Discrete & Continuous Models
& Applied Computational Science
ISSN 2658-7149 (Online), 2658-4670 (Print)

2024, 32 (3) 306–318
http://journals.rudn.ru/miph

Modeling and simulation

Research article

UDC 004.021:519.2:519.6
DOI: 10.22363/2658-4670-2024-32-3-306–318
EDN: FEMNAB

Symbolic-numeric approach for the investigation of kinetic models

Ekaterina A. Demidova[1], Daria M. Belicheva[1], Victoria M. Shutenko[1], Anna V. Korolkova[1], Dmitry S. Kulyabov[1, 2]

[1] RUDN University, 6 Miklukho-Maklaya St, Moscow, 117198, Russian Federation

[2] Joint Institute for Nuclear Research, 6 Joliot-Curie St, Dubna, 141980, Russian Federation

(received: June 4, 2024; revised: June 20, 2024; accepted: June 25, 2024)

Abstract. Our group has been investigating kinetic models for quite a long time. The structure of classical kinetic models is described by rather simple assumptions about the interaction of the entities under study. Also, the construction of kinetic equations (both stochastic and deterministic) is based on simple sequential steps. However, in each step, the researcher must manipulate a large number of elements. And once the differential equations are obtained, the problem of solving or investigating them arises. The use of symbolic-numeric approach methodology is naturally directed. When the input is an information model of the system under study, represented in some diagrammatic form. And as a result, we obtain systems of differential equations (preferably, in all possible variants). Then, as part of this process, we can investigate the resulting equations (by a variety of methods). We have previously taken several steps in this direction, but we found the results somewhat unsatisfactory. At the moment we have settled on the package Catalyst.jl, which belongs to the Julia language ecosystem. The authors of the package declare its relevance to the field of chemical kinetics. Whether it is possible to study more complex systems with this package, we cannot say. Therefore, we decided to investigate the possibility of using this package for our models to begin with standard problems of chemical kinetics. As a result, we can summarize that this package seems to us to be the best solution for the symbolic-numerical study of chemical kinetics problems.

Key words and phrases: chemical kinetics equations, stochastic differential equations, population models, one-step processes

For citation: Demidova, E. A., Belicheva, D. M., Shutenko, V. M., Korolkova, A. V., Kulyabov, D. S. Symbolic-numeric approach for the investigation of kinetic models. *Discrete and Continuous Models and Applied Computational Science* **32** (3), 306–318. doi: 10.22363/2658-4670-2024-32-3-306–318. edn: FEMNAB (2024).

1. Introduction

The chemical kinetics equations (chemical reaction networks, CRN) are a simplified version of the stochastic kinetic equations. In the works of C. W. Gardiner [1] and N. G. Van Kampen [2] the chemical kinetics equations are derived from the more general stochastic kinetic equations.

Model representation in the form of CRN is mostly used in biochemistry, theoretical chemistry and biology. Such models are based on a combination of a set of substances, which defines the state of the system, and a set of reaction events, which describe reaction rates and rules for changing the state of the system as they occur. This structure makes it easier to understand and analyze the model. Each reaction includes reagents (initial substances) and products (substances that are formed as a result of a reaction), which are written as concentrations of objects in the extended sense.

For instance, in the work [3] A. Lotka derived a system of stochastic equations from hypothetical chemical reaction. The same model was independently arrived at by V. Volterra [4]. This model describes predator–prey species interaction.

The equations of chemical kinetics are constructed according to rather primitive (but also somewhat cumbersome) rules. It seems justified to use the analytic–numerical approach for these tasks. This paper provides an overview of the main functions of the Catalyst.jl library [5, 6] for the programming language Julia [7, 8], which provides a toolkit for symbolic–numerical exploration of chemical kinetics models [9]. The Julia programming language is widely used in biology, providing a large number of tools [10, 11]. Due to Julia's support for the metaprogramming paradigm, this language is well-suited for implementing custom domain–specific languages. For example, it is possible to implement a computer algebra subsystem integrated into the language [12].

1.1. Paper structure

The section 2 briefly provides an overview of the basic principles of chemical kinetics. The section 3 provides brief information about the Catalyst.jl package. The sections 4 and 5 consider examples of constructing and solving one-dimensional and multidimensional models, respectively. The models are presented in both deterministic and stochastic forms.

