

**Кулябов Д.С.<sup>1,2</sup>, Геворкян М.Н.<sup>1</sup>, Демидова А.В.<sup>1</sup>, Королькова А.В.<sup>1</sup>, Севастьянов Л.А.<sup>1</sup>,  
Котюков М.М.<sup>1</sup>**

<sup>1</sup> Российский университет дружбы народов, г. Москва, Россия

<sup>2</sup> Объединенный институт ядерных исследований, г. Дубна, Россия

## **АНАЛИЗ ТРУДНОСТЕЙ РЕАЛИЗАЦИИ СТОХАСТИЧЕСКИХ ЧИСЛЕННЫХ МЕТОДОВ РУНГЕ—КУТТЫ\***

### **Аннотация**

*В данной статье рассматриваются стохастические численные методы типа Рунге-Кутты с слабой и сильной сходимостями для систем стохастических дифференциальных уравнений в форме Ито. Дается краткий обзор основных работ по данной теме. Также кратко излагается теоретический материал по стохастическим численным методам и сведения из теории стохастических дифференциальных уравнений. В связи с тем, что аналитическому решению поддается лишь малая доля стохастических дифференциальных уравнений, стохастические численные методы являются единственным способом получения решения. Стохастические методы Рунге-Кутта основываются на тех же идеях, что и классические методы, однако их схемы существенно сложнее своих детерминированных аналогов. Это обуславливает трудности при попытке программной реализации. Авторами мотивируется подход к реализации данных методов с помощью генерации исходного кода так как это позволяет добиться универсальности при сохранении простоты кода. Обсуждаются детали реализации и используемые языки программирования и библиотеки. Приводится ряд примеров стохастических дифференциальных уравнений, используемых для тестирования программы и оценки погрешности вычислений. Дается ссылка на репозиторий с исходным кодом программы на языке Python.*

### **Ключевые слова**

*Стохастические дифференциальные уравнения; стохастические численные методы; автоматическая генерация кода; Python; Julia; шаблонизатор.*

**Kulyabov D.S.<sup>1,2</sup>, Gevorgyan M.N.<sup>1</sup>, Demidova A.V.<sup>1</sup>, Korolkova A.V.<sup>1</sup>, Sevastianov L.A.<sup>1</sup>,  
Kotukov M.M.<sup>1</sup>**

<sup>1</sup> Peoples' Friendship University of Russia, Moscow, Russia

<sup>2</sup> Joint Institute for Nuclear Research, Dubna, Moscow region, Russia

## **IMPLEMENTATION DIFFICULTIES ANALYSIS OF STOCHASTIC NUMERICAL RUNGE-KUTTA METHODS**

### **Abstract**

*This article discusses stochastic numerical methods of Runge-Kutta methods with weak and strong convergences for systems of stochastic differential equations in the Ito form. At first section we give a brief review of main works on the topic. In next section we make short introduction into theory of stochastic numerical methods and facts from the theory of stochastic differential equations. Due to the fact that the analytical solution exists only for a small subset of stochastic differential equations, stochastic numerical methods are the only way to obtain the solution. Stochastic Runge-Kutta methods are based on the same ideas as the classic methods, but their scheme is essentially more complicated than their deterministic counterparts. This leads to difficulties when one attempts to implement them. In this paper we motivate the approach to the implementation of these methods using source code generation as it allows achieving universality while maintaining a simplicity of code. We discuss some of the implementation details and the used programming languages and libraries.*

\* Труды II Международной научной конференции «Конвергентные когнитивно-информационные технологии» (Convergent'2017), Москва, 24-26 ноября, 2017

Proceedings of the II International scientific conference "Convergent cognitive information technologies" (Convergent'2017), Moscow, Russia, November 24-26, 2017

*We give several examples of stochastic differential equations, used to test the program and evaluate the error of the calculations. We provide the link to the repository with the source code of discussed programs.*

## **Keywords**

*Stochastic differential equations; stochastic numerical methods; automatic code generation; Python; Julia; the template engine.*

## **Введение**

В статье авторов [1] описывалась реализация стохастических численных методов типа Рунге-Кутты на языке Python [2] с использованием библиотек NumPy и SciPy [3]. Выбор языка был продиктован удобством программирования и возможностью работать с многомерными массивами как с тензорами (функции `tensor_dot` и `einsum` из библиотеки NumPy). Однако производительность созданных функций была на низком уровне, и не столько из-за низкого быстродействия языка Python, сколько из-за большого числа вложенных циклов (до семи циклов). В данной статье мы рассматриваем альтернативный подход к реализации стохастических численных методов, основанный на генерации кода функций.

Данная статья состоит из трех разделов. В первом разделе дается обзор основных источников и излагаются сведения из теории стохастических дифференциальных уравнений (СДУ) и численных методов для их решения. Во втором разделе приведены стохастические численные схемы для скалярных СДУ с сильной сходимостью и для систем СДУ с сильной и слабой сходимостью. Наряду с общими схемами приводятся несколько таблиц коэффициентов, которые позволяют реализовать уже конкретный численный метод. Наконец в третьем разделе мотивируется применение генерации кода для стохастических численных методов и описываются некоторые детали реализованного нами генератора (на языке Python с использованием шаблонизатора Jinja2).

## **Обзор основных источников**

Кратко охарактеризуем основные доступные статьи, посвященные стохастическим методам. Особое внимание уделим многостадийным методам. Первым стохастические броуновские процессы для математического моделирования использовал французский математик Л. Ж.-Б. А. Башелье (1870-1946) в 1900 году в работе [4].

В качестве краткого введения в область стохастических численных методов можно использовать обзорную статью А. В. Лукшина и С. Н. Смирнова [5], в которой тезисно изложены подходы к численному решению стохастических уравнений по состоянию на 1990 год. В более новой статье [6] А. Ф. Ерешко и Д. В. Филатовой также приведены некоторые численные методы (в том числе и один метод Рунге-Кутты и методы Г. Н. Милштейна [7-9]), а также даны результаты численного эксперимента.

Используемые в данной статье численные схемы в основном были получены А. Росслером, который в работе [10] последовательно изложил теорию стохастических методов Рунге-Кутты. Автор рассматривает аппроксимацию систем СДУ Ито и Стратоновича в слабом смысле для скалярного и многомерного процесса Винера. После краткого обзора литературы автор переходит к изложению теории помеченных деревьев, которая используется для вывода условий порядка в случае детерминированных методов Рунге-Кутты [11] и которая была распространена на случай СДУ. Дальнейшие результаты Росслера изложены в статьях [12,13]. В препринте [13] продолжена классификация стохастических методов Рунге-Кутты со слабой сходимостью на основе условий порядка. Приведено несколько конкретных реализаций и даны результаты численных экспериментов. В другом препринте [14] приведены таблицы методов четвертой стадийности и сильного порядка сходимости  $p = 2.0$ .

П. М. Бурраж и К. Бурраж в серии статей [15-19] изучили методы сильного порядка  $p = 1.5$ , а также распространили теорию помеченных деревьев на стохастический случай. В статье R. Soheili и M. Namjoo [20] получены три метода с сильной сходимостью  $p = 1.0$  и проведено численное сравнение с методом из книги [21]. Дальнейшее развитие методы со слабой сходимостью получили в статьях [22] и [23]. В статье [24] получены два трехстадийных метода слабой сходимости  $p = 2.0$ , а также проведены численные эксперименты.

В виду крайней сложности дальнейшего повышения порядка точности стохастических численных схем, современные работы в основном концентрируются на получении численных схем для частных случаев СДУ. Можно выделить работы, посвященные симплектическим стохастическим численным методам Рунге-Кутта [27-29] и неявным схемам, в частности стохастическим аналогам метода Розенброка [30].

В качестве монографии, посвященной целиком стохастическим численным методам, следует упомянуть книгу [21], которая содержит подробную сводку разнообразных численных схем, а также список СДУ, которые имеют аналитические решения.

