbb{N}^n$, $k_{ij} \in \{k_1, \ldots, k_r\}$, for $1 \leq i \leq n$, $1 \leq j \leq r_i$.

Output: A Set of Polynomial Conservation Laws $\phi(\boldsymbol{x}) \in \mathbb{Z}[\boldsymbol{x}]$ for $F(\boldsymbol{k}, \boldsymbol{x})$.

1: PolCons $:= \emptyset$
2: Compute $Syz(F)$, a basis for the set of syzygies of F of all orders.
3: **for** $G(\boldsymbol{k}, \boldsymbol{x}) = (g_1(\boldsymbol{k}, \boldsymbol{x}), \ldots, g_n(\boldsymbol{k}, \boldsymbol{x})) \in Syz(F)$ **do**
4: **if** $\boldsymbol{\nabla} \times G = 0$ **then**
5: $\phi_G(\boldsymbol{k}, \boldsymbol{x}) := \int_0^{x_1} g_1(x_1, 0, \ldots, 0) dx_1 + \int_0^{x_2} g_2(x_1, x_2, 0, \ldots, 0) dx_2 + \cdots + \int_0^{x_n} g_n(x_1, \ldots, x_n) dx_n$.
6: PolCons $=$ PolCons $\cup \{\phi_G(\boldsymbol{k}, \boldsymbol{x})\}$
7: **end if**
8: **end for**
9: UncondLaws $:=$ Ideal(PolCons) $\cap \mathbb{Z}[\boldsymbol{x}]$
10: **return** UncondLaws.

Algorithm 24: PolynomialConservationLawsViaSyzygies

Remark 8.20. As $Syz(F)$ computed in line 2 of Algorithm 24 gives a basis of the syzygy module, which includes linearly independent syzygies, the output conservation laws of the algorithm are independent. Moreover, if there exists a complete set of polynomial conservation laws, then the algorithm outputs such a complete set of conservation laws.

Example 8.21 (BIOMD0000000629, [Chelliah et al., 2015]). The ODEs corresponding to

Biomodel 629 in the BioModels' repository [1] are the following;

$$\begin{aligned}
\dot{x_1} =& - k_2x_1x_3 + k_3x_2,\\
\dot{x_2} =& k_2x_1x_3 - k_3x_2 - k_4x_2x_4 + k_5x_5,\\
\dot{x_3} =& - k_2x_1x_3 + k_3x_2,\\
\dot{x_4} =& - k_4x_2x_4 + k_5x_5,\\
\dot{x_5} =& k_4x_2x_4 - k_5x_5.
\end{aligned}$$

A basis for the syzygy module is

$$\begin{aligned}
\{G_1 =& (-1, 0, 1, 0, 0)\\
G_2 =& (0, -1, 1, 1, 0)\\
G_3 =& (0, 0, 0, 1, 1)\\
G_4 =& (k_2x_1x_3 - k_4x_2x_4 - k_3x_2 + k_5x_5, k_2x_1x_3 - k_3x_2, 0, 0, 0)\}.
\end{aligned}$$

Integrating the conservative elements of the syzygy basis, we obtain

$$\begin{aligned}
\phi_1(\boldsymbol{k}, \boldsymbol{x}) =& - x_1 + x_3\\
\phi_2(\boldsymbol{k}, \boldsymbol{x}) =& - x_2 + x_3 + x_4\\
\phi_3(\boldsymbol{k}, \boldsymbol{x}) =& x_4 + x_5
\end{aligned}$$

So we have three independent polynomial conservation laws for the biomodel 629. One should note that $G_4(\boldsymbol{k}, \boldsymbol{x})$ is not conservative, $\boldsymbol{\nabla} \times G_4 \neq 0$.

Example 8.22. Consider the example in Subsection 8.17. The ODEs in that example are

$$\begin{aligned}
\dot{x_1} =& - k_1x_1,\\
\dot{x_2} =& - k_2x_2,\\
\dot{x_3} =& (k_1 + k_2)x_1x_2
\end{aligned}$$

