

One-Step Stochastic Processes Simulation Software Package

E. G. Eferina,^{*} A. V. Korolkova,[†] and M. N. Gevorkyan[‡]

*Department of Applied Probability and Informatics
Peoples' Friendship University of Russia
Miklukho-Maklaya str. 6, Moscow, 117198, Russia*

D. S. Kulyabov[§]

*Department of Applied Probability and Informatics
Peoples' Friendship University of Russia
Miklukho-Maklaya str. 6, Moscow, 117198, Russia and
Laboratory of Information Technologies
Joint Institute for Nuclear Research
Joliot-Curie 6, Dubna, Moscow region, 141980, Russia*

L. A. Sevastyanov[¶]

*Department of Applied Probability and Informatics
Peoples' Friendship University of Russia
Miklukho-Maklaya str. 6, Moscow, 117198, Russia and
Bogoliubov Laboratory of Theoretical Physics
Joint Institute for Nuclear Research
Joliot-Curie 6, Dubna, Moscow region, 141980, Russia***

Background: It is assumed that the introduction of stochastic in mathematical model makes it more adequate. But there is virtually no methods of coordinated (depended on structure of the system) stochastic introduction into deterministic models. Authors have improved the method of stochastic models construction for the class of one-step processes and illustrated by models of population dynamics. Population dynamics was chosen for study because its deterministic models were sufficiently well explored that allows to compare the results with already known ones.

Purpose: To optimize the models creation as much as possible some routine operations should be automated. In this case, the process of drawing up the model equations can be algorithmized and implemented in the computer algebra system. Furthermore, on the basis of these results a set of programs for numerical experiment can be obtained.

Method: The computer algebra system Axiom is used for analytical calculations implementation. To perform the numerical experiment FORTRAN and Julia languages are used. The method Runge–Kutta method for stochastic differential equations is used as numerical method.

Results: The program complex for creating stochastic one-step processes models is constructed. Its application is illustrated by the predator-prey population dynamic system.

Conclusions: Computer algebra systems are very convenient for the purposes of rapid prototyping in mathematical models design and analysis.

Keywords: stochastic differential equations; “predator–prey” model; master equation; Fokker–Planck equation; computer algebra software; Axiom system

I. INTRODUCTION

This work corresponds our research on mathematical models stochastization. This item is interesting due to

the following problems: the construction of population models from first principles and the introduction of the stochastic into such models (the population dynamics is studied because of similar models introduction in other areas).

The problem of stochastic term introduction arises during mathematical models stochastization. There are several ways to solve this problem. The easiest option is an in the deterministic equation. But when additive stochastic term is introduced some free parameters that require further definition appears. Furthermore, these stochastic terms usually interpreted as an external (rather than structural) random impact. In this regard, we used and improved the stochastic one-step processes models construction method, based on master equation [1, 2]. Stochastic differential equation is considered as its ap-

^{*} eg.eferina@gmail.com

[†] avkorolkova@gmail.com

[‡] mngevorkyan@sci.pfu.edu.ru

[§] yamadharm@gmail.com

[¶] leonid.sevast@gmail.com

^{**} Published in: E. G. Eferina, A. V. Korolkova, M. N. Gevorkyan, D. S. Kulyabov, L. A. Sevastyanov, One-Step Stochastic Processes Simulation Software Package, Bulletin of Peoples' Friendship University of Russia. Series “Mathematics. Information Sciences. Physics” (3) (2014) 46–59. ; Sources: <https://bitbucket.org/yamadharm/articles-2014-axiom-sdu>

proximate form. It allows to get the model equations from general principles. Furthermore, deterministic and stochastic parts are derived from the one equation so we can regard it as stochastic and deterministic parts consistency.

The aim of this work is the software complex development for rapid prototyping construction of stochastic one-step processes models. This complex consists of two blocks. The first block generates the equations of dynamic stochastic process model on the principles similar to chemical kinetic relations describing the investigated process. This block is implemented by means of the computer algebra system - system FriCAS, which is offshoot of Axiom.

The second block is used for the numerical analysis of the resulting model. For numerical solution of deterministic and stochastic models equations some Runge–Kutta different orders methods [3, 4] are used.

To illustrate the developed system the well-known population model predator–prey is used [5–7].

The structure of the paper is as follows. The basic notation and conventions are introduced in Section II. Section III is devoted to brief introduction to the one-step processes stochastization method. Further, in the Section IV the model under investigation is described. In the subsection IV A there is a brief reference to standard (deterministic) approach, and in the subsection IV B the stochastic extension of our model with the help of the one-step processes stochastization method is obtained.

In the Section V A we justify selection of the system, which implements the model equations generating unit. The actual interface of this part of the program complex is described in the Section V B.

The possibility of applying Runge–Kutta methods for the analysis of stochastic differential equations is considered in the Section VI. The software interface of the model equations numerical analysis unit is also described in this section. Calculations example is based on the predator–prey model.

II. NOTATIONS AND CONVENTIONS

1. We use abstract indices notation [8]. In this notation tensor as a whole object is denoted just as an index (e.g., x^i), components are denoted by underlined index (e.g., $x^{\underline{i}}$).
2. We will adhere to the following agreements. Latin indices of the middle of the alphabet (i, j, k) will apply to the space of the system state vectors. Latin indices from the beginning of the alphabet (a) will relate to the Wiener process space. Latin indices from the end of the alphabet (p, q) will refer to the indices of the Runge–Kutta method. Greek indices (α) will set a number of different interactions in kinetic equations.

3. A Dot over a symbol denotes differentiation with respect to time.
4. The comma in the index denotes partial derivative with respect to corresponding coordinate.

III. ONE-STEP PROCESSES MODELING

Let's briefly review the method of one-step processes stochastization on the basis of [9].

We understand one-step processes as Markov processes with continuous time with values in the domain of integers, which transition matrix allows only transitions between neighbouring portions. Also, these processes are known as birth-and-death processes.

One-step processes are subject to the following conditions:

1. If at the moment t the system is in state $i \in \mathbb{Z}_{\geq 0}$, then the probability of transition to state $i + 1$ in time interval $[t, t + \Delta t]$ is equal to $k^+ \Delta t + o(\Delta t)$.
2. If at time moment t the system is in state $i \in \mathbb{Z}_+$, then the probability of transition to state $i - 1$ in the time interval $[t, t + \Delta t]$ is equal to $k^- \Delta t + o(\Delta t)$.
3. The probability of transition to a state other than the neighbouring is equal to $o(\Delta t)$.
4. The probability to remain in the same state is equal to $1 - (k^+ + k^-) \Delta t + o(\Delta t)$.
5. State $i = 0$ is an absorbing boundary.

The idea of the one-step processes stochastization method is as follows. Based on the patterns of interaction we construct a master kinetic equation, expand it into a series, leaving only the terms up to and including the second derivative. The resulting equation is the Fokker–Planck equation. In order to get more convenient model we record corresponding Langevin equation. In fact, as we shall see, from the patterns of interaction we will immediately obtain the coefficients of the Fokker–Planck equation (and accordingly, the Langevin equation), so for practical use of the method there is no need to construct the master kinetic equation.

A. Interaction schemes

We will describe the state of the system by a state vector x^i in \mathbb{R}^n , where n is the system dimension (the state vector is considered as the set of mathematical values, fully describing system). The operator $n_j^i \in \mathbb{Z}_{\geq 0}^n \times \text{set } \mathbb{Z}_{\geq 0}^n$ defines the state of the system before the interaction and the operator $m_j^i \in \mathbb{Z}_{\geq 0}^n \times \mathbb{Z}_{\geq 0}^n$ — after the interaction. The result of interaction is a system transition to another state [1, 10].

There are s kinds of different interactions that may happen in the system, $s \in \mathbb{Z}_+$. So, instead of n_j^i and m_j^i let's consider the operators $n_j^{i\alpha} \in \mathbb{Z}_{\geq 0}^n \times \mathbb{Z}_{\geq 0}^n \times \mathbb{Z}_{\geq 0}^s$ and $m_j^{i\alpha} \in \mathbb{Z}_{\geq 0}^n \times \mathbb{Z}_{\geq 0}^n \times \mathbb{Z}_{\geq 0}^s$.

