

МЕТОД МОНТЕ-КАРЛО ДЛЯ РЕШЕНИЯ СИСТЕМ ОДУ¹

С.М.Ермаков¹, Т.М.Товстик¹

¹Санкт-Петербургский государственный университет, Российская Федерация, 199034, Санкт-Петербург, Университетская наб., 7/9.

sergey.ermakov@mail.ru, peter.tovstik@mail.ru

Рассматривается применение метода Монте-Карло для решения задач Коши для систем линейных и нелинейных обыкновенных дифференциальных уравнений. Метод Монте-Карло актуален для решения больших систем уравнений и в случае малой гладкости исходных функций. Система приводится к эквивалентной системе интегральных уравнений Вольтерра. Для линейных систем это преобразование позволяет снять ограничения, связанные со сходимостью мажорантного процесса. Приводятся примеры оценок функционалов от решения и обсуждается поведение их дисперсий. В общем случае промежуток интегрирования разбивается на конечные интервалы, на которых нелинейная функция аппроксимируется полиномом. Полученное интегральное уравнение решается с помощью ветвящихся цепей Маркова с поглощением. Обсуждаются возникающие при этом задачи распараллеливания алгоритмов. В качестве примера рассмотрено одномерное уравнение с кубической нелинейностью. Обсуждается выбор переходных плотностей ветвящегося процесса. Подробно описывается метод поколений. Дано сравнение численных результатов с решением, полученным методом Рунге-Кутты. Библиогр. 9 назв.

Ключевые слова: метод Монте-Карло, системы ОДУ, интегральные уравнения, статистическое моделирование.

1. Введение. Теория методов Монте-Карло для решения линейных краевых задач математической физики в настоящее время достаточно хорошо развита [1, 2]. Обычно задачу сводят к эквивалентному интегральному уравнению. Как известно, интегральные уравнения 2-го рода тесно связаны с цепями Маркова и наиболее распространенные алгоритмы основаны на моделировании N независимых траекторий цепи Маркова, связанной с соответствующим интегральным уравнением условием согласования. При этом предполагается существование итерационного решения мажорантного интегрального уравнения. Последнее условие может быть довольно ограничительным, и имеется ряд работ [3, 4], где предложены специальные методы для снятия этих ограничений. Для нелинейных уравнений используются различные методы последовательной аппроксимации, среди которых особое место занимает приближенная замена нелинейности общего вида полиномиальной нелинейностью, что позволяет строить алгоритмы с естественным параллелизмом.

Что касается задач с начальными условиями, то алгоритмы методов Монте-Карло для них мало исследованы даже в случае ОДУ. Можно сослаться лишь на работу [5], где имеются примеры решения методом Монте-Карло систем ОДУ. В то же время имеются важные прикладные задачи, требующие решения больших систем такого рода или систем определенного вида (например, стохастических дифференциальных уравнений). В указанных случаях метод Монте-Карло может быть более эффективным, чем классические вычислительные методы.

¹Работа поддержана грантом РФФИ №17-01-00267-а.

Начальная краевая задача для систем ОДУ приводится к системам интегральных уравнений типа Вольтерра. Это позволяет для линейных систем снять ограничения, связанные со сходимостью мажорантного процесса теоретически для всех времен $t \in [0, \infty)$. Относительно некоторых классов нелинейных систем также возможен подход, основанный на приближении полиномами, но теорема Пикара диктует ограничения на длину промежутка, где существует итерационное решение, и метод Монте-Карло работает без синхронизации только на этом промежутке. При этом возникает стохастическая погрешность, которая может накапливаться в процессе вычислений, поэтому необходим анализ ее поведения.

2. Метод, основанный на цепях Маркова. Наиболее распространенным алгоритмом метода Монте-Карло является алгоритм, основанный на моделировании цепей Маркова.

Пусть μ — вероятностная мера, и требуется решить интегральное уравнение

$$\varphi(x) = \int k(x, y) \varphi(y) \mu(dy) + f(x), \quad (\text{mod } \mu), \quad (1)$$

где k и f — заданные функции, $(\text{mod } \mu)$ означает, что равенство выполнится на носителе меры μ .

