

Dynamic modeling of biochemical reactions with application to signal transduction: principles and tools using Mathematica

János TÓTH^a, Jean-Pierre ROSPARS^{b,*}

^a *Department of Analysis, Institute of Mathematics, Budapest University of
Technology and Economics, Budapest, H-1521 Hungary*

^b *Institut National de la Recherche Agronomique, Unité de Phytopharmacie et
Médiateurs chimiques, Versailles, France, and Unité Mathématiques et Informatique
Appliquées, Jouy-en-Josas, France*

Abstract. Modeling of biochemical phenomena is based on formal reaction kinetics. This requires the translation of the original reaction systems into sets of differential equations expressing the effects of the various reaction steps. The temporal behavior of the system is obtained by solving the differential equations. We present the main concepts on which the formal approach of these two problems is based and we show how the amount of work needed to treat them can be significantly reduced by using a mathematical program package (*Mathematica*). Symbolic and numerical calculations can be performed with the programs presented and graphic presentations of the behavior of the system be obtained. The basic ideas are illustrated with three examples taken from the area of signal transduction and ion signaling.

Keywords: formal reaction kinetics, signal transduction, modeling, *Mathematica*

* **Corresponding author** : INRA, Unité de Phytopharmacie et Médiateurs chimiques,
F-78026 Versailles Cedex, France. Tel.: +33 1 30 83 33 55. Fax: +33 1 30 83 31 19.
E-mail: rospars@versailles.inra.fr.

1. Introduction

The theoretical study of reaction kinetics has led to the development of powerful concepts and methods that can be of much help for investigating dynamical models of biochemical systems. Most problems in biochemistry can be solved by going through routine steps that are almost independent of the specific system investigated. This situation prompted us to try and fill the gap between theory and practical needs. The aims of this paper are to present in a concise way the routine steps involved and to propose practical solutions based on present day mathematical program packages. Although the routine steps may involve hard or complicated analytical calculations, both exact and approximate, and also extensive numerical investigations, these packages can help shorten the time needed to do the calculations and to improve their exactness.

For introducing some of these possibilities we have selected three examples of increasing complexity taken from the field of receptor-mediated cell responses. Examples I and II, based partly on our own practice (e.g. [21], [28], [29]), are about the first two steps of cell transduction and involve only one or two reaction steps. The third one, about calcium oscillations, involves seven reaction steps with feedback relationships [24]. Besides describing chemical reactions these examples involve translocations (Example III), and illustrate several properties of chemical dynamic systems, including the tendency to stationary (Examples I and II) as well as oscillatory (Example III) behaviors. More technical aspects are also presented, such as the description of reactions that, contrary to most steps in all examples, do not obey the mass-action law (some steps in Example III), as well as the solution of normal and stiff differential equations, the numerical handling of the latter presenting difficulties. Basic notions of reaction kinetics such as e.g. stationary point, steady state, equilibrium, stable equilibrium, detailed balanced equilibrium, or molecularity and order of a reaction step, or solution, trajectory and selectivity curves of a model, are defined. We show how the usual questions addressed when studying such systems can be solved with relatively short programs which, in the present paper, are written in Mathematica.

These examples and programs are intended to be applicable to as wide a range of systems as possible. They are easy to generalize. In neurochemistry, for example, the tools described can be applied to describe the kinetics of systems as diverse as ion channels (voltage dependent or ligand gated, see [14]), transmitter release, or gating of postsynaptic receptors (see [6]).

There are three main logical *steps of investigations*. (i) The first step consists in establishing a set of chemical reactions which summarizes the knowledge of the

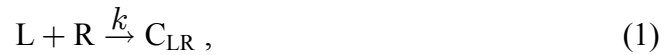
biochemical system studied, including the speed of transition along each arrow (reaction or translocation). Three examples are presented in section 2 which are analyzed in the following sections. (ii) The second step consists in translating the system of reactions into rate equations, i.e. as a set of differential equations, one for each time-dependent quantity in the system. This is treated in section 3. (iii) Finally these equations must be solved to reveal the time evolution of the system or to study the solutions from other points of view (non-negativity, monotonicity, oscillations, coexistence of chemical species etc.). This third step is described in section 4. Details on the programming aspects with extensions for the qualitative and stochastic studies of reactions can be found in [34].

2. Examples of neurochemical reactions

Three example systems taken from the field of cell signal transduction are presented.

2.1. Example I

The first step in cell signal transduction consists in the binding of a ligand L, e.g. photon in photoreceptor cells, odorant molecule in olfactory receptor cells or neurotransmitter in interneurons, to a receptor protein borne by the sensory or postsynaptic membrane. It can be modeled by the reaction



where C_{LR} stands for the ligand-receptor complex. This reaction expresses the fact that one "molecule" of the ligand L binds one molecule of the receptor R and an activated form C_{LR} of the receptor is formed. The positive real number k is the *reaction rate coefficient* which, in this case, is a constant that characterizes the velocity of the reaction (see section 3.2.1). Reaction (1) assumes that the reaction step is *irreversible*, i.e. there is no way (at least in the model) for the activated receptor to return to its original state. As a consequence such a model describes only the initial response of the system [19], [20]. A reversible reaction might be also considered as shown below.

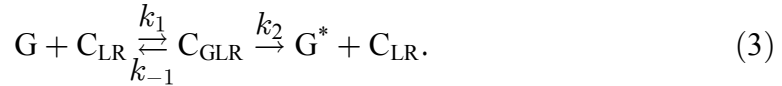
2.2. Example II

The next step of the signal transduction cascade can serve as our second example. It consists in the interaction within the cell membrane of the activated complex C_{LR} with G-proteins, denoted by G, and results in the activation of the G-protein as G^* according to the reaction scheme

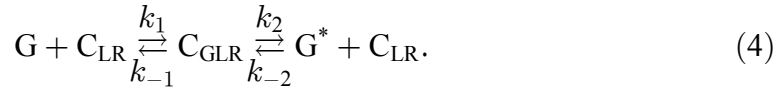


A noteworthy feature of (2) is that C_{LR} is regained at the end of the process, which means that activation according to (1) can lead to the production of several activated proteins G^* . Model (2) consists of two reaction steps. In the first step (2a) one molecule (or one active site) of the G-protein binds one molecule of the activated receptor C_{LR} . The complex produced is dissociated in second step (2b) into the activated receptor and the activated G-protein, the latter taking part in the subsequent steps of the cascade. It can be assumed that the concentration of C_{LR} is constant in this reaction. This constancy means that the stimulus is delivered as a short impulse, that the number of receptors is much larger than that of ligand L reaching the membrane and that the lifetime of C_{LR} is relatively long. In formal kinetics C_{LR} is said to be an *external species*.

It may also be worth considering variations of model (2) where the first reaction is reversible:



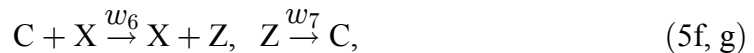
or both reactions are reversible



The above models are essentially of the Michaelis-Menten type [17] with C_{LR} playing the role of the enzyme and G that of the substrate. From a quantitative point of view a difference arises: in enzyme kinetics the usual assumption is that the number of enzyme molecules is relatively low; here this is just the opposite. As in example I the reaction rate coefficients are constants.

2.3. Example III

As a third example we consider the model of receptor-stimulated calcium spiking proposed by Meyer and Stryer [24]. It can be summarized in 7 reactions



in which X stands for the cytosolic Ca^{2+} , Y for the Ca^{2+} in the endoplasmic reticulum (ER) and Z for the inositol-1,4,5-trisphosphate (INP_3), whereas A , B and C stand for

external species (see above) acting as sources (or sinks) for X and Z . The w_r are *functions* characterizing the rate of each reaction r ; they generalize the reaction rate *constants* k_r of Examples I and II and will be formally introduced in section 3.2.2.

The reactions can be arranged in 3 groups. The first group describes the exchanges of Ca^{2+} between the ER and the cytosol: reaction (5a) represents the flux of calcium out of the ER through channels in the ER membrane, which is controlled by INP_3 , and (5b) the ATP-dependent transport of calcium into the ER, whereas (5c) summarizes the fact that the capacity of the ER is limited. The second group introduces the source (A) and sink (B) of Ca^{2+} with reaction (5d) describing the influx of calcium into the cell and (5e) its uptake by mitochondria (sequestered calcium B). The third group describes the triggering mechanism, reaction (5f) representing the phospholipase-dependent synthesis of INP_3 from its precursor C (PIP_2), and reaction (5g) the destruction of INP_3 . Reaction (5f) is controlled by vasopressin, an hormone acting as ligand L on a receptor R (see Example I), presumably via the activated G-protein G^* of Example II, and the cytosolic calcium level. The behavior of this system is more complex than that of the two previous examples, not only because it involves more reactions, but because more of these reactions are interrelated, INP_3 and cytosolic calcium acting as crosscoupled messengers. These couplings are taken into account through the functions w_i as explained below.

