САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

На правах рукописи

Дмитриев Алексей Валерьевич

Стохастические и асинхронные методы решения систем уравнений (с приложениями к задачам финансовой математики)

01.01.07 – Вычислительная математика

ДИССЕРТАЦИЯ

на соискание ученой степени

кандидата физико-математических наук

Научный руководитель

доктор физико-математических наук,

профессор

Ермаков Сергей Михайлович

Санкт-Петербург – 2017

Содержание

Введение

Актуальность работы. При решении многих прикладных задач физики, биологии, финансовой математики и других дисциплин зачастую не удаётся найти их явное решение. По этой причине возникает необходимость использования приближенных методов, после применения которых исходная задача часто сводится к решению систем уравнений большой размерности.

Решение таких задач в силу их сложности целесообразно проводить на многопроцессорных системах, что налагает определенные ограничения на класс используемых алгоритмов. Такие алгоритмы должны обладать свойством параллелизма и эффективно использовать ресурсы вычислительных систем. Алгоритмы, пригодные для использование на многопроцессорных системах, можно разделить на два типа: синхронные и асинхронные.

При использовании параллельных алгоритмов так или иначе возникает необходимость координировать действия процессоров. В случае синхронных алгоритмов эта координация осуществляется путем разделения алгоритма на общие для всех процессоров этапы. На каждом этапе процессоры производят ряд операций, зависящих от результатов вычислений на предыдущих этапах. Переход к следующему этапу осуществляется только после того, как все процессоры выполнили назначенные им в рамках этапа операции. Обмен результатами вычислений между процессорами, другими словами – синхронизация, происходит в конце этапа. Некоторые процессоры при этом могут быстрее других справляться с теми операциями, которые назначены им на текущем этапе, и в результате будут, простаивая, ожидать завершения этапа.

В асинхронных алгоритмах нет этапов общих для всех процессоров, а есть свои собственные этапы для каждого процессора. Процессорам разрешается вычислять быстрее и совершать больше итераций, чем могут совершить другие процессоры. Тот факт, что такие алгоритмы эффективно загружают

систему и имеют потенциальное преимущество в скорости, делает их объектом исследования.

Так например, в работах [1], [2] были предложены асинхронные варианты метода простых итераций для решения систем уравнений. В этих работах приведены достаточные условия, при которых асинхронные итерации сходятся к решению задачи, однако эти условия довольно ограничительные, и, как было показано в диссертации, в некоторых случаях удаётся построить асинхронные алгоритмы, гарантирующий сходимость и при более слабых условиях.

Естественными свойствами асинхронности обладают также многие разновидности метода Монте-Карло для решения систем уравнений. Исследованию вопроса применения метода Монте-Карло посвящено достаточно много работ различных авторов (см., например, работы С.М. Ермакова [3–6], Г.А. Михайлова [7–9], Дж. Холтона [10–12] и др.).

Цель диссертационной работы:

- исследование метода асинхронных итераций для задач, не удовлетворяющих достаточным условиям сходимости, указанным в [1], [2];
- построение оценок метода Монте-Карло для решения систем уравнений с использованием многопроцессорных систем, исследование вопросов их стохастической устойчивости;
- построение оценок метода Монте-Карло, обладающих свойством асинхронности, для решения систем обыкновенных дифференциальных уравнений большой размерности;
- применение разработанных алгоритмов для численного решения задачи нахождения цены американского опциона.

Теоретическая и практическая ценность. Полученные результаты являются математически обоснованными и могут успешно применяться для

решения широкого класса задач, так или иначе сводящихся к решению систем уравнений, на многопроцессорных вычислительных системах. Полученные теоретические результаты могут послужить основой для дальнейших исследований асинхронных детерминированных и стохастических асинхронных методов.

На защиту выносятся следующие основные результаты и положения: построен алгоритм метода Монте-Карло с частичной синхронизацией для решения систем уравнений вида

$$x = Ax + b, \tag{1}$$

при выполнении условий $|\lambda_1(A)| < 1$ и $\lambda_1(|A|) > 1$, где $\lambda_1(\cdot)$ – наибольшее по модулю собственное число матрицы, а $|A|$ – матрица, составленная из модулей элементов матрицы A. Получены достаточные условия стохастической устойчивости предложенного алгоритма и оценен период асинхронности.

Модифицирован метод асинхронных итераций для решения задачи (1) при условии $|\lambda_1(A)| < 1$ и $\lambda_1(|A|) > 1$. Получены оценки периода асинхронности.

Получены и формально описаны оценки метода Монте-Карло, обладающие свойством асинхронности, для решения систем обыкновенных дифференциальных уравнений большой размерности. Получены достаточные условия их стохастической устойчивости.

Построены асинхронные оценки метода Монте-Карло для нахождения стоимости американского опциона, исследованы условия их стохастической устойчивости.

Апробация работы. Основные результаты диссертации докладывались и обсуждались на семинарах кафедры статистического моделирования математико-механического факультета СПбГУ, а также на международных конференциях:

- Seventh International Workshop on Simulation, Римини, Италия, Май 21-25, 2013;

- Ninth IMACS Seminar on Monte Carlo Methods, Аннеси-ле-Вьё, Франция, Июль 15-19, 2013.

Работа над диссертацией была поддержана грантом РФФИ № 14-01-00271-а.

Научная новизна. Все основные результаты диссертации являются новыми.

Публикации. По теме диссертационной работы опубликованы работы [13], [14] и [15] в научных изданиях, включенных в Перечень рецензируемых научных изданий, рекомендованных ВАК. В статье [13] Ермаковым С.М. была поставлена задача и предложен метод её решения, а реализация метода, получение оценок метода Монте-Карло, исследование их свойств и проведение численных экспериментов полностью выполнено диссертантом. В статье [14] соискателем были доказаны лемма 1 об оценке погрешности при использовании асинхронных итераций и теорема 1 о сходимости метода частичной синхронизации, предложены оценки метода Монте- Карло в случае частичной синхронизации, сформулированы и доказаны теоремы 3 и 4 о достаточных условиях стохастической устойчивости предложенных методов. В статье [15] соискателем был построен пример расходимости асинхронных итераций для случая нелинейной системы, были сформулированы и доказаны лемма 2 об оценке погрешности при использовании асинхронных итераций для нелинейных систем уравнений и теорема 6 о сходимости метода частичной синхронизации в случае нелинейных систем уравнений.

Личный вклад автора. Содержание диссертации и основные положения, выносимые на защиту, отражают персональный вклад автора в опубликованные работы. Подготовка к публикации полученных результатов прово-

дилась совместно с соавторами, причем вклад диссертанта был определяющим.

Структура и объем диссертации. Диссертация состоит из введения, 3 глав, заключения и библиографии. Общий объем диссертации 125 страниц, из них 120 страницы текста, включая 20 рисунков. Библиография включает 52 наименования на 5 страницах.

Глава 1

Метод Монте-Карло и асинхронные итерации

При использовании параллельных или распределенных алгоритмов необходимо координировать действия различных процессоров, другими словами, необходимо наличие некоторого управляющего алгоритма. Принцип действия управляющих алгоритмов существенно различается для синхронных и асинхронных алгоритмов. Для синхронных алгоритмов процесс управления удобно представить в виде этапов, в течении которых каждый процессор должен совершить ряд вычислений на основе данных, полученных от других процессоров на предыдущих этапах. Время выполнения вычислений на каждом процессоре в рамках одного этапа в общем случае не зависит от времени аналогичных вычислений на других процессорах. Этап заканчивается в момент завершения необходимых вычислений на каждом процессоре, после чего процессоры обмениваются информацией, и выполняется переход к следующему этапу. Если бы у процессоров имелся доступ к глобальному времени, то для всех процессоров все этапы начинались и заканчивались одновременно. При отсутствии в вычислительной системе глобального времени необходимо прибегать к помощи синхронизирующих алгоритмов, работа которых основана на глобальной синхронизации или локальной синхронизации.

При использовании глобальной синхронизации каждый процессор переходит к следующему этапу вычислений только после того, как все процессоры закончат вычисления, и все отправленные сообщения будут получены адресатами. Возможные методы реализации такого подхода можно найти, например, в [16–18].

При локальной синхронизации процессор переходит к следующему этапу после того, как закончит вычисления и получит те сообщения с данными,

которые необходимы ему для перехода к следующему этапу. В этом случае не тратится время на ожидание того, когда все сообщения будут доставлены всем процессорам.

Для асинхронных алгоритмов нет такого разделения на этапы, после выполнения вычислений процессоры приступают к новым, не следя за доставкой сообщений, и используют данные от других процессоров, имеющиеся на данный момент, пусть даже они будут и не самыми актуальными. При таком подходе нет необходимости использовать глобальное время, глобальную или локальную синхронизации и время простаивания процессоров сводится к минимуму. Важные вопросы, на которые требуется ответить при использовании асинхронных методов – приведет ли такой подход к нахождению решения задачи и, если да, то будет ли выигрыш в общем времени решения задачи по сравнению с синхронным аналогом алгоритма.

Рассмотрим вычислительную систему с $n \in \mathbb{N}$ процессорами и задачу нахождения неподвижной точки, в которой ищется вектор $x = (x_1, x_2, \ldots, x_n)$, удовлетворяющий равенству

$$x_i = f_i(x_1, x_2, \ldots, x_n),\ i = 1, \ldots, n,$$

где f_i – заданные функции зависящие от n переменных. Естественно в данной ситуации распределить вычисления по процессорам таким образом, чтобы i-ый процессор обновлял переменную x_i согласно формуле

$$x_i := f_i(x_1, x_2, \ldots, x_n),\ i = 1, \ldots, n,$$

предполагая что вычисления начинаются с некоторых исходных значений.

При использовании синхронного алгоритма процессор i не приступит к k-ому обновлению пока не получит результаты $(k-1)$-ого обновления от процессоров, чьи переменные используются при расчете f_i. У такого подхода есть несколько недостатков. Во-первых, процессор i после обновления x_i

вынужден ожидать прихода результатов вычислений от других процессоров (рисунок 1.1). В частности, медленный канал обмена данными замедляет решения всей задачи (рисунок 1.2 a). Во-вторых, те процессоры, которые выполняют свои вычисления быстрее других по причине меньшей нагрузки в рамках одной итерации или же в силу большей вычислительной мощности, вынуждены ждать пока более медленные процессоры завершат вычисления. Поэтому скорость всего алгоритма будет напрямую зависеть от скорости самого медленного процессора (рисунок 1.2 b). Простой процессоров относится к так называемому штрафу за синхронизацию.

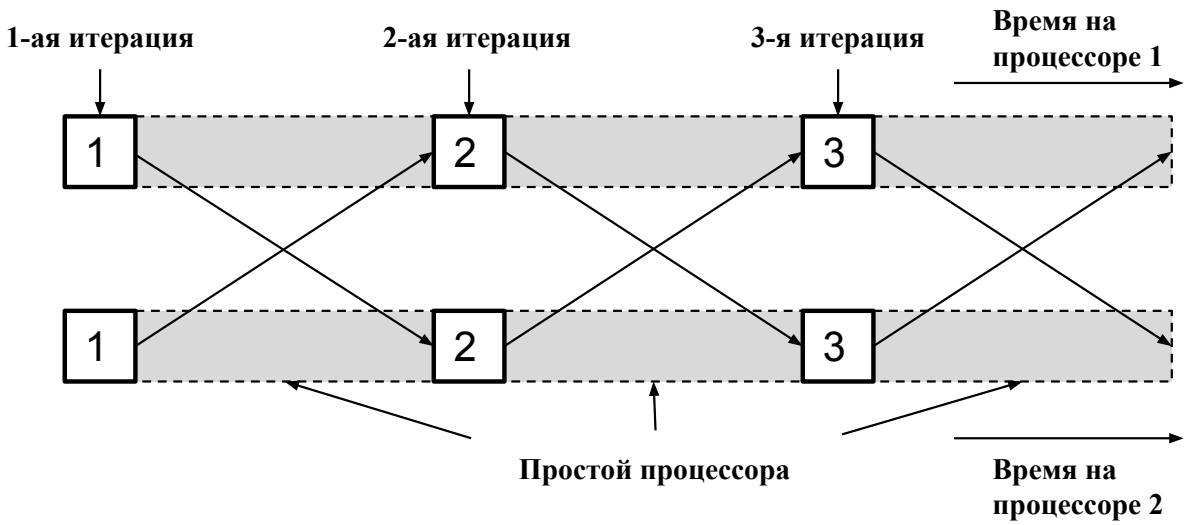

Рис. 1.1. Синхронный метод

Использование асинхронного алгоритма (рисунок 1.3) позволяет в значительной степени ослабить условия перехода для процессора i от k-ого к $(k+1)$-ому обновлению. На рисунке 1.3 представлена ситуация, когда за время обмена данными между процессорами каждый из них может успеть выполнить три итерации. Чтобы перейти к $(k+1)$-ому обновлению i-ому процессору достаточно знать некоторые прошлые результаты обновлений других процес-

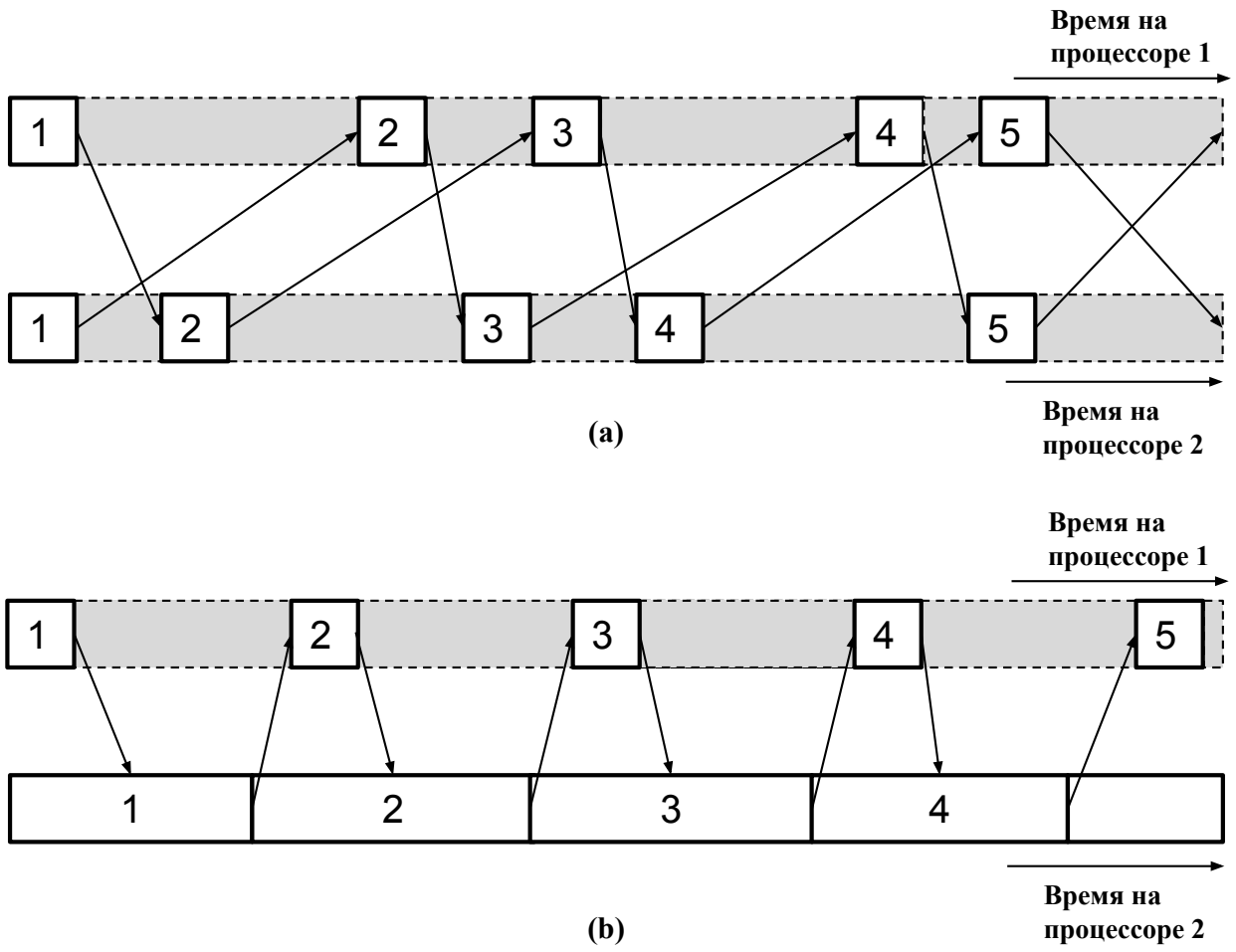

Рис. 1.2. Синхронный метод. Задержки.

соров пусть даже не самые актуальные. При таком подходе удаётся свести к минимуму штраф за синхронизацию, правда есть опасность, что использование в вычислениях устаревшей информации приведет к неэффективному алгоритму. Обсуждение этого вопроса будет приведено в последующих разделах.

Рассмотрим ещё один важный пример (см. [19]), демонстрирующий преимущество асинхронных алгоритмов. Как отмечалось выше, время вычислений в рамках одной итерации может отличаться от процессора к процессору. Представляется разумным предположить, что время вычислений на процес-

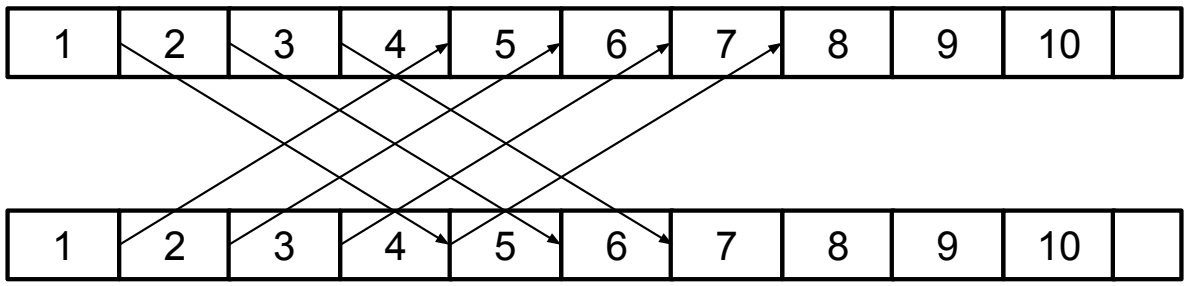

Рис. 1.3. Асинхронный метод

соре является случайной величиной. Для примера предположим, что время вычислений на любом из процессоров распределено по показательному распределению с параметром $\lambda > 0$. При синхронной реализации алгоритма переход к следующему этапу происходит после завершения вычислений всеми процессорами, то есть время этапа – это максимум из n случайных величин, если в системе n процессоров. Среднее время этапа, как нетрудно проверить, имеет выражение H_n/λ, где

$$H_n = 1 + \frac{1}{2} + \cdots + \frac{1}{n}.$$

В данном случае H_n характеризует штраф за синхронизацию. Среднее время простоя процессора равно $(H_n - 1)/\lambda$ и будет неограниченно расти при стремлении n к бесконечности. Такой итог побуждает к исследованию асинхронных алгоритмов на неоднородных системах, то есть тех системах, на которых вычислительное время или время передачи может существенно отличаться для разных процессоров.

Когда речь заходит о решении задач большой размерности, нередко возникающих при дискретизации задач математической физики, то наряду с детерминированными методами решения рассматривают и стохастические, а

именно метод Монте-Карло. Чаще всего в таких ситуациях при его использовании строится стохастическая оценка, математическое ожидание которой есть искомое решение. Далее производится моделирование большого числа случайных величин, распределенных так же, как построенная оценка. После получения достаточного количества реализаций случайной величины берется среднее смоделированных величин, что в результате и является приближением искомой величины. Привлекательность такого метода состоит в том, что моделирование делается независимо и может быть поручена разным процессорам, представляя тем самым эффективный алгоритм загрузки многопроцессорной системы произвольной архитектуры.

Далее будет рассматриваться задача нахождения неподвижной точки

$$x = F(x), \tag{1.1}$$

где $x = (x_1, x_2, \ldots, x_n)^T$ – вектор-столбец неизвестных, $F = (f_1(x), f_2(x), \ldots, f_n(x))^T$ – оператор из $\mathbb{R}^n$ в $\mathbb{R}^n$. Так как в дальнейшем будет рассматриваться последовательность векторов наряду с элементами этих векторов, введем следующие обозначения: компоненты вектора x из $\mathbb{R}^n$ будем обозначать x_i, $i = 1, \ldots, n$, а последовательность векторов из $\mathbb{R}^n$ будем обозначать $x(j)$, $j = 0, 1, \ldots$. Метод простых итераций, который лежит в основе рассматриваемых методов, запишется в предложенных обозначениях в виде

$$x(k+1) = F(x(k)),\ k = 0, 1, \ldots, \tag{1.2}$$

для некоторого начального $x(0)$.

Условия, при которых процесс (1.2) сходится к неподвижной точке оператора F, можно найти, например, в [20]. Среди всевозможных условий особо выделим следующий класс операторов и связанную с этим классом теорему.

Определение 1. *Отображение $F : D \subset \mathbb{R}^n \to \mathbb{R}^n$ называется сжимающим*

на $D_0 \subset D$, если существует такое $\alpha < 1$, что $\|F(x) - F(y)\| < \alpha\|x - y\|$ при всех $x, y \in D_0$.

Теорема 1. *Пусть отображение $F : D \subset \mathbb{R}^n \to \mathbb{R}^n$ сжимающее на замкнутом множестве $D_0 \subset D$ и $F(D_0) \subset D_0$. Тогда F имеет единственную неподвижную точку $x^* \in D_0$ и для любого начального $x(0) \in D_0$ последовательность $\{x(k)\}$, определяемая (1.2), сходится к x^*.*

1.1. Асинхронные итерации

Впервые асинхронные варианты методов итераций для решения системы (1.1), в которой оператор F являлся линейным оператором вида $Ax+b$, были предложены в [1] и носили название "схема хаотических релаксаций"(chaotic relaxation scheme). Там же было показано, что предложенная схема сходится к решению задачи (1.1) тогда и только тогда, когда $\lambda_1(|A|) < 1$, где $|A|$ – матрица, составленная из модулей элементов матрицы A, а $\lambda_1(|A|)$ – первое собственное число матрицы $|A|$.

Затем в [21] и [22] Миллоу обобщил схему хаотических релаксация на случай нелинейного оператора F и доказал сходимость обобщенного метода для сжимающих операторов. В [2] Боде ввел определение асинхронных итераций, которые обобщали понятие хаотических релаксаций, для решения систем (в том числе и нелинейных) уравнений и получил достаточные условия сходимости метода к решению системы (1.1).

Ещё более общее определение асинхронных итераций было приведено в [16], поэтому в дальнейшем будем следовать постановке из этого источника.

Пусть $X_1, X_2, \dots X_n$ – заданные множества, а X – их декартово произведение

$$X = X_1 \times X_2 \times \cdots \times X_n.$$

Элемент $x \in X$ соответственно имеет структуру

$$x = (x_1, x_2, \ldots, x_n),$$

где x_i принадлежат X_i, $i = 1, \ldots, n$. Пусть заданы функции $f_i : X \to X_i$, а функция $F : X \to X$ представляется в виде

$$F(x) = (f_1(x), f_2(x), \ldots, f_n(x)), \forall x \in X.$$

Задача состоит в нахождении неподвижной точки оператора F, то есть такой $x^* \in X$, что $x^* = F(x^*)$ или в покомпонентной записи

$$x_i^* = f_i(x^*), \; i = 1, \ldots, n.$$

Далее определим асинхронную версию метода простых итераций, которую впоследствии будем называть асинхронными итерациями.

Предположим, что множество $T = \{0, 1, 2, \ldots\}$ – это множество моментов времени, в которые одна или несколько компонент x_i вектора x обновляется некоторым процессором распределенной вычислительной системы. Обозначим через T^i множество моментов времени, в которые происходит обновление x_i.

Разумно предположить, что в распределенной системе процессор, обновляющий компоненту x_i, не всегда имеет актуальную информацию по другим компонентам вектора x. Поэтому в асинхронном случае допускается использование устаревшей информации. Этот факт можно записать в следующем виде

$$x_i(t+1) = f_i(x_1(\tau_1^i(t)), \ldots, x_n(\tau_n^i(t))), \forall t \in T^i, \tag{1.3}$$

где $\tau_i^j(t)$ – моменты времени, удовлетворяющие неравенству

$$0 \leq \tau_j^i(t) \leq t, \forall t \in T.$$

Для всех моментов $t \notin T^i$ считаем, что x_i не обновляется

$$x_i(t+1) = x_i(t), \forall t \notin T^i. \tag{1.4}$$

Элементы множества T следует рассматривать как индексы последовательности моментов реального времени, в которые происходит обновление. Процессорам, которые не обновляют компоненту x_i не обязательно знать множество T^i, так как этого не нужно для расчета итераций (1.3) и (1.4), и поэтому нет необходимости иметь в системе глобальное время. Разницу $(t - \tau_j^i(t))$ между текущим временем t и временем $\tau_i^j(t)$, когда процессором, обновляющем x_i, в последний раз была получена информация о компоненте x_j, может быть рассмотрена как задержка в передаче информации. Удобно рассматривать вычислительный процесс в данной ситуаций следующим образом: в момент $t \in T^i$ процессор, закончивший предшествующие расчеты и готовый выполнить новые, получает посредством некоторого механизма величины $x_1(\tau_1^i(t)), \ldots, x_n(\tau_n^i(t))$, а затем обновляет x_i по формуле (1.3), при этом ему совершенно не обязательно знать значения $t, \tau_1^i(t), \ldots, \tau_n^i(t)$ и элементов T^j, $j = 1, \ldots, n$.

Заметим, что такие итеративные методы для решения систем линейных уравнений, как метод Якоби и метод Гаусса-Зейделя, являются частными случаями итерации (1.3). Для метода Якоби множества T^i и моменты времени $\tau_j^i(t)$ определяются как

$$T^i = T,\ \tau_j^i(t) = t,\ \forall i, j \in \{1, \ldots, n\},\ \forall t \in T.$$

Для метода Гаусса-Зейделя множества T^i и моменты времени $\tau_j^i(t)$ определяются следующим образом

$$T^i = \{t \in T \mid (t+1)\, mod\, i = 0\},$$

$$\begin{cases} \tau_j^i(t) & = t - i + j, \text{ для } j < i, \\ \tau_j^i(t) & = t - i + j - n, \text{ для } j \geq i. \end{cases}$$

Чтобы называть итерации (1.3) - (1.4) асинхронными необходимо добавить определенные условия на множества T^i и моменты времени $\tau_j^i(t)$.

Множества T^i являются бесконечными и для любой последовательности $\{t_k\} \subseteq T^i$, стремящейся к бесконечности, выполнено $\lim_{k\to\infty} \tau_j^i(t_k) = \infty$ для $j = 1, \ldots, n$.

Данное предположение гарантирует, что каждая компонента обновится бесконечное число раз, а старая информация в конечном итоге выйдет из обработки. В дальнейшем будем считать, что это предположение выполнено.

После введения итераций (1.3) - (1.4) и сделанных относительно них предположений возникает вопрос – при каких условиях итерации сходятся к неподвижной точке оператора F? Достаточные условия (см. [16]) даёт следующая

Теорема 2. *Если существует последовательность непустых множеств $\{X(k)\}$, удовлетворяющих условиям*

- $\cdots \subset X(k+1) \subset X(k) \subset \cdots \subset X(0) \subset X$;
- $F(x) \in X(k+1)$ *для любого* k *и* $\forall x \in X(k)$. *Более того, если последовательность* $\{y(k)\}$ *такая, что* $y(k) \in X(k)$ *для* $k = 0, 1, \ldots$, *тогда каждая предельная точка* $\{y(k)\}$ *является неподвижной точкой оператора* F;
- *Для любого* $k \in \{0, 1, \ldots\}$ *существуют множества* $X_i(k) \in X_i$, *такие что*

$$X(k) = X_1(k) \times X_2(k) \times \cdots \times X_n(k),$$

- *начальное приближение* $x(0)$ *принадлежит* $X(0)$,

тогда каждая предельная точка последовательности $\{x(t)\}$, определяемой асинхронными итерациями, является неподвижной точкой оператора F.

Заметим, что первое и второе условия теоремы вместе подразумевают, что синхронные итераций $x := F(x)$, начинающаяся с некоторого начального x из $X(0)$, сходятся к неподвижной точке оператора F. Третье условие означает, что если взять два произвольных элемента $X(k)$ и поменять в них i-ые компоненты местами, то снова получатся элементы множества $X(k)$.

Далее ограничимся рассмотрением операторов вида $F : \mathbb{R}^n \to \mathbb{R}^n$. Рассмотрим следующую норму на $\mathbb{R}^n$

$$\|x\|_\omega = max_i \frac{|x_i|}{\omega_i},$$

где $\omega = (\omega_1, \omega_2, \ldots, \omega_n) \in \mathbb{R}^n$ и $w_i > 0, i = 1, \ldots, n$.

Если теперь рассматривать сжимающие отображения с параметром сжатия $\alpha < 1$ и положить в определении асинхронных итераций $X_i = \mathbb{R}$, $i = 1, \ldots, n$, а также $X = \mathbb{R}^n$, то согласно теореме 2, чтобы показать, что асинхронные итерации сходятся к неподвижной точке x^* оператора F, нужно построить последовательность множеств $\{X(k)\}$. Определим их следующим образом

$$X(k) = \{x \in \mathbb{R}^n \mid \|x - x^*\|_\omega \leq \alpha^k \|x(0) - x^*\|_\omega\}.$$

Нетрудно проверить выполнение условий теоремы.

Далее будут использовать следующие нормы:

- для вектора $x = (x_1, \ldots, x_n)^T$:

$$\|x\| = \max_{1 \leq i \leq n} |x_i|, \tag{1.5}$$

- для матрицы $A = \|a_{ij}\|_{i,j=1}^n$:

$$\|A\| = \max_{1 \leq i \leq n} \sum_{j=1}^{n} |a_{ij}|. \tag{1.6}$$

1.1.1. Система линейных уравнения

Рассмотрим случай, когда

$$F(x) = Ax + b,$$

где $A = \|a_{ij}\|_{i,j=1}^{n}$ – заданная матрица, $b \in \mathbb{R}^n$ – заданный вектор правых частей. В этом случае выполняется поиск такого x^*, что

$$x^* = Ax^* + b.$$

Асинхронные итерации (1.3)-(1.4) для системы линейных уравнений будут иметь вид

$$\begin{aligned} x_i(t+1) &= \sum_{j=1}^{n} a_{i,j} x_j(\tau_j^i(t)) + b_i,\ t \in T^i, \\ x_i(t+1) &= x_i(t),\ t \notin T^i. \end{aligned} \tag{1.7}$$

В работе [1] было доказано, что хаотические релаксации, которые являются частным случаем асинхронных итераций, для системы линейных уравнений сходятся к решению системы тогда и только тогда, когда $\lambda_1(|A|) < 1$. В [16] было получено обобщение этого результата для случая асинхронных итераций.

Теорема 3. *Пусть матрица A такая, что $I - A$ обратима. Тогда следующие утверждения эквивалентны*

1. $\lambda_1(|A|) < 1$;
2. *Для любого начального $x(0)$, для любого $b \in \mathbb{R}^n$, для любых множеств T^i, удовлетворяющих условиям из определения асинхронных итераций, для любого выбора переменных $\tau_j^i(t)$ таких, что $t - 2 < \tau_j^i(t) < t$, последовательность, порождаемая асинхронными итерациями (1.7), сходится к $(I - A)^{-1}b$.*

Таким образом, если обычный (синхронный) процесс сходится – $|\lambda_1(A)| < 1$, но $\lambda_1(|A|) > 1$, то по крайней мере асинхронные итерации некоторого вида обязательно расходятся. В этом случае можно попытаться исправить положение, осуществляя после некоторой группы асинхронных итераций определенное количество синхронных, которые уменьшают ошибку (частичная синхронизация).

Будем далее рассматривать алгоритм, который после каждых m асинхронных итераций, использует l простых. Очевидно, существует такое m, при котором этот комбинированный итерационный процесс будет сходиться, но медленнее, вообще говоря, чем процесс, полностью синхронизированный. Поэтому наш подход имеет смысл, если асинхронные итерации существенно дешевле, чем синхронные.

Оценка возможной получаемой выгоды существенно зависит от вида матрицы A и в общем случае может быть достаточно грубой. По-видимому, наиболее эффективным здесь может быть численный эксперимент. Тем не менее мы докажем лемму, которая указывает границы роста ошибки в асинхронном случае при $\lambda_1(|A|) > 1$.

Пусть $x(t), t = 0, 1, 2, \ldots$ – последовательность асинхронных итераций для системы

$$x = Ax + b,$$

а $\widetilde{x}$ – её решение. Тогда $x(t)$ можно представить в виде $x(t) = \widetilde{x} + \Delta x(t)$, где $\Delta x(t)$ – последовательность асинхронных итераций для системы

$$x = Ax.$$

Лемма 1. *Если для матрицы A выполнено $|\lambda_1(A)| < 1$ и $\lambda_1(|A|) > 1$, то*

$$\|\Delta x(k)\| \le \|A\|^k \|\Delta x(0)\| \tag{1.8}$$

для $k = 0, 1, 2, \ldots$.

