Chemical Reaction Systems, Computer Algebra and Systems Biology

François Boulier, François Lemaire, Michel Petitot, Alexandre Sedoglavic

Chemical Reaction Systems, Computer Algebra and Systems Biology*

François Boulier[1], François Lemaire[1], Michel Petitot[1], and Alexandre Sedoglavic[1]

Université Lille I, LIFL, 59655 Villeneuve d'Ascq, France

1 Introduction

In this invited paper, we survey some of the results obtained in the computer algebra team of Lille, in the domain of systems biology. So far, we have mostly focused on models (systems of equations) arising from generalized chemical reaction systems. Eight years ago, our team was involved in a joint project, with physicists and biologists, on the modeling problem of the circadian clock of the green algae *Ostreococcus tauri.* This cooperation led us to different algorithms dedicated to the reduction problem of the deterministic models of chemical reaction systems. More recently, we have been working more tightly with another team of our lab, the BioComputing group, interested by the stochastic dynamics of chemical reaction systems. This cooperation led us to efficient algorithms for building the ODE systems which define the statistical moments associated to these dynamics. Most of these algorithms were implemented in the MAPLE computer algebra software. We have chosen to present them through the corresponding MAPLE packages.

Chemical reaction systems provide a general setting for modeling in biology. More generally, chemical kinetics may be viewed as a prototype of nonlinear science, as pointed out in [33, chapter 1].

The reaction $A \xrightarrow{\quad} B$ describes the transformation of a *species* A into a different species B. Species A is the *reactant* of the reaction. Species B is the *product.* One often endows a reaction with a symbol k, which parametrizes the speed of the transformation. The reaction is then denoted $A \xrightarrow{k} B$. The reaction $A \xrightarrow{\quad} \varnothing$ describes the transformation of species A into a species which is not part of the model. This sort of reaction often occurs when one models biological phenomenons but one usually does not encounter it in chemistry, since it is not *equilibrated.* Symmetrically, the reaction $\varnothing \xrightarrow{\quad} B$ describes the entry, in the model, of a species B, from outside the model. A more complicated reaction is $A+B \xrightarrow{\quad} C$. It is interpreted as follows: when a molecule of A encounters a molecule of B, both may react and form a molecule of a third species C. Last, one sometimes encounters reactions denoted $A+B \rightleftarrows C$. Some authors consider it as a single revertible reaction. We view it as a pair of two reactions. The

* This work has benefited from the support of the French ANR (decision number ANR-2010-BLAN-0109-03).

right to the left reaction may be described as follows: every molecule of C may break itself and yield two molecules: one of species A and one of species B.

A chemical reaction system is a set of chemical reactions. Here are two examples. The first one is a classics of chemistry lectures: it is the simplest example of an enzymatic reaction. It describes the transformation of a substrate S into a product P, in the presence of some enzyme E. An intermediate complex ES is formed:

$$E + S \underset{k_{-1}}{\overset{k_1}{\rightleftarrows}} ES \xrightarrow{k_2} E + P\,. \tag{1}$$

The second example, adapted from [27, Syst. (7.19)], models a two-species oscillator:

$$\varnothing \xrightarrow{a} A\,, \qquad A \xrightarrow{k_1} \varnothing\,, \qquad 2\,A + B \xrightarrow{k_2} 3\,A\,, \qquad \varnothing \xrightarrow{b} B\,. \tag{2}$$

In order to build systems of equations, i.e. mathematical models of chemical reaction systems, one needs more precise assumptions. There are at least two families of models: the deterministic and the stochastic ones.

2 Deterministic modeling

There are different ways to associate a precise deterministic dynamics to a chemical reaction system. In this paper, we focus on the mass-action dynamics. The parameters of the reactions are called *kinetic constants*. To each species A, one associates a function $A(t)$ which represents the concentration of the species. The evolution of each concentration is given by an ordinary differential equation, in which kinetic constants appear as parameters. The system of ordinary differential equations built using the mass-action law, from a chemical reaction system, is called the *natural deterministic model* of the system. It is given by the following formula:

$$\frac{\mathrm{d}X}{\mathrm{d}t} = N \cdot V$$

where X is the column vector of the concentrations, N is the *stoichiometry matrix*, and V is the column vector of the *laws*. The law of a reaction is obtained by multiplying the kinetic constant by the product of the concentrations of the reactants. The stoichiometry matrix involves one row per species and one column per reaction. Its coefficient, row r and column c, is equal to the number of molecules of species r produced by the reaction c. This number is equal to the number of occurences of species r as a product, minus the number of occurences of species r as a reactant. On System (2), we have

$$X = \begin{pmatrix} A(t) \\ B(t) \end{pmatrix}, \qquad N = \begin{pmatrix} 1 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{pmatrix}, \qquad V = \begin{pmatrix} a \\ k_1\,A(t) \\ k_2\,A(t)^2\,B(t) \\ b \end{pmatrix}.$$

