Simulating Biological Pathways with a Continuous Petri Net Using Runge-Kutta Methods

Sunku Kang[1], Byungwoo Lee[2], Jihoon Yang[3], Seonho Kim[4], and Saejoon Kim[5]
Economic Research Institute, Samsung[1]
Department of computer Science, Sogang University[2,3,4,5]
sun9sun9@seri.org[1], *elva1212@sogang.ac.kr*[2], *yangjh@sogang.ac.kr*[3],
shkim@lex.yonsei.ac.kr[4], *saejoon@sogang.ac.kr*[5]

Abstract

The application of Petri nets to analyze and simulate biological pathways has been developed vigorously. Some of the modifications of Petri nets appeared to perform quantitative analysis of biological pathways: hybrid Petri net, timed Petri net, stochastic Petri net and so on. For a quantitative analysis of biological pathways, we propose to adopt Runge-Kutta methods to act on a continuous Petri net. We show that Runge-Kutta methods are applicable to a continuous Petri net in glycolysis and apoptosis pathways with reliable and efficient simulation results.

1. Introduction

Understanding biological pathways is a significant task in systems biology. Traditionally, the quantitative analysis of biological pathways has been based on analytical methods, which are to solve the ordinary differential equations (ODEs) based on mathematical foundation. On the other hand, the qualitative analysis is based on discrete event systems which analyze the structural properties of the biochemical reactions [1]. Reddy *et al.* introduced Petri nets to the qualitative analysis of the biochemical reaction systems based on discrete event systems [2]. They also indicated that the concurrent property of Petri nets is suitable to the qualitative analysis of biological pathways [2].

Motivated by the qualitative analysis, many applications of high-level Petri nets (e.g. hierarchical Petri nets, hybrid Petri net, timed Petri nets, and stochastic Petri nets, etc.) have been developed for various models and simulation purposes (e.g. quantitative analysis) [1]. Hofestädt introduced a self-modified Petri net, which is modified for the quantitative analysis of biochemical networks [3]. Matsuno *et al.* developed a hybrid Petri net for the simulation of gene regulation networks and signal transduction pathways [4,5]. Genrich suggested a high-level Petri net model as an efficient simulation method [6]. Against this background, we are motivated to simulate biological pathways based on high-level Petri nets.

However, the simulations of biological pathways represented by Petri nets are usually based on Euler method, which is not stable and accurate. The reason is that the method generally applies to discrete event systems, while the simulation of biological pathways is continuous. We therefore introduced Runge-Kutta methods to the simulation of the biological pathways to improve stability, accuracy and convenience.

In Section 2, we analyze the modeling method of biological pathways with high-level Petri nets used in [5-8] and explain our modifications. In Section 3, we mention the simulation methods used in [5-8] and their limitations and explain the suitability of Runge-Kutta methods to the simulation of Petri nets. In Section 4, we demonstrate the simulation results of our approach for glycolysis and apoptosis pathways. In Section 5, we close the paper with concluding remarks.

2. Representing biological pathways with a Petri net

A Petri net is a directed graph formed by two kinds of nodes, called a *place* and a *transition*, and directed edges called an *arc* [9]. An arc connects a place to a transition or a transition to a place [9]. The former is called an *input-arc*, and the latter is called an *output-arc*. As shown in Figure 1, a place is represented as a circle, a transition as a box, and an arc as an arrow. A non-negative number of tokens, which represents the state of the Petri net, may be assigned to each place [9].

* This work was supported by grant No. R01-2004-000-10689-0 from the Basic Research Program of the Korea Science & Engineering Foundation

A marking is defined as a vector whose elements represent the states of tokens. Each arc has a weight that denotes the amounts of tokens that pass after an execution. If the number of tokens in an input-place is greater than or equal to the weight, the transition can be enabled.

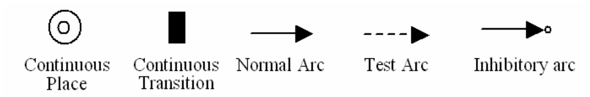

Figure 1. The representation of the elements of the continuous Petri net

Places can represent various biological elements (e.g. metabolites, proteins and genes) and the tokens that are assigned to places represent the concentration or degree of the elements. Transitions can be regarded as the reaction among input-places that are linked with arcs to a transition and output-places that are linked with arcs from a transition. Figure 2 illustrates the Henri Michaelis-Menten reaction [10] with a Petri net. In Figure 2, [X] denotes the concentration of X. S is a substrate, E is an enzyme, ES is the complex of S and E, and P is a product.