2. Chemical kinetics equations

Representation of kinetics equations in the form of chemical reactions is widely used in modeling. We will briefly examine the structure of equations of this type. Consider mixtures of chemical substances X_a, $a = \overline{1,n}$. Each substance has a concentration x_a. A reaction occurs under the influence of a catalyst, and the mixture changes. This is expressed as changes in the reaction orders with respect to the substances, i.e., changes in the amounts of substances, which are represented by the sets of coefficients N_a^A, M_a^A. The total amount of substances involved in a reaction is called the reaction order. In this case, the reaction is characterized by constant proportionality coefficients k^+ и k^-, which characterize the intensity of the processes—the reaction rate.

The equation of a reversible chemical reaction in a general form is considered [13–16]:

$$\sum_a N_a^A X_a \underset{k_A^-}{\overset{k_A^+}{\rightleftarrows}} \sum_a M_a^A X_a, \qquad A = \overline{1,m}, \quad a = \overline{1,n}. \tag{1}$$

In the equation (1), M_a^A and N_a^A represent the number of components of type X_a on the left and right sides, respectively. In the interaction of type A, a set N_a^A of components of type X_a enters, and a set M_a^A of components of type X_a, or M_b^A of type X_b $(b \neq a)$ is informed. The interaction is similar in the reverse direction.

Velocity is considered proportional to the concentration of substances [2]:

$$s_A^+ = k^+ \prod_a {x_a}^{N_a},$$

$$s_A^- = k^- \prod_a {x_a}^{M_a}.$$

The state of the system $\mathbf{x} = (x_1, x_2, \dots, x_n)$ is introduced, where x_a is the substance concentration, i.e., the number of elements of type X_a. The change in the state of the system is described by the vector $\mathbf{r}^A$:

$$\mathbf{r}^A = \mathbf{M}^A - \mathbf{N}^A.$$

To construct a system of differential equations corresponding to the interaction scheme (1), we will consider the Fokker–Planck equation [1]:

$$\partial_t p(X,t) = -\sum_a \partial_a [A_a(x) P(X,t)] + \frac{1}{2} \sum_{a,b} \partial_a \partial_b [B_{ab}(X) P(X,t)], a = \overline{1,n}, b = \overline{1,n},$$

where

$$\begin{aligned} A_a(X) &= \sum_A r_a^A [s_A^+(x) - s_A^-(x)], \\ B_{a,b}(X) &= \sum_A r_a^A r_b^A [s_A^+(x) + s_A^-(x)]. \end{aligned} \tag{2}$$

The Fokker–Planck equation is mathematically equivalent to the Langevin equation. In the Langevin equation

$$\mathrm{d}\mathbf{x} = \mathbf{a}(\mathbf{x})\,\mathrm{d}t + \mathbf{b}(\mathbf{x})\,\mathrm{d}\mathbf{W},$$

where W—n-dimensional Wiener process, the coefficient $a(x)$ ccorresponds to the coefficient $A(X)$, $b(x) = B(x)B(x)^T$ [17, 18].

We discard the stochastic term and use the coefficients of A_a from equation (2) to obtain a system of differential equations:

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \sum_A r_a^A [s_A^+(x) - s_A^-(x)].$$

3. The Catalyst library

Numerical modeling of chemical reactions written with the `Catalyst.jl` library is usually performed with the `DifferentialEquations.jl` [19] package. To work with it, the reaction system is converted into the desired problem type from this package. It contains a large number of numerical solution methods and additional functions. The obtained solutions can be visualized with the `Plots.jl` package by specifying the necessary parameters and time interval. Using `Catalyst.jl`, reaction systems can be represented as deterministic and stochastic models.

In terms of performance, the `Catalyst.jl` package significantly outperforms other chemical reaction network (CRN) modeling tools such as `BioNetGen`, `COPASI`, `GillesPy2`, `Matlab` and `SimBiology` [6]. The performance of models created with Catalyst depends on a variety of factors. For instance, Catalyst builds in all reaction conditions in the ordinary differential equations (ODE), which allows the compiler to optimize computation and reduces function call overhead.

Consider the simplest reversible reaction:

$$X + Y \underset{k^-}{\overset{k^+}{\rightleftarrows}} XY,$$

where X, Y and XY are the types of some objects, and k^+, k^- are reaction parameters. Parameters can be either constants and functions of time or other components of the system.

The `@reaction_network` macro is used to symbolize a chemical reaction.

First, we load the `Catalyst.jl` package using the command `using Catalyst`.

Then we assign the reaction to the variable `rn`, where the reaction is described using the `@reaction_network` macro:

```
rn = @reaction_network begin
1.0, X + Y --> XY
1.5, XY --> X + Y
end
```

Here, the proportionality constants k^+ and k^- are specified first, followed by the corresponding reaction, separated by comma. One reversible reaction is written as two irreversible reactions. If it is necessary to specify a reaction in which reactants are produced/destroyed from nothing, we write the necessary part as 0 (it is perceived as an empty set).