Доказательство. Проведем доказательство по индукции. Для $k = 0$ имеем

$$\|\Delta x(0)\| = \|A\|^0 \|\Delta x(0)\|,$$

и, следовательно, база индукции доказана. Пусть теперь (1.8) выполнено для всех $k \leq m$. Покажем, что (1.8) выполняется при $k = m + 1$. Согласно определению асинхронных итераций, если $m \in T^i$, то

$$\Delta x_i(m+1) = \sum_{j=1}^{n} a_{i,j} \Delta x_j(\tau_j^i(m)),$$

а если $m \notin T^i$, то

$$\Delta x_i(m+1) = \Delta x_i(m).$$

Пусть $\Delta\hat{x}$ – вектор длины $2n$, такой что $\Delta\hat{x}_i = \Delta x_i(m+1)$, $\Delta\hat{x}_{n+i} = 0$ при $m \in T^i$ и $\Delta\hat{x}_i = 0$, $\Delta\hat{x}_{n+i} = \Delta x_i(m+1)$ при $m \notin T^i$. Обозначим вектор $(\Delta x_1(\tau_1^i(m)), \ldots, \Delta x_n(\tau_n^i(m)))^T$ за $\widetilde{\Delta x}$. Для $\Delta\hat{x}$, $\widetilde{\Delta x}$ и $\Delta x(m)$ справедливо равенство

$$\Delta\hat{x} = \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix} \begin{pmatrix} \widetilde{\Delta x} \\ \Delta x(m) \end{pmatrix},$$

где A_1, A_2 – матрицы n на n. При $m \in T_j$ j-ая строка матрицы A_1 равняется j-ой строке матрицы A, а при $m \notin T_j$ j-ая строка матрицы A_1 состоит из нулей. A_2 – диагональная матрица, для которой j-й элемент диагонали равен 0, если $m \in T_j$, и равен 1 в противном случае.

Заметим, что $\|\Delta\hat{x}\| = \|\Delta x(m+1)\|$, $\|A_1\| \leq \|A\|$, и в силу того, что $\|A\| \geq |\lambda_1(|A|)| > 1$, выполнено неравенство $\|A_2\| \leq \|A\|$. Тогда справедливо неравенство

$$\|\Delta x(m+1)\| = \left\| \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix} \begin{pmatrix} \widetilde{\Delta x} \\ \Delta x(m) \end{pmatrix} \right\| \leq \|A\| \left\| \begin{pmatrix} \widetilde{\Delta x} \\ \Delta x(m) \end{pmatrix} \right\|.$$

Пусть $\|(\widetilde{\Delta x}^T, \Delta x(m)^T)\| = |\Delta x_{j'}(s')|$ для некоторого натурального $s' \leq m$ и

$1 \le j' \le n$. Так как $s' < m$, в силу предположения индукции будет выполнено

$$\left\| \begin{pmatrix} \widetilde{\Delta x} \\ \Delta x(m) \end{pmatrix} \right\| \le \|\Delta x(s')\| \le \|A\|^{s'} \|\Delta x(0)\| \le \|A\|^{m} \|\Delta x(0)\|,$$

а следовательно,

$$\|\Delta x(m+1)\| \le \|A\|^{m+1} \|\Delta x(0)\|,$$

и лемма доказана. □

Теперь легко доказать следующую теорему.

Теорема 4. *Если итерационный процесс состоит из последовательности групп m асинхронных итераций и затем l синхронных, то при достаточно большом l он сходится. При этом сходится не медленнее чем $\lambda_\varepsilon \left(\frac{\|A\|}{\lambda_\varepsilon}\right)^{\frac{m}{m+l}}$ для произвольного λ_ε, удовлетворяющего неравенству $|\lambda_1(A)| < \lambda_\varepsilon < 1$.*

Доказательство. Из леммы 1 следует, что за m асинхронных итераций ошибка может возрасти не более чем в $\|A\|^m$ раз. Поскольку $|\lambda_1(A)| < 1$, то для $\forall \varepsilon > 0$, такого что $\varepsilon < 1 - |\lambda_1(A)|$, найдется такое l_0, что для $\forall l \ge l_0$ будет выполняться неравенство $\|A^l\| < (|\lambda_1(A)| + \varepsilon)^l < 1$. Обозначим $(|\lambda_1(A)| + \varepsilon)$ за λ_ε. Нетрудно видеть, что $|\lambda_1(A)| < \lambda_\varepsilon < 1$. Норму ошибки после $m+l$ итераций (m асинхронных и l синхронных) можно оценить следующим образом:

$$\|\Delta x(m+l)\| \le \|A^l\| \|A\|^m \|\Delta x(0)\| < \lambda_\varepsilon^l \|A\|^m \|\Delta x(0)\|.$$

При достаточно большом l имеем $\lambda_\varepsilon^l \|A\|^m < 1$, что и доказывает первую часть теоремы.

Мы видим, что за $m+l$ итераций ошибка уменьшается в $\lambda_\varepsilon^l \|A\|^m$ раз. Такой результат мы имели бы при геометрической сходимости с параметром r, если бы $r^{m+l} = \lambda_\varepsilon^l \|A\|^m$. То есть

$$r = (\lambda_\varepsilon^l \|A\|^m)^{\frac{1}{m+l}} = \lambda_\varepsilon \left(\frac{\|A\|}{\lambda_\varepsilon}\right)^{\frac{m}{m+l}},$$

что доказывает вторую часть теоремы. □

Теорема 4 даёт соотношение на количество асинхронных и синхронных итераций для систем линейных уравнений вида (1). В алгоритмах, где чередуются m асинхронных и l синхронных итераций, и при выбранном соотношении m и l итерации сходятся к искомому решению, будем называть сумму $m + l$ **периодом асинхронности**. Перейдем теперь к рассмотрению более общего случая.

1.1.2. Система нелинейных уравнений

Определение 2. *Оператор F из $\mathbb{R}^n$ в $\mathbb{R}^n$ называется липшицевым (в литературе (см. [20]) также встречается название n-липшицевый) оператором на $D \subseteq \mathbb{R}^n$, если существует неотрицательная матрица A такая, что*

$$|F(x) - F(y)| \leq A|x - y|, \forall x, y \in D, \tag{1.9}$$

где операция взятия модуля применяется покомпонентно и неравенство выполняется для всех компонент.

Матрицу A из определения 2 будем называть липшицевой матрицей оператора F.

Определение 3. *Оператор F из $\mathbb{R}^n$ в $\mathbb{R}^n$ называется сжимающим (в литературе (см. [20]) также встречается название n-сжимающий) оператором на $D \subseteq \mathbb{R}^n$, если он липшицев на D и для его липшицевой матрицы A выполнено $\lambda_1(A) < 1$, где $\lambda_1(A)$ — первое собственное число матрицы A.*

Для сжимающих операторов справедлива (см. [2]) следующая

Теорема 5. *Если F – n-сжимающий оператор на замкнутом множестве $D \subseteq \mathbb{R}^n$ и $F(D) \subseteq D$, тогда произвольные асинхронные итерации сходятся к единственному решению системы (1.1).*

Таким образом, если обычный (синхронный) процесс $x_{k+1} = F(x_k)$, $k = 0, 1, \ldots$ сходится, а оператор F не удовлетворяет условиям теоремы 5, то, по крайней мере, асинхронные итерации некоторого вида обязательно расходятся. Можно попытаться исправить положение, осуществляя после некоторой группы асинхронных итераций определенное количество синхронных, которые уменьшают ошибку (частичная синхронизация).

В статье [14] был приведен пример линейной системы вида $x = Ax + b$, при решении которой методом асинхронных итераций в некоторых случаях можно наблюдать расходимость метода. Для получения аналогичного результата для нелинейной системы (1.1) достаточно рассмотреть оператор F следующего вида:

$$F(x) = 10^{-2}x^2 + Ax + b, \tag{1.10}$$

где операция возведения в квадрат поэлементная.

Рассмотрим теперь задачу нахождения корня некоторого оператора $G : \mathbb{R}^n \to \mathbb{R}^n$:

$$G(x) = 0.$$

Будем искать решение методом хорд

$$x(k+1) = x(k) - CG(x(k)),$$

где C – заданная обратимая матрица, а $x(0)$ – заданный начальный вектор. Нетрудно видеть, что метод хорд – это метод простых итераций для оператора $I - CG$, и для нахождения неподвижной точки этого оператора могут использоваться асинхронные итерации. Такой оператор G, при котором простые итерации для оператора $I - CG$ сходятся, а асинхронные итерации в некоторых случаях расходятся, может быть найден из выражения

$$x - CG(x) = F(x),$$

где оператор F из (1.10). Таким образом искомый оператор выглядит следующим образом

$$G(x) = C^{-1}(x - F(x)).$$

Расходимость асинхронных итераций продемонстрирована на рисунке 1.4.

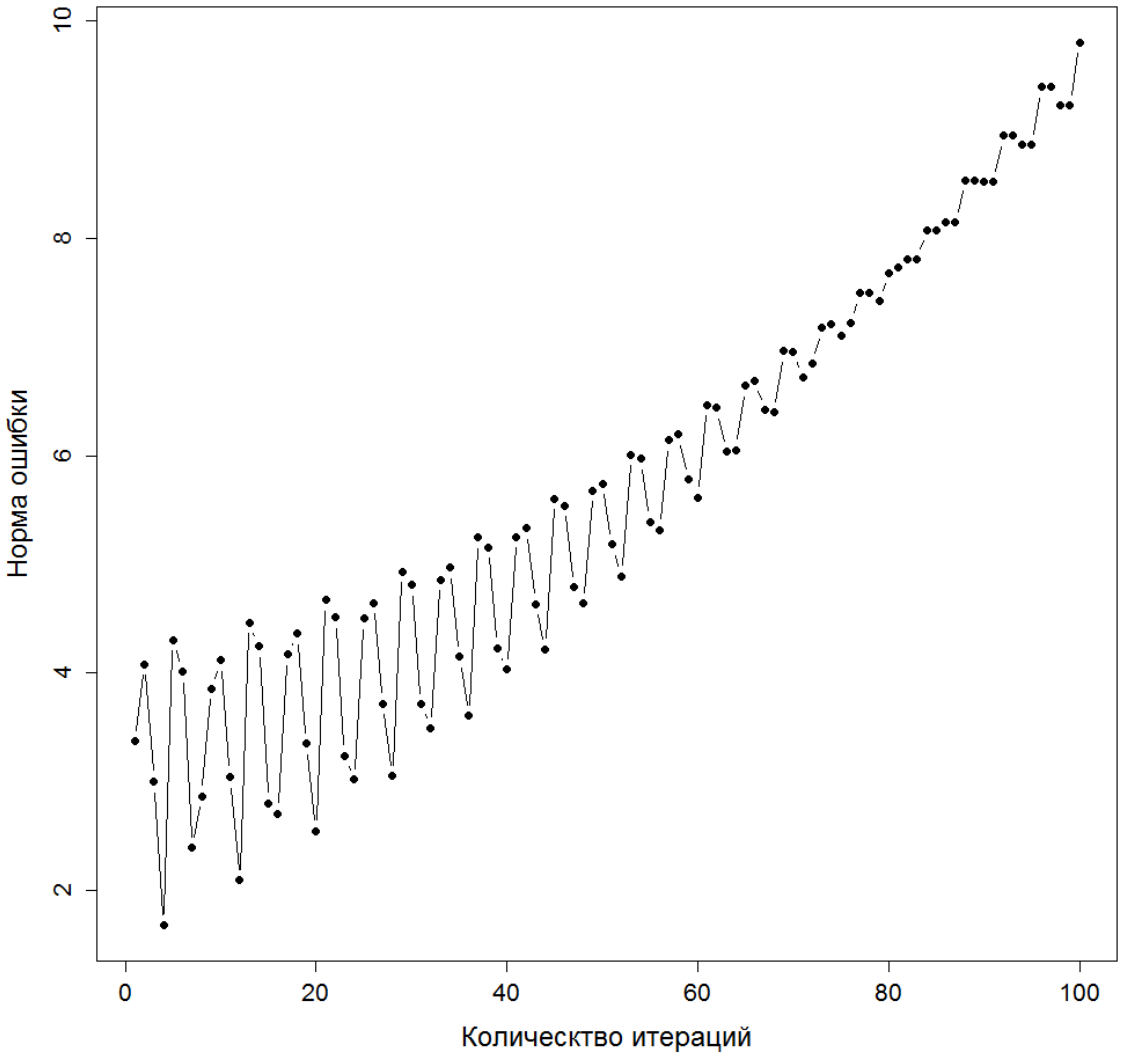

Рис. 1.4. Расходимость асинхронных итераций

Если же корень оператора G ищется при помощи метода Ньютона

$$x_{k+1} = x_k - G'(x_k)^{-1}G(x_k),$$

то на каждой итерации необходимо решать систему линейных уравнений $G'(x_k)y = G(x_k)$ или эквивалентную ей систему $y = (I - G'(x_k))y + G(x_k)$. Применение метода асинхронных итераций для решения последней системы при некоторых условиях может привести к расходимости. Так, например, если взять $G(x) = 0.5x^2 + Ax + b$, то при $x_0 = (1, 1, \ldots, 1)$ необходимо будет решить систему $y = Ay + G(x_0)$, при решении которой методом асинхронной итерации можно наблюдать расходимость.

Приведенные выше примеры свидетельствуют о том, что существуют такие нелинейные операторы для которых простые (синхронных) итерации сходятся к их неподвижной точке, в то время как асинхронные итерации могут и расходится. Возможным компромиссным решением как и в линейном случае может быть использование последовательности групп m асинхронных итераций и затем l синхронных. Число итераций m и l, при которых будет наблюдаться сходимость зависит в общем случае от свойств оператора F.

Будем далее рассматривать алгоритм, который после каждых m асинхронных итераций, использует l обычных. Очевидно, существует такое m, при котором этот комбинированный итерационный процесс будет сходиться, но медленнее, вообще говоря, чем процесс, полностью синхронизированный. Поэтому наш подход имеет смысл, если асинхронные итерации существенно дешевле, чем синхронные.

Пусть $x(t), t = 0, 1, 2, \ldots$ – последовательность асинхронных итераций для системы (1.1), а $\widetilde{x}$ – её решение. Пусть F – липшицевый оператор на $D \in \mathbb{R}^n$ с липшицевой матрицей A, но при этом F не является сжимающим, то есть $\lambda_1(|A|) > 1$. Рассматривая оператор $F(x+\widetilde{x})-\widetilde{x}$, не умаляя общности можно считать, что $\widetilde{x} = F(\widetilde{x}) = 0$. Полагая $y = \widetilde{x}$ в неравенстве (1.9), условие липшица для оператора F приобретет вид $|F(x)| \leq A|x|, \forall x \in D$.

Лемма 2. *При сделанных выше предположениях относительно оператора*

F выполняется неравенство

$$\|x(k)\| \leq \|A\|^k \|x(0)\| \tag{1.11}$$

для $k = 0, 1, 2, \ldots$.

Доказательство. Проведем доказательство по индукции. Для $k = 0$ имеем

$$\|x(0)\| = \|A\|^0 \|x(0)\|,$$

и, следовательно, база индукции доказана. Пусть теперь (1.11) выполнено для всех $k \leq m$. Покажем, что (1.11) выполняется при $k = m + 1$. Согласно определению асинхронных итераций, если $m \in T^i$, то

$$x_i(m+1) = f_i(x_1(\tau_1^i(m)), \ldots, x_n(\tau_n^i(m)))$$

а если $m \notin T^i$, то

$$x_i(m+1) = x_i(m).$$

Если $x_i(m+1) = x_i(m)$, то в силу предположения индукции выполнено

$$|x_i(m+1)| \leq \|x(m)\| \leq \|A\|^m \|x(0)\| \leq \|A\|^{m+1} \|x(0)\|.$$

Если $x_i(m+1) = f_i(x_1(\tau_1^i(m)), \ldots, x_n(\tau_n^i(m)))$, то в силу липшицевости оператора F выполнено

$$|x_i(m+1)| = |f_i(x_1(\tau_1^i(m)), \ldots, x_n(\tau_n^i(m)))| \leq \sum_{j=1}^{n} a_{ij} |x_i(\tau_j^i(m))| \leq$$

$$\leq \|A\| \|(x_1(\tau_1^i(m)), \ldots, x_n(\tau_n^i(m)))\|.$$

По определению асинхронных итераций $\tau_j^i(m) \leq m, j = 1, \ldots, n$, то по предположению индукции

$$|x_i(m+1)| \leq \|A\| \|(x_1(\tau_1(m)), \ldots, x_n(\tau_n(m)))\| \leq$$

$$\leq \|A\| \|A\|^m \|x(0)\| \leq \|A\|^{m+1} \|x(0)\|.$$

Следовательно, $|x_i(m+1)| \leq \|A\|^{m+1}\|x(0)\|$ при любом i, и лемма доказана. □

Теперь легко доказать следующую теорему для класса нелинейных операторов, описанных выше.

Теорема 6. *Если итерационный процесс состоит из последовательности групп m асинхронных итераций и затем l синхронных, то при достаточно большом l он сходится. При этом сходится не медленнее чем $r\left(\frac{\|A\|}{r}\right)^{\frac{m}{m+l}}$, где $r < 1$ – параметр геометрической сходимости итераций (1.2) к решению (1.1).*

Таким образом, для определенного класса вычислительных устройств метод частичной синхронизации и в нелинейном случае может служить удобным инструментом. Очевидно, другие методы линеаризации, метод Ньютона, например, допускает также частичную синхронизацию.

1.2. Алгоритмы метода Монте-Карло

Метод Монте-Карло зачастую без особого труда адаптируется к использованию на многопроцессорных вычислительных системах. Рассмотрим особенности его применения для решения систем уравнений, начав со случая линейных систем.

1.2.1. Система линейных уравнений

В этом разделе будут рассматриваться системы линейных алгебраических уравнений вида

$$x = Ax + b, \tag{1.12}$$

для которых сходится метод простых итераций.

Известны различные алгоритмы метода Монте-Карло для решения системы вида (1.12) (см., например, [4, 7, 23]). Мы будем рассматривать простейшие алгоритмы, состоящие в оценивании сходящегося ряда Неймана

$$(h, \widetilde{x}) = (h, \sum_{k=0}^{\infty} A^k b)$$

с помощью выбранной соответствующим образом оценки – оценки на траекториях моделируемой цепи Маркова. Здесь h – заданный вектор, а $\widetilde{x}$ – решение системы (1.12). Как известно (см., например, [4]), при

$$\lambda_1(|A|) < 1 \tag{1.13}$$

возможно вычисление $(h, \widetilde{x})$ при помощи асинхронного алгоритма. То есть на различных процессорах моделируются траектории цепи Маркова и вычисляются оценки на них, после чего оценки, полученные на всех процессорах, усредняются. Таким образом, условия несмещенности оценок метода Монте-Карло и сходимости асинхронных итераций совпадают, на что было обращено внимание в работе [24].

Как было показано в [4], нарушение условия (1.13) и использование асинхронного алгоритма приводит к стохастической неустойчивости – экспоненциальному росту дисперсии. Эта трудность, вообще говоря, может быть преодолена за счёт увеличения вычислительной работы, но её экспоненциальный рост делает алгоритм нереализуемым. Альтернативой является запоминание промежуточных результатов (синхронизация).

Предлагается в случае $\lambda_1(|A|) > 1$, "дешевых" асинхронных алгоритмов и относительно "дорогих" синхронных, как и в случае асинхронных итераций использовать смешанный алгоритм с частичной синхронизацией. Как и ранее, будем рассматривать подход, в котором чередуются асинхронные и синхронные алгоритмы. Для формального исследования такого комбинированного алгоритма мы подробно рассмотрим схемы оценивания частичных

сумм ряда Неймана в асинхронном и синхронном случаях.

1.2.2. Оценки частичных сумм ряда Неймана

Рассмотрим аналог схемы Неймана-Улама для нахождения частичной суммы ряда Неймана

$$S_m = \sum_{k=0}^{m} A^k b,$$

где $m \in \mathbb{N}$ задано. Случай $m = \infty$ подробно изучен в литературе [3]. Случай конечного m требует некоторой модификации стандартных рассуждений.

Пусть h – заданный вектор, $h = (h_1, \ldots, h_n)^T$. Алгоритм предполагает задание стохастической матрицы $\mathcal{P} = \|p_{ij}\|_{i,j=1}^{n}$, удовлетворяющей условиям согласования:

(а) $p_{i,j} > 0$, если $a_{i,j} \neq 0$, и распределение $q = (q_1, \ldots, q_n)$ такое, что $q_i > 0$, если $h_i \neq 0$, или

(б) $p_{i,j} > 0$, если $a_{j,i} \neq 0$, и распределение $p = (p_1, \ldots, p_n)$ такое, что $p_i > 0$, если $b_i \neq 0$.

Также зададим вектор $g = (g_1, \ldots, g_m)^T$, такой что $g_i > 0$, $i = 1, \ldots, m$ и

$$\sum_{i=1}^{m} g_i = 1.$$

Введем случайную величину τ, распределенную по закону

$$\begin{pmatrix} 1 & \ldots & m \\ g_1 & \ldots & g_m \end{pmatrix},$$

и рассмотрим случайные величины

$$\zeta = \frac{h_{i_0} a_{i_0,i_1} \ldots a_{i_{\tau-1},i_\tau} b_{i_\tau}}{q_{i_0} p_{i_0,i_1} \ldots p_{i_{\tau-1},i_\tau} g_\tau}, \tag{1.14}$$

$$\zeta^* = \frac{b_{j_0} a_{j_1,j_0} \ldots a_{j_\tau,j_{\tau-1}} h_{j_\tau}}{p_{j_0} p_{j_0,j_1} \ldots p_{j_{\tau-1},j_\tau} g_\tau}, \tag{1.15}$$

где $i_0 \to i_1 \to \cdots \to i_\tau$ соответствует цепи Маркова $(q, \mathcal{P})$, а $j_0 \to j_1 \to \cdots \to j_\tau$ – цепи Маркова $(p, \mathcal{P})$.

Легко проверить, что

$$\mathbb{E}\zeta = (h, \sum_{k=1}^{m} A^k b) = (h, S^m),$$

$$\mathbb{E}\zeta^* = (\sum_{k=1}^{m} b^T (A^T)^k, h) = (S_m^T, h).$$

Столь же просто вычисляются дисперсии оценок (1.14), (1.15), если учесть, что

$$\mathbb{E}\zeta^2 = \sum_{k=1}^{m} \sum_{i_0, i_1, \ldots i_k = 1}^{n} \frac{h_{i_0}^2 a_{i_0,i_1}^2 \ldots a_{i_{k-1},i_k}^2 b_{i_k}^2}{q_{i_0} p_{i_0,i_1} \cdots p_{i_{k-1},i_k} g_k} = \left(\frac{h^2}{q}, \sum_{k=1}^{m} \left(\frac{A^2}{\mathcal{P}} \right)^k \frac{b^2}{g} \right), \tag{1.16}$$

$$\mathbb{E}(\zeta^*)^2 = \sum_{k=1}^{m} \sum_{j_0, j_1, \ldots j_k = 1}^{n} \frac{b_{j_0}^2 a_{j_1,j_0}^2 \ldots a_{j_k,j_{k-1}}^2 h_{j_k}^2}{p_{j_0} p_{j_0,j_1} \cdots p_{j_{k-1},j_k} g_k} = \left(\sum_{k=1}^{m} \frac{(b^T)^2}{p} \left(\frac{(A^T)^2}{\mathcal{P}} \right)^k, \frac{h^2}{g} \right), \tag{1.17}$$

где операции возведение в квадрат векторов и матриц, как и деление матрицы на матрицу и вектора на вектор, выполняются поэлементно. В дальнейшем будет отдельно оговариваться использование подобных операций над матрицами и векторами. Оценкой всех компонент S_m может быть вектор $z = (\zeta_1, \ldots, \zeta_n)^T$, где ζ_i, $i = 1, \ldots, n$ получаются из (1.14), если полагать h равным вектору с i-ой компонентой равной единице и остальными равными нулю.

Как видно из (1.16) и (1.17) поведение дисперсии с ростом m определяется первыми собственными числами матриц $A^2/\mathcal{P}$ и $(A^T)^2/\mathcal{P}$ соответственно. Так, если выполнено $\lambda_1(A^2/\mathcal{P}) < 1$ и $\lambda_1((A^T)^2/\mathcal{P}) < 1$, то соответствующие дисперсии будут ограничены при стремлении m к бесконечности. В противном случае с ростом m будет наблюдаться экспоненциальный рост дисперсии.

Теперь для заданного вектора $y = (y_1, \ldots, y_n)^T$ введем оценки $\eta = \eta(y)$

и $\eta^* = \eta^*(y)$ такие, что

$$\mathbb{E}\eta = (h, A^{m+1}y),$$

$$\mathbb{E}\eta^* = (y^T(A^T)^{m+1}, h).$$

Оценки η и η^* строятся на траекториях цепей Маркова $(p, \mathcal{P})$ и $(q, \mathcal{P})$ длины $m+1$ и выражения для них имеют вид

$$\eta = \frac{h_{i_0} a_{i_0,i_1} \dots a_{i_m,i_{m+1}} y_{i_{m+1}}}{q_{i_0} p_{i_0,i_1} \dots p_{i_m,i_{m+1}}}, \tag{1.18}$$

$$\eta^* = \frac{y_{j_0} a_{j_1,j_0} \dots a_{j_{m+1},j_m} h_{j_{m+1}}}{p_{j_0} p_{j_0,j_1} \dots p_{j_m,j_{m+1}}}. \tag{1.19}$$

Как и ранее дисперсии оценок (1.18), (1.19) вычисляются просто, учитывая, что выражения для вторых моментов имеют вид

$$\mathbb{E}\eta^2 = \left(\frac{h^2}{q}, \left(\frac{A^2}{\mathcal{P}}\right)^{m+1} y^2\right),$$

$$\mathbb{E}(\eta^*)^2 = \left(\frac{(y^T)^2}{p} \left(\frac{(A^T)^2}{\mathcal{P}}\right)^{m+1}, h^2\right),$$

где операции возведение в квадрат векторов и матриц, как и деление матрицы на матрицу и вектора на вектор, выполняются поэлементно. Поведение дисперсии, также как и для оценок (1.14), (1.15), с ростом m определяется первыми собственными числами матриц $A^2/\mathcal{P}$ и $(A^T)^2/\mathcal{P}$.

Таким образом, при помощи пар оценок η, ζ и η^*, ζ^* для заданного вектора y и натурального m может быть получена оценка вектора $A^{m+1}y + \sum_{k=0}^{m} A^k b$. Поручая моделирование цепей Маркова и вычисление оценок по ним разным процессорам, мы вновь получим асинхронный алгоритм.

1.2.3. Оценки суммы ряда Неймана

Оценки (1.14), (1.18) и (1.15), (1.19) могут служить основой алгоритмов с частичной синхронизацией для получения решения системы (1.12) без огра-

ничений вида (1.13). Алгоритмы состоят в последовательном вычислении оценок векторов $y(j)$, где $y(j)$ связаны соотношением

$$y(0) = b,\ y(j) = A^m y(j-1) + \sum_{k=0}^{m-1} A^k b, \tag{1.20}$$

$$y(j+1) = Ay(j) + b, \ldots, y(j+l) = Ay(j+l-1) + b, j = 1, l+2, 2l+2, \ldots \tag{1.21}$$

или соотношением

$$y(0) = b,\ y(j) = A^{m+1} y(j-1) + \sum_{k=0}^{m} A^k b, j = 1, 2, \ldots \tag{1.22}$$

для некоторого фиксированного натурального m. Первый случай соответствует частичной синхронизации, когда после асинхронного алгоритма следуют синхронные, аналогично описанному выше подходу для асинхронных итераций. Если j устремить к бесконечности, то $y(j)$ в (1.20) – (1.22) будет стремиться к искомой сумме $\sum_{k=0}^{\infty} A^k b$.

При использовании метода Монте-Карло оценками векторов $y(j)$ будут случайные векторы $\Xi(j) = (\xi_1(j), \ldots, \xi_n(j))^T$. Скалярное произведение $(h, y(j))$ может оцениваться с помощью оценок вида (1.14), (1.18). В этом случае оценкой для $(h, y(j))$ будет

$$\eta(\Xi(j-1)) + \zeta + (h, b).$$

Или оценок вида (1.15), (1.19) и в этом случае $(h, y(j))$ будет оцениваться как

$$\eta^*(\Xi(j-1)) + \zeta^* + (h, b).$$

В каждом случае берется среднее из некоторого количества N реализаций оценок. Оценка всех компонент $y(j)$ может быть получена, если полагать h равным векторам с одной из компонент равной 1 и остальными равными 0.

При этом во многих случаях интерес представляет оценка $y(j)$ для $T \leq j < +\infty$ (разностные схемы).

Заметим, что один и тот же вектор (S_m в (1.22) и S_{m-1} в (1.20)–(1.21)) оценивается на каждой итерации, поэтому можно организовать алгоритм таким образом, чтобы оценка этого вектора уточнялась с каждой новой итерацией, используя оценки с предыдущих шагов итераций.

Особый интерес представляет задача оценки ковариации случайных векторов, возникающих при использовании метода Монте-Карло для оценки последовательности векторов итерационного процесса. Ковариация вектора $\Xi(j)$ с ростом j может оставаться ограниченной, а может экспоненциально расти. В последнем случае алгоритм будет стохастически неустойчивым, что делает невозможным его применение на практике.

Рассмотрим теперь итерационный процесс $y(j)$ с начальным вектором $y(0)$ и

$$y(j) = By(j-1) + d,\ j = 1, 2, \ldots, \tag{1.23}$$

где B – матрица $n \times n$, а d – вектор длины n. Пусть при каждом j вычисление слагаемых в правой части (1.23) происходит с помощью рандомизированной процедуры. Таким образом, вместо последовательности $y(j)$ возникает последовательность случайных векторов $\Xi(j) = (\xi_1(j), \ldots, \xi_n(j))^T$, $j = 0, 1, 2, \ldots$ с начальным вектором $\Xi(0)$, связанных соотношением

$$\Xi(j) = \mathbb{B}_j \Xi(j-1) + \mathbb{D}(j) \tag{1.24}$$

где $\mathbb{B}_j = \|\beta_{i,k}\|_{i,k=1}^n$ – случайные матричные операторы, $\mathbb{D}(j)$ – случайные векторы. При этом для любого натурального j выполнено $\mathbb{E}\Xi(j) = y(j) = (y_1(j), \ldots, y_n(j))^T$, $\mathbb{E}\mathbb{B}_j = B$, $\mathbb{E}\mathbb{D}(j) = d$.

Следующая лемма определяет характер поведения ковариации векторов ошибок $\mathcal{E}(j) = \Xi(j) - y(j)$.

Лемма 3. *Пусть случайные операторы $\mathbb{B}_j$, векторы $\mathbb{D}(j)$ и $\Xi(k)$ независимы в совокупности при любых $j = 0, 1, 2, \ldots$ и $k < j$, в том смысле, что случайные величины α_1, α_2, α_3, где α_1 – произвольный элемент оператора $\mathbb{B}_j$, α_2 –*

произвольный элемент $\mathbb{D}(j)$, α_3 – *произвольный элемент* $\Xi(k)$, *независимы в совокупности. Тогда для матрицы ковариации вектора* $\mathcal{E}(j)$ *справедливо соотношение*

$$\begin{aligned} vec\, cov\mathcal{E}(j) =& B \otimes B vec\, cov\mathcal{E}(j-1) + \mathbb{E}(\Delta_j \otimes \Delta_j) vec\, cov\mathcal{E}(j-1) + \\ &+ \mathbb{E}(\Delta_j \otimes \Delta_j) vec(y(j-1)y(j-1)^T) + vec\, cov\delta(j), \end{aligned} \tag{1.25}$$

где $\Delta_j = \mathbb{B}_j - B$, $\delta(j) = \mathbb{D}(j) - d$, vec – *операция векторизации матрицы, а* $\otimes$ – *операция кронекеровского произведения матриц.*

Доказательство. Подставим в (1.24) выражения для $\mathbb{B}_j$, $\mathbb{D}(j)$ и $\Xi(j)$:

$$y(j) + \mathcal{E}(j) = (B + \Delta_j)(y(j-1) + \mathcal{E}(j-1)) + d + \delta(j).$$

Поскольку $y(j)$ подчинен (1.23), получим выражения для $\mathcal{E}(j)$ и $\mathcal{E}(j)^T$:

$$\mathcal{E}(j) = B\mathcal{E}(j-1) + \Delta_j y(j-1) + \Delta_j \mathcal{E}(j-1) + \delta(j) \tag{1.26}$$

и

$$\mathcal{E}(j)^T = \mathcal{E}(j-1)^T B^T + y(j-1)^T \Delta_j^T + \mathcal{E}(j-1)^T \Delta_j^T + \delta(j)^T. \tag{1.27}$$

Далее необходимо перемножить правые и левые части равенств (1.26), (1.27) и вычислить математическое ожидание от всех членов получившегося равенства. При этом стоит отметить, что математические ожидания многих слагаемых правой части будут равны нулю. Так, например, $\mathbb{E}\Delta_j \mathcal{E}(j-1) y(j-1)^T \Delta_j^T = 0$ в силу того, что $\mathcal{E}(j-1)$ не зависит от Δ_j и $\mathbb{E}\mathcal{E}(j) = 0$. Учитывая эти соображения, имеем

$$\begin{aligned} cov\mathcal{E}(j) = & B cov\mathcal{E}(j) B^T + \mathbb{E}(\Delta_j cov\mathcal{E}(j-1)\Delta_j^T) + \\ &+ \mathbb{E}(\Delta_j y(j-1) y(j-1)^T \Delta_j^T) + cov\delta(j). \end{aligned}$$

Если столбцы матриц $Bcov\mathcal{E}(j)B^T$ составить в один столбец длины $2n$ и то же самое проделать с матрицей $cov\mathcal{E}(j-1)$, то получившиеся векторы

будут связаны соотношением

$$vec\left(Bcov\mathcal{E}(j)B^T\right) = (B \otimes B)\, vec\, cov\mathcal{E}(j-1).$$

Теперь, применяя такую операцию векторизации к матрицам $cov\mathcal{E}(j)$, $cov\mathcal{E}(j-1)$, $y(j-1)y(j-1)^T$ и $cov\delta(j)$, получим (1.25), что и доказывает лемму. □

Из леммы 3 можно вывести

Следствие 1. *Для стохастической устойчивости алгоритма (1.24) необходимо и достаточно, чтобы модуль первого собственного числа матрицы* $B \otimes B + \mathbb{E}(\Delta_j \otimes \Delta_j)$ *был меньше единицы.*

Известно (см., например, [25, 26]), что собственными числами $B \otimes B$ являются $\lambda_i(B)\lambda_k(B),\ i,k = 1,\dots,n$. Заметим также, что $\|B \otimes B\| = \|B\|^2$.