3. From chemical reactions to evolution equations

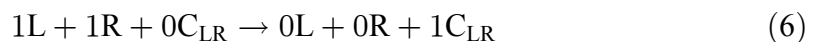
3.1. Formal chemical reactions

3.1.1. Examples of formal reactions

A formal reaction is a standard translation of the actual chemical reaction in which all chemical species not considered as external appear on both sides of the formula and are multiplied with coefficients 1 or 0 according to their presence or absence in the actual reaction; coefficients greater than 1 can also occur, see e.g. reaction (5c), but rarely. These coefficients play an important role in the modeling. An introduction to the topics can be found in [27]. A more mathematical text with many biological applications is given in [7].

Example I

The formal reaction corresponding to reaction (1) writes:



The formal linear combination on the left side of the reaction step is called the *reactant complex* and that on the right side of the reaction step the *product complex* (the meaning of "complex" here must not be confused with that in "receptor-ligand complex"). The coefficients standing in front of the chemical species are the *stoichiometric coefficients*. The stoichiometric coefficient of C_{LR} in the reactant complex is zero because it is not present on the left side of reaction (1), whereas it is 1 in the product complex. The sum of the stoichiometric coefficients in the reactant complex is the (overall) *molecularity* of the reaction. Because the molecularity of reaction (1) is 2, it is said to be a *second order* reaction.

Example II

The detailed form of step (2b) is



Here we neglect C_{LR} because it is an external species. To put it another way (2b) is a *genuine* reaction step with external species included, whereas (7) is the corresponding *formal* reaction. The *molecularity* of the reaction step is 1, so (7) is a *first order* formal reaction step. The genuine reaction being of higher order in general, the simplicity of the system under investigation strongly depends on the determination of the species that can be considered as external.

Example III

The same rules apply. Note that the formal reaction step corresponding to the fourth step (5d), $A \rightarrow X$, is $0X \rightarrow 1X$. Therefore this is a *zeroth order* step.

3.1.2. Generalization

Formal reactions, like (6) and (7) can be generalized in the following way

$$\sum_{m=1}^M \alpha_{m,r} X_m \xrightarrow{k_r} \sum_{m=1}^M \beta_{m,r} X_m, \quad (r = 1, 2, \dots, R) \quad (8)$$

where $M \in \mathbb{N}$ is the number of chemical species X_1, X_2, \dots, X_M and $R \in \mathbb{N}$ is the number of reaction steps, $\alpha_{m,r}$ and $\beta_{m,r} \in \mathbb{N}_0$ are the stoichiometric coefficients and $k_r \in \mathbb{R}^+$ are the reaction rate coefficients.

In this setting a reversible reaction $I \rightleftharpoons J$ is represented as an irreversible reaction-antireaction pair, $I \rightarrow J$ and $J \rightarrow I$. This is the *canonical* form of a reversible reaction step. Realistic detailed chemical models are usually required to consist of reversible steps. This is not the case of the examples treated here, which give only a phenomenological description of processes.

3.1.3. Associated calculations

The transformation into canonical form and the determination of the stoichiometric coefficients can be done automatically with two programs. The first one (**ToCanonical**, Table 2) rewrites reversible reactions as a set of two irreversible reactions. This standard form is interpreted by the second program (**ToStoichiometry**, Table 3) which returns as output the number of chemical species M , the number of reaction steps R , the list of species involved in a canonical (lexicographic) ordering, and the matrices of the stoichiometric coefficients α and β of the reactant and product complexes respectively. Let us present these programs on the examples.

Example I

In this case only the second program is needed because the reaction is already written in its canonical form.

```
stoiI = ToStoichiometry[{L+R → CLR}]
```

returns variable **stoiI** containing the list

$$\begin{aligned} \{ \text{species} \rightarrow \{L, R, C_{LR}\}, M \rightarrow 3, \text{reactionSteps} \rightarrow \{L+R \rightarrow C_{LR}\}, \\ R \rightarrow 1, \alpha \rightarrow \{\{1\}, \{1\}, \{0\}\}, \beta \rightarrow \{\{0\}, \{0\}, \{1\}\}, \\ \text{variables} \rightarrow \{l, r, c_{LR}\} \} . \end{aligned} \quad (9)$$

These results can be compared to (6). The list of species is automatically ordered by the program, α and β correspond to this new order. In *Mathematica* the arrow above only expresses replacement rules and has nothing to do with limits or reaction arrows, i.e. α should be replaced in calculations with the expression given above. Any part of the results can be extracted and used in further calculations, e.g. $\alpha/.stoiI$ extracts α from variable **stoiI**.

Example II

Usually reactions are written in the form of chains as in (2), (3) or (4). This is different from the canonical form (8) which is used by both the mathematical theory and most of our programs. Thus, except when the natural and canonical forms are identical (as in Example I) a preparatory program is needed to transform the natural form into the canonical one. For reaction (2) we can write

```
ToCanonical[{G+CLR → CGLR → G*+CLR}]
```

which returns

$$\{G+C_{LR} \rightarrow C_{GLR}, C_{GLR} \rightarrow C_{LR}+G^*\}$$

The program can process reversible reactions which are indicated with the double-head arrow \leftrightarrow . For example, introducing one reversible step, we have

$$\mathbf{canoII} = \mathbf{ToCanonical}[\{G+C_{LR} \leftrightarrow C_{GLR} \rightarrow G^*+C_{LR}\}]$$

which returns variable **canoII** containing

$$\{G+C_{LR} \rightarrow C_{GLR}, C_{GLR} \rightarrow G+C_{LR}, C_{GLR} \rightarrow C_{LR}+G^*\}$$

Similarly with two reversible steps one gets

$$\begin{aligned} &\mathbf{ToCanonical}[\{G+C_{LR} \leftrightarrow C_{GLR} \leftrightarrow G^*+C_{LR}\}] \\ &\{G+C_{LR} \rightarrow C_{GLR}, C_{GLR} \rightarrow G+C_{LR}, C_{GLR} \rightarrow C_{LR}+G^*, C_{LR}+G^* \rightarrow C_{GLR}\} \end{aligned}$$

Then we can apply the second program to reaction (2). For example

$$\mathbf{stoiIII} = \mathbf{ToStoichiometry}[\mathbf{canoII}, \{C_{LR}\}]$$

where the external species is indicated as the last (optional) argument. It yields

$$\begin{aligned} &\{\text{species} \rightarrow \{G, C_{GLR}, G^*\}, M \rightarrow 3, \\ &\text{reactionSteps} \rightarrow \{G \rightarrow C_{GLR}, C_{GLR} \rightarrow G, C_{GLR} \rightarrow G^*\}, R \rightarrow 3, \\ &\alpha \rightarrow \{\{1, 0, 0\}, \{0, 1, 1\}, \{0, 0, 0\}\}, \beta \rightarrow \{\{0, 1, 0\}, \{1, 0, 0\}, \{0, 0, 1\}\}, \\ &\text{variables} \rightarrow \{g, c_{GLR}, g^*\} \end{aligned} \quad (10)$$

of which α and β can be formatted to give

$$\alpha = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \beta = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Matrix α gives the stoichiometric coefficients of the reactant complexes and matrix β those of the product complexes. In these matrices the rows correspond to the M

species and the columns to the R reaction steps. Note that the sum of the coefficients in each column of α gives the order of the reaction step.

Example III

In the calcium spiking model we first write

```
canoIII = ToCanonical[{Y+Z → X+Z, X → Y,  
2Y → X+Y, A → X → B, C+X → X+Z, Z → D}]
```

```
{Y+Z → X+Z, X → Y, 2Y → X+Y, A → X, X → B, C+X → X+Z, Z → D}
```

(because of the lexical ordering of the reactions the output order may be different from the input order), then

```
stoiIII = ToStoichiometry[canoIII, {A, B, C, D}]
```

```
{species → {X, Y, Z}, M → 3, reactionSteps →  
{0 → X, X → 0, X → Y, X → X+Z, 2Y → X+Y, Z → 0, Y+Z → X+Z}, R → 7,  
α → {{0, 1, 1, 1, 0, 0, 0}, {0, 0, 0, 0, 2, 0, 1}, {0, 0, 0, 0, 0, 1, 1}},  
β → {{1, 0, 0, 1, 1, 0, 1}, {0, 0, 1, 0, 1, 0, 0}, {0, 0, 0, 1, 0, 0, 1}},  
variables → {x, y, z}} (11)
```

In the latter program the external species are again indicated as the last argument. In this example there are $M = 3$ time-dependent quantities x , y , z and $R = 7$ steps (reactions and translocations), that is why the stoichiometric coefficients α and β are 3×7 matrices.

3.2. Evolution equations

3.2.1. Examples of evolution equations

What does interest us is the temporal evolution of the concentrations of L , R and C_{LR} in the first example, of G , C_{GLR} and G^* (C_{LR} is assumed to be constant) in the second one, and of X , Y , Z in the third one. The concentrations of the individual (formal chemical) species are usually denoted by the same letters in italicized lowercase form. The concentration can be expressed in molarity or mass, with respect to a fixed mass, volume or area. The units of the reaction rate coefficients must be consistent with

those of the concentrations. For the sake of simplicity, in the rest of the paper we will consider concentrations expressed in moles per unit volume.