The `Catalyst.jl` package allows to use different ways of writing arrows. Unidirectional arrows can be written in both directions and with Unicode characters. Accordingly, one reaction can be represented as four equivalent variants:

```
rn = @reaction_network begin
  1.0, X + Y --> XY
  1.0, X + Y → XY
  1.0, XY ← X + Y
  1.0, XY <-- X + Y
end
```

As a result of executing the code, we will get:

$$X + Y \overset{1.0}{\rightarrow} XY,$$

$$X + Y \overset{1.0}{\rightarrow} XY,$$

$$X + Y \overset{1.0}{\rightarrow} XY,$$

$$X + Y \overset{1.0}{\rightarrow} XY.$$

Speaking of bidirectional arrows, both two-line and one-line entries are possible. The following variants of writing, which are also equivalent, are obtained. The first variant consisting of two reactions (forward and reverse):

```
rn = @reaction_network begin
  2.0, X + Y --> XY
  2.0, XY <-- X + Y
end
```

The second option represents a bidirectional reaction:

```
rn = @reaction_network begin
  (2.0, 2.0), X + Y <--> XY
end
```

The output will be the same in both cases:

$$X + Y \underset{2.0}{\overset{2.0}{\rightleftarrows}} XY.$$

The `Catalyst.jl` package also allows to use different ways to combine reactions. The combining of two or more reactions are considered:

- write both reactions on the same line – `1.0, S --> (P1, P2);`
- write each reaction on a separate line –

```
        1.0, S --> P1
        1.0, S --> P2
```

- combine reactions with different parameters – `(1.0, 2.0), (S1, S2) --> (P1,P2)`.

In the case where reversible reactions are to be combined, the following notation may occur:
`(1.0,(1.0,2.0)), S <--> (P1,P2);`
As a result, we get:

$$S \xrightarrow{1.0} P1,$$
$$S \xrightarrow{1.0} P2,$$
$$P1 \xrightarrow{1.0} S,$$
$$P2 \xrightarrow{2.0} S.$$

4. Example of one-dimensional model

Consider the birth–death model of kinetic reactions:

$$\begin{aligned} X &\xrightarrow{1} 2X, \\ X &\underset{100}{\overset{2}{\rightleftarrows}} 0. \end{aligned} \tag{3}$$

We set the differential equations in stochastic and deterministic forms that correspond to this chemical reaction (3). We write down the vectors describing the state of the system. In our case, they will be one-dimensional:

$$r^1 = 2 - 1 = 1,$$
$$r^2 = 0 - 1 = -1.$$

We consider the probabilities of the transitions. The first reaction is irreversible, so $s_1^- = 0$.

$$s_1^+ = 1x^1 = x,$$
$$s_2^+ = 2x^1 = 2x,$$
$$s_2^- = 100x^0 = 100.$$

Using the transition probabilities, we can write the Fokker–Planck equation:

$$\partial_t p(X,t) = -\sum_a \partial_a [A_a(x)P(X,t)] + \frac{1}{2}\sum_{a,b}[B_{ab}(X)P(X,t)],$$

where

$$A(X) = r^1 s_1^+ + r^2[s_2^+ - s_2^-] = x - 2x + 100 = 100 - x,$$
$$B(X) = r^1(r^1)^T s_1^+ + r^2(r^2)^T[s_2^+ - s_2^-] = x + 2x - 100 = 3x - 100.$$

We proceed to write a stochastic differential equation in Langevin form. To do this, we need to extract the square root of $B(x)$.

$$\mathrm{d}x = a(x)\,\mathrm{d}t + \sqrt{B(x)}\,\mathrm{d}W = (100 - x)\,\mathrm{d}t + \sqrt{3x - 100}\,\mathrm{d}W.$$

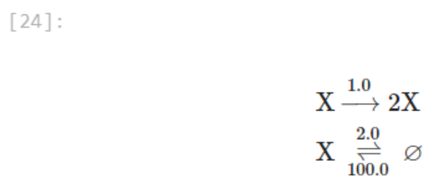


Figure 1. Result of the chemical reaction initialization code

By removing the stochastic term, we get a deterministic differential equation.

Now we will consider an example of writing and finding a solution to a birth–death model using `Catalyst.jl`. The system (3) is set by the variable `rn`:

```
rn = @reaction_network begin
  1.0, X --> 2X
  2.0, X --> 0
  100.0, 0 --> X
end
```

We get a system of chemical reactions (Fig. 1).