Предположим, что сходится последовательность $(\bar{\varphi}_n, h)$ при $n \rightarrow \infty$ для всех h из некоторого множества H функций, где $\bar{\varphi}_n$, $n = 0, 1, \dots$ — последовательность функций, определяемых равенством

$$\bar{\varphi}_n(x) = \int |k(x, y)| \bar{\varphi}_{n-1}(y) \mu(dy) + |f(x)|, \quad (\text{mod } \mu), \quad \bar{\varphi}_0 = |f| \quad (2)$$

и $(\bar{\varphi}_n, h) = \int \bar{\varphi}_n(x) h(x) \mu(dx)$. Алгоритм (2) приводит к решению уравнения

$$\bar{\varphi}(x) = \int |k(x, y)| \bar{\varphi}(y) \mu(dy) + |f(x)|, \quad (\text{mod } \mu), \quad (3)$$

которое является мажорантным по отношению к (1). Алгоритм (2) сходится, если сходится ряд Неймана и в этом случае выполняется равенство

$$\bar{\varphi}(x) = |f(x)| + \sum_{m=1}^{\infty} \int |k(x, x_1)| |k(x_1, x_2)| \cdots |k(x_{m-1}, x_m)| |f(x_m)| \otimes_{i=1}^m \mu(dx_i). \quad (4)$$

Выберем цепь Маркова, определяемую начальной плотностью распределения $p^0(x)$, переходной плотностью $p(x, y)$ (по отношению к мере μ) и вероятностью поглощения $g(x)$ так, что выполняются равенства

$$\int p(x, y) \mu(dy) = 1 - g(x), \quad (\text{mod } \mu), \quad 0 \leq g(x) \leq 1 \quad (5)$$

и условия согласования

$$\begin{aligned} p^0(x) &> 0, & \text{если} & & h(x) &\neq 0, \\ p(x, y) &> 0, & \text{если} & & k(x, y) &\neq 0, \\ g(x) &> 0, & \text{если} & & f(x) &\neq 0. \end{aligned} \quad (6)$$

Моделируя с помощью известных алгоритмов траектории этой цепи:

$$x_0 \rightarrow x_1 \rightarrow \dots \rightarrow x_{\tau-1} \rightarrow x_\tau \quad (7)$$

в предположении, что случайный момент поглощения τ конечен, с вероятностью 1 можем оценить (вычислить) величину $(h, \bar{\varphi})$, так как справедливо следующее утверждение

$$\mathbf{E} J_a = (h, \varphi), \quad J_a = \frac{h_0 k_{0,1} \dots k_{\tau-1,\tau} f_\tau}{p_{0,1}^0 \dots p_{\tau-1,\tau} g_\tau}. \quad (8)$$

Здесь $h_0 = h(x_0)$, $k_{i,i+1} = k(x_i, x_{i+1})$, $i = 0, 1, \dots, \tau - 1$, и аналогичные обозначения имеют место для f_τ , $p_{i,j}^0$, $p_{i,j}$, g_τ . Случайную величину J_a называют оценкой "по поглощениям".

Основанные на методе Монте-Карло оценки функционала $(h, \bar{\varphi})$ используют представление решения уравнения (1) в виде ряда Неймана и предполагают его абсолютную сходимость, т.е. сходимость ряда (4).

Моделируем N независимых траекторий, для каждой из которых вычисляются значения $J_a(l)$ (l — номер траектории) и (h, φ) оценивается с помощью среднего $1/N \sum J_a(l)$. Соответствующие доказательства и модификации алгоритма содержатся в [4]. Мы привели самую простую из известных Монте-Карло оценок для (h, φ) . Для нас важно отметить параллелизм алгоритма.

Известен ряд алгоритмов Монте-Карло для решения некоторых типов нелинейных интегральных уравнений [6]. Как правило, они основаны на последовательной приближенной линеаризации задачи. Это обычно ухудшает свойства параллелизма, т.к. на каждом этапе линеаризации требуется синхронизация. Исключением являются уравнения с полиномиальной нелинейностью (уравнения Ляпунова–Шмидта) вида

$$\varphi(x) = \sum_{l=1}^{\infty} \int K_l(x, x_1, \dots, x_l) \Pi_{m=1}^l \varphi(x_m) \otimes_{m=1}^l \mu(dx_m) + f(x). \quad (9)$$

При соответствующих условиях (см. [7, 8]) функционал (h, φ) от решения этого уравнения можно оценить, моделируя ветвящийся марковский процесс. На его траекториях вычисляется аналог оценки J_a и процедура не требует дополнительной синхронизации. Практический интерес представляют сравнительно небольшие степени нелинейности.