Для удобства введем следующие обозначения: $\mathcal{M}_j = \mathbb{E}(\Delta_j \otimes \Delta_j)$, $\mathcal{B}_j = B \otimes B + \mathcal{M}_j$ и $\mathcal{D}_j = \mathbb{E}(\Delta_j \otimes \Delta_j) vec(y(j-1)y(j-1)^T) + vec\, cov\delta(j)$. Тогда выражение для ковариации из леммы 3 примет вид

$$vec\, cov\mathcal{E}(j) = \mathcal{B}_j vec\, cov\mathcal{E}(j-1) + \mathcal{D}_j.$$

Элементы матрицы $\mathcal{M}_j$ имеют порядок $1/N_j$, где N_j – количество моделируемых траекторий на j-ой итерации, то есть $\mathcal{M}_j = \frac{1}{N_j}\mathcal{M}'$, где $\mathcal{M}'$ – матрица с элементами, не зависящими от N_j.

Лемма 3 и следствие из нее позволяют сделать определенные выводы относительно стохастической устойчивости рандомизированных процедур для итерационных процессов (1.20)–(1.21) и (1.22).

Теорема 7. *Если* $|\lambda_1(A)| < 1$, *и на каждой итерации (1.20)–(1.21) вычисления осуществляются при помощи метода Монте-Карло, то для любого натурального* m *существует такое* N', *что для* $\forall N_j > N'$, $j = 1, 2, \dots$, *где*

N_j – количество моделируемых траекторий цепи Маркова для оценки $y(j)$ в (1.20)–(1.21), рандомизированный итерационный процесс (1.20)–(1.21) будет стохастически устойчивым.

Доказательство. Рассмотрим (1.20) и заметим, что матрица B из Леммы 3 равна A^m, при этом в явном виде матрица A^m не известна и оценивается при помощи метода Монте-Карло. Учитывая, что $\|\mathcal{B}_j\| \leq \|B\|^2 + \frac{1}{N_j}\|\mathcal{M}'\|$, норму ковариации ошибки после вычисления (1.20) методом Монте-Карло можно оценить как

$$\|vec\, cov\mathcal{E}(j)\| \leq (\|A^m\|^2 + \frac{1}{N_j}\|\mathcal{M}'\|)\|vec\, cov\mathcal{E}(j-1)\| + \|\mathcal{D}_j\|.$$

Согласно (1.21) последующие l итераций ковариация ошибки будет изменяться в соответствии с

$$vec\, cov\mathcal{E}(j+1) = (A \otimes A + \frac{1}{N_{j+1}}\mathcal{M}')vec\, cov\mathcal{E}(j) + \mathcal{D}_{j+1}.$$

Так как $|\lambda_1(A)| < 1$, то $|\lambda_1(A \otimes A)| = \lambda_1^2(A) < 1$ и существует такое N', что для $\forall N_{j+1} > N'$ первое собственное число матрицы $A \otimes A + \frac{1}{N_{j+1}}\mathcal{M}'$ по модулю будет меньше единицы. Тогда, согласно следствию 1, рандомизированный итерационный процесс будет стохастически устойчивым. □

Оптимальный выбор параметров m, l, N' из теоремы 7 зависит от матрицы A, вида оценок и свойств многопроцессорной системы. Как видно, при достаточно большом N' предложенная процедура соответствует случаю частичной синхронизации в детерминированном случае, за исключением того, что на каждом шаге итерации возникает случайная ошибка. Помимо очевидного увеличения числа моделируемых траекторий N' дисперсия случайной ошибки может быть уменьшена путем применения различных техник уменьшения дисперсии (см. [4, 7, 27]). Важно заметить, что если количество процессоров больше порядка системы, то часть процессоров при использовании

асинхронных итераций будет простаивать, так как для расчета одной компоненты очередной итерации используется не более одного процессора. Использование метода Монте-Карло в тех же условиях позволяет эффективно использовать все имеющиеся процессоры, равномерно распределяя по ним моделируемые траектории цепей Маркова.

Похожую теорему можно сформулировать для (1.22).

Теорема 8. *Если* $|\lambda_1(A)| < 1$, *и на каждой итерации (1.22) вычисления осуществляются при помощи метода Монте-Карло, то для любого натурального* m *существует такое* N', *что для* $\forall N_{j+1} > N'$, *где* N_j – *количество моделируемых траекторий цепи Маркова для оценки* $y(j)$ *в (1.22), рандомизированный итерационный процесс (1.22) будет стохастически устойчивым.*

Доказательство. При применении метода Монте-Карло для вычисления очередного $y(j)$ в выражении (1.22) ковариация ошибки, согласно лемме 3, будет изменяться следующим образом

$$vec\, cov\mathcal{E}(j) = (A^m \otimes A^m + \frac{1}{N_j}\mathcal{M}')vec\, cov\mathcal{E}(j-1) + \mathcal{D}_j.$$

Сделанное предположение о матрице A, а именно $|\lambda_1(A)| < 1$, означает, что выполнено неравенство $|\lambda_1(A^m)| = |\lambda_1(A)|^m < 1$. Модуль первого собственного числа матрицы $A^m \otimes A^m$ равен $|\lambda_1(A)|^{2m}$. Этот факт позволяет заключить, что для любого натурального m существует такое N', что для $\forall N_{j+1} > N'$ модуль первого собственного числа матрицы $A^m \otimes A^m + \frac{1}{N_{j+1}}\mathcal{M}'$ будет меньше единицы. А это в свою очередь согласно следствию 1 означает, что процедура вычисления (1.22) методом Монте-Карло будет стохастически устойчивой. □

При использовании (1.22) и метода Монте-Карло может быть построен стохастически устойчивый алгоритм. Он будет обладать большей асинхрон-

ностью чем алгоритм, построенный на основе (1.20)–(1.21), но у такого алгоритма при равном числе моделируемых траекторий ковариация оценок будет больше.

Полученные результаты для систем линейных уравнений дают представление о периоде асинхронности применения последовательного метода Монте-Карло.

1.2.4. Система нелинейных уравнений

Основные принципы и идеи применения метода Монте-Карло к решению нелинейных уравнений можно найти, например, в [3, 4, 28]. В частности в [4] разобран пример решения интегрального уравнений с квадратичной нелинейностью. Однако в указанных источниках нет достаточно строгого и подробного описания оценок метода Монте-Карло решения систем алгебраических уравнений с полиномиальной нелинейностью.

Рассмотрим нелинейную систему размерности $n \in \mathbb{N}$, уравнениями в которой являются полиномы степени, не превосходящей $m \in \mathbb{N}$, следующего вида

$$x_i = \sum_{\alpha=(\alpha_1,\ldots,\alpha_n)} K_i^{\alpha} x_1^{\alpha_1} \ldots x_n^{\alpha_n},\ i = 1, \ldots, n, \tag{1.28}$$

где $\{\alpha = (\alpha_1, \ldots, \alpha_n)\}$ – векторы с целочисленными неотрицательными компонентами, для которых выполнено неравенство

$$\sum_{i=1}^{n} \alpha_i \leq m.$$

Далее иногда будет использоваться следующее сокращенное обозначение

$$x^{\alpha} = x_1^{\alpha_1} \ldots x_n^{\alpha_n}.$$

для векторов $x = (x_1, x_2, \ldots, x_n)$ и $\alpha = (\alpha_1, \ldots, \alpha_n)$. В таких обозначениях

система (1.28) примет вид

$$x_i = \sum_{\alpha} K_i^{\alpha} x^{\alpha},\ i = 1, \ldots, n.$$

Как и ранее получившуюся систему можно записать в векторной форме

$$x = Gx.$$

В некоторых случаях будет использоваться следующее обозначение

$$b_i = K_i^{(0,0,\ldots,0)}, i = 1, \ldots, n,$$

$$b = (b_1, \ldots, b_n).$$

Будем далее предполагать относительно оператора G, что он является сжимающим на замкнутом множестве $D \subset \mathbb{R}^n$ и выполнено $GD \subset D$. Это обеспечивает существование единственной неподвижной точки x^* оператора G на D, а также то, что итерационный процесс

$$x_{k+1} = Gx_k$$

сходится к x^* для любого начального $x_0 \in D$.

При решении линейных систем уравнений оценки метода Монте-Карло строились на линейных траекториях цепи Маркова, а ветвящиеся траектории использовались в качестве техники уменьшения дисперсии. В случае же с системами вида (1.28) оценки строятся на ветвящихся траекториях. При этом процесс, порождающий такие траектории, удобно интерпретировать как процесс эволюции популяции частиц. Изначально в популяции имеется одна частица. Каждая частица из популяции имеет единичную продолжительность жизни. В конце жизни каждая частица производит случайное количество потомков. Качественный и количественный состав потомков определяется рассматриваемой системой уравнений.

Рассмотрим связь процесса рождения/гибели частиц и системы (1.28) более подробно. Пусть имеется n типов частиц

$$T_1, T_2, \ldots, T_n,$$

каждая из которых связана с определенным уравнением системы (1.28) – частица i-ого типа связана с i-ым уравнением. Если в популяции есть частица i-ого типа, то её потомки определяются членами i-ого уравнения. Член уравнения вида

$$K_i^{\alpha} x_1^{\alpha_1} \ldots x_n^{\alpha_n}$$

предполагает рождение α_1 частиц первого типа, α_2 частиц второго типа и так далее вплоть до α_n частиц n-ого типа. Так, например, свободный член (все α_i равны нулю) подразумевает гибель частицы без рождения потомков. Конкретный же член i-ого уравнения, определяющий дальнейший сценарий, выбирается случайным образом, так или иначе согласованным с коэффициентами K_i^{α}.

Одна из возможных траекторий, связанных с системой

$$\begin{cases} x_1 = x_1^2 + 5x_1x_2 + x_1 + 1, \\ x_2 = x_2^2 + x_1x_2 + x_1^2 + 2 \end{cases}$$

приведена на рисунке 1.5.

В силу того, что указанная выше схема укладывается в понятие ветвящегося случайного процесса, для формального описания результатов разумно воспользоваться развитым математическим аппаратом этой области (см., например, [29, 30]).

Текущее состояние популяции частиц будем характеризовать вектором размерности n с целыми неотрицательными компонентами

$$\alpha = (\alpha_1, \ldots, \alpha_n),$$

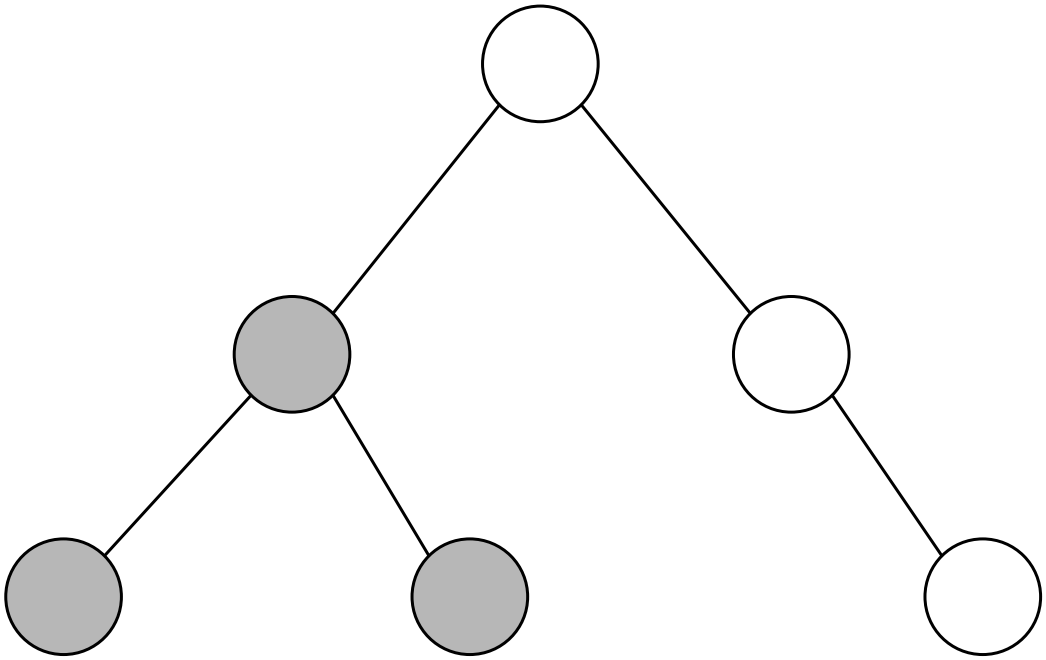

Рис. 1.5. Возможная траектория. Белые вершины соответсвуют первому уравнению системы, серые – второму.

который показывает, что популяция частиц состоит из α_1 частиц типа T_1, α_2 частиц типа T_2 и так далее.

Считая, что частица любого типа живёт единицу времени, динамика популяции будет рассматриваться в моменты времени $0, 1, 2, \ldots$, иначе называемые поколениями.

Обозначим за $P_\alpha^\beta(t_1, t_2)$ вероятность того, что система в момент времени t_2 находится в состоянии β, если в момент времени t_1 она находилась в состоянии α. Определенные для $t_1 \leq t_2$ вероятности $P_\alpha^\beta(t_1, t_2)$ удовлетворяют условиям

$$P_\alpha^\beta(t_1, t_2) \geq 0, \sum_\beta P_\alpha^\beta(t_1, t_2) = 1.$$

Здесь и далее суммирование $\sum_\beta$ производится по всем векторам с целочис-

ленными неотрицательными компонентами. Для случая $t_1 = t_2 = t$ примем

$$P_\alpha^\beta(t,t) = \begin{cases} 1, \alpha = \beta \\ 0, \alpha \neq \beta \end{cases}$$

Для марковских процессов, а именно с такими мы имеем дело, для любых $t_1 < t_2 < t_3$ вероятности $P_\alpha^\beta(\cdot,\cdot)$ удовлетворяют основному уравнению марковских случайных процессов

$$P_\alpha^\gamma(t_1,t_3) = \sum_\beta P_\alpha^\beta(t_1,t_2)P_\beta^\gamma(t_2,t_3). \tag{1.29}$$

Обозначим за $P_k^\alpha(t_1,t_2)$ вероятность того, что частица типа T_k за промежуток времени (t_1,t_2) превратится в совокупность частиц $\alpha = (\alpha_1,\ldots,\alpha_n)$. Для ветвящихся процессов вероятности $P_k^\alpha(t_1,t_2)$ не зависят от

- способа и времени возникновения исходной частицы типа T_k, предполагается лишь её наличие в момент t_1;
- частиц популяции в момент t_1, отличных от рассматриваемой частицы, и частиц, возникающих из нее в последующие поколения.

Для однородных марковских процессов, вероятности $P_\alpha^\beta(t_1,t_2)$ для которых зависят лишь от разности $t_2 - t_1$, уравнение (1.29) перепишется в виде

$$P_\alpha^\gamma(t+s) = \sum_\beta P_\alpha^\beta(t)P_\beta^\gamma(s).$$

Далее, если не будет оговорено другого, будут рассматриваться однородные марковские процессы.

Особую роль для ветвящихся процессов играют вероятности $P_k^\alpha = P_k^\alpha(1)$, вероятность того, что частица типа T_k за единицу времени превратится в совокупность частиц $\alpha = (\alpha_1,\ldots,\alpha_n)$.

Важным инструментом при исследовании ветвящихся процессов является производящая функция

$$F(t_1, t_2, x) = (F_1(t_1, t_2, x), \ldots, F_n(t_1, t_2, x)), \tag{1.30}$$

где $x = (x_1, \ldots, x_n)$,

$$F_k(t_1, t_2, x) = \sum_{\alpha} P_k^{\alpha}(t_1, t_2) x^{\alpha}. \tag{1.31}$$

Функции $F_k(t_1, t_2, x)$ определены при $t_1 \leq t_2$ и удовлетворяют неравенствам

$$|F_k(t_1, t_2, x)| \leq 1,$$

если $|x_i| \leq 1, i = 1, \ldots, n$ и равенствам

$$F_k(t_1, t_2, 1, 1, \ldots, 1) = 1.$$

Производящая функция удовлетворяет (см. [29]) функциональному уравнению

$$F(t_1, t_3, x) = F(t_1, t_2, F(t_2, t_3, x)) \tag{1.32}$$

и граничному условию

$$F(t_1, t_1, x) = x.$$

В нашем случае, то есть для однородных процессов, производящие функции будут иметь вид

$$F_k(t, x) = \sum_{\alpha} P_k^{\alpha}(t) x^{\alpha},$$

а основную роль будут играть производящие функции

$$F_k(1, x) = F_k(x) = \sum_{\alpha} P_k^{\alpha} x^{\alpha}.$$

Уравнение (1.32) для однородных процессов приобретет совсем простой вид

$$F(t + 1, x) = F(F(t, x)) = F^{(t+1)}(x),$$

то есть $F(t,x)$ является t-й итерацией функции $F(x)$. Иными словами поведение ветвящегося процесса определяется итерациями функции $F(x)$.

Как и в случае систем линейных алгебраических уравнений нам будут интересны обрывающиеся траектории процесса, и именно на них будут строиться оценки метода Монте-Карло. Процесс будет считаться выродившимся, если не осталось ни одной частицы типов $T_1, T_2, \ldots, T_n$. Вероятность того, что процесс, начавшись с одной частицы типа T_k, выродится к поколению t, равна

$$P_k^{(0,0,\ldots,0)}(t) = F_k(t,0,0,\ldots,0).$$

Вероятность того, что процесс, начавшись с одной частицы типа T_k, рано или поздно выродится, равна пределу $P_k^{(0,0,\ldots,0)}(t)$ при $t \to \infty$

$$P_k = lim_{t\to\infty} P_k^{(0,0,\ldots,0)}(t) = lim_{t\to\infty} F_k(t,0,0,\ldots,0).$$

Будем считать процесс вырождающимся, если все $P_k = 1, k = 1,\ldots,n$.

Динамику популяции, начавшейся с одно частицы типа T_k и выродившейся к моменту времени τ, можно записать следующим образом

$$\alpha(0) \to \alpha(1) \to \cdots \to \alpha(\tau),$$

где $\alpha(0) = (0,\ldots,1_k,\ldots,0)$, $\alpha(j) = (\alpha_1(j),\ldots,\alpha_n(j))$ – популяция в поколение j, а $\alpha(\tau) = (0,0,\ldots,0)$.

Данное представление, однако, не достаточно подробно для наших целей. Ему могут соответствовать сразу несколько траекторий ветвящегося процесса. Траекторию ветвящегося процесса удобно представлять в виде ориентированного графа, а точнее сказать дерева, в котором вершинами являются частицы, а направленные ребра обозначаю процесс превращения старой частицы в совокупность новых. Чтобы однозначно определить траекторию ветвящегося процесса, необходимо особым образом пронумеровать/обозначить вершины дерева, соответствующего этой траектории.

Приведем такие обозначения. Начальная частица типа T_k будет обозначаться (k), её i-ый потомок типа T_{k_1} будет обозначаться (kk_1^i), частицы n-ого поколения будут нумероваться

$$(kk_1^{i_1}k_2^{i_2}\dots k_n^{i_n}).$$

Такая запись полностью определяет путь превращений от частицы типа T_k нулевого поколения до частицы типа T_{k_n} поколения n. При этом всё дерево будет однозначно определяться метками концевых вершин, соответствующих частицам, погибшим без рождения потомков.

Для траектории с рисунка 1.5 вершины пронумеруются так, как показано на рисунке 1.6.

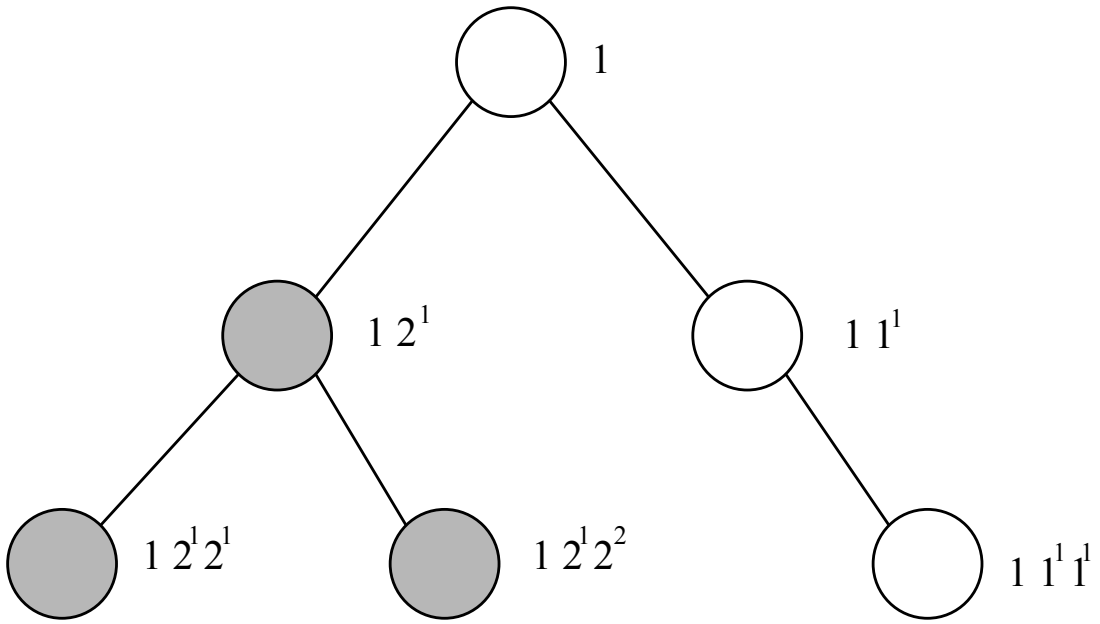

Рис. 1.6. Пронумерованные вершины

Такие обозначения в определенном смысле полезны, однако являются достаточно громоздкими для использования в выражении вероятности траектории. Для вычисления вероятности траектории важно знать в какое состояние перейдет та или иная частица из поколения.

Пронумеруем частицы типа k поколения l от 1 до $\alpha_k(l)$. Обозначим состояние, в которое перейдет j-ая частица типа T_k поколения l за $\alpha^{(k,j)}(l+1)$. В таких обозначения будет иметь место равенство

$$\sum_{k,j} \alpha^{(k,j)}(l+1) = \alpha(l+1),$$

а выражение для вероятности траектории

$$P_k^{\alpha(1)} \prod_{i_1=1}^{n} \prod_{j_1=1}^{\alpha_{i_1}(1)} P_{i_1}^{\alpha^{(i_1,j_1)}(2)} \cdots \prod_{i_\tau=1}^{n} \prod_{j_\tau=1}^{\alpha_{i_\tau}(\tau+1)} P_{i_\tau}^{\alpha^{(i_\tau,j_\tau)}(\tau+1)}. \tag{1.33}$$

Для того, чтобы вероятности (1.33) задавали вероятностную меру на пространстве траекторий необходимо, чтобы процесс был вырождающимся.

Для формулировки необходимых и достаточных условий вырождаемости процесса введем классификацию типов частиц (см. [29])

$$T_1, T_2, \ldots, T_n.$$

Для каждого типа $T_k, k = 1, \ldots, n$ выделим два подмножества A_k и B_k из совокупности типов $T_1, T_2, \ldots, T_n$. Положим $T_i \in A_k$, если при некотором t с положительной вероятностью из частицы типа T_i образуется частица типа T_k. Положим $T_j \in B_k$, если при некотором t с положительной вероятностью из частицы типа T_k образуется частица типа T_j. Пересечением этих типов назовем классом сообщающихся типов $S^{(k)} = A_k \cap B_k$. Назовем класс сообщающихся типов Ψ финальным, если каждая частица любого из типов класса Ψ всегда с вероятностью единица среди своего потомства имеет ровно одну частицу какого-либо типа из класса Ψ и может иметь еще какие-нибудь частицы типов, не входящих в Ψ.

Известна (см. [29]) следующая теорема.

Теорема 9. *Пусть случайный ветвящийся процесс с типами частиц $T_1, T_2, \ldots, T_n$ задаётся производящими функциями $F_1(x), \ldots, F_n(x)$. Для то-*

го, чтобы он был вырождающимся, необходимо и достаточно, чтобы выполнялись следующие условия

1. *Система типов $T_1, T_2, \ldots, T_n$ не содержит ни одного финального класса.*

2. *Все собственные числа матрицы $A = \|A_{i,j}\|_{i,j=1}^n$, где*

$$A_{i,j} = \frac{\partial F_i(1, 1, \ldots, 1)}{\partial x_j},$$

лежат в единичном круге $|\lambda| < 1$.

Далее будем предполагать, что для рассматриваемых ветвящихся процессов выполнены условия теоремы.

Теперь, после сделанной подготовительной работы, вернемся к поиску решения задачи (1.28) методом Монте-Карло. С системой (1.28) свяжем случайны ветвящийся процесс, определяемый производящими функциями вида (1.31), в которых вероятности P_k^α удовлетворяют условиям согласования:

- $P_k^\alpha > 0$ для всех k и α, для которых $K_k^\alpha \neq 0$.

Будем предполагать, что для уравнения (1.28) выполнено мажорантное условие: сходится метод последовательных приближений

$$x_i(j+1) = \sum_\alpha |K_i^\alpha| x^\alpha(j),\ i = 1, \ldots, n \tag{1.34}$$

при начальном $x_i(0) = b_i, i = 1, \ldots, n$.

Оценки метода Монте-Карло строятся на траекториях ветвящегося процесса, поэтому на траектории (1.33) одной из оценок может быть аналог оценки по поглощению для линейного случая, а именно

$$\xi_k = \frac{K_k^{\alpha(1)}}{P_k^{\alpha(1)}} \prod_{i_1=1}^{n} \prod_{j_1=1}^{\alpha_{i_1}(1)} \frac{K_{i_1}^{\alpha^{(i_1,j_1)}(2)}}{P_{i_1}^{\alpha^{(i_1,j_1)}(2)}} \cdots \prod_{i_\tau=1}^{n} \prod_{j_\tau=1}^{\alpha_{i_\tau}(\tau+1)} \frac{K_{i_\tau}^{(0,0,\ldots,0)}}{P_{i_\tau}^{(0,0,\ldots,0)}}. \tag{1.35}$$

Если бы ветвящийся процесс был таким, что все траектории обрывались бы к поколению M, то математическое ожидание (1.35) вычислялось бы как сумма

$$\sum_{\tau=0}^{M}\sum_{\alpha(1),\ldots,\alpha(\tau)} K_k^{\alpha(1)}\prod_{i_1=1}^{n}\prod_{j_1=1}^{\alpha_{i_1}(1)} K_{i_1}^{\alpha^{(i_1,j_1)}(2)}\cdots\prod_{i_\tau=1}^{n}\prod_{j_\tau=1}^{\alpha_{i_\tau}(\tau+1)} K_{i_\tau}^{(0,0,\ldots,0)} = \\ = (G^{(\tau)}(0))_k = (G^{(\tau-1)}(b))_k \tag{1.36}$$

Последнее равенство легко проверяется индукцией по M. Действительно,

$$G_k(0) = b_k = (G^{(0)}(b))_k,$$

а индукционный переход

$$G^{M+1}(0) = G(G^M(0)) = G(G^{M-1}(b)) = G^M(b).$$

В сделанных предположениях, хотя процесс и вырождающийся, а следовательно почти все траектории конечны, однако могут иметь сколь угодно много поколений. Поэтому, чтобы вычислить $\mathbb{E}\xi_k$ нужно перейти в (1.36) к пределу при $M \to \infty$. Учитывая мажорантное условие (1.34), получим

$$\mathbb{E}\xi_k = \lim_{M\to\infty}(G^{(M)}(b))_k = x_k^*. \tag{1.37}$$

Выражение для второго момента будет иметь вид

$$\mathbb{E}\xi_k^2 = \lim_{M\to\infty}\hat{G}^{(M)}(b), \tag{1.38}$$

где

$$\hat{G}_i(x) = \sum_\alpha \frac{(K_i^\alpha)^2}{P_i^\alpha}x^\alpha,\ i = 1,\ldots,n.$$

Таким образом, в данном разделе было дано формальное описание метода Монте-Карло для решения систем алгебраических уравнений с полиномиальной нелинейностью. Используя аппарат теории ветвящихся процессов,

был описан процесс построения оценок на ветвящихся траекториях, связанных с решаемой системой. Были исследованы свойства построенных оценок. Важно будет отметить, что как и в случае с линейными системами процесс вычисления оценки решения без труда распараллеливается путем распределения моделируемых ветвящихся траекторий по разным процессорам.

1.3. Численные эксперименты

Для асинхронных итераций численные эксперименты проводились для случайным образом смоделированной матрицы A размерности 4×4, такой что $\lambda_1(A) = 0.76$, $\lambda_1(|A|) = 1.64$, $\|A\| = 2.04$.

На рисунке 1.7 изображена норма вектора ошибок для асинхронных итераций с множествами

$$T_i = \{j \in \mathbb{N} | j \, mod \, i = 0\}, \, i = 1, \ldots, 4,$$

а $\tau_i(j) = j - 1$, $i = 1, \ldots, 4$, $j = 1, 2, \ldots$. В данном случае наблюдается экспоненциальный рост ошибки.

На рисунке 1.8 изображена норма вектора ошибок для асинхронных итераций с частичной синхронизацией. Теорема 4 гарантирует сходимость нормы ошибки к нулю при $m = 2, l = 6$, однако сходимость будет наблюдаться уже при значениях $m = 2, l = 3$.

Для метода Монте-Карло численные эксперименты проводились для случайным образом смоделированной матрицы A размерности 10×10, такой что $\lambda_1(A) = 0.79$, $\lambda_1(|A|) = 3.21$, $\|A\| = 8.45$.

На рисунке 1.9 изображено стандартное отклонение оценки при решении (1.22) методом Монте-Карло с использованием оценок (1.14) и (1.18) для параметров $m = 5$, $N_j = 100$. В этом случае наблюдается явление стохастической неустойчивости.

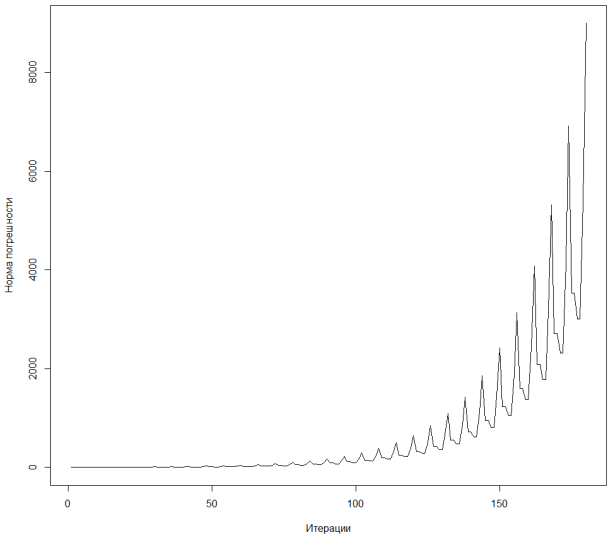

Рис. 1.7. Расходимость асинхронных итераций

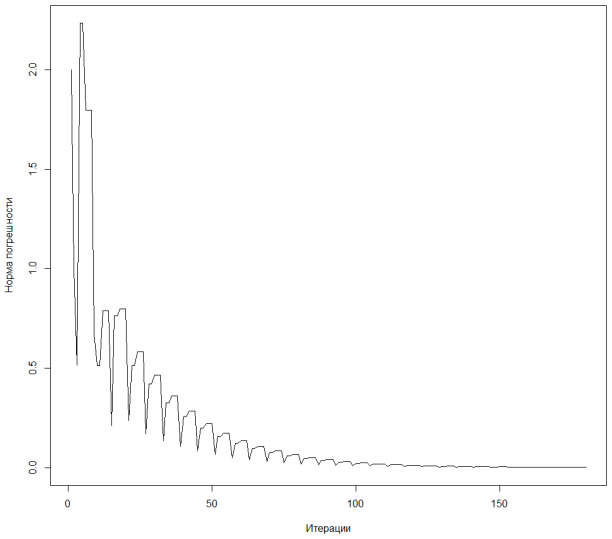

Рис. 1.8. Частичная синхронизация

Увеличение количества моделируемых траекторий до $N_j = 1000$ исправляет ситуацию, что можно наблюдать на рисунке 1.10

На рисунке 1.11 изображено стандартное отклонение оценки при решении (1.20)–(1.21) методом Монте-Карло использованием оценок (1.14) и (1.18) для параметров $m = 5, l = 3$. При этом $N_j = 1000$ при расчете (1.20) и $N_j = 1000$ при расчете каждой итерации в (1.21). Пики на графиках соответствуют периоду использования асинхронного метода.

В качестве примера использования метода Монте-Карло для решения систем с полиномиальной нелинейностью рассмотрим матричное уравнения Риккати (см. [31, 32])

$$\frac{dX}{dt} = XA(t)X + B_1(t)X + XB_2(t) + C(t), X(t_0) = X_0 \qquad (1.39)$$

где X – матрица неизвестных размерности $n \times n$, $A(t), B_1(t), B_2(t), C(t)$ – заданные матрицы, зависящие от t, размерности $n \times n$. Уравнение Риккати играет важную роль в вариационном исчислении и в квантовой теории поля.