Here, the first reaction means reproduction of species, the second one means extinction, and the third one sets a constraint on the population size.

For numerical modeling we use the package `DifferentialEquations.jl`, which allows solving a wide range of DEs. The resulting reaction system is converted into a differential equation using the ODESystem method, and then the initial condition and the time interval on which the solution is sought are set:

```
rnsys = convert(ODESystem, rn)

@variables t
@species X(t)
u0 = [X => 10]
tspan = (0, 10.0)
symsys = structural_simplify(rnsys)
rnprob = ODEProblem(symsys, u0, tspan)

sol = solve(rnprob, Tsit5())
```

The macros `@variables` and `@species` are used here, where t is time and $X(t)$ is the population concentration (number of individuals) dependent on t.

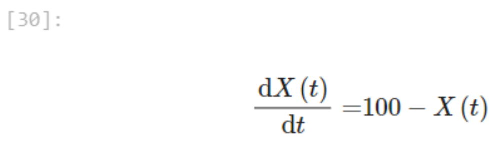
We can notice that the system of differential equations (Fig. 2) coincides with the theoretically derived deterministic part of the equation (4).

The results of modeling can be visualized using the Plots.jl package (Fig. 3).

It is also possible to convert the system of chemical reactions into a stochastic differential equation in Langevin form, which will have the form:

$$\frac{\mathrm{d}X}{\mathrm{d}t} = \big(100 - X(t)\big)\,\mathrm{d}t + 100\,\mathrm{d}W_1 - X(t)\,\mathrm{d}W_2.$$

In this case, the stochastics are different from those derived in (4).



```
[30]:
```

$$\frac{\mathrm{d}X(t)}{\mathrm{d}t} = 100 - X(t)$$

Figure 2. The result of code converting a death-birth chemical reaction systems into deterministic differential equations

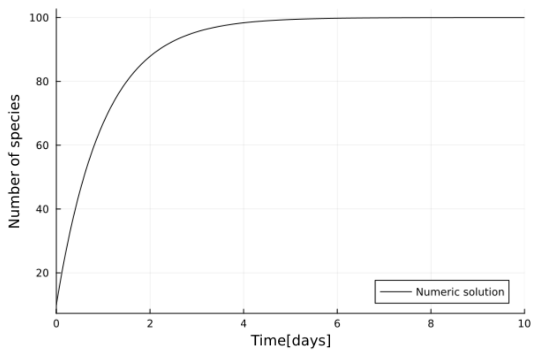


Figure 3. The birth–death model

```
sprob = SDEProblem(rn, u0, tspan)
sol_stoch = solve(sprob, EM(), dt = 0.001)
```

As a result, we get a graph of the solution (Fig. 4).

In addition, we can represent the model as a jump process. To define a jump process, we use the `JumpProblems` method. First, from the network of chemical reactions, we construct another type of problem with which transitions will be associated, namely we create a `DiscreteProblem` using Gillespie's direct method (`Direct()`), which sets constant transition rates. For the solution, a high-performance integrator `SSAStepper()` for pure jump problems (with constant transition rates) is passed to the `solve` function:

```
dprob = DiscreteProblem(rn, u0, tspan)
jprob = JumpProblem(rn, dprob, Direct())
jsol = solve(jprob, SSAStepper())
```

As a result, we get the solution graph (Fig. 5).

5. Example of a multidimensional model

Also, using this library we can set n-dimensional reactions. In general, the n-dimensional predator prey system can be written as:

$$\frac{\mathrm{d}x^i}{\mathrm{d}t} = x^i \left(b^i + \sum_{j=1}^{n} a^i_j x^j \right),$$

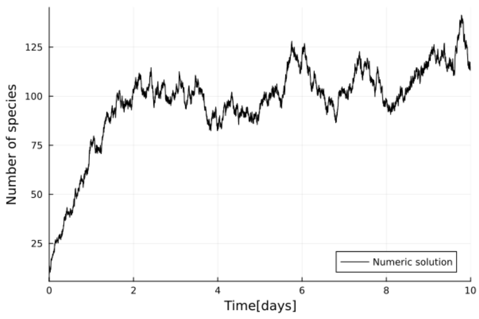


Figure 4. Stochastic birth–death model. Solution using the Euler–Maruyama method

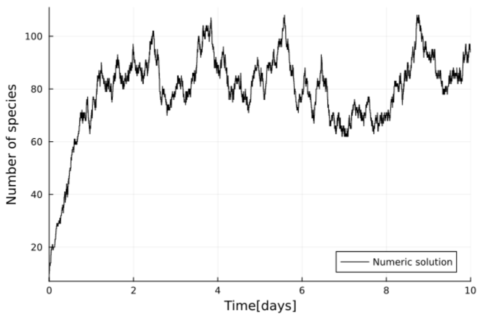


Figure 5. A stochastic birth–death model. Solution using the direct Gillespie method