Natural deterministic models of chemical reaction systems were intensively studied. Many informations can be read in the stoichiometry matrix alone: the nullspace of its transpose provides linear conservation laws. System (2) does not have any, while computing a basis of this nullspace for System (1) yields the two laws:

$$-E(t) + S(t) + P(t) = \mathrm{cst}_1\,, \qquad E(t) + ES(t) = \mathrm{cst}_2\,. \tag{3}$$

A striking property of chemical reaction systems is a very simple necessary and sufficient condition for a system of polynomial ODE to be the natural deterministic model of a chemical reaction system: it is necessary and sufficient that, in the right hand side of the ODE which describes the evolution of any concentration $A(t)$, there does not exist any monomial which is both free of $A(t)$ and endowed with a minus sign [33, section 4.7.1]. Another striking, but much more difficult result is the *zero deficiency theorem* of Feinberg, Horn and Jackson [14], which gives a sufficient condition for a system to admit a unique steady state with strictly positive coordinates. This sufficient condition, which can be formulated in the setting of the graph theory, can be implemented very easily. This theorem was much studied, recently, from different points of views. See [8, 15] and the references therein.

The size of natural deterministic models becomes so large, on all but the simplest examples, that it forbids any further analysis. The computer algebra team of Lille has focused on the problem of reducing models. We make use of two types of reduction: a reduction method which permits to approximate the natural deterministic model, under some simplifying assumptions ; and a reduction method, which permits to reduce the number of parameters, and to separate parameters which have an effect on the coordinates and the stabilities of the steady points, from the ones which have an effect on their stabilities only.

2.1 Approximating Models

The approximation method implements the well-known quasi-steady state approximation technique. See [20] for a general presentation and [39, 30] for more chemically oriented texts. Our contribution [4] consisted in formulating it, in the context of chemical reaction systems, as the result of a differential elimination process [3]. In this section, we present it over System (1), by showing how to obtain the famous Henri, Michaelis and Menten formula [18, 25]

$$\frac{\mathrm{d}S}{\mathrm{d}t}(t) = -\frac{V_{\max}\, S(t)}{K + S(t)} \tag{4}$$

where $V_{\max}$ and K are two parameters which can be expressed from the kinetic constants of the chemical reaction system. See [29] for a closely related work. Our algorithm was successfully applied over a more complicated model [6, 5], featuring two genes, arising from the modeling problem of the circadian clock of the green algae *Ostreococcus tauri* [12].

Computation using DifferentialAlgebra. Our method relies on the assumption that the chemical reactions are split into two sets: the set of the fast reactions, and the set of the slow ones. In the case of the Henri, Michaelis and Menten reduction, the revertible reaction is supposed to be fast, compared to the third one: k_1, $k_{-1} \gg k_2$. One starts by building the natural deterministic model.

```
> with (LinearAlgebra):
> X := <E(t), S(t), ES(t), P(t)>:
> V := <k[1]*E(t)*S(t), k[-1]*ES(t), k[2]*ES(t)>:
> N := <<-1, -1, 1, 0> | <1, 1, -1, 0> | <1, 0, -1, 1>>:
> X, N, V;
                  [E(t) ]  [-1     1     1]
                  [     ]  [              ]  [k[1] E(t) S(t)]
                  [S(t) ]  [-1     1     0]  [              ]
                  [     ], [              ], [ k[-1] ES(t)  ]
                  [ES(t)]  [ 1    -1    -1]  [              ]
                  [     ]  [              ]  [  k[2] ES(t)  ]
                  [P(t) ]  [ 0     0     1]
```

Then, in the vector V, one replaces the laws corresponding to the fast reactions by two unknown laws $F_1(t)$ and $F_{-1}(t)$. Denote W this new vector. The approximated system is obtained by enlarging the ODE system $\mathrm{d}X/\mathrm{d}t = N \cdot W$, with the equation $k_1\, E(t)\, S(t) = k_{-1}\, E(t)\, S(t)$, which gives the algebraic variety where the fast reactions would be equilibrated, if the slow reactions did not exist.

```
> W := <F[1](t), F[-1](t), k[2]*ES(t)>:
> equilibre := 0 = k[1]*E(t)*S(t) - k[-1]*ES(t):
> redsys := map (diff, X, t) = N . W:
> redsys := [ seq (lhs (redsys) [i] = rhs (redsys) [i],
                                i = 1 .. Dimension (X)), equilibre ];
           d
redsys := [-- E(t) = -F[1](t) + F[-1](t) + k[2] ES(t),
           dt

    d                              d
    -- S(t) = -F[1](t) + F[-1](t), -- ES(t) = F[1](t) - F[-1](t) - k[2] ES(t),
    dt                             dt

    d
    -- P(t) = k[2] ES(t), 0 = k[1] E(t) S(t) - k[-1] ES(t)]
    dt
```

