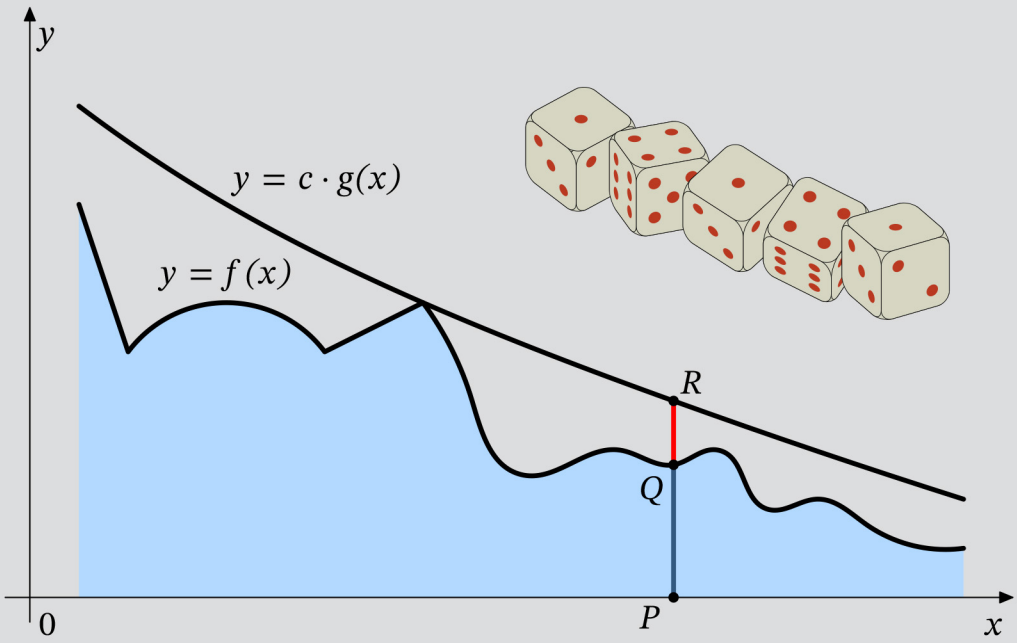
PODRĘCZNIKI

POLITECHNIKA LUBELSKA

Symulacje Monte Carlo

Teoria i zastosowania

Paweł Wlaź



Symulacje Monte Carlo

Teoria i zastosowania

Symulacje Monte Carlo

Teoria i zastosowania

Paweł Wlaź

WYDAWNICTWO
POLITECHNIKI
LUBELSKIEJ

Lublin 2025

RECENZENCI:

dr hab. **Michał Bereta**, prof. UMCS, Uniwersytet Marii Curie-Skłodowskiej w Lublinie

dr hab. inż. **Mariusz Bieniek**, prof. PK, Politechnika Krakowska

AUTOR:

dr **Paweł Wlaź**, ORCID: 0000-0003-0565-4946
Politechnika Lubelska – ROR: https://ror.org/024zjzd49

Redaktor prowadzący: **Magdalena Chołojczyk**
Korekta językowa: **Magdalena Chołojczyk**
Skład i łamanie: **Paweł Wlaź**

Publikacja została sfinansowana w ramach programu projakościowego prorektora ds. studenckich:
Konkurs na wydanie podręcznika akademickiego lub skryptu.

Ilustracja na okładce została wykonana przez autora.
Jeśli nie wskazano źródła ilustracji, stanowią one opracowanie własne autora.

Publikacja wydana za zgodą **Rektora Politechniki Lubelskiej**

ISBN: 978-83-7947-639-8 (wersja drukowana)
ISBN: 978-83-7947-640-4 (wersja elektroniczna)
DOI: 10.35784/9788379476404
Wydawca: Wydawnictwo Politechniki Lubelskiej
www.wpl.pollub.pl
ul. Nadbystrzycka 36C, 20-618 Lublin
tel. (81) 538-46-59

WYDAWNICTWO
POLITECHNIKI
LUBELSKIEJ

Nakład: 50 egz.

Spis treści

Symulacje Monte Carlo. Teoria i zastosowania

Skrypt został opracowany na podstawie notatek do wykładów i laboratoriów z przedmiotu „metoda Monte Carlo” prowadzonego na studiach pierwszego stopnia na kierunku matematyka w Politechnice Lubelskiej.

W skrypcie omawiane są sposoby generowania pseudolosowych próbek z różnych rozkładów, metody mierzenia dokładności oraz redukcji wariancji, obliczenia quasi-Monte Carlo, obliczenia związane z procesami stochastycznymi i algorytm Metropolisa–Hastingsa.

Słowa kluczowe: metoda Monte Carlo, statystyka, liczby losowe, liczby pseudolosowe, metody obliczeniowe.

Monte Carlo simulations. Theory and applications

This academic script is based on notes from lectures and laboratories for the subject Monte Carlo Method, which is taught at the first-cycle studies in the field of mathematics at the Lublin University of Technology.

The script discusses methods of generating pseudorandom samples from various distributions, methods of measuring accuracy and reducing variance, calculations using the quasi-Monte Carlo method, calculations related to stochastic processes and Metropolis–Hastings algorithm.

Keywords: Monte Carlo method, statistics, random numbers, pseudorandom numbers, computational methods.

Wykaz oznaczeń

$\mathbb{R}$	–	zbiór liczb rzeczywistych
$\mathbb{N}$	–	zbiór liczb naturalnych
$(\Omega, \mathcal{F}, \Pr)$	–	przestrzeń probabilistyczna ze zbiorem zdarzeń elementarnych Ω, σ-ciałem zbiorów mierzalnych $\mathcal{F}$ i funkcją prawdopodobieństwa $\Pr$
EX	–	wartość oczekiwana zmiennej losowej X
$\operatorname{Var} X$	–	wariancja zmiennej losowej X
$\operatorname{Cov}(X, Y)$	–	kowariancja między zmiennymi losowymi X oraz Y
$\varrho(X, Y)$	–	współczynnik korelacji między zmiennymi losowymi X oraz Y
μ	–	wartość oczekiwana zmiennej losowej
σ	–	odchylenie standardowe zmiennej losowej
$N(\mu, \sigma)$	–	rozkład normalny z wartością oczekiwaną μ i odchyleniem standardowym σ
$\varphi\colon \mathbb{R} \to \mathbb{R}$	–	funkcja gęstości rozkładu $N(0, 1)$
$\Phi\colon \mathbb{R} \to \mathbb{R}$	–	dystrybuanta rozkładu $N(0, 1)$
$\operatorname{erf}\colon \mathbb{R} \to \mathbb{R}$	–	funkcja błędu
$\overline{x}$	–	średnia arytmetyczna z próby $x_1, x_2, \ldots, x_n$
$\hat{s}^2$	–	nieobciążony estymator wariancji z próby $x_1, x_2, \ldots, x_n$
v. r. r.	–	współczynnik redukcji wariancji
$\{W(t)\}_{t\in[0,\infty)}$	–	proces Wienera

Wstęp

Metody Monte Carlo mają nie tylko atrakcyjną nazwę, ale także bardzo ciekawą historię i ważne współczesne zastosowania. Koniecznie należy wspomnieć o znakomitych postaciach związanych z tymi metodami – jak Ulam, von Neumann, Metropolis – które w latach 40. i 50. XX wieku miały wielki wkład w rozwój technologii komputerowej oraz tych działów techniki i matematyki, które do rozwoju tej technologii się przyczyniały, a również z niego wiele czerpały.

Nawet kilkadziesiąt lat po powstaniu pierwszych komputerów wciąż do podręczników opisujących metody symulacyjne dodawano tablice liczb losowych, które umożliwiały czytelnikowi – pozbawionemu na ogół swobodnego dostępu do cyfrowego środowiska obliczeniowego – przeprowadzanie na papierze eksperymentalnych symulacji. Mogły być one jednak jedynie ćwiczeniem pomagającym w rozumieniu teorii – dla osiągnięcia jakiejkolwiek przydatnej dokładności metody te wymagają bardzo wielkiej liczby prób, do których komputer jest konieczny. Zatem metody te bez użycia komputera i możliwości zaprogramowania eksperymentu nie mają praktycznego sensu.

Jest to więc dziedzina, w której najlepiej będzie się czuł ktoś, kto rozumie matematykę, rachunek prawdopodobieństwa i statystykę, ale jednocześnie potrafi zaprogramować obliczenia lub przynajmniej posłużyć się pakietem oprogramowania przeznaczonym do wspomagania obliczeń Monte Carlo. Kluczowa jest jednak biegłość z dziedziny, której eksperyment ma dotyczyć, zbudowanie modelu badanej sytuacji, zidentyfikowanie zmiennych losowych, ich rozkładów, parametrów, opracowanie możliwości pseudolosowej realizacji tych zmiennych, obliczania i uśredniania lub całkowania interesującej wielkości.

Jeżeli chcemy, by taki eksperyment poprawnie opisywał rzeczywistość, by cokolwiek pewnego (czy też wysoce prawdopodobnego) można było powiedzieć o dokładności wyników, to niezbędna jest także wiedza i umiejętności dotyczące probabilistycznych i statystycznych własności metod symulacyjnych.

Zdobyciu tych właśnie umiejętności i wiedzy ma służyć przedmiot metoda Monte Carlo. Jest to przedmiot na studiach pierwszego stopnia na kierunku matematyka (studia o profilu praktycznym) prowadzonym w Politechnice Lubelskiej. Na przedmiot ten przeznaczone jest 20 godzin wykładu i 20 godzin laboratorium. Program studiów na tym kierunku nie przewiduje niektórych ważnych dla przedmiotu dziedzin, na przykład procesów stochastycznych (występują na studiach drugiego stopnia). Elementy tej teorii wystarczające do zaprogramowania właściwego eksperymentu są w takiej sytuacji, bez nadmiernej szczegółowości, podane podczas zajęć. Nie inaczej jest w niniejszym skrypcie.

W skrypcie omawiane są podstawowe intuicje związane z obliczeniami metodą Monte Carlo, sposoby generowania pseudolosowych próbek z różnych rozkładów, metody mierzenia dokładności i powiększania dokładności za pomocą redukcji wariancji. Omówione są także obliczenia metodą quasi-Monte Carlo, obliczanie trudnych rozkładów statystycznych metodą Monte Carlo, badanie własności procesów stochastycznych, omówione zostało także próbkowanie Monte Carlo łańcuchami Markowa na przykładzie algorytmu Metropolisa–Hastingsa.

Mam nadzieję, że skrypt okaże się użyteczny także dla studentów innych kierunków – założona jest jedynie znajomość rachunku prawdopodobieństwa w zakresie najważniejszych rozkładów oraz statystyki matematycznej, zwłaszcza estymacji punktowej i przedziałowej (ale podstawowe dla dużych prób modele estymacji przedziałowej są tutaj przywołane). Nie da się też skorzystać w pełni z tego skryptu bez znajomości `C++`, bo w tym języku programowania prezentowane są listingi i tego języka dotyczą zadania, w których występuje konieczność programowania. Większość tych problemów może być jednak tak naprawdę rozwiązana w wielu innych środowiskach programistycznych. Takie zadania do samodzielnego rozwiązania znajdują się na końcu każdego z pięciu rozdziałów. Po treściach zadań umieszczono wskazówki lub odpowiedzi do wybranych zagadnień.

ROZDZIAŁ 1

Podstawowy schemat metody Monte Carlo

1.1. Intuicja

Intuicja stojąca za ogólną zasadą Monte Carlo wynika z analogii, jakie widać między miarą prawdopodobieństwa a „zwykłą" miarą, taką jak długość, powierzchnia, objętość[1]. Szczególnie dobrze widoczne jest to, gdy mamy do czynienia z tzw. prawdopodobieństwem geometrycznym, a więc w sytuacji, w której zdarzenia elementarne to punkty z ustalonego mierzalnego podzbioru $\Omega \subset \mathbb{R}^n$, natomiast prawdopodobieństwa zdarzeń – czyli podzbiorów Ω – są proporcjonalne do powierzchni (lub długości, objętości) tych podzbiorów.

To, co w probabilistyce posłużyło do określenia prawdopodobieństwa, w przypadku metody Monte Carlo działa w drugą stronę – można szacować miarę Lebesgue'a zbioru na podstawie oceny prawdopodobieństwa przynależności do danego zbioru. Ta ocena z kolei może zostać dokonana na podstawie symulacji, losowego, a częściej pseudolosowego, eksperymentu, który wybierając wielokrotnie punkt ze zbioru Ω obliczy frakcję punktów znajdujących się w badanym podzbiorze.

Prostą ilustracją tej zasady jest następujący przykład.

Przykład 1.1. *Oszacuj wartość liczby π na podstawie eksperymentu polegającego na $10\,000\,000$-krotnym jednostajnym losowaniu punktu z kwadratu i obliczeniu frakcji punktów znajdujących się w kole wpisanym w ten kwadrat.*

[1] By skrócić i uściślić dalsze zapisy będę używał pojęcia miary Lebesgue'a, wtedy gdy w n-wymiarowym przypadku zechcę użyć uogólnienia pojęcia powierzchni (długości, objętości). Do zrozumienia nie jest konieczne znajomość dokładnej definicji tej miary, wystarczy intuicyjna wiedza na temat pojęć, które ta miara uogólnia.

Oczywiście ta frakcja nie zależy od tego, jaka jest długość boku kwadratu. Tak czy inaczej wiemy, że stosunek pola koła wpisanego w kwadrat do pola tego kwadratu wynosi $\frac{\pi}{4}$, stąd nasze oszacowanie wyniesie

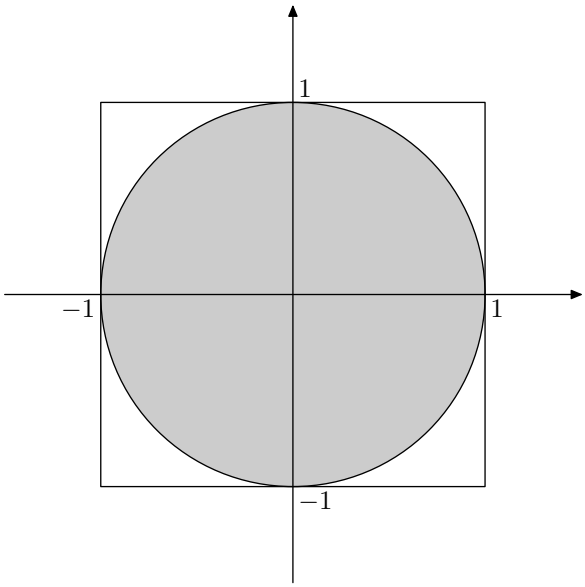
$$\pi \approx 4\frac{k}{n},$$

gdzie n jest liczbą wylosowanych punktów, natomiast k jest liczbą punktów, które znajdą się w kole.