where $i \neq j$, $\vec{x} = (x^1, x^2, \ldots, x^n)$—$n$ species, $\vec{b} = (b^1, b^2, \ldots, b^n)$—natural death or birth parameters (greater than zero for autotrophic species, less than zero for heterotrophic species); $\hat{A} := a^i_j$—parameters describing the interaction between species (if greater than zero, the individual is eaten, if less, the individual is born):

$$\hat{A} := a^i_j = \begin{pmatrix} 0 & a^1_2 & a^1_3 \\ a^2_1 & 0 & a^2_3 \\ a^3_1 & a^3_2 & 0 \end{pmatrix}.$$

Bazykin's work [20] examines various types of interactions between three populations, consider the producer-consultant-predator system (4).

$$\begin{aligned} X_i &\overset{p_{i0}}{\to} 2X_i, \\ X_i + X_j &\overset{p_{ij}}{\to} X_j, \end{aligned} \tag{4}$$

where $i, j = 1, \ldots, n$.

We set it with specific parameters using `Catalyst` and find the solution using the numerical method `Tsit5()`:

```
lv = @reaction_network begin
  1.5, X + Y --> 2*Y
  3, Y --> 0
  1.5, X + Z --> 2*X
  3, Z --> 2*Z
  end

  @variables t
  @species X(t) Y(t) Z(t)
  u0 = [X => 1.0, Y => 1.0, Z => 1.0]
  tspan = (0.0, 16.0)
  symsys = structural_simplify(lvsys)
  lvoprob = ODEProblem(symsys, u0, tspan)
  sol_lv_ode = solve(lvoprob, Tsit5())
```

[5]:

$$
\begin{aligned}
\frac{\mathrm{d}X\left(t \right)}{\mathrm{d}t} &= - X\left(t \right) - 1.5Y\left(t \right)X\left(t \right) + 1.5X\left(t \right)Z\left(t \right) \\
\frac{\mathrm{d}Y\left(t \right)}{\mathrm{d}t} &= - 3Y\left(t \right) + 1.5Y\left(t \right)X\left(t \right) \\
\frac{\mathrm{d}Z\left(t \right)}{\mathrm{d}t} &= 3Z\left(t \right) - 1.5X\left(t \right)Z\left(t \right)
\end{aligned}
$$

Figure 6. The result of code converting a three-dimensional predator–prey chemical reaction systems into deterministic differential equations

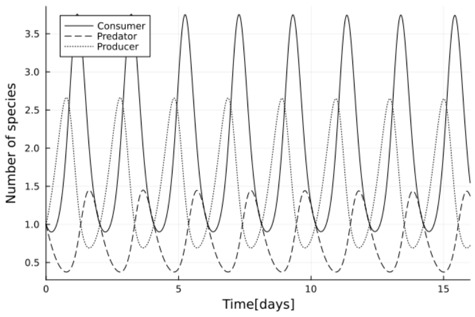


Figure 7. Solutions of the predator–prey three-dimensional system

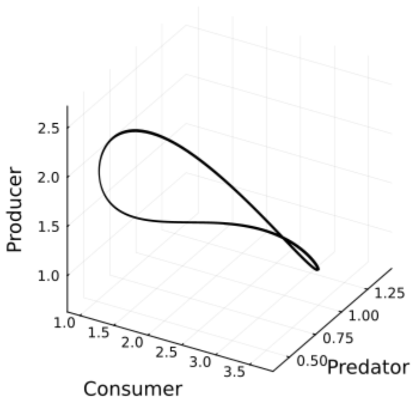


Figure 8. Phase portrait of the predator–prey three-dimensional system

In the form of differential equations, this system of reactions, transformed with `Catalyst`, looks as follows (Fig. 6).

As a result, using the `solve` method, we get a graph of the solution (Fig. 7), and we can display a phase portrait (Fig. 8).