This differential-algebraic equation can now be simplified, by means of a differential elimination software, such as the DifferentialAlgebra package or the recent [1]. With the DifferentialAlgebra package, the output involves three cases. We only give the equation which expresses the evolution of $S(t)$, in the general case.

```
> with (DifferentialAlgebra):
> Field := field (generators = [k[1],k[-1],k[2]]);
```

```
                Field := field(generators = [k[1], k[-1], k[2]])

> R := DifferentialRing
         (blocks = [[F[1],F[-1]], [ES,E,P,S], [k[1](),k[-1](),k[2]()]],
          derivations = [t]);
                             R := differential_ring

> ideal := RosenfeldGroebner (redsys, basefield = Field, R):
ideal := [regular_differential_chain, regular_differential_chain,

    regular_differential_chain]

> Equations (ideal[1], solved, leader=diff(S(t),t));
                              2     2
         d           E(t) S(t)  k[1]  k[2] + E(t) S(t) k[1] k[-1] k[2]
        [-- S(t) = - -------------------------------------------------]
         dt                                                         2
                        E(t) k[1] k[-1] + S(t) k[1] k[-1] + k[-1]
```

This equation is not yet the Henri, Michaelis and Menten formula, since some minor hypotheses are missing: one still needs to take conservation laws (3) and some initial values into account. Last, one needs to rename $K = k_1/k_{-1}$ and $V_{\max} = k_2\,E(0)$ and to neglect some monomial $K\,E(0)$, assuming $S(0) \gg E(0)$. Here is a sequence of computations, which takes these hypotheses into account and leads to the sought formula.

```
> conservation_laws :=
       [E(t) + ES(t) = E0 + ES0,
        S(t) + ES(t) + P(t) = S0 + ES0 + P0]:
> hypotheses := [P0 = 0, ES0 = 0, op (conservation_laws)]:
> R := DifferentialRing (
            blocks = [[F[1],F[-1]], [ES,E,P,S],
                      [ES0(),E0(),P0(),S0()], [k[1](),k[-1](),k[2]()]],
            derivations = [t]):
> ideal := RosenfeldGroebner ([ op(redsys), op(hypotheses) ],
                                               R, basefield = Field):
> formula := Equations (ideal[1], solved, leader = diff (S(t),t)):
> formula := subs (k[-1]=K*k[1], formula):
> formula := normal (formula):
> formula := algsubs (k[2]*E0=Vmax, formula):
> formula := normal (subs (K*E0=0, formula));

                                   d           Vmax S(t)
                       formula := [-- S(t) = - ---------]
                                   dt          S(t) + K
```

Computation using MABSys. The same computation can also be performed by the dedicated MABSys package [24, 23], which relies on the MAPLE RegularChains [21] and ELPS [36] packages. The MABSys package gathers as input a chemical reaction system. Each reaction can be defined as fast or slow. The package permits to obtain directly the stoichiometry matrix, the vector of the laws, the system of the equilibria, the conservation laws that can be read from the stoichiometry matrix and the natural deterministic model.

```
> with(MABSys):
> R1 := NewReaction (E+S, C, MassActionLaw(k1), fast=true):
> R2 := NewReaction (C, E+S, MassActionLaw(km1), fast=true):
> R3 := NewReaction (C, E+P, MassActionLaw(k2)):
> RS := [R1,R2,R3]:

> ConservationLaws(RS);
                [C + P + S - C_0 - P_0 - S_0, C + E - C_0 - E_0]

> sys := ReactionSystem2ODEs(RS, [E,S,C,P]);


        d
sys := [-- E(t) = -k1 E(t) S(t) + km1 C(t) + k2 C(t),
        dt


    d
    -- S(t) = -k1 E(t) S(t) + km1 C(t),
    dt

    d                                            d
    -- C(t) = k1 E(t) S(t) - km1 C(t) - k2 C(t), -- P(t) = k2 C(t)]
    dt                                           dt
```

The package provides a function that performs the quasi-steady state approximation using the information provided by the `fast` boolean.

```
> output := ModelReduce(RS, [E,C,P,S], useConservationLaws=true):
> red_sys := output[1][1]:
> red_sys := subs (C_0=0, P_0=0, red_sys):
> red_sys[4];

           d                  E_0 k2 k1 S(t) (k1 S(t) + km1)
           -- S(t) = - ---------------------------------------------
           dt            2     2                                   2
                       k1  S(t)  + 2 S(t) km1 k1 + km1 k1 E_0 + km1
```