W tego typu zadaniu ważne jest, by procedura prowadząca do oszacowania pewnej wielkości nie korzystała w żaden sposób z tej wielkości. Dokładnie tak jest w tym przypadku. Ważne jest też, by losowanie było `jednostajne`. To oznacza, że prawdopodobieństwo realizacji w danym zborze jest proporcjonalne do miary Lebesgue'a tego zbioru. By rozwiązanie zaimplementować, przyjmijmy, że rozważany kwadrat ma wierzchołki o współrzędnych

$$(-1,-1),\ (-1,1),\ (1,1),\ (1,-1),$$

jak na rysunku 1.1.



Rys. 1.1. Położenie figur z przykładu 1.1

Zestaw funkcji realizujących losowanie punktu z kwadratu (dokładnie to z wnętrza kwadratu), sprawdzających, czy punkt ten leży wewnątrz koła i na tej podstawie dokonujących oszacowania liczby π przedstawiony jest na listingu 1.1. Zauważmy, że gdy mowa o mierze bądź prawdopodobieństwie, to bez znaczenia będzie, czy figury w tym zadaniu traktujemy jako domknięte czy jako otwarte, ale w praktycznym losowaniu na coś trzeba się zdecydować.

```
#include <iostream>
#include <random>
double runif_o() {
// wylosuj liczbę z przedziału otwartego (-1,1)
    static std::mt19937_64 gen{}; // static - inicjowana tyko raz
    static std::uniform_real_distribution<double> u(-1.0, 1.0);
    double x;
    do {
            x = u(gen);
    } while (x == -1.0);
    return x;
}
int ile_sukcesów(bool Y(), int n) {
// uruchom n-krotnie funkcje Y i zwróć liczbę wartości TRUE
    int ile=0;
    for (int i=0; i<n; i++)
        ile += Y();
    return ile;
}
bool punkt_w_kole() {
// wylosuj punkt i zwróć TRUE jeśli punkt leży wewnątrz kole
    double x = runif_o(), y = runif_o();
    return x*x+y*y < 1.0;
}
int main() {
    int n = 10'000'000;
    int k = ile_sukcesów( punkt_w_kole, n );
    std::cout << "liczba prób " << n << ", liczba trafień: " <<
        k << ",\noszacowanie liczby π: " << k/double(n)*4.0 <<
        ".\n";
}
```

Listing 1.1. Rozwiązanie dla przykładu 1.1

Jeszcze krótki komentarz do kodu. Losowanie jednostajne w takim obszarze prostokątnym łatwo uzyskujemy, losując z rozkładu jednostajnego w odpowiednich przedziałach dla każdej współrzędnej. Najczęściej generowanie liczb pseudolosowych w językach programowania, w przypadku losowania jednostajnego z przedziału rzeczywistego, sprowadza się do losowania z przedziału lewostronnie domkniętego, prawostronnie otwartego. Jest to z wielu powodów wygodne, o czym jest mowa w dalszej części skryptu. Akurat w tym przykładzie zręczniejsze jest (ze względu na matematyczną treść i symetrię rozważanych figur) losowanie jednostajne z przedziału obustronnie domkniętego lub obustronnie otwartego. Przyjąłem tę drugą opcję i kod prezentuje jeden z możliwych sposobów rozwiązania tego dylematu.

Wracając do części obliczeniowej: tekst, jaki wyświetli program z listingu 1.1 na poprzedniej stronie, to

```
liczba prób 10000000, liczba trafień: 7854766,
oszacowanie liczby π: 3.14191.
```

Jak widać, wynik nie jest zbyt imponujący. Różni się o około 0.0009 od π i to mimo znacznej liczby prób. Co gorsza, otrzymaliśmy tylko oszacowanie punktowe i na tym etapie nie mamy żadnych gwarancji, że na przykład używając innego generatora lub tego samego generatora, ale z innym ziarnem[2], wynik nie będzie się znacznie różnił od otrzymanego wcześniej. Ocenę dokładności uzyskamy na przykład stosując teorię przedziałów ufności.

Na zakończenie rozważań związanych z tym przykładem dodam, że gdyby liczbę $10\,000\,000$ zamienić na 100 w treści przykładu (oraz w programie), to uzyskalibyśmy 76 trafień w koło i oszacowanie liczby π wynoszące około 3.04.

Rozważania dotyczące domkniętości/otwartości przedziału także nie mają w tym przypadku istotnego znaczenia. Gdybyśmy w listingu 1.1 na poprzedniej stronie linijki 7–12 zastąpili uproszczoną wersją ignorującą ten problem, czyli

[2] Ziarno to stan początkowy generatora. O metodach pozyskiwania liczb pseudolosowych można przeczytać w rozdziale 2 oraz (zwłaszcza w kontekście języka C++) w podrozdziale 3.4.

```
    return u(gen);
```

to omawiane wyniki nie uległyby żadnej zmianie. Wartość równa lewemu końcowi przedziału dla rozkładu jednostajnego generowana jest tak rzadko (dla typu `double`), że jej wpływ na wyniki takiego eksperymentu można zaniedbać.

..... *koniec przykładu 1.1* .....

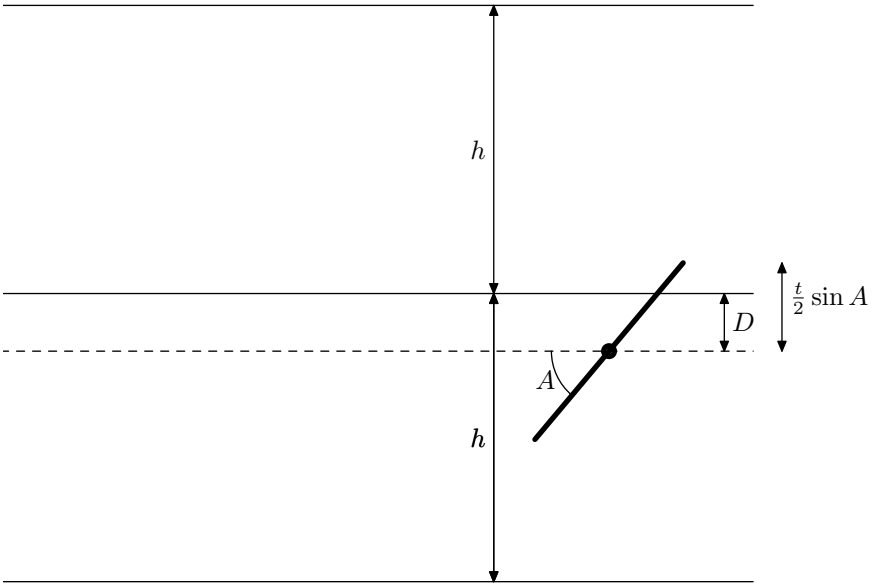
Rzecz jasna liczba π badana w powyższym przykładzie jest nam znana – przynajmniej jeżeli chodzi o jej przybliżenia z dokładnością do bardzo wielu cyfr znaczących. Chodziło o jasny, czytelny przykład odwołujący się do pojęć powszechnie znanych. Gdyby sobie jednak postawić pytanie o realizację tego eksperymentu bez techniki komputerowej, to natychmiast dotrzemy do zasadniczych problemów. I wcale nie chodzi o to, że trudno „ręcznie" przeprowadzać wielokrotne losowanie. Ważniejsze jest nawet pytanie o to, jak choćby jednokrotnie przeprowadzić prawdziwie jednostajne losowanie z kwadratu. Rzucanie z bliska niewątpliwie spowoduje, że niektóre obszary będą bardziej intensywnie wybierane. Z kolei rzucanie z daleka spowoduje, że szansa trafienia w kwadrat zbliży się do zera.

Nie sposób przy okazji nie wspomnieć słynnego zadania Buffona, które zostanie przedstawione w kolejnym przykładzie. W zadaniu tym, a właściwie w jego kontynuacji, eksperyment przeprowadzony „naturalną" metodą jest jak najbardziej możliwy.

Przykład 1.2 (Igła Buffona)**.** *Słynny problem dotyczący prawdopodobieństwa geometrycznego, autorem opublikowanego w 1777 roku rozwiązania jest Georges-Louis Leclerc, hrabia de Buffon [2].*

Igła długości t rzucana jest (w przypadkowy sposób i z wysoka) na płaszczyznę z narysowanymi równoległymi liniami, odległość między sąsiednimi liniami to h, $h > t$. Wyznacz prawdopodobieństwo, że igła przetnie jedną z linii.

Sytuacja została zilustrowana na rysunku 1.2 na następnej stronie. Jeśli założy się, że kąt ostry A między igłą a liniami jest zmienną losową o rozkładzie jednostajnym w przedziale $[0, \frac{\pi}{2}]$, a odległość D między środkiem igły a najbliższą linią jest zmienną losową o rozkładzie jednostajnym w przedziale $[0, \frac{h}{2}]$ – założenia usprawiedliwione treścią



Rys. 1.2. Igła Buffona

zadania – to można zauważyć, że warunkiem przecięcia przez igłę linii jest, by

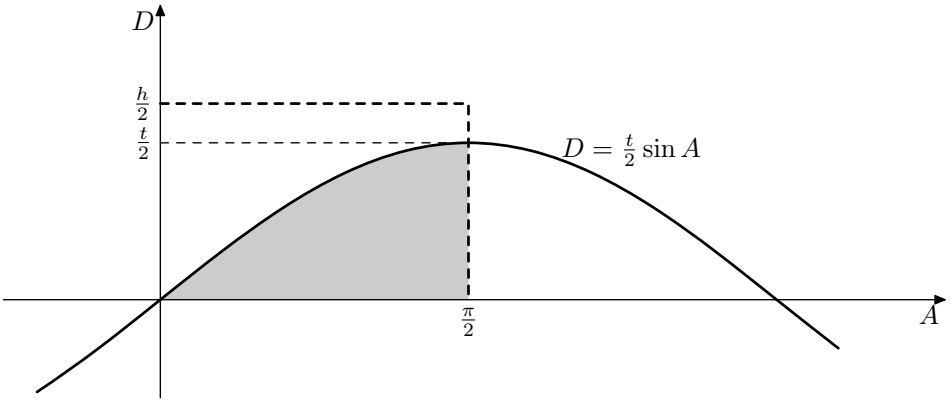
$$\frac{t}{2}\sin A > D. \tag{1.1}$$

Szkic zbioru możliwych wartości dwuwymiarowej zmiennej losowej (A, D), wraz z zaznaczonym zbiorem punktów spełniających warunek (1.1) można znaleźć na rysunku 1.3 na następnej stronie.

Obie zmienne losowe mają rozkład jednostajny. Zmienne te są niezależne, stąd rozkład w prostokącie jest także jednostajny, a prawdopodobieństwo jest proporcjonalne do powierzchni.

Ostatecznie więc mamy

$$\Pr(\text{„igła przetnie linię”}) = \frac{\int_0^{\pi/2} h/2 \cdot \sin x \, dx}{\frac{\pi}{2} \cdot \frac{h}{2}} = \frac{2t}{\pi h}.$$



Rys. 1.3. Problem Igły Buffona sprowadzony do prawdopodobieństwa geometrycznego

Na tym kończy się zadanie probabilistyczne, dla nas ciekawsza jest jego kontynuacja. W 1812 roku Laplace zauważył [19], że na podstawie liczby przecięć w wielu próbach można oszacować wielkość π, zastępując prawdopodobieństwo częstością i przekształcając powyższy wzór do postaci

$$\pi \approx \frac{2t}{h}\frac{n}{k}$$

gdzie n jest liczbą prób, k jest liczbą uzyskanych przecięć.

Znaleźli się nawet chętni do przeprowadzenia takiego eksperymentu: w 1901 roku Lazzarini [20] opublikował pracę, w której raportował, że w 3408 rzutach igłą o długości 2.5 cm, przy liniach oddalonych o 3 cm, uzyskał 1808 trafień w linię, co dało oszacowanie liczby π wynoszące $3.141\,593$. To znakomite przybliżenie. Jest to doskonałe zaokrąglenie do siedmiu cyfr znaczących liczby π. Warto też dodać, że liczba trafień 1808 była najlepszą możliwą, każda inna dawała gorsze przybliżenie π. Rzetelne (przeprowadzone z wykorzystaniem komputera) eksperymenty pokazują, że uzyskiwane oszacowania mają przy 3408 próbach przedziały ufności długości około 0.2. Wrócimy do tego w zadaniu 4.2 na stronie 94. Można także na drodze teoretycznej wykazać, jak nieprawdopodobnie wielkie szczęście musiałoby prowadzić do trafienia właśnie 1808 razy. Praca Lazzariniego została oczywiście w późniejszych latach wielokrotnie krytycznie oceniona w aspekcie wiarygodności [2].

koniec przykładu 1.2

1.2. Trochę teorii

Fakt, że uśrednianie takie, jak w przykładzie 1.1 na stronie 13, zbliża nas do prawidłowego teoretycznie wyniku, może być tłumaczony na wiele sposobów.

1.2.1. Rozważania statystyczne

Liczba trafień może być traktowana jako liczba sukcesów w schemacie Bernoulliego. Ze statystyki matematycznej [4] wiadomo, że estymatorem prawdopodobieństwa sukcesu jest właśnie $\frac{K_n}{n}$, gdzie K_n to zmienna losowa równa liczbie sukcesów w n próbach. Jest to estymator zgodny, nieobciążony, efektywny.

1.2.2. Prawa wielkich liczb

Przypomnijmy kilka praw wielkich liczb [16].

Twierdzenie 1.1 (Prawo wielkich liczb Bernoulliego)**.** *Jeżeli K_n jest liczbą sukcesów w n próbach, to dla dowolnego $\varepsilon > 0$*

$$\lim_{n\to\infty} \Pr\left(\left|\frac{K_n}{n} - p\right| > \varepsilon\right) = 0,$$

gdzie p jest prawdopodobieństwem sukcesu w pojedynczej próbie.

Twierdzenie to oznacza właśnie zgodność estymatora $\frac{K_n}{n}$.

Twierdzenie 1.2 (Mocne prawo wielkich liczb)**.** *Jeżeli $X_1, X_2, \ldots$ jest ciągiem niezależnych zmiennych losowych o tym samym rozkładzie, to na to, by*

$$\Pr\left(\lim_{n\to\infty} \frac{1}{n}\sum_{k=1}^{n} X_k = a\right) = 1$$

potrzeba i wystarcza, by $EX_i = a$.

W przypadku igły Buffona, przyjmując $X_i = 1$, gdy w i-tej próbie igła przetnie linię, i 0 w przeciwnym wypadku, mocne prawo wielkich

liczb przyjmie postać

$$\Pr\left(\lim_{n\to\infty}\frac{K_n}{n}=a\right)=1\iff a=\frac{2t}{\pi h}.$$

1.2.3. Twierdzenie ergodyczne

Podejście statystyczne oraz prawa wielkich liczb jednak zakładają jakąś przestrzeń probabilistyczną. Tymczasem używając (na przykład tak jak w przedstawionych wcześniej listingach) liczb pseudolosowych, korzystamy przy obliczeniach w istocie rzeczy z deterministycznego algorytmu – nie ma tam zmiennej losowej. Jak taka sytuacja ma się do tej opisywanej za pomocą przestrzeni probabilistycznej? Jak uzasadnić poprawność uzyskiwanej przybliżonej równości?