Running Algorithm 24, we first compute a basis for the syzygy of the steady state ideal I:

$$Syz(I) = \{(x_2, x_1, 1), (-k_2x_2, k_1x_1, 0)\}.$$

The first element of the syzygy basis is conservative and integrates to the conservation law $\phi_1 = x_1x_2 + x_3$. The second element is not conservative $\boldsymbol{\nabla} \times (-k_2x_2, k_1x_1, 0) \neq 0$, because $\frac{\partial -k_2x_2}{\partial x_2} - \frac{\partial k_1x_1}{\partial x_1} = -(k_1 + k_2) \neq 0$. There is only one conservation law and one can easily check its completeness. Indeed, the jacobian matrix $D_{\boldsymbol{x}}(-k_1x_1, -k_2x_2, x_1x_2 + x_3)^{\mathsf{T}}$, which is

$$\begin{pmatrix} -k_1 & 0 & 0\\ 0 & -k_2 & 0\\ x_2 & x_1 & 1 \end{pmatrix}$$

have rank 3, then it is for $D_{\boldsymbol{x}}(-k_1x_1, -k_2x_2, (k_1 + k_2)x_1x_2, x_1x_2 + x_3)^{\mathsf{T}}$.

[1] https://www.ebi.ac.uk/biomodels

Example 8.23. Let

$$\begin{aligned}\dot{x}_1 &= x_2,\\ \dot{x}_2 &= x_1,\\ \dot{x}_3 &= x_1 x_2.\end{aligned}$$

One can compute the following basis for the syzygy module:

$$\{(0, x_2, -1), (x_1, -x_2, 0)\}.$$

Both elements are conservatives and integrate to the following independent conservation laws:

$$\begin{aligned}&\frac{1}{2}x_2^2 - x_3,\\ &\frac{1}{2}x_1^2 - \frac{1}{2}x_2^2.\end{aligned}$$

The conservation laws obtained are polynomial and one can easily check that this is a complete set of conservation laws. Indeed, $D_{\boldsymbol{x}}(x_2, x_1, \frac{1}{2}x_2^2 - x_3)^\mathsf{T}$ is

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & x_2 & -1 \end{pmatrix}$$

which have rank 3.

Algorithm 24 first computes a polynomial $\phi(\boldsymbol{k}, \boldsymbol{x}) \in \mathbb{R}[\boldsymbol{k}, \boldsymbol{x}]$ and then obtains conservation laws via eliminating $\boldsymbol{k}$. This may lead to extra computations, in particular when there is no conservation law $\phi(\boldsymbol{x})$. One idea to avoid computations in $\mathbb{R}[\boldsymbol{k}, \boldsymbol{x}]$, but just to do computations in $\mathbb{R}[\boldsymbol{x}]$ is not to compute a basis for the syzygy module, but to ansatz a conservation law $\phi(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$ and check if $\boldsymbol{\nabla}\phi(\boldsymbol{x}) \cdot F = 0$. Algorithm 25 below uses this idea, starting from degree two polynomials and continuing to higher degrees until a conservation law is found. The algorithm stops when the first conservation law is found and does not necessarily compute a complete set of conservation laws. In worst case, the degree of the ansatz polynomial is the max degree of the polynomials in a syzygy plus one. Instead of stopping once a conservation law is found, one can also continue running the algorithm until the degree bound on syzygies, so that all the conservation laws are found.

Input: A polynomial ODE system $F(\boldsymbol{k}, \boldsymbol{x}) = (f_1(\boldsymbol{k}, \boldsymbol{x}), \ldots, f_n(\boldsymbol{k}, \boldsymbol{x}))$ whose r.h.s. are polynomials in $\mathbb{Z}[\boldsymbol{k}, \boldsymbol{x}]$, homogeneous of degree one in $\boldsymbol{k}$:

$$\dot{x}_i = f_i(\boldsymbol{k}, \boldsymbol{x}) = \sum_{j=1}^{r_i} z_{ij} k_{ij} \boldsymbol{x}^{\boldsymbol{\alpha}_{ij}}$$

where $z_{ij} \in \mathbb{Z}$, $\boldsymbol{\alpha}_{ij} \in \mathbb{N}^n$, $k_{ij} \in \{k_1, \ldots, k_r\}$, for $1 \leq i \leq n$, $1 \leq j \leq r_i$.

Output: A Set of Polynomial Conservation Laws $\phi(\boldsymbol{x}) \in \mathbb{Z}[\boldsymbol{x}]$ for $F(\boldsymbol{k}, \boldsymbol{x})$.

