

МЕТОДЫ СТАТИСТИЧЕСКОГО МОДЕЛИРОВАНИЯ (метод Монте-Карло)

Имитационное моделирование представляет собой численный метод проведения на ЭВМ вычислительных экспериментов с математическими моделями, имитирующими поведение реальных объектов, процессов и систем во времени в течение заданного периода. При этом функционирование таких объектов и процессов разбивается на элементарные явления, подсистемы и модули. Функционирование этих элементарных явлений, подсистем и модулей описывается набором алгоритмов, которые имитируют элементарные явления с сохранением их логической структуры и последовательности.

При исследовании сложных систем, подверженных случайным возмущениям используются вероятностные имитационные модели, в которых влияние случайных факторов учитывается с помощью задания вероятностных характеристик случайных процессов (законы распределения вероятностей, спектральные плотности или корреляционные функции). При этом результаты, полученные при воспроизведении на имитационной модели рассматриваемого процесса, являются случайными реализациями. Поэтому для нахождения объективных и устойчивых характеристик процесса требуется его многократное воспроизведение, с последующей статистической обработкой полученных данных. Именно поэтому **исследование сложных процессов и систем, подверженных случайным возмущениям, с помощью имитационного моделирования принято называть статистическим моделированием.**

Статистическая модель случайного процесса – это алгоритм, с помощью которого имитируют работу сложной системы, подверженной случайным возмущениям; имитируют взаимодействие элементов системы, носящих вероятностный характер. Тогда статистическое моделирование можно определить как способ изучения сложных процессов и систем, подверженных случайным возмущениям, с помощью имитационных моделей.

Методика статистического моделирования, который часто называют методом Моне-Карло, состоит из следующих этапов:

1. Моделирование на ЭВМ псевдослучайных последовательностей с заданной корреляцией и законом распределения вероятностей (метод Монте-Карло), имитирующих на ЭВМ случайные значения параметров при каждом испытании;
2. Использование полученных числовых последовательностей в имитационных математических моделях.
3. Статистическая обработка результатов моделирования.

Проиллюстрируем суть метода Монте-Карло относительно простым примером. Пусть требуется оценить надежность системы (рис.1).

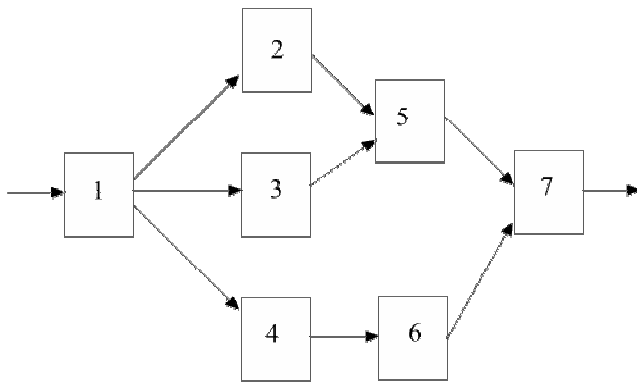


Рис.1. Блочная структура системы

$p_i(\tau_i), i = \overline{1,7}$ Какова надежность системы в целом?

Рассмотрим случайную величину

$$\gamma = \min \{ \tau_1, \max [\min (\tau_4, \tau_6), \min [\max (\tau_2, \tau_3)], \tau_7] \}, \quad (1)$$

где γ – время безотказной работы системы. В одном опыте разыгрываются значения всех $\tau_i, i = \overline{1,7}$ в соответствии с $p_i(\tau_i), i = \overline{1,7}$ Используя полученные реализации $\tau_i, i = \overline{1,7}$, по (1) вычисляем реализацию γ . Один опыт дает одну реализацию (одно выборочное значение) γ . Проводим M опытов (испытаний), получаем “статистический” материал (выборку). Берем среднее арифметическое времени безотказной работы системы γ с p в качестве оценки надежности системы. При необходимости можно построить закон распределения вероятностей случайной величины γ в виде соответствующей гистограммы.

