

Глава 12

Экспериментальная физика и моделирование

В настоящее время компьютерное моделирование эксперимента¹ стало рутинным и необходимым этапом его подготовки, особенно в тех случаях, когда полный цикл физического эксперимента (проектирование и создание установки, проведение измерений, обработка и анализ данных, получение и публикация физических результатов) занимает годы или даже десятки лет.

Как было видно из рассмотрения понятий фазового объема и диаграммы Далица, один из наиболее распространенных приемов моделирования реакции взаимодействия частиц состоит в том, чтобы для определенного кинематического состояния начальных частиц (пучка и мишени) случайным образом определить кинематические параметры тех частиц конечного состояния, которые интересуют экспериментатора. При этом должны быть выполнены все известные законы сохранения (энергии-импульса, квантовых чисел и т. п.). Затем прослеживается их "судьба" в экспериментальной установке.

Моделируемые события взаимодействия по своим кинематическим параметрам распределены в фазовом пространстве реакции (Часть 9) по некоторому закону и при моделировании необходимо

¹ Одним из пионеров и энтузиастов этого метода был Г.И.Копылов [2].

обеспечить достаточно плотное заполнение разрешенной кинематической области фазового пространства (фазовый объем). В простейшем случае (например, когда отвлекаются от динамики реакции), моделируемые события должны быть равномерно распределены по диаграмме Далица или, для многочастичных реакций, по многомерному фазовому объему. Во многих случаях такой подход оправдан. Однако, нередко требуется учесть наличие "пустот", или "каверн" в фазовом пространстве [2], причины появления которых очевидны, а некоторые примеры уже были рассмотрены. Как учесть наличие этих "каверн" наиболее эффективным, в смысле скорости вычислений, образом? Иными словами, как обеспечить розыгрыш событий моделируемой реакции наиболее экономным образом, только в кинематически (в широком смысле) разрешенной области фазового пространства? На этот вопрос не всегда просто ответить, и поэтому представляется полезным познакомить начинающих исследователей с некоторыми практическими приемами, помогающими (или облегчающими) нахождение ответа на него.

Опыт показывает, что приходя в современные коллективы экспериментаторов, работающих в физике элементарных частиц и ядер, физики-новички нередко "изобретают велосипед", когда включаются в моделирование эксперимента. Происходит это потому, что соответствующей учебной литературы немного или она труднодоступна (как, например, хорошее введение в "азбуку" моделирования [71]). Это обстоятельство и послужило мотивом для включения краткого обсуждения способов генерации случайных событий по заранее заданному закону распределения в курс основ кинематики элементарных процессов. Обсуждение опирается на работу [71].

12.1 Псевдослучайные числа.

В основе подавляющего большинства генераторов чисел, распределенных по некоторому, заранее заданному закону, лежит генератор равномерно распределенных на отрезке $(0, 1)$ случайных чисел.

Главные особенности большинства таких генераторов состоят в следующем:

1. Генераторы используют хорошо определенные алгоритмы, по которым создается последовательность чисел, где каждое сле-

дующее число получается из предыдущего. Начальное число хорошо определено.

Это означает, что при нескольких последовательных запусках задачи генерации, пользователь проходит через ту же самую последовательность случайных чисел, если каждый раз начинает генерацию чисел с самого начала.

2. Стартовое число может зависеть от типа компьютера, так что разные компьютеры будут генерировать различные последовательности случайных чисел. В конкретных реализациях граничные значения 0 и 1 могут либо включаться, либо не включаться во множество генерируемых чисел.
3. Название и конкретный алгоритм конкретного генератора случайных чисел, равномерно распределенных на отрезке $(0, 1)$, зависят от применяемой программной и аппаратной платформ.