1: $i = 2$
2: **while** $\boldsymbol{\nabla}\phi \cdot F \neq 0$ **do**
3: (Ansatz)Let $\phi(\boldsymbol{x}) := \sum c_\alpha x^\alpha$ with indeterminate coefficients c_α and of degree i
4: Obtain linear equations in the indeterminates c_α by putting the coefficient of each monomial equal to zero in $\sum \frac{c_\alpha}{|\alpha_i|} \frac{x^\alpha}{x_i} f_i(\boldsymbol{k}, \boldsymbol{x})$.
5: Find real solutions of the linear equations and obtain c_α and consequently $\phi(\boldsymbol{x})$.
6: $i = i + 1$
7: **end while**
8: **return** $\phi(\boldsymbol{x})$.

Algorithm 25: PolynomialConservationLawsViaAnsatz

Bibliography

[Ábrahám et al., 2016] Ábrahám, E., Abbott, J., Becker, B., Bigatti, A. M., Brain, M., Buchberger, B., Cimatti, A., Davenport, J. H., England, M., Fontaine, P., Forrest, S., Griggio, A., Kroening, D., Seiler, W. M., and Sturm, T. (2016). SC^2: Satisfiability checking meets symbolic computation. In Kohlhase, M., Johansson, M., Miller, B., de Moura, L., and Tompa, F., editors, *Proc. CICM 2016*, volume 9791 of *LNCS*, pages 28–43. Springer.

[Akian et al., 2006] Akian, M., Bapat, R., and Gaubert, S. (2006). Max-plus algebra. *Handbook of linear algebra*, 39.

[Andrieux et al., 2012] Andrieux, G., Fattet, L., Le Borgne, M., Rimokh, R., and Théret, N. (2012). Dynamic regulation of tgf-b signaling by tif1γ: a computational approach. *PloS one*, 7(3):e33761.

[Antoneli et al., 2018] Antoneli, F., Golubitsky, M., and Stewart, I. (2018). Homeostasis in a feed forward loop gene regulatory motif. *Journal of theoretical biology*, 445:103–109.

[Araya and Neveu, 2018] Araya, I. and Neveu, B. (2018). lsmear: a variable selection strategy for interval branch and bound solvers. *Journal of Global Optimization*, 71(3):483–500.

[Araya et al., 2014] Araya, I., Trombettoni, G., Neveu, B., and Chabert, G. (2014). Upper bounding in inner regions for global optimization under inequality constraints. *Journal of Global Optimization*, 60(2):145–164.

[Aubin, 2009] Aubin, J.-P. (2009). *Viability theory*. Springer Science & Business Media.

[Barr et al., 2019] Barr, K., Reinitz, J., and Radulescu, O. (2019). An in silico analysis of robust but fragile gene regulation links enhancer length to robustness. *PLoS computational biology*, 15(11).

[Barrett et al., 2017] Barrett, C., Fontaine, P., and Tinelli, C. (2017). The SMT-LIB standard: Version 2.6. Technical report, Department of Computer Science, The University of Iowa.

[Bartlett and Wong, 2020] Bartlett, S. and Wong, M. L. (2020). Defining lyfe in the universe: From three privileged functions to four pillars. *Life*, 10(4):42.

[Becker et al., 1993] Becker, T., Weispfenning, V., and Kredel, H. (1993). *Gröbner Bases, a Computational Approach to Commutative Algebra*, volume 141 of *Graduate Texts in Mathematics*. Springer.

[Benhamou et al., 1999a] Benhamou, F., Goualard, F., Granvilliers, L., and Puget, J.-F. (1999a). Revising Hull and Box Consistency. In *Proc. ICLP*, pages 230–244.

[Benhamou et al., 1999b] Benhamou, F., Goualard, F., Granvilliers, L., and Puget, J.-F. (1999b). Revising hull and box consistency. In *Int. Conf. on Logic Programming*. Citeseer.

[Bergman, 1971] Bergman, G. M. (1971). The logarithmic limit-set of an algebraic variety. *Transactions of the American Mathematical Society*, 157:459–469.

[Bernard, 1879] Bernard, C. (1879). *Leçons sur les phénomènes de la vie commune aux animaux et aux végétaux*, volume 2. Baillière.

[Bessiere, 1991] Bessiere, C. (1991). Arc-consistency in dynamic constraint satisfaction problems. In *AAAI Conference on Artificial Intelligence*, volume 1, pages 221–226.

[Bessiere, 2006] Bessiere, C. (2006). Constraint propagation. In *Foundations of Artificial Intelligence*, volume 2, pages 29–83. Elsevier.