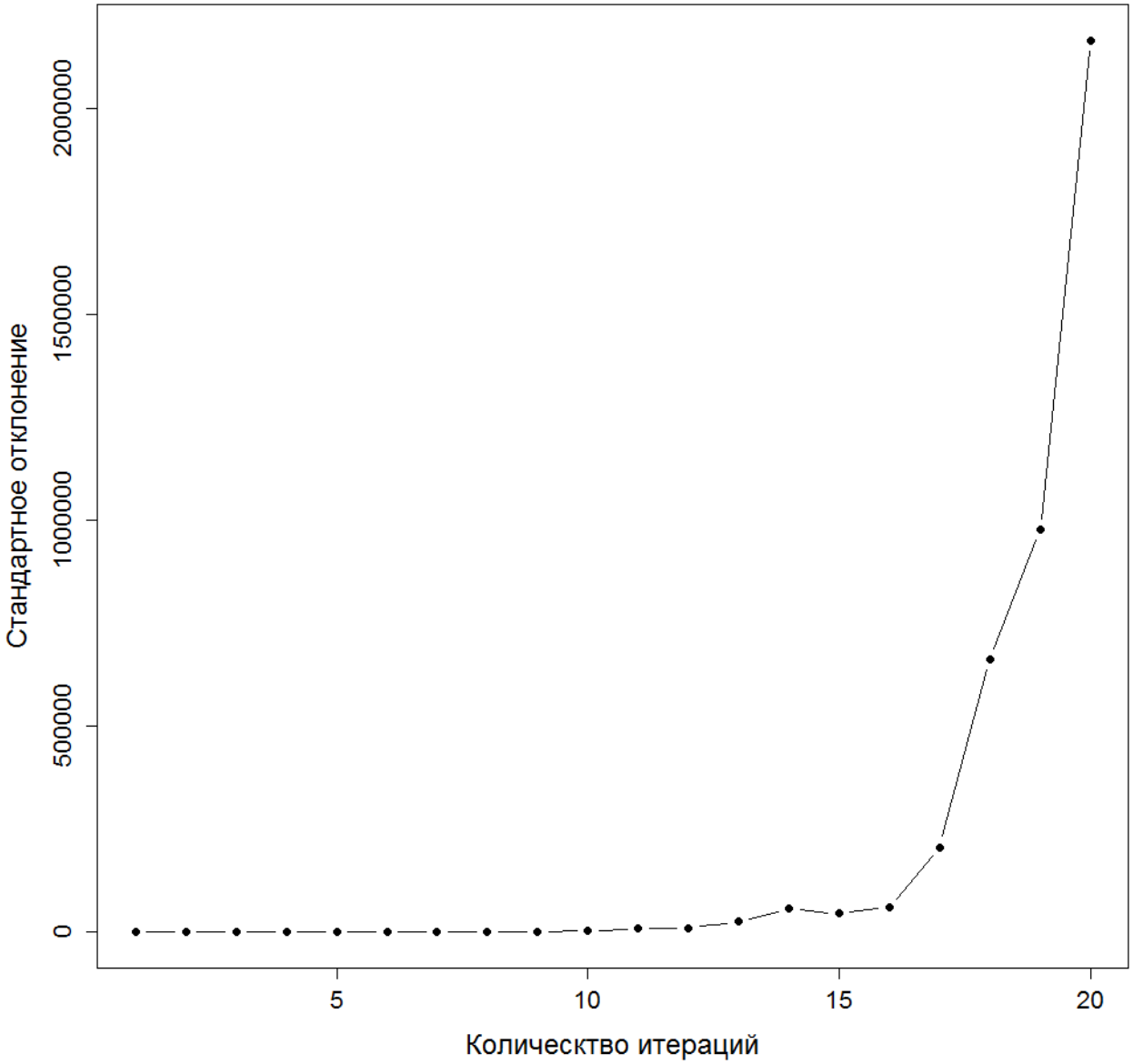

Рис. 1.9. Стохастическая неустойчивость при малом кол-ве моделируемых траекторий

Необходимость решать систему нелинейных уравнений может возникнуть, например, при использовании неявного метода Рунге-Кутты (см. [33, 34]):

$$X_k = \Delta t \left(X_k A(t_k) X_k + B_1(t_k) X_k + X_k B_2(t_k) + C(t_k) \right) + X_{k-1}, k = 1, 2, \ldots \tag{1.40}$$

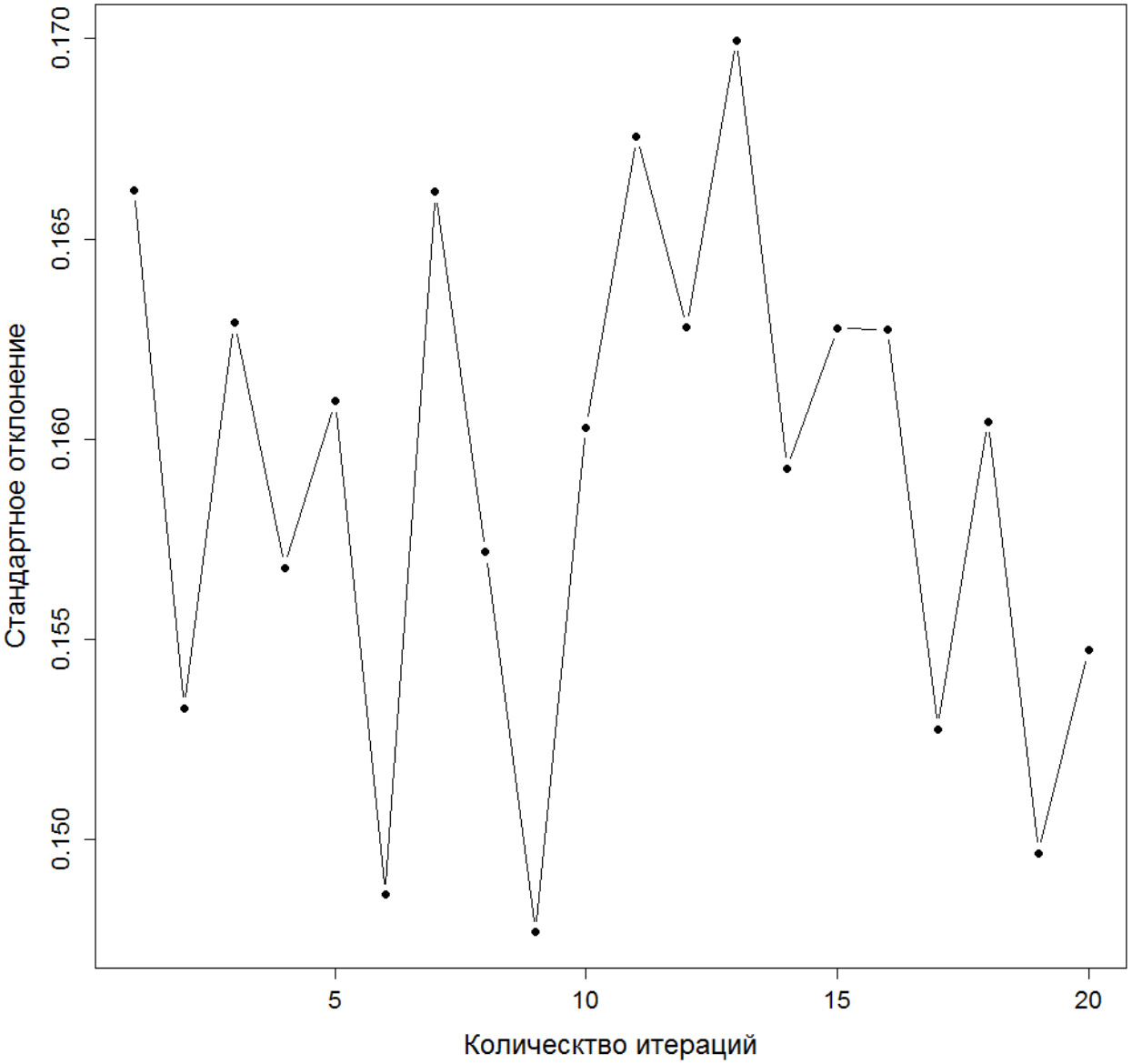

Рис. 1.10. Ограниченное отклонение

где Δt – шаг по времени, $t_k = t_0 + k\Delta t_k$, $X_k = X(t_k)$.

В модельном примере будем считать матрицы $A(t), B_1(t), B_2(t), (t)$, не зависящими от времени. Результаты эксперимента приведены в таблице 1.1

Одновременное выполнение условий $|\lambda_1(A)| < 1$ и $\lambda_1(|A|) > 1$ может создать определенные трудности при использовании как детерминированных,

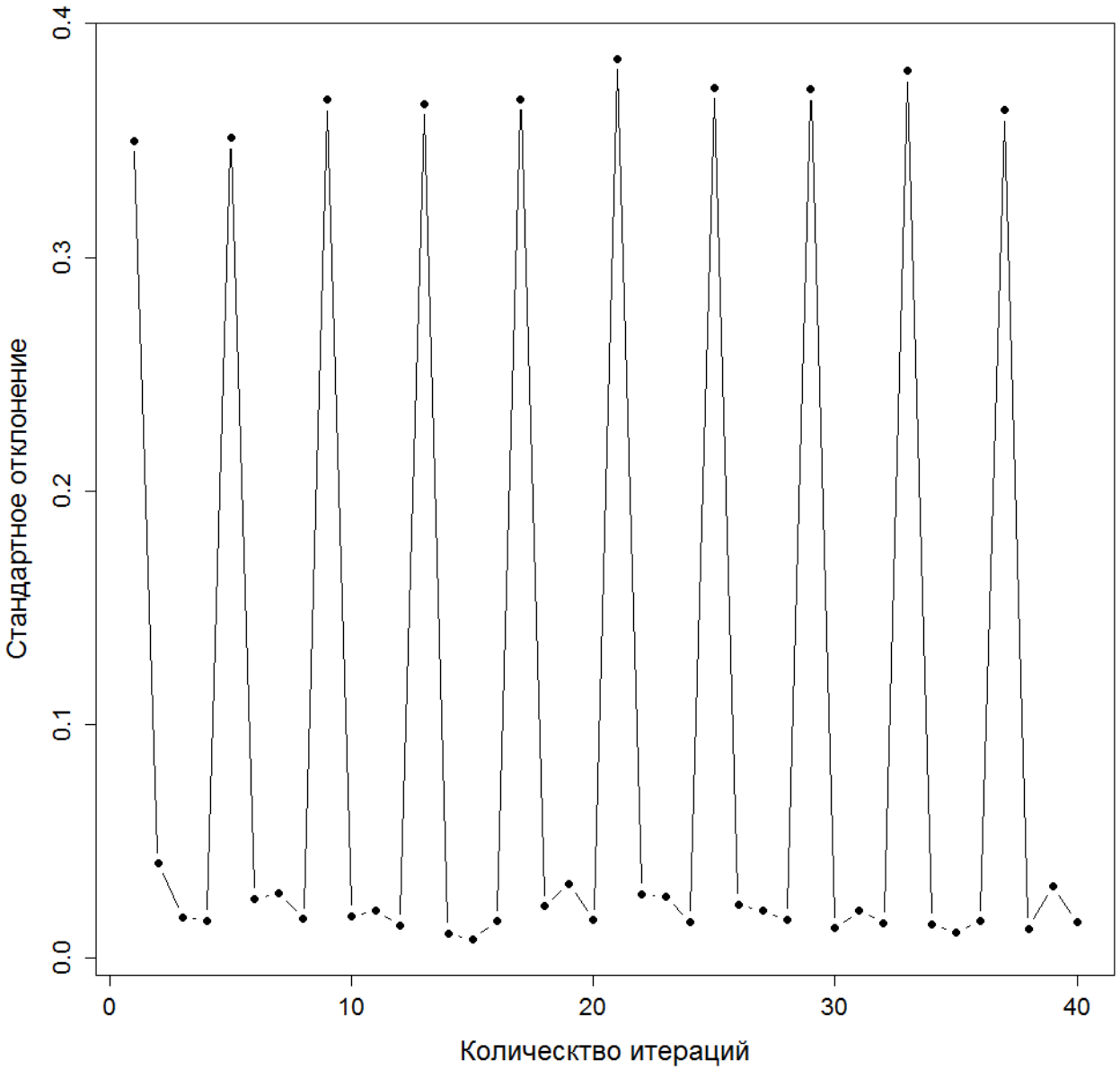

Рис. 1.11. Оценка с частичной синхронизацией

так и стохастических асинхронных методов для решения системы (1.12).

В детерминированном случае возможным выходом является использование частичной синхронизации, когда асинхронные итерации чередуются с синхронными. Представление о соотношении количества синхронных и асинхронных итерации даёт теорема 4. Стоит отметить, что такой подход, вообще

Таблица 1.1

Количество траекторий	N = 10	N = 100	N = 1000	N = 10000
Норма ошибки	0.18	0.03	0.0017	0.0057
Стандартное отклонение	0.094	0.028	0.009	0.003

говоря, вписывается в концепцию асинхронных итераций.

Комбинируя алгоритмы асинхронного типа и алгоритмы с синхронизацией метода Монте-Карло, мы, как и в детерминированном случае, можем расширить сферу асинхронности при использовании стохастических методов. Случайный характер ошибки создаёт свои особенности, которые, как мы видели, легко учесть. Богатый арсенал средств понижения дисперсии оценок создаёт много возможностей для совершенствования алгоритмов и, по мнению автора, для больших систем уравнений метод Монте-Карло может быть предпочтительнее детерминированных асинхронных итераций.

Глава 2

Глава 2. Решение систем обыкновенных дифференциальных уравнений методом Монте-Карло

В данном разделе речь пойдет о решении систем обыкновенных дифференциальных уравнений. Основное внимание будет сосредоточено на алгоритмах метода Монте-Карло, однако один из параграфов будет посвящен методу асинхронных итераций. Рассмотрение рандомизированных методов обусловлено потенциально большой размерностью системы и необходимостью проводить вычисления на многопроцессорных системах, а именно в этих аспектах метод Монте-Карло зарекомендовал себя в качестве эффективного метода.

Рассмотрим систему обыкновенных дифференциальных уравнений первого порядка, записанную в векторной форме

$$\dot{x} = f(t, x), \tag{2.1}$$

где $x = x(t) = (x_1(t), x_2(t), \ldots, x_n(t))^T$ – искомая вектор-функция, $t \in \mathbb{R}$, $f(t, x) = (f_1(t, x), f_2(t, x), \ldots, f_n(t, x))^T$ – оператор из $\mathbb{R}^{n+1}$ в $\mathbb{R}^n$, $n \in \mathbb{N}$ – размерность системы. Пусть также задано начальное условие

$$x(t_0) = x_0. \tag{2.2}$$

Относительно функций $f_i(t, x), i = 1, \ldots, n$ предполагается, что они непрерывны на области определения. Точно так же будет предполагаться, что частные производные

$$\frac{\partial f_i(t, x_1, \ldots, x_n)}{\partial x_j}, i, j = 1, \ldots, n$$

существуют и непрерывны на области определения.

Идея предлагаемого метода состоит в том, чтобы заменить систему дифференциальных уравнений (2.1) на систему интегральных уравнений, имеющую то же решение, что и исходная система. А затем решать получившуюся систему интегральных уравнений методом Монте-Карло.

Вероятно, самой простой из таких замен, не требующих никаких дополнительных преобразований правой части уравнения (2.1), является интегральное уравнение

$$x(t) = \int_{t_0}^{t} f(\tau, x(\tau))d\tau + x_0. \tag{2.3}$$

Впрочем, в некоторых случаях представляется выгодным выделить линейную часть у функции $f(t, x)$. Рассмотрим равносильную (2.1) систему следующего вида

$$\dot{x} = A(t)x + f(t, x) - A(t)x, \tag{2.4}$$

где $A(t)$ – матрица $n \times n$, зависящая от t. Введем следующие обозначения

$$g(t, x) = f(t, x) - A(t)x,$$

$$B(t) = \int_{t_0}^{t} A(t)dt,$$

где операция интегрирования выполняется поэлементно, и перейдем к интегральному уравнению

$$x(t) = e^{B(t)} \int_{t_0}^{t} e^{-B(\tau)} g(\tau, x(\tau))d\tau + e^{B(t)}x_0. \tag{2.5}$$

Нетрудно проверить, что если $x = \varphi(t)$ – некоторое решение уравнения (2.4), определенное на интервале $r_1 < t < r_2$ и удовлетворяющее начальному условию $\varphi(t_0) = x_0$, то для функции $\varphi(t)$ на всем интервале $r_1 < t < r_2$ выполнено интегральное тождество (2.5). Обратно, если для некоторой непрерывной

функции $x = \varphi(t)$ на интервале $r_1 < t < r_2$ выполнено интегральное тождество (2.5), то функция $x = \varphi(t)$ дифференцируема, является решением уравнения (2.4) и удовлетворяет начальному условию (2.2).

Далее будут исследованы алгоритмы метода Монте-Карло для решения системы (2.1), в которой компоненты функций f или g являются полиномами от x степени, не превосходящей $m \in \mathbb{N}$. В этом случае будем считать, что тем или иным способом исходная система (2.1) преобразуется к системе интегральных уравнений вида

$$x_i(t) = \int_{t_0}^{t} \sum_{\alpha=(\alpha_1,\ldots,\alpha_n)} K_i^{\alpha}(t,\tau) x_1^{\alpha_1}(\tau) \ldots x_n^{\alpha_n}(\tau) d\tau + K_i^{(0,\ldots,0)}(t),\ i = 1,\ldots,n, \quad (2.6)$$

где $\{\alpha = (\alpha_1, \ldots, \alpha_n)\}$ – векторы с целочисленными неотрицательными компонентами, для которых выполнено неравенство

$$\sum_{i=1}^{n} \alpha_i \leq m.$$

Далее, как и в случае с системами уравнений из первой главы, иногда будет использоваться следующее сокращенное обозначение

$$x^{\alpha} = x_1^{\alpha_1} \ldots x_n^{\alpha_n}.$$

для векторов $x = (x_1, x_2, \ldots, x_n)$ и $\alpha = (\alpha_1, \ldots, \alpha_n)$. В таких обозначениях систему (2.6) можно записать в виде

$$x_i(t) = \int_{t_0}^{t} \sum_{\alpha} K_i^{\alpha}(t,\tau) x^{\alpha}(\tau) d\tau + K_i^{(0,\ldots,0)}(t),\ i = 1,\ldots,n.$$

Будем считать, что свободные члены x_0 из уравнения (2.3) и $e^{B(t)}x_0$ из уравнения (2.5) входят в $K_i^{(0,\ldots,0)}(t)$, который для краткости будем обозначать $K_i^0(t)$.

2.1. Частные случаи

Начнем рассмотрение с частного случая системы (2.6) - с линейного интегрального уравнения

$$x(t) = \int_{t_0}^{t} K(t,\tau)x(\tau)d\tau + c(t), \tag{2.7}$$

Применение метода Монте-Карло для решения интегрального уравнения второго рода подробно описано в [4, 35]. Используем указанную методику для решения уравнения (2.7).

Отличительной особенностью уравнения (2.7) является наличие переменного верхнего предела в интеграле. Эта особенность влияет на свойства цепи Маркова, используемой для оценки решения уравнения.

Пусть $\varphi(t)$ – решение уравнения (2.7), удовлетворяющее начальному условию $\varphi(t_0) = x_0$. Далее уравнению (2.7) сопоставляется цепь Маркова с множеством состояний $[t_0, t]$. Цепь задаётся плотностью начального распределения $p^0(t)$ и плотностью перехода из состояния t в состояние τ: $p(t,\tau)$. Относительно плотности $p(t,\tau)$ предполагается, что она субстохастична при любом $t \in [t_0, \infty)$

$$\int_{t_0}^{t} p(t,\tau)d\tau = 1 - g(t),\ 0 \le g(t) < 1, \tag{2.8}$$

где $g(t)$ – вероятность поглощения (обрыва траектории). На траекториях цепи Маркова $\omega_l = \tau_0 \to \tau_1 \to \cdots \to \tau_l$ вводится функционал

$$\mathcal{J}(\omega_l) = \frac{h(\tau_0)K(\tau_0,\tau_1)\dots K(\tau_{l-1},\tau_l)c(\tau_l)}{p^0(\tau_0)p(\tau_0,\tau_1)\dots p(\tau_{l-1},\tau_l)g(\tau_l)}. \tag{2.9}$$

Как было показано в [4] для того, чтобы оценка (2.9) была несмещенной оценкой функционала (φ, h) необходимо и достаточно, чтобы сходился ряд

$$\sum_{l=0}^{\infty} \int_{t_0}^{t}\int_{t_0}^{\tau_0} \dots \int_{t_0}^{\tau_{l-1}} |h(\tau_0)K(\tau_0,\tau_1)\dots K(\tau_{l-1},\tau_l)c(\tau_l)|d\tau_l \dots d\tau_0 \tag{2.10}$$

и выполнялись условия согласования:

- $p^0(t) > 0$ для тех t, для которых $h(t) \neq 0$,
- $p(t,\tau) > 0$ для тех $(t,\, \tau)$, для которых $K(t,\tau) \neq 0$,
- $g(t) > 0$ для тех t, для которых $c(t) \neq 0$.

Доказательство теоремы существования (см., например, [36]) решения уравнения (2.1) носит конструктивный характер, решение ищется при помощи метода последовательных приближений. В ходе доказательства находятся такие положительные константы q и r, что ряд

$$\sum_{l=0}^{\infty} \int_{t_0}^{t} \int_{t_0}^{\tau_0} \dots \int_{t_0}^{\tau_{l-1}} |K(\tau_0,\tau_1) \dots K(\tau_{l-1},\tau_l) c(\tau_l)| d\tau_l \dots d\tau_0$$

сходится при условии $|t - t_0| < q$ и $|x(t) - x_0| < r$. Отсюда следует также и сходимость ряда (2.10).

Наличие переменного верхнего предела и субстохастической плотности $p(t,\tau)$ означает, что последовательность моделируемых состояний в цепи Маркова будет не возрастать.

Следующим по сложности случаем системы (2.6) – система линейных интегральных уравнений

$$\begin{cases} x_1(t) = \sum_{j=1}^{n} \int_{t_0}^{t} K_{1,j}(t,\tau) x_j(\tau) d\tau + c_1(t), \\ \vdots \\ x_n(t) = \sum_{j=1}^{n} \int_{t_0}^{t} K_{n,j}(t,\tau) x_j(\tau) d\tau + c_n(t). \end{cases} \tag{2.11}$$

или в векторной форме

$$x(t) = \int_{t_0}^{\infty} K(t,\tau) x(\tau) d\tau + c(t), \tag{2.12}$$

где $K(t,\tau)$ – матрица $n \times n$.

В этом случае предлагается рассматривать векторную марковскую цепь вида

$$\omega_l = \begin{pmatrix} i_0 \\ \tau_0 \end{pmatrix} \to \begin{pmatrix} i_1 \\ \tau_1 \end{pmatrix} \to \cdots \to \begin{pmatrix} i_l \\ \tau_l \end{pmatrix}$$

с множеством состояний $\{1, \ldots, n\} \times [t_0, t)$. Предполагается, что для первой компоненты вектора заданы плотность начального распределения $q^0 = (q_1^0, q_2^0, \ldots, q_n^0)$ и матрица переходных вероятностей $\mathcal{Q}(t) = \|q_{i,j}(t)\|_{i,j=1}^n$, которая является субстохастической для любого $t \in [t_0, \infty)$

$$\sum_{j=1}^{n} q_{i,j}(t) = 1 - g_i(t),\ 0 \le g_i(t) < 1,\ i = 1, \ldots, n,$$

где $g(t) = (g_1(t), \ldots, g_n(t))$ – вероятности поглощения (обрыва траектории). Для второй компоненты задана матрица переходных плотностей $\mathcal{P}(t, \tau) = \|p_{i,j}(t, \tau)\|_{i,j=1}^n$, таких что для $t \in [t_0, \infty)$

$$\int_{t_0}^{t} p_{i,j}(t, \tau) d\tau = 1,\ i, j = 1, \ldots, n,$$

а τ_0 всегда равен t.

Моделирование такой цепи осуществляется следующим образом:

Шаг 0. Инициализация. k:=0;

Шаг 1. Моделируется плотность q^0. Вычисляется i_0;

Шаг 2. По неравенству $\alpha < g_{i_k}(\tau_k)$, где α – равномерно распределенная на отрезке $[0, 1]$ случайная величина, проверяется является ли состояние (i_k, τ_k) поглощающим. В случае выполнения неравенства траектория обрывается, и моделирование заканчивается. Иначе выполняется переход к шагу 3;

Шаг 3. Моделируется распределение $(q_{i_k,1}(\tau_k),\dots,q_{i_k,n}(\tau_k))$. Вычисляется i_{k+1}. Моделируется плотность $p_{i_k,i_{k+1}}(t,\tau)$. Вычисляется τ_k, k заменяется на $k+1$ и выполняется переход к шагу 2.

Плотность распределения такой траектории есть

$$q^0_{i_0}q_{i_0,i_1}(t)p_{i_0,i_1}(\tau_0,\tau_1)\dots q_{i_{l-1},i_l}(\tau_{l-1})p_{i_{l-1},i_l}(\tau_{l-1},\tau_l)g_{i_l}(\tau_l).$$

На траекториях цепи Маркова вводится функционал

$$\mathcal{J}(\omega_l)=\frac{h(\tau_0)K_{i_0,i_1}(\tau_0,\tau_1)\dots K_{i_{l-1},i_l}(\tau_{l-1},\tau_l)c_{i_l}(\tau_l)}{q^0_{i_0}q_{i_0,i_1}(t)p_{i_0,i_1}(\tau_0,\tau_1)\dots q_{i_{l-1},i_l}(\tau_{l-1})p_{i_{l-1},i_l}(\tau_{l-1},\tau_l)g_{i_l}(\tau_l)}. \tag{2.13}$$

Вычислим математическое ожидание $\mathcal{J}(\omega_l)$:

$$\mathbb{E}\mathcal{J}(\omega_l)=\sum_{l=1}^{\infty}\sum_{i_0=1}^{n}\cdots\sum_{i_l=1}^{n}\int_{t_0}^{t}\dots\int_{t_0}^{\tau_{l-1}}h_{i_0}(\tau_0)K_{i_0,i_1}(\tau_0,\tau_1)\dots$$

$$\dots K_{i_{l-1},i_l}(\tau_{l-1},\tau_l)c_{i_l}(\tau_l)d\tau_l\dots d\tau_1=\sum_{i_0=1}^{n}(h_{i_0},x_{i_0})$$

Условия согласования в этом случае примут вид

- $q^0_i>0$ для тех i, для которых $h_i\neq 0$;
- $q_{i,j}(t)>0$ для тех i,j,t, для которых $\int_{t_0}^{t}K_{i,j}(t,\tau)d\tau\neq 0$;
- $p_{i,j}(t,\tau)>0$ для тех $(t,\,\tau)$, для которых $K_{i,j}(t,\tau)\neq 0$, $i,j=1,\dots,n$;
- $g_i(t)>0$ для тех t, для которых $c_i(t)\neq 0$, $i=1,\dots,n$.

Выражение для второго момента (2.13), полезное для многих целей, может быть получено путем прямых выкладок.

$$\mathbb{E}\mathcal{J}^2(\omega_l)=\sum_{l=1}^{\infty}\sum_{i_0,\dots,i_l=1}^{n}\int_{t_0}^{t}\dots$$

$$\dots\int_{t_0}^{\tau_{l-1}}\frac{h^2(\tau_0)K^2(\tau_0,\tau_1)\dots K^2(\tau_{l-1},\tau_l)c^2_{i_l}(\tau_l)}{q^0_{i_0}q_{i_0,i_1}(t)p_{i_0,i_1}(\tau_0,\tau_1)\dots q_{i_{l-1},i_l}(\tau_{l-1})p_{i_{l-1},i_l}(\tau_{l-1},\tau_l)g_{i_l}(\tau_l)}d\tau_l\dots d\tau_1 \tag{2.14}$$

В предположении, что указанный ряд сходится, выражение для дисперсии будет иметь вид

$$\mathbb{D}\mathcal{J}(\omega_l) = \left(\frac{h^2}{q^0}, \psi(t)\right) - (h, x(t))^2,$$

где $\psi(t) = (\psi_1(t), \ldots, \psi_n(t))^T$ – итерационное решение системы уравнений

$$\psi(t) = \int_{t_0}^{\infty} \frac{K^2(t,\tau)}{\mathcal{Q}(t)\mathcal{P}(t,\tau)}\psi(\tau)d\tau + \frac{c^2(t)}{g(t)},$$

умножение и деление матрицы на матрицу и вектора на вектор выполняются здесь поэлементно.

Как видим, переход от одного линейного интегрального уравнения к системе линейных интегральных уравнений хоть и усложнил схему, но не значительно. Иначе дело обстоит с общим случаем.

2.2. Случай полиномиальной нелинейности

Итак, теперь приступим к рассмотрению общего случая системы уравнений (2.6). При решении линейных систем интегральных уравнений оценки метода Монте-Карло строились на линейных траекториях цепи Маркова. В случае же с системами вида (2.6) оценки строятся на ветвящихся траекториях. При этом процесс, порождающий такие траектории, удобно интерпретировать как процесс эволюции популяции частиц. Изначально в популяции имеется одна частица, продолжительность жизни такой частицы является случайной величиной. В конце жизни каждая частица производит случайное количество потомков. Качественный и количественный состав потомков определяется рассматриваемой системой уравнений.

Подробно опишем связь между системой (2.6) и процессом рождения/-гибели частиц. Пусть имеется n типов частиц

$$T_1, T_2, \ldots, T_n,$$

каждая из которых связана с определенным уравнением системы (2.6) – частица i-ого типа связана с i-ым уравнением. Если в популяции есть частица i-ого типа, то её потомки и время жизни определяются членами i-ого уравнения. Член уравнения вида

$$\int_{t_0}^{t} K_i^{\alpha}(t,\tau)x_1^{\alpha_1}\dots x_n^{\alpha_n}d\tau$$

предполагает рождение α_1 частиц первого типа, α_2 частиц второго типа и так далее вплоть до α_n частиц n-ого типа. Так, например, свободный член (все α_i равны нулю) подразумевает гибель частицы без рождения потомков. Конкретный же член i-ого уравнения, определяющий дальнейший сценарий, выбирается случайным образом, так или иначе согласованным с коэффициентами $K_i^{\alpha}(t,\tau)$.

Подобные ветвящиеся процессы описываются, например, в [29] и [37]. Отличительной особенностью рассматриваемого ветвящегося процесса будет то, что время имеет обратный ход. Другими словами, частица, родившаяся в момент времени $t > t_0$, погибнет в момент времени из промежутка $[t, t_0]$.

Обозначим за $P_k^{\alpha}(t)$ вероятность того, что частица типа T_k, родившаяся в момент времени t, после гибели превратится в совокупность частиц $\alpha = (\alpha_1, \dots, \alpha_n)$. Предполагается, что для вероятностей $P_k^{\alpha}(t)$ выполнено условие нормировки для каждого $t > t_0$

$$\sum_{\alpha} P_k^{\alpha}(t) = 1, k = 1, \dots, n.$$

Плотность распределения времени гибели частицы типа T_k, родившейся в момент времени t и превращающуюся после гибели в совокупность частиц α, обозначим за $p_k^{\alpha}(t,\tau)$. Для плотностей $p_k^{\alpha}(t,\tau)$ предполагается выполнение

$$\int_{t_0}^{t} p_k^{\alpha}(t,\tau)d\tau = 1, k = 1, \dots, n$$

при любом α.

Если решение системы (2.6) или некоторый функционал от него ищется в момент времени $t > t_0$, то пространство элементарных событий Ω_t состоит из ветвящихся траекторий следующего вида. В момент t в популяции единственная частица типа T_{k_0}, где k_0 является случайной величиной с распределением $q^0 = (q_1^0, \ldots, q_n^0)$. Далее для каждой частицы из популяции по её типу T_k и времени рождения τ_0 определяется совокупность частиц α, в которую она перейдёт после гибели, по распределению $\{P_k^\alpha(\tau_0)\}_\alpha$. Предполагается, что все частицы из α рождаются в один момент времени, который задаётся плотностью распределения $p_k^\alpha(\tau_0, \tau_1)$. Если частица погибает без рождения потомков, то рассматривать время их рождения не имеет смысла, поэтому плотности $p_k^{(0,\ldots,0)}(t, \tau)$ не используются, так же, как и плотности $p_k^\alpha(t, \tau)$ для таких k, α и t, при которых $P_k^\alpha(t) \equiv 0$.

В дальнейшем будем считать, что вероятности $P_k^\alpha(t)$ определены на $[0, \infty)$, а плотности $p_k^\alpha(t, \tau)$ определены на $[0, \infty)^2$ для всех $k = 1, \ldots, n$ и α. Будем также предполагать, что количество $P_k^\alpha(t)$ тождественно не равных нулю конечно.

Траекторию ветвящегося процесса удобно представлять в виде ориентированного графа, а точнее сказать дерева, в котором вершинами являются частицы, а направленные ребра обозначают процесс превращения старой частицы в совокупность новых. Чтобы однозначно определить траекторию ветвящегося процесса, необходимо особым образом пронумеровать/обозначить вершины дерева, соответствующего этой траектории.

Приведем такие обозначения. Начальная частица типа T_k будет обозначаться 1^k, её i-ый потомок типа T_{k_1} будет обозначаться $(1^k i^{k_1})$, и в общем случае обозначение

$$(1^k i_1^{k_1} \ldots i_n^{k_n})$$

будет соответствовать i_n-ому потомку типа T_{k_n} ... i_1-ого потомка типа T_{k_1} начальной частицы типа T_k. У каждой частицы из дерева помимо её номера есть также и время её рождения. Для частицы с номером $(1^k i_1^{k_1} \dots i_n^{k_n})$ время её рождения будем обозначать $t_{1^k i_1^{k_1} \dots i_n^{k_n}}$. Состояние, в которое перейдет частица $(1^k i_1^{k_1} \dots i_n^{k_n})$, будем обозначать $\alpha(1^k i_1^{k_1} \dots i_n^{k_n})$.