Example I

The usual system of evolution equations for reaction (1) is:

$$\dot{l}(t) = -kl(t)r(t) \quad \dot{r}(t) = -kl(t)r(t) \quad \dot{c}_{LR}(t) = kl(t)r(t), \quad (12)$$

where k is the reaction rate coefficient, $l(t)$, $r(t)$ and $c_{LR}(t)$ denote the concentrations at time t of the species L, R and C_{LR} respectively, and $\dot{l}(t)$, $\dot{r}(t)$ and $\dot{c}_{LR}(t)$ denote their derivatives with respect to time.

The differential equations (12) are based on the assumption that the individual reactions proceed according to the (*kinetic*) *mass-action law* (**Assumption 1**). It means that the velocity of reaction (1) is the product of the quantities of the reactants $l(t)$ and $r(t)$ multiplied by the reaction rate coefficient. One can also say that here $l(t)$ and $r(t)$ are raised to the power 1, because this is the value of the stoichiometric coefficients of L and R in the reactant complex.

The signs in the equations reflect the net changes of the corresponding species. The net changes of L, R and C_{LR} are $0 - 1$, $0 - 1$ and $1 - 0$ respectively, which means that L and R are destroyed in the reaction whereas C_{LR} is created. (This seemingly cumbersome notation is justified by the fact that writing -1 , -1 , 1 would entail a loss of information because 1 results also from e.g. $2 - 1$.)

Example II

In the case of reaction (3) the system of evolution equations is:

$$\dot{g}(t) = -k_1g(t) + k_{-1}c_{GLR}(t) \quad (13a)$$

$$\dot{c}_{GLR}(t) = k_1g(t) - (k_{-1} + k_2)c_{GLR}(t) \quad (13b)$$

$$\dot{g}^*(t) = k_2c_{GLR}(t) \quad (13c)$$

$$g(0) = g_0 \quad c_{GLR}(0) = g^*(0) = 0, \quad (14)$$

where $g(t)$, $c_{GLR}(t)$ and $g^*(t)$ denote the concentrations at time t of the species G, C_{GLR} and G^* respectively. The positive real number k_{-1} and k_2 are the reaction rate coefficients characteristic of the velocity of the corresponding reaction steps, while the positive real number k_1 is actually the product of a *genuine rate coefficient* κ_1 and the (assumed to be constant) concentration of C_{LR} , $k_1 = \kappa_1 c_{LR}$. So, one has to be careful because the units depend on the stoichiometric coefficients of the species: in a first order reaction the unit of a reaction rate coefficient (reciprocal of time unit) is

different from that of a second order one (reciprocal of the product of time and concentration units). Here, it is assumed in the kinetic differential equations (13) that the effects of different reaction steps are additive. This is a fundamental assumption of reaction kinetics (**Assumption 2**).

The essential *similarity* between (12) and (13) is that the rates of the steps are multiplied by the net changes of the species involved. The net changes of G , C_{GLR} , G^* in reaction (2a) are $0 - 1$, $1 - 0$ and $0 - 0$, and the net changes of G , C_{GLR} , G^* in (2b) are $0 - 0$, $0 - 1$ and $1 - 0$. Note that there is no equation for the quantity of the external species C_{LR} . The essential *difference* between (12) and (13)–(14) is that (12) is a mere differential equation, whereas (13)–(14) is an *initial value* or *Cauchy problem*, i. e. it also contains restrictions (14) on the initial values of the variables. These restrictions imply that the solution of (13)–(14) is unique whereas that of (12) is not.

We simplified reaction (3) assuming that the concentration of C_{LR} is constant. Another (often used) approximation consists in assuming that the rate of change of the concentration of C_{GLR} is close to zero. First let us write down the full system of equations:

$$\dot{g}(t) = -k_1 g(t) c_{\text{LR}}(t) + k_{-1} c_{\text{GLR}}(t) \quad (15a)$$

$$\dot{c}_{\text{LR}}(t) = -k_1 g(t) c_{\text{LR}}(t) + (k_{-1} + k_2) c_{\text{GLR}}(t) \quad (15b)$$

$$\dot{c}_{\text{GLR}}(t) = k_1 g(t) c_{\text{LR}}(t) - (k_{-1} + k_2) c_{\text{GLR}}(t) \quad (15c)$$

$$\dot{g}^*(t) = k_2 c_{\text{GLR}}(t) . \quad (15d)$$

Then, using the conservation relation,

$$c_{\text{LR}}(t) + c_{\text{GLR}}(t) = c_{\text{LR}}(0) + c_{\text{GLR}}(0) = c_{\text{LR}}(0) \quad (16)$$

which is a mathematical consequence of (15b) and (15c) and *not* a physical assumption imposed from outside, we get

$$\dot{c}_{\text{GLR}}(t) = k_1 g(t) (c_{\text{LR}}(0) - c_{\text{GLR}}(t)) - (k_{-1} + k_2) c_{\text{GLR}}(t) . \quad (17)$$

Assuming that the time derivative of c_{GLR} is zero, c_{GLR} can be expressed as

$$c_{\text{GLR}}(t) = \frac{g(t) c_{\text{LR}}(0)}{g(t) + K} \quad \text{where } K = \frac{k_{-1} + k_2}{k_1} . \quad (18)$$

Inserting this expression into (15d) a rational function is obtained as the rate of change of g^* :

$$\dot{g}^*(t) = k_2 \frac{g(t)c_{LR}(0)}{g(t) + K}. \quad (19)$$

Although this procedure seems to be not well-founded from a mathematical point of view, it can be supported by the theory of singular perturbation, see references in [7], [17], [37] or [41]. We do not investigate this model further; it is intended as an introduction to the line of thought which lies behind the reaction rates of the rational type found in the next example.

Example III

The kinetic mass-action law of Assumption 1 is verified experimentally in many cases but it is far from being a general law of Nature. Example III offers a good illustration of this statement because many of the kinetic differential equations found by Meyer and Stryer [24] do not follow this law. These equations are:

$$\dot{x}(t) = c_1 \left(\frac{z(t)}{K_1 + z(t)} \right)^3 y(t) - c_2 \left(\frac{x(t)}{x(t) + K_2} \right)^2 + c_3 y(t)^2 + c_4 - c_4 \left(\frac{x(t)}{c_5} \right)^{3.3} \quad (20)$$

$$\dot{y}(t) = -c_1 \left(\frac{z(t)}{K_1 + z(t)} \right)^3 y(t) + c_2 \left(\frac{x(t)}{x(t) + K_2} \right)^2 - c_3 y(t)^2 \quad (21)$$

$$\dot{z}(t) = c_6 \rho \frac{x(t)}{x(t) + K_3} - c_7 z(t) \quad (22)$$

Eq. (20) gives the rate of change of the cytosolic concentration of Ca^{2+} with the first three terms (reaction rates) describing the cytosol-ER exchanges (reactions (5a)-(5c)), and the last two the source and sink (reactions (5d), (5e)). Eq. (21) gives the rate of change of the endoplasmic concentration of Ca^{2+} which is merely the reverse of (5a)-(5c). Eq. (22) yields the rate of change of the INP_3 with the two reaction rates corresponding respectively to its synthesis (5f) and destruction (5g). Parameter ρ in (22) expresses the relative level of activation of phospholipase C which ultimately depends on vasopressin concentration. The functions describing each reaction and their constants (Table 1) were obtained mostly from experimental measurements.

It can be seen here that some of the rates of the individual steps are of the mass-action type (products in components 3, 4 and 7, see Table 1) but the others are not (ratios or non-integer exponents in components 1, 2, 5 and 6, Table 1; components 1, 2 and 5 being of Michaelis–Menten type). However, the reaction rates are again multiplied by the net changes (± 1) of the corresponding species, so Assumption 2 still holds. The components of the rate $w(x, y, z)$ of the individual steps at the concentration vector (x, y, z) and the corresponding multipliers are shown in Table 1.

Table 1 approximately here

The reactions that do not obey the mass-action law are usually considered to be less fundamental than those that obey it. They can be generally interpreted as the result of lumping of several (unknown) reactions or species into a single one (that used in the model) [32]. They can be also obtained by fitting to an empirical function. Nevertheless these non-mass-action rates are kinetic rates in the sense that they are positive, continuous, monotonous functions of the corresponding concentrations [39] or [31].

Formal kinetics provides a framework to treat in a unified way chemical reactions and some transport phenomena under certain circumstances, and this is also exemplified here. The second step $X \rightarrow Y$ expresses the belief that translocation of Ca^{2+} between the cytosol and the endoplasmic reticulum proceeds as if it were a first order transformation between two different species. This **Assumption 3** enlarges the modeling capacity of formal reaction kinetics and is quite common [15].