[Bohr, 1920] Bohr, N. (1920). Über die serienspektra der elemente. *Zeitschrift für Physik*, 2(5):423–469.

[Boltzmann, 1964] Boltzmann, L. (1964). *Lectures on gas theory*. U. of California Press, Berkeley, CA, USA.

[Bose, 1995] Bose, N. K. (1995). *Gröbner Bases: An Algorithmic Method in Polynomial Ideal Theory*, pages 89–127. Springer Netherlands, Dordrecht.

[Boulier et al., 2011] Boulier, F., Lefranc, M., Lemaire, F., and Morant, P.-E. (2011). Model reduction of chemical reaction systems using elimination. *Mathematics in Computer Science*, 5(3):289–301.

[Bradford et al., 2020] Bradford, R., Davenport, J. H., England, M., Errami, H., Gerdt, V., Grigoriev, D., Hoyt, C., Košta, M., Radulescu, O., Sturm, T., et al. (2020). Identifying the parametric occurrence of multiple steady states for some biological networks. *Journal of Symbolic Computation*, 98:84–119.

[Briggs and Haldane, 1925] Briggs, G. E. and Haldane, J. B. S. (1925). A note on the kinetics of enzyme action. *Biochemical journal*, 19(2):338.

[Buchberger, 1965] Buchberger, B. (1965). *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*. Doctoral dissertation, Mathematical Institute, University of Innsbruck, Innsbruck, Austria.

[Cardin and Teixeira, 2017] Cardin, P. T. and Teixeira, M. A. (2017). Fenichel theory for multiple time scale singular perturbation problems. *SIAM Journal on Applied Dynamical Systems*, 16(3):1425–1452.

[Chabert, 2020] Chabert, G. (2020). `www.ibex-lib.org`.

[Chabert and Jaulin, 2009] Chabert, G. and Jaulin, L. (2009). Contractor programming. *Artificial Intelligence*, 173(11):1079–1100.

[Chelliah et al., 2015] Chelliah, V., Juty, N., Ajmera, I., Ali, R., Dumousseau, M., Glont, M., Hucka, M., Jalowicki, G., Keating, S., Knight-Schrijver, V., Lloret-Villas, A., Nath Natarajan, K., Pettit, J.-B., Rodriguez, N., Schubert, M., Wimalaratne, S. M., Zhao, Y., Hermjakob, H., Le Novère, N., and Laibe, C. (2015). BioModels: Ten-year anniversary. *Nucl. Acids Res.*

[Collins et al., 2003] Collins, F. S., Morgan, M., and Patrinos, A. (2003). The human genome project: lessons from large-scale biology. *Science*, 300(5617):286–290.

[Conradi et al., 2017] Conradi, C., Feliu, E., Mincheva, M., and Wiuf, C. (2017). Identifying parameter regions for multistationarity. *PLoS computational biology*, 13(10):e1005751.

[Cooper, 2008] Cooper, S. J. (2008). From claude bernard to walter cannon. emergence of the concept of homeostasis. *Appetite*, 51(3):419–427.

[Cox et al., 2006] Cox, D. A., Little, J., and O'shea, D. (2006). *Using algebraic geometry*, volume 185. Springer Science & Business Media.

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

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

[Dolzmann et al., 1998] Dolzmann, A., Sturm, T., and Weispfenning, V. (1998). Real quantifier elimination in practice. In Matzat, B. H., Greuel, G.-M., and Hiss, G., editors, *Algorithmic Algebra and Number Theory*, pages 221–247. Springer.

[Feinberg, 1987] Feinberg, M. (1987). Chemical reaction network structure and the stability of complex isothermal reactors—i. the deficiency zero and deficiency one theorems. *Chemical engineering science*, 42(10):2229–2268.

[Feinberg, 1988] Feinberg, M. (1988). Chemical reaction network structure and the stability of complex isothermal reactors—ii. multiple steady states for networks of deficiency one. *Chemical Engineering Science*, 43(1):1–25.

[Feinberg and Horn, 1974] Feinberg, M. and Horn, F. J. (1974). Dynamics of open chemical systems and the algebraic structure of the underlying reaction network. *Chemical Engineering Science*, 29(3):775–787.

[Feliu and Wiuf, 2012] Feliu, E. and Wiuf, C. (2012). Preclusion of switch behavior in networks with mass-action kinetics. *Applied Mathematics and Computation*, 219(4):1449–1467.