Перечисленные выше вычислительные методы и их модификации представляют интерес при решении многомерных задач, если функция $\varphi(x)$ зависит от большого числа переменных и (или) требуется решить большие системы уравнений.

Отметим, что важные в прикладном отношении краевые задачи для уравнений в частных производных сводятся к задачам, перечисленным выше. В этой области имеется обширная литература.

Вместе с тем стохастические методы для решения задачи Коши для больших систем обыкновенных дифференциальных уравнений исследованы мало. Первые результаты в этом направлении получены в [5], где намечен общий подход, но не обсуждалась сравнительная эффективность метода и его параллелизм. Задача Коши для ОДУ однако обладает своеобразием, которое можно проследить на простейшем примере, линейной однородной задачи Коши.

Решение простейшей системы обыкновенных дифференциальных уравнений

$$Y' = AY, \quad Y = (y_1(t), \dots, y_n(t))^T, \quad Y(0) = (1, \dots, 1), \quad (10)$$

$A = \|a_{i,j}\|_{i,j=1}^n$ есть матричная экспонента, представленная рядом

$$Y(t) = \exp(At) = \sum_{l=0}^{\infty} \frac{A^l t^l}{l!}, \quad (11)$$

который абсолютно сходится при $t \in [0, \infty)$, и никаких мажорантных условий не требуется.

По аналогии со случаем интегральных уравнений функционал $\int_0^\infty h(t)Y(t) dt$ можно вычислять методом Монте-Карло, моделируя цепь Маркова. Аналогом оценки (8) в этом случае может быть

$$\tilde{J}_a = \frac{h_{i_0} a_{i_0, i_1} \dots a_{i_{\tau-1}, i_\tau} t^\tau}{p_{i_0}^\circ p_{i_0, i_1} \dots p_{i_{\tau-1}, i_\tau} g_{i_\tau} \cdot \tau!} \quad (12)$$

Дисперсия этой оценки возрастает, однако, экспоненциально с ростом t . Оценка не учитывает того факта, что соответствующее интегральное уравнение является вольтерровским. Существо дела видно при рассмотрении неоднородной задачи. Рассмотрим линейную неоднородную задачу Коши

$$\frac{dY}{dt} = A(t)Y + F(t), \quad Y(0) = Y_0, \quad (13)$$

$$A(t) = \|a_{i,j}(t)\|_{i,j=1}^n, \quad F = (f_1, \dots, f_n)^T.$$

Решение ее, как известно, может быть выражено в явном виде. Оно равно сумме решения однородного уравнения (10) и частного решения, которое представимо в виде ряда, аналогичного (4)

$$Y(t) = Y_0 e^{\int_0^t A(\tau) d\tau} + J(t) \quad (14)$$

где

$$J(t) = F(t) + \sum_{k=1}^{\infty} \int_0^t A(\tau_1) d\tau_1 \dots \int_0^{\tau_{k-1}} A(\tau_k) F(\tau_k) d\tau_k. \quad (15)$$

Наличие переменных верхних пределов в равенстве (15) требует для вычисления этого ряда привлечь неоднородную цепь Маркова. В частности, выберем в качестве такой цепи цепь с начальным распределением $p^\circ(t)$ и переходными матрицами

$$\mathcal{P}^{(k)}(\tau) = \|p_{i,j}^{(k)}(\tau)\|_{i,j=1}^n, \quad \tau \in (0, \tau_k), \quad k \geq 0, \quad \tau_0 = t, \quad (16)$$

удовлетворяющими условиям

$$p_{i,j}^{(k)}(\tau) > 0, \quad \text{если } a_{ij}(\tau) \neq 0, \quad (17)$$

и условию, налагаемому на вероятность поглощения после k -го перехода ($k=1, 2, \dots$)

$$\int_0^{\tau_k} \sum_{j=1}^n p_{i,j}^{(k)}(\tau) d\tau = 1 - g_i^{(k)}(\tau_k), \quad (18)$$

где τ_k момент k -го перехода цепи.