It also provides a function that permits to reduce the number of parameters of the system. Over this example, the parameter k_1 is removed, using the assumption that it is strictly positive. The change of coordinates is provided, following the syntax: “new parameter = rational function of the old parameters”.

```
> output := InvariantizeByScalings (red_sys, [k1], [km1,k2,E,C,P,S]):
> red_sys2 := output[1]:
> red_sys2[4];
```

```
                d                S(t) k2 E_0 (S(t) + km1)
                -- S(t) = - -----------------------------------
                dt              2                             2
                            S(t)  + 2 km1 S(t) + E_0 km1 + km1
```

```
> output[2];
                                         km1
                                  [km1 = ---]
                                         k1
```

Other functions, which are still prototypes, hence not yet integrated in MABSys, permit to perform the final approximation which yields the Henri, Michaelis and Menten formula.

2.2 Reducing and Reparametrizing Models

This section, in which we give some more details on the method which permitted us to remove the k_1 parameter, is much inspired from [38, chapter 2] and [22]. The goal consists in reducing and reparametrizing an initial model into a reduced one (the initial model possibly comes from the quasi-steady state approximation method). The reduced model is equivalent to the initial one, by an invertible change of coordinates, which preserves the most important properties: the positive steady points of the initial model are in bijection with the positive steady points of the reduced one, the initial model presents an oscillating behaviour if, and only if, the reduced model does, and so on.

The computed changes of coordinates are given by Lie symmetries of the initial system of the simplest type: they are restricted to scalings. Moreover, the method distinguishes the parameters which are supposed to be strictly positive (dividing by them is allowed) from the ones which are only supposed to be nonnegative (dividing by them is forbidden). The computations of the scalings is performed by the ELPS package. It is important to point out that this package computes, in general, a restricted set of the scalings of the input system, in order to preserve a worst case polynomial complexity [35].

The method is illustrated over System (2), whose natural deterministic model [27, Eqs. (7.20)] depends on two variables and four parameters.

```
> ODS := [
>     diff(A(t),t) = a - k1 * A(t) + k2 * A(t)^2 * B(t),
>     diff(B(t),t) = b - k2*A(t)^2*B(t)
> ];
          d                              2       d                    2
  ODS := [-- A(t) = a - k1 A(t) + k2 A(t)  B(t), -- B(t) = b - k2 A(t)  B(t)]
          dt                                     dt
```

The four parameters are supposed to be strictly positive. The steady points of the natural deterministic model are the zeros of non differential system obtained by equating to zero, the right hand sides of the model equations. The steady points obviously depend on the four parameters.

The reduction step. It makes use of the scalings of the natural deterministic model. These scalings can be described by the two following changes of coordinates. The first one depends on a parameter ν_1. The second one depends on a parameter ν_2.

$$(t,\, A,\, B,\, k_1,\, k_2,\, a,\, b) \longrightarrow \left(t,\, A\,\nu_1,\, B\,\nu_1,\, k_1,\, \frac{k_2}{\nu_1^2},\, a\,\nu_1,\, b\,\nu_1\right)\,.$$

$$(t,\, A,\, B,\, k_1,\, k_2,\, a,\, b) \longrightarrow \left(\frac{t}{\nu_2^2},\, A\,\nu_2,\, B\,\nu_2,\, k_1\,\nu_2^2,\, k_2,\, a\,\nu_2^3,\, b\,\nu_2^3\right)\,.$$

Assuming that the four parameters are strictly positive, these scalings permit to rewrite the natural deterministic model in a new coordinate set. The new system depends on two parameters instead of four. The change of coordinates is given using the syntax: "new variables = rational fractions of the old ones".

```
> out := InvariantizeByScalings (ODS, [a,b,k1,k2]):
> ODS_reduced := out[1];
ODS_reduced :=

     d                            2       d                    2
    [-- A(t) = 1 - A(t) + k2 A(t)  B(t), -- B(t) = b - k2 A(t)  B(t)]
     dt                                  dt

> Change_of_coordinates := out[2];
                                              2
                                          k2 a                 A k1      B k1
  Change_of_coordinates := [b = b/a, k2 = -----, t = t k1, A = ----, B = ----]
                                             3                  a         a
                                           k1
```

The reparametrization step. It makes use of the scalings of the system which gives the steady points of `ODS_reduced`. These scalings can be described by the following change of coordinates, which depends on a parameter ν_3.

$$(A,\, B,\, k_2,\, b) \longrightarrow \left(A,\, B\,\nu_3,\, \frac{k_2}{\nu_3},\, b\right)\,.$$