## Случайный процесс Винера и программное генерирование его траекторий

Случайный процесс  $W(t), t \geq 0$  называется скалярным процессом Винера, если выполняются следующие условия [25, 32, 33]:

- $P\{W(0) = 0\} = 1$  иначе говоря,  $W(0) = 0$  почти наверное;
- $W(t)$  — процесс с независимыми приращениями, то есть приращения  $\Delta W_i$  являются независимыми случайными величинами

- $\Delta W_i = W(t_{i+1}) - W(t_i) \sim \mathcal{N}(0, t_{i+1} - t_i)$ , где  $0 \leq t_{i+1} < t_i < T, i = 0, 1, \dots, N - 1$ .

Обозначение  $\Delta W_i \sim \mathcal{N}(0, \Delta t_i)$  говорит о том, что  $\Delta W_i$  — нормально распределенная случайная величина с математическим ожиданием  $\mathbb{E}[\Delta W_i] = \mu = 0$  и дисперсией  $\mathbb{D}[\Delta W_i] = \sigma^2 = \Delta t_i$

Винеровский процесс является моделью броуновского движения (хаотического блуждания). Если рассмотреть процесс  $W(t)$  в те моменты времени  $0 = t_0 \leq t_1 < t_2 < \dots < t_{N-1} < t_N$ , когда он испытывает случайные аддитивные изменения, то непосредственно из определения следует:

$$W(t_1) = W(t_0) + \Delta W_0, W(t_2) = W(t_1) + \Delta W_1, \dots, W(t_N) = W(t_{N-1}) + \Delta W_{N-1},$$

где по  $\Delta W_i \sim \mathcal{N}(0, \Delta t_i), \forall i = 0, \dots, N - 1$ .

Запишем  $W(t_N)$  в виде накопленной суммы:

$$W(t_N) = W(t_0) + \sum_{i=0}^{N-1} \Delta W_i$$

и учтем, что  $\mathbb{E}[\Delta W_i] = 0$  и  $\mathbb{D}[\Delta W_i] = \Delta t_i$  можно показать, что сумма нормально распределенных случайных чисел  $\Delta W_i$  также является нормально распределенным случайным числом:

$$\mathbb{E} \sum_{i=0}^{N-1} \Delta W_i = 0, \quad \mathbb{D} \sum_{i=0}^{N-1} \Delta W_i = \sum_{i=0}^{N-1} \Delta t_i = T, \quad \sum_{i=0}^{N-1} \Delta W_i \sim \mathcal{N}(0, T).$$

Многомерный винеровский процесс  $W(t): \Omega \times [t_0, T] \rightarrow \mathbb{R}^m$  определяют как случайный процесс составленный из совместно независимых одномерных винеровских процессов  $W_1(t), \dots, W_m(t)$ . Приращения  $\Delta W_i^\alpha, \forall \alpha = 1, \dots, m$  являются совместно независимыми нормально распределенными случайными величинами. С другой стороны? вектор  $\Delta W_i^\alpha$  можно представить как многомерную нормально распределенную случайную величину с вектором математических ожиданий  $\mu = 1$  и диагональной матрицей ковариаций. Рассмотрим теперь вопрос генерации траектории процесса Винера на компьютере. Для симулирования одномерного винеровского процесса необходимо сгенерировать  $N$  нормально распределенных случайных чисел  $\mu = \varepsilon_1, \dots, \varepsilon_N$  и построить их накопленные суммы  $\varepsilon_1, \varepsilon_1 + \varepsilon_2, \varepsilon_1 + \varepsilon_2 + \varepsilon_3$  и так далее, то мы получим выборочную траекторию винеровского процесса  $W(t)$  (независимого случайного блуждания).

В случае многомерного случайного процесса следует сгенерировать уже  $m$  последовательностей из  $N$  нормально распределенных случайных величин.

## Стохастические интегралы и СДУ для скалярного процесса Винера

Формулировка стохастического дифференциального уравнения начинается с введения операции стохастического интегрирования.

**Определение.** Пусть  $g(t, x(t))$  непрерывная скалярная функция, зависящая от случайного процесса  $x(t)$ ,  $W(t)$  — процесс Винера,  $t \in [t_0, T]$ ,  $0 \leq t_0 < t_1 < t_2 < \dots < t_{n-1} < t_n \leq T < \infty$  — разбиение отрезка  $[t_0, T]$ , тогда

$$I(\theta) =: \int_{t_0}^T g(t, x(t)) dW(t) \stackrel{\text{def}}{=} \lim_{n \rightarrow \infty} \sum_{i=0}^{n-1} g(\theta t_{i+1} + (1 - \theta)t_i, \theta x(t_{i+1}) + (1 - \theta)x(t_i))(W(t_{i+1}) - W(t_i)),$$

где  $\theta \in [0, 1]$ .  $I(\theta)$  называется *стохастическим интегралом*. Строгое обоснование существования данного интеграла для широкого класса функций см. [25, глава 3].

В физике и прикладной математике нашли применение два частных случая стохастического интеграла:

- *интеграл Ито* (назван в честь японского математика К. Ито)

$$I = I(0) = \lim_{n \rightarrow \infty} \sum_{i=0}^{n-1} g(t_i, x(t_i))(W(t_{i+1}) - W(t_i)),$$

- *интеграл Стратоновича* (назван в честь советского физика Р. Л. Стратоновича)

$$I_{0.5} = I(0.5) = \lim_{n \rightarrow \infty} \sum_{i=0}^{n-1} g\left(\frac{t_{i+1} + t_i}{2}, \frac{x(t_{i+1}) + x(t_i)}{2}\right)(W(t_{i+1}) - W(t_i)).$$

После введения операции стохастического интегрирования, можно записать интегрального уравнения для случайного процесса  $x(t)$  [25].

$$x(t_k) = x(t_0) + \int_0^t f(\tau, x(\tau))d\tau + \underbrace{\int_0^t g(\tau, x(\tau))dW}_{\text{стохастический интеграл}}$$

Вышеприведенному интегральному уравнению соответствует стохастическое дифференциальное уравнение:

$$dx(t) = f(t, x(t))dt + g(t, x(t))dW.$$

При этом необходимо помнить, что дифференциал типа  $dx$  не является обычным «малым» приращением функции  $x(t)$  это случайная величина. Фактически СДУ является символической записью предела итерационной схемы.

Для численных схем необходимо уметь вычислять стохастические интегралы Ито следующего вида:

$$\int_{t_0}^t W(\tau)dW(\tau) = \frac{W^2(t)}{2} - \frac{t}{2}.$$

Мы сразу привели конечный результат так как полная процедура вычисления данного интеграла громоздка.

### СДУ в форме Ито для многомерного процесса Винера

Введем вероятностное пространство  $(\Omega, \mathcal{A}, \mathbb{P})$ , где  $\Omega$  — пространство элементарных событий,  $\mathcal{A}$  — сигма-алгебра подмножеств пространства  $\Omega$ ,  $\mathbb{P}$  — вероятностная мера. На отрезке  $[t_0, T] \in \mathbb{R}^1$  определена переменная  $t$  имеющая физический смысл времени.

Рассмотрим случайный процесс  $\mathbf{x}(t) = (x^1(t), \dots, x^d(t))^T$ , где  $\mathbf{x}(t)$  принадлежит функциональному пространству  $L^2(\Omega)$  с нормой  $\|\cdot\|$ . Будем считать, что случайный процесс  $\mathbf{x}(t)$  является решением для стохастического дифференциального уравнения (СДУ) в форме Ито [1,29,30]:

$$d\mathbf{x}(t) = \mathbf{f}(t, \mathbf{x}(t))dt + \mathbf{G}(t, \mathbf{x}(t))d\mathbf{W},$$

где  $\mathbf{W} = (W^1, \dots, W^m)^T$  — многомерный винеровский процесс, называемый *ведущим (driver) процессом* СДУ. Функция  $\mathbf{f}: [t_0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$  называется *вектором сноса*, а матричнозначная функция  $\mathbf{G}: [t_0, T] \times \mathbb{R}^d \times \mathbb{R}^m \rightarrow \mathbb{R}^d \times \mathbb{R}^m$  — *матрицей диффузии*. Кроме того,  $\mathbf{f}(t, \mathbf{x}(t)) = (f^1(t, \mathbf{x}(t)), \dots, f^d(t, \mathbf{x}(t)))^T$ , а матрица  $\mathbf{G}$  имеет вид:

$$\mathbf{G}(t, \mathbf{x}(t)) = \begin{bmatrix} g_1^1(t, \mathbf{x}) & \dots & g_m^1(t, \mathbf{x}) \\ \vdots & \ddots & \vdots \\ g_1^d(t, \mathbf{x}) & \dots & g_m^d(t, \mathbf{x}) \end{bmatrix}$$

То же уравнение можно переписать в индексном виде:

$$x^\alpha(t) = f^\alpha(t, x^\gamma(t))dt + \sum_{\beta=1}^m g_\beta^\alpha(t, x^\gamma(t))dW^\beta,$$

где  $\alpha, \gamma = 1, \dots, d$ ,  $\beta = 1, \dots, m$ , а  $f^\alpha(t, x^\gamma(t)) = f^\alpha(t, x^1(t), \dots, x^d(t))$ .