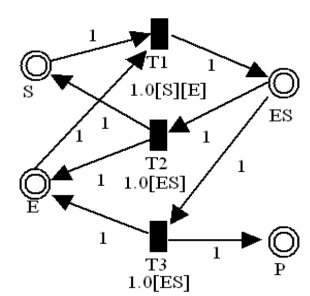

Figure 2. Henri Michaelis-Menten reaction with a Petri net

For a quantitative simulation, other elements have been added to the low-level Petri net. A hybrid Petri net [1] is added with continuous places, which have non-negative real number of tokens. In functional Petri nets, the reaction function, like Michaelis-Menten equation[†], is assigned to an arc or a transition. For modeling various pathways, inhibitory arcs and test arcs are also introduced. An inhibitory arc controls the firing: when the token of input place is greater than the weight, the reaction is prevented. An inhibitory arc can be used to represent an inhibitor. A test arc does not pass the tokens to a transition. A test arc can be used to represent an element whose amount is not changed (e.g. an enzyme).

In a continuous Petri net, the continuous elements are used only since discrete elements are naturally included in its continuous counterparts. In previous models, an arc weight is the ratio of the removal and addition of elements in a reaction, and at the same time, an arc weight specifies the firing conditions. This might cause confusion in the use of arc weights and is also inflexible in modeling inhibitions. To relieve this problem, each inhibitory arc has an activation range, which controls the transition connected by an inhibitory arc to be activated as long as the token of the place connected by the inhibitory arc is between the activation ranges.

3. Simulating biological pathways with a Petri net

The simulation algorithms of Petri nets are described in the literature [3,5-7] but they did not explain the overall procedures in detail. We integrate the methods in [3,5-7] as follows.

Let P be a finite set of places ($P = \{p_1, p_2, \ldots, p_n\}$) and T be a finite set of transitions ($T = \{t_1, t_2, \ldots, t_m\}$). Let A_{in} be an $m \times n$ matrix of input-arc weights of which the columns represent transitions and the rows represent places. Let A_{out} be an $m \times n$ matrix of output-arcs weights. Let $M(t)$ be the marking at time t (a vector of tokens as described in Section 2): $M(t) = (m_{p1}(t), m_{p2}(t), \ldots, m_{pn}(t))^T$. Let $V(M(t))$ be a vector with the reaction functions that are assigned to transitions. $V(M(t))$ denotes the reaction speed at time t given $M(t)$. The Henri Michaelis-Menten reaction in Figure 2 can be represented like the following:

$$P = \{S, E, ES, P\}$$

$$T = \{T1, T2, T3\}$$

$$M(t) = (1\ \ 1\ \ 1\ \ 0)$$

$$A_{in} = \begin{array}{cc} \begin{array}{ccc} \mathrm{T1} & \mathrm{T2} & \mathrm{T3} \end{array} & \\ \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix} & \begin{array}{l} S \\ E \\ ES \\ P \end{array} \end{array}$$

$$A_{in} = \begin{array}{cc} \begin{array}{ccc} \mathrm{T1} & \mathrm{T2} & \mathrm{T3} \end{array} & \\ \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix} & \begin{array}{l} S \\ E \\ ES \\ P \end{array} \end{array}$$

$$V(M(t)) = \begin{pmatrix} 1.0m_s(t)m_E(t) \\ 1.0m_{ES}(t) \\ 1.0m_{ES}(t) \end{pmatrix}$$

[†] $dP/dt = -dS/dt = k_{cat} \times E \times S / (K_m + S)$, where k_{cat} is the turnover number, and K_m is a constant.

Then, using the notations, the simulation procedures can be described by the following pseudo-code:

```
$M0$: initial marking
$dM$: change of marking
$dt$: interval within an execution
$count$: counter
$t$: simulation time
$t = count = 0$
$M(0) = M0$
do
  $dM = -A_{in} \times V(M(count)) + A_{out} \times V(M(count))$
        $M(count + 1) = M(count) + dM \times dt$
        $count{+}{+}$
        $t = t + dt$
until $t <= finish\_time$
```

The simulation methods described by the pseudo-code are similar to Euler method. Because Euler method is neither accurate nor stable, it is not recommended in practice [11]. Figure 3 is the result of the simulation of the Henri Michaelis-Menten reaction in Figure 2 with Euler method when the simulation interval is 0.5. Figure 3 shows that the simulation is instable with an arbitrary large simulation interval. Therefore, we have to choose a simulation interval small enough to obtain reliable results.