Таким образом, испытания реальной системы заменены на испытания математической модели. Каждое испытание сопровождается расчетом. Поэтому имитационное моделирование и называют численным экспериментом на ЭВМ с математической моделью (модель выступает как объект исследования).

Метод Монте-Карло можно определить как метод моделирования случайных величин с целью вычисления характеристик их распределений. Как правило, предполагается, что моделирование осуществляется с помощью электронных вычислительных машин (ЭВМ), хотя в некоторых случаях можно добиться успеха, используя приспособления типа рулетки, карандаша и бумаги.

Термин "метод Монте-Карло" (предложенный Дж. Фон Нейманом и С. М. Уламу в 1940-х) относится к моделированию процессов с использованием генератора случайных чисел. Термин Монте-Карло (город, широко известный своими казино) произошел от того факта, что "число шансов" (методы моделирования Монте-Карло) было использовано с целью нахождения интегралов от сложных уравнений при разработке первых

Система выполняет свою функцию, если работают цепочки блоков: 1,2,5,7; 1,3,5,7; 1,4,6,7. Какие-то блоки могут отказать. Каждый блок характеризуется временем безотказной работы $\tau_i, i = \overline{1,7}$. Пусть заданы плотности распределения

ядерных бомб (интегралы квантовой механики). С помощью формирования больших выборок случайных чисел из, например, нескольких распределений, интегралы этих (сложных) распределений могут быть аппроксимированы из (сгенерированных) данных.

Возникновение идеи использования случайных явлений в области приближенных вычислений принято относить к 1878 г., когда появилась работа Холла об определении чисел π с помощью случайных бросаний иглы на разграфленную параллельными линиями бумагу. Сущность дела заключается в том, чтобы экспериментально воспроизвести событие, вероятность которого выражается через число π , и приближенно оценить эту вероятность.

Первоначально метод Монте-Карло использовался главным образом для решения задач нейтронной физики, где традиционные численные методы оказались мало пригодными. Далее его влияние распространилось на широкий класс задач статистической физики, очень разных по своему содержанию. К разделам науки, где все в большей мере используется метод Монте-Карло, следует отнести задачи теории массового обслуживания, задачи теории игр и математической экономики, задачи теории передачи сообщений при наличии помех и ряд других.

Метод Монте-Карло оказал и продолжает оказывать существенное влияние на развитие метода вычислительной математики (например, развитие методов численного интегрирования) и при решении многих задач успешно сочетается с другими вычислительными методами и дополняет их. Его применение оправдано в первую очередь в тех задачах, которые допускают теоретико-вероятностное описание. Это объясняется как естественностью получения ответа с некоторой заданной вероятностью в задачах с вероятным содержанием, так и существенным упрощением процедуры решения. В тех случаях, когда имеется теоретико-вероятностное описание задачи, использование метода Монте-Карло может существенно упростить упомянутые ее решение. Впрочем, как будет следовать из дальнейшего, во многих случаях полезно и для задач строго детерминированных строить вероятностную модель (рандомизовать исходную задачу) с тем, чтобы далее использовать метод Монте-Карло.

2. Общая схема метода Монте-Карло

Предположим, что нам требуется вычислить некоторую неизвестную величину m , и мы хотим сделать это, рассматривая случайную величину ξ такую, что ее математическое ожидание $M\xi = m$. Пусть при этом дисперсия данной случайной величины $D\xi = b$.