System elements interaction will be described with the interaction schemes similar to chemical kinetic schemes [11]:

$$n_j^{i\alpha} x^j \xrightleftharpoons[k_\alpha]{k_\alpha^+} m_j^{i\alpha} x^j, \quad (1)$$

Here Greek indexes specify the number of interactions and Latin ones specify dimensionality of the system. The state change is given by the operator

$$r_j^{i\alpha} = m_j^{i\alpha} - n_j^{i\alpha}. \quad (2)$$

Thus, one step interaction $\underline{\alpha}$ in forward and opposite directions can be written as

$$\begin{aligned} x^i &\rightarrow x^i + r_j^{i\alpha} x^j, \\ x^i &\rightarrow x^i - r_j^{i\alpha} x^j. \end{aligned}$$

We can write (1) not in the form of vector equations, but in the more traditional form sums:

$$n_j^{i\alpha} x^j \delta_i \xrightleftharpoons[k_\alpha]{k_\alpha^+} m_j^{i\alpha} x^j \delta_i, \quad (3)$$

where $\delta_i = (1, \dots, 1)$.

Also, we will use the following notation:

$$n^{i\alpha} := n_j^{i\alpha} \delta^j, \quad m^{i\alpha} := m_j^{i\alpha} \delta^j, \quad r^{i\alpha} := r_j^{i\alpha} \delta^j.$$

B. Master equation

Transition probabilities per unit of time from the state x^i to the state $x^i + r_j^{i\alpha} x^j$ (to the state $x^i - r_j^{i\alpha} x^j$) are proportional to the number of x^i combination from a set of $n^{i\alpha}$ elements (of x^i — combinations from a set of $m^{i\alpha}$) and are given by:

$$\begin{aligned} s_\alpha^+ &= k_\alpha^+ \prod_{i=1}^n \frac{x^i!}{(x^i - n^{i\alpha})!}, \\ s_\alpha^- &= k_\alpha^- \prod_{i=1}^n \frac{x^i!}{(x^i - m^{i\alpha})!}. \end{aligned} \quad (4)$$

Thus, the general form of the master kinetic equation for the states vector x^i (it changes by $r_j^{i\alpha} x^j$ per step), takes the form:

$$\frac{\partial p(x^i, t)}{\partial t} =$$

$$\begin{aligned} &= \sum_{\alpha=1}^m \left\{ \left[s_\alpha^-(x^i + r^{i\alpha}, t) p(x^i + r^{i\alpha}, t) - s_\alpha^+(x^i) p(x^i, t) \right] + \right. \\ &\quad \left. + \left[s_\alpha^+(x^i - r^{i\alpha}, t) p(x^i - r^{i\alpha}, t) - s_\alpha^-(x^i) p(x^i, t) \right] \right\}. \quad (5) \end{aligned}$$

C. Fokker–Planck equation

With the help of the Kramers–Moyal expansion, the Fokker–Planck equation [11] is obtained. For this purpose we will make several assumptions:

1. there are only small jumps, ie $s_\alpha(x^i)$ is a slowly varying function with the change of x^i ;
2. $p(x^i, t)$ also slowly changes with the change of x^i .

Then in Fokker–Planck equation (5) one can shift from the point $(x^i \pm r_j^{i\alpha} x^j)$ to the point x^i , and by expanding the right-hand side in a Taylor series and dropping terms of order higher than the second, we obtain Fokker–Planck equation:

$$\frac{\partial p}{\partial t} = -\partial_i [A^i p] + \frac{1}{2} \partial_i \partial_j [B^{ij} p], \quad (6)$$

where

$$\begin{aligned} A^i &:= A^i(x^k, t) = r^{i\alpha} [s_\alpha^+ - s_\alpha^-], \\ B^{ij} &:= B^{ij}(x^k, t) = r^{i\alpha} r^{j\alpha} [s_\alpha^+ - s_\alpha^-], \quad \underline{\alpha} = \overline{1, m}. \end{aligned} \quad (7)$$

As seen from (7), the coefficients of the Fokker–Planck equation can be obtained directly from (2) and (4), i.e. in practical calculations, there is no need to write the master equation.

D. Langevin equation

The Langevin equation corresponds to the Fokker–Planck equation:

$$dx^i = a^i dt + b_a^i dW^a, \quad (8)$$

where $a^i := a^i(x^k, t)$, $b_a^i := b_a^i(x^k, t)$, $x^i \in \mathbb{R}^n$ — is the system state vector, $W^a \in \mathbb{R}^m$ — m -dimensional Wiener process. Wiener process is implemented as $dW = \varepsilon \sqrt{dt}$, where $\varepsilon \sim N(0, 1)$ is normal distribution with average 0 and variance 1. Latin indexes from the middle of the alphabet denote the values related to the state vectors (dimension of the space is n), and Latin indexes from the beginning of the alphabet denote the values related to the Wiener process vector (dimension of the space is $m \leq n$).

The connection between the equation (6) and (8) expressed by the following relationships:

$$A^i = a^i, \quad B^{ij} = b_a^i b_a^j.$$

We will use Ito interpretation. Under the Ito interpretation, differential of complex functions does not obey the standard formulas of analysis. To calculate it *rule* or *Ito lemma* are used.

Let $f := f(x^k, t)$ is a function of a random process $x^k(t)$, $f \in C^2$. Then the formula of the differential is [12]:

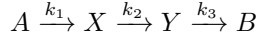
$$df = \left[\partial_t f + a^i f_{,i} + \frac{1}{2} b_a^i b^{ia} f_{,ij} \right] dt + b_a^i f_{,i} dW^a,$$

where $f := f(x^k, t)$, $a^i := a^i(x^k, t)$, $b_a^i := b_a^i(x^k, t)$, and $dW^a := dW^a(t)$.

IV. PREDATOR-PREY MODEL

A. Deterministic predator-prey model

Systems with the interaction of two predator-prey populations types are extensively studied and there are a lot of various models for these systems. The very first predator-prey model is considered to be a model which was obtained independently by A. Lotka and V. Volterra. Lotka in [13] described some hypothetical chemical reaction:



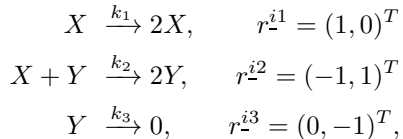
where X, Y are intermediates substances, coefficients k_1, k_2, k_3 are rates of chemical reactions, A is a initial reagent, and B is a resultant. As a result was a system of differential equations:

$$\begin{cases} \dot{x} = k_1 x - k_2 xy, \\ \dot{y} = k_2 xy - k_3 y. \end{cases}$$

This system is identical to the system of differential equations, obtained by Volterra, who considered the growth mechanism of two populations with predator-prey interaction type. In order to get equations [5] Volterra made a series of idealized assumptions about nature of intraspecific and interspecific relationships in the predator-prey system.

B. Stochastic predator-prey model

Consider a model of predator-prey system, consisting of two individuals species, one of which hunts, second is provided with inexhaustible food resources. Let's introduce the notation, where X is a prey and Y is a predator, then we can write the possible processes (3) for the state vector $x^i = (X, Y)^T$ [14–17]:



which have the following interpretation. The first relation means that the prey which eats the food unit immediately reproduced. The second relation describes the case when predator absorbs the prey and than it is instantaneous reproduced. Only such possibility of prey death is considered. Last ratio is a natural predator death.

All processes are irreversible, so $s_{\alpha}^- = 0$, and

$$\begin{aligned} s_1^+(x, y) &= k_1 \frac{x!}{(x-1)!} \frac{y!}{y!} = k_1 x, \\ s_2^+(x, y) &= k_2 \frac{x!}{(x-1)!} \frac{y!}{(y-1)!} = k_2 xy, \\ s_3^+(x, y) &= k_3 \frac{x!}{x!} \frac{y!}{(y-1)!} = k_3 y. \end{aligned}$$

With the help of the formula (6) we have the Fokker-Planck equation:

$$\frac{\partial p(x, y)}{\partial t} = -\partial_i (A^i(x, y)p(x, y)) + \frac{1}{2} \partial_i \partial_j (B^{ij}(x, y)p(x, y)),$$

where

$$\begin{aligned} A^i(x, y) &= s_{\alpha}^+(x, y) r^{i\alpha}, \\ B^{ij}(x, y) &= s_{\alpha}^+(x, y) r^{i\alpha} r^{j\alpha}. \end{aligned}$$

As a result:

$$\begin{aligned} A^i(x, y) &= \begin{pmatrix} 1 \\ 0 \end{pmatrix} k_1 x + \begin{pmatrix} -1 \\ 1 \end{pmatrix} k_2 xy + \\ &\quad + \begin{pmatrix} 0 \\ -1 \end{pmatrix} k_3 y = \begin{pmatrix} k_1 x - k_2 xy \\ k_2 xy - k_3 y \end{pmatrix}, \\ B^{ij}(x, y) &= \begin{pmatrix} 1 \\ 0 \end{pmatrix} (1, 0) k_1 x + \begin{pmatrix} -1 \\ 1 \end{pmatrix} (-1, 1) k_2 xy + \\ &\quad + \begin{pmatrix} 0 \\ -1 \end{pmatrix} (0, -1) k_3 y = \begin{pmatrix} k_1 x + k_2 xy & -k_2 xy \\ -k_2 xy & k_2 xy + k_3 y \end{pmatrix}. \end{aligned} \quad (9)$$

In order to write a stochastic differential equation in Langevin form (8) for predator-prey model, it is enough to take the square root of the resulting matrix B^{ij} in Fokker-Planck equation.

$$\begin{aligned} d \begin{pmatrix} x \\ y \end{pmatrix} &= \begin{pmatrix} k_1 x - k_2 xy \\ k_2 xy - k_3 y \end{pmatrix} dt + b_a^i \begin{pmatrix} dW^1 \\ dW^2 \end{pmatrix}, \\ b_a^i b_a^{ja} &= B^{ij} = \begin{pmatrix} k_1 x + k_2 xy & -k_2 xy \\ -k_2 xy & k_2 xy + k_3 y \end{pmatrix}. \end{aligned}$$

It should be noted that the specific form of the matrix b_a^i is not written out because of the extreme awkwardness of the expression. However, with further studies we will need not actually matrix b_a^i , but its square, i.e. the matrix B^{ij} .