На отрезке  $[t_0, T]$  введем сетку  $t_0 < t_1 < \dots < t_N = T$  с шагом  $h_n = t_{n+1} - t_n$ , где  $n = 0, \dots, N-1$  и максимальный шаг сетки  $h = \max\{h_{n-1}\}_1^N$ . Далее будем полагать, что сетка равномерная, то  $h_n = h = \text{const}$ .  $\mathbf{x}_n$  — сеточная функция, аппроксимирующая случайный процесс  $\mathbf{x}(t)$  так что  $\mathbf{x}_0 = \mathbf{x}(t_0)$ ,  $\mathbf{x}_n \approx \mathbf{x}(t_n)$ ,  $\forall n = 1, \dots, N$ .

### Вычисление и аппроксимация кратных интегралов Ито

В общем случае для конструирования численных схем, порядок сильной сходимости которых был бы больше чем 0,5, необходимо включение в формулы этих схем однократных и двукратных интегралов Ито:

$$I^i(t_n, t_{n+1}) = I^i(h_n) = \int_{t_n}^{t_{n+1}} dW^i, \quad I^{ij}(t_n, t_{n+1}) = I^{ij}(h_n) = \int_{t_n}^{t_{n+1}} \int_{t_n}^{\tau_1} dW^i(\tau_2)dW^j(\tau_1),$$

где  $i, j = 1, \dots, m$  и  $W_i$  — компоненты многомерного винеровского процесса.

Возникает задача выражения этих интегралов через приращения  $\Delta W_n^i = W^i(t_{n+1}) - W^i(t_n)$ . В случае однократного интеграла это можно сделать для любого индекса  $i$ :  $I^i(h_n) = W_n^i$  в случае же двукратного интеграла  $I^{ij}(h_n)$  точная формула имеет место лишь при  $i = j$ :

$$I^{ii}(h_n) = \frac{1}{2}((\Delta W_n^i)^2 - \Delta t_n),$$

в остальных же случаях при  $i \neq j$  выразить  $I^{ij}(h_n)$  через приращения  $\Delta W_n^\alpha$  и  $\Delta t_n$  в конечном виде не представляется возможным, поэтому остается лишь использовать численную аппроксимацию.

В книге [21] приведены следующие формулы для аппроксимации двукратного интеграла Ито  $I^{ij}$ :

$$I^{ij}(h_n) = \frac{\Delta W_n^i \Delta W_n^j - h_n \delta^{ij}}{2} + A^{ij}(h_n),$$

$$A^{ij}(h_n) = \frac{h}{2\pi} \sum_{k=1}^{\infty} \frac{1}{k} (V_k^i (U_k^j + \sqrt{2/h_n} \Delta W_n^j) - V_k^j (U_k^i + \sqrt{2/h_n} \Delta W_n^i)),$$

$\Delta W_n^i \sim \mathcal{N}(0, h)$ ,  $V_k^i \sim \mathcal{N}(0, h)$ ,  $U_k^i \sim \mathcal{N}(0, h)$ ,  $i, j = 1, \dots, m$ ;  $k = 1, \dots, \infty$ ;  $j = 1, \dots, N$ . Из формул видно, что в случае  $i = j$  получаем конечное выражение для  $I^{ij}$ , но в случае  $i \neq j$  приходится суммировать бесконечный ряд  $A^{ij}$ . Данный алгоритм дает ошибку аппроксимации порядка  $O(h^2/n)$ , где  $n$  — число оставленных слагаемых бесконечного ряда  $A^{ij}$ .

В статье [26] предложен матричный вид аппроксимирующих формул. Пусть  $\mathbf{1}_{m \times m}$ ,  $\mathbf{0}_{m \times m}$  — единичная и нулевая матрицы  $m \times m$ , тогда

$$\mathbf{I}(h_n) = \frac{\Delta \mathbf{W}_n \Delta \mathbf{W}_n - h_n \mathbf{1}_{m \times m}}{2} + \mathbf{A}(h_n),$$

$$\mathbf{A}(h_n) = \frac{h}{2\pi} \sum_{k=1}^{\infty} \frac{1}{k} (\mathbf{V}_k (\mathbf{U}_k + \sqrt{2/h_n} \Delta \mathbf{W}_n)^T - (\mathbf{U}_k + \sqrt{2/h_n} \Delta \mathbf{W}_n) \mathbf{V}_k^T),$$

где  $\Delta \mathbf{W}_n$ ,  $\mathbf{V}_k$ ,  $\mathbf{U}_k$  — независимые нормально распределенные многомерные случайные величины:

$$\Delta \mathbf{W}_n = (\Delta W_n^1, \Delta W_n^2, \dots, \Delta W_n^m)^T \sim \mathcal{N}(\mathbf{0}_{m \times m}, h, \mathbf{1}_{m \times m}),$$

$$\mathbf{V}_k = (V_k^1, V_k^2, \dots, V_k^m)^T \sim \mathcal{N}(\mathbf{0}_{m \times m}, \mathbf{1}_{m \times m}), \quad \mathbf{U}_k = (U_k^1, U_k^2, \dots, U_k^m)^T \sim \mathcal{N}(\mathbf{0}_{m \times m}, \mathbf{1}_{m \times m}),$$

### Слабая и сильная сходимости аппроксимирующей функции

Необходимо определить критерий точности аппроксимации процесса  $\mathbf{x}(t)$  функцией  $\mathbf{x}_n$ . Таких критериев принято выделять два: слабый и сильный [25, 32, 33].

Последовательность сеточных аппроксимирующих функций  $\mathbf{x}_1, \dots, \mathbf{x}_N$  сходится в *сильном смысле* с порядком  $p$  к решению  $\mathbf{x}(t)$  СДУ в момент времени  $T$  если существует константа  $C > 0$  и  $\delta_0 > 0$  такая что  $\forall h \in (0, \delta_0]$  выполняется условие

$$\mathbb{E}(\|\mathbf{x}(T) - \mathbf{x}_n\|) \leq Ch^p.$$

Последовательность сеточных аппроксимирующих функций  $\mathbf{x}_1, \dots, \mathbf{x}_N$  сходится в *слабом смысле* с порядком  $p$  к решению  $\mathbf{x}(t)$  СДУ в момент времени  $T$  если существует константа  $C_F > 0$  и  $\delta_0 > 0$  такая что  $\forall h \in (0, \delta_0]$  выполняется условие

$$|\mathbb{E}[F(\mathbf{x}(T))] - \mathbb{E}[F(\mathbf{x}_n)]| \leq Ch^p.$$

Здесь  $F \in C_p^{2(p+1)}(\mathbb{R}, \mathbb{R}^d)$  — непрерывный дифференцируемый функционал с полиномиальным ростом.

Если матрица  $\mathbf{G}$  обращается в ноль, то условие сильной сходимости равносильно условию сходимости для детерминированного случая. В отличие от детерминированного случая порядок сильной сходимости не обязательно является натуральным числом и может принимать дробно-рациональные значения.

Важно отметить, что выбор типа сходимости зависит от решаемой задачи. Увеличение порядка строгой сходимости ведет к увеличению точности аппроксимации траекторий решения  $\mathbf{x}(t)$  СДУ. Если же требуется вычислить, например, момент случайного процесса  $\mathbf{x}(t)$  или обобщенный функционал вида  $\mathbb{E}[F(\mathbf{x}(t))]$ , то следует повышать порядок слабой сходимости.

### Стохастические методы Рунге–Кутты

Перейдем к рассмотрению стохастических методов Рунге–Кутты. Мы приведем формулы для трех численных схем, записав для каждой из них формулы, таблицу Бутчера в общем виде и несколько таблиц с конкретным набором коэффициентов.