Имеет место следующее утверждение.

Пусть $A(t)$ имеет ограниченную в некоторой метрике норму (например, $\max_{i,t} \sum_{j=1}^n |a_{i,j}(t)| < M$), для цепи Маркова $p^\circ(t)$, $\mathcal{P}^{(k)}(\tau)$ выполнены условия (16), (17) и

$$g_i^{(k)}(t) > 0, \quad \text{если} \quad f_i(t) \neq 0,$$

и для некоторого заданного на $[0, T]$ вектора $h(t) = (h_1(t), \dots, h_n(t))$ выполнено условие $p_i(t) > 0$, если $h_i(t) \neq 0$. Тогда для случайной величины

$$\xi = \frac{h_0 a_{0,1} \dots a_{l-1,l} f_l}{p_0^\circ p_{0,1} \dots p_{l-1,l} g_l},$$

определенной на траекториях $(\tau_0, i_0) \rightarrow (\tau_1, i_1) \rightarrow \dots \rightarrow (\tau_l, i_l)$ цепи, где использованы сокращения $h_0 = h_{i_0}(\tau_0)$, $a_{k-1,k} = a_{i_{k-1}, i_k}(\tau_k)$, $f_l = f_{i_l}(\tau_l)$, $g_l = g_{i_l}^{(l)}(\tau_l)$ и т.д., справедливо равенство

$$E\xi = \int_0^t h(\tau) J(\tau) d\tau. \quad (19)$$

Доказательство этого равенства вполне аналогично доказательству такого же равенства для оценки (8), если учесть, что плотность вероятности траектории $(\tau_0, i_0) \rightarrow \dots \rightarrow (\tau_l, i_l)$ есть $p_0^\circ p_{0,1} \dots p_{l-1,l} g_l$ с учетом введенных выше сокращенных обозначений.

Очевидно, что аналогичные обобщения (введение переменной вероятности обрыва траектории) могут быть сделаны для уравнений с полиномиальной нелинейностью.

Таким образом, для решения задачи Коши для систем обыкновенных дифференциальных уравнений (ОДУ) можно предложить следующий приближенный алгоритм (общая схема).

Задача

$$\frac{d\vec{X}}{dt} = \vec{f}(t, X), \quad X = (x_1, \dots, x_s), \quad \vec{f} = (f_1, \dots, f_s), \quad X(t_0) = X_0 = (x_1^0, \dots, x_s^0) \quad (20)$$

аппроксимируется задачей

$$\frac{d\vec{X}}{dt} = \vec{P}(t, X), \quad X(t_0) = X_0,$$

где

$$\vec{P}(t, X) = (P_1(t, X), \dots, P_s(t, X)) \quad \text{и} \quad P_k(t, X) = \sum_{i_1 \dots i_s=1}^m c_{i_1 \dots i_s}(t) x_1^{i_1} \dots x_s^{i_s}.$$

Функции $c_{i_1 \dots i_s}$ подбираются из условия минимизации $\|\vec{f} - \vec{P}\|$ — некоторой выбранной исследователем нормы разности $\vec{f} - \vec{P}$ на промежутке $(t_0 - h, t_0 + h)$. Если в линейном случае при выполнении условий теоремы Пикара [9] о существовании решения системы (20) h может быть практически неограниченным, то в случае нелинейном — и условия теоремы Пикара, и требование малости нормы $\vec{f} - \vec{P}$ могут налагать на h довольно

жесткие условия. Теоретические соображения и, возможно, пробные вычисления должны дать возможность выбрать подходящие m и $\Delta t < h$ для вычисления решения системы в точке $t_0 + \Delta t$.

Если ставится задача оценки $X(T)$ при $T > (t_0 + \Delta t)$, то промежуток $(t_0, t_0 + T)$ делится на части, имеющие длину порядка Δt и решаются последовательно задачи с начальными данными $X_0, \tilde{X}(t_0 + \Delta t), \tilde{X}(t_0 + 2\Delta t), \dots$. Волна обозначает, что соответствующее значение X имеет систематическую ошибку, вызванную полиномиальной аппроксимацией и случайную ошибку метода Монте-Карло. Анализ поведения погрешности в самом общем случае вряд ли может быть содержательным. Как обычно, удовлетворительных результатов можно ожидать в случае вычислительной устойчивости алгоритма.