This system can also be solved by set a stochastic problem using the function `SDEProblem` (see Fig. 9, 10):

```
lvoprob = SDEProblem(lvsys, u0, tspan)
```

We also display phase portraits (see Figs. 11, 12). Here additionally, a constraint was set to ensure that the number of individuals does not fall below zero:

```
function condition(u, t, integrator)
    any(u .< 0)
end

function affect!(integrator)
     integrator.u .= max.(integrator.u, 0)
end
```

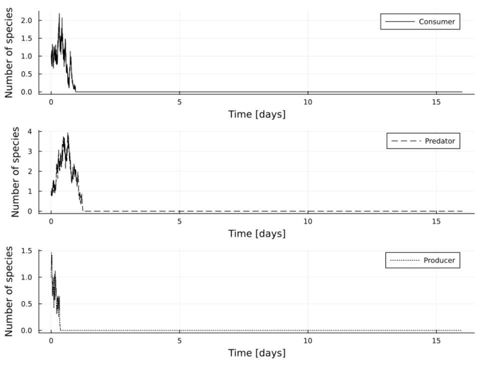


Figure 9. Solution of a three-dimensional stochastic predator–prey system. The case of extinction of all species

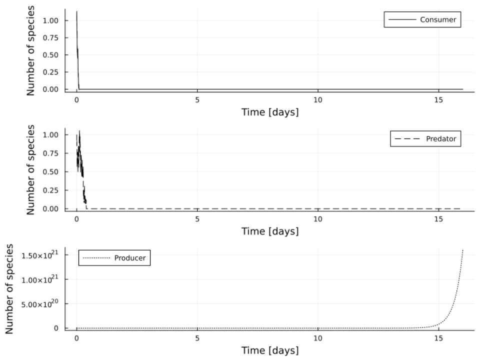


Figure 10. Solution of a three-dimensional stochastic predator–prey system. The case of unrestricted growth of the producer population

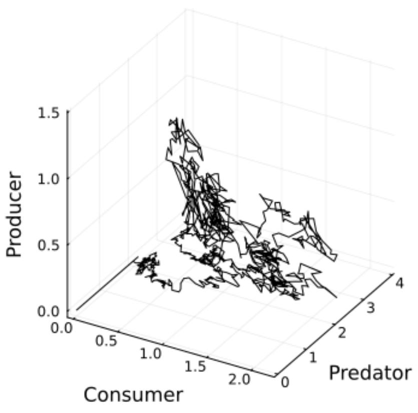


Figure 11. Phase portrait of a three-dimensional stochastic predator–prey system. Case of unrestricted growth of the producer population

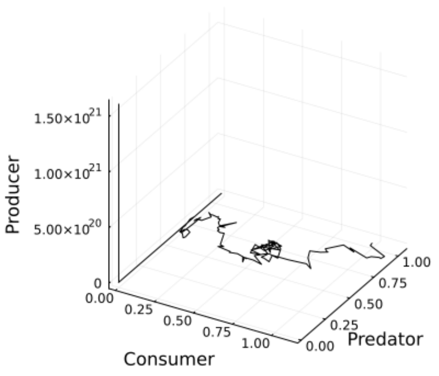


Figure 12. Phase portrait of a three-dimensional stochastic predator–prey system. The case of unrestricted growth of the producer population

```
cb = DiscreteCallback(condition, affect!)
```

```
sol_lv_sde = solve(lvsdeprob, EM(), dt = 0.001, callback = cb)
```

We can notice that in the stochastic case, the system can behave in two ways: all species go extinct or the consumers and predators go extinct and the producers multiply unrestrictedly.

6. Conclusion

The paper reviews the tools of the Catalyst.jl library for working with symbolic–numerical notation of kinetic equations, which allows to describe and analyze kinetic processes in a convenient and flexible form. Examples of constructing and solving one-dimensional and multidimensional models in the form of ODEs and SDEs are considered, specifically, birth–death and three-dimensional predator–prey models are constructed and their numerical solutions are found.

Future research plan include exploring the extended functionality of Catalyst.jl with the application of neural networks. For example, in this library it is possible to set reaction rate parameters not only as constants but also as neural networks. Also, complex CRN structures can be approximated using deep learning methods.

Author Contributions: Conceptualization, methodology, Dmitry S. Kulyabov; writing—original draft preparation, Ekaterina A. Demidova, Daria M. Belicheva; writing—review and editing, Victoria M. Shutenko, Anna V. Korolkova. All authors have read and agreed to the published version of the manuscript.

Funding: This work was carried out in the framework of grant support of the RUDN University, project 021934-0-000 (Anna V. Korolkova) and was supported by the program of strategic academic leadership of the RUDN University.