3.2.2. Formal presentation

If the concentration of species X_m at time $t \in \mathbb{R}_0^+$ is denoted by $x_m(t)$, then the usual kinetic differential equation (assuming mass-action type kinetics for the time evolution of these quantities) is the following (polynomial autonomous) ordinary differential equation with the same notations as in (8) :

$$\dot{x}_m(t) = \sum_{r=1}^R (\beta_{m,r} - \alpha_{m,r}) k_r \prod_{p=1}^M x_p(t)^{\alpha_{p,r}}, \quad (m = 1, 2, \dots, M) \quad (23)$$

or in vectorial form

$$\dot{\mathbf{x}}(t) = (\boldsymbol{\beta} - \boldsymbol{\alpha}).(\mathbf{k} \otimes \mathbf{x}(t)^\alpha), \quad (24)$$

where \otimes denotes componentwise product of vectors [15].

In the case when the reaction rates are not of the mass-action type the r th reaction step has a reaction rate $w_r(\mathbf{x}(t))$, if the vector of concentrations at time t is $\mathbf{x}(t)$. Then the form of the kinetic differential equation (still corresponding to Assumption 2) is:

$$\dot{x}_m(t) = \sum_{r=1}^R (\beta_{m,r} - \alpha_{m,r}) w_r(\mathbf{x}(t)). \quad (m = 1, 2, \dots, M) \quad (25)$$

or, in matrix notation

$$\dot{\mathbf{x}}(t) = (\boldsymbol{\beta} - \boldsymbol{\alpha}).\mathbf{w}(\mathbf{x}(t)). \quad (26)$$

In order for Eq. (25) to have a kinetic meaning further regularity conditions (mentioned above) are required for the functions w_r .

3.2.3. Calculation of the relevant quantities

The procedure summarized in section 3.2.2 can be fully automated when the mass-action law can be applied, and partially when it cannot. Program **RHS** (see Table 4) calculates the right hand side of a kinetic differential equation if the matrices α , β provided by the program **ToStoichiometry** (see section 3.1.3) and the vector of reaction rate coefficients k are given. **RHS** works no matter what is the number of chemical species and elementary steps, and is thus very helpful for solving models involving many species and steps. In the case of a mass-action type kinetics **RHS** displays each reaction step and asks the corresponding rate constant one after the other. In the case of non-mass-action type kinetics it asks the rates w themselves.

The mass-action type systems of kinetic differential equations are *polynomial differential equations*: usually more complicated than linear equations, still tractable enough from many points of view. This feature is unessential for simple models as those of Examples I and II, but it becomes essential for introducing mass-action kinetics in full generality. Moreover, these are the arguments beneath the program **RHS**. Let us present this program on the examples.

Example I

The stoichiometric coefficients α and β were given by the program **ToStoichiometry** and stored in **stoiI**. Then

```
rhsI = RHS[stoiI]
```

returns the right-hand sides of the differential equations

$$\{-k_{lr}, -k_{lr}, k_{lr}\}$$

These results are those given in equations (12) provided we give the rate coefficients or the rates of the reaction steps in the corresponding order.

Example II

Again, using the coefficients given by the function **ToStoichiometry** we can write

```
rhsII = RHS[stoiII]
```

and get

$$\{ -g k_1 + k_{-1} c_{\text{GLR}}, g k_1 - k_2 c_{\text{GLR}} - k_{-1} c_{\text{GLR}}, k_2 c_{\text{GLR}} \}$$

These results are those given in equations (13).

Example III

In this non mass-action type example, the reaction rates $w(x, y, z)$ at the concentration vector (x, y, z) given in Table 1 (second column) must be used as components of the vector w and the (last) optional argument must be set to false.

```
rhsIII = RHS[stoiIII, MassAction → False]
```

We do not reproduce the results as they are equivalent to the right-hand sides of eqs. (20)–(22).

4. From evolution equations to system states

Typically dynamic systems present stationary, transient non-periodical, oscillatory or chaotic trajectories. The three examples chosen allow one to illustrate the first three of these behaviors and to present tools to decide which behavior is displayed by a given system. Having investigated the problem of stationary states (section 4.1) we will consider the evolution in time of the system which is a more complicated problem (section 4.2).

4.1. Stationary states

We consider first the existence of solutions that do not change in time, then we examine if such solutions are stable or not.

4.1.1. Calculation of the stationary points

The (seemingly) simplest question about a kinetic differential equation is: which are the constant solutions of this equation? Constant solutions are interpreted as stationary (or in some cases, equilibrium) solutions. They describe states where the system stays forever if undisturbed. To find these states means solving three problems. First, the system of differential equations must be solved when their right hand sides are set equal to zero, which expresses that the velocities of formation of all species are zero.

Second, the *law of mass conservation* must also hold for the stationary points. Third, it is also necessary to eliminate non-physical solutions with negative values at this step. The properties that a concentration vector (i) makes the species formation vector zero, (ii) obeys the law of mass conservation and (iii) is nonnegative, are independent, as can easily be shown by simple examples.

The program **StationaryPoints** (fourth in Table 4) calculates the stationary solutions using their first two properties but not the third, i.e. it gives constant solutions of the kinetic differential equations obeying the mass conservation relations that may happen to be negative. Its input arguments are the stoichiometric matrices, the names of the variables and their initial values. Internally it calls sequentially two programs, **Kinetic** and **MassConservation** that can be also used separately.

The program, **Kinetic** (second in Table 4), calculates the zeros of the right hand side of a kinetic differential equation given the matrices of stoichiometric coefficients, the vector of reaction rate coefficients and the names of the variables to be used. The calculations depend also on the type of rates (mass-action or not). Results are presented in table form. It is more appropriate to use **Reduce** than **Solve**, because the latter would only give the coarse solutions, those which do not take into account special values of the parameters (like zero); e.g. for eq. $ax + b = 0$, the coarse solution is $x = -b/a$, whereas the fine one consists of three possibilities (i) $x = -b/a$ for $a \neq 0$, (ii) x undetermined for $a = 0$, $b = 0$, (iii) no solution otherwise ($a = 0$, $b \neq 0$). **Solve** and **Reduce** are based on the algorithm for constructing Gröbner basis, which is one of the most important results in the last decades in theoretical computer science. This algorithm enables one to rewrite a polynomial system of equations into a triangular form, in almost the same way as it is done in the process of Gauss elimination for linear systems of equations (Buchberger, 1970). (Triangular here means that the first equation only contains a single variable, the second one only two etc.) Then such equations can be solved, sometimes symbolically and almost always numerically. This procedure can also be extended to the case when the rates are rational functions.

The other program, **MassConservation** (third in Table 4), finds conditions to be fulfilled in order that the law of mass conservation be valid. It needs the stoichiometric matrices as input, the names of the variables to be used and the initial values of the variables.

As to the third problem, that of eliminating the non-physical solutions, the package **Algebra** of *Mathematica* might help decide whether a candidate stationary solution has non-negative coordinates. The programs in this package are designed to solve polynomial inequalities in any dimensions based on new theoretical developments. In

practice, a faster answer can be obtained if the parameters are given numerical values; then the number of non-numerical parameters can be gradually increased.

Example I

The stationary solutions are given by **StationaryPoints** $[\alpha/.stoiI, \beta/.stoiI, \{k\}, \{l, r, c_{LR}\}, \{l_0, r_0, c_0\}]$ which yields

$$\left\{ \left\{ c_{LR} \rightarrow c_0 + l_0, r \rightarrow -l_0 + r_0, l \rightarrow 0 \right\}, \right. \\ \left. \left\{ c_{LR} \rightarrow c_0 + r_0, r \rightarrow 0, l \rightarrow l_0 - r_0 \right\} \right\}$$

These results give the coordinates c_{LR}, r, l of the stationary points, either as $c_0 + l_0, r_0 - l_0, 0$ or as $c_0 + r_0, 0, -r_0 + l_0$. The first one is accepted if $l_0 \leq r_0$, and the second if $l_0 \geq r_0$, (both are accepted and identical if $l_0 = r_0$) because these are the conditions assuring non-negativity.

For the null velocities, it is enough to consider one equation from the set of three derived from (5)

$$0 = -klr \quad 0 = -klr \quad 0 = klr \quad (27)$$

The first solution is $r = 0$ with arbitrary concentrations of the two other species, and the second solution is $l = 0$ with arbitrary concentrations c_{LR} and r . The species formation rates are equal to zero if and only if concentration vectors are of these types. The same result is given with program **Kinetic** $[\alpha/.stoiI, \beta/.stoiI, \{k\}, \{l, r, c_{LR}\}]$ which returns

$$\left\{ \{l \rightarrow 0\}, \{r \rightarrow 0\} \right\}$$

The conservation of mass can be written:

$$c_{LR}(t) + l(t) = c_{LR}(0) + l(0), c_{LR}(t) + r(t) = c_{LR}(0) + r(0) \quad (28)$$

Program **MassConservation** $[\alpha/.stoiI, \beta/.stoiI, \{l, r, c_{LR}\}, \{l_0, r_0, c_0\}]$ gives

$$\left\{ \{l \rightarrow c_0 - c_{LR} + l_0, r \rightarrow c_0 - c_{LR} + r_0\} \right\}$$

which is equivalent to (28).