Следующее замечание относительно параллелизма алгоритма должно быть сделано.

В пределах одного промежутка длины Δt алгоритм метода Монте-Карло состоит в моделировании независимых (может быть ветвящихся) траекторий и обладает неограниченным параллелизмом. Для перехода на следующий интервал требуется синхронизация. Это соображение также может быть важным при выборе Δt , m и точности полиномиальной аппроксимации.

Подробному анализу поведения погрешности метода для определенных классов задач авторы предполагают посвятить дальнейшие публикации. Наиболее перспективным представляется применение его к решению стохастических дифференциальных уравнений, где он может служить источником новых алгоритмов.

Ниже приводятся иллюстративные примеры.

3. Примеры. Пусть

$$\dot{x} = \beta x^n - ax, \quad x(t_0) = x_0, \quad a > 0, \quad t_0 \leq t \leq T. \quad (21)$$

Это уравнение при $T = t$ эквивалентно интегральному

$$x(t) = \beta \int_{t_0}^t e^{-a(t-\tau)} x(\tau)^n d\tau + e^{-a(t-t_0)} x_0. \quad (22)$$

Нелинейное интегральное уравнение (22) запишем в виде

$$x(t) = \int_{t_0}^t k(t, \tau) x(\tau)^n d\tau + f(t), \quad k(t, \tau) = \beta e^{-a(t-\tau)}, \quad f(t) = e^{-a(t-t_0)} x_0. \quad (23)$$

Для вычисления решения уравнения (23) методом Монте-Карло, согласно изложенным выше алгоритмам, нужно использовать ветвящийся процесс, у которого в каждой точке ветвления рождается ровно n новых частиц. Весь интервал $(0, T)$ интегрирования нужно разбить на малые подинтервалы и искать решение на них. Будем вычислять

$$x(t_k), \quad t_k = k\Delta t, \quad k = 1, \dots, L = [T/\Delta t]. \quad (24)$$

Для нормы ядра $k(t_1, t_2)$, $t_0 \leq t_2 \leq t_1 \leq t$, должно выполняться неравенство

$$n \|k(t_1, t_2)\| = \sup_{t_0 \leq t_1 \leq t} \int_{t_0}^{t_1} n |k(t_1, t_2)| dt_2 = \frac{n|\beta|}{|a|} (1 - e^{-a(t-t_0)}) < 1, \quad (25)$$

из которого, следует найти ограничения на параметры β, a, t, t_0 .

Воспользуемся методом "поколений." В рассматриваемой задаче в начальный момент времени в точке t имеется одна частица, которая может погибнуть и тогда алгоритм заканчивается. Если частица не гибнет, то она превращается в n новых частиц, которые образуют первое поколение.

Далее каждая частица независимо от других ведет себя так же, как исходная частица. Выжившие частицы образуют следующее поколение. Алгоритм заканчивается, если на очередное поколение не переходит ни одной частицы.

Каждая частица, появившись в новом поколении в момент s , может либо погибнуть с вероятностью $g(s)$, либо превратиться в n новых частиц, в случайные моменты времени

$$\tau_s^{(k)}, \quad k = 1, \dots, n, \quad (26)$$

которые одинаково распределены и независимы. Плотность распределения $r(s, t_2)$, $t_0 \leq t_2 \leq s$, этих случайных величин, которая называется условной плотностью перехода частицы из точки s в точку $\tau_s^{(k)}$ удовлетворяет равенству

$$\int_{t_0}^s r(s, t_2) dt_2 = 1.$$

Функции $g(s)$ и $r(s, t_2)$ нужно задать и найти переходную плотность, которая равна

$$p(t_1, t_2) = (1 - g(t_1)) r(t_1, t_2), \quad (27)$$

Моделирующие формулы случайных величин $\tau_s^{(k)}$, $k = 1, \dots, n$, при любом k получаются из уравнений

$$\int_{\tau_s^{(k)}}^s r(s, t_2) dt_2 = \alpha(k), \quad (28)$$

в которых $\alpha(k)$ - независимые случайные величины, равномерно распределенные на $(0, 1)$.