Jednym z mocno przemawiających argumentów jest twierdzenie ergodyczne Birkhoffa [7].

Najpierw zostanie przypomniane pojęcie ergodyczności.

Definicja 1.1. Niech (X, F, μ) będzie przestrzenią probabilistyczną. Przekształcenie $T\colon X\to X$ zachowujące miarę jest `ergodyczne`, jeżeli dowolny zbiór niezmienniczy tego przekształcenia jest miary zero lub pełnej miary.

Twierdzenie 1.3 (Ergodyczne Birkhoffa)**.** *Jeżeli przekształcenie*

$$T\colon X\to X$$

zachowuje miarę i jest ergodyczne, a funkcja $f\colon X\to\mathbb{R}$ *jest całkowalna, to*

$$\lim_{n\to\infty}\frac{1}{n}\sum_{k=0}^{n-1}f(T^k x)=\int_X f(y)\,d\mu(y)$$

dla wszystkich x *poza zbiorem miary* 0.

Można to interpretować następująco. Powiedzmy, że x jest stanem początkowym (`ziarnem`) generatora liczb pseudolosowych, $T(\cdot)$ jest przekształceniem, algorytmem, który na podstawie stanu generatora dokonuje przejścia do kolejnego stanu. Wtedy uśredniona dla dużego n wartość funkcji $f(T^k x)$ zbliża się do jej całki.

Opisana wyżej koncepcja jest podstawą tzw. prostej metody Monte Carlo (ang. *crude Monte Carlo, CMC*). Aby obliczyć przybliżenie wartości oczekiwanej

$$Ef(X)$$

dla zmiennej losowej X, należy wygenerować za pomocą generatora liczb pseudolosowych n liczb „pasujących" do realizacji ciągu niezależnych zmiennych losowych o rozkładzie identycznym jak rozkład X i uśrednić wartości funkcji f na tym ciągu liczb. Oczywiście dla dużej wartości n.

Pozostaną kwestie takie jak:

- problemy generowania liczb pseudolosowych o odpowiednim rozkładzie,
- szacowanie błędów w takich obliczeniach,
- modyfikacja CMC prowadząca do szybszej zbieżności,
- szczególne sposoby postępowania w wybranych dziedzinach zastosowań.

Problemom tym poświęcone zostaną kolejne rozdziały skryptu.

Zadania

Zadanie 1.1. Zapisz wartość $\int_0^2 \exp(-x^2)\,dx$ jako wartość oczekiwaną z $f(X)$ dla odpowiednio dobranej formuły $f(\cdot)$, gdy

(a) X jest zmienną losową o rozkładzie jednostajnym w $[0, 2]$,

(b) X jest zmienną losową o rozkładzie określonym gęstością

$$g(x) = \begin{cases} -0.5 \cdot x + 1, & \text{dla } x \in [0, 2], \\ 0, & \text{dla } x \notin [0, 2]. \end{cases}$$

W każdym z tych przypadków wylosuj 1000 wartości $f(X)$. Oczywiście spodziewamy się podobnej średniej w obu powyższych przypadkach. Czy można to samo powiedzieć o wariancji uzyskanych prób?

Zadanie 1.2. Dla całki określonej na przedziale nieskończonym

$$\int_0^\infty f(x)\,dx$$

możemy, dokonując zamiany zmiennych, sprowadzić całkę do przedziału skończonego, a następnie zamienić tę całkę na wartość oczekiwaną funkcji zmiennej losowej przyjmującej wartości z tego przedziału (na przykład – ale niekoniecznie – zmiennej losowej o rozkładzie jednostajnym). Inna możliwość to losowanie zmiennej mającej gęstość dodatnią w całym przedziale nieskończonym, którego dotyczy oryginalna całka (np. wykładniczy, normalny, lognormalny).

Przedstaw całkę

$$\int_0^\infty \frac{1}{1+x^{3.5}}\,dx$$

jako wartość oczekiwaną funkcji zmiennej losowej X o rozkładzie:

(a) wykładniczym o wartości oczekiwanej 1,
(b) wykładniczym o wartości oczekiwanej 10,
(c) normalnym $N(0,1)$,
(d) jednostajnym w przedziale $[0,1]$, zamiany zmiennej w całce dokonaj podstawieniem $t = \frac{1}{1+x^2}$,
(e) jednostajnym w przedziale $[0,1]$, zamiany zmiennej w całce dokonaj podstawieniem $t = \frac{1}{1+x^{3.5}}$,
(f) jednostajnym w przedziale $[0,1]$, zamiany zmiennej w całce dokonaj podstawieniem $t = e^{-x}$.

Zadanie 1.3. Wiadomo z podstawowej analizy matematycznej, że

$$\int_1^2 \frac{1}{x} = \ln 2,$$

a zatem pole powierzchni figury ograniczonej liniami

$$y = \frac{1}{x}, \qquad y = 0, \qquad x = 1, \qquad x = 2$$

wynosi $\ln 2$. Dokonując modyfikacji programu przedstawionego na listingu 1.1 na stronie 15, oszacuj liczbę $\ln 2$. Użyj $10\,000\,000$ punktów wylosowanych z prostokąta o wierzchołkach

$$(1,0), \qquad (2,0), \qquad (2,1), \qquad (1,1).$$

Odpowiedzi, wskazówki

Zadanie 1.1. Należy skorzystać z ogólnej zasady, że jeśli X ma gęstość $g(\cdot)$ to

$$Ef(X) = \int_{\mathbb{R}} f(x)g(x)\,dx.$$

Wariancja powinna być mniejsza w przypadku (b), zostanie to wyjaśnione w podrozdziale 4.4. Prosty sposób na generowanie próbek z rozkładu o zadanej gęstości podany jest w podrozdziale 3.1.4.

Zadanie 1.2. Wartość tej całki znamy z analizy matematycznej:

$$\frac{\pi/3.5}{\sin(\pi/3.5)}.$$

Zadanie 1.3. Wynik: $0.693\,29$.

ROZDZIAŁ 2

Generowanie liczb pseudolosowych

2.1. Liczby losowe

Przez generator liczb losowych rozumiemy pewien sposób, mechanizm otrzymywania ciągu zmiennych losowych

$$U_1, U_2, \dots,$$

przy czym

- każde U_i ma rozkład jednostajny na odcinku $[0, 1]$,
- zmienne są niezależne.

Pierwsza własność jest tylko pewną wygodną normalizacją (przy czym matematycznie domykanie przedziałów jest dla zmiennej typu ciągłego bez znaczenia).

Druga własność jest istotniejsza, oznacza w szczególności, że dla dowolnego i wartość U_i nie może w żaden sposób być przewidziana na podstawie wartości $U_1, U_2, \dots, U_{i-1}$.

W jaki sposób tak rozumiana losowość może być realizowana za pomocą komputera? Czy w ogóle jest to możliwe?

W znacznej mierze zależy to od sprzętu, który posiadamy, systemu operacyjnego oraz używanego oprogramowania.

Źródłem losowości, na podstawie którego można dokonać obliczeń prowadzących do ciągu realizacji zmiennych generatora, mogą być elementy takie jak: stan systemowego zegara, odstępy czasowe obserwowane podczas używania klawiatury, ruchy urządzeniem wskazującym, przerwania systemowe bądź sieciowe. System operacyjny może tego rodzaju przekształcane w źródło losowych bitów informacje zbierać i udostępniać aplikacjom uruchamianym na danej platformie.

Przykładem takiego właśnie podejścia jest `/dev/random` – urządzenie używane w systemie Linux od połowy lat dziewięćdziesiątych XX wieku [13].

Jednak źródło tak pojętej losowości jest mało wydajne i jest nieprzydatne w obliczeniach prowadzonych metodą Monte Carlo. Obliczenia numeryczne wymagają źródła o statystycznych właściwościach zbliżonych do źródła losowego, jednak dającego się przewidzieć, kontrolować, powtórzyć, bardzo szybko generować.

2.2. Liczby pseudolosowe

Generator liczb pseudolosowych jest algorytmem, który na podstawie zadanych parametrów wylicza wartości ciągu

$$u_1, u_2, \dots, u_k,$$

w taki sposób, by był on trudny do odróżnienia od realizacji zmiennych $U_1, U_2, \dots, U_k$ generatora liczb losowych.

Oznacza to między innymi, że jeżeli k jest dużą liczbą, to liczba wartości ciągu, które znajdą się w dowolnie ustalonym podprzedziale $[0, 1]$, powinna być proporcjonalna do jego długości. To odpowiada pierwszej z własności ciągu losowego – jednostajności rozkładu.

Natomiast druga własność – niezależność, powinna objawić się tym, że wśród liczb wygenerowanych ciągiem pseudolosowym nie wyłaniają się zauważalne wzorce, co oznacza zwłaszcza, że statystyczne testy nie mogą dawać podstaw do odrzucenia hipotezy o niezależności.

Pozostałe oczekiwania od generatorów liczb pseudolosowych to:

- duży okres generowanego ciągu[1],
- duża szybkość,
- możliwość powtórzenia eksperymentu (powtórnego wygenerowanie dokładnie tego samego ciągu),

[1] Ciąg $a_1, a_2, \dots$ nazywany jest okresowym, jeśli istnieje liczba $T > 0$ taka, że $a_{n+T} = a_n$ dla $n \in \mathbb{N}$. Najmniejsza taka liczba T jest nazywana okresem ciągu.

- przenośność (nie powinien być bardzo uzależniony od szczegółów operacji arytmetycznych wykonywanych na różnych platformach systemowo-sprzętowych).

Przykład 2.1 (Generator von Neumanna). *W 1949 roku von Neumann opisał metodę (wcześniej już znaną, gdyż koncepcja taka była opisywana już w XIII wieku) zwaną* middle square. *Niech m będzie ustaloną dodatnią liczbą całkowitą parzystą. Ciąg zaczyna się od dowolnej co najwyżej m-cyfrowej dodatniej liczby całkowitej x_0, następnie, aby otrzymać liczbę x_n, liczba x_{n-1} podnoszona jest do kwadratu, ewentualnie uzupełniana zerami z lewej strony do $2m$ cyfr, po czym jako x_n przyjmuje się liczbę utworzoną z m „środkowych" cyfr.*

Liczby tak tworzone mają właściwości, które sprawiają, że obecnie wyklucza się ich poważne zastosowanie.

Dzieląc każdą z tych liczb przez 10^m uzyskamy pseudolosowy ciąg liczb z przedziału $(0, 1)$. Na przykład dla $m = 4$, przyjmując jako pierwszy wyraz 894, dostaniemy kolejne wyrazy 7992, 8720, 384, 1474. Po podzieleniu przez 10^m będzie to ciąg 0.7992, 0.8720, 0.0384, 0.1474.

Losowość tego typu ciągu może być zilustrowana na diagramie, w którym dla ciągu $x_1, x_2, \ldots x_{2k}$ na wykresie zostanie umieszczonych k par postaci

$$(x_1/10^m, x_2/10^m), (x_3/10^m, x_4/10^m), \ldots, (x_{2k-1}/10^m, x_{2k}/10^m).$$

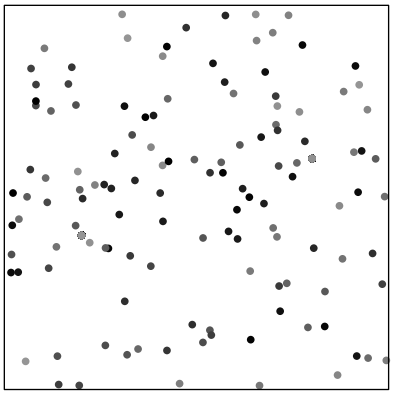
Dla $m = 6$, $k = 1000$, przyjmując jako stan początkowy x_0 liczbę $234\,569$, otrzymamy ciąg przedstawiony na rysunku 2.1 na następnej stronie.

Na rysunku tym mamy jednak zaskakująco mało punktów, jest ich 128. Wynika to z faktu, że tego rodzaju ciągi względnie szybko dochodzą do małych cykli i ostatecznie liczba różnych elementów w ciągu jest niewielka. Gdybyśmy taką liczbę różnych elementów uzyskiwanych dla poszczególnych ziaren (łącznie z tymi ziarnami) uśrednili, to na przykład dla $m = 2$ wynosi ona około 5.8, dla $m = 4$ około 43.7. Odpowiednie wielkości dla $m = 6$ (wyjaśniające małą liczbę punktów na rysunku 2.1 na następnej stronie) oraz $m = 8$ są rozwiązaniem zadania 2.2.

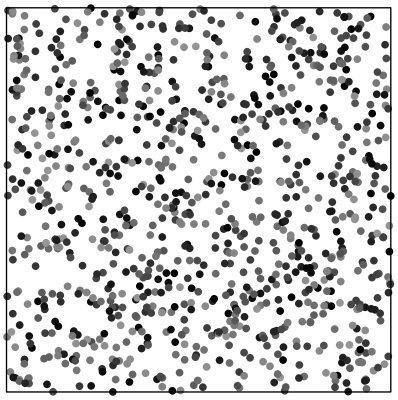
Oczywiście zwiększając m możemy doprowadzić analogiczne rysunki do bardziej oczekiwanej formy. Na przykład rysunek 2.2 na następnej stronie jest wykonany dla 1000 par wziętych z ciągu

von Neumanna dla $m = 12$. Nie wygląda gorzej od rysunku 2.3 na następnej stronie, przy tworzeniu którego użyto standardowego generatora `mt19937_64` w połączeniu z metodą uzyskiwania rozkładu jednostajnego z biblioteki standardowej `C++11` (zob. podrozdział 3.4).

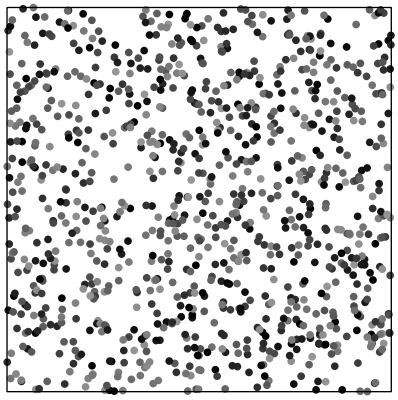
koniec przykładu 2.1



Rys. 2.1. 1000 kolejnych par wylosowanych generatorem von Neumanna dla 6 cyfr – różnych elementów jest tu wielokrotnie mniej niż 1000



Rys. 2.2. 1000 punktów z kwadratu jednostkowego wylosowanych generatorem von Neumanna (12 cyfr)



Rys. 2.3. 1000 punktów z kwadratu jednostkowego wylosowanych przy użyciu generatora `mt19937_64` w C++

2.3. Liniowe generatory liczb pseudolosowych

Najpopularniejsze generatory liczb pseudolosowych nazywane są liniowymi generatorami liczb pseudolosowych, uzyskuje się je za pomocą wzoru [29]

$$x_{i+1} = (ax_i + c) \mod m; \qquad u_{i+1} = x_{i+1}/m, \tag{2.1}$$

przy czym m jest ustaloną (dużą) liczbą naturalną. Zatem

$$x_i \in \{0, 1, \ldots, m-1\},$$

natomiast $u_i \in [0, 1)$. Na liczby u_i trzeba patrzeć po prostu jak na unormowany ciąg x_i, niekoniecznie zaś jak na docelowy ciąg pseudolosowy o rozkładzie jednostajnym (czy też dokładniej mówiąc naśladujący realizację ciągu niezależnych zmiennych o rozkładzie jednostajnym). Rozkład jednostajny w przedziale może być otrzymany na podstawie rozkładu równomiernego bardziej wyrafinowaną metodą niż dzieleniem ciągu (x_i) przez m. Wrócimy do tego w podrozdziale 3.4 przy okazji omawiania zagadnienia generowania liczb pseudolosowych w C++.