Data Availability Statement: Data sharing is not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gardiner, C. W. *Handbook of Stochastic Methods: for Physics, Chemistry and the Natural Sciences* (Springer Series in Synergetics, 1985).
2. Van Kampen, N. G. *Stochastic Processes in Physics and Chemistry* (Elsevier Science, 2011).
3. Lotka, A. J. *Elements of Physical Biology* 435 pp. (Williams and Wilkins Company, Baltimore, 1925).
4. Volterra, V. *Leçons sur la Théorie mathématique de la lutte pour la vie* French (Gauthiers-Villars, Paris, 1931).
5. Loman, T. E., Ma, Y., Ilin, V., Gowda, S., Korsbo, N., Yewale, N., Rackauckas, C. & Isaacson, S. A. *Catalyst: Fast Biochemical Modeling with Julia* Aug. 2022. doi:10.1101/2022.07.30.502135. bioRxiv: 2022.07.30.502135.
6. Loman, T. E., Ma, Y., Ilin, V., Gowda, S., Korsbo, N., Yewale, N., Rackauckas, C. & Isaacson, S. A. Catalyst: Fast and flexible modeling of reaction networks. *PLOS Computational Biology* **19** (ed Ouzounis, C. A.) e1011530.1–19. doi:10.1371/journal.pcbi.1011530 (Oct. 2023).
7. Bezanson, J., Karpinski, S., Shah, V. B. & Edelman, A. Julia: A Fast Dynamic Language for Technical Computing, 1–27. arXiv: 1209.5145 (2012).
8. Bezanson, J., Edelman, A., Karpinski, S. & Shah, V. B. Julia: A fresh approach to numerical computing. *SIAM Review* **59,** 65–98. doi:10.1137/141000671. arXiv: 1411.1607 (Jan. 2017).
9. Fedorov, A. V., Masolova, A. O., Korolkova, A. V. & Kulyabov, D. S. *Implementation of an analytical-numerical approach to stochastization of one-step processes in the Julia programming language* in *Workshop on information technology and scientific computing in the framework of the XI International Conference Information and Telecommunication Technologies and Mathematical Modeling of High-Tech Systems (ITTMM-2021)* (eds Kulyabov, D. S., Samouylov, K. E. & Sevastianov, L. A.) **2946** (Moscow, Apr. 2021), 92–104.
10. Roesch, E., Greener, J. G., MacLean, A. L., Nassar, H., Rackauckas, C., Holy, T. E. & Stumpf, M. P. H. *Julia for Biologists* 2021. doi:10.48550/ARXIV.2109.09973. arXiv: 2109.09973.
11. Pal, S., Bhattacharya, M., Dash, S., Lee, S.-S. & Chakraborty, C. A next-generation dynamic programming language Julia: Its features and applications in biological science. *Journal of Advanced Research,* 1–12. doi:10.1016/j.jare.2023.11.015 (Nov. 21, 2023).

12. Kulyabov, D. S. & Korol'kova, A. V. Computer Algebra in JULIA. *Programming and Computer Software* **47,** 133–138. doi:10.1134/S0361768821020079. arXiv: 2108.12301 (Mar. 2021).
13. Laidler, K. J. *Chemical Kinetics* 3rd ed. 531 pp. (Prentice Hall, Inc., Jan. 17, 1987).
14. Korolkova, A. V. & Kulyabov, D. S. One-step Stochastization Methods for Open Systems. *EPJ Web of Conferences* **226** (eds Adam, G., Buša, J. & Hnatič, M.) 02014.1–4. doi:10.1051/epjconf/202022602014 (Jan. 2020).
15. Doi, M. Stochastic theory of diffusion-controlled reaction. *Journal of Physics A: Mathematical and General* **9,** 1479–1495. doi:10.1088/0305-4470/9/9/009 (1976).
16. Schlögl, F. Chemical reaction models for non-equilibrium phase transitions. *Zeitschrift für Physik* **253,** 147–161. doi:10.1007/BF01379769 (1972).
17. Hnatič, M., Eferina, E. G., Korolkova, A. V., Kulyabov, D. S. & Sevastyanov, L. A. Operator Approach to the Master Equation for the One-Step Process. *EPJ Web of Conferences* **108** (eds Adam, G., Buša, J. & Hnatič, M.) 58–59. doi:10.1051/epjconf/201610802027. arXiv: 1603.02205 (2016).
18. Korolkova, A. V., Eferina, E. G., Laneev, E. B., Gudkova, I. A., Sevastianov, L. A. & Kulyabov, D. S. *Stochastization Of One-Step Processes In The Occupations Number Representation* in *Proceedings 30th European Conference on Modelling and Simulation* (ECMS, Regensburg, Germany, June 2016), 698–704. doi:10.7148/2016-0698.
19. Rackauckas, C. & Nie, Q. DifferentialEquations.jl - A Performant and Feature-Rich Ecosystem for Solving Differential Equations in Julia. *Journal of Open Research Software* **5.** doi:10.5334/jors.151 (2017).
20. Bazykin, A. D. *Nonlinear Dynamics of Interacting Populations* ed. by Khibnik, A. I. Ed. by Krauskopf, B. doi:10.1142/2284 (World Scientific, Singapore, May 1998).