Example II

Of the three equations derived from (13)

$$0 = -k_1 g + k_{-1} c_{\text{GLR}} \quad 0 = k_1 g - (k_{-1} + k_2) c_{\text{GLR}} \quad 0 = k_2 c_{\text{GLR}} \quad (29)$$

only two are independent. **Kinetic** $[\alpha/.stoiii, \beta/.stoiii, \{k_1, k_{-1}, k_2\}, \{g, c_{\text{GLR}}, g^*\}]$ gives

$$\{\{g \rightarrow 0, c_{\text{GLR}} \rightarrow 0\}\}$$

which means that there is only one solution, the first two components of which are zero and the third one, G^* , can be arbitrary.

The conservation of mass is

$$c_{\text{GLR}}(t) + g(t) + g^*(t) = c_{\text{GLR}}(0) + g(0) + g^*(0). \quad (30)$$

There is no mass conservation equation for the quantity of the receptors because of our assumptions above. Program **MassConservation** $[\alpha/.stoiii, \beta/.stoiii, \{g, c_{\text{GLR}}, g^*\}, \{g_0, c_0, g_0^*\}]$ returns (30) in a different form

$$\{\{g \rightarrow c_0 - c_{\text{GLR}} + g_0 - g^* + g_0^*\}\}$$

Let us apply now the specific initial conditions (14), i.e. no GLR-complex and no activated G-protein are present at time 0, to calculate the stationary points.

StationaryPoints $[\alpha/.stoii, \beta/.stoii, \{k_1, k_{-1}, k_2\}, \{g, c_{\text{GLR}}, g^*\}, \{g_0, 0, 0\}]$

$$\{\{g^* \rightarrow g_0, g \rightarrow 0, c_{\text{GLR}} \rightarrow 0\}\}$$

This expresses that all G-proteins are activated but does not mean that this state of the system can be reached (see below).

Example III

If one tries to determine the stationary point(s) of this example by the program above then no result will probably be obtained within a reasonable time. High school tricks might help *Mathematica* do the calculations in this specific case leading to a result as long as 5 pages. Finding all the solutions of such an equation numerically is a very hard task. There are general methods to obtain all the roots of multivariate polynomials and these are used by the built-in **NSolve** of *Mathematica*, where **N** stands for numerical. However, our system of equations cannot be transformed into a polynomial system of equations, not because of the presence of rational functions but because of the presence of the non-integer exponent 3.3. To treat such cases one has to

turn to specialized programs written to find roots of nonlinear systems. Such an additional package has also been written for *Mathematica* (**Statistics`NonLinearFit`**). In some cases one may use the ideas developed for this special situation and choose a simpler way as follows.

(1) Modify the exponent 3.3 to 3. In the present case it will not change the stationary point of the system (when calculating the stationary point use the third equation to simplify the first one; the reduced first equation $c_4(1 - (\frac{x(t)}{c_5})^{3.3}) = 0$ obviously has a solution independent of the exponent in question). This gives **wmodified** as a result.

(2) Apply **Kinetic**, giving the zeros of the right-hand side, to the modified equations as follows

Table 5 approximately here

```
Kinetic[ $\alpha$ , $\beta$ ,wmodified/.meyerconst,variables,  
MassAction → False]
```

where **meyerconst** is the actual value of all the constants occurring in the rates shown in Table 5. The result will be a table containing 6 solutions, most of them being complex ones, only one is real positive. It is

x	y	z
0.6	5.63422	0.07875

This is the solution we are looking for because of the specific structure of the equations. As can be checked numerically using the program **NDeterministic** below, a solution starting from this point remains there.

(3) This solution can be used as an initial estimate given to the built-in function **FindRoot** as follows

```
spIII=FindRoot[rhsIII/.meyerconst,  
{x,0.6},{y,5.63},{z,0.079}]
```

which ends with the same result as before (in general this will not be the case). The same result is obtained if we put the values of the constants into the 5-page long symbolic result mentioned above.

Finally, mass conservation implies no restrictions on the stationary points as `MassConservation[$\alpha/.stoiIII,\beta/.stoiIII,\{x,y,z\},\{x_0,y_0,z_0\}]$` returns `{ {} }` which means that the set of restrictions is empty.

4.1.2. Linear stability analysis

So, solutions which do not change with time can be found. This is not sufficient however because these solutions are based on the assumption that no perturbations are present, which is not realistic in practice. Some constant solutions change drastically under small perturbations, and therefore have no physical significance. Other constant solutions are significant, because they resist small disturbances and are consequently stable. A simple way (among many) to decide whether a stationary point is stable is *linear stability analysis* [1], [10]. This is what we do here automatically.

Even if a stationary point is stable it may not obey the law (sometimes called Principle) of Detailed Balance (or the principle of microscopic reversibility) which is thought to be very important in chemical thermodynamics [22]. The content of this law is that at equilibrium all the individual subprocesses are equilibrated, which in the case of chemical reactions means that the reaction steps should be reversible and should have the same reaction rate in both directions. Obviously this property can only hold if all the elementary steps are reversible which is often not the case for *formal* models (as opposed to models expressing detailed chemistry) used here. In order to check this property for reversible reactions one might use the necessary and sufficient condition provided by Feinberg [11].

The essence of linear stability analysis is that one takes the linear approximation of the right hand side (the Jacobian Matrix) and evaluates it at the stationary points. If all the eigenvalues of this matrix have negative real part then the stationary point is asymptotically stable; if there is at least one eigenvalue with positive real part then the stationary point is unstable. Further investigations are needed if some of the real parts are negative and some are zeros. Program **LinearizedEigenvalues** (sixth in Table 4) calculates the eigenvalues of the linearized equation.

Example

`LinearizedEigenvalues[rhsI,variables/.stoiI,spI]//Simplify`
returns

$$\{\{0, 0, k (r_0 - l_0)\}, \{0, 0, k (-r_0 + l_0)\}\},$$

which shows that either the first or the second stationary point is unstable, depending on whether $r_0 > l_0$ or $r_0 < l_0$. The case when they are equal needs further

investigation (cf. [1]). If all the eigenvalues were negative, the stationary points would be easily seen to be asymptotically stable. Note that the eigenvalues depend on both the reaction rate coefficient and some of the initial concentrations.

Example II

Here the eigenvalues only depend on the reaction rate coefficients,

LinearizedEigenvalues[rhsII,variables/.stoiII,spII]

$$\left\{ \left\{ 0, \frac{-k_{-1}-k_1-k_2-\sqrt{-4k_1k_2+(k_{-1}+k_1+k_2)^2}}{2}, \frac{-k_{-1}-k_1-k_2+\sqrt{-4k_1k_2+(k_{-1}+k_1+k_2)^2}}{2} \right\} \right\}$$

The stability of the stationary point cannot be decided based on linear stability analysis; further investigations are needed. (Negativity of two eigenvalues suggest that it is probably stable.)

In the general case the negative real parts cannot be so easily investigated as here. More complicated methods must be used, the best known of which is the Routh-Hurwitz test. A review of these methods together with the *Mathematica* codes of the corresponding algorithms can be found in [33].

Example III

Because **LinearizedEigenvalues** calculates the stationary points itself one cannot expect that it is able to calculate the linearized eigenvalues automatically. What can be done is to apply the program with the calculated stationary point in the following way

**LinearizedEigenvalues[rhsIII/.meyerconst,
variables/.stoiIII,{spIII}]**

$$\{-7.33505, -0.0582115 + 0.0650087 \text{ I}, -0.0582115 - 0.0650087 \text{ I}\}$$

The presence of a conjugate pair of eigenvalues with real parts close to zero suggests the possibility of Andronov-Hopf bifurcation leading to periodic solutions [10].

4.2. Temporal evolution of relevant concentrations

The temporal evolution of the concentrations is of foremost importance, l , r , c_{LR} in Example I, g , c_{GLR} , g^* in Example II, and x , y , z in Example III. In order to calculate these quantities as functions of time the initial values of each variable need be known

on the basis of measurements or theoretical arguments (although precise values are not always required, see below).

It can be proven from the form of kinetic differential equations alone (without any physical reasoning) that the solutions of kinetic differential equations are always non-negative [39]. (However, it may happen that a numerical method provides negative quantities, which prompted new developments in numerical mathematics aimed at overcoming this difficulty, see e.g. [8], [9], [16]).

The kinetic differential equation together with the initial values make up an initial value (or Cauchy) problem. This problem can be solved in many cases either symbolically with **DSolve** or numerically with **NDSolve**. Symbolic solutions can also be based on recent results on the explicit Taylor representation of the solutions of polynomial differential equations [3], [4].

4.2.1. Symbolic solution (Example II)

Program **Deterministic** (number 7 in Table 4) calculates in analytical form the solutions of a kinetic differential equation given the matrices of stoichiometric coefficients, the vector of reaction rate coefficients, the variable names and initial values of the variables. The code uses **DSolve**. An analytical solution can only be obtained in simple cases when the right hand side is linear or if there is only a single concentration involved in the model. Let us try it on Example (2a, b):

```
solIIab = Deterministic[rhsIIab,variables/.stoiIIab,{g0,0,0}]
```

$$\left\{ \left\{ g[t] \rightarrow E^{-k_1 t} g_0, c_{GLR}[t] \rightarrow \frac{(-E^{-k_1 t} + E^{-k_2 t}) g_0 k_1}{k_1 - k_2} \right\}, \right. \\ \left. g^*[t] \rightarrow \frac{g_0 (k_1 - E^{-k_2 t} k_1 + (-1 + E^{-k_1 t}) k_2)}{k_1 - k_2} \right\} \}$$

which is the complete explicit solution.