[Fenichel, 1979] Fenichel, N. (1979). Geometric singular perturbation theory for ordinary differential equations. *Journal of differential equations*, 31(1):53–98.

[Feret et al., 2009] Feret, J., Danos, V., Krivine, J., Harmer, R., and Fontana, W. (2009). Internal coarse-graining of molecular systems. *Proceedings of the National Academy of Sciences*, 106(16):6453–6458.

[Goldman, 1987] Goldman, L. (1987). Integrals of multinomial systems of ordinary differential equations. *Journal of pure and applied algebra*, 45(3):225–240.

[Golubitsky and Stewart, 2017] Golubitsky, M. and Stewart, I. (2017). Homeostasis, singularities, and networks. *Journal of mathematical biology*, 74(1-2):387–407.

[Gorban et al., 2001] Gorban, A. N., Karlin, I. V., Ilg, P., and Öttinger, H. C. (2001). Corrections and enhancements of quasi-equilibrium states. *Journal of non-newtonian fluid mechanics*, 96(1-2):203–219.

[Gorban and Radulescu, 2007] Gorban, A. N. and Radulescu, O. (2007). Dynamical robustness of biological networks with hierarchical distribution of time scales. *IET Systems Biology*, 1(4):238–246.

[Gorban et al., 2010] Gorban, A. N., Radulescu, O., and Zinovyev, A. Y. (2010). Asymptotology of chemical reaction networks. *Chemical Engineering Science*, 65(7):2310–2324.

[Hansen, 1980] Hansen, E. (1980). Global optimization using interval analysis—the multidimensional case. *Numerische Mathematik*, 34(3):247–270.

[Hansen and Walster, 2003] Hansen, E. and Walster, G. W. (2003). *Global optimization using interval analysis: revised and expanded*, volume 264. CRC Press.

[Hansen, 1979] Hansen, E. R. (1979). Global optimization using interval analysis: the one-dimensional case. *Journal of Optimization Theory and Applications*, 29(3):331–344.

[Hoops et al., 2006] Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes, P., and Kummer, U. (2006). Copasi—a complex pathway simulator. *Bioinformatics*, 22(24):3067–3074.

[Hoppensteadt, 1967] Hoppensteadt, F. (1967). Stability in systems with parameter. *Journal of Mathematical Analysis and Applications*, 18(1):129–134.

[Hoppensteadt, 1969a] Hoppensteadt, F. (1969a). On systems of ordinary differential equations with several parameters multiplying the derivatives. *Journal of Differential Equations*, 5(1):106–116.

[Hoppensteadt, 1969b] Hoppensteadt, F. (1969b). On systems of ordinary differential equations with several parameters multiplying the derivatives. *J. Differ. Equations*, 5(1):106–116.

[Hoppensteadt, 1966] Hoppensteadt, F. C. (1966). Singular perturbations on the infinite interval. *Transactions of the American Mathematical Society*, 123(2):521–535.

[Jaulin et al., 2001] Jaulin, L., Kieffer, M., Didrit, O., and Walter, E. (2001). *Applied Interval Analysis*. Springer.

[Johnson and Goody, 2011] Johnson, K. A. and Goody, R. S. (2011). The original michaelis constant: translation of the 1913 michaelis–menten paper. *Biochemistry*, 50(39):8264–8269.

[Kamihira et al., 2000] Kamihira, M., Naito, A., Tuzi, S., Nosaka, A. Y., and Saito, H. (2000). Conformational transitions and fibrillation mechanism of human calcitonin as studied by high-resolution solid-state 13 c nmr. *Protein Science*, 9(5):867–877.

[Košta, 2016] Košta, M. (2016). *New Concepts for Real Quantifier Elimination by Virtual Substitution*. Doctoral dissertation, Saarland University, Germany.

[Krivulin, 2014] Krivulin, N. (2014). Tropical optimization problems. *arXiv preprint arXiv:1408.0313*.

[Kruff et al., 2020] Kruff, N., Lüders, C., Radulescu, O., Sturm, T., and Walcher, S. (2020). Algorithmic reduction of biological networks with multiple time scales. *Mathematics in Computer Science, in press*.

[Kruff et al., 2021] Kruff, N., Lüders, C., Radulescu, O., Sturm, T., and Walcher, S. (2021). Algorithmic reduction of biological networks with multiple time scales. *Mathematics in Computer Science*, pages 1–36.