Простейшим численным методом для решения как скалярных уравнений, так и систем СДУ, является метод Эйлера–Маруямы, названный так в честь Гиширо Маруямы (Gisiro Maruyama), который распространил классический метод Эйлера для ОДУ на случай СДУ [27]. Метод легко обобщается на случай многомерного винеровского процесса:

$$x_n^\alpha = x^\alpha(t_n),$$

$$x_{n+1}^\alpha = x_n^\alpha + f^\alpha(t_n, x_n^\alpha) h_n + \sum_{\gamma=1}^d G_\beta^\alpha(t_n, x_n^\gamma) \Delta W_n^\beta.$$

Как видно из формул на каждом шаге для вычисления следующего значения аппроксимирующей функции требуется лишь соответствующее данному шагу приращение процесса  $W_n^\beta$ . Метод имеет сильный порядок  $(p_d, p_s) = (1.0, 0.5)$ . Величина  $p_d$  обозначает порядок точности детерминированной части численного метода, то есть той точности, которую будет давать численный метод при применении его к уравнению функцией  $G(t, x^\alpha(t)) \equiv 0$ . Величина  $p_s$  обозначает порядок приближения стохастической части уравнения.



В препринте Росслера [21] указаны две таблицы Бутчера для метода третьей стадийности  $s = 3$ :

SRK1Wm:	<table style="border-collapse: collapse; width: 100%;"> <tr><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">0</td></tr> <tr><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">0</td></tr> <tr><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">0</td></tr> <tr style="border-top: 1px solid black;"><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">0</td></tr> <tr><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">1</td><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">0</td></tr> <tr><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">-1</td><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">0</td></tr> <tr style="border-top: 1px solid black;"><td style="padding: 2px 5px;"></td><td style="padding: 2px 5px;">1</td><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">1</td><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">0</td></tr> <tr><td style="padding: 2px 5px;"></td><td style="padding: 2px 5px;"><math>\frac{1}{2}</math></td><td style="padding: 2px 5px;"><math>\frac{1}{2}</math></td></tr> </table>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	-1	0	0		1	0	0	1	0	0						$\frac{1}{2}$	$\frac{1}{2}$	SRK2Wm:	<table style="border-collapse: collapse; width: 100%;"> <tr><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">0</td></tr> <tr><td style="padding: 2px 5px;">1</td><td style="padding: 2px 5px;">1</td><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">0</td></tr> <tr><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">0</td></tr> <tr style="border-top: 1px solid black;"><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">0</td></tr> <tr><td style="padding: 2px 5px;">1</td><td style="padding: 2px 5px;">1</td><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">1</td><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">0</td></tr> <tr><td style="padding: 2px 5px;">1</td><td style="padding: 2px 5px;">1</td><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">-1</td><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">0</td></tr> <tr style="border-top: 1px solid black;"><td style="padding: 2px 5px;"></td><td style="padding: 2px 5px;"><math>\frac{1}{2}</math></td><td style="padding: 2px 5px;"><math>\frac{1}{2}</math></td><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">1</td><td style="padding: 2px 5px;">0</td><td style="padding: 2px 5px;">0</td></tr> <tr><td style="padding: 2px 5px;"></td><td style="padding: 2px 5px;"><math>\frac{1}{2}</math></td><td style="padding: 2px 5px;"><math>-\frac{1}{2}</math></td></tr> </table>	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	1	1	0	0	-1	0	0		$\frac{1}{2}$	$\frac{1}{2}$	0	1	0	0						$\frac{1}{2}$	$-\frac{1}{2}$
0	0	0	0	0	0	0																																																																																																													
0	0	0	0	0	0	0																																																																																																													
0	0	0	0	0	0	0																																																																																																													
0	0	0	0	0	0	0																																																																																																													
0	0	0	0	1	0	0																																																																																																													
0	0	0	0	-1	0	0																																																																																																													
	1	0	0	1	0	0																																																																																																													
					$\frac{1}{2}$	$\frac{1}{2}$																																																																																																													
0	0	0	0	0	0	0																																																																																																													
1	1	0	0	0	0	0																																																																																																													
0	0	0	0	0	0	0																																																																																																													
0	0	0	0	0	0	0																																																																																																													
1	1	0	0	1	0	0																																																																																																													
1	1	0	0	-1	0	0																																																																																																													
	$\frac{1}{2}$	$\frac{1}{2}$	0	1	0	0																																																																																																													
					$\frac{1}{2}$	$-\frac{1}{2}$																																																																																																													

Метод SRK1Wm имеет сильный порядок  $(p_d, p_s) = (1.0, 1.0)$ , а метод SRK2Wm сильный порядок  $(p_d, p_s) = (1.0, 2.0)$ .

Стохастические методы Рунге-Кутты слабой сходимости  $p = 2.0$  Численные методы со слабой сходимостью хорошо аппроксимируют характеристики распределения случайного процесса  $x^\alpha(t)$ . Слабый численный метод не нуждается в информации о траектории винеровского процесса  $W_n^\alpha$  и случайные величины для этих методов могут быть сгенерированы на другом вероятностном пространстве. Поэтому данный метод можно использовать с легко моделируемом на компьютере распределением. Формулы данной численной схемы самые громоздкие из трех рассматриваемых нами схем [13]:

$$X^{0i\alpha} = x_n^\alpha + \sum_{j=1}^s A_{0j}^i f^\alpha(t_n + c_0^j h_n, X^{0j\beta}) h_n + \sum_{j=1}^s \sum_{l=1}^m B_{0j}^i G_l^\alpha(t_n + c_1^j h_n, X^{lj\beta}) \hat{I}^l,$$

$$X^{ki\alpha} = x_n^\alpha + \sum_{j=1}^s A_{1j}^i f^\alpha(t_n + c_0^j h_n, X^{0j\beta}) h_n + \sum_{j=1}^s B_{1j}^i G_l^\alpha(t_n + c_1^j h_n, X^{lj\beta}) \sqrt{h_n},$$

$$\hat{X}^{kia} = x_n^\alpha + \sum_{j=1}^s A_{2j}^i f^\alpha(t_n + c_0^j h_n, X^{0j\beta}) h_n + \sum_{j=1}^s \sum_{l=1, l \neq k}^m B_{2j}^i G_l^\alpha(t_n + c_1^j h_n, X^{lj\beta}) \frac{\hat{I}^{kl}}{\sqrt{h_n}},$$

$$x_{n+1}^\alpha = x_n^\alpha + \sum_{i=1}^s a_i f^\alpha(t_n + c_1^i h_n, X^{ki\beta}) h_n$$

$$+ \sum_{i=1}^s \sum_{k=1}^m \left( b_i^1 \hat{I}^k + b_i^2 \frac{\hat{I}^{kk}}{\sqrt{h_n}} \right) G_k^\alpha(t_n + c_1^i h_n, X^{ki\beta}) + \sum_{i=1}^s \sum_{k=1}^m \left( b_i^3 \hat{I}^k + b_i^4 \sqrt{h_n} \right) G_k^\alpha(t_n + c_2^i h_n, \hat{X}^{ki\beta})$$

где  $n = 0, 1, \dots, N - 1$ ;  $i = 1, \dots, s$ ;  $\beta, k = 1, \dots, m$ ;  $\alpha = 1, \dots, d$ .