4.2.2. Numeric solutions (Example III)

However, in many cases one cannot solve the kinetic differential equation analytically. This is the case of the Meyer-Stryer model because its right-hand side contains three variables and is highly nonlinear. If **Deterministic** is applied to this system it returns the input unevaluated, which is the common behavior of *Mathematica* programs. Then it is useful to have Program 8 (**NDeterministic**), which is another version of Program 7 using **NDSolve** instead of **DSolve**. In this case the program structure and its arguments are different; the input arguments are the right-hand side, the names of the variables, the initial values of the variables, an initial time

(which has been given the default value 0), a final time and optional arguments to be passed to **NDSolve**. The output is a list of functions numerically interpolating the concentration vs. time curves.

The use of this program can be illustrated with the Meyer-Stryer model (values of all the coefficients are given in Table 5). Variable **rhsIII**, determined in section 3.2.3, contains the right hand side of the model in symbolic form. The concentration variables being $x(t)$, $y(t)$ and $z(t)$, a possible set of initial conditions is $x_0 = 1.2$, $y_0 = 0.2$, $z_0 = 13$. Although these values have not been provided by the authors, they reproduce qualitatively their results because of the special structure of the model (see below). Let the final time be 1200 and the number of calculation steps be 10000 (this is done using an optional parameter to **NDSolve**), as follows:

```
solIII = NDSolve[rhsIII/.meyerconst,  
variables/.stoiIII,{1.2,0.2,13},1200,  
MaxSteps → 10000];
```

The program stops with a message indicating that 10000 steps were not enough. If the results up to 1200 is really needed then **MaxSteps** must be changed to 15000. Here (see figures below) it is sufficient to do the calculations up to time 1000, for which 10000 steps are more than enough. The variable **solIII** contains

```
{ {x[t] → InterpolatingFunction[{{0.,1000.}}, "<>"][t],  
  y[t] → InterpolatingFunction[{{0.,1000.}}, "<>"][t],  
  z[t] → InterpolatingFunction[{{0.,1000.}}, "<>"][t]} }
```

For using these solutions one has to write **x[t]/.solIII** etc.

4.3. Presentation of results: solutions, trajectories, selectivity curves

Plotting can hardly be automated: since each case needs individual considerations, presentation of the solution of a kinetic differential equation is not a trivial task. The plots are of two kinds, kinetic curves (quantities vs. time) showing the temporal evolution of the individual concentrations, and *selectivity curves* (evolution in the state space, i.e. concentration vs. concentration) obtained from pairs or triplets of concentrations. The evolution of the vector of concentrations as a function of time is called the *solution*, which can only be visualized in the case of one or two species. If all coordinates are involved in the case of a selectivity curve, then it is the *trajectory* of the solution.

Example II

Fig. 1 approximately here

Fig. 1 shows the solutions $g(t)$, $g^*(t)$ and $c_{\text{GLR}}(t)$. The plot for the longer period indicates that the model reflects a relatively fast coupling thus the precoupling hypothesis, that G and C_{LR} are bound together from the beginning, is not needed.

Fig. 2 approximately here

Fig. 2 shows the trajectory in 3D space (g , c_{GLR} and g^*). The selectivity curves (g vs. c_{GLR} , c_{GLR} vs. g^* and g vs. g^*) are monotonous, thus they are not so interesting. They could have been produced by applying **ParametricPlot** to two of the coordinate functions.

Example III

Fig. 3 approximately here

Fig. 4 approximately here

Fig. 5 approximately here

The kinetic curves are shown in Figs. 3 and 4. The selectivity curve (Fig. 5) is closed which means that the solution is periodic. It suggests that no matter how the concentrations start they will move to a closed curve, which is a *stable limit cycle*. It explains why the choice of initial conditions is usually of minor importance and why they often need not be provided. Fig. 6 shows the trajectory in 3D space; it reflects again the periodicity of the solution.

Fig. 6 approximately here

5. Discussion

5.1. Extensions

This paper provides an overview of the analysis of biochemical systems illustrated by specific examples. Programs written in the *Mathematica* language are given for

solving the main problems met in this kind of analysis. (It is not a hard task to modify the programs so as to use version 2.2 or earlier; some of the programs may be improved for version 4.) They are intended to be as general as possible and can be applied to chemical reaction *stricto sensu* but also to conformational changes and to translocations. With data storage in files (see [40]), they can be applied to large models with several dozens (or even hundreds) of steps and species. However difficulties may be met when applying them to similar or more complicated cases than treated here.

We have considered the case of reaction rate coefficients that do not *depend on time*. However, it frequently occurs, even in the case of kinetics of the mass-action type, that the rate coefficients do depend on time. This is common in neurochemistry because the rate of a reaction step often depends on the presence or concentration of a regulator molecule (e.g. [38]) or on the voltage of the membrane (as for the axonal sodium channel in the Hodgkin and Huxley model, see e.g. [18]; [35]). Such reactions are usually highly nonlinear and characterized by thresholds. Because such models have no explicit solutions, they must be solved numerically. Example III on calcium spikes shows that the programs presented can be used also in this case.

Explicit symbolic solution (or even some first integrals) of a kinetic differential equation can only be obtained in rare cases. It is easier to get numerical solutions although this may also be tricky. Moreover, numerical solutions lack generality because they show the behavior of the system only under special circumstances. Another approach is the *qualitative theory of differential equations* which makes it possible to draw general conclusions even in cases when symbolic solutions cannot be obtained. Several methods are available (see e.g. [1], [10], [13], [26]; or on applications: [30], [25]), one of which, linear stability analysis, is considered in the present paper. Sensitivity analysis may also be considered to belong to this theory; it shows how sensitive to changes of some parameters, including initial conditions, the results are [37].

5.2. Limitations

Special *delay* models are used in several areas of reaction kinetics. However, no general delay model is presently available in formal kinetics. Consequently, the only practical approach at present might be writing a program to solve delay differential equations using a step-by-step method.

Spatial effects are often important. We have taken them into account in Example III, although in an incomplete way, because translocations were assumed to proceed between spatially homogeneous compartments following virtually first order kinetics.

In mathematical terms we have only dealt with ordinary differential equations and not with partial differential equations. In engineering terms the same idea is expressed in saying that we only used concentrated parameter models as opposed to distributed parameter models. The spatial distribution of the concentrations has not been taken into account, only the total quantity within a given, supposedly homogeneous compartment, was. Turning to partial differential equation models (more specifically, to reaction diffusion equations) makes all the investigations more complicated and less easy to automatize.

Another limitation of our present approach is that we have considered only the deterministic case. This is a satisfactory description for systems involving large numbers of molecules. For systems where at least some species playing an important role are few in number, it is necessary to turn to *stochastic models* ([7], [12], [23]). In this case one would like to formulate and partially solve (symbolically or numerically) the evolution equations for probability distributions, transition probabilities, moments etc. Then the numerical investigations of the deterministic model are replaced by *simulations* providing a picture of the time evolution of the selected model.

Acknowledgements

This work was partly supported by joint cooperation project Balaton No 835YJ between France and Hungary and by grants from the Hungarian National Scientific Foundation, No. T 025472 and T 037491 to J.T.

Note

A *Mathematica* notebook containing all the programs presented in Tables 2–4 and their extended and deconstructed versions, with more examples and more detailed comments is available at the web site www.math.bme.hu/~jtoth. In case *Mathematica* 3.0 or higher is not available, the program *MathReader* for reading *Mathematica* notebooks can be downloaded from the web site www.wri.com/MathSource.

References

- Arrowsmith, D.K., Place C.M., 1982. Ordinary Differential Equations. A Qualitative Approach with Applications. Chapman and Hall, London, New York .
- Atkins, P., 1977. Physical Chemistry, 6th ed., Freeman, New York.
- Barton, D., Willers, I.M., Zahar, R.V.M., 1971. The Automatic Solution of Systems of Ordinary Differential Equations by the Method of Taylor Series. In: O.R. Petrocelli (Ed.), The Best Computer Papers of 1971, pp. 147–163.
- Brenig, L., 1988. Complete factorisation and analytic solutions of generalized Lotka-Volterra equations. Phys. Lett. 1334, 378–382.
- Buchberger, B., 1970. Ein algorithmisches Kriterium für die Lösbarkeit eines algebraischen Gleichungssystems. Aequationes Math. 4, 374–383.
- Destexhe, A., Mainen, Z.F, Sejnowski, T.J., 1994. Synthesis of models for excitable membranes, synaptic transmission and neuromodulation using a common kinetic formalism. J comput Neurosci 1, 195–230.
- Érdi, P., Tóth, J., 1989. Mathematical Models of Chemical Reactions. Theory and Application of Deterministic and Stochastic Models. Princeton University Press, Princeton.
- Faragó, I., 1995. Nonnegativity of the difference schemes. Pure Math. Appl. 6, 147–159.
- Faragó, I., Komáromi, N., 1990. Nonnegativity of the numerical solution of parabolic problems. In: Numerical methods, Proc. Conf., Miskolc/Hungary 1990, Colloq. Math. Soc. Janos Bolyai 59, 173–179.
- Farkas, M., 1994. Periodic motions. Springer, New York (1994).
- Feinberg, M., 1989. Necessary and sufficient conditions for detailed balancing in mass action systems of arbitrary complexity. Chem Engng Sci 44, 1819–1827.
- Goss, P.J.E., Peccoud, J. 1998. Quantitative modeling of stochastic systems in molecular biology by using stochastic Petri nets. Proc. Natl. Acad. Sci. USA 95, 6750–6755.