Проиллюстрируем такой подход к нумерации вершин на конкретном примере дерева, в котором два типа вершин: первый соответствует белым вершинам, а второй - серым (рисунок 2.1).

Имея такую нумерацию вершин, можно представить дерево в виде последовательности

$$(\alpha(1^k), t_{1^k}; \alpha(1^k 1^{k_1}), t_{1^k 1^{k_1}}; \alpha(1^k 2^{k_1}), t_{1^k 2^{k_1}}; \dots).$$

Множество всевозможных последовательностей такого вида и составляет Ω_t. Индекс t здесь свидетельствует о том, с какого момента времени начинаются траектории.

Для каждого дерева $\omega \in \Omega_t$ определим последовательность множеств $I_0(\omega), I_1(\omega), I_2(\omega), \dots$, называемых поколениями. Нулевое поколение $I_0(\omega)$ состоит из одного элемента – корневой вершины дерева 1^k. Поколение $I_j(\omega)$ состоит из последовательностей $(1^k i_1^{k_1} \dots i_j^{k_j})$ таких, что $(1^k i_1^{k_1} \dots i_{j-1}^{k_{j-1}}) \in I_{j-1}(\omega)$ и $i_j \le \alpha_{k_j}(1^k i_1^{k_1} \dots i_{j-1}^{k_{j-1}})$.

Дерево на рисунке 2.1 имеет три поколения: $I_0(\omega) = \{1^1\}$, $I_1(\omega) = \{1^1 1^1, 1^1 1^2\}$, $I_2(\omega) = \{1^1 1^1 1^1, 1^1 1^1 2^1, 1^1 1^2, 1^1 1^2 1^2, 1^1 1^2 2^2\}$, $I_3(\omega) = \{1^1 1^2, 1^2 1^1\}$.

Из определения поколений следует, что если $I_j(\omega)$ пусто для некоторого j, то все поколения с индексом больше j также будут пустыми. Объединение поколений $\cup_{j=1}^{\infty} I_j(w)$ будем называть семьей и обозначать $I(\omega)$. В общем случае $I(\omega)$ может быть и бесконечной, это означает, что дерево ω никогда не оборвётся, то есть в каждом поколение $I_j(\omega), j = 1, 2, \dots$ найдётся хотя бы

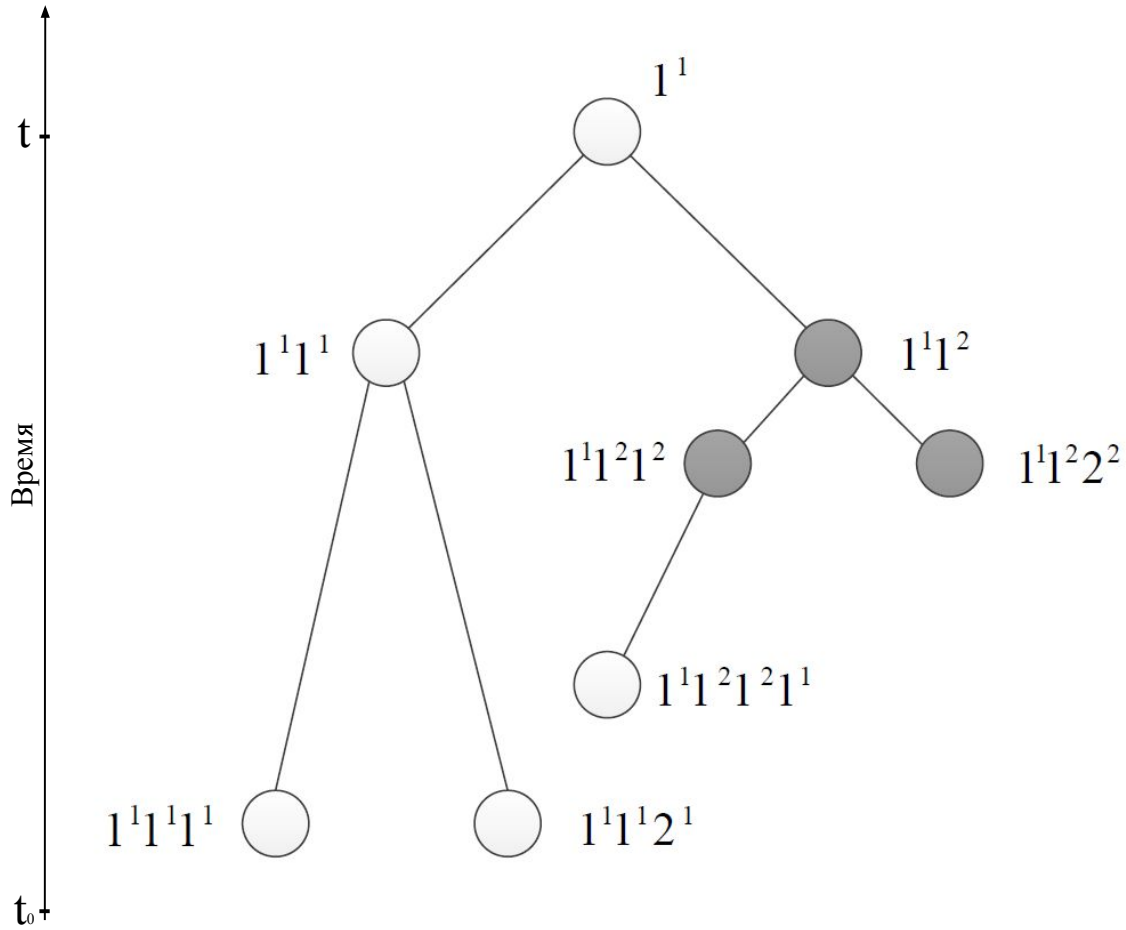

Рис. 2.1. Нумерация вершин дерева

один элемент.

Рассмотрим множества $\Omega_t^j \subset \Omega_t$, состоящие из деревьев, имеющих ровно j поколений.

$$\Omega_t^j = \{\omega \in \Omega_t | I_j(\omega) \neq \emptyset \,\&\, I_{j+1}(\omega) = \emptyset\},$$

Вместе с тем введем множество $\Omega_t^\infty \subset \Omega_t$, состоящее из деревьев с бесконечным количеством поколений. Объединение множеств $\cup_{j=1}^\infty \Omega_t^j \cup \Omega_t^\infty$ даёт всё множество Ω_t.

В качестве σ-алгебры $\mathcal{F}_t$ возьмем минимальную σ-алгебру, содержащую подмножества $\Omega_t^j, j = 1, 2, \ldots$ и Ω_t^∞.

Введем вероятностную меру на $\mathcal{F}_t$. Сначала рассмотрим деревья $\omega \in \Omega_t^j$.

Такие траектории распределены следующим образом

$$q_0^k P_k^\alpha(t) p_k^\alpha(t,\tau) \prod_{i_1 \in I_1(\omega)} P_{i_1}^{\alpha(i_1)}(t_{i_1}) p_{i_1}^{\alpha(i_1)}(t_{i_1},\tau_{i_1}) \cdots \prod_{i_j \in I_j(\omega)} P_{i_j}^{(0,\ldots,0)}(t_{i_j}). \tag{2.15}$$

Если теперь провести суммирование по всем деревьям с конечным числом поколений и всем возможным $\alpha(i), t_i$, то получится выражение

$$\sum_{j=0}^{\infty} \sum_{k_0=1}^{n} q_{k_0} \int_{t_0}^{t} \sum_{\alpha} P_{k_0}^\alpha(t) p_{k_0}^\alpha(t,\tau) \prod_{i_1 \in I_1(\omega)} \int_{t_0}^{t_{i_1}} \sum_{\alpha(i_1)} P_{i_1}^{\alpha(i_1)}(t_{i_1}) p_{i_1}^{\alpha(i_1)}(t_{i_1},\tau_{i_1}) \ldots \cdots \prod_{i_j \in I_j(\omega)} P_{i_j}^{(0,\ldots,0)}(t_{i_j}) d\tau_{i_{j-1}} \ldots d\tau_{i_1} d\tau \tag{2.16}$$

или же, если записать это в виде скалярного произведения,

$$(q^0, Q(t)),$$

$$Q(t) = (Q_1(t), \ldots, Q_n(t)),$$

где компоненты вектора $Q(t)$ имеют вид

$$Q_k(t) = \sum_{j=0}^{\infty} \int_{t_0}^{t} \sum_{\alpha} P_k^\alpha(t) p_k^\alpha(t,\tau) \prod_{i_1 \in I_1(\omega)} \int_{t_0}^{t_{i_1}} \sum_{\alpha(i_1)} P_{i_1}^{\alpha(i_1)}(t_{i_1}) p_{i_1}^{\alpha(i_1)}(t_{i_1},\tau_{i_1}) \ldots \cdots \prod_{i_j \in I_j(\omega)} P_{i_j}^{(0,\ldots,0)}(t_{i_j}) d\tau_{i_{j-1}} \ldots d\tau_{i_1} d\tau.$$

Если показать, что сумма ряда (2.16) равна единице, то почти все деревья будут конечными, а это наиболее интересный случай с практической точки зрения.

Рассмотрим вектор частичных сумм рядов из $Q(t)$ до j равного l и обозначим его за $Q(t,l)$. Как нетрудно видеть

$$Q(t,0) = (P_1^{(0,\ldots,0)}(t), \ldots, P_n^{(0,\ldots,0)}(t)). \tag{2.17}$$

Покажем, что $Q(t,l+1)$ и $Q(t,l)$ связаны соотношением

$$Q(t,l+1) = \int_{t_0}^{t} \sum_{\alpha} P_k^\alpha(t) p_k^\alpha(t,\tau) Q^\alpha(\tau,l) d\tau, l = 0,1,\ldots. \tag{2.18}$$

Для l равного нулю это показывается довольно просто

$$Q_k(t,1) = \int_{t_0}^{t} \sum_{\alpha} P_k^{\alpha}(t) p_k^{\alpha}(t,\tau) \prod_{i_1 \in I_1(\omega)} P_{i_1}^{(0,\ldots,0)}(t_{i_1}) d\tau =$$

$$= \int_{t_0}^{t} \sum_{\alpha} P_k^{\alpha}(t) p_k^{\alpha}(t,\tau) \prod_{i_1=1}^{n} \prod_{r=1}^{\alpha_i} P_{i_1}^{(0,\ldots,0)}(\tau) d\tau =$$

$$= \int_{t_0}^{t} \sum_{\alpha} P_k^{\alpha}(t) p_k^{\alpha}(t,\tau) Q^{\alpha}(\tau,0) d\tau.$$

Случай произвольного l требует дополнительных соображений. В данном случае полезно отметить тот факт, что для дерева $\omega \in \Omega_t$ с количеством поколений $l+1$, у которого $I_1(\omega) \neq \emptyset$, все частицы первого поколения рождаются в некоторый момент τ и являются в свою очередь корневыми вершинами деревьев из Ω_τ с количеством поколений, не превосходящим l (рисунок 2.2). Воспользуемся этим для доказательства (2.18).

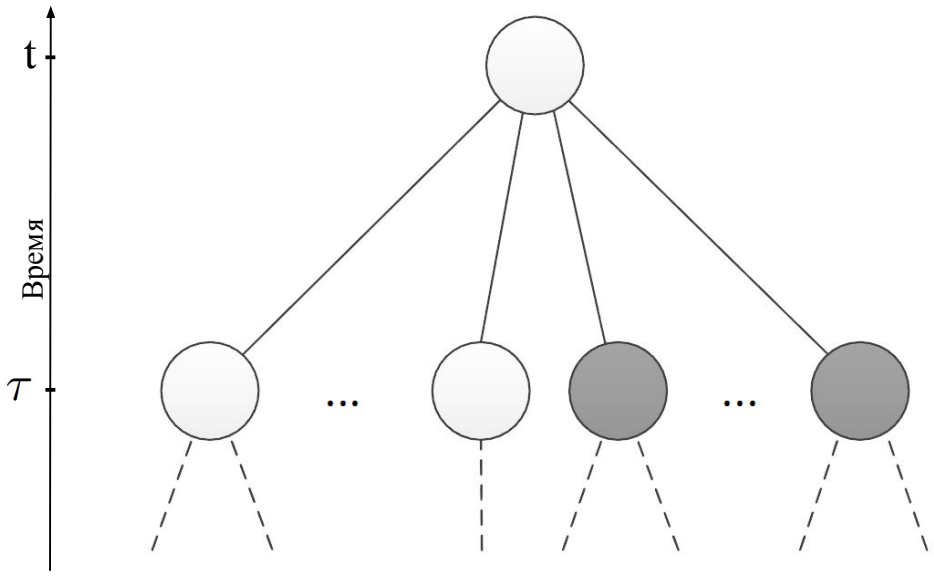

Рис. 2.2. Первое поколение

$$Q_k(t,l+1)=\sum_{j=1}^{l+1}\int_{t_0}^{t}\sum_{\alpha}P_k^{\alpha}(t)p_k^{\alpha}(t,\tau)\prod_{i_1\in I_1(\omega)}\int_{t_0}^{t_{i_1}}\sum_{\alpha(i_1)}P_{i_1}^{\alpha(i_1)}(t_{i_1})p_{i_1}^{\alpha(i_1)}(t_{i_1},\tau_{i_1})\dots$$

$$\cdots\prod_{i_j\in I_j(\omega)}P_{i_j}^{(0,\dots,0)}(t_{i_j})d\tau_{i_{j-1}}\dots d\tau_{i_1}d\tau=$$

$$=\int_{t_0}^{t}\sum_{\alpha}P_k^{\alpha}(t)p_k^{\alpha}(t,\tau)\prod_{i_1=1}^{n}\prod_{r=1}^{\alpha_{i_1}}\sum_{j=1}^{l}\int_{t_0}^{\tau}\sum_{\alpha(i_1)}P_{i_1}^{\alpha(i_1)}(t_{i_1})p_{i_1}^{\alpha(i_1)}(t_{i_1},\tau_{i_1})\dots$$

$$\cdots\prod_{i_j\in I_j(\omega)}P_{i_j}^{(0,\dots,0)}(t_{i_j})d\tau_{i_{j-1}}\dots d\tau_{i_1}d\tau=$$

$$=\int_{t_0}^{t}\sum_{\alpha}P_k^{\alpha}(t)p_k^{\alpha}(t,\tau)Q^{\alpha}(\tau,l)d\tau.$$

Таким образом (2.18) доказано. Нетрудно видеть, что (2.18) являются простыми итерациями для системы интегральных уравнений

$$Q_k(t)=\int_{t_0}^{t}\sum_{\alpha}P_k^{\alpha}(t)p_k^{\alpha}(t,\tau)Q^{\alpha}(\tau)d\tau,k=1,\dots,n \tag{2.19}$$

с начальным условием $Q(t_0)=(1,\dots,1)$.

Если взять $Q_k(t)\equiv 1,k=1,\dots,n$, то они в силу свойств вероятностей P_k^{α} и плотностей $p_k^{\alpha}(t,\tau)$ будут удовлетворять системе уравнений (2.19) и являться его единственным решением. Следовательно, если итерации (2.18) сходятся при некотором $t\geq t_0$, то сходятся они к единственному решению системы уравнений (2.19). В этом случае вектор-функция $Q(t)\equiv(1,\dots,1)$, откуда следует, что сумма (2.16) будет равна единице и почти все траектории конечны.

Далее будем предполагать, что для рассматриваемых t вероятности $P_k^{\alpha}(t)$ и плотности $p_k^{\alpha}(t,\tau)$ выбраны таким образом, что итерационный процесс (2.18), начинающийся с (2.17), сходится к $(1,\dots,1)$.

Вернемся к поиску решения системы уравнений (2.6) и рассмотрим связанные с ней итерации

$$x_i(t,l+1)=\int\limits_{t_0}^{t}\sum_{\alpha}K_i^{\alpha}(t,\tau)x^{\alpha}(\tau,l)d\tau+K_i^0(t),\ i=1,\ldots,n,l=0,1,\ldots, \quad (2.20)$$

начинающиеся с $x(t,0)\equiv x_0$.

Наряду с (2.20) будем рассматривать мажорантный итерационный процесс

$$y_i(t,l+1)=\int\limits_{t_0}^{t}\sum_{\alpha}|K_i^{\alpha}(t,\tau)|y^{\alpha}(\tau,l)d\tau+|K_i^0(t)|,\ i=1,\ldots,n,l=0,1,\ldots, \quad (2.21)$$

начинающиеся с $y(t,0)\equiv|x_0|$.

Существует такое t', что для любого $t_0\le t\le t'$ итерации (2.21) сходятся, а вместе с ними сходится итерационный процесс (2.20), при этом сходится он к решению системы (2.6). Далее будем предполагать, что $t_0\le t\le t'$.

На деревьях $\omega\in\Omega_t$ построим оценку метода Монте-Карло решения системы (2.6). Оценка по поглощению для траектории $\omega\in\Omega_t^j$ будет иметь вид

$$\mathcal{J}(\omega)=\frac{h_k}{q_0^k}\frac{K_k^{\alpha}(t,\tau)}{P_k^{\alpha}(t)p_k^{\alpha}(t,\tau)}\prod_{i_1\in I_1(\omega)}\frac{K_{i_1}^{\alpha(i_1)}(t_{i_1},\tau_{i_1})}{P_{i_1}^{\alpha(i_1)}(t_{i_1})p_{i_1}^{\alpha(i_1)}(t_{i_1},\tau_{i_1})}\cdots\prod_{i_j\in I_j(\omega)}\frac{K_{i_j}^{(0,\ldots,0)}(t_{i_j})}{P_{i_j}^{(0,\ldots,0)}(t_{i_j})}. \quad (2.22)$$

Теперь может быть сформулирована

Теорема 10. *Пусть выполнены условия согласования*

- $q_i^0>0$ *для тех* t, *для которых* $h_i\ne0$, $i=1,\ldots,n$;
- $P_k^{\alpha}(t)>0$ *для тех* t, *для которых* $\int_{t_0}^{t}K_k^{\alpha}(t,\tau)d\tau\ne0$, *для всех* α *и* $k=1,\ldots,n$;
- $p_k^{\alpha}(t,\tau)>0$ *для тех* t,τ, *для которых* $K_k^{\alpha}(t,\tau)\ne0$, *для всех* α *и* $k=1,\ldots,n$;

и сходится мажорантный процесс (2.21), тогда (2.22) является несмещенной оценкой скалярного произведения $(q^0, x^*(t))$, где $x^*(t)$ – решение уравнения (2.6).

Доказательство. Выражение для $\mathbb{E}\mathcal{J}(\omega)$ имеет вид

$$\mathbb{E}\mathcal{J}(\omega) = \sum_{j=1}^{n} \sum_{k} \sum_{\alpha} \cdots \sum_{\alpha(i_{j-1})} \int_{t_0}^{t} \int_{t_0}^{t_{i_1}} \cdots \int_{t_0}^{t_{i_{j-1}}} \mathcal{J}(\omega)\mathcal{P}(\omega) d\tau_{i_{j-1}} \ldots d\tau.$$

Знаменатель в $\mathcal{J}(\omega)$ есть не что иное, как $\mathcal{P}(\omega)$, поэтому условия согласования дают возможность выполнить сокращение. Сходимость же мажорантного процесса позволяет делать перестановки знаков сумм и интегрирования. После преобразований получим следующее

$$\mathbb{E}\mathcal{J}(\omega) = \sum_{j=1}^{\infty} \sum_{k_0=1}^{n} q_{k_0} \int_{t_0}^{t} \sum_{\alpha} K_{k_0}^{\alpha}(t,\tau) \prod_{i_1 \in I_1(\omega)} \int_{t_0}^{t_{i_1}} \sum_{\alpha(i_1)} K_{i_1}^{\alpha(i_1)}(t_{i_1}, \tau_{i_1}) \ldots$$
$$\cdots \prod_{i_j \in I_j(\omega)} K_{i_j}^{(0,\ldots,0)}(t_{i_j}) d\tau_{i_{j-1}} \ldots d\tau_{i_1} d\tau = (q^0, x^*(t)),$$

и теорема доказана. □

Выражение для второго момента оценки (2.22) имеет следующий вид

$$\mathbb{E}(\mathcal{J}(\omega))^2 = \left(\frac{h(t)}{q^0}, \psi(t) \right),$$

где операция деления вектора на вектор выполняется поэлементно, а $\psi(t)$ – итерационное решение системы уравнений

$$\psi_i(t) = \int_{t_0}^{t} \sum_{\alpha} \frac{(K_i^{\alpha}(t,\tau))^2}{P_i^{\alpha}(\tau) p_i^{\alpha}(t,\tau)} \psi^{\alpha}(\tau) d\tau, \; i = 1, \ldots, n, \tag{2.23}$$

которое, вообще говоря, может и не существовать. Дисперсия оценки (2.22) при условии конечности второго момента имеет вид

$$\mathbb{D}\mathcal{J}(\omega) = \left(\frac{h(t)}{q^0}, \psi(t) \right) - (q^0, x^*(t))^2.$$

Таким образом оценка (2.22) может быть использована для таких $t > t_0$, при которых сходится мажорантный итерационный процесс (2.21) и сходится итерационный процесс, связанный с уравнением (2.23). Предположим, удалось найти некоторое t', такое что указанные итерационные процессы сходятся для $t_0 \leq t \leq t'$, а следовательно применима предложенная оценка метода Монте-Карло.

Возникает вопрос - как же найти решение (2.6) для $t > t'$? В этом случае предлагается использовать схему с запоминанием. На первом этапе при помощи метода Монте-Карло находится $\tilde{x}$ – оценка $x(t')$, а затем система (2.1) решается с новым начальным условием $x(t') = \tilde{x}$. Далее процедура повторяется. При таком подходе каждая новая система уравнений решается с начальными условиями, содержащими случайную ошибку, а это означает, что есть риск столкнуться с явлением стохастической неустойчивости алгоритма метода Монте-Карло.

В дальнейших рассуждениях будем считать, что исходная система дифференциальных уравнений (2.1) заменена системой интегральных уравнений (2.3). Пусть решение задачи ищется в моменты времени $t_k, k = 0, 1, \ldots$ такие, что

$$t_0 < t_1 < t_2 \ldots .$$

Точное решение задачи обозначим за $x^*(t)$. Для него выполнено

$$x^*(t) = \int_{t_0}^{t} f(\tau, x^*(\tau))d\tau + x^*(t_0), \tag{2.24}$$

где $x^*(t_0)$ – заданное начальное значение.

Последовательный метод Монте-Карло предполагает в данной ситуации, что по известной оценке решения в момент времени t_{k-1}, начиная с $k = 1$, ищется оценка решения в момент времени t_k, после чего k увеличивается на единицу и процедура повторяется. Такой процесс в предположении, что

при каждом t_k находится точное решение, соответствует последовательному решению интегральных уравнений вида

$$x(t_k) = \int_{t_{k-1}}^{t_k} f(\tau, x(\tau))d\tau + x(t_{k-1}), k = 1, 2, \ldots, \tag{2.25}$$

где $x(t_0) = x^*(t_0)$. Легко убедиться в том, что $x^*(t)$ удовлетворяет уравнениям (2.25).

В действительности же из-за вычислительных ошибок будут решаться системы

$$x_k(t_k) = \int_{t_{k-1}}^{t_k} f(\tau, x_k(\tau))d\tau + \tilde{x}_{k-1}(t_{k-1}), k = 1, 2, \ldots, \tag{2.26}$$

где $\tilde{x}_{k-1}(t_{k-1})$ – оценка решения для момента времени t_{k-1}, которую можно представить в виде

$$\tilde{x}_{k-1}(t_{k-1}) = x_{k-1}(t_{k-1}) + \delta_{k-1},$$

где $x_{k-1}(t_{k-1})$ – точное решение (2.26) для момента времени t_{k-1}, а δ_{k-1} – вычислительная ошибка, являющейся суммой детерминированной ошибки $\delta_{k-1}^{(d)}$, возникающей, например, из-за ошибок округления, и случайной составляющей $\delta_{k-1}^{(r)}$, обусловленной использованием метода Монте-Карло. Предполагается, что используются несмещенные оценки, поэтому $\mathbb{E}\delta_{k-1}^{(r)} = 0$.

Обозначим за $\Delta_k(t_k)$ разность между точным решением (2.26) и $x^*(t_k)$

$$\Delta_k(t_k) = x_k(t_k) - x^*(t_k).$$

Будем так же считать, что для $t_{k-1} \leq t \leq t_k$ имеет место равенство

$$\Delta_k(t) = x_k(t) - x^*(t).$$

Вычтем теперь из системы уравнений (2.26) систему (2.25), подставив в нее x^*

$$\Delta_k(t_k) = \int_{t_{k-1}}^{t_k} f(\tau, x_k(\tau)) - f(\tau, x^*(\tau))d\tau + \Delta_k(t_{k-1}) + \delta_{k-1}. \tag{2.27}$$

Рассмотрим отдельно разность $f(\tau, x_k(\tau)) - f(\tau, x^*(\tau))$. При сделанных предположениях относительно вектор-функции $f(t,x)$ для нее существует производная по Гато $f_x(t,x)$, и указанную разность можно представить в виде интеграла

$$f(\tau, x_k(\tau)) - f(\tau, x^*(\tau)) = \int_0^1 f_x(\tau, \omega x_k(\tau) + (1-\omega)x^*(\tau))\Delta_k(\tau)d\omega =$$

$$= \int_0^1 f_x(\tau, x^*(\tau) + \omega\Delta_k(\tau))d\omega\Delta_k(\tau).$$

Теперь сделаем предположение относительно интеграла

$$\int_0^1 f_x(\tau, x^*(\tau) + \omega\Delta_k(\tau))d\omega.$$

Будем считать, что для выбранных интервалов времени $[t_{k-1}, t_k]$ его можно довольно точно приблизить по формуле прямоугольников в нуле, то есть

$$\int_0^1 f_x(\tau, x^*(\tau) + \omega\Delta_k(\tau))d\omega \approx f_x(\tau, x^*(\tau)), t_{k-1} \le \tau \le t_k. \tag{2.28}$$

После такой замены уравнение (2.27) становится линейным относительно $\Delta_k(t_k)$, и в этом случае решение системы (2.27) будет выглядеть следующим образом

$$\Delta_k(t_k) = \exp\left\{\int_{t_{k-1}}^{t_k} f_x(\tau, x^*(\tau))d\tau\right\}(\Delta_k(t_{k-1}) + \delta_{k-1}). \tag{2.29}$$

Обозначим теперь за $\varepsilon(t_k)$ – полную ошибку в момент времени t_k, которая состоит из $\Delta_k(t_k)$ и δ_k. Используя (2.29), легко проверяется, что

$$\varepsilon(t_k) = \exp\left\{\int_{t_{k-1}}^{t_k} f_x(\tau, x^*(\tau))d\tau\right\}\varepsilon(t_{k-1}) + \delta_k. \tag{2.30}$$

Представим теперь $\Delta_k(t_k)$ и $\varepsilon(t_k)$ в виде суммы двух компонент случайной и детерминированной

$$\Delta_k(t_k) = \Delta_k^{(d)}(t_k) + \Delta_k^{(r)}(t_k)$$

$$\varepsilon(t_k) = \varepsilon^{(d)}(t_k) + \varepsilon^{(r)}(t_k),$$

при этом $\mathbb{E}\Delta_k^{(r)}(t_k) = \mathbb{E}\varepsilon^{(r)}(t_k) = 0$.

Возьмем математическое ожидание от левой и правой частей уравнения (2.30) и получим уравнение

$$\varepsilon^{(d)}(t_k) = \exp\left\{\int\limits_{t_{k-1}}^{t_k} f_x(\tau, x^*(\tau))d\tau\right\}\varepsilon^{(d)}(t_{k-1}) + \delta_k^{(d)}. \tag{2.31}$$

Вычтем из (2.30) уравнение (2.31) и получим связь случайной ошибки в момент t_k со случайной ошибкой в момент t_{k-1}

$$\varepsilon^{(r)}(t_k) = \exp\left\{\int\limits_{t_{k-1}}^{t_k} f_x(\tau, x^*(\tau))d\tau\right\}\varepsilon^{(r)}(t_{k-1}) + \delta_k^{(r)}. \tag{2.32}$$

Таким образом получилось два уравнения, описывающих поведение ошибок – одно для детерминированных ошибок, другое для стохастических. Введем сокращенное обозначение

$$A_k = \exp\left\{\int\limits_{t_{k-1}}^{t_k} f_x(\tau, x^*(\tau))d\tau\right\}.$$

Тогда (2.31) и (2.32) примут вид

$$\varepsilon^{(d)}(t_k) = A_k\varepsilon^{(d)}(t_{k-1}) + \delta_k^{(d)}, \tag{2.33}$$

$$\varepsilon^{(r)}(t_k) = A_k\varepsilon^{(r)}(t_{k-1}) + \delta_k^{(r)}. \tag{2.34}$$

Транспонируем векторы в левой и правой частях (2.34)

$$(\varepsilon^{(r)}(t_k))^T = (\varepsilon^{(r)}(t_{k-1}))^T A_k^T + (\delta_k^{(r)})^T. \tag{2.35}$$

Чтобы оценить ковариацию $\varepsilon^{(r)}(t_k)$, умножим (2.34) на (2.35) и возьмем математическое ожидание от левой и правой частей получившегося равенства, учитывая, что математическое ожидание части слагаемых в правой части окажется равным нулю

$$cov\varepsilon^{(r)}(t_k) = A_k cov\varepsilon^{(r)}(t_k)A_k^T + cov\delta_k^{(r)}, k = 1, 2, \ldots. \tag{2.36}$$

Теорема 11. *Если для всех натуральных k выполнены неравенства $\|A_k\| \le q_1 < 1$, $\|A_k\|\|A_k^T\| \le q_2 < 1$, $\|\delta_k^{(d)}\| < m_1 < \infty$ и $\|cov\delta_k^{(r)}\| < m_2 < \infty$, то верны следующие оценки*

$$\lim_{k\to\infty} \|\varepsilon^{(d)}(t_k)\| \le \frac{m_1}{1-q_1},$$

$$\lim_{k\to\infty} \|cov\varepsilon^{(r)}(t_k)\| \le \frac{m_2}{1-q_2}.$$

Доказательство. Используя выражения (2.33) и (2.36) выпишем неравенства для норм $\varepsilon^{(d)}(t_k)$ и $cov\varepsilon^{(r)}(t_k)$

$$\|\varepsilon^{(d)}(t_k)\| \le \|A_k\|\|\varepsilon^{(d)}(t_{k-1})\| + \|\delta_k^{(d)}\| \le \cdots \le q_1^k\|\varepsilon^{(d)}(t_0)\| + \sum_{i=1}^{k-1} q_1^i m_1.$$

$$\|cov\varepsilon^{(r)}(t_k)\| \le \|A_k\|\|cov\varepsilon^{(r)}(t_{k-1})\|\|A_k^T\| + \|cov\delta_k^{(d)}\| \le \ldots$$
$$\ldots \le q_2^k\|cov\varepsilon^{(d)}(t_0)\| + \sum_{i=1}^{k-1} q_2^i m_2.$$

После перехода к пределу при $k \to \infty$ в левых и правых частях неравенств получим утверждение теоремы. □

Таким образом, при сделанном предположении (2.28) и выполнении условий теоремы 11 последовательный алгоритм Монте-Карло будет стохастически устойчивым.

Одним из преимуществ алгоритмов метода Монте-Карло является простота их реализации. Однако усложнение алгоритма для деревьев по сравнению с линейным случаем очевидно, и рассмотрение алгоритма и особенностей его эффективной реализации становится необходимым.

Моделирование ветвящейся траектории, как нетрудно видеть, носит рекуррентный характер. Всё начинается с рождения корневой частицы. Для неё определяется состав потомков и время гибели. Затем для каждого из потомков выполняется аналогичная операция, далее то же следует для потомков потомков и так далее. Можно было бы организовать алгоритм в виде рекуррентного вызова функции $func(T_i, \tau_i)$, которая бы по времени рождения и типу частицы моделировала время гибели и состав потомков, а также рассчитывала оценку (2.22), делая рекуррентный вызов

$$func(T_i, \tau_i) = \frac{K_i^\alpha(\tau_i, \tau_{i+1})}{P_i^\alpha(\tau_i) p_i^\alpha(\tau_i, \tau_{i+1})} \overbrace{func(T_1, \tau_{i+1}) \dots func(T_1, \tau_{i+1})}^{\alpha_1} \dots$$
$$\dots \underbrace{func(T_n, \tau_{i+1}) \dots func(T_n, \tau_{i+1})}_{\alpha_n},$$

в случае непустого состава потомков, и, обрывая рекурсию по формуле

$$func(T_i, \tau_i) = \frac{K_i^0(\tau_i)}{P_i^0(\tau_i)},$$

в случае гибели частицы без рождения потомков. Однако даже небольшое количество ветвлений при сравнительно малой вероятности поглощения может привести к переполнению стека вызова функций.

Чтобы избежать такого эффекта предлагается следующий подход. В алгоритме будет использоваться структура, описывающая вершину в моделируемом дереве. Эта структура хранит информацию о типе частице, времени её рождения, времени гибели, составе потомков и ссылку на предка. В ходе алгоритма будут моделироваться потомки, соответствующие им вершины будут добавляться в память, а после их обработки удаляться из неё. Помимо

этого будет использоваться указатель на текущую обрабатываемую вершину. Оценку метода Монте-Карло обозначим за W. Алгоритм моделирования дерева и расчета по нему оценки можно записать в следующем виде:

Шаг 0 . Инициализация. По распределение q^0 моделируется тип исходной частицы k. $W := h_k/q_k^0$. Создаётся вершина с типом T_k временем рождения t и неопределенными временем гибели, составом потомков и без ссылки на предка. Указатель текущей вершины указывает на корневую вершину.