The key idea is now very simple: apply these scalings, not on the system which gives the steady points of `ODS_reduced`, but on `ODS_reduced` itself. In the resulting system, the parameter k_2 is still present. However, it has no control anymore on the location of the steady points: it can only control their stabilities.

```
> out := SemiRectifySteadyPoints (ODS_reduced, [b,k2,A,B], []):
> Reparamatrized_system := out[1][1];
Reparamatrized_system :=
     d                        2       d                  2
    [-- A(t) = 1 - A(t) + A(t)  B(t), -- B(t) = (b - A(t)  B(t)) k2]
     dt                               dt

> Change_of_coordinates := out[1][3];
                      Change_of_coordinates := [B = k2 B]
```

3 Stochastic Modeling

The dynamics of chemical reaction systems can also be studied with a stochastic point of view. The parameters of the reactions are *stochastic constants*, which are probabilities that reactions occur per unit of time [17, page 2342]. To each species A, one associates a random variable $A(t)$ which counts the number of molecules of A. In a celebrated paper [17], Gillespie provided a numerical simulation algorithm, based on a strong rigorous analysis, under some assumptions ; the main one being that the chemically reacting system keeps being well-stirred over the time.

In the context of biological modeling, Gillespie's main assumption is very unlikely. However, the analysis of the stochastic behaviour of chemical reaction systems is a very important counterpart of the deterministic analysis. In a living cell, some molecules may occur in a very little number: if one views a gene, in the "active" state, as a chemical species, the number of molecules is zero or one, and handling this number as a real valued concentration is contestable. In such cases, it is known that there exist many different stochastic behaviours which are, at least, difficult to reproduce by a deterministic model. See [19, page 454]. Moreover, stochastic simulations may reproduce the surprising effects of the "noise" on the models, as pointed out in [41] and the references therein.

It is well-known that the evolution, over the time t, of the moments (mean, variance, covariance) of the random variable associated to each species may be described by a system of ordinary differential equations, at least for chemical systems the reactions of which do not involve more than one reactant [31, 32]. For more general systems, the differential system is, in general, infinite, and approximating it by a finite ODE system, by performing a so-called *moment closure*, is a difficult problem [37, 16]. In this area, our contribution [40] consisted in showing how to build these ODE systems by using Weyl algebra methods. The use of Weyl algebra led us to a few algorithmic improvements (reducing expressions swells and taking advantage of linear conservation laws at an early stage of the formula generation process).

4 Analysis of Statistical Moments

Before proceeding, one needs to associate a precise stochastic dynamics to a chemical reaction system. This can be achieved using stochastic Petri nets[1], endowed with their standard temporisation [40] ; or by relying on the Gillespie's stochastic simulation algorithm.

Define the *state* ν of a given chemical reaction system as a vector of non-negative integers (one coordinate per chemical species, each coordinate being a number of molecules) ; $\pi_\nu(t)$ as the probability that the chemical reaction system

[1] According to some sources, cited on the *Wikipedia*, Petri nets were invented by Carl Adam Petri to model chemical reaction systems at the age of 13.

be in the state ν at time t ; and the probability generating function

$$\phi(t,\, z) = \sum_{\nu \geq 0} \pi_\nu(t)\, z^\nu \,, \tag{5}$$

where z^ν stands for $z_1^{\nu_1}\, z_2^{\nu_2} \cdots z_n^{\nu_n}$ (one “symbol” z_i per chemical species). Given any chemical reaction system, it is possible to compute a general equation for ϕ [33, Eq. (5.60)] as follows. To each reaction

$$\alpha_1\, A_1 + \alpha_2\, A_2 + \cdots + \alpha_n\, A_n \xrightarrow{c} \beta_1\, A_1 + \beta_2\, A_2 + \cdots + \beta_n\, A_n \,,$$

associate the differential operator

$$\frac{c}{\alpha!}\, \left(z^\beta - z^\alpha\right) \left(\frac{\partial}{\partial\, z}\right)^{\alpha} . \tag{6}$$

The very same operator can be denoted using *Euler operators* $\theta_i = z_i\, \partial / \partial\, z_i$ instead of partial derivatives

$$\frac{c}{\alpha!}\, \left(z^{\beta - \alpha} - 1\right)\, \theta^{\underline{\alpha}} , \tag{7}$$

where $\theta^{\underline{\alpha}}$ denotes the product of the $\theta_i^{\underline{\alpha_i}} = \theta_i\, (\theta_i - 1) \cdots (\theta_i - \alpha_i + 1)$. Define H as the sum of the differential operators (6), or (7), for all reactions of the considered chemical reaction system. The general equation for ϕ is:

$$\frac{\partial}{\partial\, t}\, \phi(t,\, z) = H\, \phi(t,\, z) \,. \tag{8}$$

With partial derivatives, the differential operator H, for System (2), is

$$a\, (z_A - 1) + k_1\, (1 - z_A)\, \frac{\partial}{\partial\, z_A} + \frac{k_2}{2}\, \left(z_A^3 - z_A^2\, z_B\right)\, \frac{\partial^3}{\partial\, z_A^2\, \partial\, z_B} + b\, (z_B - 1) \,.$$