- Guckenheimer, J., Holmes, Ph., 1983. *Nonlinear Oscillations, Dynamical Systems, and Bifurcation of Vector Fields*. Springer-Verlag, New York.
- Hille, B., 1992. *Ionic Channels of Excitable Membranes*. Sinauer, Sunderland.
- Horn, F., Jackson, R., 1972. General mass action kinetics. *Arch. Ratl. Mech. Anal.* 47, 81–116.
- Horváth, Z., 1998. Positivity of Runge–Kutta and diagonally split Runge–Kutta methods. *Appl. Numer. Math.* 28, 309–326.
- Keleti, T., 1986. *Basic Enzyme Kinetics*. Akadémiai Kiadó, Budapest.
- Koch, C., 1999. *Biophysics of Computation*. Oxford University Press, Oxford, 1999.
- Lamb, T.D., Pugh Jr, E.N., 1992. A quantitative account of the activation steps involved in phototransduction in amphibian photoreceptors. *J. Physiol.* 449, 719–758.
- Lamb, T.D., Pugh Jr, E.N., 1992. G-protein cascades: gain and kinetics. *Trends Neurosci.* 15, 291–298.
- Lánský, P., Krivan, V., Rospars, J.P., 2001. Ligand-receptor interaction under periodic stimulation: a modeling study of concentration chemoreceptors. *European Biophysics J.*, 30, 110–120.
- Lewis, G.N. (1925) A new principle of equilibrium. *Proc. Natl. Acad. Sci. USA* 11, 179–183.
- Lowe, G., Gold, G.H., 1995. Olfactory transduction is intrinsically noisy. *Proc. Natl. Acad. Sci. USA* 92, 7864–7868.
- Meyer, T., Stryer, L., 1988. Molecular model for receptor-stimulated calcium spiking. *Proc. Natl. Acad. Sci. USA* 85, 5051–5055.
- Novák, B., Tyson, J.J., 1997, Modeling the control of DNA replication in fission yeast. *Proc. Natl. Acad. Sci. USA* 94, 9147–9152.
- Perko, L., 1996. *Differential Equations and Dynamical Systems*. 2nd ed. Springer, New York.
- Pilling, M.J., Seakins, P.W., 1997. *Reaction Kinetics*. Oxford University Press, Oxford.

- Rospars, J.P., Lánský, P., Tuckwell, H., Vermeulen, A. (1996) Coding of odor intensity in a steady-state deterministic model of an olfactory receptor neuron. *J. comput. Neurosci.* 3, 51–72.
- Rospars, J.P., Krivan, V., Lánský, P., 2000. Perireceptor and receptor events in olfaction. Comparison of concentration and flux detectors: a modeling study. *Chem. Senses* 25, 293–311.
- Schneider, K.R., Wegner, B., Tóth, J., 1987. Qualitative analysis of a model for synaptic slow waves. *J. Math. Chem.* 1, 219–234.
- Szili L., Tóth, J., 1997. On the origin of Turing instability. *J. Math. Chem.* 22, 39–53.
- Tóth, J., Li, G., Rabitz, H., Tomlin, A.S., 1997. The effect of lumping and expanding on kinetic differential equations. *SIAM J. Appl. Math.* 57, 1531–1556.
- Tóth, J., Szili, L., Zachár, A., 1998. Stability of polynomials. *Math. Educ. Res.* 7, 5–12.
- Tóth, J., 2002. Formal kinetics with applications, In: *Proc. 6th World Multiconference on Systemic, Cybernetics and Informatics*, N. Callaos, M. Margenstern and B. Sanchez (Eds.), Vol. XI (Computer Science II), Orlando FL, pp. 573-576.
- Tuckwell, H., 1988. *Introduction to Theoretical Neurobiology. Vol. 2, Nonlinear and stochastic theories*, Cambridge University Press, Cambridge.
- Turányi, T., 1990. Sensitivity analysis of complex kinetic systems. Tools and applications. *J. Math. Chem.* 5, 203–248.
- Turányi, T., Tóth, J., 1992. Comments to an article of Frank-Kamenetski on the Quasi-Steady-State Approximation. *ACH–Models in Chemistry* 129, 903–914.
- Tyson, J.J., Novák, B., Odell, G.M., Chen, K., Thron, C.D., 1996. Chemical kinetic theory: understanding cell-cycle regulation. *Trends Biochem. Sci.* 21, 89–96.
- Volpert, A.I., Khudyaev, S.I., 1985. *Analysis in the Classes of Discontinuous Functions and Equations of Mathematical Physics*. Martinus Nijhoff Publishers, Dordrecht, Boston, Lancaster.
- Wolfram, S., 2000. *The Mathematica Book (Fourth edition)*. Wolfram Media & Cambridge University Press, Cambridge.

Zachár, A., 1998. Comparison of transformations from nonkinetic to kinetic models
ACH–Models in Chemistry 135, 425–434.

Tables

Table 1. Rates and multipliers of rates for species^a in Example III^b

Component no. of rate vector \mathbf{w}	Rate components of $\mathbf{w}(x, y, z)$	Multiplier of X	Multiplier of Y	Multiplier of Z	Eqn. no.
1	$c_1 \frac{z^3}{(K_1+z)^3} y$	1 – 0	0 – 1	1 – 1	(5a)
2	$c_2 \frac{x^2}{(x+K_2)^2}$	0 – 1	1 – 0	0 – 0	(5b)
3	$c_3 y^2$	1 – 0	1 – 2	0 – 0	(5c)
4	c_4	1 – 0	0 – 0	0 – 0	(5d)
5	$\frac{c_4}{c_5^{3.3}} x^{3.3}$	0 – 1	0 – 0	0 – 0	(5e)
6	$c_6 \rho \frac{x}{x+K_3}$	1 – 1	0 – 0	1 – 0	(5f)
7	$c_7 z$	0 – 0	0 – 0	0 – 1	(5g)

^a x, y, z are concentrations of X (cytosolic Ca^{2+}), Y (Ca^{2+} in endoplasmic reticulum) and Z (inositol-1,4,5-trisphosphate). c_1 to c_7 and K_1, K_1, K_3 are constants (see Table 5). $0 \leq \rho \leq 1$ in component 6 indicates the degree of receptor-dependent activation.

^b From Meyer and Stryer (1988). Notations were modified.

Table 2 : Source code of the `ToCanonical` *Mathematica* function

```

ToCanonical[chains_] :=
  Module[{rulebreak={
(* Rules to break up chains *)
( $\mathbf{x} \leftrightarrow \mathbf{y} \leftrightarrow \mathbf{z}$ )  $\rightarrow \{\mathbf{x} \leftrightarrow \mathbf{y}, \mathbf{y} \leftrightarrow \mathbf{z}\}$ ,
( $\mathbf{x} \leftrightarrow \mathbf{y} \rightarrow \mathbf{z}$ )  $\rightarrow \{\mathbf{x} \leftrightarrow \mathbf{y}, \mathbf{y} \rightarrow \mathbf{z}\}$ ,
( $\mathbf{x} \leftrightarrow \mathbf{y} \leftarrow \mathbf{z}$ )  $\rightarrow \{\mathbf{x} \leftrightarrow \mathbf{y}, \mathbf{y} \leftarrow \mathbf{z}\}$ ,
( $\mathbf{x} \rightarrow \mathbf{y} \leftrightarrow \mathbf{z}$ )  $\rightarrow \{\mathbf{x} \rightarrow \mathbf{y}, \mathbf{y} \leftrightarrow \mathbf{z}\}$ ,
( $\mathbf{x} \leftarrow \mathbf{y} \leftrightarrow \mathbf{z}$ )  $\rightarrow \{\mathbf{x} \leftarrow \mathbf{y}, \mathbf{y} \leftrightarrow \mathbf{z}\}$ ,
( $\mathbf{x} \rightarrow \mathbf{y} \rightarrow \mathbf{z}$ )  $\rightarrow \{\mathbf{x} \rightarrow \mathbf{y}, \mathbf{y} \rightarrow \mathbf{z}\}$ ,
( $\mathbf{x} \rightarrow \mathbf{y} \leftarrow \mathbf{z}$ )  $\rightarrow \{\mathbf{x} \rightarrow \mathbf{y}, \mathbf{y} \leftarrow \mathbf{z}\}$ ,
( $\mathbf{x} \leftarrow \mathbf{y} \rightarrow \mathbf{z}$ )  $\rightarrow \{\mathbf{x} \leftarrow \mathbf{y}, \mathbf{y} \rightarrow \mathbf{z}\}$ ,
( $\mathbf{x} \leftarrow \mathbf{y} \leftarrow \mathbf{z}$ )  $\rightarrow \{\mathbf{x} \leftarrow \mathbf{y}, \mathbf{y} \leftarrow \mathbf{z}\}$ ,
(* Rules to arrange arrow directions *)
rulerev={( $\mathbf{x} \leftarrow \mathbf{y}$ )  $\rightarrow (\mathbf{y} \rightarrow \mathbf{x})$ , ( $\mathbf{x} \leftrightarrow \mathbf{y}$ )  $\rightarrow \{\mathbf{x} \rightarrow \mathbf{y}, \mathbf{y} \rightarrow \mathbf{x}\}$ }},
(* Rules applied *)
Flatten[chains//.rulebreak/.rulerev1]]