Szczególny przypadek $c = 0$ był rozważany i dyskutowany kilka lat wcześniej [21].

Angielska nazwa generatorów określonych formułą (2.1) to *linear congruential pseudo random number generator (LCG)*.

W książce *Sztuka programowania* [18] D.E. Knuth podaje, że aby uzyskać maksymalny okres ciągu wystarczy dla $c \neq 0$ spełnić poniższe warunki

- c oraz m są względnie pierwsze,
- każdy pierwszy dzielnik m jest także dzielnikiem $a-1$,
- jeżeli $4|m$ to także $4|(a-1)$.

Przykład 2.2. *Korzystając z powyższych wskazówek, dla $m = 32$ można przyjąć $a = 9$, $c = 15$. Wtedy, jeśli przyjmiemy początkowy wyraz $x_0 = 5$, to otrzymamy ciąg kolejnych wartości x_i:*

$$28, 11, 18, 17, 8, 23, 30, 29, 20, 3, 10, 9, 0, 15, 22, 21,$$
$$12, 27, 2, 1, 24, 7, 14, 13, 4, 19, 26, 25, 16, 31, 6, 5,$$
$$28, 11, 18, \ldots,$$

natomiast przyjmując $x_0 = 12$, otrzymamy

$$27, 2, 1, 24, 7, 14, 13, 4, 19, 26, 25, 16, 31, 6, 5, 28,$$
$$11, 18, 17, 8, 23, 30, 29, 20, 3, 10, 9, 0, 15, 22, 21, 12,$$
$$27, 2, 1, \ldots$$

Wartości u_i odpowiadające pierwszemu z nich to ciąg

$$0.875, 0.343\,75, 0.5625, 0.531\,25, 0.25, 0.718\,75, 0.9375, 0.906\,25,$$
$$0.625, 0.093\,75, 0.3125, 0.281\,25, 0, 0.468\,75, 0.6875, 0.656\,25,$$
$$0.375, 0.843\,75, 0.0625, 0.031\,25, 0.75, 0.218\,75, 0.4375, 0.406\,25,$$
$$0.125, 0.593\,75, 0.8125, 0.781\,25, 0.5, 0.968\,75, 0.1875, 0.156\,25,$$
$$0.875, 0.343\,75, 0.5625, \ldots$$

W jakiś prosty sposób te liczby mogą naśladować realizację ciągu zmiennych o rozkładzie jednostajnym w $[0, 1]$. Zwróćmy uwagę na parzystość liczb x_i, zmienia się ona zbyt regularnie. Wyrażając tę własność za pomocą liczb u_i: ciąg $\lfloor 32 \cdot u_i \rfloor \bmod 2$ to ciąg

$$0, 1, 0, 1, 0, 1, \ldots,$$

a więc ciąg okresowy o okresie 2. Teraz u_i już nie wydają się tak dobrze naśladować realizacji ciągu niezależnych zmiennych o rozkładach jednostajnych.

Ta obserwacja pokazuje, że generatory LCG mają pewne oczywiste słabości, jeśli chodzi o statystyczne własności, wiele z nich dotyczy właśnie młodszych bitów. Dlatego niektóre praktyczne generatory korzystające z koncepcji LCG pracują (i przechowują stan) na słowach $2k$ bitowych, natomiast na wyjściu dla użytkownika umieszczają tylko k starszych bitów.

Jeżeli $c = 0$, zaś m jest liczbą pierwszą, to [18] dla maksymalizacji okresu wystarczy, by liczba a była pierwiastkiem pierwotnym modulo m, to znaczy, by

- $a^{m-1} = 1 \mod m$,
- $a^j \neq 1 \mod m$, dla $j = 1, 2, \ldots m-2$.

Jeśli taki pierwiastek istnieje, to maksymalny okres wynosi $m-1$ (liczba 0 w oczywisty sposób musi być z tak generowanego ciągu wykluczona). Pierwiastki modulo m istnieją tylko w szczególnych przypadkach: $m = 2$, $m = 4$, m będące potęgą liczby pierwszej nieparzystej, m będące podwojoną potęgą liczby pierwszej nieparzystej [27].

Na przykład dla $m = 31$ można przyjąć za a dowolną z liczb

$$3, 11, 12, 13, 17, 21, 22, 24.$$

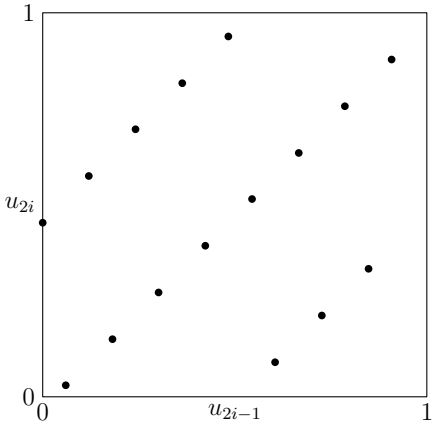
Kilka przykładowych sprawdzonych układów stałych dla LCG (gwarantują one maksymalny okres oraz dobre właściwości wielu testów statystycznych [18]) umieszczono w tabeli 2.1.

Tab. 2.1. Przykładowe stałe używane w generatorach LCG

m	a	c	uwagi
$2^{31}-1$	16 807	0	`minstd_rand0` w C++11
$2^{31}-1$	48 271	0	`minstd_rand` w C++11
$2^{31}-1$	1 226 874 159	0	
2^{32}	134 775 813	1	Turbo Pascal 4.0
2^{24}	16 598 013	12 820 163	Microsoft Visual Basic 6

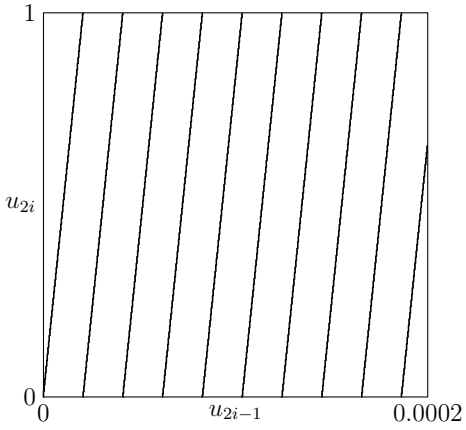
Poważną wadą, sprzeczną z intuicyjnym rozumieniem generowania liczb pseudolosowych, jest liniowość struktur otrzymywanych w kostce d-wymiarowej. Łatwo możemy to zaobserwować, robiąc wykres punktów (u_{2i-1}, u_{2i}) dla $i = 1, 2, \ldots$, gdzie u_i są liczbami rzeczywistymi z przedziału $[0, 1)$ otrzymywanymi według wzoru (2.1). Przykładowa

ilustracja dla generatora z przykładu 2.2 pokazana została na rysunku 2.4. Możemy takie struktury zaobserwować także w przypadku realnych stałych – rysunek 2.5.



$a = 9,\ c = 15,\ m = 32$

Rys. 2.4. Punkty postaci (u_{2i-1}, u_{2i}) dla generatora z przykładu 2.2



$a = 48\,271,\ c = 0,\ m = 2\,147\,483\,647$

Rys. 2.5. Punkty postaci (u_{2i-1}, u_{2i}) dla generatora z przykładu 2.2

2.3.1. Inne metody generowania ciągów pseudolosowych

Wśród innych sposobów tworzenia generatorów warto wymienić:

- użycie rekurencji wyższego rzędu, np.

$$x_{i+1} = (a_0 x_i + a_1 x_{i-1} + \ldots a_p x_{i-p}) \mod m,$$

  co pozwala na uzyskiwanie dużo większych okresów (przy tym samym zakresie liczb),
- użycie kilku generatorów i podawanie jako wyniku pewnej ich funkcji (np. sumy),
- tworzenie generatorów dokonujących obliczeń w strukturach ciał Galois GF(2) lub GF(2^p), które dają jako wynik strumień bitów – na przykład rejestry przesuwające [17].

Zadania

Zadanie 2.1. Dla stałych z tabeli 2.1 na stronie 31 sprawdź, czy spełnione są warunki do osiągnięcia maksymalnej wielkości okresu ciągu.

Zadanie 2.2. Rozważmy użycie generatora von Neumanna dla liczb m cyfrowych i niech x będzie liczbą całkowitą mniejszą niż 10^m i niech k_x będzie liczbą różnych elementów w wygenerowanym metodą *middle square* ciągu

$$x_0, x_1, x_2, \ldots$$

dla $x_0 = x$. Wyznacz średnią wartość k_x, to znaczy

$$\frac{1}{10^m} \sum_{x=0}^{10^m - 1} k_x,$$

gdy
(a) $m = 6$,
(b) $m = 8$.

Zadanie 2.3. Zaprogramuj generowanie $1\,000\,000$-elementowego ciągu liczb rzeczywistych z przedziału $[0, 1)$, na podstawie współczynników ostatniego wiersza tabeli 2.1 na stronie 31, i oblicz, ile elementów

ciągu znalazło się w przedziałach

$$[0, 0.25), \quad [0.25, 0.5), \quad [0.5, 0.75), \quad [0.75, 1).$$

Odpowiedzi, wskazówki

Zadanie 2.2. Dla $m = 6$ średnia wartość to 326.198, dla $m = 8$ jest to 4403.29.

Zadanie 2.3. Wyniki: $250\,555$, $249\,525$, $249\,492$, $250\,428$.

ROZDZIAŁ 3

Generowanie próbek z różnych rozkładów

W rozdziale tym rozważymy zarówno generowanie próbek z rozkładów jednowymiarowych, jak i wielowymiarowych. Na szczególną uwagę zasługują metody uniwersalne (odwracanie dystrybuanty, metoda akceptacji/odrzucania) oraz metody związane z generowaniem rozkładów normalnych. Wszystkie one sprowadzą się do pewnego przekształcania zmiennych losowych o rozkładach jednostajnych, dla generowania których bazą są ciągi pseudolosowe omawiane w rozdziale 2.

3.1. Rozkłady jednowymiarowe

3.1.1. Rozkład dyskretny równomierny

Fakt 3.1. *Niech m będzie liczbą całkowitą dodatnią i niech U ma rozkład jednostajny w $[0,1]$. Wtedy*

$$X = \lfloor U \cdot m \rfloor$$

ma rozkład równomierny w

$$\{0, 1, \ldots, m-1\}.$$

Dowód. Rzeczywiście, dla dowolnego $k \in \{0, 1, \ldots, m-1\}$ otrzymujemy

$$\begin{aligned}\Pr(X = k) &= \Pr(\lfloor U \cdot m \rfloor = k) = \Pr(k \leqslant U \cdot m < k+1) = \\ &= \Pr(U \in [k/m, k/m + 1/m)) = 1/m.\end{aligned}$$

□

3.1.2. Rozkład zero-jedynkowy i rozkład dwumianowy

Fakt 3.2. *Niech $p \in (0,1)$ i niech U ma rozkład jednostajny w $[0,1]$. Wtedy*

$$X = \begin{cases} 1, & U < p, \\ 0, & U \geqslant p, \end{cases}$$

ma rozkład zero-jedynkowy z parametrem p.

Powyższy fakt jest oczywisty, gdyż $\Pr(U < p) = p$ dla każdego $p \in (0,1)$.

Rozkład dwumianowy[1] dla n prób możemy uzyskać na przykład sumując n zmiennych o rozkładzie zero-jedynkowym. Przy dużej liczbie n skuteczniejsze może być skorzystanie z ogólnej metody generowania próbek z rozkładu dyskretnego, omówionej w punkcie 3.1.3.

3.1.3. Dowolny rozkład dyskretny o wartościach ograniczonych od dołu

Fakt 3.3. *Załóżmy, że X przyjmuje wartości*

$$x_1 < x_2 < \ldots$$

z prawdopodobieństwami $p_1, p_2, \ldots$ i niech U ma rozkład jednostajny w $[0,1]$. Wtedy zmienna losowa

$$Y = \begin{cases} x_1, & U < f_1, \\ x_2, & U \in [f_1, f_2), \\ \ldots, & \ldots \end{cases}$$

gdzie $f_i = \sum_{j=1}^{i} p_j, i = 1, 2, \ldots$, ma rozkład identyczny z rozkładem X.

Rzeczywiście, przyjmując $f_0 = 0$ mamy dla $k \in \{1, 2, \ldots\}$, że

$$\Pr(Y = k) = \Pr(U \in [f_{i-1}, f_i)) = f_i - f_{i-1} = p_i.$$

[1] Inaczej nazywany rozkładem Bernoulliego, czyli rozkład liczby sukcesów w schemacie n niezależnych prób, w których w każdej osiąga się sukces z prawdopodobieństwem p.

3.1.4. Metoda odwracania dystrybuanty

Uniwersalną metodą w przypadku rozkładów jednowymiarowych (nie zawsze najbardziej wydajną) jest metoda wykorzystująca odwracanie dystrybuanty.

Szczególnie prostą formę uzyskuje się w przypadku, gdy dystrybuanta jest funkcją ściśle rosnącą.

Twierdzenie 3.1. *Niech F będzie ściśle rosnącą dystrybuantą, a U niech będzie zmienną losową o rozkładzie jednostajnym w $[0,1]$. Wtedy $F^{-1}(U)$ ma rozkład zadany dystrybuantą F.*

Dowód. Rzeczywiście, niech $x \in \mathbb{R}$. Wtedy

$$\begin{aligned}\Pr(F^{-1}(U) \leqslant x) &= \Pr(F(F^{-1}(U)) \leqslant F(x)) = \\ &= \Pr(U \leqslant F(x)) = F(x).\end{aligned}$$

□

W bardzo wielu przypadkach, nawet dla rozkładów ciągłych, dystrybuanta w pewnych odcinkach jest stała. Taka funkcja, w ścisłym sensie tego słowa, nie daje się odwrócić.

Nawet tak podstawowy rozkład ciągły jak rozkład wykładniczy nie ma dystrybuanty ściśle rosnącej, gdyż w całym przedziale $(-\infty, 0]$ ma ona wartość zero, a więc przeciwobraz zera ma nieskończenie wiele elementów.