Ясно, что создаваемые таким образом случайные числа "не вполне случайны", хотя бы из-за конечной длины машинного слова. Поэтому они называются **псевдослучайными**. Отсюда следует также, что генерируемые числа повторяются с некоторым квазипериодом. Если длина генерируемой последовательности меньше этого квазипериода, то распределение генерируемых чисел будет достаточно близким к равномерному, а сами числа будут независимы в статистическом смысле. **Важно знать, каков период генератора на вашем компьютере.** На практике, пользователь всегда должен работать с последовательностями, длина которых намного меньше периода генератора. Только тогда можно быть уверенным в том, что числа действительно случайны в статистическом смысле.

12.2 Генерация событий по заданному закону.

Подробно, и с большим количеством часто встречающихся на практике примеров и алгоритмов, эти вопросы изложены в работе[71]. Простейшие примеры, рассмотренные здесь, взяты именно оттуда.

Одной из типичных и часто встречающихся задач в процессе моделирования эксперимента методом Монте-Карло является проблема генерации случайной величины x , распределенной на интервале

$[A, B]$ по заданному (например, экспериментально измеренному) закону $f(x)$ (здесь $f(x)$ – плотность вероятности).

Очевидно, что удобнее всего иметь такой генератор, который выдает нужное случайное число чаще всего там, где плотность вероятности максимальна. Наиболее эффективная идея решения поставленной задачи основана именно на этой здравой мысли и состоит в следующем.

Разделим интервал $[A, B]$ на N подынтервалов; обозначим конец подынтервала i как x_i ; $x_0 = A$; $x_N = B$. Вероятность того, что x принадлежит подынтервалу i есть $\Delta W_i \simeq f(x_i) \cdot (x_i - x_{i-1})$, точнее:

$$\Delta W_i = \int_{x_{i-1}}^{x_i} f(x) dx \quad ; \quad \sum_{i=1}^N \Delta W_i = 1 . \quad (12.1)$$

Расположим (последовательно) все числа ΔW_i на интервале $[0, 1]$ (поскольку вероятность есть неотрицательное число, не превосходящее 1 по определению). Очевидно, что если *равномерно* распределенное случайное число w , генерируемое соответствующей стандартной программой (условно назовем ее RANF) попадает в интервал j (т.е. удовлетворяет условию $\Delta W_{j-1} < w \leq \Delta W_j$), то соответствующее значение случайной величины x есть x_j (или $(x_j + x_{j-1})/2$, что кажется для многих предпочтительнее). Это и есть суть идеи, которую легко запрограммировать.

Другой, нередко используемый способ, состоит в том, чтобы приписать событию с выпадением числа x , генерированного программой RANF², вес $w(x)$, но это – как правило – неудобно, тем более, что основная часть машинного времени уйдет на генерацию чисел вдали от максимума (максимумов) распределения. Кроме того, возникает масса других неудобств, если используются критерии отбора событий по тем или иным признакам и необходимо сравнивать выходы событий после таких отборов.

Итак, основная идея – генерировать равномерно распределенное случайное число y с последующим нахождением требуемого числа x по правилу

$$y = \int_A^x f(x') dx' \quad ; \quad \int_A^B f(x') dx' = 1 . \quad (12.2)$$

² Примем для определенности, что генератор случайных чисел, равномерно распределенных на отрезке $(0, 1)$, называется именно так.

Можно поступить и по-другому: найти такую новую переменную $\xi = g(x)$, что $g'(x) = f(x)$. В этом случае $d\xi = g'(x)dx = f(x)dx$, где величина ξ распределена равномерно на интервале $[0, 1]$. Получив ξ с помощью генератора RANF, необходимо решить уравнение $g(x) = \xi$ для того, чтобы получить значение переменной x , распределенной согласно заданной плотности вероятности. Этот метод носит название **метода обратного преобразования**.

Пример 1: экспоненциальное распределение. Проиллюстрируем только что сказанное. Пусть распределение плотности вероятности имеет вид $\exp(-x/X_0)$, где X_0 есть некая константа. Надо построить генератор, дающий вам случайное число, распределенное по этому закону и подсчитать, сколько чисел из N_0 "выброшенных" генератором окажется больше заранее заданного X_{max} .