Such a choice of these numerical values for approximation is made because the Wiener process is implemented as $dW = \varepsilon\sqrt{dt}$. You should also pay attention to double summation in the third term of both numerical scheme formulas as well as the fact that each number J^1, \dots, J^n should be generated separately.

The method coefficients, as well as for the classical analogue, can be grouped into a table called the *Butcher table*:

$$\begin{array}{c|c} R_{ij} & \hat{R}_{ij} \\ \hline r_j & \hat{r}_j \end{array}$$

For calculations we used a method with the table

$$\begin{array}{c|ccc|ccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2/3 & 0 & 0 & 2/3 & 0 & 0 & 0 \\ -1 & 1 & 0 & -1 & 1 & 0 & 0 \\ \hline 0 & 3/4 & 1/4 & 0 & 3/4 & 1/4 & 0 \end{array}$$

B. Software implementation description

The purposes of the programs complex were to automate the SDE coefficients A^i and B^{ij} computation with the help of general principles described above, and to find a numerical solution of the equation obtained by means of stochastic Runge–Kutta methods. From a programming standpoint we can derive three subtasks:

1. coefficients A^i and B^{ij} generation using the computer algebra system;
2. generation of source code in languages Fortran and Julia, implementing the SDE on the basis of the coefficients, saved as a text file;
3. writing subroutines/functions implementing stochastic Runge–Kutta methods in Fortran and Julia, and their subsequent compilation together with automatically generated source codes.

As a result of its work Axiom module creates a text file which contains the coefficients A^i and B^{ij} in the following form:

```
# A
A[1]
...
A[N]
# B
B[1,1] B[1,2] .. B[1,N]
...
B[N,1] B[N,2] .. B[N,N]
```

Matrix $b_\alpha^i = \sqrt{b_\alpha^i b^{j\alpha}} = \sqrt{B^{ij}}$ is calculated numerically with the help of the singular value matrix decomposition (a subroutine DGESVD from library LAPACK is used).

For the second subtask scripting language Python was chosen (version 3). This language has a wide set of tools to work with strings and text files. Except matrices A^i and B^{ij} additional information about the mathematical model was specified as dictionary (standard data type in Python), with model name, list of variables, list of parameters, initial values of variables, parameters values and parameters of the numerical method (integration section and step size).

On the basis of these data, the script automatically generates two files `functions.f90` and `main.f90`, where the first is a module with functions defining the SDE, and the second one is a main program file. While compiling these files the third additional module with auxiliary procedures with Stochastic Runge–Kutta method is added.

C. The numerical experiment description

For the programs complex work verification a well-known predator–prey model was chosen with vector a^i components

$$a^1 = \alpha x - \beta xy, \quad a^2 = -\gamma y + \delta xy$$

and matrix B^{ij} :

$$\begin{bmatrix} \alpha x + \beta xy & -\beta xy \\ -\beta xy & \beta xy + \gamma x \end{bmatrix},$$

x is the number of preys, y is the number of predators. Coefficients also have the following physical (biological) meaning: α is the growth rate of the prey population, β is a frequency of predators and prey meetings, γ is an intensity of predators death or migration in a lack of preys, δ is a predator population growth rate on the assumption of the excess of the prey.

During numerical simulations it was taken into account that the value of variables x, y could not be less than zero (program stop working when one of the variables becomes equal to zero).

Numerical simulation shows that the addition of stochastic to the classical predator–prey model leads to the fact that after a certain time death of one of the competing species comes. So, for the following parameters: $\alpha = 10$, $\beta = 1, 5$, $\gamma = 8, 5$, $\delta = 1, 8$ and the initial values: $x = 9, 7$, $x = 6, 77$, victims are first to die, and after that predators die due to lack. This case is illustrated in Fig. 2. For comparison in Fig. 3 is a graph for the deterministic case.

Under other conditions ($\alpha = 10$, $\beta = 1, 5$, $\gamma = 8, 5$, $\delta = 0, 5$, $x = 22$, $y = 6, 76$) predators die, and the number of victims is increasing rapidly, as for their model assumes an infinite source of food. Graphics for this case are shown in Fig. 4, and Fig. 5 shown for comparison with

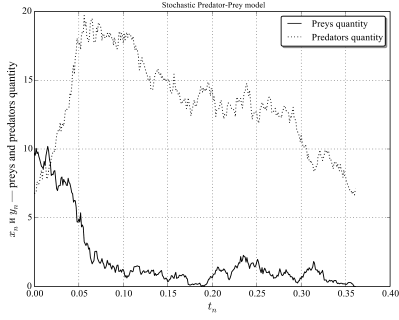


Figure 2. Stochastic predator-prey model, prey die

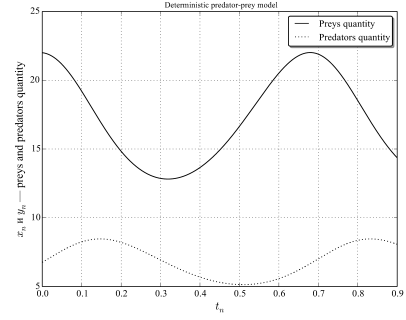


Figure 5. Deterministic predator-prey model

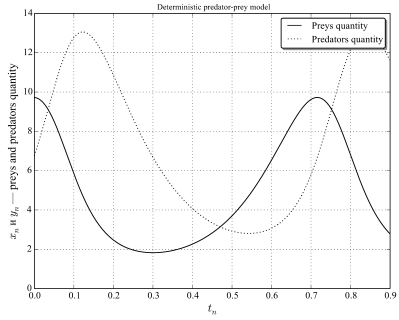


Figure 3. Deterministic predator-prey model

deterministic case.

VII. CONCLUSIONS

This work demonstrates the application of the developed initial physical system formalization method. The system is presented in the form of one or more one-step processes. Formalization of the system is done by introducing the evolution operator. Wherein the analytical description of the model requires a lot of routine operations. To simplify the work we propose to use the computer algebra system (Axiom fork FriCAS).

We have developed an analytical software package block that receives inlet evolution operator and produces the SDE, which describes the original model. For numerical studies of obtained SDE system a second software unit that converts the resulting system of equations into the program code in fortran and gives its numerical solution was developed. Thus, the software system is applicable for both analytical and numerical study of the original model.

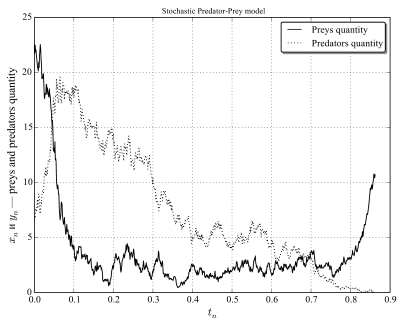


Figure 4. Stochastic predator-prey model, predators die

Currently the software package does not cover all possibilities, incorporated in the proposed method of formalizing the original physical system. Since the original system description uses ODE, we should introduce the boundary conditions by ties or indicator functions. Partial differential equations can help to solve this problem. Further objective is the development of a complete software complex for a method of one-step original physical system model construction.