Jednoznaczność uzyskamy przyjmując w miejsce odwrotności funkcję kwantylową $Q(p)$ zdefiniowaną dla

$$p \in (0,1) \tag{3.1}$$

wzorem

$$Q(p) = \inf\{t \in \mathbb{R} \colon p \leqslant F(t)\}. \tag{3.2}$$

W przedziałach, w których dystrybuanta jest rosnąca, Q niczym nie różni się od F^{-1}. Dla $p = 0$ oraz $p = 1$ w wielu przypadkach wzór (3.2) każe liczyć infimum ze zbioru pustego bądź ze zbioru zawierającego dowolnie małe wartości, stad ograniczenie (3.1). Ograniczenia te nie są istotne, w praktyce oczywiście by obliczać $Q(U)$ musimy czasem uzupełnić wartość Q dla 0 lub 1 – ale możemy to robić tak, jak jest nam wygodnie, bo nie wpłynie to na rozkład zmiennej $Q(U)$.

Tak określana funkcja kwantylowa często nazywana jest uogólnioną funkcją odwrotną do dystrybuanty, bywa też oznaczana tak samo jak funkcja odwrotna do dystrybuanty.

Funkcja $Q(\cdot)$ ma własność

$$Q(p) \leqslant x \iff p \leqslant F(x) \qquad \text{dla dowolnego } p \in [0,1]. \tag{3.3}$$

Prawa strona implikuje lewą wprost z określenia (3.2). Natomiast wynikanie w drugą stronę jest prostym wnioskiem z prawostronnej ciągłości dystrybuanty.

Twierdzenie 3.2. *Niech F będzie dystrybuantą, Q jej funkcją kwantylową i niech U będzie zmienną losową o rozkładzie jednostajnym w $[0,1]$. Wtedy $Q(U)$ ma rozkład zadany dystrybuantą F.*

Dowód. Niech $x \in \mathbb{R}$. Wtedy

$$\Pr(Q(U) \leqslant x) = [\text{na mocy (3.3)}] = \Pr(U \leqslant F(x)) = F(x).$$

□

Z wielu względów – co widać na przykładzie podrozdziału 3.1.3 – wygodnie jest rozważać dla podstawowego rozkładu jednostajnego tylko przedziały lewostronnie domknięte, prawostronnie otwarte. Taka też jest tradycja w wielu językach programowania, które oferują programistom pseudolosowe generowanie rozkładu jednostajnego, dając wyniki właśnie w przedziale lewostronnie domkniętym, prawostronnie otwartym.

Tymczasem funkcja kwantylowa (co wynika z jej definicji oraz z prawostronnej ciągłości dystrybuanty) jest lewostronnie ciągła. Najprostszym sposobem poradzenia sobie z tym jest wyznaczenie odpowiedniej funkcji kwantylowej i – w przypadku gdy jej definicja będzie oparta o przedziały otwarte bądź prawostronnie domknięte – zmienienie jej na niemal taką samą funkcję $h(\cdot)$, ale opartą na przedziałach lewostronnie domkniętych. Brakującą w punkcie 0 wartość można uzupełnić dowolnie, ilustruje to przykład 3.1.

Przykład 3.1. *Rozważmy dystrybuantę*

$$F(x) = \begin{cases} 0, & \text{dla } x < 1, \\ 0.3(x-1), & \text{dla } x \in [1,2), \\ 0.5 + 0.4(x-2)^2, & \text{dla } x \in [2,3), \\ 1 & \text{dla } x \geqslant 3. \end{cases}$$

Dystrybuanta określona w tym przykładzie jest jak widać dystrybuantą pewnej zmiennej losowej mieszanej z dodatnimi prawdopodobieństwami realizacji wartości 2 oraz 3.

Funkcja kwantylowa konstruowana jest dość łatwo. W otwartych przedziałach, w których dystrybuanta rośnie, zwyczajnie ją odwracamy. Dla wartości p, dla których istnieje wiele x, dla których $F(x) = p$, bierzemy infimum tych wartości x. Korzystamy z gwarantowanej lewostronnej ciągłości i ostatecznie dostajemy dla $p \in (0,1)$

$$Q(p) = \begin{cases} \frac{10}{3}p + 1, & \text{dla } p \in (0, 0.3], \\ 2 & \text{dla } p \in (0.3, 0.5], \\ 2 + \sqrt{\frac{5}{2}(p-0.5)}, & \text{dla } p \in (0.5, 0.9], \\ 3, & \text{dla } p \in (0.9, 1). \end{cases}$$

Korzystając z uwag na temat domykania przedziałów (i preferowania domykania lewostronnego przedziałów), można określić dla wszystkich $p \in [0,1)$ funkcję

$$h(p) = \begin{cases} \frac{10}{3}p + 1, & \text{dla } p \in [0, 0.3), \\ 2 & \text{dla } p \in [0.3, 0.5), \\ 2 + \sqrt{\frac{5}{2}(p-0.5)}, & \text{dla } p \in [0.5, 0.9), \\ 3, & \text{dla } p \in [0.9, 1). \end{cases}$$

Oczywiste jest, że jeśli U ma rozkład jednostajny w przedziale $[0,1]$ to $h(U)$ oraz $Q(U)$ mają identyczny rozkład (ich wartości różnią się na zbiorze miary zero). Funkcja $h(\cdot)$ jest nieco bardziej konsekwentna (określona na jednakowego typu przedziałach) od $Q(\cdot)$ oraz lepiej pasuje do praktycznie spotykanej sytuacji, w której system programowania dostarcza funkcji generującej jednostajnie wartości właśnie z $[0,1)$.

W C++ realizacja $h(\cdot)$ mogłaby wyglądać na przykład tak – sprawdzane są tylko proste warunki z ostrymi nierównościami, liczba wyborów warunkowych jest o jeden mniejsza niż liczba przypadków w opisie funkcji, w wyborach warunkowych wystarczy tylko rozważanie przypadku spełnienia warunku:

```
double h(double x) {
    if (x<0.3)
        return 10/3.0*x + 1.0;
    if (x<0.5)
        return 2.0;
    if (x<0.9)
        return 2.0 + sqrt(5.0/2.0*(x-0.5));
    return 3;
}
```

koniec przykładu 3.1

Przykład 3.2. *Rozważmy rozkład wykładniczy z parametrem częstotliwości μ, to znaczy rozkład o dystrybuancie*

$$F(x) = \begin{cases} 0, & \text{dla } x < 0, \\ 1 - \exp(-\mu x), & \text{dla } x \geqslant 0. \end{cases}$$

Funkcja kwantylowa dla tego rozkładu to

$$Q(p) = \frac{-\ln(1-p)}{\mu}, \qquad \text{dla } p \in (0,1).$$

Jeżeli U ma rozkład jednostajny na $[0,1]$, to $Q(U)$ ma rozkład wykładniczy z parametrem μ. Oczywiście $Q(\cdot)$ powinno być uzupełnione w 0 i w 1, ale może być uzupełnione dowolnie (te punkty są dla U osiągane z prawdopodobieństwem 0, a więc nie mają wpływu na rozkład).

Dla praktycznej sytuacji, gdy generator będzie dostarczał wartości z przedziału $[0,1)$, konieczne jest tylko uzupełnienie w 0 (konieczne, bo ta wartość może się trafić, prawdopodobieństwo jest małe, ale nie jest zerowe dla generatora liczb pseudolosowych).

W tym wypadku naturalne jest uzupełnienie wartością 0 dla $p = 0$, uzyskany wzór to

$$h(p) = \frac{-\ln(1-p)}{\mu}, \qquad \text{dla } p \in [0, 1).$$

Na koniec zauważmy jeszcze, że jeśli U ma rozkład jednostajny w $[0, 1]$, to taki sam rozkład ma również $1 - U$. Zatem skoro $\frac{-\ln(1-U)}{\mu}$ ma rozkład wykładniczy, to wynika z tego, że także $\frac{-\ln U}{\mu}$ ma rozkład wykładniczy z parametrem μ. Pamiętajmy tylko, że dla praktycznej sytuacji tym razem musimy zadbać, żeby U `nie` osiągnęło wartości 0.

Jeśli chcielibyśmy korzystać z obu wzorów do generowania rozkładu wykładniczego, to znaczy z

$$\frac{-\ln(1-U)}{\mu} \qquad \text{oraz} \qquad \frac{-\ln U}{\mu},$$

– a może to być potrzebne choćby przy korzystaniu z metody zmiennych antytetycznych (podrozdział 4.2) – to najwygodniejszy byłby generator liczb pseudolosowych o rozkładzie jednostajnym dający wartości tylko z przedział otwartego $(0, 1)$. Można go zorganizować na przykład tak, jak to pokazano w listingu 1.1 na stronie 15. W każdym razie problemu tego zignorować nie wolno, bo mogłoby to doprowadzić do nieprawidłowego zachowania się programu.

..... *koniec przykładu 3.2* .....

Metoda odwracania dystrybuanty jest szczególnie atrakcyjna, gdy mamy zwartą postać dystrybuanty i to taką, którą można łatwo odwrócić analitycznie. Takie były dystrybuanty w przykładach 3.2 oraz 3.1. W innych sytuacjach może być trudniej, na przykład dla rozkładu normalnego dystrybuanta nie ma postaci zwartej. Nawet w takiej sytuacji da się tę metodę stosować: skoro da się dystrybuantę obliczać (choćby numerycznym całkowaniem), to da się ją numerycznie odwracać (na przykład ogólnymi metodami rozwiązywania równań nieliniowych).

Jednak dla szczególnych sytuacji – jak choćby właśnie rozkładu normalnego, kluczowego dla wielu statystycznych modeli oraz dla przeprowadzania symulacji i oceny ich wyników – opracowano szczególne sposoby, całkowicie odchodzące od koncepcji odwracania dystrybuanty. Takie zagadnienia są tematem podrozdziału 3.1.5.

3.1.5. Rozkład normalny

Rozkład normalny $N(\mu, \sigma)$ ze średnią μ i odchyleniem standardowym σ ma funkcję gęstości daną wzorem

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-\mu)^2}{2\sigma^2}}.$$

Dystrybuanta jest więc równa

$$F(x) = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{x} e^{\frac{-(t-\mu)^2}{2\sigma^2}}\, dt,$$

nie ma ona zwartej postaci, nie jest więc łatwa w odwracaniu.

Zauważmy jeszcze tylko, że ze względu na możliwość standaryzacji, czyli ze względu na to, że

$$Y \sim N(\mu, \sigma) \quad \Longleftrightarrow \quad \frac{Y-\mu}{\sigma} \sim N(0,1), \quad \text{dla dow. } \mu \in \mathbb{R}, \sigma > 0,$$

przy generowaniu próbek rozkładu normalnego wystarczy „nauczyć się" generowania próbek rozkładu $N(0,1)$. Mnożąc te próbki przez σ i dodając do nich μ dostaniemy próbki z rozkładu $N(\mu, \sigma)$.

Ze względu na szczególną przydatność rozkładu $N(0,1)$, jego gęstość oraz dystrybuanta będą oznaczane zgodnie z tradycją przez φ oraz Φ, to znaczy

$$\varphi(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-x^2}{2}}, \qquad \Phi(x) = \int_{-\infty}^{x} \varphi(t)\, dt. \tag{3.4}$$

Metody bazujące na Centralnym Twierdzeniu Granicznym

Centralne Twierdzenie Graniczne ma wiele form i odmian. Przypomnijmy, że ich istotą jest to, że przy odpowiednich założeniach dystrybuanta sumy ciągu zmiennych losowych dla dużej liczby składników bliska jest dystrybuanty rozkładu normalnego.

Poniżej przytoczona zostanie jedna z podstawowych postaci tej rodziny twierdzeń.

Twierdzenie 3.3 (Centralne twierdzenie Lindeberga–Lévy'ego [3]). *Jeżeli $X_1, X_2, X_3, \ldots$ jest ciągiem niezależnych zmiennych losowych o jednakowym rozkładzie o wartości przeciętnej μ i skończonej wariancji $\sigma^2 > 0$, to*

$$\lim_{n\to\infty} \Pr\left(\frac{\sum_{i=1}^{n} X_i - n\mu}{\sqrt{n}\sigma} \leqslant x\right) = \Phi(x), \qquad \textit{dla dowolnego } x \in \mathbb{R}.$$

Suma wielu zmiennych o tym samym rozkładzie (o ile mają skończoną i dodatnią wariancję) ma rozkład zbliżony do normalnego, więc na przykład prosta do generowania suma

$$X_1 + X_2 + \ldots X_{12} - 6$$

– gdzie X_i mają rozkład jednostajny na $[0, 1]$ – ma rozkład zbliżony do $N(0, 1)$. Zgodność średniej i wariancji jest oczywista, w przypadku zmiennych niezależnych wariancja sumy to suma wariancji, zaś wariancja rozkładu jednostajnego w przedziale $[a, b]$ to $\frac{(b-a)^2}{12}$.

Alternatywnie może to być suma

$$X_1 + X_2 + \ldots X_{12},$$

gdzie X_i są niezależnymi zmiennymi losowymi o rozkładzie jednostajnym w $[-0.5, 0.5]$.

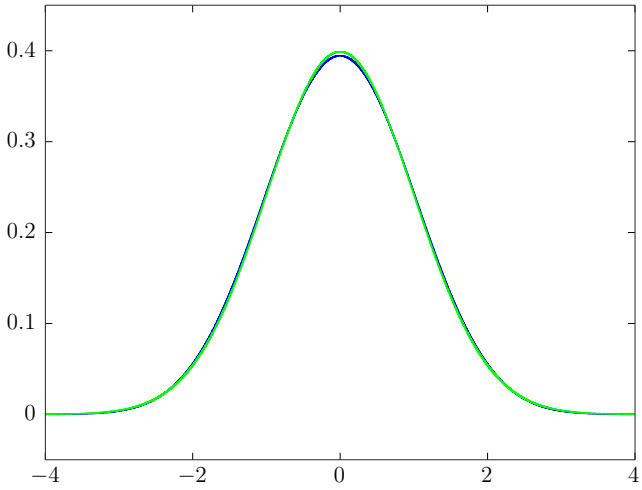
Oczywiście mogłoby się nasunąć pytanie czy 12 to naprawdę już „duża" liczba. Okazuje się, że w przypadku sumowania rozkładów jednostajnych, 12 zmiennych daje dobre przybliżenie rozkładu normalnego, gęstości pokazane są na rysunku 3.1 na następnej stronie.

Na tym rysunku linia zielona to gęstość rozkładu $N(0, 1)$, natomiast linia niebieska to gęstość sumy 12 zmiennych jednostajnych na $[0, 1]$ pomniejszonej o 6.

Oczywiście powiększając liczbę zmiennych X_i (jednocześnie dokonując odpowiedniego skalowania), udoskonalamy podobieństwo do rozkładu normalnego, jednak efektywność takiego rozwiązania spada.

Transformacja Boxa–Mullera

Duża część efektywnych metod generowania próbek rozkładu normalnego oparta jest na transformacji Boxa–Mullera. Przekształcenie to zostało opisane w twierdzeniu 3.4.