With Euler operators, the differential operator H, for System (1), is

$$k_1\, \left(\frac{z_{ES}}{z_E\, z_S} - 1\right)\, \theta_E\, \theta_S + k_{-1}\, \left(\frac{z_E\, z_S}{z_{ES}} - 1\right)\, \theta_{ES} + k_2\, \left(\frac{z_E\, z_P}{z_{ES}} - 1\right)\, \theta_{ES} \,.$$

By differentiating the probability generating function (5) and evaluating it at $z_1 = z_2 = \cdots = z_n = 1$, one gets formulae which bind ϕ and the moments of the random variables which count the numbers of molecules. A mere evaluation yields: $\phi(t, z) \mid_{z=1} = 1$. Differentiating (5) with respect to any z_i and evaluating at $z = 1$ provides the expected value of the number of molecules of the chemical species associated to z_i.

These ideas are illustrated using a prototype software available at [34], on System (1). Variables are numbered, with the convention $(E,\, S,\, ES,\, P) = (1,\, 4,\, 2,\, 3)$. The differential operator H is displayed using Euler operators.

```
> cr1 := ChemicalReaction (E+S = ES, k[1], ""):
> cr2 := ChemicalReaction (ES = E+S, k[-1], ""):
> cr3 := ChemicalReaction (ES = E+P, k[2], ""):
> CRS := ChemicalSystem (cr1,cr2,cr3):
> HSyst := HamiltonianSystem (CRS, z, theta, x(t));
                          HSyst := hamiltonian_system

> map (numero, [E,S,ES,P], HSyst);
                                  [1, 4, 2, 3]

> H := Hamiltonian (HSyst);
          /  z[2]       \                           /z[1] z[4]    \
H := k[1] |--------- - 1| theta[1] theta[4] + k[-1] |--------- - 1| theta[2]
          \z[1] z[4]    /                           \  z[2]       /

            /z[1] z[3]    \
     + k[2] |--------- - 1| theta[2]
            \  z[2]       /
```

The next command permits to generate an ODE system for the expected values and the covariances of the random variables which count the molecules of E, S, ES and P (the variable $x_1(t)$ is the expected value of the random variable which counts the molecules of the enzyme E ; the variable $x_{1,4}(t)$ is the covariance of the random variables which count, respectively, the molecules of the enzyme E and the product P). Since some reactions of System (1) involve two reactants, the ODE system is infinite. The function truncates it. The parameter iv receives some information on initial values, for a further numerical integration.

```
> ODEs := mean_equations (HSyst, 'iv');

         d
ODEs := [-- x[1](t) = -%2 + (k[-1] + k[2]) x[2](t) - %1,
         dt

    d                                               d
    -- x[2](t) = %2 + (-k[-1] - k[2]) x[2](t) + %1, -- x[3](t) = k[2] x[2](t),
    dt                                              dt

    d                                      d
    -- x[4](t) = -%2 + k[-1] x[2](t) - %1, -- x[1, 1](t) = ...
    dt                                     dt

%1 := k[1] x[1, 4](t)

%2 := k[1] x[4](t) x[1](t)
```

5 Conclusion

Beyond the improvements of the underlying theories, a lot of improvements can be brought to the software. Since this paper is software oriented, let us focus on that single issue. First, our packages could certainly be polished and merged. Second, they miss very important tools dedicated to the solving problem of polynomial systems in the field of the real numbers, such as [10, 7, 42, 11, 26]. See [28] for a recent study of the sizes of the problems that can be investigated using some of the best available such tools. However, even the nice integrated MAPLE package that we could foresee would miss two features: sort of a *model checking* functionality, which would permit to the practitioner to query models, as [13, 2] do ; and a user interface of the same quality as that of, say, *Cytoscape* [9]. These two missing features, which are both user interface related, may seem a bit irrelevant to the traditional audience of a computer algebra conference. The need for them is however pretty obvious for any lecturer, teaching symbolic techniques close to ours, among other systems biology approaches.

Bibliography

[1] Thomas Bächler, Vladimir Gerdt, Markus Lange-Hegermann, and Daniel Robertz. Thomas Decomposition of Algebraic and Differential Systems. In *Proceedings of CASC 2010*, volume 6244 of *LNCS*, pages 31–54, 2010.