-
- [1] A. V. Demidova and D. S. Kulyabov. The Introduction of an Agreed Term in the Equation of Stochastic Population Model. *Bulletin of Peoples Friendship University of Russia. Series Mathematics. Information Sciences. Physics*, (3):69–78, 2012. In Russian.
- [2] Anastsasiya Vyacheslavovna Demidova, Leonid Antonovich Sevastianov, and Dmitry Sergeevich Kulyabov. Application of Stochastic Differential Equations to Model Population Systems. In *Third International Conference on Mathematical Modelling of Social and Economical Dynamics MMSED-2010*, pages 92–94. Russian State Social University, 2010. In Russian.
- [3] Kristian Debrabant and Andreas Rößler. Classification of Stochastic Runge–Kutta Methods for the Weak Approximation of Stochastic Differential Equations. *Mathematics and Computers in Simulation*, 77(4):408–420, 2008. ISSN 0378-4754. doi: <http://dx.doi.org/10.1016/j.matcom.2007.04.016>.
- [4] A. Tocino and R. Ardanuy. Runge–Kutta Methods for Numerical Solution of Stochastic Differential Equations. *Journal of Computational and Applied Mathematics*, 138(2):219–241, 2002. ISSN 0377-0427. doi: [http://dx.doi.org/10.1016/S0377-0427\(01\)00380-6](http://dx.doi.org/10.1016/S0377-0427(01)00380-6). URL <http://www.sciencedirect.com/science/article/pii/S0377042701003806>.
- [5] Vito Volterra. *Mathematical Theory of the Struggle for Existence*. Nauka, 1976.
- [6] A. D. Bazykin. *Nonlinear Dynamics of Interacting Populations*. World Scientific, 1998.
- [7] A. S. Bratus, A. S. Novozhilov, and A. P. Platonov. *Dynamical systems and biology models*. Fizmatlit, M., 2010. In Russian.
- [8] Roger Penrose and Wolfgang Rindler. *Spinors and Space-Time: Two-Spinor Calculus and Relativistic Fields*, volume 1. Cambridge University Press, 1984.
- [9] A. V. Demidova, M. N. Gevorkyan, A. D. Egorov, A. V. Korolkova, D. S. Kulyabov, and L. A. Sevastyanov. Influence of stochastization to one-step model. *Bulletin of Peoples’ Friendship University of Russia. Series Mathematics. Information Sciences. Physics*, (1):71–85, 2014. In Russian.
- [10] Anastsasiya Vyacheslavovna Demidova, Anna Vladislavovna Korolkova, Dmitry Sergeevich Kulyabov, and Leonid Antonovich Sevastianov. The Method of Stochastization of One-Step Processes. In *Mathematical Modeling and Computational Physics*, page 67. JINR, 2013.
- [11] C. W. Gardiner. *Handbook of Stochastic Methods: for Physics, Chemistry and the Natural Sciences*. Springer Series in Synergetics, 1985.
- [12] B. K. Øksendal. *Stochastic Differential Equations: An Introduction with Applications*. Springer, Berlin, 2003.
- [13] A. J. Lotka. *Elements of Physical Biology*. BiblioBazaar, 2011. ISBN 9781178508116. URL <http://books.google.ru/books?id=tFN9pwAACAAJ>.
- [14] A. V. Demidova. The Equations of Population Dynamics in the Form of Stochastic Differential Equations. *Bulletin of Peoples’ Friendship University of Russia. Series Mathematics. Information Sciences. Physics*, (1):67–76, 2013. In Russian.
- [15] A. V. Demidova. The method of stochastization of mathematical models for the example of the “predator–prey”. In *A Scientific Session NRNU MEPHI-2013*, page 127, 2013. In Russian.
- [16] Kulyabov D. S. Korolkova, A. V. Methods of Stochastization of Mathematical Models on the Example of Peer to Peer Networks. In *A Scientific Session NRNU MEPHI-2013*, page 131, Moscow, 2013. MEPHI. In Russian.
- [17] Anastsasiya Vyacheslavovna Demidova, Dmitry Sergeevich Kulyabov, and Leonid Antonovich Sevastianov. The agreed stochastic term in population models. In *XI Belarusian Mathematical Conference*, page 39, Minsk, 2012. Institute of Mathematics of the National Academy of Sciences of Belarus. In Russian.

Программный комплекс стохастического моделирования одношаговых процессов

Е. Г. Еферина,^{*} А. В. Королькова,[†] and М. Н. Геворкян[‡]

*Кафедра прикладной информатики и теории вероятностей,
Российский университет дружбы народов,
ул. Миклухо-Маклая, д.6, Москва, Россия, 117198*

Д. С. Кулябов[§]

*Кафедра прикладной информатики и теории вероятностей,
Российский университет дружбы народов,
ул. Миклухо-Маклая, д.6, Москва, Россия, 117198 and
Лаборатория информационных технологий,
Объединённый институт ядерных исследований,
ул. Жолио-Кюри 6, Дубна,
Московская область, Россия, 141980*

Л. А. Севастьянов[¶]

*Кафедра прикладной информатики и теории вероятностей,
Российский университет дружбы народов,
ул. Миклухо-Маклая, д.6, Москва, Россия, 117198 and
Лаборатория теоретической физики,
Объединённый институт ядерных исследований,
ул. Жолио-Кюри 6, Дубна,
Московская область, Россия, 141980***

Предпосылки: Нашим коллективом разработана методика согласованного (зависящего от структуры системы) введения стохастики в детерминистические модели. На данном этапе методика ограничена классом одношаговых процессов.

Цель: Для оптимизации работы по созданию моделей следует автоматизировать как можно больше рутинных операций. В данном случае процесс составления уравнений модели можно алгоритмизировать и реализовать в системе компьютерной алгебры. Кроме того, на базе этих результатов можно получить и набор программ для проведения численного эксперимента.

Методы: Для реализации аналитических расчётов используется система компьютерной алгебры Axiom. Для проведения численного эксперимента используются языки FORTRAN и Julia. В качестве численного метода используется метод Рунге–Кутты для стохастических дифференциальных уравнений.

Результаты: Создан программный комплекс для создания стохастических моделей одношаговых процессов. Проиллюстрировано его применение на примере системы популяционной динамики типа «хищник–жертва». Детерминистические модели для таких процессов достаточно хорошо исследованы, что позволяет сравнить полученные результаты с уже известными.

Выводы: Системы компьютерной алгебры очень удобны для целей быстрого прототипирования при создании и исследовании математических моделей.

Keywords: стохастические дифференциальные уравнения; модель «хищник–жертва»; основное кинетическое уравнение; уравнение Фоккера–Планка; системы компьютерной алгебры; система Axiom

I. ВВЕДЕНИЕ

Данная работа находится в русле проводимых нами исследований по стохастизации математических моделей. Данная тематика интересна из-за следующих проблем: построение популяционных моделей из первых принципов и введение стохастики в модели данного вида (следует заметить, что обратились мы к популяционной динамике потому, что нами исследовались схожие модели в других областях).

При стохастизации математических моделей возникает проблема введения стохастического члена. Существует несколько способов сделать это. Самый простой вариант — аддитивное добавление стохастическо-

^{*} eg.eferina@gmail.com

[†] avkorolkova@gmail.com

[‡] mngevorkyan@sci.pfu.edu.ru

[§] yamadharm@gmail.com

[¶] leonid.sevast@gmail.com

^{**} Опубликовано в: One-Step Stochastic Processes Simulation Software Package / E. G. Eferina, A. V. Korolkova, M. N. Gevorkyan, D. S. Kulyabov, L. A. Sevastyanov // Bulletin of Peoples' Friendship University of Russia. Series «Mathematics. Information Sciences. Physics». — 2014. — No. 3. — P. 46–59. ; Исходные тексты: <https://bitbucket.org/yamadharm/articles-2014-axiom-sdu>

го члена к детерминистическому уравнению. Однако при таком введении возникают свободные параметры, требующие дальнейшего определения из дополнительных соображений. Кроме того, данные стохастические члены обычно интерпретируются как внешнее (а не структурное) случайное воздействие. В связи с этим мы использовали и усовершенствовали метод построения стохастических моделей одношаговых процессов на основе основного кинетического уравнения [1, 2]. Стохастическое дифференциальное уравнение рассматривается как его приближённая форма. Это позволяет получить модельные уравнения из общих принципов. Кроме того, детерминистическая и стохастическая части получаются из одного и того же уравнения, что рассматривается нами как согласованность стохастической и детерминистической частей.

Целью данной работы является разработка программного комплекса для быстрого прототипирования построения стохастических моделей одношаговых процессов. Данный комплекс состоит из двух блоков. Первый блок по соотношениям, аналогичным соотношениям химической кинетики, описывающим исследуемый процесс, генерирует уравнения динамической стохастической модели процесса. Этот блок реализован на системе компьютерной алгебры. В качестве системы для реализации выбрана система FriCAS — ответвление системы Axiom.