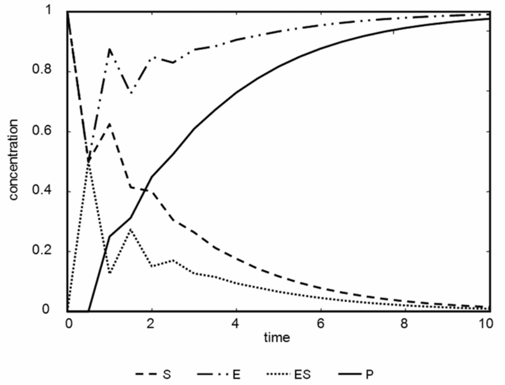

Figure 3. Simulation results of the Henri Michaelis-Menten reaction using Euler method with simulation interval of 0.5

To perform more accurate and stable simulations, we introduce Runge-Kutta methods. Runge-Kutta methods are one of the popular methods to solve ordinary differential equations, which are based on midpoint method [11]. Even though Runge-Kutta methods do not guarantee very high accuracy, adapting them to the simulation of biological pathways based on Petri nets can be done naturally since we only need to calculate the midpoints. In order to calculate the midpoints, we placed sub-executions (i.e. the inner executions to calculate the deviations of the midpoints) at each execution. The following pseudo-code is our extension incorporating the 4th order Runge-Kutta method:

```
$M0$: initial marking
$dM$: change of marking
$dt$: interval within an execution
$count$: counter
$div = \{0.0, 0.5, 0.5, 1.0\}$
$sub(4)$: variables from sub-execution
$t$: simulation time
$t = count = 0$
$M(0) = M0$
do
  $sub(0) = -A_{in} \times V(M(count)) + A_{out} \times V(M(count)) \times dt$
  for $i$ from 1 to 3
  begin
    $sub(i) = (-A_{in} \times V(M(count) + su([i–1] \times div(i)) +$
        $A_{out} \times V(M(count) + sub[\mathrm{i}–1] \times div(i))) \times dt$
 end
  $dM = sub(0)/6 + sub(1)/3 + sub(2)/3 + sub(3)/6$
  $count{+}{+}$
 $t = t + dt$
 $M(count+1) = M(count) + dM$
until t $< finish\_time$
```

As shown in the pseudo-code, the Runge-Kutta methods need some modification: the space for midpoints and 4-fold calculation for the midpoints. As we can see clearly, the results in Figure 4 with the 4th order Runge-Kutta method are more stable than the results in Figure 3. (We will emphasize this point again in Section 4 with another experiment.)

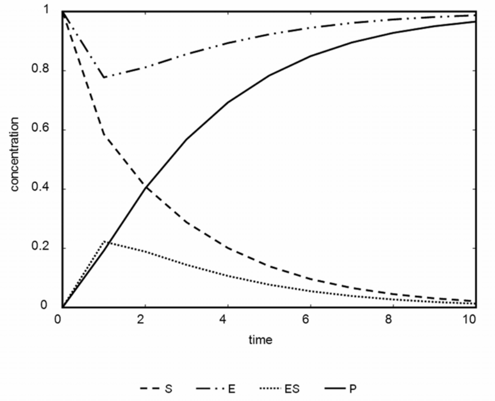

Figure 4. Simulation results of the Henri Michaelis-menten reaction using the 4th order Runge-Kutta method with simulation interval of 0.5

Both of the 4th order Runge-Kutta and Euler methods are required to select the simulation interval with an arbitrary small number. If not, the result will be wrong. On the contrary, if we select too small a number, the additional computations will be prohibitive. Therefore, these methods are not appropriate for the problems of optimization or inference of pathways (i.e. searching the optimal constants to maximize the production of certain elements or inferring the biological pathways). To relieve the problem, Genrich suggested that the change of the simulation interval (corresponding to the state of marking) to reduce the computational overhead [6]. In [6], he defined several modes, speed factor and model time, that are switched dynamically according to the state of the marking. However, they did not clearly explain how to make choices among them.