Обобщенная таблица Бутчера имеет вид

$c_0^i$	$A_{0j}^i$	$B_{0j}^i$	
$c_1^i$	$A_{1j}^i$	$B_{1j}^i$	
$c_2^i$	$A_{2j}^i$	$B_{2j}^i$	
	$a_i$	$b_i^1$	$b_i^2$
		$b_i^3$	$b_i^4$

0	0	0	0	0	0	0	0	0	0	0	0	0	
1	1	0	0	$\frac{1}{3}$	0	0	1	1	0	0	1	0	0
$\frac{5}{12}$	$\frac{25}{144}$	$\frac{35}{144}$	0	$-\frac{5}{6}$	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\frac{1}{4}$	$\frac{1}{4}$	0	0	$\frac{1}{2}$	0	0	1	1	0	0	1	0	0
$\frac{1}{4}$	$\frac{1}{4}$	0	0	$-\frac{1}{2}$	0	0	1	1	0	0	-1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	1	0	0
0	0	0	0	-1	0	0	0	0	0	0	-1	0	0
	$\frac{1}{10}$	$\frac{3}{14}$	$\frac{24}{35}$	1	-1	-1	0	1	-1		$\frac{1}{2}$	$-\frac{1}{2}$	
				$\frac{1}{2}$	$-\frac{1}{4}$	$-\frac{1}{4}$	0	$\frac{1}{2}$	$-\frac{1}{2}$		$\frac{1}{2}$	$-\frac{1}{2}$	

В численной схеме слабого стохастического метода Рунге-Кутты используются следующие случайные величины [13]:

$$\hat{f}^{kl} = \begin{cases} \frac{1}{2}(\hat{f}^k \hat{f}^l - \sqrt{h_n} \tilde{f}^k), & k < l, \\ \frac{1}{2}(\hat{f}^k \hat{f}^l - \sqrt{h_n} \tilde{f}^l), & l < k, \\ \frac{1}{2}((\hat{f}^k)^2 - h_n), & k = l, \end{cases}$$

Случайная величина  $\hat{f}^{kl}$  имеет трехточечное распределение. Это означает, что  $\hat{f}^{kl}$  может принимать три значения:  $\{-\sqrt{3h_n}, 0, \sqrt{3h_n}\}$  с вероятностями  $1/6, 2/3$  и  $1/6$  соответственно. В свою очередь случайная величина  $\hat{f}^k$  имеет двухточечное распределение и принимает два значения  $\{-\sqrt{h_n}, \sqrt{h_n}\}$  с вероятностями  $1/2$  и  $1/2$ .

### Анализ трудностей реализации стохастических численных методов Рунге-Кутты

Как видно уже из формул, стохастические методы Рунге-Кутты значительно сложнее своих классических аналогов. Кроме громоздкости формул, можно выделить еще следующие факторы, усложняющие реализацию стохастических методов в программном виде, а также их применение для численного решения СДУ.

- При выборе конкретного метода надо учитывать, какой тип сходимости необходимо обеспечить для данной конкретной задачи, а также какое из стохастических уравнений необходимо решать — в форме Ито или в форме Стратоновича. Это увеличивает количество алгоритмов, которые нужно запрограммировать.
- Для методов с сильной сходимостью большей единицы на каждом шаге необходимо решать ресурсоемкую задачу по аппроксимации двукратных стохастических интегралов.
- В численной схеме присутствуют не только матрицы и векторы, но и тензоры (четырёхмерные массивы), с которыми необходимо совершать операцию свертки по нескольким индексам. Реализация свертки через суммирование с помощью обычных циклов приводит к существенному падению производительности.
- Для использования слабых методов необходимо применять метод Монте Карло, проводя несколько серий по множеству испытаний в каждой. Так как метод Монте-Карло сходится приблизительно как  $1/\sqrt{N}$ , где  $N$  — число испытаний, то для достижения точности хотя бы  $10^{-3}$ , необходимо провести минимум  $10^6$  испытаний.

Наиболее существенное падение производительности происходит при реализации универсального алгоритма, то есть такой программы, которая может произвести расчет, используя произвольную таблицу коэффициентов. В этом случае приходится использовать большое количество вложенных циклов, для того чтобы организовать суммирование. Наличие в схемах для систем СДУ двойных сумм и сложной комбинации индексов в множителях, находящихся под знаком этих сумм, еще более усложняет задачу и число вложенных циклов вырастает до шести.

Кроме этих специфических особенностей стохастических численных методов, стоит упомянуть также несколько причин падения производительности, присущих также и классическим схемам, которые в стохастическом случае также играют роль. Очевидным способом хранения коэффициентов методов является использование массивов. Однако у явных методов, которые мы рассматриваем, матрица является ниже-диагональной и хранение ее в виде двумерного массива приводит к тому, что более чем половина выделенной для массива памяти тратится на хранение нулей.

Если изучить исходные коды популярных подпрограмм, реализующих классические явные вложенные методы Рунге—Кутты [28], то можно обнаружить, что в этих программах для хранения коэффициентов метода используется набор именованных констант, а не массивов. Это вызвано также и тем, что операции со скалярными величинами в большинстве языков программирования проводятся быстрее, чем операции с массивами.

При сохранении требования универсальности создаваемого кода и вместе с тем желание увеличить скорости вычислений и уменьшить расход памяти, привели нас к решению использовать автоматическую генерацию кода по одному шаблону для каждого отдельного метода.

Кроме выигрыша в производительности, автоматическая генерация кода, позволяет добавлять или изменять все функции за раз путем редактирования одного лишь шаблона, а не каждой отдельной функции. Это позволяет как уменьшить количество ошибок, так и генерировать различные варианты функций для разных целей.

### Краткое описание автоматической генерации кода

В качестве языка для написания генератора кода используется язык Python. Исходный код и документация созданных в нашем коллективе программ, доступны по ссылке

[https://bitbucket.org/mngev/sde\\_num\\_generation](https://bitbucket.org/mngev/sde_num_generation). Данный репозиторий содержит модуль stochastic, в котором реализован винеровский случайный процесс и численные методы типа Рунге-Кутты для скалярных и многомерных стохастических дифференциальных уравнений в форме Ито. Большая часть кода для этого модуля генерируется набором скриптов из директории generator. Вручную написан класс, реализующий многомерный винеровский процесс и методы Эйлера-Маруямы.

Для работы генератора кода мы использовали библиотеку для обработки шаблонов Jinja2 [29]. Данный шаблонизатор (template engine) разрабатывался изначально для генерации HTML страниц, однако он обладает очень гибким синтаксисом и может использоваться как универсальное средство для генерации текстовых файлов любого вида, в том числе и исходных кодов на любых языках программирования. Кроме Jinja2 мы также использовали библиотеку NumPy для работы с массивами и ускорения вычислений.

Кроме перечисленных выше двух внешних библиотек, был использован стандартный модуль fraction, который позволяет задать коэффициенты метода в виде рациональных дробей, а потом уже преобразовывать их в вещественный вид с нужным порядком точности, а также модуль typing для аннотации типов аргументов функций.

Шаблоны для генерации представляют собой файлы с исходным кодом на языке Python со вставками специальных команд шаблонизатора Jinja2. Информация о коэффициентах методов хранится отдельно, в структурированном виде в формате JSON. Это позволяет легко добавлять новые методы и изменять старые путем редактирования и JSON файлов. В настоящее время генерируются методы с коэффициентами, представленными в работах [12,13] и [21].

В качестве языка для уже сгенерированных функций используется сам python с активным применением библиотеки NumPy, которая позволяет получить приемлемую производительность. Однако генерируемый код легко можно переформатировать так, чтобы он соответствовал синтаксису любого другого языка программирования. Мы планируем доработать программу, для генерации кода на языке Julia (julia.org). Данный язык был представлен в 2012 году и изначально ориентирован на научные вычисления. В настоящее время он интенсивно развивается и набирает популярность. На сегодняшний день актуальной является версия 0.6.1. Julia позволяет получить производительность, сравнимую с C++ и Fortran, но при этом является динамическим языком с возможностью интерактивной командной строки (REPL) наподобие IPython и может интегрироваться в интерактивную среду Jupyter. Текущая версия библиотеки превосходит описанную авторами в статье [1]. Применение автогенерации позволит избавиться от большого количества вложенных циклов, выделения лишней памяти для хранения нулевых коэффициентов, а также значительно упростило код.

### Тестирование созданных программ

Для тестирования созданных программ мы использовали стохастические численные методы с известными аналитическими решениями. Большой список скалярных СДУ с аналитическими решениями представлен в книге [21], а для многомерного случая мы использовали уравнение Блека-Шоулза в форме Ито, представленное в книге [36]. Для проверки корректности работы генератора, был создан шаблон для генерации формул в формате LaTeX, которые затем проверялись на корректность визуально.