Второй блок служит для численного анализа полученной модели. Для численного решения уравнений детерминированной и стохастической моделей используются методы Рунге–Кутты разных порядков [3, 4].

Для демонстрации работы разработанного комплекса используется хорошо известная популяционная модель «хищник–жертва» [5–7].

Структура статьи следующая. В разделе II введены основные обозначения и соглашения. В разделе III даётся краткое введение в метод стохастизации одношаговых процессов. Далее, в разделе IV описывается исследуемая модель. При этом в подразделе IV A даётся краткая справка по стандартному (детерминистическому) подходу, а в подразделе IV B мы получаем стохастическое расширение данной модели по методу стохастизации одношаговых процессов.

В разделе V A обосновывается выбор системы для реализации блока генерации уравнений модели. В разделе V B описывается собственно интерфейс работы с этой частью программного комплекса.

В разделе VI рассматривается возможность применения методов Рунге–Кутты для анализа стохастических дифференциальных уравнений и описывается программный интерфейс блока численного анализа модельных уравнений. Пример расчётов основан на модели «хищник–жертва».

II. ОБОЗНАЧЕНИЯ И СОГЛАШЕНИЯ

1. В работе используется нотация абстрактных индексов [8]. В данной нотации тензор как целостный объект обозначается просто индексом (например, x^i), компоненты обозначаются подчёркнутым индексом (например, $x^{\dot{i}}$).
2. Будем придерживаться следующих соглашений. Латинские индексы из середины алфавита (i, j, k) будут относиться к пространству векторов состояний системы. Латинские индексы из начала алфавита (a) будут относиться к пространству винеровского процесса. Латинские индексы из конца алфавита (p, q) будут относиться к индексам метода Рунге–Кутты. Греческие индексы (α) будут задавать количество разных взаимодействий в кинетических уравнениях.
3. Точкой над символом обозначается дифференцирование по времени.
4. Запятой в индексе обозначается частная производная по соответствующей координате.

III. МОДЕЛИРОВАНИЕ ОДНОШАГОВЫХ ПРОЦЕССОВ

Дадим краткий обзор метода стохастизации одношаговых процессов на базе работы [9].

Под одношаговыми процессами мы будем понимать марковские процессы с непрерывным временем, принимающие значения в области целых чисел, матрица перехода которых допускает только переходы между соседними участками. Также эти процессы известны под названиями процессов рождения–гибели.

Одношаговые процессы подчиняются следующим условиям.

1. Если в момент времени t система находится в состоянии $i \in \mathbb{Z}_{\geq 0}$, то вероятность перехода в состояние $i + 1$ в интервале времени $[t, t + \Delta t]$ равна $k^+ \Delta t + o(\Delta t)$.
2. Если в момент времени t система находится в состоянии $i \in \mathbb{Z}_+$, то вероятность перехода в состояние $i - 1$ в интервале времени $[t, t + \Delta t]$ равна $k^- \Delta t + o(\Delta t)$.
3. Вероятность перехода в состояние, отличное от соседних равна $o(\Delta t)$.
4. Вероятность сохранения прежнего состояния равна $1 - (k^+ + k^-) \Delta t + o(\Delta t)$.
5. Состояние $i = 0$ есть поглощающая граница.

Идея метода стохастизации одношаговых процессов состоит в следующем. На основании схем взаимодействия мы строим основное кинетическое уравнение, раскладываем его в ряд, оставляя только члены до второй производной включительно. Получившееся уравнение будет уравнением Фоккера–Планка. Для получения более привычного вида модели записываем соответствующее ему уравнение Ланжевена. На самом деле, как мы увидим, из схем взаимодействия мы сразу получаем коэффициенты уравнения Фоккера–Планка (и, соответственно, уравнения Ланжевена), поэтому при практическом применении метода строить основное кинетическое уравнение нет необходимости.

А. Схемы взаимодействия

Состояние системы будем описывать вектором состояния $x^i \in \mathbb{R}^n$, где n — размерность системы (под вектором состояния будем понимать множество математических величин, полностью описывающих систему). Оператор $n_j^i \in \mathbb{Z}_{\geq 0}^n \times \mathbb{Z}_{\geq 0}^n$ задаёт состояние системы до взаимодействия, оператор $m_j^i \in \mathbb{Z}_{\geq 0}^n \times \mathbb{Z}_{\geq 0}^n$ — после. В результате взаимодействия происходит переход системы в другое состояние [1, 10].

В системе может происходить s видов различных взаимодействий, где $s \in \mathbb{Z}_+$. Поэтому вместо операторов n_j^i и m_j^i будем рассматривать операторы $n_j^{i\alpha} \in \mathbb{Z}_{\geq 0}^n \times \mathbb{Z}_{\geq 0}^n \times \mathbb{Z}_{\geq 0}^s$ и $m_j^{i\alpha} \in \mathbb{Z}_{\geq 0}^n \times \mathbb{Z}_{\geq 0}^n \times \mathbb{Z}_{\geq 0}^s$.

Взаимодействие элементов системы будем описывать с помощью схем взаимодействия, подобным схемам химической кинетики [11]:

$$n_j^{i\alpha} x^j \xrightleftharpoons[k_{\alpha}^{-}]{k_{\alpha}^{+}} m_j^{i\alpha} x^j, \quad (1)$$

Здесь греческие индексы задают количество взаимодействий, а латинские — размерность системы. Изменение состояния будет задаваться оператором

$$r_j^{i\alpha} = m_j^{i\alpha} - n_j^{i\alpha}. \quad (2)$$

Таким образом, один шаг взаимодействия α в прямом и обратном направлениях можно записать в виде

$$\begin{aligned} x^i &\rightarrow x^i + r_j^{i\alpha} x^j, \\ x^i &\rightarrow x^i - r_j^{i\alpha} x^j. \end{aligned}$$

Мы также можем записывать (1) не в форме векторных уравнений, а в виде более традиционных сумм:

$$n_j^{i\alpha} x^j \delta_i \xrightleftharpoons[k_{\alpha}^{-}]{k_{\alpha}^{+}} m_j^{i\alpha} x^j \delta_i, \quad (3)$$

где $\delta_i = (1, \dots, 1)$.

Также мы будем использовать следующие обозначения:

ния:

$$n^{i\alpha} := n_j^{i\alpha} \delta^j, \quad m^{i\alpha} := m_j^{i\alpha} \delta^j, \quad r^{i\alpha} := r_j^{i\alpha} \delta^j.$$

В. Основное кинетическое уравнение

Вероятности перехода в единицу времени из состояния x^i в состояние $x^i + r_j^{i\alpha} x^j$ (в состояние $x^i - r_j^{i\alpha} x^j$) пропорциональны соответственно числу способов выбора комбинации из x^i по $n^{i\alpha}$ (из x^i по $m^{i\alpha}$) и определяются выражениями:

$$\begin{aligned} s_{\alpha}^{+} &= k_{\alpha}^{+} \prod_{i=1}^n \frac{x^i!}{(x^i - n^{i\alpha})!}, \\ s_{\alpha}^{-} &= k_{\alpha}^{-} \prod_{i=1}^n \frac{x^i!}{(x^i - m^{i\alpha})!}. \end{aligned} \quad (4)$$

Таким образом, общий вид основного кинетического уравнения для вектора состояний x^i , изменяющегося шагами длины $r_j^{i\alpha} x^j$, принимает вид:

$$\begin{aligned} \frac{\partial p(x^i, t)}{\partial t} &= \\ &= \sum_{\alpha=1}^m \left\{ \left[s_{\alpha}^{-}(x^i + r^{i\alpha}, t) p(x^i + r^{i\alpha}, t) - s_{\alpha}^{+}(x^i) p(x^i, t) \right] + \right. \\ &\quad \left. + \left[s_{\alpha}^{+}(x^i - r^{i\alpha}, t) p(x^i - r^{i\alpha}, t) - s_{\alpha}^{-}(x^i) p(x^i, t) \right] \right\}. \end{aligned} \quad (5)$$

С. Уравнение Фоккера–Планка

Далее, используя разложение Крамерса–Мойала, получаем уравнение Фоккера–Планка [11]. Для этого делается несколько предположений:

1. имеют место только малые скачки, т.е. $s_{\alpha}(x^i)$ является функцией, медленно изменяющейся с изменением x^i ;
2. $p(x^i, t)$ также медленно изменяется с изменением x^i .