The application of the 4th order Runge-Kutta method can be extended to an adaptive step-size method. Using this adaptive step-size Runge-Kutta method, we can control the simulation interval guaranteeing the error bound as described in [6], without trying to select an appropriate simulation interval. The additional calculations are not necessary in case of excessively small intervals. The main idea of adaptive step-size Runge-Kutta method is that if the error (the difference between the 4th order method and the 5th order method) is smaller than the given threshold, then we can either increase or decrease the simulation interval according to the error automatically [11]. The resulting adaptive step-size Runge-Kutta method to a Petri net can be represented similar to the aforementioned pseudo-code with slight modifications.

4. Experiments

We developed software for representing and simulating biological pathways by the three methods described earlier: Euler method, 4th order Runge-Kutta, and adaptive step-size Runge-Kutta methods. We compare the performance of the methods with the glycolysis and apoptosis pathways.

4.1. Glycolysis pathway

Glycolysis pathway is a well-known metabolic pathway. We chose the textbook Glycolysis pathway, and obtained the Michaelis-Menten constants from BRENDA[‡] (The database for biological pathways). In Figure 5, we show the glycolysis pathway.

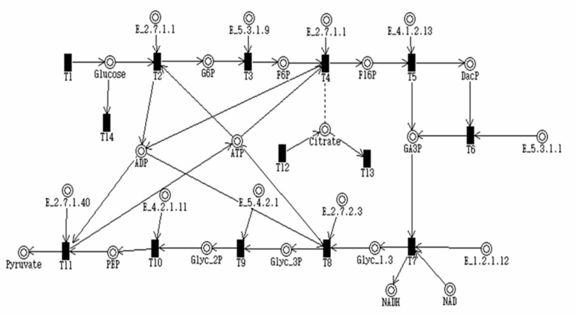

Figure 5. Glycolysis pathway represented with a continuous Petri net

With the Euler method used in [3,5-7], the simulation interval was to be less than 0.008 to get proper results. If the simulation interval is larger than 0.008, the accumulation of errors causes abnormal values like negative or extremely large values. On the other hand, the 4th order Runge-Kutta can get similar results when the simulation interval is 0.008. (Since the simulation result is same with that of adaptive step-size Runge-Kutta, we omit the graph displaying the simulation result for this case.) We can thus conclude that the 4th order Runge-Kutta method is feasible in loose condition, as verified in the glycolysis pathway.

As shown in Figure 6, Euler method resulted in negative or very large concentrations for some elements. On the contrary, as in Figure 7, the adaptive step-size Runge-Kutta method resulted in appropriate values.

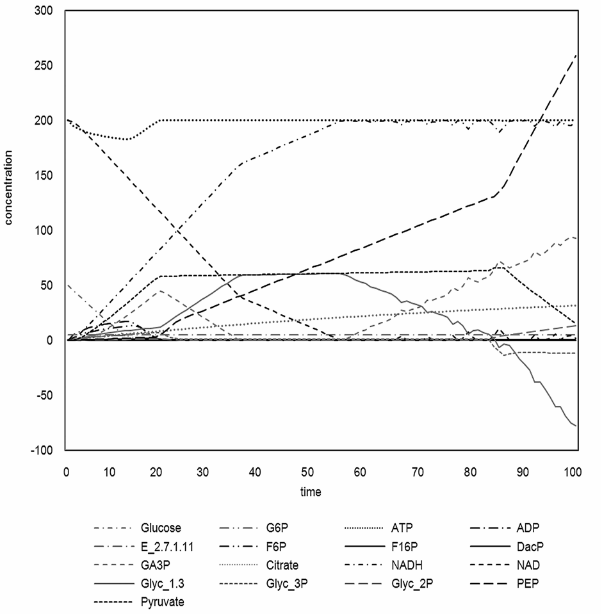

Figure 6. The results of simulating glycolysis pathway using Euler method with assigning simulation interval to 0.005