Рассмотрим N случайных независимых величин $\xi_1, \xi_2, \dots, \xi_N$, распределения которых совпадают с распределением рассматриваемой случайной величины ξ . Согласно центральной предельной теореме теории вероятностей, распределение суммы $\rho_N = \xi_1 + \xi_2 + \dots + \xi_N$ будет приблизительно

нормальным со средним, равным $\mu = Nm$, и дисперсией $\sigma^2 = Nb^2$. Согласно «правилу трех сигма», каковы бы ни были m и σ ,

$$P\{\mu - 3\sigma < P_N < \mu + 3\sigma\} = 0.997,$$

т. е.

$$P\left\{m - \frac{3b}{\sqrt{N}} < \frac{P_N}{N} < m + \frac{3b}{\sqrt{N}}\right\} \approx 0.997$$

Последнее соотношение можно переписать в виде

$$P\left\{\frac{1}{N} \sum_{j=1}^N \xi_j - m < \frac{3b}{\sqrt{N}}\right\} \approx 0.997$$

Полученная формула дает метод расчета m и оценку погрешности этого метода.

Сущность применения метода Монте-Карло заключается в определении результатов на основании статистики, получаемой к моменту принятия некоторого решения. Поэтому достоверность результатов, получаемых при использовании метода Монте-Карло, решающим образом определяется качеством генератора случайных чисел.

Для получения случайных чисел на ЭВМ используются способы генерирования, которые обычно основаны на много кратном повторении некоторой операции. Полученной таким образом последовательности более соответствует название псевдослучайных чисел, поскольку генерируемая последовательность является периодичной и, начиная с некоторого момента, числа начнут повторяться.

Для языка программирования высокого уровня, например C++, часто используют следующий алгоритм:

1. В начале программы целой переменной X присваивается некоторое значение X_0 . Затем случайные числа генерируются по правилу

$$X = (a X) \bmod m.$$

Например, можно принять $a = 48271$ и $m = 2^{31} - 1$.

3. Вычисление определенных интегралов

Статистический метод вычисления интегралов основан на вычислении математических ожиданий некоторых случайных величин, поскольку эти математические ожидания представляют собой обычные интегралы. Пусть $z = f(\xi)$ неслучайная функция от непрерывного случайного аргумента ξ определена на отрезке $[0,1]$. Тогда ее математическое ожидание $M[z]$ есть

$$M[z] = I = \int_0^1 f(\xi)p(\xi)d\xi,$$

где $p(\xi)$ - плотность вероятности распределения случайной величины на данном отрезке. Если случайная величина ξ распределена равномерно, то $p(\xi) \equiv 1$

и тогда выражение для $M[z]$ есть обыкновенный интеграл. Если ξ является дискретной равномерно распределенной случайной величиной, т.е. может принимать значения $\xi_1, \xi_2, \dots, \xi_n$, то в этом случае $M[z] = \bar{z}_n = \frac{1}{n} \sum_{i=1}^n f(\xi_i)$ и

тогда приближенно

$$\int_0^1 f(\xi) d\xi \approx \frac{1}{n} \sum_{i=1}^n f(\xi_i).$$

Согласно центральной предельной теореме разность между левой и правой частями этого приближенного равенства стремится к нулю по вероятности при $n \rightarrow \infty$.

Если дисперсия $D[z] = \sigma^2 = M[(M[z] - z)^2]$ конечная, то при достаточно большом числе испытаний n ($n > 10$) имеет место правило «трех сигм»

$$P(|I - \bar{z}_n| < \frac{3\sigma}{\sqrt{n}}) \approx 0.997,$$

т.е. неравенство $|I - \bar{z}_n| < \frac{3\sigma}{\sqrt{n}}$ выполняется с вероятностью 0.997. Если σ неизвестно, то при $n > 10$ ее можно приближенно оценить по формуле

$$\sigma \approx \sqrt{\frac{1}{n} \sum_{i=1}^n (z_i - \bar{z}_n)^2}.$$

Таким образом, если ξ - равномерно распределенная на $[0,1]$ дискретная случайная величина, то

$$\int_0^1 f(x) dx \approx \bar{z}_n = \frac{1}{n} \sum_{i=1}^n f(\xi_i),$$

где ξ_i независимые реализации случайной величины ξ .