[Lam and Goussis, 1994] Lam, S. and Goussis, D. (1994). The csp method for simplifying kinetics. *International journal of chemical kinetics*, 26(4):461–486.

[Le Novere et al., 2006] Le Novere, N., Bornstein, B., Broicher, A., Courtot, M., Donizelli, M., Dharuri, H., Li, L., Sauro, H., Schilstra, M., Shapiro, B., et al. (2006). Biomodels database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. *Nucleic acids research*, 34(suppl_1):D689–D691.

[Lemaire and Temperville, 2014] Lemaire, F. and Temperville, A. (2014). On defining and computing "good" conservation laws. In *International Conference on Computational Methods in Systems Biology*, pages 1–19. Springer.

[Litvinov and Maslov, 1998] Litvinov, G. L. and Maslov, V. P. (1998). The correspondence principle for idempotent calculus and some computer applications. *Idempotency*, 11:420–443.

[Lotka, 1925] Lotka, A. J. (1925). *Elements of physical biology*. Williams & Wilkins.

[Lüders, 2020] Lüders, C. (2020). Computing tropical prevarieties with satisfiability modulo theories (smt) solvers. *arXiv preprint arXiv:2004.07058*.

[Lüders et al., 2020] Lüders, C., Radulecsu, O., et al. (2020). Computational algebra oriented crn collection of models. *in preparation*.

[Lund, 1965] Lund, E. W. (1965). Guldberg and waage and the law of mass action. *Journal of Chemical Education*, 42(10):548.

[Maas and Pope, 1992] Maas, U. and Pope, S. B. (1992). Simplifying chemical kinetics: intrinsic low-dimensional manifolds in composition space. *Combustion and flame*, 88(3-4):239–264.

[Maclagan and Sturmfels, 2015] Maclagan, D. and Sturmfels, B. (2015). *Introduction to tropical geometry*, volume 161. American Mathematical Soc.

[Mahdi et al., 2017] Mahdi, A., Ferragut, A., Valls, C., and Wiuf, C. (2017). Conservation laws in biochemical reaction networks. *SIAM Journal on Applied Dynamical Systems*, 16(4):2213–2232.

[Malthus, 1872] Malthus, T. R. (1872). *An Essay on the Principle of Population..*

[Markov, 2010] Markov, S. (2010). Biomathematics and interval analysis: a prosperous marriage. In *AIP Conference Proceedings*, volume 1301, pages 26–36. American Institute of Physics.

[Misener and Floudas, 2014] Misener, R. and Floudas, C. (2014). ANTIGONE: Algorithms for coNTinuous / Integer Global Optimization of Nonlinear Equations. *J. Global Optimization (JOGO)*, 59(2–3):503–526.

[Moore, 1966] Moore, R. E. (1966). *Interval analysis*, volume 4. Prentice-Hall Englewood Cliffs.

[Neveu et al., 2015] Neveu, B., Trombettoni, G., and Araya, I. (2015). Adaptive Constructive Interval Disjunction: Algorithms and Experiments. *Constraints Journal*, 20(4):452–467.

[Neveu et al., 2016] Neveu, B., Trombettoni, G., and Araya, I. (2016). Node selection strategies in interval branch and bound algorithms. *Journal of Global Optimization*, 64(2):289–304.

[Nieuwenhuis et al., 2006] Nieuwenhuis, R., Oliveras, A., and Tinelli, C. (2006). Solving SAT and SAT modulo theories: From an abstract Davis–Putnam–Logemann–Loveland procedure to DPLL(T). *J. ACM*, 53(6):937–977.

[Ninin, 2015] Ninin, J. (2015). Global optimization based on contractor programming: An overview of the ibex library. In *International Conference on Mathematical Aspects of Computer and Information Sciences*, pages 555–559. Springer.

[Noble, 2003] Noble, D. (2003). The future: putting humpty-dumpty together again. *Biochemical Society Transactions*, 31(1):156–158.

[Noel et al., 2014] Noel, V., Grigoriev, D., Vakulenko, S., and Radulescu, O. (2014). Tropicalization and tropical equilibration of chemical reactions. In Litvinov, G. L. and Sergeev, S. N., editors, *Tropical and Idempotent Mathematics and Applications*, volume 616 of *Contemporary Mathematics*, pages 261–277. AMS.