Шаг 1 . Если для текущей вершины не определён состав потомков, то переходим к шагу 2, иначе переходим к шагу 3.

Шаг 2 . Для текущей вершины по распределению $P_i^\alpha(\tau_i)$ моделируется α.

Если $\alpha = (0, \ldots, 0)$, то

$$W := W\frac{K_i^0(\tau_i)}{P_i^0(\tau_i)},$$

вершина считается обработанной и удаляется из памяти, а указатель перемещается на предка, если он есть, текущей вершины, после чего выполняется переход к шагу 1. Отсутствие предка означает, что мы обработали все вершины с их потомками и алгоритм завершается.

Если $\alpha \neq (0, \ldots, 0)$, то по распределению $p_i^\alpha(\tau_i, \tau)$ находится время τ_{i+1} гибели текущей частицы. Рассчитывается оценка

$$W := W\frac{K_i^\alpha(\tau_i, \tau_{i+1})}{P_i^\alpha(\tau_i)p_i^\alpha(\tau_i, \tau_{i+1})}.$$

Выполняется переход к шагу 1.

Шаг 3 . Если для текущий вершины есть необработанные потомки, то из них выбирается первый. По его типу создаётся вершина. Количество потомков, данного типа, которые необходимо обработать для текущей верши-

ны уменьшается на 1. Указатель перемещается на созданную вершину. Выполняется переход к шагу 1.

Если все потомки текущей вершины обработаны, то указатель переходит на его предка, а сама вершина удаляется, затем выполняется переход к шагу 1. В случае его отсутствия все вершины обработаны и алгоритм завершается.

Основным преимуществом такого алгоритма является тот факт, что для деревьев $\omega \in \Omega_t^j$ количество единовременно хранимых в памяти вершин вне зависимости от степени ветвления процесса не превосходит j. Простота реализации и естественный параллелизм метода Монте-Карло делают алгоритм эффективным средством вычисления оценок (2.22) на многопроцессорных системах.

2.3. Общий случай

До этого момента обсуждалось решения интегральных уравнений с полиномиальной нелинейностью. Перейдем к рассмотрению уравнений вида (2.3) с произвольной нелинейностью. Его предлагается заменить на приближенное

$$x_i(t) = \int_{t_0}^{t} \sum_{\alpha=(\alpha_1,\ldots,\alpha_n)} K_i^{\alpha}(t,\tau) x_1^{\alpha_1}(\tau) \ldots x_n^{\alpha_n}(\tau) d\tau + K_i^{(0,\ldots,0)}(t),\ i = 1, \ldots, n.$$

Приближенное уравнение можно решать методом Монте-Карло, способами изложенными в предыдущих пунктах.

Как известно, метод Монте-Карло позволяет находить оценку решения для задачи

$$y = \tilde{G}y, \tag{2.37}$$

где $\tilde{G}$ – оператор с полиномиальной нелинейностью. Однако в прикладных задачах часто встречаются случаи с операторами произвольной нелинейности.

Пусть требуется решить задачу

$$x = Gx \tag{2.38}$$

с произвольным нелинейным оператором G, сжимающим на некоторой области D, с постоянной сжатия $\alpha < 1$. И пусть x^* – единственная на D неподвижная точка оператора G.

В данном случае разумно предложить заменить исходный нелинейный оператор G на приближенный оператор $\tilde{G}$, который в свою очередь будет обладать свойством полиномиальной нелинейности. И далее вместо исходной задачи (2.38) решать систему (2.37) методом Монте-Карло.

В общем случае оператор $\tilde{G}$ может и не иметь неподвижных точек в области D или же иметь несколько неподвижных точек. Далее будем считать, что $\tilde{G}$ – сжимающий на замкнутом множестве $D_0 \subset D$, и $\tilde{G}D_0 \subset D_0$. Это гарантирует существование единственной неподвижной точки y^* у $\tilde{G}$ на множестве D_0. Вообще говоря, y^* может и не совпадать с x^*.

Запишем оператор G в виде

$$Gx = \tilde{G}x + \Delta Gx.$$

Нетрудно показать, что $x^* = y^*$ тогда и только тогда, когда $\Delta Gx^* = 0$. То есть, чтобы, решая приближенное уравнение (2.37), получилось решение близкое к решению системы (2.38), нужно выбирать приближение таким образом, чтобы значение остатка ΔG в точке x^* по норме было как можно меньше. Ситуацию осложняет тот факт, что x^* является предметом поиска и нельзя заранее проверить адекватность выбора $\tilde{G}$ и ΔG.

В связи с этим соображением рассмотрим следующую постановку. Предположим, что оператор G может быть приближен оператором с полиномиальной нелинейностью с заданной точностью $\varepsilon \geq 0$, иными словами

$$\|\Delta Gx\| < \varepsilon, \, \forall x \in D.$$

Далее возникает вопрос – на сколько решение задачи (2.37) отличается от решения (2.38). Ответ на этот вопрос даёт следующая оценка. Рассмотрим итерационный процесс

$$y_{k+1} = \tilde{G} y_k,$$

который при сделанных выше предположениях сходится к y^*. Верно следующее

$$\|y_{k+1} - x^*\| = \|y_{k+1} - \tilde{G} y_k - \Delta G y_k + G y_k - G y_{k+1} + G y_{k+1} - x^*\|,$$

$$\|y_{k+1} - x^*\| \le \|y_{k+1} - \tilde{G} y_k\| + \|\Delta G y_k\| + \|G y_k - G y_{k+1}\| + \|G y_{k+1} - G x^*\| \le$$

$$\le \|y_{k+1} - \tilde{G} y_k\| + \|\Delta G y_k\| + \alpha \|y_k - y_{k+1}\| + \alpha \|y_{k+1} - x^*\|.$$

И в итоге

$$\|y_{k+1} - x^*\| \le \frac{1}{1-\alpha} (\|y_{k+1} - \tilde{G} y_k\| + \|\Delta G y_k\| + \alpha \|y_k - y_{k+1}\|). \tag{2.39}$$

Переходя к пределу при $k \to \infty$ в левой и правой частях неравенства (2.39) получим оценку

$$\|y^* - x^*\| \le \frac{1}{1-\alpha} \|\Delta G y^*\| \le \frac{\varepsilon}{1-\alpha}. \tag{2.40}$$

Таким образом, ответ на поставленный вопрос получен.

Теперь перейдем к ряду примеров применения предложенных методов для решения задач, которые либо являются системами обыкновенных дифференциальных уравнений вида (2.1), либо могут быть к ней сведены.

2.3.1. Уравнение теплопроводности

Начнем с рассмотрения уравнения теплопроводности (см. [38, 39])

$$\frac{\partial u}{\partial t} = a^2 \frac{\partial^2 u}{\partial x^2}, \tag{2.41}$$

где $u = u(t, x)$, $x \in [0, 1]$, $t \ge 0$.

Начальные и граничные условия примем в виде

$$u(0,x) = \sin \pi x,$$

$$u(t,0) = 0,$$

$$u(t,1) = 0.$$

Введем на отрезке $[0,1]$ сетку $x_0 = 0, x_i = x_0 + i\Delta x, i = 1,\ldots,n+1$, $x_{n+1} = 1$ и заменим уравнение (2.41) на приближенные в точках сетки.

$$\dot{u}_i = a^2 \frac{u_{i-1} - 2u_i + u_{i+1}}{\Delta x^2}, i = 1,\ldots,n, \tag{2.42}$$

где $u_i = u(t,x_i)$. Начальные и граничные условия перепишутся в виде

$$u_i(0) = sin(\pi x_i),\ i = 1,\ldots,n, \tag{2.43}$$

$$u_0(t) = 0,\ u_{n+1} = 0. \tag{2.44}$$

Таким образом исходная задача свелась к системе обыкновенных дифференциальных уравнений вида (2.1). Теперь, вводя обозначения $y = y(t) = (u_1,\ldots,u_n)^T$, запишем получившуюся систему в векторном виде

$$\dot{y} = Ay, \tag{2.45}$$

где A – трёхдиагональная матрица $n \times n$, имеющая следующую структуру

$$\begin{pmatrix} -2a^2/\Delta x^2 & a^2/\Delta x^2 & & & 0 \\ a^2/\Delta x^2 & -2a^2/\Delta x^2 & a^2/\Delta x^2 & & \\ & \ddots & \ddots & \ddots & \\ & & a^2/\Delta x^2 & -2a^2/\Delta x^2 & a^2/\Delta x^2 \\ 0 & & & a^2/\Delta x^2 & -2a^2/\Delta x^2 \end{pmatrix}.$$

Далее полученную систему дифференциальных уравнений заменяем на равносильную систему интегральных уравнений. Сделать это можно, например, следующим образом. Представим матрицу A в виде суммы матриц

$A_1 + A_2$, где A_1 представляет собой диагональную матрицу с диагональю матрицы A, а матрица $A_2 = A - A_1$. Далее запишем равносильное интегральное уравнение

$$y(t) = \int_0^t e^{A_1(t-\tau)} A_2 y(\tau) d\tau + e^{A_1 t} y(0)$$

или для покомпонентной записи

$$y_i(t) = \int_0^t e^{-2(t-\tau)a^2/\Delta x^2} \frac{a^2}{\Delta x^2} (y_{i-1}(\tau) + y_{i+1}(\tau)) d\tau + e^{-2a^2/\Delta x^2 t} y_i(0),\ i = 1, \ldots, n.$$

Далее задача решается согласно схеме, описанной выше. Вообще говоря, построение оценки носит итеративный характер, который можно записать в следующем виде

$$W_j = W_{j-1} \frac{a^2/\Delta x^2\, e^{-2a^2(t_{j-1}-t_j)/\Delta x^2}}{q_{i_{j-1},i_j} p_{i_{j-1},i_j}(t_{j-1}, t_j)}, j = 1, 2, \ldots,$$

где $W_0 = h(x_0)/q^0_{i_0} p^0_{i_0}(x_0)$. В качестве переходной плотности рассмотрим усеченное экспоненциальное распределение

$$p_{i,j}(t, \tau) = \frac{a^2/\Delta x^2 e^{-2a^2(t-\tau)}}{1 - e^{-2a^2 t/\Delta x^2}}.$$

В этом случае итерационная процедура преобразуется к виду

$$W_j = W_{j-1} \frac{1 - e^{-2a^2 t_{j-1}/\Delta x^2}}{2 q_{i_{j-1},i_j}}, j = 1, 2, \ldots,$$

Известно (см., например, [3]), что если умножение на каждом шаге итерации происходит на величину меньшую по модулю единицы, то дисперсия такой оценки будет конечной. Нетрудно проверить, что при выборе вероятности g_j поглощения траектории цепи Маркова для каждого j такой, что выполнено неравенство

$$g_j < e^{-2a^2 t_{j-1}/\Delta x^2},$$

умножение в итерационной схеме будет на величину по модулю меньшую единицы. Следовательно дисперсия такой оценки будет конечной.

Вернёмся к рассмотрению системы (2.45). Вообще говоря, можно было и обойтись без представления матрицы A в виде суммы и сразу выписать решение

$$y(t) = e^{At}y(0).$$

В данном случае проблемой является вычисление матричной экспоненты от матрицы с параметром t. Как известно (см., например, [40]), собственные числа для A имеют представление:

$$\lambda_i = -\frac{4a^2}{\Delta x^2}\sin^2\frac{\pi i}{2(n+1)}, i = 1, \ldots, n,$$

а собственные векторы

$$v_k^{(i)} = \sin\frac{\pi k i}{n+1}, k, i = 1, \ldots, n.$$

Для матрицы с известным спектром можно выписать (см., например, [25]) следующее выражение для матричной экспоненты

$$e^{At} = \sum_{j=1}^{n} e^{\lambda_j t}\mathcal{L}_j(A),$$

где $\mathcal{L}_j(A)$ не зависят от t и имеют следующий вид

$$\mathcal{L}_j(A) = \frac{\prod_{k\neq j}(A - \lambda_k I)}{\prod_{k\neq j}(\lambda_j - \lambda_k)}.$$

В получившейся конечной сумме параметр t уже стоит под обычной экспонентой, а для оценки матриц $\mathcal{L}_j(A)$, которые со временем не меняются, требуется построение стохастического алгоритма.

В рассматриваемом же случае можно вычислить матричную экспоненту e^{At} еще проще. Дело в том, что матрица A является симметрической вещественной матрицей и, следовательно, допускает представление

$$A = Q\Lambda Q^T,$$

где матрица Q – ортогональная матрица, столбцами которой являются собственные векторы $v^{(i)}$ матрицы A, а Λ – диагональная матрица с собственными числами матрицы A на диагонали. Известно (см., например, [25]), что для произвольной обратимой матрицы T выполнено равенство

$$e^{TAT^{-1}} = Te^{A}T^{-1}.$$

В рассматриваемом случае в силу того, что $Q^T = Q^{-1}$, верно следующее

$$e^{At} = Qe^{\Lambda t}Q^T.$$

Обозначив матрицу e^{At} за $C = \{c_{i,j}\}_{i,j=1}^{n}$, нетрудно выписать выражения для элементов матрицы C:

$$c_{i,j} = \sum_{k=1}^{n} e^{\lambda_k t} \sin\frac{\pi ik}{n+1} \sin\frac{\pi jk}{n+1}.$$

Так же просто можно выписать собственные числа матрицы e^{At}, которые получаются взятием экспоненты от собственных чисел матрицы At, а именно

$$e^{\lambda_i t} = e^{-\frac{4a^2 t}{\Delta x^2}\sin^2\frac{\pi i}{2(n+1)}}, i = 1, \ldots, n.$$

Заметим, что для любого $t > 0$ выполнено неравенство

$$e^{\lambda_i t} < 1.$$

Любопытно отметить, что, решая одну задачу, мы столкнулись с другой, не менее интересной – оценкой матричной экспоненты от матрицы, зависящей от параметра.

2.4. Асинхронные релаксации

Как отметалось в начале главы, часть повествования будет уделена методу асинхронных итераций или иначе – асинхронных релаксаций.

Рассмотрим динамическую систему, состоящую из m взаимосвязанных подсистем. Пусть $x(t) = (x_1(t), \ldots, x_m(t)) \in \mathbb{R}^n$ – вектор состояния системы, $x_i(t) \in \mathbb{R}^{n_i}$ – вектор состояния подсистемы i в момент времени t, $n = \sum_{i=1}^{m} n_i$. Предполагается, что задано начальное состояние $x_i(0)$ для каждой из подсистем, а сама система описывается дифференциальными уравнениями вида

$$\frac{d}{dt}x_i(t) + F_i(x_i, t) = G_i(x, t) + u_i(t),\ i = 1, \ldots, m, \tag{2.46}$$

где $F_i(x_i, t)$, $G_i(x, t)$, $u_i(t)$ – заданные функции, образы которых лежат в $\mathbb{R}^{n_i}$. В такой постановке за динамику подсистемы отвечают функции F_i, а за взаимосвязь между системами функции G_i.

Применение асинхронных методов для решения таких систем было рассмотрено в [41] и [16]. Суть предложенных методов заключается в том, что подсистемы распределяются между процессорами и поиск решения выполняется при помощи итераций. На каждом шаге итераций для i-ой подсистемы предполагается наличие некоторого приближения искомого решения. Это приближение подставляется в правую часть i-ого уравнения в (2.46), после чего решается получившееся дифференциальное уравнение. Решение i-ого уравнения даст таким образом новое приближение, после чего процесс повторяется до тех пор пока не будет выполнен критерий остановки алгоритма.

Рассмотрим линейный случай уравнения (2.46)

$$\frac{d}{dt}x_i(t) + D_i(t)x_i(t) = \sum_{j=1}^{m} B_{i,j}(t)x_j(t) + u_i(t),\ i = 1, \ldots, m, \tag{2.47}$$

где $D_i(t)$ и $B_{i,j}(t)$ – матрицы размерности $n_i \times n_i$ и $n_i \times n_j$ соответственно. Будем далее предполагать, что все элементы матриц $D_i(t)$, $B_{i,j}(t)$ и векторов $u_i(t)$ являются непрерывными и ограниченными функциями на $[0, \infty)$. Это предположение обеспечивает существование и единственность решения уравнения (2.47) при любом начальном условии.

Обозначим за X_i множество непрерывных на интересующем нас интервале времени ($[0,T]$ или $[0,\infty)$) функций $x_i(t)$, которые согласуются с начальными условиями задачи (2.47). Пусть $X = X_1 \times X_2 \times \cdots \times X_m$. Определим отображения $f_i : X \to X_i$ следующим образом. Для функций $x_j(t) \in X_j$, $j = 1, \ldots, m$ пусть

$$y_i(t) = f_i(x_1(t), \ldots, x_m(t))$$

является решением дифференциального уравнения

$$\frac{d}{dt}y_i(t) + D_i(t)y_i(t) = \sum_{j=1}^{m} B_{i,j}(t)x_j(t) + u_i(t) \tag{2.48}$$

с начальным условием $y_i(0) = x_i(0)$. При сделанных предположениях относительно матриц $D_i(t)$, $B_{i,j}(t)$ и вектора $u_i(t)$ уравнение (2.48) имеет единственное решение, принадлежащее X_i. Как обсуждалось в предыдущих разделах $y_i(t)$ имеет следующее представление

$$y_i(t) = e^{-C_i(t)} \int_{t_0}^{t} e^{C_i(\tau)} \left[\sum_{j=1}^{m} B_{i,j}(t)x_j(\tau) + u_i(\tau) \right] d\tau + e^{-C_i(t)}x_i(0), \tag{2.49}$$

где

$$C_i(t) = \int_{0}^{t} D_i(\tau)d\tau.$$

Теперь, после того как определены функции f_i, $i = 1, \ldots, m$, определена и функция $f : X \to X$ такая, что $f = (f_1, f_2, \ldots, f_m)$. Пусть $x^*(t)$ – решение системы (2.47), удовлетворяющее начальным условиям. Тогда $x^*(t)$ удовлетворяет интегральному уравнению

$$x_i^*(t) = e^{-C_i(t)} \int_{t_0}^{t} e^{C_i(\tau)} \left[\sum_{j=1}^{m} B_{i,j}(t)x_j^*(\tau) + u_i(\tau) \right] d\tau + e^{-C_i(t)}x_i(0), \tag{2.50}$$

и, как не трудно видеть из уравнения (2.49), является неподвижной точкой оператора f.

Далее можно воспользоваться математическим аппаратом первой главы. Считаем, что итерационный процесс начинается с функций $x_i(t) \in X_i$, а функция f_i обновляет компоненту x_i в рамках одного процессора. Предполагается, что уравнение (2.48) решается с заданной точностью некоторым численным методом, природа которого на данном этапе не важна.

В [16] было показано, что при сделанных предположениях относительно матриц $D_i(t)$, $B_{i,j}(t)$ и векторов $u_i(t)$ асинхронные итерации для оператора f, начинающиеся с некоторой непрерывной функции, удовлетворяющей начальным условиям, сходятся равномерно на $[0, T]$, $T < \infty$ к решению системы (2.47).

На каждом шаге асинхронных релаксаций необходимо решать систему дифференциальных уравнений (2.48). В данном случае дополнительная возможность распараллеливания видится в применении метода Монте-Карло описанного ранее в этой главе. Уместно в такой ситуации будет говорить о комбинации двух методов – метода Монте-Карло и асинхронных релаксаций.

2.5. Численные эксперименты

В качестве примера использования метода Монте-Карло для решения систем обыкновенных дифференциальных уравнений с полиномиальной нелинейностью рассмотрим матричное уравнения Риккати

$$\frac{dX}{dt} = XA(t)X + B_1(t)X + XB_2(t) + C(t), X(t_0) = X_0 \qquad (2.51)$$

где X – матрица неизвестных размерности $n \times n$, $A(t), B_1(t), B_2(t), C(t)$ – заданные матрицы, зависящие от t, размерности $n \times n$. Уравнение Риккати играет важную роль в вариационном исчислении и в квантовой теории поля.

В модельном примере будем считать матрицы $A(t), B_1(t), B_2(t), (t)$, не зависящими от времени.

Система (2.51) естественным образом приводится к векторной форме, в результате чего получится система из n^2 уравнений.

Результаты моделирования приведены на рисунках 2.3 - 2.6. На рисунках 2.3 и 2.5 приведены графики точного решения (черная сплошная линия) для отдельных компонент решения и серия оценок метода Монте-Карло для этих компонент. На рисунках 2.4 и 2.6 изображены норма ошибки (синим цветом) и ее доверительные интервалы (красным цветом).

Решение систем обыкновенных дифференциальных уравнений методом Монте-Карло можно условно разделить на два этапа. Первый этап заключается в переходе от исходной системы к равносильной системе интегральных уравнений. Второй этап состоит собственно в применении метода Монте-Карло для нахождения решения получившейся системы интегральных уравнений. В основе предложенного метода лежат оценки, построенные на ветвящихся траекториях, которые удобно представлять как процесс рождения/-гибели частиц. Такое представление сразу же подключает аппарат теории случайных процессов, что несомненно является полезным при формальном описании траекторий, порождаемых такими процессами, и их свойств. Предложенные оценки являются несмещенными и простыми для реализации, однако имеют некоторые ограничения. Условия на сходимость мажоритарного итерационного процесса налагает ограничения на интервал интегрирования. Такое препятствие может быть преодолено использованием последовательного метода Монте-Карло, однако его применимость, а именно стохастическая устойчивость, как мы видели, зависит от свойств F_x вблизи искомого решения. Другое ограничение связано с тем, что алгоритм метода Монте-Карло подходит для задач с полиномиальной нелинейностью, но, как известно, в прикладных задачах нередко встречаются задачи с произвольной нелинейностью. В этом случае можно попытаться свести задачу к уже решенной, приблизив нелинейный оператор оператором с полиномиальной нелинейностью,

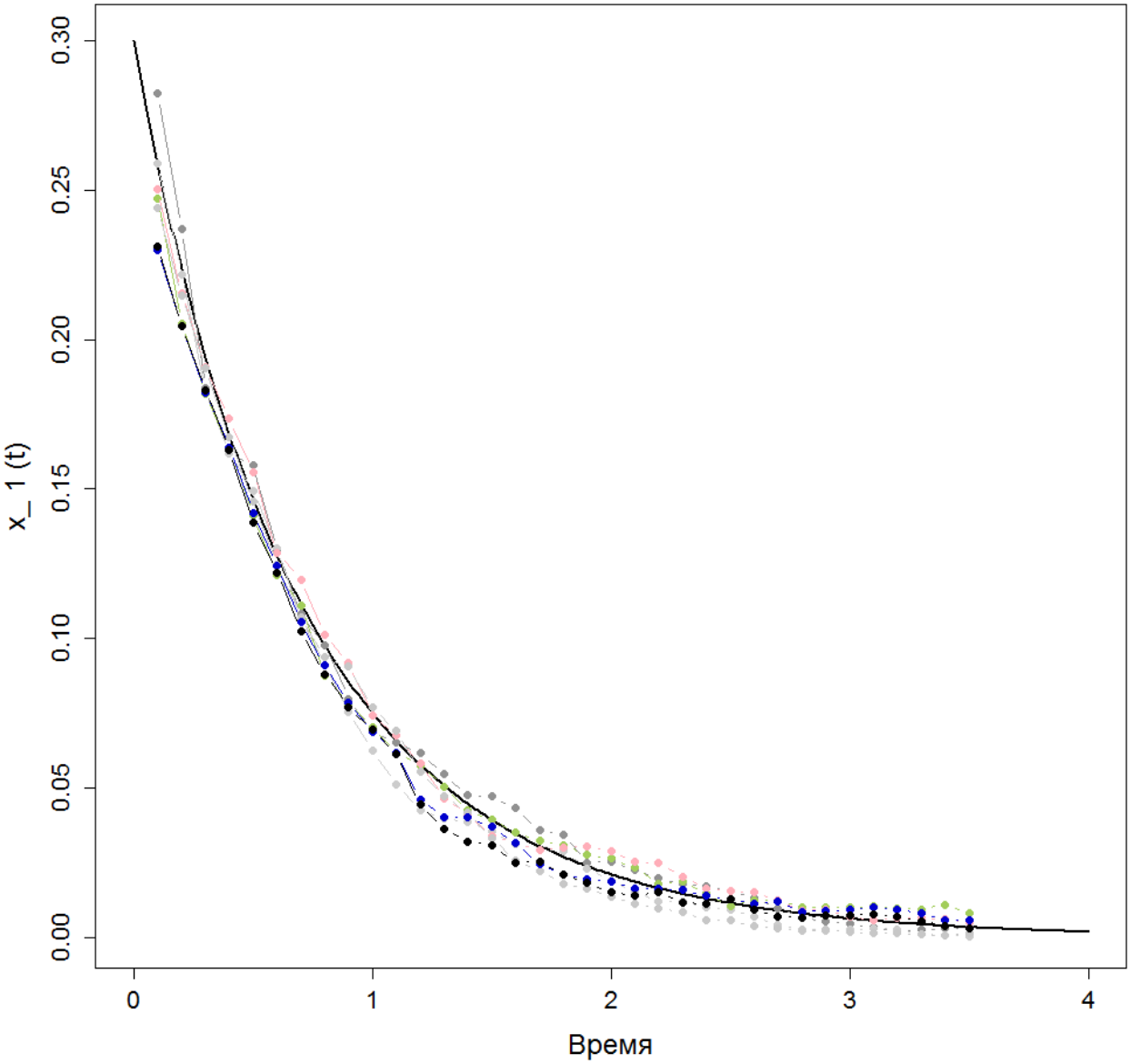

Рис. 2.3. Серия оценок первой компоненты

не забывая при этом, что в результате такой замены возникает дополнительная погрешность.

Предложенные оценки метода Монте-Карло, как отмечалось выше, довольно просты в реализации и легко адаптируются для использования на параллельных вычислительных системах. Метод релаксаций для решения си-

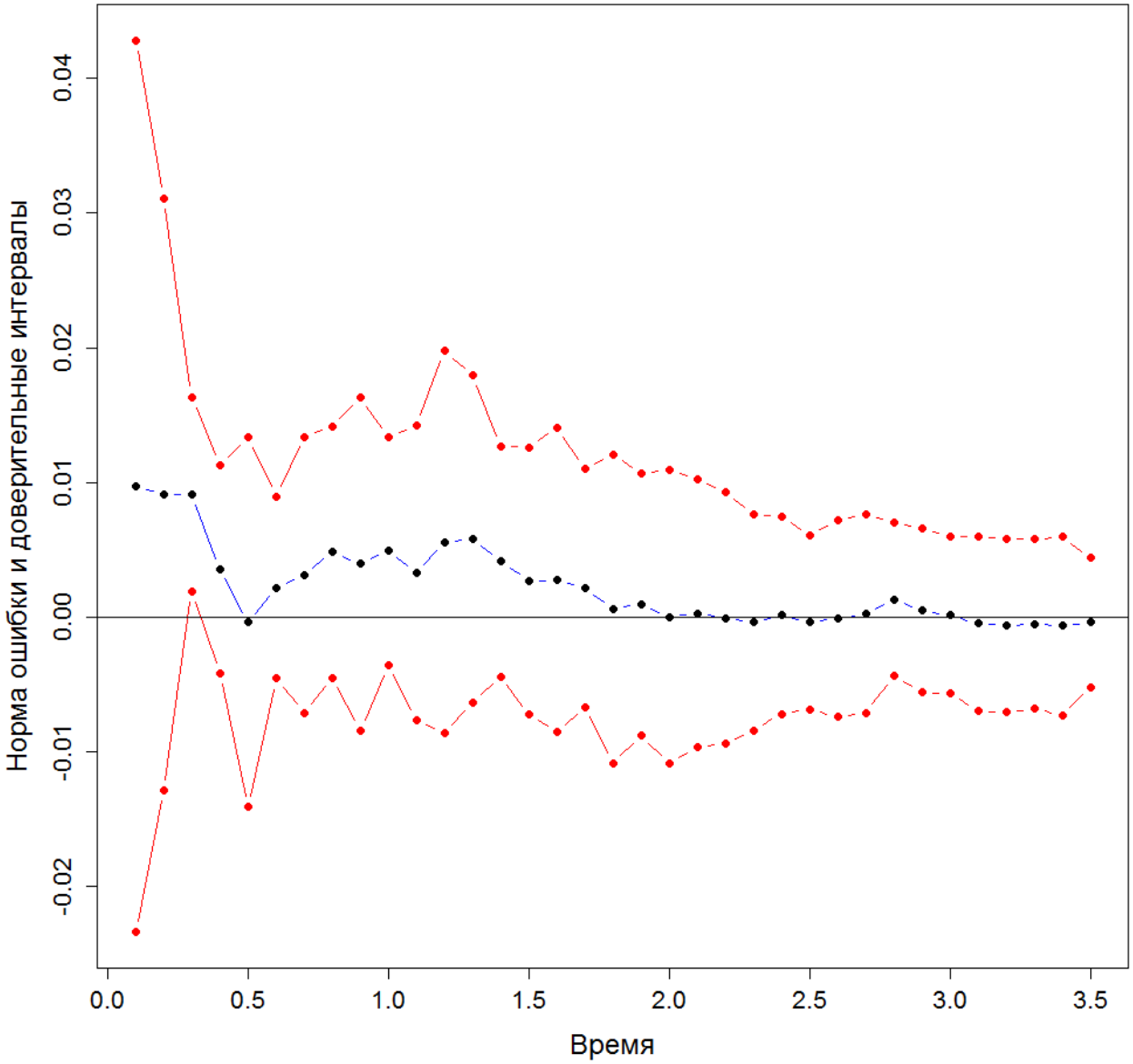

Рис. 2.4. Оценка ошибки для первой компоненты

стем обыкновенных дифференциальных уравнений является связкой с идеями и методами первой главы, которая открывает дополнительные возможности для асинхронного решения задач на многопроцессорных системах.