Тогда можно выполнить сдвиг в (5) из точки $(x^i \pm r_j^{i\alpha} x^j)$ в точку x^i и, разложив правую часть в ряд Тейлора и отбросив члены порядка выше второго, получим уравнение Фоккера–Планка:

$$\frac{\partial p}{\partial t} = -\partial_i [A^i p] + \frac{1}{2} \partial_i \partial_j [B^{ij} p], \quad (6)$$

где

$$\begin{aligned} A^i &:= A^i(x^k, t) = r^{i\alpha} \left[s_{\alpha}^+ - s_{\alpha}^- \right], \\ B^{ij} &:= B^{ij}(x^k, t) = r^{i\alpha} r^{j\alpha} \left[s_{\alpha}^+ - s_{\alpha}^- \right], \quad \alpha = \overline{1, m}. \end{aligned} \quad (7)$$

Как видно из (7), коэффициенты уравнения Фоккера–Планка можно получить сразу из (2) и (4), то есть в практических расчётах записывать основное кинетическое уравнение нет необходимости.

Д. Уравнение Ланжевена

Уравнению Фоккера–Планка соответствует уравнение Ланжевена:

$$dx^i = a^i dt + b_a^i dW^a, \quad (8)$$

где $a^i := a^i(x^k, t)$, $b_a^i := b_a^i(x^k, t)$, $x^i \in \mathbb{R}^n$ — вектор состояния системы, $W^a \in \mathbb{R}^m$ — m -мерный винеровский процесс. Винеровский процесс реализуется как $dW = \varepsilon \sqrt{dt}$, где $\varepsilon \sim N(0, 1)$ — нормальное распределение со средним 0 и дисперсией 1. Латинскими индексами из середины алфавита обозначаются величины, относящиеся к векторам состояний (размерность пространства n), а латинскими индексами из начала алфавита обозначаются величины, относящиеся к вектору винеровского процесса (размерность пространства $m \leq n$).

При этом связь между уравнениями (6) и (8) выражается следующими соотношениями:

$$A^i = a^i, \quad B^{ij} = b_a^i b^{ja}.$$

Мы будем использовать интерпретацию Ито. В рамках интерпретации Ито дифференциал от сложной функции не подчиняется стандартным формулам анализа. Для его вычисления используется *правило* или *лемма Ито*.

Пусть $f := f(x^k, t)$ — функция от случайного процесса $x^k(t)$, $f \in C^2$. Тогда формула дифференциала будет выглядеть следующим образом [12]:

$$df = \left[\partial_t f + a^i f_{,i} + \frac{1}{2} b_a^i b^{ia} f_{,ij} \right] dt + b_a^i f_{,i} dW^a,$$

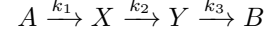
где $f := f(x^k, t)$, $a^i := a^i(x^k, t)$, $b_a^i := b_a^i(x^k, t)$ и $dW^a := dW^a(t)$.

IV. МОДЕЛЬ «ХИЩНИК-ЖЕРТВА»

А. Детерминистическая модель «хищник-жертва»

Системы с взаимодействием двух видов популяций типа «хищник-жертва» широко исследованы и для

таких систем существует большое количество разнообразных моделей. Самой первой моделью «хищник-жертва» принято считать модель, полученную независимо друг от друга А. Лоткой и В. Вольтеррой. Лотка в своей работе описывал некоторую гипотетическую химическую реакцию [13]:



где X, Y — промежуточные вещества, коэффициенты k_1, k_2, k_3 — скорости химических реакций, A — исходный реагент, а B — продукт реакции. В результате была получена система дифференциальных уравнений вида:

$$\begin{cases} \dot{x} = k_1 x - k_2 x y, \\ \dot{y} = k_2 x y - k_3 y. \end{cases}$$

Эта система полностью совпадает с системой дифференциальных уравнений, полученной Вольтеррой, который рассматривал механизм роста численности двух популяций с взаимодействием типа «хищник-жертва». Для получения уравнений Вольтерры делает ряд идеализированных предположений о характере внутривидовых и межвидовых отношений в системе «хищник-жертва» [5].

В. Стохастическая модель «хищник-жертва»

Рассмотрим модель системы «хищник-жертва», состоящую из особей двух видов, причём один из них охотится, второй — обеспечен неисчерпаемыми пищевыми ресурсами. Введя обозначения X — жертва, Y — хищник, можно записать возможные процессы (3) для вектора состояния $x^i = (X, Y)^T$ [14–17]:

$$\begin{aligned} X &\xrightarrow{k_1} 2X, & r^{i1} &= (1, 0)^T \\ X + Y &\xrightarrow{k_2} 2Y, & r^{i2} &= (-1, 1)^T \\ Y &\xrightarrow{k_3} 0, & r^{i3} &= (0, -1)^T, \end{aligned}$$

которые имеют следующую интерпретацию. Первое соотношение означает, что жертва, которая съедает единицу пищи, немедленно репродуцируется. Второе соотношение описывает поглощение жертвы хищником и мгновенное репродуцирование хищника. Это единственная рассматриваемая возможность гибели жертвы. Последнее соотношение — это естественная смерть хищника.

Все процессы необратимы, поэтому $s_{\alpha}^- = 0$, а

$$\begin{aligned} s_1^+(x, y) &= k_1 \frac{x!}{(x-1)!} \frac{y!}{y!} = k_1 x, \\ s_2^+(x, y) &= k_2 \frac{x!}{(x-1)!} \frac{y!}{(y-1)!} = k_2 x y, \\ s_3^+(x, y) &= k_3 \frac{x!}{x!} \frac{y!}{(y-1)!} = k_3 y. \end{aligned}$$

Воспользовавшись формулой (6) имеем уравнение Фоккера–Планка:

$$\frac{\partial p(x, y)}{\partial t} = -\partial_i (A^i(x, y)p(x, y)) + \frac{1}{2}\partial_i\partial_j (B^{ij}(x, y)p(x, y)),$$

где

$$A^i(x, y) = s_\alpha^+(x, y) r^{i\alpha}, \\ B^{ij}(x, y) = s_\alpha^+(x, y) r^{i\alpha} r^{j\alpha}.$$

Таким образом мы получили:

$$A^i(x, y) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} k_1 x + \begin{pmatrix} -1 \\ 1 \end{pmatrix} k_2 xy + \\ + \begin{pmatrix} 0 \\ -1 \end{pmatrix} k_3 y = \begin{pmatrix} k_1 x - k_2 xy \\ k_2 xy - k_3 y \end{pmatrix}, \\ B^{ij}(x, y) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} (1, 0) k_1 x + \begin{pmatrix} -1 \\ 1 \end{pmatrix} (-1, 1) k_2 xy + \\ + \begin{pmatrix} 0 \\ -1 \end{pmatrix} (0, -1) k_3 y = \begin{pmatrix} k_1 x + k_2 xy & -k_2 xy \\ -k_2 xy & k_2 xy + k_3 y \end{pmatrix}. \quad (9)$$

Для того чтобы записать стохастическое дифференциальное уравнение в форме Ланжевена (8) для модели «хищник–жертва», достаточно извлечь квадратный корень из полученной матрицы B^{ij} в уравнении Фоккера–Планка.

$$d \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} k_1 x - k_2 xy \\ k_2 xy - k_3 y \end{pmatrix} dt + b_a^i \begin{pmatrix} dW^1 \\ dW^2 \end{pmatrix}, \\ b_a^i b_a^{ja} = B^{ij} = \begin{pmatrix} k_1 x + k_2 xy & -k_2 xy \\ -k_2 xy & k_2 xy + k_3 y \end{pmatrix}.$$

Следует заметить, что конкретный вид матрицы b_a^i не выписан из-за крайней громоздкости выражения. Впрочем, при дальнейших исследованиях нам понадобится не собственно матрица b_a^i , а её квадрат, то есть матрица B^{ij} .

V. РЕАЛИЗАЦИЯ МОДЕЛИ ОДНОШАГОВЫХ СТОХАСТИЧЕСКИХ ПРОЦЕССОВ В СИСТЕМЕ КОМПЬЮТЕРНОЙ АЛГЕБРЫ

A. Обоснование выбора системы компьютерной алгебры

Рассмотрим системы аналитических вычислений *Maxima* и *Axiom*. *Maxima* является самой первой системой аналитических вычислений, написана она на языке *Lisp*. *Maxima* успешно работает на всех современных операционных системах: *Windows*, *Linux* и *UNIX*, *Mac OS* и даже на КПК под управлением *Windows CE/Mobile*. Документация интегрирована в программу в виде справочника, оснащенного средствами поиска. В *Maxima* нет разделения на объекты и данные, нет четкого разграничения между оператором

и функцией. В системе нет встроенной графической отрисовки.