Rys. 3.1. Porównanie gęstości rozkładu normalnego(linia zielona) z gęstością sumy 12 niezależnych zmiennych o rozkładzie jednostajnym (linia niebieska)

Twierdzenie 3.4 (Box–Muller [5]). *Jeżeli D jest zmienną losową o rozkładzie wykładniczym z wartością oczekiwaną 2 i jeżeli A jest zmienną losową o rozkładzie jednostajnym w $[0, 2\pi]$, D i A niezależne, to zmienne losowe*

$$X = \sqrt{D} \cdot \cos(A), \qquad Y = \sqrt{D} \cdot \sin(A)$$

są niezależne i obie mają rozkład $N(0, 1)$.

Dowód. Para zmiennych losowych (X, Y) powstaje z pary zmiennych (D, A), co można zapisać jako

$$(X, Y) = \psi(D, A).$$

W takiej sytuacji dla gęstości mamy

$$f_{X,Y}(x, y) = f_{D,A}(\psi^{-1}(x, y)) \cdot \det J_{\psi^{-1}}(\psi^{-1}(x, y)), \tag{3.5}$$

gdzie J_h oznacza macierz Jacobiego przekształcenia h.

Funkcja $\psi(\cdot, \cdot)$ to w naszym przypadku

$$\psi(d, a) = (\sqrt{d} \cos a, \sqrt{d} \sin a),$$

jej jakobian to

$$\det J_{\psi}(d,a) = \begin{vmatrix} \frac{1}{2\sqrt{d}}\cos a & \frac{1}{2\sqrt{d}}\sin a \\ -\sqrt{d}\sin a & \sqrt{d}\cos a \end{vmatrix} = \frac{1}{2},$$

stąd

$$\det J_{\psi^{-1}}(x,y) = 2.$$

Zatem, korzystając z niezależności D i A oraz ze znajomości ich rozkładów, możemy zapisać

$$f_{X,Y}(x,y) = \frac{1}{2}e^{-\frac{x^2+y^2}{2}} \cdot \frac{1}{2\pi} \cdot 2 = \frac{1}{2\pi}e^{-\frac{x^2+y^2}{2}}. \tag{3.6}$$

Mając gęstość łączną wektora (X,Y), możemy policzyć gęstość X:

$$\begin{aligned} f_X(x) &= \int_{-\infty}^{\infty} \frac{1}{2\pi}e^{-\frac{x^2+y^2}{2}}\,dy = \frac{1}{\sqrt{2\pi}}e^{\frac{-x^2}{2}} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}}e^{\frac{-y^2}{2}}\,dy = \\ &= \frac{1}{\sqrt{2\pi}}e^{\frac{-x^2}{2}}. \end{aligned}$$

W podobny sposób wykażemy, że

$$f_Y(y) = \frac{1}{\sqrt{2\pi}}e^{\frac{-y^2}{2}}.$$

Tak więc X oraz Y mają rozkład $N(0,1)$. W połączeniu z formułą (3.6) dostajemy

$$f_{X,Y}(x,y) = f_X(x) \cdot f_Y(y),$$

a więc została wykazana także niezależność X i Y. □

Powyższe twierdzenie daje następującą metodę wygenerowania dwu liczb z_1, z_2 pseudolosowych z rozkładu $N(0,1)$ (wykorzystany został do tego sposób generowania zmiennej o rozkładzie wykładniczym, omówiony w przykładzie 3.2 na stronie 40):

1. Wygeneruj u_1, u_2 pseudolosowe z rozkładu jednostajnego na $[0,1)$.
2. Oblicz $r = \sqrt{-2 \cdot \ln(1-u_1)}$.
3. Oblicz $a = 2\pi \cdot u_2$.
4. Oblicz $z_1 = r\cos a$, $z_2 = r\sin a$.

Na listingu 3.1 pokazana jest przykładowa realizacja w C++, która na przemian generuje dwie liczby i zwraca pierwszą z nich, albo nic nie generuje, a jedynie zwraca drugą liczbę z poprzednio wygenerowanej pary. Stan przechowywany jest w lokalnej zmiennej statycznej, która charakteryzuje się tym, że jej wartość nie jest tracona w momencie zakończenia funkcji.

```
#include <random>
#include <cmath>
using namespace std;
double runif() {
// przykładowa funkcja dająca wyniki pseudolosowe
// z rozkładu jednostajnego na [0,1)
    static mt19937_64 gen;
    static uniform_real_distribution<double> u;
    return u(gen);
}
double norm() {
    static bool generuj = false;
    static double n2;
    generuj = !generuj;
    if (generuj) {
        double u1 = runif();
        double u2 = runif();
        double r = sqrt(-2*log(1-u1));
        double a = 2*M_PI*u2;
        double n1 = cos(a)*r;
        n2 = sin(a)*r;
        return n1;
    } else
        return n2;
}
```

Listing 3.1. Realizacja algorytmu wynikającego z twierdzenia 3.4

Metoda Marsaglia–Braya

Łącząc idee transformacji Boxa–Mullera i jej graficznej interpretacji oraz obliczania rozkładu warunkowego, Marsaglia i Bray opracowali

modyfikację metody, w której nie wymagane jest obliczania funkcji trygonometrycznych [23]:

1. Wygeneruj x, y z rozkładu jednostajnego na $(-1, 1)$.
2. Oblicz $u := x^2 + y^2$.
3. Jeżeli $u \geqslant 1$ powróć do kroku 1.
4. $t := \sqrt{-2\ln(1-u)/u}$.
5. $z_1 := xt$, $z_2 := yt$.

Uzasadnienie sformułowane zostanie jako twierdzenie 3.5.

Zauważmy jeszcze tylko, że kroki 1, 2, 3 tej metody generują parę (x, y), będącą realizacją zmiennej losowej dwuwymiarowej o rozkładzie jednostajnym w kole jednostkowym.

Twierdzenie 3.5. *Niech (X, Y) będzie dwuwymiarową zmienną losową o rozkładzie jednostajnym w kole*

$$\{(x, y) \in \mathbb{R}^2 \colon x^2 + y^2 < 1\}$$

i niech

$$U = X^2 + Y^2, \quad T = \sqrt{-2\ln(1-U)/U}, \quad Z_1 = X \cdot T, \quad Z_2 = Y \cdot T.$$

Zmienne losowe Z_1 oraz Z_2 są niezależne i mają rozkład $N(0, 1)$.

Dowód. Para (X, Y) ma rozkład jednostajny w kole, dlatego prawdopodobieństwo realizacji w danym podzbiorze koła jest proporcjonalne do jego powierzchni, stąd

$$\Pr(U \leqslant s) = \frac{[\text{pole koła o promieniu } \sqrt{s}]}{[\text{pole koła jednostkowego}]} = s, \qquad \text{dla } s \in [0, 1]$$

zatem U ma rozkład jednostajny w $[0, 1]$. Niech A będzie kątem promienia wodzącego punktu (X, Y). Zauważmy, że

$$\Pr(A \leqslant t) = \frac{t}{2\pi}, \qquad \text{dla } t \in [0, 2\pi],$$

a więc zatem A ma rozkład jednostajny w $[0, 2\pi]$.

Rozważenie dystrybuanty łącznej

$$\Pr(U \leqslant s, A \leqslant t) = \frac{\pi \cdot (\sqrt{s})^2 \cdot \frac{t}{2\pi}}{\pi} = \Pr(U \leqslant s) \cdot \Pr(A \leqslant t)$$

wykazuje, że U oraz A są niezależne.

Zauważmy, że w takim razie $D = -2\ln(1-U)$ ma rozkład wykładniczy z wartością oczekiwaną 2 i D nie zależy od A. Ponadto

$$Z_1 = \frac{X}{\sqrt{X^2+Y^2}} \cdot \sqrt{D} = \cos(A) \cdot \sqrt{D},$$
$$Z_2 = \frac{Y}{\sqrt{X^2+Y^2}} \cdot \sqrt{D} = \sin(A) \cdot \sqrt{D}$$

i teza twierdzenia wynika z twierdzenia 3.4. □

Jakie mogą być korzyści ze stosowania procedury będącej konsekwencją twierdzenia 3.5 zamiast tej wynikającej z twierdzenia 3.4? Z jednej strony przy stosowaniu metody Marsaglia–Braya nie trzeba obliczać funkcji trygonometrycznych. Z drugiej strony traci się część generowanych pomocniczych liczb pseudolosowych – średnio

$$1 - \frac{\pi}{4} \approx 21\%.$$

Innymi słowy dla uzyskania próby 1000 realizacji rozkładu normalnego potrzeba średnio około 1273 liczb z rozkładu jednostajnego.

Stosowanie metody Marsaglia–Braya zamiast Boxa–Mullera jest uzależnione od tego, jak w danym systemie przedstawia się sytuacja kosztów poszczególnych składowych całego procesu – a więc obliczania funkcji trygonometrycznych, logarytmicznych, różnych operacji arytmetycznych, losowania z rozkładu jednostajnego.

3.2. Rozkład wielowymiarowy normalny

Zmienna losowa n-wymiarowa X ma rozkład normalny wielowymiarowy $N(\boldsymbol{\mu}, \Sigma)$, gdzie $\boldsymbol{\mu}$ jest kolumną n wartości oczekiwanych, a Σ jest $n \times n$ macierzą kowariancji (która jest symetryczna i dodatnio określona), wtedy i tylko wtedy gdy gęstość X wyraża się wzorem

$$f_X(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^n \det \Sigma}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x}-\boldsymbol{\mu})\right).$$

Przy działaniach elementy z $\mathbb{R}^n$ traktujemy jako macierze jednokolumnowe.

Niech teraz zmienna losowa n-wymiarowa Y dana będzie formułą

$$Y = AX + B, \tag{3.7}$$

gdzie A – macierz $n \times n$, B – kolumna n-elementowa.

Jakobianem przekształcenia (3.7) jest $\det A$, stąd gęstość Y – przy wykorzystaniu formuły (3.5) – wynosi

$$\begin{aligned} f_Y(\boldsymbol{y}) =& \frac{1}{\det A} \cdot \frac{1}{\sqrt{(2\pi)^n \det \Sigma}} \cdot \\ & \cdot \exp\left(-\frac{1}{2}(A^{-1}(\boldsymbol{y} - B) - \boldsymbol{\mu})^T \Sigma^{-1} (A^{-1}(\boldsymbol{y} - B) - \boldsymbol{\mu})\right) = \\ =& \frac{1}{\sqrt{(2\pi)^n \det(A\Sigma A^T)}} \cdot \\ & \cdot \exp\left(-\frac{1}{2}(\boldsymbol{y} - A\boldsymbol{\mu} - B)^T (A\Sigma A^T)^{-1} (\boldsymbol{y} - A\boldsymbol{\mu} - B)\right), \end{aligned}$$

czyli

$$Y \sim N(A\boldsymbol{\mu} + B, A\Sigma A^T). \tag{3.8}$$

Przyjmijmy teraz, że mamy dodatnio określoną symetryczną $n \times n$ macierz Σ oraz kolumnę $\boldsymbol{\mu} \in \mathbb{R}^n$. Korzystając z (3.8), dla zmiennej losowej X mającej rozkład $N(\mathbf{0}, I_n)$ można zapisać

$$Y = AX + \boldsymbol{\mu} \sim N(\boldsymbol{\mu}, \Sigma),$$

o ile tylko

$$AA^T = \Sigma. \tag{3.9}$$

Macierz A spełniającą równość (3.9) można uzyskać na przykład dokonując rozkładu Choleskiego [10], co zostanie przedstawione poniżej. Natomiast generowanie pojedynczej próbki z rozkładu wielowymiarowego $N(\mathbf{0}, I_n)$, to po prostu wygenerowanie próbki n niezależnych zmiennych o rozkładzie $N(0, 1)$, co już zostało objaśnione w poprzednich podrozdziałach.

Równość (3.9) chcemy zapewnić dla dolnie trójkątnej[2] macierzy A, zatem

$$\begin{aligned} A_{11}^2 &= \Sigma_{11}, \\ A_{21}A_{11} &= \Sigma_{21}, \\ A_{21}^2 + A_{22}^2 &= \Sigma_{22} \\ &\vdots \end{aligned}$$

lub ogólnie

$$\sum_{k=1}^{j} A_{ik}A_{jk} = \Sigma_{ij}, j \leqslant i,$$

stąd mamy metodę wyznaczania A_{ij}:

$$\begin{aligned} A_{ij} &= (\Sigma_{ij} - \sum_{k=1}^{j-1} A_{ik}A_{jk})/A_{jj}, \quad j < i, \\ A_{ii} &= \sqrt{\Sigma_{ii} - \sum_{k=1}^{i-1} A_{ik}^2}, \quad j = i. \end{aligned}$$

Przykład 3.3. *Dysponujemy następującymi realizacjami zmiennych X, Y, Z (każdy wiersz odpowiada jednej realizacji rozkładu trójwymiarowego).*

x	y	z
4.0	2.0	0.60
4.2	2.1	0.59
3.9	2.0	0.58
4.3	2.1	0.62
4.1	2.2	0.63

Zakładamy, że (X, Y, Z) ma rozkład trójwymiarowy normalny, wektor wartości oczekiwanych wyznaczymy jako zwykłą średnią z realizacji, natomiast macierz kowariancji oszacujemy jako kowariancję z próby.

Chcemy wyznaczyć macierz Choleskiego, która pozwoli generować trójwymiarowe próbki z rozkładu $N(\mu, \Sigma)$.

[2] Alternatywnie można wynik rozkładu otrzymać w formie macierzy górnie trójkątnej.

Obliczamy

$$\Sigma = \begin{pmatrix} 2.50 \cdot 10^{-2} & 7.50 \cdot 10^{-3} & 1.75 \cdot 10^{-3} \\ 7.50 \cdot 10^{-3} & 7.00 \cdot 10^{-3} & 1.35 \cdot 10^{-3} \\ 1.75 \cdot 10^{-3} & 1.35 \cdot 10^{-3} & 4.30 \cdot 10^{-4} \end{pmatrix}.$$

Następnie, postępując zgodnie z algorytmem, otrzymujemy

$$A_{11} = \sqrt{2.50 \cdot 10^{-2}} \approx 1.581\,138\,83 \cdot 10^{-1},$$

potem

$$\begin{aligned} A_{21} =& 7.50 \cdot 10^{-3} / A_{11} \approx 4.743\,416\,49 \cdot 10^{-2}, \\ A_{22} =& \sqrt{7.00 \cdot 10^{-3} - A_{21}^2} \approx 6.892\,024\,38 \cdot 10^{-2}, \\ A_{31} =& 1.75 \cdot 10^{-3} / A_{11} \approx 1.106\,797\,18 \cdot 10^{-2}, \\ A_{32} =& (1.35 \cdot 10^{-3} - A_{31} A_{21}) / A_{22} \approx 1.197\,035\,81 \cdot 10^{-2}, \\ A_{33} =& \sqrt{4.30 \cdot 10^{-4} - A_{31}^2 - A_{32}^2} \approx 1.281\,446\,55 \cdot 10^{-2}. \end{aligned}$$

Gdyby jeszcze dodatkowo (zgodnie z treścią zadania) wektor średnich $\boldsymbol{\mu} = [\mu_1, \mu_2, \mu_3]^T$ oszacować jako średnią z kolumn, czyli przyjąć

$$\boldsymbol{\mu} = [4.1, 2.08, 0.604]^T,$$

to uzyskalibyśmy poniższą prostą metodę generowania liczb o zadanym rozkładzie wielowymiarowym.