На случайных моментах (26), в которых происходит изменение состава частиц, строится весовая функция

$$W(t_1, t_2) = \frac{k(t_1, t_2)}{p(t_1, t_2)}.$$

Посмотрим, какой вклад в оценку вносит исходная частица. Она с вероятностью $g(t)$ гибнет, тогда алгоритм заканчивается. Если частица не гибнет, то появляются n новых частиц. Вклад начальной частицы таков

$$J_0 = \begin{cases} \frac{f(t)}{g(t)} & \text{с вероятностью } g(t), \\ \prod_{k=1}^n \zeta^{(k)}, \quad \zeta^{(k)} = W(t, \tau_t^{(k)}) & \text{с вероятностью } 1 - g(t), \end{cases}$$

здесь $\tau_t^{(k)}$ - независимые случайные величины, которые моделируются с помощью формулы (28).

Пусть $\{t_1^{(k)}, \dots, t_{n_k}^{(k)}\}$ - координаты множества частиц $T^{(k)}$, образующих k -ое поколение. В $T^{(k)} = G^{(k)} \cup H^{(k)}$ входят частицы $G^{(k)}$, из которых образуется $k+1$ поколение

и частицы $H^{(k)}$, которые гибнут. Частицы k -го поколения вносят с соответствующими вероятностями следующий вклад в оценку

$$J_k = \prod_{t_j^{(k)} \subset G^{(k)}} \prod_{i=1}^n W(t_j^{(k)}, \tau_{t_j^{(k)}}^{(i)}) \prod_{t_l^{(k)} \subset H^{(k)}} \frac{f(t_l^{(k)})}{g(t_l^{(k)})},$$

а оценка, учитывающая состояние системы с нулевого по k -ое поколения, равна $\prod_{r=0}^k J_r$. Алгоритм заканчивается на K поколении, если множество $G^{(K+1)}$ оказывается пустым, а $G^{(K)}$ - нет. В результате, одна реализация, оценивающая решение уравнения (23), принимает вид

$$V^{(1)} = \prod_{k=0}^{K^{(1)}} J_k, \quad (29)$$

здесь $K^{(1)}$ - число поколений в первой реализации. Этот алгоритм нужно проделать N раз и при больших значениях N получаем

$$x(t) \approx \hat{x}(t)_N = \sum_{j=1}^N V^{(j)} / N.$$

Условие (25) приводит к гибели ветвящегося процесса с вероятностью единица и, следовательно, к конечности числа поколений $K^{(j)}$ при любом j .

Рассмотрим конкретную задачу, которая имеет вид (21)

$$\dot{x} = 0.4x^3 - 0.6x, \quad x(0) = 0.7, \quad \beta = 0.4, \quad a = 0.6, \quad 0 \leq t \leq 4. \quad (30)$$

Учитывая вид ядра $k(t_1, t_2) = 0.4e^{-0.6(t_1-t_2)}$, выбираем

$$r(t_1, t_2) = \frac{be^{-b(t_1-t_2)}}{1 - e^{-b(t_1-t_0)}}, \quad b \geq a, \quad t_0 \leq t_2 \leq t_1.$$

В частности, можно выбрать $b = a$. Так как переходная плотность удовлетворяет равенству (27), то в качестве функции $g(s)$ - вероятности поглощения в точке s , берем $g(s) = e^{-c(s-t_0)}$. Заметим, что с ростом номера поколения вероятности поглощения в нем возрастают.

Для сходимости алгоритма достаточно выполнения условия $|W(t_1, t_2)| < 1$, которое приводит к следующему неравенству

$$|W(t_1, t_2)| = \frac{|\beta|e^{-(a-b)(t_1-t_2)}(1 - e^{-b(t_1-t_0)})}{b(1 - e^{-c(t_1-t_0)})} < 1. \quad (31)$$

Неравенство (31) накладывает условия: $a \geq b$, $c > b$.

Нужно задать параметры b и c . Основываясь на идее существенной выборки при задании величины параметра b , можно взять $b = a$ или число близкое к a . Параметр c определяет поглощение $g(t)$ в начальный момент времени и, так как с этой вероятностью алгоритм не дойдет до первого поколения, следовательно, $g(t)$ не должно быть слишком большим.