В качестве одномерного уравнения для тестирования методов с сильной сходимостью выбрано уравнение логарифмического блуждания (применяется в финансовой математике для моделирования рынков). Оно имеет следующий вид:

$$dx(t) = \mu x(t)dt + \sigma x(t)dW(t), x(0) = x_0,$$

где  $W(t)$  — скалярный процесс Винера, а  $x_0$  — начальное значение. Его аналитическое решение выражается через процесс Винера в конечном виде:

$$x(t) = x_0 \exp((\mu - 0.5\sigma^2)t + \sigma W(t)).$$

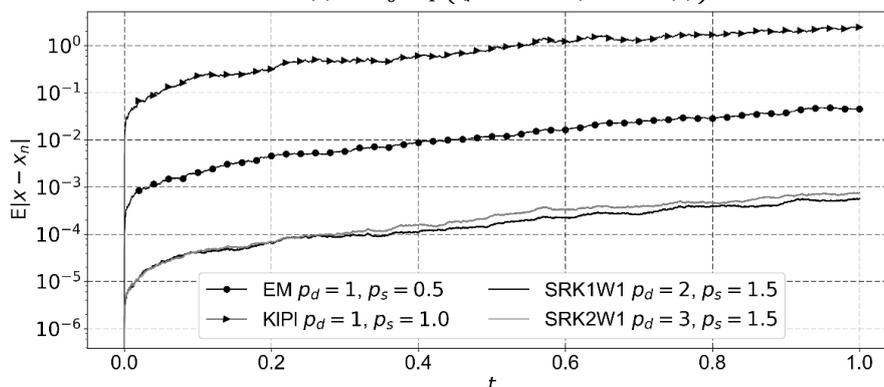


Рис. 1 Абсолютная локальная погрешность скалярных стохастических методов с сильной сходимостью

При вычислениях будем использовать коэффициенты  $\mu = 2$ ,  $\sigma = 1$  и  $x_0 = 1$ . Результаты тестирования погрешности представлены на рисунке 1. Аббревиатурой EM обозначен метод Эйлера-Маруямы, коэффициенты трех других методов представлены выше в пункте «Стохастические методы Рунге-Кутты». Шаг дискретизации  $h$  равен 0.001. Результаты согласуются с теоретической оценкой сильной погрешности.

Уравнение Блека-Шоулза в форме Ито имеет следующие вектор сдвига и матрицы диффузии:

$$f(x_1, x_2) = \begin{pmatrix} a_1 x_1 \\ a_2 x_2 \end{pmatrix}, G(x_1, x_2) = \begin{bmatrix} b_1 x_1 & 0 \\ b_2 \rho x_2 & b_2 \sqrt{1 - \rho^2} \end{bmatrix},$$

Значения параметров выберем следующие  $a_1 = a_2 = 0.1$ ,  $b_1 = b_2 = 0.2$ ,  $\rho = 0.8$ . Начальные значения:  $x_1(0) = x_2(0) = 1$ . Аналитическое решение имеет следующий вид:

$$\begin{cases} x^1(t) = x_0^1 \exp((a_1 - 0.5b_1^2)t + b_1 W^1(t)) \\ x^2(t) = x_0^2 \exp((a_2 - 0.5b_2^2)t + b_2(\rho W^1(t) + \sqrt{1 - \rho^2} W^2(t))) \end{cases}$$

Погрешность для метода, обозначенного нами выше как SRK1Wm изображена на рисунке 2. Для второго метода (SRK2Wm) погрешность вычислений аналогична.

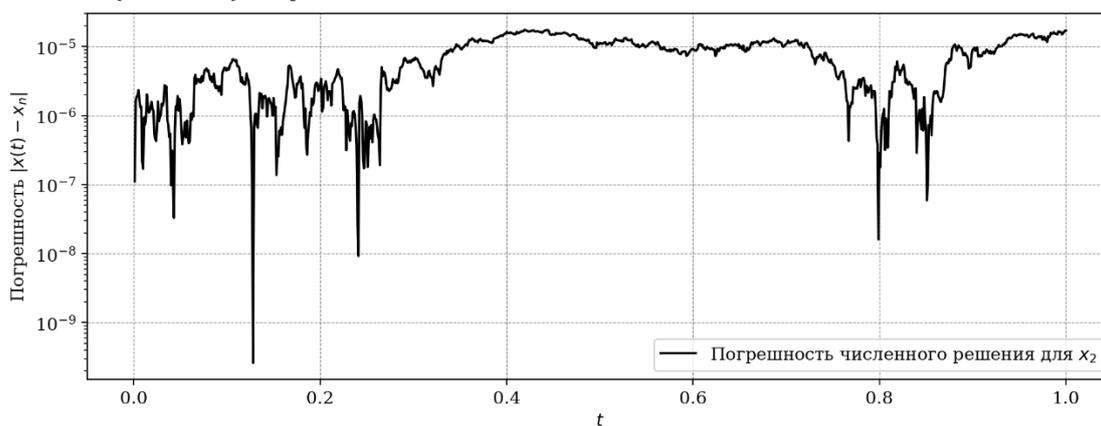


Рис. 2 Абсолютная локальная погрешность векторного стохастического метода SRK1Wm

Проведенные тесты позволяют убедиться в корректности генерируемого кода. С помощью «магической» команды %timeit оболочки iPython мы также оценили производительность векторных численных стохастических методов. Для случая уравнения Блека-Шоулза время работы функции strong\_srklw2 (реализует метод SRK2Wm для размерности 2) на сравнительно медленном процессоре Intel i3-4100 составило около секунды, что вполне приемлемо в случае разовых вычислений.

## Заключение

В данной работе рассмотрены стохастические численные схемы с порядком сходимости выше 0.5. Показано, что такие методы существенно сложнее эквивалентных численных методов для систем обыкновенных дифференциальных уравнений. Выделены их особенности, делающие эффективную программную реализацию таких методов не тривиальной задачей. В заключительной части статьи обсуждается подход, основанный на автоматической генерации программного кода, который позволяет получить эффективную реализацию рассматриваемых методов, давая возможность использовать любую таблицу коэффициентов. Дано краткое описание программы, созданной авторами и ссылка на репозиторий с исходными кодами.

## Благодарности

Публикация подготовлена при поддержке программы РУДН «5-100» и при финансовой поддержке РФФИ в рамках научных проектов РФФИ № 15-07-08795, 16-07-00556.

## Литература

1. Gevorkyan M. N., Velieva T. R., Korolkova A. V., Kulyabov D.S., Sevastyanov L.A. Stochastic Runge-Kutta Software Package for Stochastic Differential Equations // Dependability Engineering and Complex Systems. -- Springer International Publishing, 2016. -- Vol. 470. -- P. 169-179. -- 1606.06604
2. Python Reference Manual: Rep. ; Executor: Guido Rossum. — Amsterdam, The Netherlands, The Netherlands: 1995.
3. Jones Eric, Oliphant Travis, Peterson Pearu et al. SciPy: Open source scientific tools for Python. — 2001-. — [Online; accessed 19.01.2017]. Access mode: <http://www.scipy.org/>
4. Bachelier L. Théorie de la spéculation // Annales Scientifiques de l'École Normale Supérieure. — 1900. — Vol. 3, no. 17. — P. 21-86.
5. Лукшин А. В., Смирнов С. Н. Численные методы решения стохастических дифференциальных уравнений // Вычислительные алгоритмы и методы. -- 1990. -- Т. 2, № 11. -- С. 108-121.