Используем метод обратного преобразования. Имеем:

$$d\xi = -\frac{1}{X_0}e^{-x/X_0}dx; \quad \xi = -\frac{1}{X_0} \int_0^x e^{-x'/X_0} dx' = 1 - e^{-x/X_0}$$

$$x = -X_0 \cdot \ln(1 - \xi) . \quad (12.3)$$

Теперь алгоритм выглядит так:

- ```
-- Инициализация; N0 - полное число
генерируемых чисел, X0 - параметр распределения,
Xmax - заданная граница, Nout - искомое количество
"выброшенных" генератором чисел, превзошедших Xmax.
Nout=0
-- Начало цикла по i от 1 до N0
ksi=RANF(1) ; ksi - равномерно распределенное случайное число
x=-X0*Alog(1.-ksi)
Если (x.ge.Xmax) Nout=1+Nout
-- Конец цикла по i от 1 до N0
Вывод Nout
-- КОНЕЦ ПРОГРАММЫ
```

**Пример 2: нормальное распределение.** Иногда бывает полезным, для решения задачи генерации случайных чисел в "одномерном" множестве, рассмотреть задачу в *большем* числе измерений, чем в изначальной ее формулировке<sup>3</sup>. Проблема генерации

---

<sup>3</sup> Следует отметить, что это – один из мощных приемов решения самых разнообразных проблем, в том числе – изобретательских.

**нормально распределенного** случайного числа является превосходной иллюстрацией действенности этого приема.

Итак, пусть надо получить случайное число, распределенное по нормальному закону со средним значением  $X = 0$  и параметром дисперсии  $\sigma = 1$  (эти два дополнительных условия никак не ограничивают общности рассмотрения), т. е. случайное число  $x$  распределено согласно плотности вероятности

$$\frac{1}{\sqrt{2\pi}} \exp(-x^2/2). \quad (12.4)$$

Выйдем в пространство чисел более высокой размерности ("из линии на плоскость") и рассмотрим задачу на плоскости  $(x, y)$ , где  $y$  — тоже случайное число, распределенное по тому же закону, что  $x$ , но **статистически независимое** от него<sup>4</sup>. Тогда вероятность иметь **пару** чисел  $x$  и  $y$  в интервале  $(x, x + dx; y, y + dy)$  есть

$$\begin{aligned} dW_{xy} &= \frac{1}{2\pi} \exp(-(x^2 + y^2)/2) dx dy = \\ &= \frac{1}{2\pi} \exp(-\rho^2/2) \rho d\rho d\varphi = dW_{\rho\varphi}. \end{aligned} \quad (12.5)$$

Можно проинтегрировать по переменной  $\varphi$  и получить следующее распределение для случайной величины  $\rho$ :

$$dW_{\rho} = \exp(-\rho^2/2) \rho d\rho \Rightarrow W_{\rho} = \exp(-\rho^2/2); \quad (12.6)$$

Как видно, использован именно метод "обратного преобразования":  $W_{\rho}$  лежит в единичном интервале  $(0, 1)$ . Разыгрывая величину  $W_{\rho}$  по равномерному закону в этом интервале, находим случайные величины  $x$  и  $y$ :

$$\begin{aligned} \rho &= \sqrt{-2 \ln(W_{\rho})} \\ x &= \rho \cdot \cos(\varphi) \\ y &= \rho \cdot \sin(\varphi) \end{aligned} \quad (12.7)$$

где  $\varphi$  также равномерно распределенная в интервале  $(0, 2\pi)$  величина. В итоге, получаем **два** нормально распределенных случайных числа  $x$  и  $y$ , вместо одного, как это требовалось вначале!

---

<sup>4</sup> Этот алгоритм известен столь давно и широко, что трудно даже найти ссылки на его авторов.