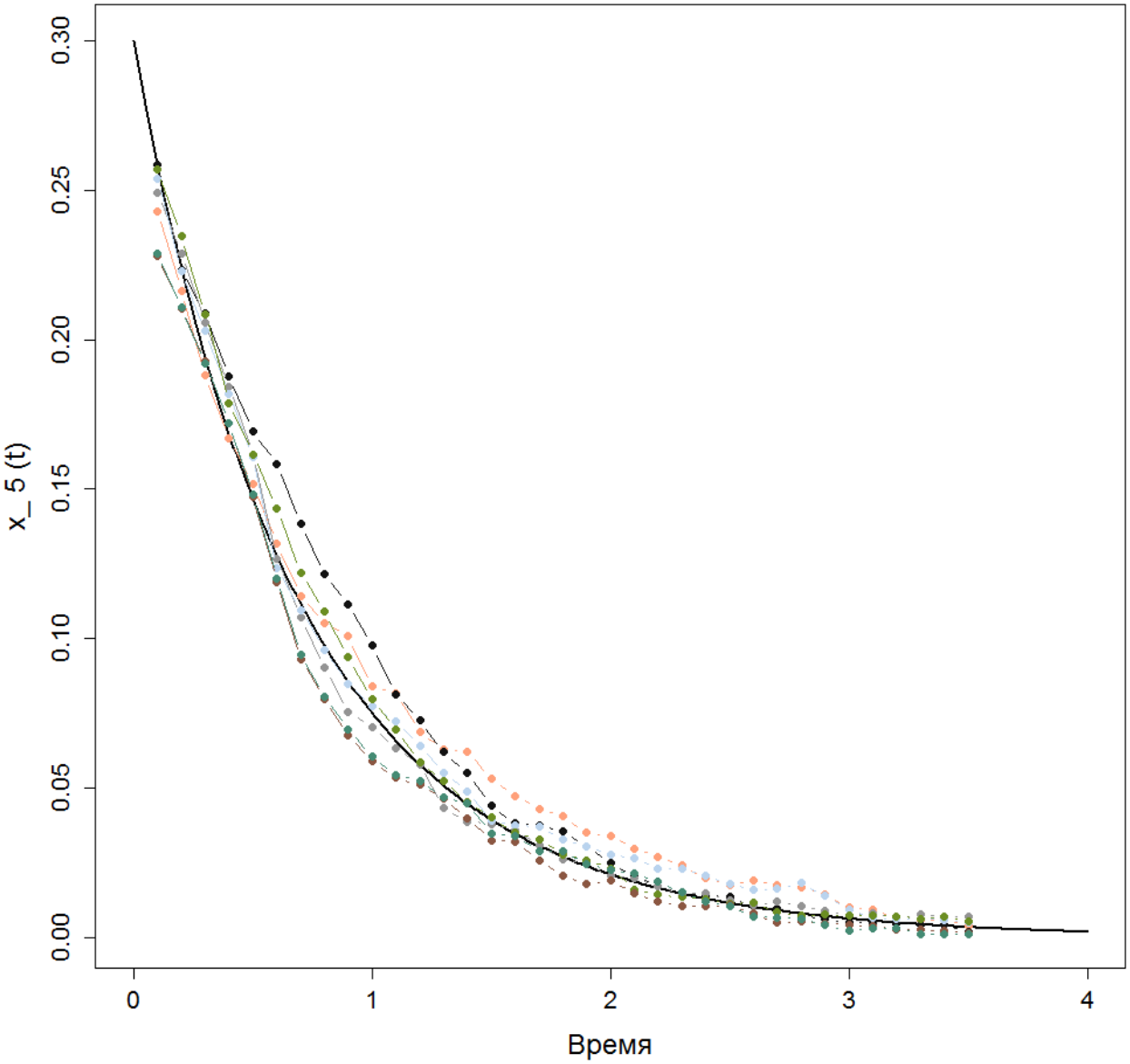

Рис. 2.5. Серия оценок пятой компоненты

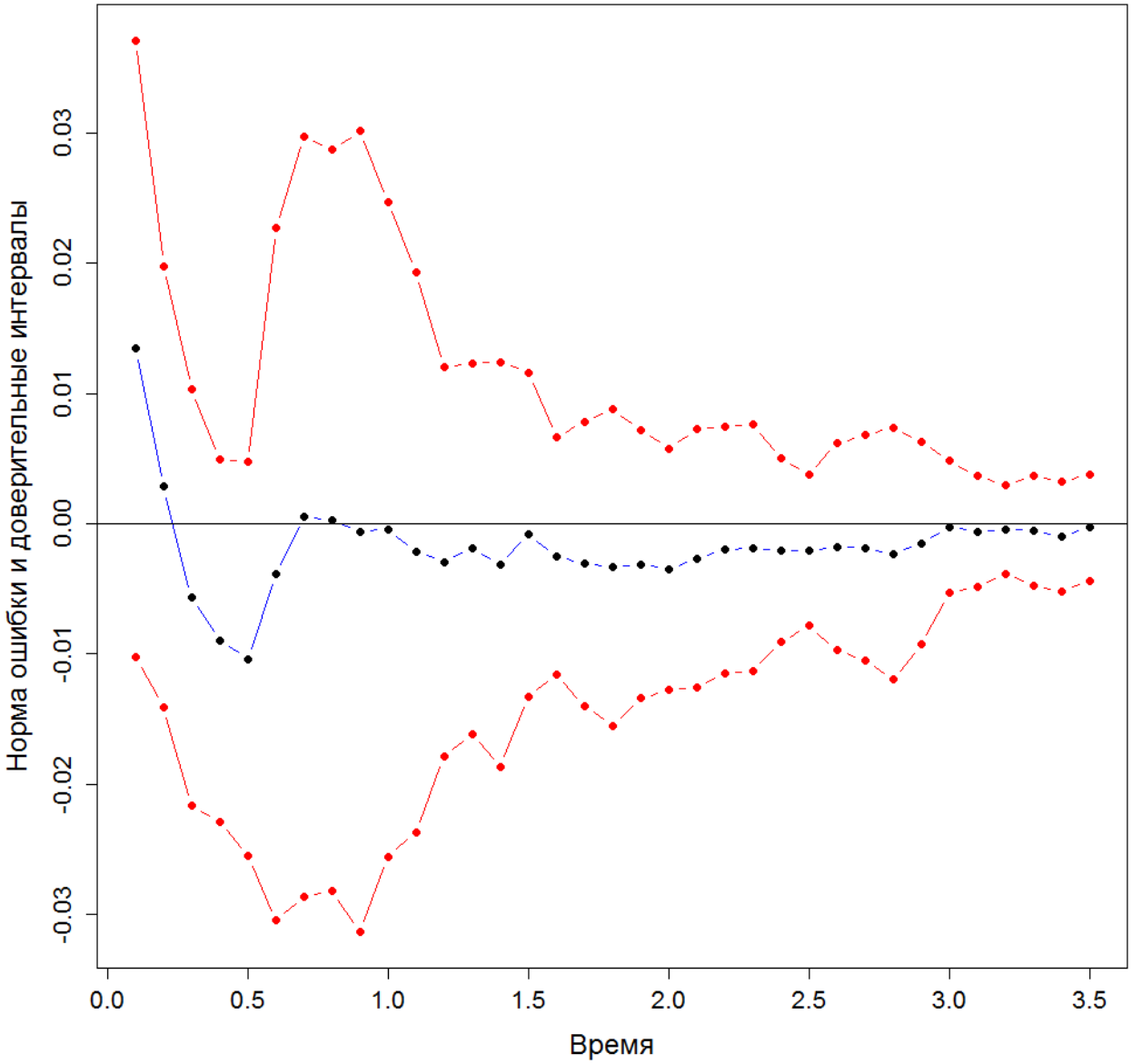

Рис. 2.6. Оценка ошибки для пятой компоненты

Глава 3

Оценка Американских опционов методом Монте-Карло

Оценка опционов является одной из важных задач финансовой математики, и разрешению этой проблемы посвящено множество работ [42–46]. Известно, что в общем случае цена опциона зависит от цены исполнения, величины базового актива, волатильности, выплачиваемых дивидендов, процентной ставки и момента исполнения. Таким образом для корректной оценки опциона необходимо учитывать все эти факторы. Более того, для Американских опционов стоит также рассматривать возможность досрочного исполнения. Это часто приводит к очень сложным вычислениям и редко удается найти аналитическое решение.

Когда не удается решить задачу аналитически, оправдано использование численных методов. Одним из известнейших численных методов для нахождения цены опциона является биномиальный метод (см., например, [47]). Не смотря на то, что в биномиальной модели можно реализовать возможность досрочного исполнения, вычисления становятся невероятно трудоемкими при росте количества стохастических параметров. Проблема в том, что количество узлов, необходимых для вычислений, растет экспоненциально, если мы хотим учитывать в модели такие стохастические параметры как процентная ставка, дивиденды, волатильность или сложный базовый актив.

Альтернативный подход – использование дифференциальных уравнений: стохастических или в частных производных. Недостатком стохастических дифференциальных уравнений является тот факт, что с ростом по времени растет дисперсия решения. Этого недостатка лишены решения уравнений в частных производных. При численном решении таких уравнений используют

методы, который сводят исходное уравнение к системе алгебраических уравнений, при этом происходит дискретизация области определения. Заметим, что при увеличении количества параметров, от которых зависит цена опциона, растет размерность системы, что приводит к большой трудоемкости в случае использования детерминированных методов. В данном ситуации становится оправданным использование метода Монте-Карло. Стоит отметить также, что коэффициенты в уравнении могут носить случайный характер, а для решения таких задач особенно эффективным является метод Монте-Карло, который к тому же обладает свойствами параллелизма.

3.1. Основы опционов

Опцион – это ценная бумага, предоставляющая своему владельцу право купить или продать некоторый базовый актив в установленный период или момент времени на заранее оговариваемых условиях. Они являются, таким образом, производной ценной бумагой, поскольку их стоимость зависит от стоимости базового актива. В роли базового актива могут выступать акции, индексы акций, опционы, иностранная валюта, закладные и.т.п.

Существует два основных типа опционов: колл (Call option) и пут (Put option). Опцион колл дает своему владельцу право купить определенное количество базового актива по заранее фиксированной цене исполнения или цене покупки. Опцион пут дает его держателю право продать определенное количество базового актива по фиксированной цене продажи. Тот, кто продает или выписывает опцион, называется продавцом или выписывающей стороной. Чтобы приобрести опцион, его будущий владелец платит выписывающей стороне премию. Когда исполняется опцион колл, его владелец платит продавцу, скажем цену исполнения в обмен на акцию, и опцион прекращает свое существование. В случае опциона пут, его владелец получает от выписывающей

стороны цену исполнения в обмен на акцию.

Различают также Американские и Европейские опционы. Они отличаются лишь способом исполнения. Американский опцион можно исполнить в любой момент до окончания его срока действия, в то время как Европейский – лишь в момент его окончания.

Для определенности будем предполагать, что речь идет об опционах, построенных на акциях, стоимость которых обозначим $S = S(t)$. Также будем полагать, что период существования опциона есть временной интервал $[0, T]$.

Рассмотрим теперь, для примера, опцион-колл Европейского типа со временем исполнения T. Такой опцион характеризуется фиксированной в момент его покупки ценой K (цена исполнения), по которой покупатель может купить акции, фактическая стоимость которых $S(T)$ в момент T может, и существенным образом, отличаться от K.

Если $S(T) > K$, то эта ситуация окажется благоприятной для покупателя, поскольку ему по условиям контракта дано право купить акции по цене K, что он может и сделать с немедленной затем их продажей по рыночной цене $S(T)$. Доход от этой операции составит $S(T) - K$.

Если же окажется, что $S(T) < K$, то данное покупателю право покупки по цене K ему ничего не дает, поскольку он может купить акции по более низкой цене $S(T)$.

Таким образом доход покупателя в момент T составит $\max(S(T) - K, 0)$. Разумеется, за покупку такого финансового инструмента надо заплатить некоторую премию C. Таким образом чистый доход покупателя опциона-колл будет равен $\max(S(T) - K, 0) - C$. Соответственно доход продавца –
$C - \max(S(T) - K, 0)$.

В данном случае ключевыми являются два вопроса – какова “справедливая” цена C продажи-покупки опциона и как должен действовать продавец опциона, чтобы выполнить условия контракта. Нас будет интересовать

первый вопрос, то есть как определить “справедливую” цену C, которая бы устроила и продавца, и покупателя.

На практике большинство торгуемых опционов являются опционами Американского типа, которые дают больше свободы покупателю, допуская выбор момента исполнения. В случае же Европейских опционов этот момент заранее определен. В случае Американских опционов наряду с вопросом о “справедливой” цене возникает вопрос выбора момента исполнения. Например, если для Американского опциона-колл в момент времени $\tau < T$ доход $\max(S(\tau) - K, 0) - C$ окажется больше 0, как лучше поступить: предъявить опцион к исполнению или же ждать дальнейшего роста S.

3.2. Модель Блэка-Шоулса

Оценка стоимости опционов является важной задачей финансовой математики, и решению этой проблемы посвящено много работ (см., например, [48]). Среди них важную роль играет модель Блэка-Шоулса оценки стоимости опциона. Блэк и Шоулс показали (см. [49]), что цена опциона удовлетворяет следующему уравнению в частных производных

$$\frac{\partial U}{\partial t} + \frac{\sigma^2 S^2}{2}\frac{\partial^2 U}{\partial S^2} + rS\frac{\partial U}{\partial S} - rU = 0,$$

где

- t – время,
- S – цена акции,
- $U = U(S, t)$ – цена опциона,
- σ – постоянная волотильность,
- r – безрисковая ставка.

Это уравнение справедливо при следующих предположениях:

- Разрешена короткая продажа акций;
- Торговля ценными бумагами (базовым активом) ведется непрерывно;
- Нет транзакционных затрат, связанных с покупкой или продажей акции или опциона;
- Краткосрочная безрисковая процентная ставка r известна и является постоянной в течение всего срока действия опциона;
- Любой покупатель ценной бумаги может получать ссуды по краткосрочной безрисковой ставке для оплаты любой части ее цены.

Начальные и граничные условия для этого уравнения зависят от типа опциона.

3.3. Метод подвижной границы

Европейский опцион, как известно, может быть исполнен лишь в момент окончания его срока действия, в то время как Американский опцион может быть исполнен в любой момент до окончания его срока. Пусть $P = P(S,t)$ – цена Американского опциона-пут в момент времени t. Тогда P удовлетворяет следующему уравнению

$$\frac{\partial P}{\partial t} + \frac{\sigma^2 S^2}{2}\frac{\partial^2 P}{\partial S^2} + rS\frac{\partial P}{\partial S} - rP = 0, \quad S > B(t), \quad 0 \le t < T. \tag{3.1}$$

Здесь $B(t)$ – подвижная граница. Смысл $B(t)$ следующий: если цена акции $S(t)$ опустится ниже $B(t)$, то необходимо предъявить опцион к исполнению. Однако проблема в том, что $B(t)$ не известна заранее. Предполагается, что дивиденды не выплачивают на протяжении существования опциона.

Для уравнения (3.1) необходимо задать начальные и граничные условия. В момент окончания срока действия опциона T условия следующие:

$$P(S,T) = \max(K - S, 0), \quad S \geq 0,$$

где K – цена исполнения опциона.

Если цену акции S устремить к бесконечности, то рано или поздно S станет больше K, и чем больше будет разница $S - K$, тем меньше будет стоить опцион. Таким образом условие на бесконечности:

$$\lim_{S \to \infty} P(S,t) = 0, \quad 0 \leq t \leq T.$$

Так как после пересечения границы $B(t)$ опцион исполняется, то условия на подвижной границе выглядят следующим образом

$$P(B(t),t) = K - B(t), \quad \frac{\partial P}{\partial S}(B(t),t) = -1.$$

Также определим подвижную границу в момент окончания срока действия опциона:

$$B(T) = K.$$

И наконец, цена опциона в области $0 \leq S < B(t)$:

$$P(S,t) = \max(K - S, 0).$$

Получившаяся задача весьма специфична в силу своих граничных условий. Один из подходов к решению этой задачи – это замена переменных (см., например, [50, 51]), которая приведет к задаче с фиксированной областью определения. При этом возникнет другая трудность – исходное уравнение становится нелинейным.

Сделаем замену переменных

$$x = \frac{S}{B(t)}$$

и

$$p(x,t) = P(S,t) = P(xB(t),t).$$

Получим $x \in [1,+\infty)$ для $S \in [B(t),\infty)$. Нашей целью является получение уравнений для $p(x,t)$ при $x \geq 1$, $0 \leq t \leq T$.

Частные производные будут иметь следующий вид

$$\frac{\partial P}{\partial S} = \frac{\partial p}{\partial x}\frac{\partial x}{\partial S} + \frac{\partial p}{\partial t}\frac{\partial t}{\partial S} = \frac{\partial p}{\partial x}\frac{1}{B(t)}, \tag{3.2}$$

$$\frac{\partial P}{\partial t} = \frac{\partial p}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial p}{\partial t}\frac{\partial t}{\partial t} = -\frac{\partial p}{\partial x}\frac{B'(t)}{B^2(t)} + \frac{\partial p}{\partial t}, \tag{3.3}$$

$$\frac{\partial^2 P}{\partial S^2} = \frac{\partial}{\partial x}\left(\frac{\partial p}{\partial S}\frac{1}{B(t)}\right) = \frac{\partial}{\partial x}\left(\frac{\partial p}{\partial x}\frac{1}{B^2(t)}\right) = \frac{1}{B^2(t)}\frac{\partial^2 p}{\partial x^2}. \tag{3.4}$$

Подставив выражения (3.2)-(3.4) в исходное уравнение (3.1), получим

$$\frac{\partial p}{\partial t} + \frac{\sigma^2 x^2}{2}\frac{\partial^2 p}{\partial x^2} + x\left[r - \frac{B'(t)}{B(t)}\right]\frac{\partial p}{\partial x} - rp = 0, \quad 1 < x < \infty, \quad 0 \leq t < T. \tag{3.5}$$

Граничные условия приобретут вид:

$$\begin{aligned} &p(x,T) = 0, \quad x \geq 1, \\ &\lim_{x\to\infty} p(x,t) = 0, \\ &\frac{\partial p}{\partial x}(1,t) = -B(t), \\ &p(1,t) = K - B(t), \\ &B(T) = K. \end{aligned} \tag{3.6}$$

Заметим, что теперь в уравнении две неизвестных: цена опциона P и граница $B(t)$, а само уравнение стало нелинейным. Решив задачу (3.5)-(3.6) относительно p и B, можно будет вычислить цену Американского опциона следующим образом:

$$P(S,t) = \begin{cases} p(S/B(t),t), & \text{при } S/B(t) \geq 1; \\ K - S, & \text{при } 0 \leq S/B(t) < 1. \end{cases}$$

Для численного решения рассмотренных задач (3.5)-(3.6) могут быть использованы различные методы, их обзор можно найти в [34, 48]. В настоящей работе будет использован метод конечных разностей.

Для решения задачи (3.5)-(3.6) методом конечных разностей на области $1 \leq x < \infty$, $0 \leq t \leq T$ введем сетку с шагом Δx по x и Δt по t. Такую что

$$\Delta x = \frac{x_\infty}{M}, \quad \Delta t = \frac{T}{N},$$
$$x_i = 1 + i\Delta x, \quad i = 0, ..., M\,,$$
$$t_j = j\Delta t, \quad j = 0, ..., N.$$

Для численного решения задачи было введено большое число x_∞, для которого выполнено условие $p(x_\infty, t) = 0$. Введем обозначения $p_{i,j} = p(x_i, t_j)$ и $B_j = B(t_j)$. Для уравнения (3.5) получим неявную разностную схему:

$$\frac{p_{i,j+1} - p_{i,j}}{\Delta t} + \frac{x_i^2\sigma^2}{2}\frac{p_{i-1,j} - 2p_{i,j} + p_{i+1,j}}{(\Delta x)^2} + x_i\left[r - \frac{B_{j+1} - B_j}{B_j \Delta t}\right]\frac{p_{i+1,j} - p_{i-1,j}}{2\Delta x} -$$
$$- rp_{i,j} = 0, \quad i = 1, ..., M-1, \quad j = 0, ..., N-1. \tag{3.7}$$

Со следующими граничными условиями

$$p_{i,N} = 0, \quad i = 0, \ldots, M,$$
$$p_{M,j} = 0, \quad j = 0, \ldots, N-1,$$
$$\frac{p_{1,j} - p_{0,j}}{\Delta x} = -B_j, \quad j = 0, \ldots, N-1,$$
$$p_{0,j} = K - B_j, \quad j = 0, \ldots, N-1,$$
$$B_N = K.$$

Получилась система нелинейных алгебраических уравнений. Можно записать эту систему в более компактном виде, удобном для применения метода Нью-

тона. Для каждого шага по времени $j = N-1, \ldots, 0$ представим систему (3.7) в виде:

$$F(y) = 0,$$
$$y = (p_{0,j}, ..., p_{M-1,j}, B_j)^T.$$

Итерационный процесс метода Ньютона для данного уравнения будет выглядеть так

$$y_{k+1} = y_k - J^{-1}(y_k)F(y_k), \quad k \geq 0,$$

где J – якобиан F, имеющий следующую структуру

$$J = \begin{pmatrix} 1 & 0 & 0 & \ldots & 0 & 1 \\ -1/\Delta x & 1/\Delta x & 0 & \ldots & 0 & 1 \\ a_1 & b_1 & c_1 & \ldots & 0 & d_1 \\ 0 & a_2 & b_2 & c_2 & \ldots & d_2 \\ \vdots & & & & \ddots & \vdots \\ 0 & \ldots & 0 & a_{M-1} & b_{M-1} & d_{M-1} \end{pmatrix},$$

где

$$a_i = \frac{x_i^2\sigma^2}{2(\Delta x)^2} - \frac{x_i}{2(\Delta x)}\left(r - \frac{B_{j+1} - B_j}{B_j \Delta t}\right),$$
$$b_i = -\frac{1}{\Delta t} - \frac{x_i^2\sigma^2}{(\Delta x)^2} - r,$$
$$c_i = \frac{x_i^2\sigma^2}{2(\Delta x)^2} + \frac{x_i}{2(\Delta x)}\left(r - \frac{B_{j+1} - B_j}{B_j \Delta t}\right),$$
$$d_i = x_i \frac{B_{j+1}}{B_j^2 \Delta t} \frac{p_{i+1,j} - p_{i-1,j}}{2\Delta x},$$
$$i = 1, \ldots, M-1.$$

Решая последовательно эту задачу на каждом шаге по времени с $j = N-1$ до 0, мы найдем значение цены опциона в начальный момент времени.

На каждом шаге итерации будет решаться линейная система вида $J(y_k)Y = -F(y_k)$. Представим систему в виде

$$Y = A(y_k)Y + F(y_k),$$

где матрица A имеет структуру

$$A = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 1 \\ -1 & 0 & 0 & \dots & 0 & \Delta x \\ a_1/c_1 & b_1/c_1 & 0 & \dots & 0 & d_1/c_1 \\ 0 & a_2/c_2 & b_2/c_2 & 0 & \dots & d_2/c_2 \\ \vdots & & & & \ddots & \vdots \\ 0 & \dots & 0 & a_{M-1}/d_{M-1} & b_{M-1}/d_{M-1} & 0 \end{pmatrix},$$

Однако методы решения систем линейных уравнений, предложенные в предыдущих пунктах, в данном случае не дают положительных результатов. Это происходит в силу того, что соответствующие итерационные процессы не сходятся. По этой причине перейдем к рассмотрению альтернативного метода.

3.4. Метод штрафной функции

Метод штрафной функции для решения задач нахождения цены опциона впервые был предложен в статье [52]. Суть метода заключается в добавлении к уравнению в частных производных небольшого слагаемого, которое представляет собой непрерывную функцию $f(P)$, нелинейно зависящую от цены опциона P. При этом получается нелинейное дифференциальное уравнение с фиксированной областью определения. Как показано в работе [50], решение этого уравнение будет обладать свойствам аналитического решения задачи нахождения цены Американского опциона при достаточной малости добавляемого слагаемого. Рассмотрим уравнение Блэка-Шоулса с такой добавкой:

$$\frac{\partial P_\varepsilon}{\partial t} + \frac{\sigma^2 S^2}{2}\frac{\partial^2 P_\varepsilon}{\partial S^2} + rS\frac{\partial P_\varepsilon}{\partial S} - rP_\varepsilon + f_\varepsilon(P_\varepsilon) = 0 \qquad (3.8)$$

с граничными условиями

$$\begin{aligned} &P_\varepsilon(S,T) = \max(K-S,0), \\ &\lim_{S\to\infty} P_\varepsilon(S,t) = 0, \\ &P_\varepsilon(0,t) = K, \end{aligned} \tag{3.9}$$

где

$$f_\varepsilon(P_\varepsilon) = \frac{\varepsilon C}{P_\varepsilon + \varepsilon - K + S}, \tag{3.10}$$

$C \geq rK$ – положительная константа. Здесь индексация призвана подчеркнуть зависимость решения уравнения (3.8)-(3.9) от ε. Справедлива следующая теорема (см. [50]):

Теорема 12. *Пусть P – единственное решение уравнения Блэка-Шоулза, а P_ε – единственное решение уравнения (3.8) для $\varepsilon > 0$. Тогда $P_\varepsilon \to P$ в $L_\infty(\overline{Q}_T)$ при $\varepsilon \to 0$, где $Q_T = [0,T]\times Q$, $Q \subset R^m$ – открытое и ограниченное.*

Далее индекс ε будет опускаться. Пусть $0 < \varepsilon \ll 1$ и

$$\frac{\partial P}{\partial t} + \frac{\sigma^2 S^2}{2}\frac{\partial^2 P}{\partial S^2} + rS\frac{\partial P}{\partial S} - rP + f(P) = 0, \quad S \geq 0, \quad 0 \leq t < T. \tag{3.11}$$

Как уже отмечалось, при стремлении ε к нулю, решение задачи (3.11), (3.10), (3.9) будет стремиться к цене американского опциона. Получившаяся задача является нелинейной с фиксированной областью определения.

Для численного решения рассмотренной задачи (3.11), (3.10), (3.9) могут быть использованы различные методы, их обзор можно найти в [48]. Мы рассмотрим метод конечных разностей.

Обратим внимание также на следующий интересный факт. При дискретизации мы переходим от задачи, где время представлено непрерывно, к дискретному времени. Рассматривая список временных моментов, когда опцион может быть исполнен, мы переходим, таким образом, к формальному рассмотрению так называемого бермудского опциона.

Для того, чтобы численно решить систему (3.11), (3.10), (3.9) введем S_∞ – большое значение S, в котором предполагается выполнение граничного условия на бесконечности, а именно $P(S_\infty, t) = 0$. Для решения задачи (3.11), (3.10), (3.9) методом конечных разностей в области $S \geq 0$, $0 \leq t \leq T$ введем сетку:

$$\Delta S = \frac{S_\infty}{M}, \quad \Delta t = \frac{T}{N},$$
$$S_i = i\Delta S, \quad i = 0, ..., M,$$
$$t_j = j\Delta t, \quad j = 0, ..., N,$$
$$P_{i,j} = P(S_i, t_j).$$

Полунеявная разностная схема, используемая в [50], для уравнения (3.11) имеет вид

$$\frac{P_{i,j+1} - P_{i,j}}{\Delta t} + \frac{(i\Delta S)^2\sigma^2}{2}\frac{P_{i-1,j} - 2P_{i,j} + P_{i+1,j}}{(\Delta S)^2} + i\Delta S r \frac{P_{i+1,j} - P_{i-1,j}}{2\Delta S} - rP_{i,j} +$$
$$+ \frac{\varepsilon C}{P_{i,j+1} + \varepsilon - K + i\Delta S} = 0 \quad , i = 1 \dots M-1, \quad j = 0 \dots N-1,$$

$$P_{i,N} = \max(K - i\Delta S, 0), \quad i = 1 \dots M-1,$$
$$P_{M,j} = 0, \quad j = 0 \dots N-1,$$
$$P_{0,j} = K, \quad j = 0 \dots N-1.$$

После перегруппировки слагаемых получим

$$P_{i,j} = \frac{i^2\sigma^2\Delta t - ri\Delta t}{2(1 + i^2\sigma^2\Delta t + r\Delta t)}P_{i-1,j} + \frac{i^2\sigma^2\Delta t + ri\Delta t}{2(1 + i^2\sigma^2\Delta t + r\Delta t)}P_{i+1,j} +$$
$$+ \frac{P_{i,j+1}}{(1 + i^2\sigma^2\Delta t + r\Delta t)} + \frac{\Delta t\varepsilon C}{(1 + i^2\sigma^2\Delta t + r\Delta t)(P_{i,j+1} + \varepsilon - K + i\Delta S)},$$
$$i = 1 \dots M-1, \quad j = 0 \dots N-1.$$

Введем обозначение $X_j = (P_{1,j}, P_{2,j}, \dots, P_{M-1,j})^T$ и заметим, что в общем эта система является нелинейной, однако на каждом шаге по времени

получается линейная система вида

$$X_j = AX_j + F(X_{j+1}), \tag{3.12}$$

$$A = \begin{pmatrix} 0 & c_1 & 0 & \dots & 0 & 0 \\ a_2 & 0 & c_2 & \dots & 0 & 0 \\ 0 & a_3 & 0 & \dots & 0 & 0 \\ . & . & . & . & . & . \\ 0 & 0 & 0 & \dots & 0 & c_{M-2} \\ 0 & 0 & 0 & \dots & a_{M-1} & 0 \end{pmatrix}, \tag{3.13}$$

где

$$a_i = \frac{i^2\sigma^2\Delta t - ri\Delta t}{2(1 + i^2\sigma^2\Delta t + r\Delta t)},$$
$$c_i = \frac{i^2\sigma^2\Delta t + ri\Delta t}{2(1 + i^2\sigma^2\Delta t + r\Delta t)},$$
$$S_i = i\Delta S,$$
$$i = 1, \dots, M-1,$$

а вектор-функция F:

$$F(X_{j+1}) = (f_1(P_{1,j+1}), f_2(P_{2,j+1}), \dots, f_{M-1}(P_{M-1,j+1}))^T,$$

$$f_1(P_{1,j+1}) = \frac{\sigma^2\Delta t - r\Delta t}{(1 + \sigma^2\Delta t + r\Delta t)}K + \frac{P_{1,j+1}}{(1 + \sigma^2\Delta t + r\Delta t)} + \\ + \frac{\Delta t\varepsilon C}{(P_{1,j+1} + \varepsilon - K + \Delta S)(1 + \sigma^2\Delta t + r\Delta t)},$$

$$f_i(P_{i,j+1}) = \frac{P_{i,j+1}}{(1 + i^2\sigma^2\Delta t + r\Delta t)} + \frac{\Delta t\varepsilon C}{(P_{i,j+1} + \varepsilon - K + i\Delta S)(1 + i^2\sigma^2\Delta t + r\Delta t)},$$

$i = 2, \dots, M-1.$

При каждом фиксированном j система (3.12) является системой линейных алгебраических уравнения (X_{j+1} – вычислено на предыдущем шаге, X_j – неизвестно). Таким образом, последовательно решив N систем уравнений

начиная с j равного $N-1$ и до 0, мы найдем цену опциона в начальный момент времени. Чтобы применять метод Монте-Карло для решения системы при каждом фиксированном j, необходимо убедиться в том, что спектральный радиус матрицы $|A|$ по модулю меньше единицы. Докажем следующую лемму

Лемма 4. *При каждом фиксированном j существует $\Delta t > 0$ такое, что спектральный радиус матрицы $|A|$ (3.13) меньше единицы.*

Доказательство. Рассмотрим сумму элементов строки матрицы $|A|$, состоящей из модулей элементов матрицы A:

$$|a_i| + |c_i| = \frac{\Delta t(|(i^2\sigma^2 - ri)| + i^2\sigma^2 + ri)}{2(1 + i^2\sigma^2\Delta t + r\Delta t)}, \quad i = 1, \dots M-1.$$

При фиксированном ΔS, уменьшая Δt, можно сделать сумму меньше единицы для любого i, сделав норму (в качестве нормы берётся максимум из сумм модулей элементов строк) матрицы $|A|$, а тем самым и ее спектральный радиус, меньше единицы. Можно получить оценку для Δt,

$$\frac{\Delta t(|(i^2\sigma^2 - ri)| + i^2\sigma^2 + ri)}{2(1 + i^2\sigma^2\Delta t + r\Delta t)} = \frac{\Delta t \max(i^2\sigma^2, ri)}{1 + i^2\sigma^2\Delta t + r\Delta t} < 1. \tag{3.14}$$

Заметим, что неравенство

$$\frac{i^2\sigma^2\Delta t}{1 + i^2\sigma^2\Delta t + r\Delta t} < 1$$

выполнено для любых Δt и ΔS. Рассмотрим второе неравенство

$$\frac{ri\Delta t}{1 + i^2\sigma^2\Delta t + r\Delta t} < 1.$$

Напомним, что $1 \le i \le M$, поэтому

$$\frac{ri\Delta t}{1 + i^2\sigma^2\Delta t + r\Delta t} \le \frac{rM\Delta t}{1 + \sigma^2\Delta t + r\Delta t} < 1.$$

После преобразований получим

$$\Delta t(r(M-1) - \sigma^2) < 1.$$

При $r(M-1)-\sigma^2 \leq 0$ предыдущее неравенство, а следовательно и (3.14), выполнено. Если же $r(M-1)-\sigma^2 > 0$, то для выполнения неравенства (3.14), достаточно следующего условия на Δt:

$$\Delta t < \frac{1}{r(M-1)-\sigma^2}.$$

□

Доказанная лемма означает, что можно находить решение системы методом Монте-Карло (возможно также применение квази Монте-Карло). Оба эти метода обладают свойствами естественного параллелизма.

Теперь при каждом фиксированном j мы будем находить X_j с помощью метода Монте-Карло. Это означает, что будет строиться последовательность оценок ξ_n, таких что

$$\xi_j = \widetilde{(I-A)}^{-1} F(\hat{\xi}_{j+1}(N_{j+1})), \tag{3.15}$$

где волна обозначает, что соответствующий результат обращения матрицы и умножения её на вектор осуществляется посредством метода Монте-Карло, а $\hat{\xi}_{j+1}(N_{j+1})$ – среднее N_{j+1} испытаний на $(j+1)$-ом слое. При переходе со слоя на слой мы будем иметь дело с двумя видами ошибок: случайной и детерминированной. При решении системы (3.12) используются несмещенные оценки, однако, в силу нелинейности функции $F(X)$, будет возникать смещение при переходе со слоя на слой. Возьмем математическое ожидание от правой части (3.15):

$$E(\widetilde{(I-A)}^{-1} F(\hat{\xi}_{j+1}(N_{j+1}))) = E(\widetilde{(I-A)}^{-1}) E(F(\hat{\xi}_{j+1}(N_{j+1}))) = \\ = (I-A)^{-1} E(F(\hat{\xi}_{j+1}(N_{j+1}))).$$

Первое равенство выполнено в силу того, что на разных слоях используются

независимые траектории. Рассмотрим разложение $F(X)$ в ряд Тейлора:

$$F(\hat{\xi}_{j+1}(N_{j+1})) =$$
$$= F\left(\frac{1}{N_{j+1}}\sum_{l=1}^{N_{j+1}}\xi_{j+1}^{(l)}\right) = F(E\xi_{j+1}) + F'(E\xi_{j+1})\left(\frac{1}{N_{j+1}}\sum_{l=1}^{N_{j+1}}\xi_{j+1}^{(l)} - E\xi_{j+1}\right) +$$
$$+\frac{1}{2}F''(E\xi_{j+1})\left(\frac{1}{N_{j+1}}\sum_{l=1}^{N_{j+1}}\xi_{j+1}^{(l)} - E\xi_{j+1}\right)^2 + \dots.$$

Возьмем математическое ожидание от левой и правой частей получившегося равенства, предварительно отметив, что $E\xi_{j+1}^{(l)} = E\xi_{j+1}$

$$E(F(\hat{\xi}_{j+1}(N_{j+1}))) = F(E(\xi_{j+1})) + F''(E\xi_{j+1})\frac{E\xi_{j+1}^2 - (E\xi_{j+1})^2}{2N_{j+1}} + \dots,$$

где $F''(X_{j+1})$ – диагональная матрица $(M-1)\times(M-1)$ с элементами

$$f_i'' = \frac{\Delta t\varepsilon C}{(1+i^2\sigma^2\Delta t + r\Delta t)(P_{i,j+1}+\varepsilon - K + i\Delta S)^3}, \quad i = 1,\dots,M-1.$$

В статье [50] было показано, что при $\Delta t \le \varepsilon/C$ решение уравнения (3.12) удовлетворяет неравенству

$$P_{i,j} \ge \max(K - S_i, 0) \quad \forall i, j.$$

Оценим f_i'':

$$f_i'' \le \frac{\Delta t\varepsilon C}{(1+i^2\sigma^2\Delta t + r\Delta t)(\max(K-S_i,0)+\varepsilon - K + i\Delta S)^3} \le$$
$$\le \frac{\Delta t C}{(1+i^2\sigma^2\Delta t + r\Delta t)\varepsilon^2} \le \frac{1}{(1+i^2\sigma^2\Delta t + r\Delta t)\varepsilon}.$$

Из этого следует, что при достаточно больших N_j (порядка $1/\varepsilon$ и больше) смещением можно будет пренебречь.