С числом испытаний $N = 1000$ были промоделированы два варианта при шаге $\Delta t = 0.15$ и $\Delta t = 0.25$. Для проверки точности были вычислены выборочные среднеквадратические отклонения $\hat{\sigma}_N$ оценок $\hat{x}(t)_N$ и найдены величины

$$\chi = \sum_{k=1}^L I(|x(t_k) - \hat{x}(t_k)_N| > 2\hat{\sigma}_N), \quad \delta = \max_{1 \leq k \leq L} |x(t_k) - \hat{x}(t_k)_N|. \quad (32)$$

В каждом из вариантов найдены величины максимального $K_{\max}$ и среднего $\bar{K}$ (округленная величина) числа поколений, которые проявились при вычислении оценок (24).

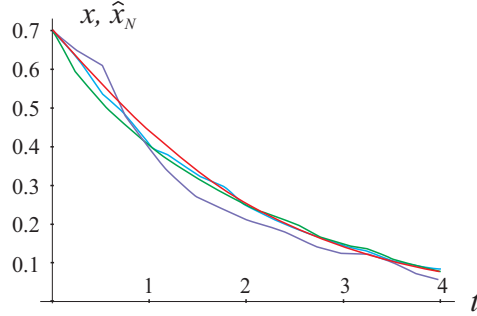


Рис.1. $\Delta t = 0.25$, $b = 0.6$.

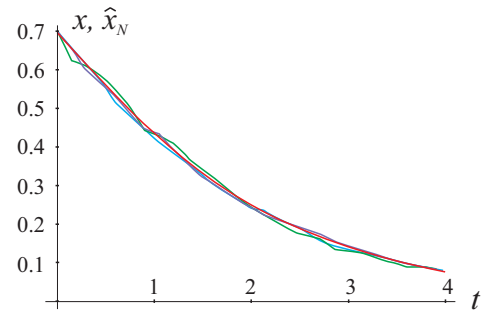


Рис.2. $\Delta t = 0.15$, $b = 0.5$.

Исходные данные для трех вариантов параметров и результаты моделирования приведены, соответственно, при $\Delta t = 0.25$ в таб.1 (рис.1) и при $\Delta t = 0.15$ таб.2 (рис.2). На каждом из рисунков построены графики $\hat{x}(t_k)_N$ в виде синей, зеленой и фиолетовой линий, которые соответствуют данным с номерами 1–3 в таблицах. Точное решение на каждом графике отмечено красной линией.

Таблица 1.

$\Delta t = 0.25$, $b = 0.6$						
№	c	$g(t)$	δ	$K_{\max}$	$\bar{K}$	χ
1	4.0	0.368	0.035	11	3	1
2	6.6	0.192	0.048	13	4	0
3	9.0	0.105	0.097	17	5	6

Таблица 2.

$\Delta t = 0.15$, $b = 0.5$						
№	c	$g(t)$	δ	$K_{\max}$	$\bar{K}$	χ
1	2.5	0.678	0.020	6	1	3
2	4.0	0.549	0.017	8	2	0
3	6.5	0.377	0.036	13	2	0

Однако не при всех значениях параметра c алгоритм приводит к хорошему приближению. Чтобы устранить это препятствие, можно, используя описанный метод, найти оценки $\hat{x}_N$ при разных значениях c и затем, усреднив их, получить $\bar{x}_N$.

В нашем примере, чтобы получить $\bar{x}_N$, найдем и усредним приближения при $c_0 = 2$, $c_i = c_0 + i1.5$, $1 \leq i \leq 4$. При $\Delta t = 0.15$ величина δ , полученная по формуле (32) с заменой $\hat{x}_N$ на $\bar{x}_N$, дает: $\delta(b = 0.6) = 0.056$ и $\delta(b = 0.5) = 0.024$, а при $\Delta t = 0.25$ получено: $\delta(b = 0.6) = 0.056$ и $\delta(b = 0.5) = 0.018$. Отсюда и из анализа обеих таблиц делаем вывод, что лучшая оценка получается, если взять $b = 0.5$.