[‡] http://www.brenda.uni-koeln.de/

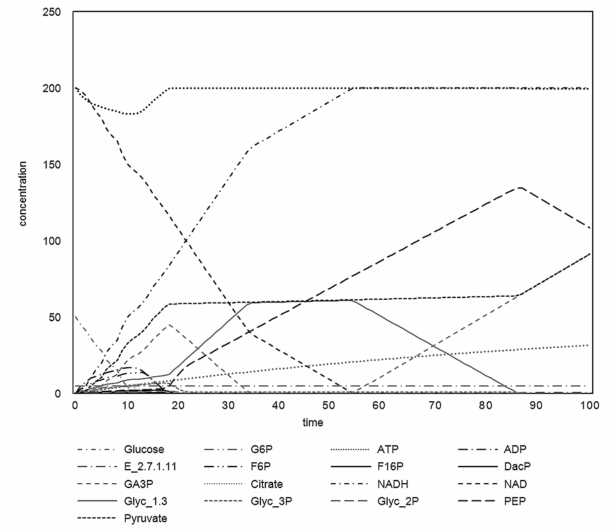

Figure 7. The result of simulating glycolysis pathway with adaptive step-size Runge-Kutta method

Table 1 exhibit the execution time for different methods. We can see that each method has comparable execution times, though Runge-Kutta methods caused slightly higher computational overhead. However, these additional computations are acceptable considering the stable results verified in Figure 6 and 7.

Table 1. The computation time of simulating glycolysis pathway (average of 1000 trials)

Simulation method	Simulation interval	Computation time(sec)
Euler	0.005	0.268
4th order Runge-Kutta	0.007	0.325
Adaptive step-size Runge-Kutta	-	0.317

4.2. Apoptosis pathway

Matsuno *et al.* simulated the apoptosis pathway, which means programmed cell-death, induced by Fas-Ligand with a hybrid Petri net [5]. We included additional features to their model, − the effect of caspase7, caspase10 and IAP. Figure 8 shows the entire apoptosis pathway.

We applied our simulation methods in the apoptosis pathway. Figure 9 is the result of the simulation of the apoptosis pathway with the adaptive step-size Runge-Kutta method. In case of Euler method, the simulation was feasible with a simulation interval less than 0.01. And as to the 4th order Runge-Kutta, it was feasible with a simulation interval less than 0.02 (looser than the interval used in Euler method). Since our experiment produced almost identical results for all the three methods (i.e. Euler, 4th order Runge-Kutta, adaptive step-size Runge-Kutta methods), we only show the results of the adaptive step-size Runge-Kutta methods in Figure 9.

Table 2 exhibits the execution time for different methods. As in the glycolysis pathway, we can see that each method has comparable execution times, though Runge-Kutta methods caused slightly higher computational overhead. We thus verified that the adaptive step-size Runge-Kutta method is also suitable for the analysis of reasonably complex (or middle-scale) networks like the apoptosis pathway.

Table 2. The computation time of simulating apoptosis pathway (average of 1000 trials)

Simulation method	Simulation interval	Computation time(sec)
Euler	0.01	1.293
4th order Runge-Kutta	0.02	1.452
Adaptive step-size Runge-Kutta	-	1.481

5. Conclusion

The Petri net provides an intuitive model to biological pathways so that it makes biological pathways easier to understand. Moreover, the concurrent property of the Petri net is appropriate to describe biological pathways. However, because its quantitative modeling for time-course simulation is based on the Euler method, it is neither stable nor convenient.

The methods that are used in solving ordinary differential equations (ODEs) are necessary to perform more stable quantitative modeling of biological pathway with a Petri net. Applying the Runge-Kutta method to a continuous Petri net is simple and suitable. We proposed an adaptive step-size Runge-Kutta method that controls the simulation interval in a Petri net dynamically and automatically, while keeping the reliability of results. This relieves the inconvenience from giving an arbitrary small simulation interval, while guaranteeing the accuracy of the simulation. And this will be very useful in a variety of simulation-based optimization problems or inference problems. We verified the feasibility of our approach in the simulations of both simple and complex biological pathways, including glycolysis and apoptosis pathways.

6. References

[1] J. W. Pinney, D. R. Westhead, and G. A. McConkey, “Petri Net Representations in Systems Biology”, *Biochemical Society Transactions*, 2003, 31, pp. 1513–1515.