```

Table 3 : Source code of the `ToStoichiometry` *Mathematica* function

```

ToStoichiometry[r_List, externalspecies_List:{}] :=
  Module[{RemoveExternal, ReactionSteps, Species,
    Complexes, ComplexVectors, Reactants, ReactantVectors,
    Products, ProductVectors,
  (* Creates the rules to remove signs from complexes *)
    RemoveSigns = {(x_ -> y_) -> {x, y}, n_Integer x_ -> x,
    x_ + y_ -> {x, y}}, ff},
  (* Creates the rules to remove external species *)
  RemoveExternal = {A_ /; MemberQ[externalspecies, A] -> 0};
  (* Removes external species *)
  ReactionSteps = Sort[r /. RemoveExternal];
  (* Collects complexes from both sides of reaction steps *)
  Complexes = Union[Cases[ReactionSteps, _, {2}]];
  (* Collects species excluding external ones *)
  Species
    = Sort[Complement[Flatten[r /. RemoveSigns], externalspecies]];
  (* Collects reactant and product complexes *)
  Reactants = First/@ReactionSteps;
  Products = Last/@ReactionSteps;
  (* Gives the vectors describing different complexes *)
  {ComplexVectors, ReactantVectors, ProductVectors}
    = (Outer[Coefficient[#1, #2] &, #1, Species, 1] &)
    /@{Complexes, Reactants, Products};
  (* Creates the matrices of stoichiometric coefficients *)
  {alpha, beta} = Transpose/@{ReactantVectors, ProductVectors};
  (* Creates the names of variables *)
  ff[x_Symbol] := ToExpression[ToLowerCase[ToString[x]]];
  ff[x_] := MapAt[ff, x, {{1}}];
  (* Collects and formats the result *)
  {species -> Species, M -> Length[Species],
    reactionSteps -> ReactionSteps, R -> Length[ReactionSteps],
    alpha -> alpha, beta -> beta,
    variables -> ff/@Species}

```

Table 4 : Source code of the other *Mathematica* programs

```

1a. Options[RHS] = {MassAction → True};
   RHS[stres_,opts___?OptionQ]:=Module[{w,rate},
     Print[reactionSteps/.stres];
     rate=Input[
       "Provide the reaction rates in order of reaction steps."];
     w=If[MassAction /. {opts} /. Options[RHS],
       rate (Times@@(variables)α/.stres, rate);((β−α)/.stres).w];
2. Options[Kinetic] = {MassAction → True,NonZeros → {}};
   Kinetic[α_,β_,rate_,vars_,opts___?OptionQ]:=Module[{eqs,w},
     w=If[MassAction /. {opts} /. Options[Kinetic],
       rate (Times@@varsα, rate);
     eqs=If[MassAction /. {opts} /. Options[Kinetic],
       Flatten[{Thread[[rate ≠ 0],Thread[(β−α).w==0]}],
       Flatten[{Thread[(NonZeros/.{opts}/.Options[Kinetic]) ≠ 0],
       Thread[(β−α).w==0]}]];
     Union[{ToRules[Reduce[eqs, vars]]}]
3. MassConservation[α_,β_,vars_,vars0_]:=
   Module[{ns=NullSpace[Transpose[β−α]]},
     Solve[If[ns ≠ {},Thread[ns.(vars−vars0)==0],{}],vars]]
4. Options[StationaryPoints] = {MassAction → True,NonZeros → {}};
   StationaryPoints[α_,β_,rate_,vars_,vars0_,opts___?OptionQ]:=
     Module[{w,ns=NullSpace[Transpose[β−α]]},
       w=If[MassAction → /.{opts}/.Options[StationaryPoints],
         rate Times@@varsα, rate];
     Solve[
       Flatten[{If[MassAction /. {opts} /. Options[StationaryPoints],
       Flatten[{Thread[[rate ≠ 0],Thread[(β−α).w==0]}],
       Flatten[{Thread[(NonZeros/.{opts}/.Options[StationaryPoints]) ≠ 0],
       Thread[(β−α).w==0]}]],
       If[ns ≠ {},Thread[ns.(vars−vars0)==0],{}]]],vars]]
5. LinearizedEigenvalues[rhs_,vars_,sp_]:=
   Map[Eigenvalues, Outer[D,rhs,vars]/.sp]
6. Deterministic[rhs_,vars_,vars0_,opts___]:=Module[
   {timevars=#[t]&@vars},SetAttributes[D, Listable];

```

```

Simplify[DSolve[Flatten[{
  Thread[D[timevars,t]==(rhs/.Thread[vars → timevars])],
  Thread[(timevars/.t → 0)==vars0]}],timevars,t,opts]]]
7. NDeterministic[rhs_,vars_,vars0_,tfinal_,opts___]:=
Module[{timevars=#[t]&/@vars},
SetAttributes[D, Listable];
de=Flatten[{Thread[D[timevars,t]
  ==(rhs/.Thread[vars → timevars])],
  Thread[(timevars/.t → 0)==vars0]}];
NDSolve[de,timevars,{t,0,tfinal},opts]]

```

^a The code of **RHS** is similar to the mathematical expression (23), except that multiplication (called **Times** in the program) being applied to the row vectors of a matrix, the special operation **@@**, a shorthand for the *Mathematica* function **Apply**, must be used.

^b In order to express the equality of vectors **Thread** is needed here.

Table 5 : Values of the coefficients used for numerical study of Example III ^a

```

meyerconst =
{c1 → 6.64, K1 → 0.10, c2 → 5.00, c3 → 3.13 10-5, K2 → 0.15,
 c4 → 0.50, c5 → 0.60,
 c6 → 1.00, K3 → 1.00, c7 → 2.00, ρ → 0.42};

```

^a From Meyer and Stryer (1988), except ρ (notations were modified). c_1, c_5 in s^{-1} , c_2, c_4, c_6 in μMs^{-1} , c_7 in μM , c_3 in $\mu M^{-1}s^{-1}$, K_1, K_2, K_3 in μM , ρ without dimensions (degree of receptor-dependent activation).

Figure Legends

Figure 1. Kinetic curves in Example II. Evolution of the concentrations g of the G-protein (solid line), c_{GLR} of the GLR complex (dashed line) and g^* of the activated G-protein G^* (dotted line) as a function of time. Plot obtained with `Plot[Evaluate[{g[t],c[t]}/.solII,{t,0,500}]`. Parameters: $k_1 = 0.06$, $k_{-1} = 0.01$, $k_2 = 0.001$, $l_0 = 0$, $g_0 = 1000$, $c_0 = 0$.

Figure 2. Selectivity curve in Example II. Trajectory of the solution in state space with concentrations of G , C_{GLR} and G^* as the three coordinates. Plot obtained with `ParametricPlot3D[Evaluate[{g[t],cGLR[t],g*[t]}/.solII,{t,1,30}]`.

Figure 3. Kinetic curves in Example III. Evolution in time of the concentrations of calcium in cytosol X (dashed line) and of inositol trisphosphate Z (solid line). The interspike interval decreases when ρ increases. Plot obtained with `Plot[Evaluate[{x[t],z[t]}/.solIII],{t,0,100}]`. Parameters: see Table 5.

Figure 4. Kinetic curve in Example III. Evolution in time of the concentration endoplasmic reticulum Y . Plot obtained with `Plot[Evaluate[{y[t]}/.solIII],{t,0,100}]`; Parameters: see Table 5.

Figure 5. Selectivity curve in Example III. Evolution in state space of the concentrations of calcium in cytosol X and in endoplasmic reticulum Y . Plot obtained with `ParametricPlot[Evaluate[{x[t],y[t]}/.solIII],{t,0,100}]`; Parameters: see Table 5.

Figure 6. Time evolution in state space of the concentrations of cytosolic calcium X and inositol triphosphate Z in Example III. Plot obtained with `ParametricPlot3D[Evaluate[{t,x[t],z[t]}/.solIII],{t,0,120}]`; Parameter: see Table 5.