Jeżeli V_1, V_2, V_3 są niezależnymi zmiennymi o rozkładach $N(0,1)$, to wektor losowy (X, Y, Z), gdzie

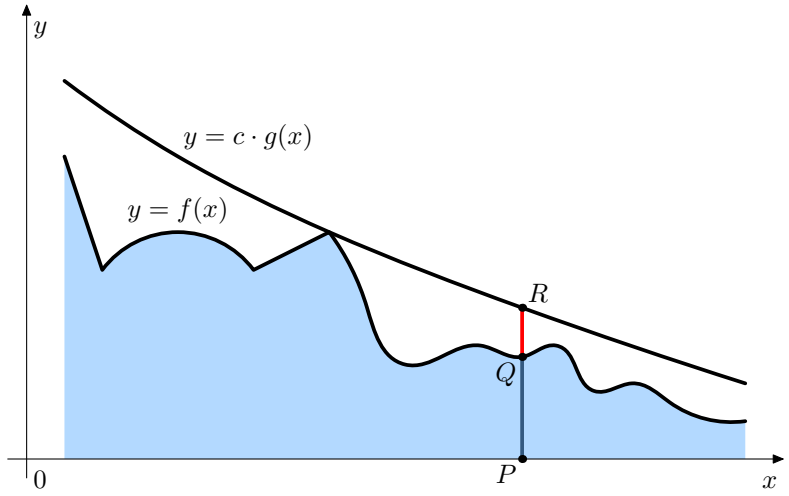
$$\begin{aligned} X &= A_{11} V_1 + \mu_1, \\ Y &= A_{21} V_1 + A_{22} V_2 + \mu_2, \\ Z &= A_{31} V_1 + A_{32} V_2 + A_{33} V_3 + \mu_3, \end{aligned}$$

ma wielowymiarowy rozkład normalny o obliczonej macierzy kowariancji Σ, gdzie A to macierz rozkładu Choleskiego macierzy Σ.

..... *koniec przykładu 3.3* .....

3.3. Metoda akceptacji/odrzucania

Omawiana w tym podrozdziale metoda jest metodą uniwersalną, która pozwala na generowanie próbek z interesującego nas rozkładu, jeśli tylko potrafimy obliczać wartość gęstości tego rozkładu i mamy możliwość generowania próbek innego rozkładu, którego gęstość potrafimy obliczyć i która jest niezerowa wszędzie, gdzie niezerowa jest gęstość interesującego nas rozkładu. Metoda ta może być stosowana do szerokiej gamy rozkładów. Nie ma żadnych przeszkód, by wykorzystywać ją do generowania próbek rozkładów wielowymiarowych.



Rys. 3.2. Metoda akceptacji/odrzucania: linia ograniczająca od góry to przeskalowana gęstość $g(\cdot)$ rozkładu pomocniczego

Ilustracją niech będzie rysunek 3.2. Na rysunku tym linia „górna” jest przeskalowaną czynnikiem c gęstością $g(\cdot)$ rozkładu, który potrafimy generować innym sposobem. Linia „dolna” to gęstość rozkładu $f(\cdot)$, „trudnego”. Jeżeli przy losowaniu z rozkładu o gęstości $g(\cdot)$ trafi nam się wartość odpowiadająca odciętej punktu P, to będziemy ją odrzucać z prawdopodobieństwem odpowiadającym stosunkowi długości odcinków QR i PR. Tym samym będziemy tę wartość akceptować z prawdopodobieństwem odpowiadającym stosunkowi długości odcinków PQ i PR. Wtedy kształt histogramu akceptowanych punktów będzie się – w miarę zwiększania jego dokładności i zwiększania liczby punktów – zbliżał do kształtu pożądanej gęstości.

Te nieformalne rozważania mogą zostać sformułowane w postaci następującego twierdzenia.

Twierdzenie 3.6. *Niech $f(\cdot)$ będzie gęstością zmiennej losowej n-wymiarowej i niech $g(\cdot)$ będzie taką gęstością, że dla pewnej stałej $c > 0$*

$$f(x) \leqslant cg(x), \qquad \text{dla } x \in \mathbb{R}^n. \tag{3.10}$$

Niech X ma rozkład zadany gęstością $g(\cdot)$, niech U ma rozkład jednostajny w $[0,1]$ oraz niech U i X będą niezależne. Wtedy warunkowy rozkład zmiennej X przy warunku

$$U \leqslant \frac{f(X)}{cg(X)}$$

zadany jest gęstością $f(\cdot)$.

Dowód. Niech X będzie zmienną losową o gęstości $g(\cdot)$. Ze względu na warunek (3.10) mamy

$$\Pr\left(U \leqslant \frac{f(X)}{cg(X)}\right) = \frac{f(X)}{cg(X)}.$$

Następnie zauważmy, że

$$\begin{aligned}\Pr\left(U \leqslant \frac{f(X)}{cg(X)}\right) &= \int_{\mathbb{R}^n} \Pr\left(U \leqslant \frac{f(x)}{cg(x)}\right) g(x)\,dx = \\ &= \int_{\mathbb{R}^n} \frac{f(x)}{cg(x)} g(x)\,dx = \int_{\mathbb{R}^n} \frac{1}{c} f(x)\,dx = \frac{1}{c}.\end{aligned} \tag{3.11}$$

Niech A będzie dowolnym borelowskim podzbiorem $\mathbb{R}^n$. Wtedy

$$\begin{aligned}\Pr\left(X \in A | U \leqslant \frac{f(X)}{cg(X)}\right) &= \frac{\Pr\left(X \in A, U \leqslant \frac{f(X)}{cg(X)}\right)}{\Pr\left(U \leqslant \frac{f(X)}{cg(X)}\right)} = \\ &= \frac{\int_A \Pr\left(U \leqslant \frac{f(x)}{cg(x)}\right) g(x)\,dx}{\frac{1}{c}} = \\ &= c\int_A \frac{f(x)}{cg(x)} g(x)\,dx = \int_A f(x)\,dx.\end{aligned}$$

□

Powyższe twierdzenie prowadzi do następującej metody generowania liczb pseudolosowych y o rozkładzie $f(\cdot)$:

1. Wyznacz liczbę pseudolosową x z rozkładu o gęstości $g(\cdot)$.
2. Wyznacz liczbę pseudolosową u z rozkładu jednostajnego na $[0, 1]$.
3. Jeżeli $u < \frac{f(x)}{c \cdot g(x)}$, to $y := x$, w przeciwnym przypadku powróć do kroku 1.

Metoda ta jest jest użyteczna, gdy generowanie liczb z rozkładu $g(\cdot)$ jest istotnie łatwiejsze od generowania liczb z rozkładu $f(\cdot)$.

Zauważmy jeszcze, że jeśli znajdziemy stałą spełniającą warunek (3.10), to każda stała większa będzie także formalnie prawidłowa. Jednak ze względu na równość (3.11) korzystna jest jak najmniejsza wartość c.

Przykład 3.4. *Jeżeli za gęstość $f(\cdot)$ przyjmiemy gęstość rozkładu normalnego $N(0, 1)$, a jako gęstość $g(\cdot)$ – gęstość rozkładu dwustronnego wykładniczego (Laplace'a) ze średnią 0, to optymalnym wyborem okazuje się wybór parametru skali dla rozkładu Laplace'a 1 oraz stałej dla metody akceptacji/odrzucania $c = \sqrt{\frac{2e}{\pi}}$.*

W uzyskanej metodzie prawdopodobieństwo akceptacji wylosowanej liczby z rozkładu Laplace'a jako liczby z rozkładu normalnego wynosi

$$\frac{1}{c} \approx 76\%.$$

3.4. Generowanie liczb pseudolosowych w C++11

3.4.1. Generatory i rozkłady

Nawet gdy pominiemy metody symulacyjne, liczby pseudolosowe odgrywają dużą rolę w programowaniu, będąc częścią wielu ważnych algorytmów (choćby *QuickSort* [1]).

W standardzie języka `C++` przed `C++11` programista dysponował jedynie surowym zestawem dostarczanym przez bibliotekę `cstdlib`:

- stała `RAND_MAX` – maksymalna wartość generatora, o wartości różnej dla różnych kompilatorów (na przykład $2^{31} - 1$, ale można spotkać nawet tak małe wartości jak $2^{15} - 1$),
- `rand()` – funkcja dająca wyniki będące liczbami całkowitymi ze zbioru $\{0, 1, \ldots, \texttt{RAND\_MAX}\}$,
- `srand(int)` – funkcja pozwalająca na ustalenie stanu generatora.

Zestaw ten daje możliwość generowania liczb całkowitych pseudolosowych o rozkładzie równomiernym, stąd można już łatwo wyprowadzić generowanie liczb przybliżających próbki z rozkładu jednostajnego w przedziale. Na tej podstawie można następnie generować inne potrzebne rozkłady.

Wady tego (dość popularnego i wciąż stosowanego) rozwiązania to między innymi brak standaryzacji, wątpliwa jakość w niektórych realizacjach, problemy przy przetwarzaniu równoległym, konieczność dokonywania uzupełnień – na przykład przy generowaniu podstawowych rozkładów – dla wielu typowych sytuacji.

Wymagało to więc albo dodatkowego nakładu pracy, albo korzystania z bibliotek spoza standardu, jak `boost` – (zob. https://www.boost.org/).

Wersja `C++11` (i oczywiście nowsze standardy) dostarcza bibliotekę `random`, która umożliwia nowy, obiektowy sposób generowania liczb pseudolosowych.

W koncepcji tej `generator`, leżący u podstaw procesu pozyskiwania liczb pseudolosowych, jest oddzielony od `rozkładu`, który przy pomocy generatora oblicza kolejne liczby o zadanych własnościach.

Generatory są obiektami, które co do zasady generują liczby całkowite o rozkładzie równomiernym, mają metody `max()` i `min()`, konstruktor, który może przyjmować (między innymi) parametr takiego typu jak wyniki generatora, służący do ustawienia `ziarna` generatora, operator `operator()`, który sprawia, że generator staje się obiektem funkcyjnym (może być wywoływany jak funkcja), także metodę `seed` pozwalającą na zmianę ziarna.

Jest zdefiniowany cały szereg takich generatorów, na przykład: `default_random_engine`, `ranlux48`, `mt19937_64`, `knuth_b`.

Obiekty tego rodzaju mogą być zapisywane i odczytywane ze strumieni za pomocą operatorów `<<`, `>>`.

`Rozkład` z kolei jest obiektem z odpowiedniej klasy, którego konstruktor ma zestaw parametrów charakterystycznych dla konkretnej klasy (na przykład w przypadku rozkładu wykładniczego jest to parametr częstotliwości, w przypadku rozkładu jednostajnego – oba końce przedziału, dla rozkładu normalnego – średnia i odchylenie standardowe itd.). `Rozkład` ma zdefiniowany operator `operator()`, przyjmujący jako parametr `generator` i dający w wyniku liczbę z właściwego rozkładu.

Przykłady rozkładów:

- `uniform_int_distribution`,
- `uniform_real_distribution`,
- `poisson_distribution`,
- `gamma_distribution`,
- `exponential_distribution`

i wiele innych.

Obiekty tego rodzaju – podobnie jak generatory – mogą być zapisywane i odczytywane ze strumieni za pomocą operatorów `<<`, `>>`.

Przykład 3.5. *Zasymulowanie funkcji* `rand()` *oraz* `srand()` *może wyglądać następująco:*

```
#include<random>
#include<cstdlib>
std::mt19937_64 __mg;
void srand(int s = 1) {
     __mg.seed(s);
}
int rand() {
     static std::uniform_int_distribution<int> uni(0,RAND_MAX);
     return uni(__mg);
}
```

jednak w nowo tworzonych programach lepiej oczywiście unikać `rand()` *oraz* `RAND_MAX`.

3.4.2. Liczby prawdziwie losowe

Istnieje też szczególna klasa generatorów – `random_device` – której obiekty mają generować wyniki niepowtarzalne i nieprzewidywalne. Nie każdy kompilator potrafi ją prawidłowo zrealizować, zatem użycie jej powinno być ograniczone na przykład do jednokrotnego zainicjowania innego generatora – już zwykłego, pseudolosowego. Nawet i to może nie dać oczekiwanych rezultatów, bo zdarzają się kompilatory z deterministycznym działaniem `random_device` – na przykład tak jest w przypadku kompilatora `MinGW-W64` dla Windows, w wersji 8.1.0 – jednego z zalecanych do użycia z aktualną (w chwili publikacji skryptu) wersją 20.03 popularnego środowiska `Code::blocks`.

Przy założeniu poprawnego działania klasy `random_device`, w miejsce starego

```
srand(time(0));
```

można w tym kontekście użyć na przykład

```
moj_generator.seed(random_device{}());
```

lub wprost przy konstrukcji np.

```
mt19937_64 moj_generator(random_device{}());
```

3.4.3. Dystrybuanta rozkładu normalnego i jej odwrotność w C++

Biblioteka `<cmath>` dostarcza funkcję błędu `erf(double)` (zwaną także funkcją błędu Gaussa) taką, że

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\, dt.$$

Na tej podstawie można wyprowadzić, że

$$\Phi(x) = \frac{\operatorname{erf}\left(\frac{x}{\sqrt{2}}\right) + 1}{2}, \tag{3.12}$$

gdzie $\Phi(\cdot)$ jest dystrybuantą rozkładu normalnego $N(0,1)$.

Jej odwrotności niestety w standardzie aktualnym nie ma, jest w bibliotekach typu `boost`. Do jej odwrócenia można użyć ogólnych metod rozwiązywania równań nieliniowych (jak bisekcja lub metoda stycznych), są także dla tego szczególnie ważnego przypadku opracowane efektywne metody wykorzystujące jego specyfikę.

By uniknąć konieczności korzystania z dodatkowych bibliotek (jak `boost`, co nie zawsze jest wygodne), dla dalszych rozważań i obliczeń zostanie w skrypcie zaproponowany prosty w użyciu niewielki plik `qnorm.cpp` – listing 3.2. Plik ten może zostać dołączony do projektu

wieloplikowego w C++ jako niezależnie kompilowany plik źródłowy. Nowo definiowane nazwy umieszczone są w przestrzeni nazw mc, dla skojarzenia z tematyką skryptu.

Plik ten wprowadza dwie użyteczne funkcje C++:

- `double pnorm(double)` – dla danego parametru oblicza wartość dystrybuanty rozkładu $N(0,1)$, jest zbudowana przy wykorzystaniu erf zgodnie z formułą (3.12),
- `double qnorm(double)` – dla danego parametru oblicza wartość funkcji kwantylowej rozkładu $N(0,1)$, oparta jest na koncepcji zrealizowanej w programie *Matlab* funkcji `erfinv`, kod funkcji `NewtonRefinementInverseError` z listingu 3.2 jest kopią kodu funkcji ze strony `http://www.mimirgames.com/articles/programming/approximations-of-the-inverse-error-function/`.