6. Ерешко А. Ф., Филатова Д. В. Анализ явных численных методов решения стохастических дифференциальных уравнений // Труды ИСА РАН. Динамика неоднородных систем.—2008.—Т. 32, № 2.—С. 164–173.
7. Milstein G.N. Approximate Integration of Stochastic Differential Equations // *Theory Probab. Appl.*—1974.—no. 19.—P. 557–562.
8. Milstein G. N. A Method of Second-Order Accuracy Integration of Stochastic Differential Equations // *Theory Probab. Appl.*—1979.—no. 23.—P. 396–401.
9. Milstein G. N. Weak Approximation of Solutions of Systems of Stochastic Differential Equations // *Theory Probab. Appl.*—1986.—no. 30.—P. 750–766.
10. Rößler Andreas. Runge-Kutta Methods for the Numerical Solution of Stochastic Differential Equations : Ph.D. thesis / Andreas Rößler ; Technischen Universität Darmstadt. — Darmstadt, 2003.— februar.
11. Hairer E., Nørsett S.P., Wanner G. Solving Ordinary Differential Equations I. — 2edition. — Berlin : Springer, 2008.—ISBN: 978-3-540-56670-0 .
12. Debrabant K., Rößler A. Continuous weak approximation for stochastic differential equations // *Journal of Computational and Applied Mathematics.* — 2008. — no. 214. — P. 259–273.
13. Debrabant K., Rößler A. Classification of Stochastic Runge-Kutta Methods for the Weak Approximation of Stochastic Differential Equations / Technische Universität Darmstadt, Fachbereich Mathematik.—2013.— Mar.— arXiv:1303.4510v1.
14. Rößler A. Strong and Weak Approximation Methods for Stochastic Differential Equations – Some Recent Developments / Department Mathematik. Schwerpunkt Mathematische Statistik und Stochastische Prozesse.—2010
15. Burrage K., Burrage P. M. High strong order explicit Runge-Kutta methods for stochastic ordinary differential equations // *Appl. Numer. Math.* — 1996. — no. 22. — P. 81–101.
16. Burrage K., Burrage P. M., Belward J. A. A bound on the maximum strong order of stochastic Runge-Kutta methods for stochastic ordinary differential equations. // *BIT.* — 1997. — no. 37. — P. 771–780.
17. Burrage K., Burrage P. M. General order conditions for stochastic Runge-Kutta methods for both commuting and non-commuting stochastic ordinary differential equation systems // *Appl. Numer. Math.*—1998.—no. 28.— P. 161–177.
18. Burrage P. M. Runge-Kutta Methods for Stochastic Differential Equations: Ph.D. thesis /P. M. Burrage ; University of Queensland.— Australia, 1999.
19. Burrage K., Burrage P. M. Order conditions of stochastic Runge-Kutta methods by B-series // *SIAM J. Numer. Anal.*—2000.— no. 38.— P. 1626–1646.
20. Soheili A. R., Namjoo M. Strong approximation of stochastic differential equations with Runge-Kutta methods // *World Journal of Modelling and Simulation.* — 2008. — Vol. 4, no. 2. — P. 83–93.
21. Kloeden Peter E., Platen Eckhard. Numerical Solution of Stochastic Differential Equations.— 2 edition.—Berlin Heidelberg New York : Springer, 1995. — ISBN: 3-540-54062-8
22. Komori Y., Mitsuri T. Stable ROW-Type Weak Scheme for Stochastic Differential Equations// *RIMS Kokyuroku.*—1995.—no. 932.— P. 29–45.
23. Mackevičius V. Second-order weak approximations for stratonovich stochastic differential equations // *Lithuanian Mathematical Journal.* — 1994. — Vol. 34, no. 2. — P. 183–200. — Access mode: <http://dx.doi.org/10.1007/BF02333416>
24. Tocino A., Ardanuy R. Runge-Kutta methods for numerical solution of stochastic differential equations // *Journal of Computational and Applied Mathematics.* — 2002. — No. 138. — P. 219–241.
25. Øksendal Bernt. Stochastic differential equations. An introduction with applications. — 6 edition.—Berlin Heidelberg New York: Springer, 2003. — ISBN: 3-540-04758-1
26. Wiktorsson M. Joint characteristic function and simultaneous simulation of iterated Itô integrals for multiple independent Brownian motions // *The Annals of Applied Probability.* — 2001. — Vol. 11, no. 2.—P. 470–487.
27. Maruyama G. Continuous Markov processes and stochastic equations // *Rendiconti del Circolo Matematico.*—1955.—no. 4. —P. 48–90.
28. Burrage Kevin, Burrage Pamela M. Low rank Runge-Kutta methods, symplecticity and stochastic Hamiltonian problems with additive noise // *Journal of Computational and Applied Mathematics.* — 2012. — Vol. 236, no. 16. — P. 3920–3930. — Access mode: <http://www.sciencedirect.com/science/article/pii/S0377042712001240>.
29. Ma Qiang, Ding Xiaohua. Stochastic symplectic partitioned Runge-Kutta methods for stochastic Hamiltonian systems with multiplicative noise // *Applied Mathematics and Computation.* — 2015. — Vol. 252, no. Supplement C. — P. 520–534. — Access mode: <http://www.sciencedirect.com/science/article/pii/S0096300314016993>.
30. Stochastic symplectic Runge-Kutta methods for the strong approximation of Hamiltonian systems with additive noise / Weien Zhou, Jingjing Zhang, Jialin Hong, Songhe Song// *Journal of Computational and Applied Mathematics.* — 2017. — Vol. 325, no. Supplement C. — P. 134–148. — Access mode: <http://www.sciencedirect.com/science/article/pii/S0377042717302285>.
31. Amiri Sadegh, Hosseini S. Mohammad. Stochastic Runge-Kutta Rosenbrock type methods for SDE systems // *Applied Numerical Mathematics.* — 2017. — Vol. 115, no. Supplement C. — P. 1–15. — Access mode: <http://www.sciencedirect.com/science/article/pii/S0168927416302355>.
32. Булинский А.В., Ширяев А. Н. Теория случайных процессов. --5 изд.--Москва: ФИЗМАТЛИТ, 2003.--ISBN: 5-9221-0335-0
33. Кузнецов Д. Ф. Стохастические Дифференциальные Уравнения: Теория и Практика Численного Решения. -- 4 изд. -- Санкт-Петербург : Издательство Политехнического Университета, 2010.
34. Fortran and Matlab Codes. — Access mode: <https://www.unige.ch/~hairer/software.html>.
35. Jinja2 official site.—Access mode: <http://http://jinja.pocoo.org>.
36. Platen Eckhard, Bruti-Liberati Nicola. Numerical Solution of Stochastic Differential Equations with Jumps in Finance . — 2 edition. — Heidelberg Dordrecht London New York : Springer, 2010.—ISBN: 978-3-642-12057-2

## References

1. Gevorkyan, M. N.; Veliava, T. R.; Korolkova, A. V.; Kulyabov, D. S. & Sevastyanov, L. A. (2016), Stochastic Runge-Kutta Software Package for Stochastic Differential Equations'Dependability Engineering and Complex Systems', Springer International Publishing, , pp. 169--179.
2. Rossum, G. (1995), 'Python Reference Manual', CWI (Centre for Mathematics and Computer Science), Amsterdam, The Netherlands, The Netherlands.
3. Jones, E.; Oliphant, T.; Peterson, P. & others (2001), 'SciPy: Open source scientific tools for Python', [Online; accessed 19.01.2017].
4. Bachelier, L. (1900), 'Théorie de la spéculation', *Annales Scientifiques de l'École Normale Supérieure* 3(17), 1986.
5. Lukshin A., V. & Smirnov S., N. (1990), 'Chislennyye metody reshenija stohasticheskikh differencial'nyh uravnenij', *Vychislitel'nye algoritmy i metody* 2(11), 108–121.
6. Ereshko A., F. & Filatova D., V. (2008), 'Analiz javnyh chislennyh metodov reshenija stohasticheskikh differencial'nyh uravnenij',