В отличие от *Maxima*, язык *Axiom* строго типизирован для лучшего отображения математических объектов и взаимосвязей. Вся математическая база написана на языке *Spad*. Переносимость *Axiom* чуть хуже: система работает под *Linux*, *UNIX*, а под *Windows* не работает построение графиков. *Axiom* обладает собственной графической подсистемой.

В 2007 году у *Axiom* появилось два форка с открытым исходным кодом: *OpenAxiom* и *FriCAS*. *OpenAxiom* разрабатывается, придерживаясь идеологии *Axiom*, устраняются проблемы, которые встречались в *Axiom*. Разработчики проекта *FriCAS* реорганизовали процесс сборки, расширили функционал. Кроме того, *FriCAS* поддерживает не только *GCL*, который работает на ограниченном количестве платформ, но и *ECL*, *Clisp*, *sbc1* или *openmcl*, что позволяет запускать *FriCAS* на более широком множестве платформ.

B. Описание реализации в системе компьютерной алгебры *Axiom*

Метод стохастизации одношаговых процессов организован в виде модуля для системы компьютерной алгебры *FriCAS*. Для вывода всех вычислений на экран использована переменная `SHOWCALC := true`. Для вызова метода нужно воспользоваться основной функцией, которая имеет следующий вид:

```
osp(Matrix(Integer), Matrix(Integer),
     Vector, Vector, Vector)
```

где первый аргумент — матрица n_j^i состояний до взаимодействия, второй аргумент — матрица m_j^i после взаимодействия, третий аргумент — вектор k_α^+ , четвертый аргумент — вектор k_α^- , пятый аргумент — вектор состояний x^i . Рассмотрим особенности языка *FriCAS* на вспомогательных функциях. Например, функция `calcProd` используется для упрощения вычислений s_α^+ и s_α^- . В реализации функции используется оператор условия и встроенная функция `reduce`:

```
calcProd : (Matrix(Integer), Vector, Integer,
            Integer) -> Void
calcProd (n, x, a, i) ==
  nai:Integer := n(a,i)
  if nai = 0 then 1 else reduce(*,
                                [x(i) - j for j in 0..(nai-1)])
```

В функции `Bi` производятся промежуточные вычисления для элементов матрицы B^{ij} :

```
Bi (rv, sp, sm, i) == rv(i) *
  (transpose rv(i)) * (sp(i) + sm(i))
```

Чтобы воспользоваться модулем для расчёта модели системы «хищник–жертва», вызываем функцию со следующими аргументами:

```

Buffer File Edit Insert Text Paragraph Document Project Options Help
[Icons]
osp ([[1,0],[1,1],[0,1]], [[2,0],[0,2],[0,0]], vector([k1,k2,k3]), vector([0,0,0]), vector([x,y]))

FriCAS will attempt to step through and interpret the code.
Cannot compile map: calcProd
We will attempt to interpret the code.
Compiling function Ai with type (List(Matrix(Integer)), List(Polynomial(Integer)), List(Polynomial(Integer)), NonNegativeInteger) -> Matrix(Polynomial(Integer))
Compiling function A with type (List(Matrix(Integer)), List(Polynomial(Integer)), List(Polynomial(Integer)), NonNegativeInteger) -> Matrix(Polynomial(Float))
Compiling function Bi with type (List(Matrix(Integer)), List(Polynomial(Integer)), List(Polynomial(Integer)), NonNegativeInteger) -> Matrix(Polynomial(Integer))
Compiling function B with type (List(Matrix(Integer)), List(Polynomial(Integer)), List(Polynomial(Integer)), NonNegativeInteger) -> Matrix(Polynomial(Float))
(-k2xy + k1x -k3 + k2xy) * (k2xy + k1x -k2xy -k2xy k3y + k2xy)

```

$$\begin{vmatrix} R_{ij} & \hat{R}_{ij} \\ r_j & \hat{r}_j \end{vmatrix}.$$

Для вычислений нами был использован метод с таблицей

0	0	0	0	0	0	0
2/3	0	0	2/3	0	0	
-1	1	0	-1	1	0	
0	3/4	1/4	0	3/4	1/4	

Рис. 1. Результат работы модуля для модели «хищник-жертва» в графической оболочке T_EXmacs

```

osp ([[1,0],[1,1],[0,1]], [[2,0],[0,2],[0,0]],
      vector([k1,k2,k3]), vector([0,0,0]),
      vector([x,y]))

```

На рис. 1 представлен результат, получающийся в оболочке T_EXmacs. Фактически, мы повторили результаты, полученные в (9).

VI. ЧИСЛЕННЫЙ ЭКСПЕРИМЕНТ В РАМКАХ ПРОГРАММНОГО КОМПЛЕКСА

A. Стохастические методы Рунге–Кутты

Одним из хорошо известных численных методов решения СДУ является метод Эйлера–Маруямы, который, является частным случаем более общих стохастических методов Рунге–Кутты. Классический метод Рунге–Кутты может быть обобщен на случай системы СДУ (8) следующим образом [3, 4]:

$$\begin{cases} X_k^i = x_0^i + h R_k^l a^i(X_{\underline{l}}^1, \dots, X_{\underline{l}}^n) + \hat{R}_k^l J^\alpha b_\alpha^i(X_{\underline{l}}^1, \dots, X_{\underline{l}}^n), \\ x_1^i = x_0^i + h r^l a^i(X_{\underline{l}}^1, \dots, X_{\underline{l}}^n) + \hat{r}^l J^\alpha b_\alpha^i(X_{\underline{l}}^1, \dots, X_{\underline{l}}^n) \end{cases}$$

Индексы $k = 1, \dots, s$ и $l = 1, \dots, n$ относятся к стохастическому методу Рунге–Кутты. $J \sim N(0, h)$ или $J \sim \sqrt{h}\varepsilon$, $\varepsilon \sim N(0, 1)$ — нормально распределенные случайные величины. Выбор таких величин для численной аппроксимации объясняется тем, что винеровские процесс реализуется как $dW = \varepsilon \sqrt{dt}$. Также следует обратить внимание на двойное суммирование в третьем слагаемом обеих формул численной схемы, а также на то, что каждое число J^1, \dots, J^n должно генерироваться отдельно.

Коэффициенты метода, также как и для классического аналога, можно сгруппировать в таблицу, называемую *таблицей Батчера*:

B. Описание программной реализации

При написании комплекса программ ставилась задача как автоматизации вычисления коэффициентов A^i и B^{ij} СДУ с помощью описываемых выше общих принципов, так и численное решение полученного уравнения с помощью стохастических методов Рунге–Кутты. С точки зрения программирования задача распалась на три подзадачи:

1. генерация коэффициентов A^i и B^{ij} при помощи системы компьютерной алгебры;
2. генерация исходных текстов программ на языках Fortran и Julia, реализующих СДУ на основе полученных коэффициентов, сохраненных в виде текстового файла;
3. написание подпрограмм/функций, реализующих стохастические методы Рунге–Кутты на языках Fortran и Julia, и дальнейшая компиляция их вместе с автоматически сгенерированными исходными текстами.

Модуль Axiom в результате своей работы создает текстовый файл, в котором содержатся коэффициенты A^i и B^{ij} в следующем виде:

```

# A
A[1]
...
A[N]
# B
B[1,1] B[1,2] .. B[1,N]
...
B[N,1] B[N,2] .. B[N,N]

```

Матрица $b_\alpha^i = \sqrt{b_\alpha^i b^{j\alpha}} = \sqrt{B^{ij}}$ вычисляется численно при помощи сингулярного разложения матрицы (использована подпрограмма DGESVD библиотеки LAPACK).

Для решения второй подзадачи был выбран интерпретируемый язык Python (версии 3). Данный язык

обладает широким набором средств для работы со строками и текстовыми файлами. Кроме матриц A^i и B^{ij} была задана дополнительная информация о математической модели в виде словаря (стандартный тип данных в Python), где были указаны: название модели, список переменных, список параметров, начальные значения переменных, значения параметров и параметры численного метода (отрезок интегрирования и величина шага).

На основе этих данных сценарий автоматически генерирует два файла `functions.f90` и `main.f90`, где первый — это модуль, содержащий функции, задающие СДУ, а второй — основной файл программы. При компиляции к этим файлам добавляются еще три модуля со вспомогательными процедурами, в числе которых содержится стохастический метод Рунге-Кутты.