Information about the authors

Ekaterina A. Demidova (Russian Federation)—Student of Department of Probability Theory and Cyber Security of RUDN University (e-mail: 1032216451v@rudn.ru, phone: +7 (495) 955-09-27, ORCID: 0009-0005-2255-4025)

Daria M. Belicheva (Russian Federation)—Student of Department of Probability Theory and Cyber Security of RUDN University (e-mail: 1032216453@rudn.ru, phone: +7 (495) 955-09-27, ORCID: 0009-0007-0072-0453)

Victoria M. Shutenko (Russian Federation)—Student of Department of Probability Theory and Cyber Security of RUDN University (e-mail: shutenkovika@yandex.ru, phone: +7 (495) 955-09-27, ORCID: 0000-0003-3922-4805)

Anna V. Korolkova (Russian Federation)—Docent, Candidate of Sciences in Physics and Mathematics, Associate Professor of Department of Probability Theory and Cyber Security of RUDN University (e-mail: korolkova-av@rudn.ru, phone: +7(495) 952-02-50, ORCID: 0000-0001-7141-7610, ResearcherID: I-3191-2013, Scopus Author ID: 36968057600)

Dmitry S. Kulyabov (Russian Federation)—Professor, Doctor of Sciences in Physics and Mathematics, Professor of Department of Probability Theory and Cyber Security of Peoples' Friendship University of Russia named after Patrice Lumumba (RUDN University); Senior Researcher of Laboratory of Information Technologies, Joint Institute for Nuclear Research (e-mail: kulyabov-ds@rudn.ru, phone: +7 (495) 952-02-50, ORCID: 0000-0002-0877-7063, ResearcherID: I-3183-2013, Scopus Author ID: 35194130800)

УДК 004.021:519.2:519.6
DOI: 10.22363/2658-4670-2024-32-3-306–318 EDN: FEMNAB

Символьно-численный подход для исследования кинетических моделей

Е. А. Демидова[1], Д. М. Беличева[1], В. М. Шутенко[1], А. В. Королькова[1], Д. С. Кулябов[1, 2]

[1] Российский университет дружбы народов, ул. Миклухо-Маклая, д. 6, Москва, 117198, Российская Федерация

[2] Объединённый институт ядерных исследований, ул. Жолио-Кюри, д. 6, Дубна, 141980, Российская Федерация

Аннотация. Наша группа достаточно долго исследует кинетические модели. Структура классических кинетических моделей описывается достаточно простыми предположениями о взаимодействии исследуемых сущностей. Также построение кинетических уравнений (как стохастических, так и детерминистических) основывается на простых последовательных шагах. Однако на каждом шаге исследователь должен манипулировать большим количеством элементов. А после получения дифференциальных уравнений возникает проблема их решения или исследования. Естественным образом напрашивается использование методологии символьно–численного подхода. Когда на входе представляется информационная модель исследуемой системы, представленная в каком-либо диаграммном виде. А в результате мы получаем системы дифференциальных уравнений (желательно, во всех возможных вариантах). Далее, в рамках этого процесса мы можем исследовать полученные уравнения (разнообразными методами). Ранее нами было предпринято несколько шагов в этом направлении, однако результаты нам показались несколько неудовлетворительными. На данный момент мы остановились на пакете Catalyst.jl, принадлежащему экосистеме языка Julia. Авторы пакета декларируют соответствие пакета области химической кинетики. Возможно ли исследовать с помощью этого пакета более сложные системы, мы сказать не можем. Поэтому исследование возможности применения данного пакета для наших моделей мы решили начать со стандартных задач химической кинетики. В результате мы можем резюмировать, что данный пакет видится нам наилучшим решением для символьно–численного исследования задач химической кинетики.

Ключевые слова: уравнения химической кинетики, стохастические дифференциальные уравнения, популяционные модели, одношаговые процессы