- Trudy ISA RAN. Dinamika neodnorodnyh sistem 32(2), 164–173.
7. Milstein, G. N. (1986), 'Weak Approximation of Solutions of Systems of Stochastic Differential Equations', Theory Probab. Appl.(30), 750--766.
  8. Milstein, G. N. (1979), 'A Method of Second-Order Accuracy Integration of Stochastic Differential Equations', Theory Probab. Appl.(23), 396--401.
  9. Milstein, G. N. (1974), 'Approximate Integration of Stochastic Differential Equations', Theory Probab. Appl.(19), 557--562.
  10. Rößler, A. (2003), 'Runge-Kutta Methods for the Numerical Solution of Stochastic Differential Equations', PhD thesis, Technischen Universität Darmstadt.
  11. Hairer, E.; Nørsett, S. P. & G.Wanner (2008), Solving Ordinary Differential Equations I, Springer, Berlin.
  12. Debrabant, K. & Rößler, A. (2013), 'Classification of Stochastic Runge–Kutta Methods for the Weak Approximation of Stochastic Differential Equations', Technische Universität Darmstadt, Fachbereich Mathematik, arXiv:1303.4510v1.
  13. Debrabant, K. & Rößler, A. (2008), 'Continuous weak approximation for stochastic differential equations', Journal of Computational and Applied Mathematics(214), 259--273.
  14. Burrage, K. & Burrage, P. M. (2000), 'Order conditions of stochastic Runge-Kutta methods by B-series', SIAM J. Numer. Anal.(38), 1626--1646.
  15. Rößler, A. (2010), 'Strong and Weak Approximation Methods for Stochastic Differential Equations – Some Recent Developments', Department Mathematik. Schwerpunkt Mathematische Statistik und Stochastische Prozesse.
  16. Burrage, K. & Burrage, P. M. (1998), 'General order conditions for stochastic Runge-Kutta methods for both commuting and non-commuting stochastic ordinary differential equation systems', Appl. Numer. Math.(28), 161--177.
  17. Burrage, K. & Burrage, P. M. (1996), 'High strong order explicit Runge-Kutta methods for stochastic ordinary differential equations', Appl. Numer. Math.(22), 81--101.
  18. Burrage, K.; Burrage, P. M. & Belward, J. A. (1997), 'A bound on the maximum strong order of stochastic Runge-Kutta methods for stochastic ordinary differential equations.', BIT(37), 771-780.
  19. Burrage, P. M. (1999), 'Runge-Kutta Methods for Stochastic Differential Equations', PhD thesis, University of Queensland.
  20. Soheili, A. R. & Namjoo, M. (2008), 'Strong approximation of stochastic differential equations with Runge–Kutta methods', World Journal of Modelling and Simulation 4(2), 83--93.
  21. Kloeden, P. E. & Platen, E. (1995), Numerical Solution of Stochastic Differential Equations, Springer, Berlin Heidelberg New York.
  22. Komori, Y. & Mitsuri, T. (1995), 'Stable ROW-Type Weak Scheme for Stochastic Differential Equations', RIMS Kokyuroku(932), 29--45.
  23. Mackevičius, V. (1994), 'Second-order weak approximations for stratonovich stochastic differential equations', Lithuanian Mathematical Journal 34(2), 183-200.
  24. Tocino, A. & Ardanuy, R. (2002), 'Runge–Kutta methods for numerical solution of stochastic differential equations', Journal of Computational and Applied Mathematics(138), 219--241.
  25. Øksendal, B. (2003), Stochastic differential equations. An introduction with applications, Springer, Berlin Heidelberg New York.
  26. Wiktorsson, M. (2001), 'Joint characteristic function and simultaneous simulation of iterated Itô integrals for multiple independent Brownian motions', The Annals of Applied Probability 11(2), 470-487.
  27. Maruyama G. Continuous Markov processes and stochastic equations // Rendiconti del Circolo Matematico.—1955.—no. 4. —P. 48–90.
  28. Burrage Kevin, Burrage Pamela M. Low rank Runge–Kutta methods, symplecticity and stochastic Hamiltonian problems with additive noise // Journal of Computational and Applied Mathematics. — 2012. — Vol. 236, no. 16. — P. 3920–3930. — Access mode: <http://www.sciencedirect.com/science/article/pii/S0377042712001240>.
  29. Ma Qiang, Ding Xiaohua. Stochastic symplectic partitioned Runge–Kutta methods for stochastic Hamiltonian systems with multiplicative noise // Applied Mathematics and Computation. — 2015. — Vol. 252, no. Supplement C. — P. 520–534. — Access mode: <http://www.sciencedirect.com/science/article/pii/S0096300314016993>.
  30. Stochastic symplectic Runge–Kutta methods for the strong approximation of Hamiltonian systems with additive noise / Weien Zhou, Jingjing Zhang, Jialin Hong, SongheSong // Journal of Computational and Applied Mathematics. — 2017. — Vol. 325, no. Supplement C. — P. 134– 148. — Access mode: <http://www.sciencedirect.com/science/article/pii/S0377042717302285>.
  31. Amiri Sadegh, Hosseini S. Mohammad. Stochastic Runge–Kutta Rosenbrock type methods for SDE systems // Applied Numerical Mathematics. — 2017. — Vol. 115, no. Supplement C. — P. 1–15. — Access mode: <http://www.sciencedirect.com/science/article/pii/S0168927416302355>.
  32. Bulinskij A.V., Shirjaev A. N. Teorija sluchajnyh processov. --5 izd. --Moskva: FIZMATLIT, 2003.--ISBN: 5-9221-0335-0
  33. Kuznecov D. F. Stohasticheskie Differencial'nye Uravnenija: Teorija i Praktika Chislennogo Reshenija. -- 4 izd. -- Sankt-Peterburg: Izdatel'stvo Politehnicheskogo Universiteta, 2010.
  34. Maruyama, G. (1955), 'Continuous Markov processes and stochastic equations', Rendiconti del Circolo Matematico (4), 48--90.
  35. Fortran and Matlab Codes <http://jinja.pocoo.org>
  36. Platen Eckhard, Bruti-Liberati Nicola. Numerical Solution of Stochastic Differential Equations with Jumps in Finance. — 2 edition. — Heidelberg Dordrecht London New York: Springer, 2010.—ISBN: 978-3-642-12057-2

#### Об авторах:

**Кулябов Дмитрий Сергеевич**, кандидат физико-математических наук, доцент, доцент кафедры прикладной информатики и теории вероятностей, Российский университет дружбы народов; сотрудник Лаборатории информационных технологий, Объединенный институт ядерных исследований, [kulyabov\\_ds@rudn.university](mailto:kulyabov_ds@rudn.university)

**Геворкян Мигран Нельсонович**, кандидат физико-математических наук, доцент кафедры прикладной информатики и теории вероятностей, Российский университет дружбы народов, [gevorkyan\\_mn@rudn.university](mailto:gevorkyan_mn@rudn.university)

**Демидова Анастасия Вячеславовна**, кандидат физико-математических наук, старший преподаватель кафедры прикладной информатики и теории вероятностей, Российский университет дружбы народов, [demidova\\_av@rudn.university](mailto:demidova_av@rudn.university)

**Королькова Анна Владиславовна**, кандидат физико-математических наук, доцент, доцент кафедры прикладной информатики и теории вероятностей, Российский университет дружбы народов,

[korolkova\\_av@rudn.university](mailto:korolkova_av@rudn.university)

**Севастьянов Леонид Антонович**, доктор физико-математических наук, профессор, профессор кафедры прикладной информатики и теории вероятностей, Российский университет дружбы народов, [sevastianov\\_la@rudn.university](mailto:sevastianov_la@rudn.university)

**Котюков Михаил Михайлович**, аспирант кафедры прикладной информатики и теории вероятностей, Российский университет дружбы народов, [kotukov\\_mm@rudn.university](mailto:kotukov_mm@rudn.university)

**Note on the authors:**

**Kulyabov Dmitry S.**, candidate of physics and mathematics, associate professor of Department of Applied Probability and Informatics, Peoples' Friendship University of Russia; Employee of the Laboratory of Information Technologies, Joint Institute for Nuclear Research, [kulyabov\\_ds@rudn.university](mailto:kulyabov_ds@rudn.university)

**Gevorkyan Migran N.**, candidate of physics and mathematics, associate professor of Department of Applied Probability and Informatics, Peoples' Friendship University of Russia, [gevorkyan\\_mn@rudn.university](mailto:gevorkyan_mn@rudn.university)

**Demidova Anastasiya V.**, candidate of physics and mathematics, assistant professor of Department of Applied Probability and Informatics, Peoples' Friendship University of Russia, [demidova\\_av@rudn.university](mailto:demidova_av@rudn.university)

**Korolkova Anna V.**, candidate of physics and mathematics, associate professor of Department of Applied Probability and Informatics, Peoples' Friendship University of Russia, [korolkova\\_av@rudn.university](mailto:korolkova_av@rudn.university)

**Sevastianov Leonid A.**, doctor of physics and mathematics, full professor of Department of Applied Probability and Informatics, Peoples' Friendship University of Russia, [sevastianov\\_la@rudn.university](mailto:sevastianov_la@rudn.university)

**Kotukov Mikhail M.**, postgraduate student, professor of Department of Applied Probability and Informatics, Peoples' Friendship University of Russia, [kotukov\\_mm@rudn.university](mailto:kotukov_mm@rudn.university)