С. Описание численного эксперимента

В качестве модели для верификации работы комплекса программ была выбрана хорошо известная модель «хищник-жертва», задаваемая вектором a^i с компонентами

$$a^1 = \alpha x - \beta xy, \quad a^2 = -\gamma y + \delta xy$$

и матрицей B^{ij} :

$$\begin{bmatrix} \alpha x + \beta xy & -\beta xy \\ -\beta xy & \beta xy + \gamma x \end{bmatrix},$$

x — число жертв, y — число хищников. Коэффициенты же имеют следующий физический (биологический) смысл: α — интенсивность роста популяции жертв, β — частота встречи хищников и жертв, γ — интенсивность смерти или миграции хищников в условиях недостатка жертв, δ — интенсивность роста популяции хищников при условии избытка жертв.

При численном моделировании учитывалось, что значение переменных x, y не может быть меньше нуля, и при обращении в ноль одной из переменных работа программы прекращалась.

Численное моделирование показало, что добавление стохастики в классическую модель «хищник-жертва» приводит к тому, что по прошествии определенного времени наступает смерть одного из конкурирующих видов. Так, при следующих значениях параметров: $\alpha = 10$, $\beta = 1,5$, $\gamma = 8,5$, $\delta = 1,8$ и следующих начальных значениях: $x = 9,7$, $x = 6,77$ первыми погибают жертвы, а за ними и хищники ввиду недостатка пищи. Данный случай проиллюстрирован на рис. 2. Для сравнения на рис. 3 представлен график для детерминированного случая.

При других условиях ($\alpha = 10$, $\beta = 1,5$, $\gamma = 8,5$, $\delta = 0,5$, $x = 22$, $y = 6,76$) погибают хищники, а численность жертв стремительно возрастает, так как для них модель предполагает бесконечный источник

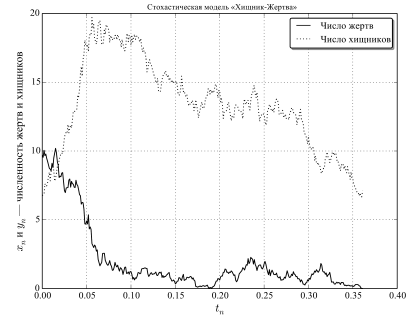


Рис. 2. Стохастическая модель «хищник-жертва» — смерть жертв

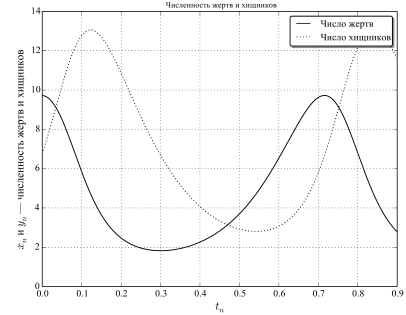


Рис. 3. Детерминированная модель «хищник-жертва»

пищи. Графики для данного случая изображены на рис. 4, а на рис. 5 для сравнения показан детерминированный случай.

ВІІ. ЗАКЛЮЧЕНИЕ

В работе продемонстрировано применение разработанного метода формализации исходной физической системы, представимой в виде одного или нескольких одношаговых процессов. Формализация системы про-

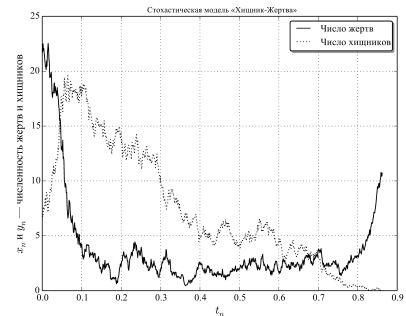


Рис. 4. Стохастическая модель «хищник-жертва» — смерть хищников

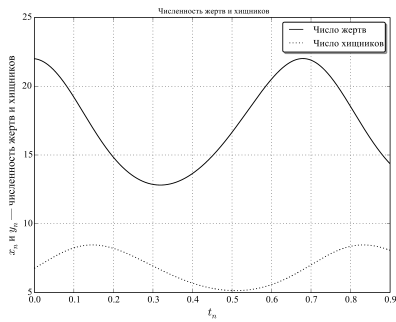


Рис. 5. Детерминированная модель «хищник-жертва»

изводится путем введения оператора эволюции. При этом для получения аналитического описания модели требуется применение большого числа рутинных операций. Для упрощения работы предложено использовать средства компьютерной алгебры — клон Axiom FriCAS.

Нами был разработан аналитический программный блок пакета, получающий на входе оператор эволюции, а на выходе выдающий систему СДУ, описывающую исходную модель. Для численного исследования полученной таким образом системы СДУ разработан второй программный блок, который преобразует полученную в FriCAS систему уравнений в программный код на языке fortran и на выходе дает ее численное решение. Таким образом, разработанный программный комплекс позволяет провести одновременно аналитическое и численное исследование исходной модели.

На данный момент разработанный программный комплекс охватывает не все возможности, заложенные в предложенном методе формализации исходной физической системы. Так в случае описания исходной системы с помощью ОДУ, граничные условия приходится вводить с помощью связей или индикаторных функций. В тоже время применение дифференциальных уравнений в частных производных может позволить решить эту проблему. Дальнейшие задачи — разработка полного программного комплекса применения методики построения одношаговых моделей исходной физической системы.

-
- [1] Кулябов Д. С., Демидова А. В. Введение согласованного стохастического члена в уравнение модели роста популяций // Вестник РУДН. Серия «Математика. Информатика. Физика». — 2012. — № 3. — С. 69–78.
 - [2] Demidova A. V., Sevastianov L. A., Kulyabov D. S. Application of stochastic differential equations to model population systems // Труды Третьей Международной конференции «Математическое моделирование социальной и экономической динамики (MMSD-2010)» / Российский государственный социальный университет. — 2010. — С. 92–94.
 - [3] Debrabant K., Rößler A. Classification of Stochastic Runge–Kutta Methods for the Weak Approximation of Stochastic Differential Equations // Mathematics and Computers in Simulation. — 2008. — Vol. 77, no. 4. — P. 408–420.
 - [4] Tocino A., Ardanuy R. Runge–Kutta Methods for Numerical Solution of Stochastic Differential Equations // Journal of Computational and Applied Mathematics. — 2002. — Vol. 138, no. 2. — P. 219–241. — URL: <http://www.sciencedirect.com/science/article/pii/S0377042701003806>.
 - [5] Volterra V. Mathematical Theory of the Struggle for Existence. — Nauka, 1976.
 - [6] Базыкин А. Д. Нелинейная динамика взаимодействующих популяций. — Москва-Ижевск : Институт компьютерных исследований, 2003.
 - [7] Братусь А. С., Новожилов А. С., Платонов А. П. Динамические системы и модели биологии. — М. : Физматлит, 2010.
 - [8] Пенроуз Р., Риндлер В. Спиноры и пространство-время. Два-спинорное исчисление и релятивистские поля. — М. : Мир, 1987. — Т. 1. — 528 с.
 - [9] Влияние стохастизации на одношаговые модели / Анастасия Вячеславовна Демидова, Мигран Нельсонович Геворкян, Александр Дмитриевич Егоров и др. // Вестник РУДН. Серия «Математика. Информатика. Физика». — 2014. — № 1. — С. 71–85.
 - [10] The Method of Stochastization of One-Step Processes / Anastsasiya Vyacheslavovna Demidova, Anna Vladislavovna Korolkova, Dmitry Sergeevich Kulyabov, Leonid Antonovich Sevastianov // Mathematical Modeling and Computational Physics. — JINR, 2013. — P. 67.
 - [11] Гардинер К. В. Стохастические методы в естественных науках. — Мир, 1986.
 - [12] Øksendal B. K. Stochastic Differential Equations: An Introduction with Applications. — Berlin : Springer, 2003.
 - [13] Lotka A. J. Elements of Physical Biology. — BiblioBazaar, 2011. — 492 p. — ISBN: 9781178508116. — URL: <http://books.google.ru/books?id=tFN9pwAACAAJ>.
 - [14] Демидова А. В. Уравнения динамики популяций в форме стохастических дифференциальных уравнений // Вестник РУДН. Серия «Математика. Информатика. Физика». — 2013. — № 1. — С. 67–76.
 - [15] Демидова А. В. Метод стохастизации математических моделей на примере системы «хищник–жертва» // Научная сессия НИЯУ МИФИ-2013. — 2013. — С. 127.
 - [16] Королькова А. В., Кулябов Д. С. Методы стохастизации математических моделей на примере пиринговых сетей // Научная сессия НИЯУ МИФИ-2013. Аннотации докладов. В 3 томах. — Москва : МИФИ, 2013. — С. 131.
 - [17] Демидова А. В., Кулябов Д. С., Севастьянов Л. А. Согласованный стохастический член в популяционных моделях // XI Белорусская математическая конференция. — Минск : Институт математики НАН Беларуси, 2012. — С. 39.