[2] Grégory Batt, Michel Page, Irène Cantone, Gregor Goessler, Pedro Monteiro, and Hidde de Jong. Efficient parameter search for qualitative models of regulatory networks using symbolic model checking. *ECCB*, 26:i603–i610, 2010.

[3] François Boulier, Daniel Lazard, François Ollivier, and Michel Petitot. Computing representations for radicals of finitely generated differential ideals. *Applicable Algebra in Engineering, Communication and Computing*, 20(1):73–121, 2009. (1997 Techrep. IT306 of the LIFL).

[4] François Boulier, Marc Lefranc, François Lemaire, and Pierre-Emmanuel Morant. Model Reduction of Chemical Reaction Systems using Elimination, 2007. Presented at the international conference MACIS 2007, `http://hal.archives-ouvertes.fr/hal-00184558`.

[5] François Boulier, Marc Lefranc, François Lemaire, and Pierre-Emmanuel Morant. Applying a rigorous quasi-steady state approximation method for proving the absence of oscillations in models of genetic circuits. In K. Horimoto et al., editor, *Proceedings of Algebraic Biology 2008*, number 5147 in LNCS, pages 56–64. Springer Verlag Berlin Heidelberg, 2008.

[6] François Boulier, Marc Lefranc, François Lemaire, Pierre-Emmanuel Morant, and Aslı Ürgüplü. On proving the absence of oscillations in models of genetic circuits. In K. Horimoto H. Anai and T. Kutsia, editors, *Proceedings of Algebraic Biology 2007*, volume 4545 of *LNCS*, pages 66–80. Springer Verlag Berlin Heidelberg, 2007. `http://hal.archives-ouvertes.fr/hal-00139667`.

[7] Christopher W. Brown. QEPCAD B: a program for computing with semi-algebraic sets using CADs. *SIGSAM Bulletin*, 37(4):97–108, 2003. `http://www.cs.usna.edu/~qepcad/B/QEPCAD.html`.

[8] Madalena Chaves and Eduardo D. Sontag. State-Estimators for Chemical Reaction Networks of Feinberg-Horn-Jackson Zero Deficiency Type. *European Journal Control*, 8:343–359, 2002.

[9] The Cytoscape Consortium. Cytoscape: An Open Source Platform for Complex Network Analysis and Visualization. `http://www.cytoscape.org`, 2001-2010.

[10] Andreas Dolzmann and Thomas Sturm. Redlog: computer algebra meets computer logic. *SIGSAM Bulletin*, 31(2):2–9, 1997.

[11] Mohab Safey El Din. RAGLib: A library for real solving polynomial systems of equations and inequalities. `http://www-salsa.lip6.fr/~safey/RAGLib`, 2007.

[12] Évelyne Derelle and others. Genome Analysis of the smallest free-living eukaryote *Ostreococcus tauri* unveils many unique features. *Proceedings of the National Academy of Science of the USA*, 103(31), August 2006.

[13] François Fages, Sylvain Soliman, and Nathalie Chabrier-Rivier. Modelling and querying interaction networks in the biochemical abstract machine BIOCHAM. *Journal of Biological Physics and Chemistry*, 4:64–73, 2004.

[14] Martin Feinberg. The Existence and Uniqueness of Steady States for a Classe of Chemical Reaction Networks. *Arch. Rational Mech. Anal.*, 132:311–370, 1995.

[15] Karin Gatermann, Markus Eiswirth, and Anke Sensse. Toric ideals and graph theory to analyze Hopf bifurcations in mass action systems. *Journal of Symbolic Computation*, 40(6):1361–1382, 2005.

[16] Colin S. Gillespie. Moment-closure approximations for mass-action models. *Systems Biology, IET*, 3(1), 2009.

[17] Daniel T. Gillespie. Exact Stochastic Simulation of Coupled Chemical Reactions. *Journal of Physical Chemistry*, 81(25):2340–2361, 1977.

[18] Victor Henri. *Lois générales de l'Action des Diastases.* Hermann, Paris, 1903.

[19] Mads Kœrn, Timothy C. Elston, William J. Blake, and James J. Collins. Stochasticity in gene expression: from theories to phenotypes. *Nature*, 6:451–464, 2005.

[20] Petar Kokotovic, Hassan K. Khalil, and John O'Reilly. *Singular Perturbation Methods in Control: Analysis and Design.* Classics in Applied Mathematics 25. SIAM, 1999.

[21] François Lemaire, Marc Moreno Maza, and Yuzhen Xie. The RegularChains library in MAPLE 10. In Ilias S. Kotsireas, editor, *The MAPLE conference*, pages 355–368, 2005.