[Noether, 1918] Noether, E. (1918). Invariante variationsprobleme. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*, 1918:235–257.

[O'Malley, 1971] O'Malley, R. (1971). On initial value problems for nonlinear systems of differential equations with two small parameters. *Archive for Rational Mechanics and Analysis*, 40(3):209–222.

[Radulescu et al., 2012] Radulescu, O., Gorban, A. N., Zinovyev, A., and Noel, V. (2012). Reduction of dynamical biochemical reactions networks in computational biology. *Frontiers in genetics*, 3:131.

[Radulescu et al., 2015a] Radulescu, O., Vakulenko, S., and Grigoriev, D. (2015a). Model reduction of biochemical reactions networks by tropical analysis methods. *Mathematical Modelling of Natural Phenomena*, 10(3):124–138.

[Radulescu et al., 2015b] Radulescu, O., Vakulenko, S., and Grigoriev, D. (2015b). Model reduction of biochemical reactions networks by tropical analysis methods. *Math. Model. Nat. Pheno.*, 10(3):124–138.

[Ramakrishnan and Bhalla, 2008] Ramakrishnan, N. and Bhalla, U. S. (2008). Memory switches in chemical reaction space. *PLoS computational biology*, 4(7).

[Richey et al., 1987] Richey, B., Cayley, D., Mossing, M., Kolka, C., Anderson, C. F., Farrar, T. C., and Record Jr, M. (1987). Variability of the intracellular ionic environment of escherichia coli. differences between in vitro and in vivo effects of ion concentrations on protein-dna interactions and gene expression. *Journal of Biological Chemistry*, 262(15):7157–7164.

[Rizk et al., 2009] Rizk, A., Batt, G., Fages, F., and Soliman, S. (2009). A general computational method for robustness analysis with applications to synthetic gene networks. *Bioinformatics*, 25(12):i169–i178.

[Rowley and Marsden, 2000] Rowley, C. W. and Marsden, J. E. (2000). Reconstruction equations and the karhunen–loève expansion for systems with symmetry. *Physica D: Nonlinear Phenomena*, 142(1-2):1–19.

[Samal et al., 2015] Samal, S. S., Grigoriev, D., Fröhlich, H., Weber, A., and Radulescu, O. (2015). A geometric method for model reduction of biochemical networks with polynomial rate functions. *Bulletin of Mathematical Biology*, 77(12):2180–2211.

[Samal et al., 2016] Samal, S. S., Naldi, A., Grigoriev, D., Weber, A., Théret, N., and Radulescu, O. (2016). Geometric analysis of pathways dynamics: Application to versatility of tgf-β receptors. *Biosystems*, 149:3–14.

[Schneider and Wilhelm, 2000] Schneider, K. R. and Wilhelm, T. (2000). Model reduction by extended quasi-steady-state approximation. *J. Math. Biol.*, 40(5):443–450.

[Schreyer, 1991] Schreyer, F. O. (1991). A standard basis approach to syzygies of canonical curves. *Journal für die reine und angewandte Mathematik*, 421:83–124.

[Schuster and Höfer, 1991] Schuster, S. and Höfer, T. (1991). Determining all extreme semi-positive conservation relations in chemical reaction systems: a test criterion for conservativity. *Journal of the Chemical Society, Faraday Transactions*, 87(16):2561–2566.

[Segel and Slemrod, 1989] Segel, L. A. and Slemrod, M. (1989). The quasi-steady-state assumption: A case study in perturbation. *SIAM Rev.*, 31(3):446–477.

[Seidl, 2006] Seidl, A. (2006). *Cylindrical Decomposition Under Application-Oriented Paradigms*. Doctoral dissertation, University of Passau, Germany.

[Seidl and Sturm, 2003] Seidl, A. M. and Sturm, T. (2003). Boolean quantification in a first-order context. In Ganzha, V. G., Mayr, E. W., and Vorozhtsov, E. V., editors, *Computer Algebra in Scientific Computing. Proceedings of the CASC 2003*, pages 329–345. Institut für Informatik, Technische Universität München, München, Germany.

[Shinar and Feinberg, 2010] Shinar, G. and Feinberg, M. (2010). Structural sources of robustness in biochemical reaction networks. *Science*, 327(5971):1389–1391.

[Soliman, 2012] Soliman, S. (2012). Invariants and other structural properties of biochemical models as a constraint satisfaction problem. *Algorithms for Molecular Biology*, 7(1):1–9.