Кратные интеграла методом Монте-Карло вычисляются аналогично

$$\int_0^1 \int_0^1 f(x, y) dx dy \approx \frac{1}{n} \sum_{i=1}^n f(\eta_i, \mu_i),$$

где η_i, μ_i - независимые реализации случайных величин, равномерно распределенных на $[0,1]$. Заметим, что произвольный отрезок интегрирования $[a, b]$ приводится к единичному отрезку простым линейным преобразованием $\zeta = (x - a)/(b - a)$.

Если погрешность квадратурной формулы оценить как $\varepsilon \approx C_1 N^{-m}$, где m - точность формулы, N - число узлов, то для заданной точности ε требуется $N \approx C_2 \varepsilon^{-1/m}$ итераций, а для k -мерного интеграла $N \approx C_2 \varepsilon^{-k/m}$. В методе же Монте-Карло $\varepsilon \approx \frac{3\sigma}{\sqrt{n}}$, т.е. требуемое число испытаний $n \approx C_3 \varepsilon^{-2}$ и не зависит

от кратности интеграла. Поэтому, несмотря на более низкую точность, методы Монте-Карло широко применяются при вычислении интегралов высокой кратности.

Основным недостатком метода Монте-Карло является вероятностный характер результата, т.е. отсутствие строгих, стремящихся к нулю при $n \rightarrow \infty$ оценок погрешности.

4. Пример расчета системы массового обслуживания методом Монте-Карло

Рассмотрим простейшую систему массового обслуживания (СМО), которая состоит из n линий (иначе называемых каналами или пунктами обслуживания). В случайные моменты времени в систему поступают заявки. Каждая заявка поступает на линию № 1. Если в момент поступления заявки T_k эта линия свободна, заявка обслуживается время t_3 (время занятости линии). Если линия занята, заявка мгновенно передается на линию № 2 и т. д. Если все n линий в данный момент заняты, то система выдает отказ.

Естественной является задача определения характеристик данной системы, по которым можно оценить ее эффективность: среднее время ожидания обслуживания, доля времени простоя системы, среднюю длину очереди и т. д.

Для подобных систем практически единственным методом расчета является метод Монте-Карло.

Среднее число заданий, поступающих в систему обслуживания за единицу времени, называется **интенсивностью поступления заданий** и определяется следующим соотношением:

$$\lambda = \frac{1}{T},$$

где T - среднее значение интервала между поступлением очередных заданий. Для многих реальных процессов, как показали многочисленные исследования в этой области, поток заданий достаточно хорошо описывается законом распределения Пуассона. В этом случае функция распределения вероятности имеет вид $P(t) = 1 - \exp(-\lambda t)$, $0 \leq t < \infty$, а ее функция плотности распределения есть

$$p(t) = \lambda \exp(-\lambda t) .$$

При расчете времени обслуживания и времени поступления очередного задания в систему необходимо решать интегральные уравнения вида

$$\int_0^{\tau} p(t) dt = \gamma, \tag{4}$$

где τ – искомый момент времени, $p(t)$ – функция плотности вероятности, γ – случайная величина, равномерно распределенная на отрезке $[0, 1]$. Если

распределение является Пуассоновским, то интеграл (4) вычисляется аналитически и величина τ определяется следующим образом

$$\tau = -(1/\lambda) \ln(\gamma). \quad (2)$$

где случайная величина γ распределена равномерно.

Каждой линии ставится в соответствие ячейка ЭВМ, в которую записывается момент освобождения линии. Пусть в момент времени $t = T_1$, который мы примем за начало отсчета, все линии свободны. За время окончания расчета примем $T = T_{\text{кон}}$.

Первая заявка поступает на линию № 1 и тут же обслуживается, поскольку линия свободна. Следовательно, в течение времени τ_1 эта линия будет занята. Поэтому заменяем t на новое значение:

$t_1 \leftarrow T_1 + \tau_1$ и добавляем 1 к счетчику выполненных заявок. Затем переходим к рассмотрению следующей заявки. Для этого генерируем значение γ , равномерно распределенное на $[0, 1]$, и по формуле (2) вычисляем очередное значение $\tau = \tau_2$. Вычисляем момент поступления второй заявки:

$$t_2 = T_1 + \tau_2.$$

Проверяем, свободна ли в этот момент первая линия, т.е. выполнено ли условие

$$t_1 \leq t_2.$$

Если это условие выполнено, линия свободна и приступает к обслуживанию второй заявки. Заменяем t_1 на $t_2 + \tau_3$, добавляем 1 к счетчику выполненных заявок и переходим к рассмотрению следующей заявки.