Литература

1. Ермаков С.М., Сипин А.С. Метод Монте-Карло и параметрическая разделимость алгоритмов. СПб.: Изд-во Санкт-Петерб. ун-та, 2014, 248 с.
2. Михайлов Г.А., Войтишек А.В. Численное статистическое моделирование. Методы Монте-Карло. М: Академия, 2006. 368 с.

3. *Ermakov S.M., Wagner W.* Monte Carlo difference schemes for the wave equation. Monte-Carlo Methods and Appl. **8** (1), 2002, pp. 1–30.
4. *Ermakov S.M.* Monte Carlo method in Метод Монте-Карло в вычислительной математике (Introduction). Невский Диалект, Бином. Лаборатория знаний, 2009, 192 с.
5. *Akhtar M.N., Durad M.H., Ahmed A.* Solving initial value ordinary differential equations by Monte Carlo method. Proc. of IAM. **4** (2), 2015, pp. 149–174.
6. *Halton J.H.* Sequential Monte Carlo techniques for solving non-linear systems // Monte Carlo Methods And Applications. **12** (2), 2006, pp. 113–141.
7. *Ермаков С. М.* Об аналоге схемы Неймана-Улама в нелинейном случае. ЖВМ и МФ. **13** (3), 1973, 564–573 с.
8. *Ермаков С.М.* Метод Монте-Карло и смежные вопросы. М.: Наука, 1975. 472 с.
9. *Понтрягин Л.С.* Обыкновенные дифференциальные уравнения. М.: Наука, 1974. 331 с.

Monte Carlo method for solution of systems ODE

S.M.Ermakov, T.M.Tovstik

St.Petersburg State University, Russian Federation, 199034, St.Petersburg, Universitetskaya nab., 7/9.

sergey.ermakov@mail.ru, peter.tovstik@mail.ru

An application of the Monte Carlo method to solution of problems Cauchy for system of linear and nonlinear ordinary differential equations is considered. the Monte Carlo method is topical for solution of large systems of equations and in the cases with given functions are not smooth enough. A system is reduced to an equivalent system of integral equations of the Volterra type. For linear systems this transformation allows to avoid restrictions, connected with a convergence of a majorizing process. Examples of estimates of functionals of solutions are given, and a behavior of their dispersions are discussed. In the general case an interval of solution is divided into finite subintervals in that nonlinear functions are approximated by polynomials. An obtained integral equations are solved by using branching Markov chains with an absorption. Appearing problems of parallel algorithms are discussed. As an example an one dimensional cubic equation is considered. A method of levels of branching process is discussed in details. a comparison of numerical results with a solution, obtained by the Runge-Kutta method, is presented.

Keywords: Monte Carlo method, systems ODE, integral equations, statistic simulation

References

1. *Ermakov S.M., Sipin A.S.* Monte Carlo method and parametric separation of algorithms. St.Petersburg Univ. publ. 2014, p. 248. [in Russian]
2. *Mikhailov G.A., Voytishek A.V.* Numerical statistical modeling. Monte Carlo methods. Moscow. Akademiya, 2006, p. 368. [in Russian]
3. *Ermakov S.M., Wagner W.* Monte Carlo difference schemes for the wave equation // Monte-Carlo Methods and Appl. Vol. 8, №1, 2002, pp. 1–30.
4. *Ermakov S.M.* Monte Carlo method in quantitative mathematics (Introduction). Nevskii Dialekt, Binom. Laboratoria znaniy, 2009. 192 p. [in Russia]
5. *M.N. Akhtar, M.H. Durad, A. Ahmed* Solving initial value ordinary differential equations by Monte Carlo method. Proceeding of IAM. V.4, № 2, 2015, pp. 149–174.
6. *Halton J. H.* Sequential Monte Carlo techniques for solving non-linear systems // Monte Carlo Methods And Applications. 2006. Vol. 12. №2, pp. 113–141.

7. *Ермаков С. М.* On analogue of Neuman-Ulam scheme in the nonlinear case. JVM&MF (J vychislitel'noy matematiki i matematicheskoy fiziki) 1973. Vol. 13. №3, pp. 564–573. [in Russian]
8. *Ермаков С.М.* Monte Carlo method and related topics. Moscow: Nauka. 1975. P. 472.[in Russian]
9. *Pontryagin L.S.* Ordinary Differential Equations. Addison-Wesley, 1962. P. 304.