Перейдем к рассмотрению случайной ошибки. При переходе по времени со слоя на слой эта ошибка может оставаться ограниченной, а может неограниченно расти с ростом N. Поэтому необходимо исследовать её поведение. Обозначим $\mathcal{E}_j = \xi_j - X_j$ – вектор-столбец случайных ошибок на каждом слое. Выразим $cov\mathcal{E}_j$ через $cov\mathcal{E}_{j+1}$ в удобной для дальнейшего анализа форме. Из (3.15) следует

$$X_j + \mathcal{E}_j = \widetilde{(I-A)}^{-1}(F(X_{j+1}+\mathcal{E}_{j+1})). \tag{3.16}$$

Напишем разложение функции F в ряд Тейлора:

$$F(X_j + \mathcal{E}_j) = F(X_j) + F'(X_j)\mathcal{E}_j + F''(X_j)\mathcal{E}_j^2 + \dots.$$

Отметим, что при достаточно малых ε и Δt слагаемым $F''(X_j)\mathcal{E}_j^2$ и всеми последующими можно будет пренебречь. То есть далее считаем, что $F(X_j + \mathcal{E}_j) \approx F(X_j) + F'(X_j)\mathcal{E}_j$. Положим также $\widetilde{(I-A)}^{-1} = (I-A)^{-1} + \delta$, где $E\delta = 0$. Равенство (3.16) перепишется

$$X_j + \mathcal{E}_j = ((I-A)^{-1} + \delta)(F(X_{j+1}) + F'(X_{j+1})\mathcal{E}_{j+1}).$$

Учитывая, что X_j удовлетворяет (3.12), получим для $\mathcal{E}_j$ следующее выражение

$$\mathcal{E}_j = \delta F(X_{j+1}) + (I-A)^{-1}F'(X_{j+1})\mathcal{E}_{j+1} + \delta F'(X_{j+1})\mathcal{E}_{j+1} \tag{3.17}$$

и соответственно для $\mathcal{E}_j^T$, заметив прежде, что F' – диагональная матрица и следовательно $(F')^T = F'$,

$$\mathcal{E}_j^T = F(X_{j+1})^T\delta^T + \mathcal{E}_{j+1}^T F'(X_{j+1})((I-A)^{-1})^T + \mathcal{E}_{j+1}^T F'(X_{j+1})\delta^T. \tag{3.18}$$

Теперь необходимо перемножить левые и правые части равенств (3.17) и (3.18) соответственно и вычислить математическое ожидание всех членов получившегося равенства. При этом стоит отметить, что математическое ожидание многих слагаемых в правой части равны нулю. Так, $E(\delta F(X_{j+1})\mathcal{E}_{j+1}^T F'(X_{j+1})\delta^T) = 0$ в силу того, что $\mathcal{E}_{j+1}^T$ не зависит от δ и $E\mathcal{E}_{j+1}^T = 0$. Учитывая эти соображения, получим

$$\begin{aligned} E(\mathcal{E}_j\mathcal{E}_j^T) = (I-A)^{-1}F'(X_{j+1})E(\mathcal{E}_{j+1}\mathcal{E}_{j+1}^T)((I-A)^{-1}F'(X_{j+1}))^T + \\ + E(\delta F'(X_{j+1})\mathcal{E}_{j+1}\mathcal{E}_{j+1}^T F'(X_{j+1})\delta^T) + \mathcal{F}_{j+1}, \end{aligned}$$

где матрица $\mathcal{F}_{j+1}$ объединяет слагаемые, не зависящие от $\mathcal{E}_{j+1}$. Далее получим

$$\begin{aligned} cov\mathcal{E}_j = (I-A)^{-1}F'(X_{j+1})cov\mathcal{E}_{j+1}((I-A)^{-1}F'(X_{j+1}))^T + \\ + E(\delta F'(X_{j+1})cov\mathcal{E}_{j+1}F'(X_{j+1})\delta^T) + \mathcal{F}_{j+1}. \end{aligned} \tag{3.19}$$

Теперь было бы удобно привести получившееся равенство к виду $Y^j = BY^{j+1} + C$, где Y^j, Y^{j+1}, C – векторы, а B – матрица. Согласно определению операции умножения матриц, из (3.19) имеем

$$\begin{aligned} \|Y^j_{l,k}\|^{M-1}_{l,k=1} = \Big\| \sum_{i_1=1}^{M-1} \sum_{i_2=1}^{M-1} \alpha_{i_0,i_1} Y^{j+1}_{i_1,i_2} \alpha_{i_3,i_2} \Big\|^{M-1}_{i_0,i_3=1} + \\ + E \Big\| \sum_{i_1=1}^{M-1} \sum_{i_2=1}^{M-1} \beta_{i_0,i_1} Y^{j+1}_{i_1,i_2} \beta_{i_3,i_2} \Big\|^{M-1}_{i_0,i_3=1} + \mathcal{F}_{j+1}, \end{aligned} \tag{3.20}$$

где $\alpha_{l,k}$ – элементы матрицы $(I - A)^{-1} F'(X_{j+1})$, $Y^j_{l,k}$ – матрицы $\mathcal{E}_j$ и $\beta_{l,k}$ – матрицы $\delta F'(X_{j+1})$. В силу того, что матрица F' диагональная, элементы $\delta F'(X_{j+1})$ будут иметь вид: $\beta_{l,k} = f'_k \delta_{l,k}$, где f'_k – элемент диагонали матрицы F', а $\delta_{l,k}$ – элементы матрицы δ. Если теперь ввести мультииндекс $L = (l, k)$ принимающий $(M-1)^2$ значений от $(1,1)$ до $(M-1, M-1)$, то равенство (3.20) перепишется

$$\begin{aligned} \|Y^j_L\|^{(M-1,M-1)}_{L=(1,1)} = \Big\| \sum_{L=(i_1,i_2)} \alpha_{i_0,i_1} \alpha_{i_3,i_2} Y^{j+1}_L \Big\| + \\ + \Big\| \sum_{L=(i_1,i_2)} E(\delta_{i_0,i_1} \delta_{i_3,i_2}) f'_{i_1} f'_{i_2} Y^{j+1}_L \Big\| + \mathcal{F}_{j+1}. \end{aligned} \tag{3.21}$$

Теперь вытянем матрицу $cov\mathcal{E}_j$ в столбец и будем рассматривать далее $\|Y^j_L\|$ как вектор длинны $(M-1)^2$, тогда последнее равенство (3.21) перепишется в искомой нами форме

$$\|Y^j_L\| = (\mathcal{A} + \mathcal{D}) \|Y^{j+1}_L\| + \mathcal{F}_{j+1}, \tag{3.22}$$

где $\mathcal{A} = \|\alpha_{i_0,i_1} \alpha_{i_3,i_2}\|$, $\mathcal{D} = \|f'_{i_1} E(\delta_{i_0,i_1} \delta_{i_3,i_2}) f'_{i_2}\|$ – матрицы $(M-1)^2 \times (M-1)^2$, i_0, i_1, i_2, i_3 меняются от 1 до $M-1$. Покажем, что собственные векторы $\mathcal{A}$ есть $\varphi_i \varphi_j^T$, $i, j = 1, \ldots, M-1$, где φ_i – собственный вектор-столбец матрицы $(I - A)^{-1} F'$, а собственные числа $\mathcal{A}$ есть $\lambda_i \lambda_j$, $i, j = 1, \ldots, M-1$, где λ_i –

собственные числа матрицы $(I-A)^{-1}F'$. Пусть φ_i, φ_j – собственные векторы $(I-A)^{-1}F'$ и λ_i, λ_j – соответствующие им собственные числа. Вытянем матрицу $\varphi_i\varphi_j^T=\phi$ в один столбец и рассмотрим $\mathcal{A}\phi$ – вектор длинны $(M-1)^2$

$$\mathcal{A}\phi=\Big\|\sum_{(i_1,i_2)=(1,1)}^{(M-1,M-1)}\alpha_{i_0,i_1}\alpha_{i_3,i_2}\varphi_i^{i_1}\varphi_j^{i_2}\Big\|_{(i_0,i_3)=(1,1)}^{(M-1,M-1)}. \tag{3.23}$$

Правую часть можно мыслить как матрицу $(M-1)\times(M-1)$, а именно

$$\Big\|\sum_{i_1=1}^{M-1}\sum_{i_2=1}^{M-1}\alpha_{i_0,i_1}\alpha_{i_3,i_2}\varphi_i^{i_1}\varphi_j^{i_2}\Big\|_{i_0,i_3=1}^{M-1}=(I-A)^{-1}F'\varphi_i\varphi_j^T((I-A)^{-1}F')^T=$$

$$=((I-A)^{-1}F'\varphi_i)((I-A)^{-1}F'\varphi_j)^T=\lambda_i\lambda_j\varphi_i\varphi_j^T.$$

Если снова растянуть $\varphi_i\varphi_j^T$ в столбец, получим $\mathcal{A}\phi=\lambda_i\lambda_j\phi$. Что и требовалось доказать. Рассмотрим теперь матрицу $\mathcal{D}$. В действительности $\mathcal{D}$ представляет собой произведение $\mathcal{M}\hat{F}$, где $\hat{F}$ – диагональная матрица с диагональю $\|f_i'f_j'\|_{(i,j)=(1,1)}^{(M-1,M-1)}$, а элементы матрицы $\mathcal{M}$ есть ковариации вектора, составленного из элементов матрицы погрешностей δ. Элементы матрицы $\mathcal{M}$ имеют порядок $1/N_j$, где N_j – количество независимых траекторий на слое с номером j. Оценим f_i', выражение для которого имеет вид

$$f_i'=\frac{1}{1+i^2\sigma^2\Delta t+r\Delta t}\left(1-\frac{\Delta t\varepsilon C}{(P_{i,j+1}+\varepsilon-K+i\Delta S)^2}\right)$$

найдем условие на Δt, при котором разность, стоящая в скобках, больше нуля

$$1-\frac{\Delta t\varepsilon C}{(P_{i,j+1}+\varepsilon-K+i\Delta S)^2}\geq 1-\frac{\Delta t\varepsilon C}{(\varepsilon-K)^2}\geq 0$$

откуда получаем условие

$$\Delta t\leq\frac{(\varepsilon-K)^2}{\varepsilon C}. \tag{3.24}$$

Таким образом норма матрицы F' меньше единицы при Δt из (3.24). Учитывая малость ε и то, что $C\geq rK$, это условие на Δt является несущественным. Заметим, что вообще говоря, матрица F' зависит от момента времени, поэтому в действительности $F'=F_j'$.

Из (3.22) и вышесказанного следуют следующие утверждения:

Утверждение 1. *Для стохастической устойчивости алгоритма (3.15) необходимо и достаточно, чтобы максимум из модулей первых собственных чисел матриц* $\mathcal{L} + \frac{1}{N_j}\mathcal{M}'$, $j = N-1 \dots 1$ *при любых натуральных* N *был меньше единицы.*

Утверждение 2. *Если первые собственные числа матриц* $(I-A)^{-1}F'_j$ *по модулю строго меньше единицы, то существует такой набор* N_j, $j = N-1, \dots 1$, *что при всех* $N_k > N_j$ *алгоритм (3.15) будет стохастически устойчивым.*

В силу того, что норма матрицы A меньше единицы, то матрица $(I-A)^{-1}$ допускает следующее представление

$$(I-A)^{-1} = \sum_{i=0}^{\infty} A^i, \quad A^0 = I. \tag{3.25}$$

При фиксированном ΔS и малых Δt будет выполнено $(I-A)^{-1} \approx I + A$. Из этого следует следующее утверждение

Утверждение 3. *Если* $\max_j(\rho((I+A)F'_j)) < 1$, *где* $\rho(\cdot)$ – *спектральный радиус, то существует такой набор* N_j, $j = N-1, \dots 1$, *что при всех* $N_k > N_j$ *алгоритм (3.15) будет стохастически устойчивым.*

Таким образом получены достаточные условия применимости и стохастической устойчивости алгоритма метода Монте-Карло для решения задач нахождения цены Американских опционов методом штрафной функции.

Важным замечанием является то, что метод Монте-Карло обладает свойством естественного параллелизма. Если число испытаний N_1 такое, что алгоритм стохастически устойчив (дисперсия не растет экспоненциально), то можно осуществить N_2 моделирований с N_1 повторениями каждое на разных процессорах и результат осреднить.

3.5. Численные эксперименты

Ниже приведены результаты решения систем (3.12) методом Монте-Карло и детерминированным методом. Расчеты производились для следующих параметров: $T = 0.75$, $S_\infty = 100$, $M = 100$, $N = 700$, $\sigma \equiv 0.15$, $r \equiv 0.055$, $K = 35$, $\varepsilon = 0.001$, $C = 2$. На графиках изображена цена опциона в зависимости от ценны акции в момент времени $t = 0$. Количество моделируемых траекторий: 1000 – для рисунка 3.1, 5000 – для рисунка 3.2, 9000 – для рисунка 3.3.

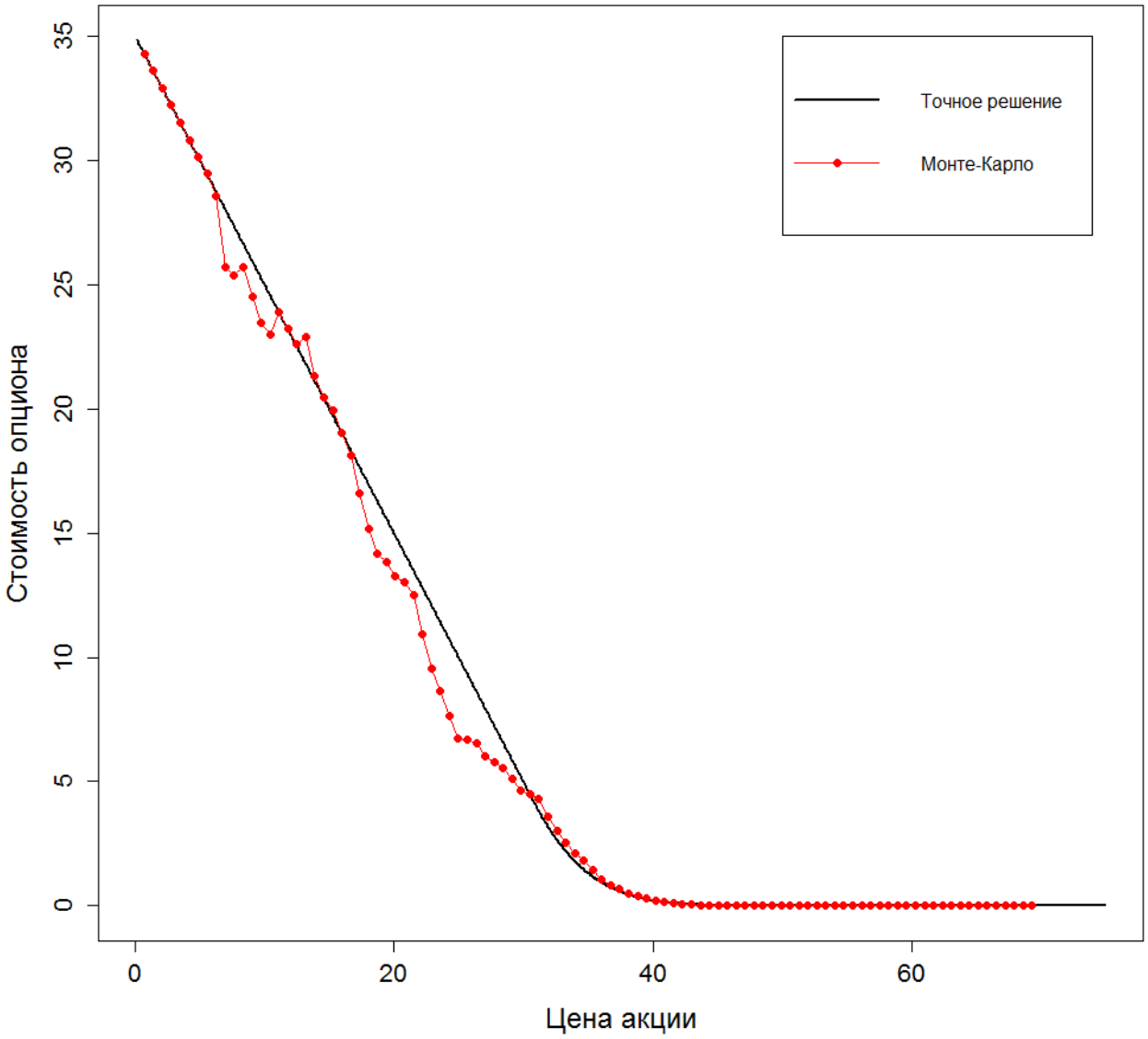

Рис. 3.1. Оценка стоимости опциона. Кол-во траекторий – 1000

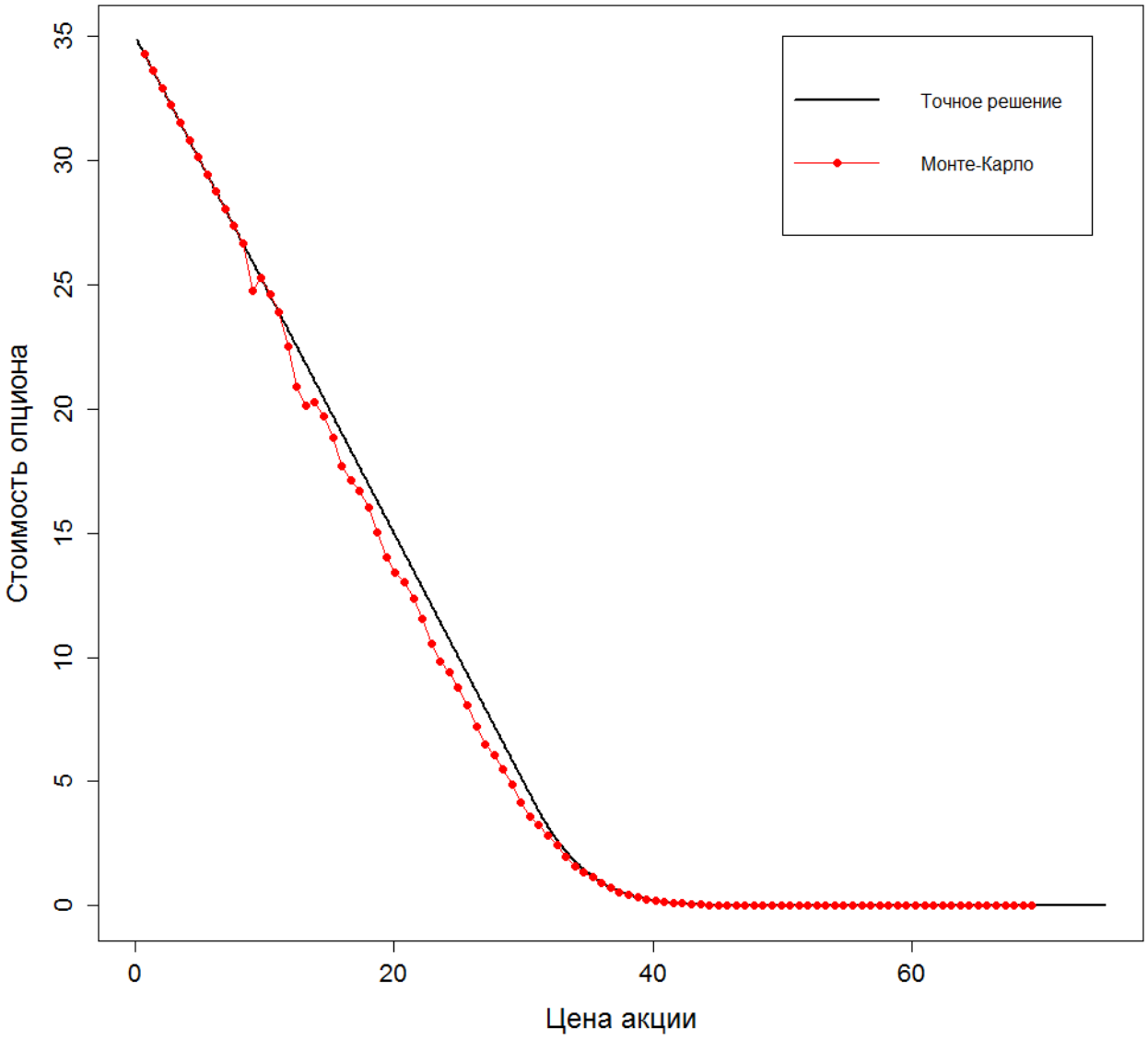

Рис. 3.2. Оценка стоимости опциона. Кол-во траекторий – 5000

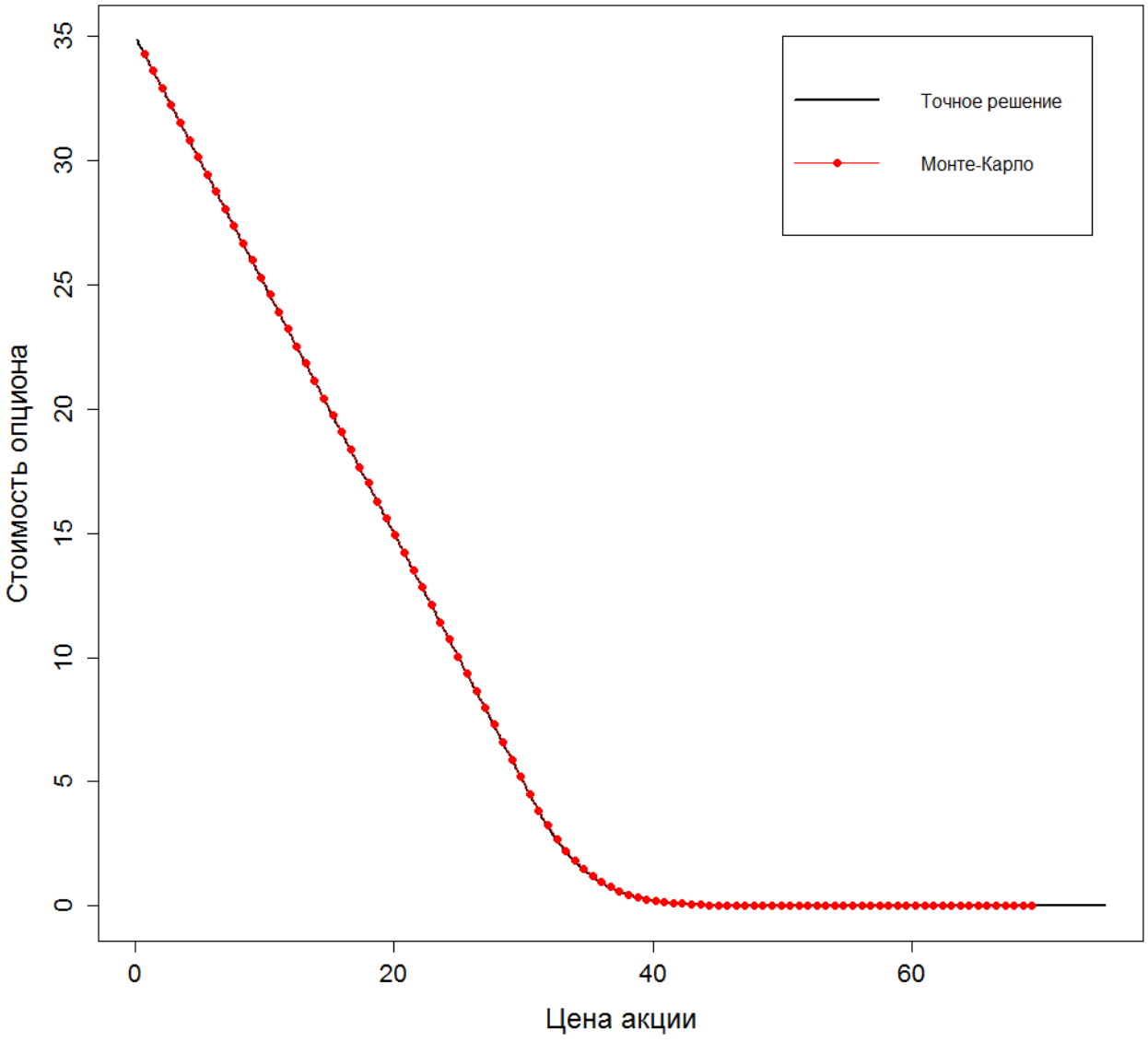

Рис. 3.3. Оценка стоимости опциона. Кол-во траекторий – 9000

Заключение

При решении систем уравнений большой размерности, к которым нередко сводятся прикладные задачи, неизменно упоминаются многопроцессорные системы, являющиеся важным инструментом быстрого поиска решения таких систем уравнений. Координирование действий процессоров при этом является одним из ключевых вопросов. Асинхронные алгоритмы могут значительно упростить координацию и одновременно с этим эффективно использовать доступные вычислительные ресурсы. Исследованию таких алгоритмов, а именно методу Монте-Карло и методу асинхронных итераций, для решения систем уравнений и было посвящено диссертационное исследование.

В диссертации построен алгоритм метода Монте-Карло с частичной синхронизацией для решения систем уравнений вида

$$x = Ax + b, \tag{3.26}$$

при выполнении условий $|\lambda_1(A)| < 1$ и $\lambda_1(|A|) > 1$, где $\lambda_1(\cdot)$ – наибольшее по модулю собственное число матрицы, а $|A|$ – матрица, составленная из модулей элементов матрицы A. Получены достаточные условия стохастической устойчивости предложенного алгоритма и оценен период асинхронности.

Модифицирован метод асинхронных итераций для решения задачи (3.26) при условии $|\lambda_1(A)| < 1$ и $\lambda_1(|A|) > 1$. Получены оценки периода асинхронности.

Получены и формально описаны оценки метода Монте-Карло, обладающие свойством асинхронности, для решения систем обыкновенных дифференциальных уравнений большой размерности. Получены достаточные условия их стохастической устойчивости.

Построены асинхронные оценки метода Монте-Карло для нахождения стоимости американского опциона, исследованы условия их стохастической устойчивости.

Полученные результаты позволяют решать широкий спектр прикладных задач из различных областей науки, предоставляя при этом возможность эффективно использовать многопроцессорные вычислительные системы, что в обстановке непрерывного развития вычислительной техники выгодно выделяет исследованные в диссертации методы и алгоритмы. Кроме того, указанные теоретические результаты могут послужить основой для дальнейших исследований асинхронных методов вычислений.

Литература

1. Chazan D., Miranker W. Chaotic relaxation // Linear Algebra and its Applications. 1969. Vol. 2, no. 2. P. 199–222.

2. Baudet G. M. Asynchronous Iterative Methods for Multiprocessors // J. ACM. 1978. Vol. 25, no. 2. P. 226–244.

3. Ермаков С.М. Метод Монте-Карло и смежные вопросы. Москва: Наука, 1975. С. 472.

4. Ермаков С.М. Метод Монте-Карло в вычислительной математике (Вводный курс). Невский Диалект, Бином. Лаборатория знаний, 2009. С. 192.

5. Ермаков С.М. Параметрически разделимые алгоритмы // Вестник СПбГУ, Сер.1, вып. 4,. 2010. С. 25–31.

6. Ермаков С.М., Михайлов Г.А. Статистическое моделирование. М.:Наука, 1982. С. 296.

7. Михайлов Г.А., Войтишек А.В. Численное статистическое моделирование. Методы Монте-Карло. Академия, 2006. С. 368.

8. Михайлов Г.А. Оптимизация весовых методов Монте-Карло. М.:Наука, 1987. С. 240.

9. Михайлов Г.А. Весовые методы Монте-Карло. Новосибирск: Изд-во СО РАН, 2000. С. 248.

10. Halton J. H. A retrospective and prospective survey of the Monte Carlo method // SIAM Review. 1970. Vol. 12, no. 1. P. 1–63.

11. Halton J. H. Sequential Monte Carlo techniques for the solution of linear systems // Journal of Scientific Computing. 1994. Vol. 9, no. 2. P. 213–257.

12. Halton J. H. Sequential Monte Carlo techniques for solving non-linear systems // Monte Carlo Methods and Applications. 2006. Vol. 12, no. 2. P. 113–141.

13. Дмитриев А.В., Ермаков С.М. Параллельный Монте-Карло метод оценки американских опционов // Вестник СПбГУ, Серия 1, Выпуск 1. 2013. С. 72–82.

14. Дмитриев А.В., Ермаков С.М. Метод Монте-Карло и асинхронные итерации // Вестник СПбГУ, Серия 1, Том 1 (59) Выпуск 4. 2014. С. 517–528.

15. Дмитриев А.В., Ермаков С.М. О частичной синхронизации итерационных методов // Вестник СПбГУ. 2016. Т. 3 (61), № 3. С. 393–401.

16. Bertsekas D. P., Tsitsiklis J. N. Parallel and Distributed Computation: Numerical Methods. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989. ISBN: 0-13-648700-9.

17. Verjus J.-P., Herman D. c. e. i., Andre F., Howlett J. Synchronization of parallel programs. Oxford, GB: North Oxford Academic, 1985. ISBN: 0-946536-20-1. Trad. de Synchronisation de programmes paralleles, Dunod, 1983.

18. Quinn M. J. Designing Efficient Algorithms for Parallel Computers. New York, NY, USA: McGraw-Hill, Inc., 1986. ISBN: 0-070-51071-7.

19. Baudet G. M. The design and analysis of algorithms for asynchronous multiprocessors. 1978. no. CMU-CS-78-116.

20. Ортега Дж., Рейнболдт В. Итерационные методы решения нелинейных систем уравнений со многими неизвестными. Москва: Мир, 1975.

21. Miellou J.-C. Algorithmes de relaxation chaotique a retards. 1975. no. 9 R-1. P. 55–82.

22. Miellou J.-C. Iterations chaotiques a retards; etudes de la convergence dans le cas d'espaces partiellement ordonnes. 1975. no. A 280. P. 233–236.

23. Ökten G. Solving Linear Equations by Monte Carlo Simulation // SIAM J. Sci. Comput. 2005. Vol. 27, no. 2. P. 511–531.

24. Ермаков С.М. Метод Монте Карло и асинхронные вычисления // Тезисы 1-ой международной конференции общества Бернулли. Т. 6. 1987. С. 462.

25. Ланкастер П. Теория матриц. Наука, 1973. С. 282.

26. Гантмахер Ф.Р. Теория матриц. М.: Наука, 1967. С. 576.

27. Медведев И.Н., Михайлов Г.А. Исследование весовых алгоритмов метода Монте-Карло с ветвлением // Журнал вычислительной математики и математической физики. 2009. Т. 49, № 3. С. 441–452.

28. Ермаков С.М. Об аналоге схемы Неймана-Улама в нелинейном случае // Журнал вычислительной математики и математической физики. 1973. Т. 13, № 3. С. 564–573.

29. Севастьянов Б. А. Теория ветвящихся случайных процессов // УМН. 1951. Т. 6. С. 47–99.

30. Athreya K. B., Ney P. Branching Processes. Springer-Verlag, NewYork–Heidelberg, 1972.

31. Зеликин М. И. Однородные пространства и уравнение Риккати в вариационном исчислении. М.: Факториал, 1998. С. 351.

32. Reid W. T. Riccati Differential Equations. New York: Academic Press, 1972. P. 216.

33. Butcher J. Numerical Methods for Ordinary Differential Equations. New York: John Wiley & Sons, 2008. P. 482.

34. Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы. М.: Бином. Лаборатория знаний, 2003. С. 632.

35. Михайлов Г.А. Минимаксные методы Монте-Карло для решения интегральных уравнений II рода // Журнал вычислительной математики и математической физики. 1989. Т. 29, № 11. С. 1650–1661.

36. Понтрягин Л. С. Обыкновенные дифференциальные уравнения. Наука, 1974. С. 331.

37. Harris T. E. The theory of branching processes: Tech. rep. Berlin, Gottingen, Heidelberg: 1963.

38. Тихонов А.Н., Самарский А.А. Уравнения математической физики. М.: Изд-во МГУ, 2004. С. 798.

39. Владимиров В.С. Уравнения математической физики. М.: Наука, 1981. С. 512.

40. Ermakov S. M., Wagner W. Monte Carlo difference schemes for the wave equation // Monte Carlo Methods and Applications. 2002. Vol. 8, no. 1. P. 1–30.

41. Mitra D. Asynchronous Relaxations for the Numerical Solution of Differential Equations by Parallel Processors // SiAM J. Sci. Stat. Comput. 1987. Vol. 8. P. 43–58.

42. Hull J. Options, Futures and Other Derivatives. Pearson/Prentice Hall, 2009. P. 822.

43. Brennan M., Schwartz E. The Valuation of American Put Options // The Journal of Finance. 1977. Vol. 32, no. 2. P. 449–462.

44. Lengwiler Y. Microfoundations of Financial Economics: An Introduction to General Equilibrium Asset Pricing. Princeton Series in Finance, 2004.

45. Bjork T. Arbitrage Theory in Continuous Time. Oxford University Press, 1999. P. 560.

46. Merton R. Theory of Rational Option Pricing // The Bell Journal of Economics and Management Science. 1973. Vol. 4, no. 1. P. 141–183.

47. Люу Ю-Д. Методы и алгоритмы финансовой математики. М.: Бином. Лаборатория знаний, 2007. С. 751.

48. Duffy D. Finite difference methods in financial engineering: a partial differential equation approach. John Wiley & Sons Ltd, 2006. P. 423.

49. Black F., Scholes M. The Pricing of Options and Corporate Liabilities // The Journal of Political Economy. 1973. Vol. 81, no. 3. P. 637–654.

50. Nielson B.F, Skavhaug O., Tvelto A. Penalty and front-fixing methods for the numerical solution of American option problems // J. Comp. Finan. 2002. no. 4.

51. Crank J. Free and Moving Boundary Problems. New York: Oxford University Press, 1987. P. 424.

52. Zvan R., Forsyth P., Vetzal K. Penalty methods for american options with stochastic volatility // Journal of Computational and Applied Mathematics. 1998. no. 218. P. 91–199.