Если линия занята, то проверяем, свободна ли вторая линия, и т. д. Если в какой-то момент времени заняты все линии, добавляем 1 в счетчик отказов и переходим к выполнению следующей заявки.

После каждого вычисления t_k проверяем условие окончания опыта: $t_k > T_{\text{кон}}$.

Если это условие выполнено, опыт заканчивается. Такой опыт повторяется N раз, и результаты всех опытов усредняются.

Аналогично можно рассматривать более сложные задачи. Например, величина t_3 может быть случайной или различной для различных линий, что позволяет отразить различие в мощности оборудования и/или квалификации обслуживающего персонала.

Подобные расчеты могут быть полезны при решении вопроса об увеличении линий, необходимости повышения квалификации персонала и т. д.

5. Решение задачи Дирихле для двумерного уравнения Лапласа методом «блужданий по сферам».

Требуется найти численное решение задачи Дирихле (первая краевая задача) для уравнения Лапласа

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \quad (1)$$

в произвольной точке прямоугольной области $D\{0 \leq x \leq 1, 0 \leq y \leq 2\}$ с граничными условиями на границе $\Gamma: u(s) = g(s)$, где s – длина дуги границы Γ , отсчитываемая от некоторой начальной точки.

Алгоритм метода Монте-Карло здесь следующий. Для выбранной точки $P_0(x_0, y_0) \in D$ строится случайная траектория $Q_0 \rightarrow Q_1 \rightarrow \dots \rightarrow Q_n \rightarrow \dots$ следующим образом: положим $Q_0 = P_0$, далее, если точка Q_n известна, то строится окружность произвольного радиуса R_n , целиком расположенная внутри D , и на этой окружности выбирается случайная точка $P_{n+1}(x_n + R_n \cos(\varphi_n), y_n + R_n \sin(\varphi_n))$, где случайный угол φ_n равномерно распределен в интервале $(0, 2\pi)$, и далее полагается $Q_{n+1} = P_{n+1}$. Случайные траектории строятся до тех пор, пока точка P_{n+1} не попадет в заданную малую окрестность ε границы области D . Пусть это будет точка P_v . Тогда приближенно полагаем $u(P_v) \approx g(P_v)$. Построив N таких траекторий, исходящих из точки P_0 , получим N значений $g(P_{v1}), g(P_{v2}), \dots, g(P_{vN})$, по которым находится приближенное значения решения уравнения (1) в точке P_0 :

$$u(P_0) \approx \frac{1}{N} \sum_{k=1}^N g(P_{vk})$$

6. Моделирование процесса прохождения нейтронов через пластину

Рассматривается задача об облучении нейтронами пластины конечных размеров, внутри которой происходит поглощение, рассеивание и отражение нейтронов. Предполагается, что перед пластиной все нейтроны имеют одинаковую скорость, которая остается постоянной (тепловая скорость), однако внутри пластины она может менять свое направление при взаимодействии с веществом пластины. Рассматриваются только два вида взаимодействия – рассеяние и поглощение, причем сечения этих взаимодействий Σ_s, Σ_c предполагаются известными, индикатриса рассеяния (условная плотность распределения направлений рассеяния) направлена «вперед». Кроме того, происходит отражение нейтронов при выходе их на отражающую верхнюю или нижнюю границу пластины. Геометрия области приведена на рисунке. Внутри пластины находится поглощающий элемент, при попадании в который нейтроны полностью поглощаются.