[Soliman et al., 2014] Soliman, S., Fages, F., and Radulescu, O. (2014). A constraint solving approach to model reduction by tropical equilibration. *Algorithms for Molecular Biology*, 9(1):1–11.

[Sommese and Wampler, 2005] Sommese, A. and Wampler, C. I. (2005). *The Numerical solution of systems of polynomials arising in engineering and science*. World Scientific.

[Song et al., 2015] Song, S., Ghosh, J., Mainigi, M., Turan, N., Weinerman, R., Truongcao, M., Coutifaris, C., and Sapienza, C. (2015). Dna methylation differences between in vitro- and in vivo-conceived children are associated with art procedures rather than infertility. *Clinical epigenetics*, 7(1):1–10.

[Sturm, 2017] Sturm, T. (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, 2018] Sturm, T. (2018). Thirty years of virtual substitution. In *Proc. ISSAC 2018*, pages 11–16. ACM.

[Tawarmalani and Sahinidis, 2005a] Tawarmalani, M. and Sahinidis, N. V. (2005a). A Polyhedral Branch-and-Cut Approach to Global Optimization. *Mathematical Programming*, 103(2):225–249.

[Tawarmalani and Sahinidis, 2005b] Tawarmalani, M. and Sahinidis, N. V. (2005b). A polyhedral branch-and-cut approach to global optimization. *Mathematical programming*, 103(2):225–249.

[Tikhonov, 1952a] Tikhonov, A. N. (1952a). Systems of differential equations containing small parameters in the derivatives. *Matematicheskii sbornik*, 73(3):575–586.

[Tikhonov, 1952b] Tikhonov, A. N. (1952b). Systems of differential equations containing small parameters in the derivatives. *Mat. Sb. (N. S.)*, 73(3):575–586.

[Trombettoni et al., 2011a] Trombettoni, G., Araya, I., Neveu, B., and Chabert, G. (2011a). Inner regions and interval linearizations for global optimization. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*.

[Trombettoni et al., 2011b] Trombettoni, G., Araya, I., Neveu, B., and Chabert, G. (2011b). Régions intérieures et linéarisations par intervalles en optimisation globale. In *JPFC 2011*, pages 299–306.

[Tucker et al., 2007] Tucker, W., Kutalik, Z., and Moulton, V. (2007). Estimating parameters for generalized mass action models using constraint propagation. *Mathematical biosciences*, 208(2):607–620.

[Van Hentenryck et al., 1997] Van Hentenryck, P., Michel, L., and Deville, Y. (1997). *Numerica : A Modeling Language for Global Optimization*. MIT Press.

[Vasil'eva and Butuzov, 1980] Vasil'eva, A. B. and Butuzov, V. F. (1980). Singularly perturbed equations in the critical case. Technical report, WISCONSIN UNIV-MADISON MATHEMATICS RESEARCH CENTER.

[Verhulst, 2005a] Verhulst, F. (2005a). *Methods and applications of singular perturbations*. Springer.

[Verhulst, 2005b] Verhulst, F. (2005b). *Methods and Applications of Singular Perturbations: Boundary Layers and Multiple Timescale Dynamics*, volume 50. Springer Science & Business Media.

[Verhulst, 1838] Verhulst, P.-F. (1838). Notice sur la loi que la population poursuit dans son accroissement. *Corresp. Math. Phys*, 10:113–121.

[Viro, 2001] Viro, O. (2001). Dequantization of real algebraic geometry on logarithmic paper. In Casacuberta, C., Miró-Roig, R. M., Verdera, J., and Xambó-Descamps, S., editors, *European Congress of Mathematics*, volume 201 of *Progress in Mathematics*, pages 135–146. Springer.

[Volterra, 1926] Volterra, V. (1926). Variazioni e fluttuazioni del numero d'individui in specie animali conviventi.

[Wei and Prater, 1962] Wei, J. and Prater, C. D. (1962). The structure and analysis of complex reaction systems. In *Advances in Catalysis*, volume 13, pages 203–392. Elsevier.

[Westerhoff and Palsson, 2004] Westerhoff, H. V. and Palsson, B. O. (2004). The evolution of molecular biology into systems biology. *Nature biotechnology*, 22(10):1249–1252.

[Zhang, 2006] Zhang, F. (2006). *The Schur complement and its applications*, volume 4. Springer Science & Business Media.