[22] François Lemaire and Aslı Ürgüplü. A method for semi-rectifying algebraic and differential systems using scaling type Lie point symmetries with linear algebra. In *Proceedings of ISSAC 2010*, pages 85–92, München, Germany, August 2010.

[23] François Lemaire and Aslı Ürgüplü. Mabsys: Modeling and analysis of biological systems. In Katsuhisa Horimoto, Masahiko Nakatsui, and Nikolaj Popov, editors, *Proceedings of Algebraic and Numeric Biology 2010*, Castle of Hagenberg, Austria, August 2010.

[24] François Lemaire and Aslı Ürgüplü. The MABSys MAPLE package. `http://www.lifl.fr/~lemaire/MABSys`, 2010.

[25] Leonor Michaelis and Maud Menten. Die kinetik der invertinwirkung. *Biochemische Zeitschrift*, 49:333–369, 1973. Partial translation in english on `http://web.lemoyne.edu/~giunta/menten.html`.

[26] Guillaume Moroz and Fabrice Rouillier. DV: Un logiciel de classification des solutions réelles d'un système paramétré. `http://www-spiral.lip6.fr/~moroz/fr/software.html`, 2009.

[27] J. D. Murray. *Mathematical Biology I. An Introduction*, volume 17 of *Interdisciplinary Applied Mathematics.* Springer Verlag, third edition, 2002.

[28] Wei Niu. *Qualitative Analysis of Biological Systems Using Algebraic Methods.* PhD thesis, Université Paris VI, Paris, June 2011.

[29] Anna Lena Nöthen. *Quasistationarität und Fast-Invariante Mengen Gewönlicher Differentialgleichungen.* PhD thesis, Rheinisch-Westfälischen Technischen Hochschule, 2008.

[30] Hans G. Othmer. Analysis of Complex Reaction Networks in Signal Transduction, Gene Control and Metabolism. `http://www.ricam.oeaw.ac.at/publications/download/summerschool/LectureNotes_Othmer.pdf`, 2006.

[31] Johan Paulsson. Models of stochastic gene expression. *Physics of Life Reviews*, 2:157–175, 2005.

[32] Johan Paulsson and Johan Elf. Stochastic Modeling of Intracellular Kinetics. In Zoltan Szallasi, Jörg Stelling, and Vipul Periwal, editors, *System Modeling in Cellular Biology: From Concepts to Nuts and Bolts*, pages 149–175. Cambridge, Massachussets: The MIT Press, 2006.

[33] Péter Érdi and János Tóth. *Mathematical models of chemical reactions: theory and applications of deterministic and stochastic models.* Princeton University Press, 1989.

[34] Michel Petitot. The MAGNUS MAPLE software. `http://www.lifl.fr/~petitot/recherche/exposes/ANB2010`, 2010.

[35] Alexandre Sedoglavic. Reduction of Algebraic Parametric Systems by Rectification of their Affine Expanded Lie Symmetries. In K. Horimoto H. Anai and T. Kutsia, editors, *Proceedings of Algebraic Biology 2007*, volume 4545 of *LNCS*, pages 277–291, 2007.

[36] Alexandre Sedoglavic and Aslı Ürgüplü. Expanded Lie Point Symmetry (MAPLE package). `http://www.lifl.fr/~sedoglav/Software`, 2007.

[37] Abhyudai Singh and João Pedro Hespanha. Lognormal Moment Closures for Biochemical Reactions. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 2063–2068, 2006.

[38] Aslı Ürgüplü. *Contribution to Symbolic Effective Qualitative Analysis of Dynamical Systems; Application to Biochemical Reaction Networks.* PhD thesis, University Lille I, Lille, France, 2010.

[39] Vincent Van Breusegem and George Bastin. Reduced order dynamical modelling of reaction systems: a singular perturbation approach. In *Proceedings of the 30th IEEE Conference on Decision and Control*, pages 1049–1054, Brighton, England, December 1991.

[40] Samuel Vidal, Michel Petitot, François Boulier, François Lemaire, and Céline Kuttler. Models of Stochastic Gene Expression and Weyl Algebra. In Katsuhisa Horimoto, Masahiko Nakatsui, and Nikolaj Popov, editors, *Proceedings of Algebraic and Numeric Biology 2010*, pages 50–67, Castle of Hagenberg, Austria, August 2010.

[41] José M. G. Vilar, Hao Yuan Kueh, Naama Barkai, and Stanislas Leibler. Mechanisms of noise-resistance in genetic oscillators. *Proceedings of the National Academy of Science of the USA*, 99(9):5988–5992, 2002.

[42] Bican Xia. DISCOVERER: A tool for solving semi-algebraic systems. In *ACM Commun. Comput. Algebra*, volume 41, pages 102–103, 2007.