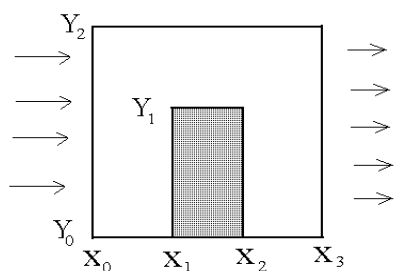


Рис.1. Схема области взаимодействия

Отражающими нейтроны являются верхняя граница области и часть нижней ($[x_0, x_1], [x_2, x_3]$). Целью моделирования является определение распределения вероятности прохождения нейтронов через такую пластину как функции от ординаты начальной точки их вылета.

Алгоритм расчета

Для моделирования рассматриваемого процесса применяется статистический подход, основанный на методе Монте-Карло. Входное сечение пластины разбивается на M равномерно распределенных по ее сечению точек с координатами $x_0 = X_0$, $y_j = Y_0 + (j-1) \cdot (Y_2 - Y_0) / (M-1)$, $j=1, \dots, M$. Из каждой такой точки выпускается N нейтронов, направления скоростей которых равномерно распределены в интервале углов $[-\pi/2, \pi/2]$ (индикатриса рассеяния «вперед»). Для каждого нейтрона вводится величина η_{jk} ($k=1, \dots, N$), описывающая «судьбу» каждого нейтрона, начальное значение которой полагается равной единице. Далее определяется случайная величина свободного пробега нейтрона $\xi_0 = -(1/\Sigma) \cdot \ln(\gamma)$, где $\Sigma = \Sigma_s + \Sigma_c$ - полное сечение взаимодействия и его новое положение на плоскости, γ - случайная величина, равномерно распределенная на интервале $[0,1]$. Затем определяются его новые координаты: $x_1 = x_0 + \xi_0 \cos(\theta_0)$ и $y_1 = y_0 + \xi_0 \sin(\theta_0)$. Если нейтрон попал в заштрихованную область, его траектория на этом обрывается и величина η_{jk} полагается равной нулю. Если он пересек отражающую границу, то его новые координаты определяются из условий зеркального отражения от этой границы. Если нейтрон вылетел за пределы области, то η_{jk} полагается равной нулю, если это левая граница. Если же нейтрон остался внутри области, разыгрывается его новая «судьба» - либо рассеяние, либо поглощение. Вероятность рассеяния нейтрона при взаимодействии его с материалом пластины в любой точке области есть Σ_s / Σ , вероятность поглощения есть Σ_c / Σ . Считается, что нейтрон рассеялся, если $\gamma < \Sigma_s / \Sigma$ и что он поглотился в противном случае. В случае поглощения величина η_{jk} полагается равной нулю и «судьба» нейтрона на этом обрывается. В остальных случаях разыгрывается его новое направление движения в предположении равномерности распределения нового угла направления вектора скорости внутри директрисы рассеяния по формуле $\theta_1 = \theta_0 + \pi/2(2\gamma - 1)$, новая длина свободного пробега $\xi_1 = -(1/\Sigma) \cdot \ln(\gamma)$ и

определяются новые координаты $x_2 = x_1 + \xi_1 \cos(\theta_1)$ и $y_2 = y_1 + \xi_1 \sin(\theta_1)$. Далее все повторяется до обрыва траектории данного k -го нейтрона. После расчета всех N нейтронов, вылетевших из одной j -той точки, определяется вероятность их прохождения через пластину $p_j \approx \sum_{k=1}^N \eta_{jk} / N$.

Заключение.

1. Метод Монте-Карло обладает высокой степенью параллелизма при его реализации как на MPI, так и на OpenMP, программы на его основе позволяют равномерно загружать вычислительные узлы кластерных систем.
2. Недостатками являются:
 - а) Границы погрешности не определены точно, но включают некую случайность. Это, однако, более психологическая, чем реальная, трудность.
 - б) Статическая погрешность убывает медленно.
 - в) Необходимость иметь случайные числа.