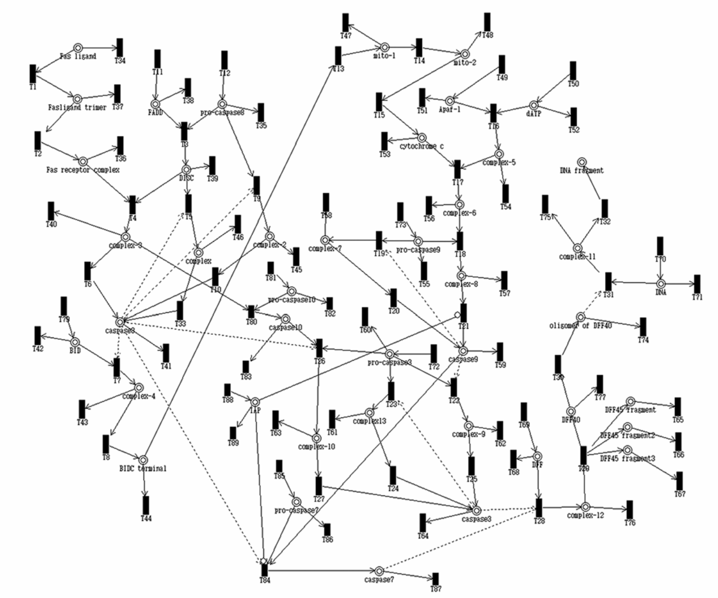

Figure 8. Apoptosis pathway represented with a continuous Petri net

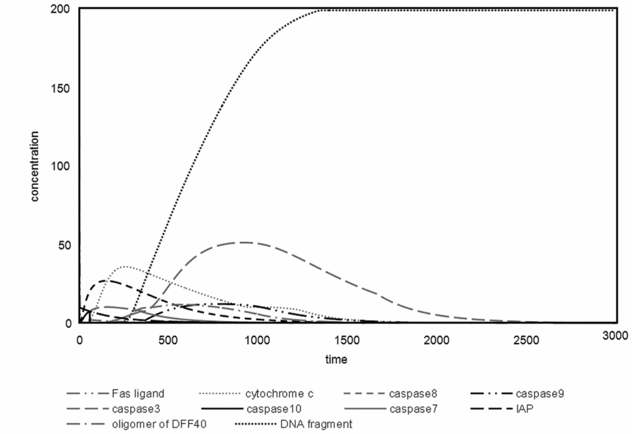

Figure 9. The result of simulating Apoptosis pathway using adaptive step-size Runge-Kutta method

[2] V. N. Reddy, M. L. Mavrovouniotis, and M. N. Liebman, "Petri Net Representation in Metabolic Pathways", *Proceedings of First International Conference on Intelligent System for Molecular Biology,* 1993, pp. 328–336.

[3] R. Hofestädt, "A Petri Net Application to Model Metabolic Processes", *Syst. Anal. Mod. Simul.*, 1994, 16, pp. 113–122.

[4] H. Matsuno, A. Doi, M. Nagasaki, and S. Miyano, "Hybrid Petri Net Representation of Gene Regulatory Network", *Pacific Symposium on Biocomputing*, 2000, 5, pp. 338–349.

[5] H. Matsuno, Y. Tanaka, H. Aoshima, A. Doi, M. Matsui, and S. Miyano, "Biopathway Representation and Simulation on Hybrid Functional Petri Net", *In Silico Biology*, 2003, 3, pp. 389–404.

[6] H. Genrich, R. Küffner, and K. Voss, "Executable Petri Net Models for the Analysis of Metabolic Pathways", *International Journal on Software Tools for Technology Transfer*, 2001, 3, pp. 394–404.

[7] M. Chen and R. Hofestädt, "Quantitative Petri Net Model of Gene Regulated Metabolic Networks in the Cell, *In Silico Bio*logy, 2003, 3, pp. 347–365.

[8] R. Kuffner, R. Zimmer, and T. Lengauer, "Pathway Analysis in Metabolic Databases via Differential Metabolic Display (DMD)", *Bioinfomatics*, 2000, 16, pp. 825–836.

[9] V. N. Reddy, M. N. Liebman, and M. L. Mavrovouniotis, "Qualitative Analysis of Biochemical Reaction Systems", *Compt. Biol. Med.*, 1996, 26, pp. 9–24.

[10] V. Henri, "Lois Générales de l'Action des Diastases", *Compt. Rend. Hebd. Acad. Sci. Paris*, 1903, 135, pp. 916–919.

[11] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. A. Vetterling, *Numerical Receipes in C. 2nd ed.,* Cambridge University Press, 1992.