Dla ułatwienia nagłówki tych dwu funkcji zostały umieszczone w pliku `qnorm.h`, którego treść zaprezentowana jest na listingu 3.3.

Przykładowy program korzystający z tych funkcji, a właściwie testujący ich działanie dla wybranych (w tym często używanych w statystyce) kwantyli, umieszczony jest na listingu 3.4. Oto wynik jego działania:

```
qnorm(0.005) = -2.57583, pnorm(qnorm(0.005)) = 0.005
qnorm(0.01) = -2.32635, pnorm(qnorm(0.01)) = 0.01
qnorm(0.05) = -1.64485, pnorm(qnorm(0.05)) = 0.05
qnorm(0.1) = -1.28155, pnorm(qnorm(0.1)) = 0.1
qnorm(0.5) = 0, pnorm(qnorm(0.5)) = 0.5
qnorm(0.9) = 1.28155, pnorm(qnorm(0.9)) = 0.9
qnorm(0.95) = 1.64485, pnorm(qnorm(0.95)) = 0.95
qnorm(0.99) = 2.32635, pnorm(qnorm(0.99)) = 0.99
qnorm(0.995) = 2.57583, pnorm(qnorm(0.995)) = 0.995
```

Wynik ten warto porównać ze znanymi wartościami z tabel kwantyli rozkładu normalnego bądź z wartościami uzyskiwanymi w programach takich jak *R* czy *Octave*.

Gdybyśmy jednak w danym projekcie decydowali się na użycie biblioteki `boost`, to z plików `qnorm.h` i `qnorm.cpp` możemy zrezygnować i posłużyć się kodem z listingu 3.5. Pokazuje on, jak w takiej sytuacji można prosto zdefiniować funkcje `pnorm(double)` oraz `qnorm(double)`, które obliczają dystrybuantę oraz funkcje kwantylową rozkładu $N(0,1)$.

```cpp
#include <cmath>
// funkcje sign, NewtonRefinementInverseError skopiowane z
// http://www.mimirgames.com/articles/programming
//         /approximations-of-the-inverse-error-function/
namespace mc {
    double sign(double x){ return (x > 0) - (x < 0); }
    double NewtonRefinementInverseError(const double x){
        static const double pi = 4.0 * atan(1.0);
        double r;
        double a[] =
          {0.886226899,-1.645349621,0.914624893,-0.140543331};
        double b[] =
          {1,-2.118377725,1.442710462,-0.329097515,0.012229801};
        double c[] =
          {-1.970840454,-1.62490649,3.429567803,1.641345311};
        double d[] = {1, 3.543889200, 1.637067800};
        double z = sign(x) * x;
        if(z <= 0.7){
            double x2=z*z;
            r=z*(((a[3]*x2+a[2])*x2+a[1])*x2+a[0]);
            r/=(((b[4]*x2+b[3])*x2+b[2])*x2+b[1])*x2+b[0];
         } else {
            double y=sqrt(-log((1-z)/2));
            r=(((c[3]*y+c[2])*y+c[1])*y+c[0]);
            r/=((d[2]*y+d[1])*y+d[0]);
         }
         r=r*sign(x);
         z=z*sign(x);
         r-=(erf(r)-z)/(2/sqrt(pi)*exp(-r*r));
         r-=(erf(r)-z)/(2/sqrt(pi)*exp(-r*r));
         return r;
    }
    auto inv_erf = NewtonRefinementInverseError;
    const double sq2 = sqrt(2.0);
    double qnorm(double x) {
        return mc::sq2*mc::inv_erf(2.0*x-1.0);
    };
    double pnorm (double x) {
        return (erf(x/mc::sq2) + 1.0)/2.0;
    }
}
```

Listing 3.2. Zawartość pliku `qnorm.cpp`

```
#ifndef __QNORM__H__
#define __QNORM__H__
namespace mc {
    double qnorm(double);
    double pnorm(double);
}
#endif // __QNORM__H__
```

Listing 3.3. Zawartość pliku `qnorm.h`

```
#include "qnorm.h"
#include <iostream>
int main() {
    for (auto p: {0.005, 0.01, 0.05, 0.1, 0.5,
                  0.9, 0.95, 0.99, 0.995}) {
        auto q = mc::qnorm(p);
        auto xp = mc::pnorm(q);
        std::cout << "qnorm(" << p << ") = " <<
            q << ", pnorm(qnorm(" << p << ")) = " <<
            xp << "\n";
    }
}
```

Listing 3.4. Plik główny programu dokonującego prostego testowania funkcji `pnorm` oraz `qnorm`

Wyniki, jakie dostaniemy programami z listingów 3.4 oraz 3.5, są dokładnie takie same.

Zadania

Zadanie 3.1. Zaimplementuj własną metodę w `C++` wyznaczania próbek rozkładu dwumianowego (bazującą na standardowej metodzie wyznaczającej próbki rozkładu jednostajnego):
(a) opartą na sumowaniu próbek zero-jedynkowych,
(b) opartą na funkcji kwantylowej.
Porównaj ich efektywność dla różnych wartości parametrów rozkładu dwumianowego.

```cpp
#include "boost/math/distributions.hpp"
#include <iostream>

double qnorm(double p) {
        static boost::math::normal_distribution<double> mn(0,1);
        return quantile(mn, p);
}
double pnorm(double q) {
        static boost::math::normal_distribution<double> mn(0,1);
        return cdf(mn, q);
}

int main() {
    for (auto p: {0.005, 0.01, 0.05, 0.1, 0.5,
                  0.9, 0.95, 0.99, 0.995}) {
        auto q = qnorm(p);
        auto xp = pnorm(q);
        std::cout << "qnorm(" << p << ") = " <<
            q << ", pnorm(qnorm(" << p << ")) = " <<
            xp << "\n";
    }
}
```

Listing 3.5. Plik główny programu dokonującego prostego testowania funkcji `pnorm` oraz `qnorm`, zdefiniowanych przy wykorzystaniu `boost`

Zadanie 3.2. Opracuj (wykorzystując funkcję kwantylową) i zaimplementuj w C++ metodę generowania pseudolosowych próbek rozkładu geometrycznego z parametrem $p \in (0, 1)$.

Zadanie 3.3. Niech

$$F(x) = \begin{cases} 0, & \text{dla } x < 0, \\ \frac{1}{10}x^2, & \text{dla } x \in [0, 2), \\ \frac{1}{4}x, & \text{dla } x \in [2, 3), \\ \frac{9}{10}, & \text{dla } x \in [3, 4), \\ 1, & \text{dla } x \in [4, \infty) \end{cases}$$

będzie dystrybuantą pewnej zmiennej losowej (typu mieszanego).

(a) Wyznacz (korzystając z funkcji kwantylowej) funkcję h taką, by zmienna losowa $Y = h(X)$ miała dystr. F, gdzie X jest zmienną o rozkładzie jednostajnym w $[0, 1]$.
(b) Oblicz EY oraz $\operatorname{Var} Y$.
(c) Metodą Monte Carlo – dla $10\,000\,000$ powtórzeń – zweryfikuj poprawność generowania, porównując średnią i wariancję z próby z wartościami obliczonymi w (b).
(d) Oszacuj – metodą Monte Carlo dla $100\,000\,000$ prób – wartość

$$\Pr(A + B + C + D + E + F \in (7, 14]),$$

gdzie A, B, C, D, E, F są niezależnymi zmiennymi losowymi o jednakowym rozkładzie danym dystrybuantą F.

Zadanie 3.4. Przetestuj funkcję z listingu 3.1 na stronie 46, na podstawie $10\,000\,000$ wygenerowanych liczb z rozkładu $N(0, 1)$, szacując momenty zwykłe do rzędu 6 włącznie i porównując je z wartościami teoretycznymi dla rozkładu $N(0, 1)$.

Zadanie 3.5. Korzystając z metody Marsaglia–Braya, opracuj funkcję w języku `C++` obliczającą liczbę losową z rozkładu $N(0, 1)$.

Do tworzenia próbek z rozkładu jednostajnego użyj obiektu klasy `uniform_real_distribution<double>`. Generatorem niech będzie obiekt klasy `mt19937_64` inicjowany domyślnie.

(a) Wygeneruj próbkę dla $n = 50$, zapisz wyniki w pliku tekstowym. Oblicz (na przykład w `R`) *p-value* testu Kołmogorowa–Smirnowa dla hipotezy H_0, według której próbka pochodzi z rozkładu normalnego.
(b) Wygeneruj 10^4 takich próbek (jedna próbka będzie zapisana w jednym wierszu). Sprawdź, jaki procent próbek ma *p-value* testu Kołmogorowa–Smirnowa poniżej 0.05.

Zadanie 3.6. Wróć do przykładu 3.4 na stronie 54 i uzasadnij wybór parametru skali dla rozkładu Laplace'a oraz wybór stałej c wskazanej w przykładzie. Następnie zaimplementuj funkcję generującą w `C++` i dokonaj testowania, podobnie jak w zadaniu 3.4.

Zadanie 3.1. W przypadku (b), skoro funkcja kwantylowa jest przedziałami stała, to dla dużego n na pewno warto zoptymalizować wyszukiwanie pasującego przedziału, na przykład jako wyszukiwanie binarne.

Zadanie 3.2. W rozkładzie tym

$$\Pr(X = k) = (1-p)^{k-1}p, \quad \text{dla } k \in \{1, 2, 3 \ldots\}.$$

Zadanie 3.3. (b) Wyniki: $EY = \frac{53}{24}$, $\operatorname{Var} Y = \frac{2467}{2880}$, (d) około 0.63.

Zadanie 3.4. Dla rozkładu $N(0,1)$ momenty rzędu $k \in \{1, 2, 3, \ldots\}$ dane są formułą

$$M_k = \begin{cases} 0 & \text{dla } k \text{ nieparzystego,} \\ (k-1)!! & \text{dla } k \text{ parzystego.} \end{cases}$$

Zadanie 3.5. Odpowiedzi: (a) $0.319\,831$, (b) 4.76%.

ROZDZIAŁ 4

Dokładność pomiaru i redukcja wariancji

Rozdział ten poświęcony jest zagadnieniom związanym z precyzją oszacowań uzyskiwanych metodą Monte Carlo i połączonym z tym problemem redukowania wariancji. Podstawowa koncepcja leżąca u podstaw badania dokładności uzyskiwanych oszacowań to oczywiście estymacja przedziałowa. Jej wybrane elementy zostaną omówione w następującym podrozdziale.

4.1. Estymacja przedziałowa

Istnieje wiele modeli – zależnych od założeń wobec rozkładu, od liczebności próby i szacowanego parametru rozkładu – estymacji przedziałowej. Niektóre z nich w metodzie Monte Carlo są mniej istotne, bo – na przykład – na ogół możemy przyjąć, że próba, jaką utworzymy metodą generowania liczb pseudolosowych, jest duża, zatem modele dotyczące prób o małej liczności możemy pominąć.

W podstawowej metodzie Monte Carlo szacujemy wartość oczekiwaną, zatem ważniejsze są modele dla wartości średniej niż modele dla wariancji czy kwantyli. W krótkim przypomnieniu skupię się na dwu modelach: przedział ufności dla średniej oraz przedział ufności dla frakcji wyróżnionych elementów – oba modele zostaną tu rozważone dla przypadku licznej próby.

4.1.1. Przedział ufności dla średniej

Rozważamy model, w którym mamy niezależne zmienne losowe

$$X_1, X_2, X_3, \ldots, X_n$$

o dowolnym, jednakowym rozkładzie, który ma skończoną wartość oczekiwaną μ oraz skończoną wariancję, $n \geqslant 100$ [4].

Niech

$$\overline{X} = \frac{1}{n} \sum_{i=1}^{n} X_i$$

będzie średnią próby, zaś

$$\hat{S}^2 = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \overline{X})^2$$

niech będzie wariancją próby. Niech $\alpha \in (0,1)$. Przedziałem ufności dla wartości μ na poziomie ufności $1-\alpha$ jest przedział postaci

$$\left(\overline{X} - \frac{\hat{S}}{\sqrt{n}} z_\alpha, \overline{X} + \frac{\hat{S}}{\sqrt{n}} z_\alpha \right), \tag{4.1}$$

gdzie z_α jest kwantylem rzędu $1 - \frac{\alpha}{2}$ rozkładu $N(0,1)$.

Interpretacja przedziału ufności jest taka, że stosując konkretną procedurę uzyskiwania przedziału ufności dla danego n (dla różnych realizacji zmiennych losowych), frakcja osiąganych sukcesów – gdzie sukcesem jest to, że nieznana wartość μ znajdzie się w wyznaczonym przedziale – będzie na poziomie ufności $1-\alpha$. Rozważany model jest oczywiście przybliżony ze względu na to, że dopuszcza się dowolne (byle o skończonej wariancji) rozkłady, formuła (4.1) wyprowadzana jest w oparciu o Centralne Twierdzenie Graniczne [11].

Jeżeli więc za α przyjmiemy małą liczbę rzeczywistą – na przykład 0.05 lub 0.01, to można z dużą pewnością założyć, że szacowana wartość znajduje się w wyznaczonym przedziale. Innymi słowy można środek przedziału (czyli wartość średnią z próby) uznać za oszacowanie szukanej wartości, natomiast połowę długości przedziału ufności – jako oszacowanie wartości bezwzględnej popełnianego błędu.

Zatem oszacowanie wartości bezwzględnej błędu to

$$\frac{\hat{S}}{\sqrt{n}} z_\alpha, \tag{4.2}$$

czyli wielkość proporcjonalna do odchylenia standardowego z próby, odwrotnie proporcjonalna do pierwiastka kwadratowego z liczebności próby i proporcjonalna do kwantyla rzędu $1-\frac{\alpha}{2}$ rozkładu normalnego.

Kwantyl rzędu $1-\frac{0.05}{2}$ rozkładu $N(0,1)$ wynosi około $1.959\,964$, natomiast kwantyl rzędu $1-\frac{0.01}{2}$ tego rozkładu to około $2.575\,829$. Stąd czasem przyjmuje się uproszczony wzór, w którym zamiast mnożnika z_α użyta jest wartość 2 w formule (4.2). Na przykład w podręczniku *Metody Monte Carlo* [32] wartość

$$2\frac{\hat{S}}{\sqrt{n}}$$

nazywana jest błędem metody Monte Carlo.

Przykład 4.1. *Rozwiązując zadanie 1.1(a) uśredniamy* 1000 *realizacji wyrażenia* $2\exp(-X^2)$*, gdzie* X *ma rozkład jednostajny w* $[0,2]$*. Uzyskany wynik (zakładam* `C++`*, generator* `mt19937_64`*) to*

$$\overline{x} = 0.879\,172, \qquad \hat{s} \approx 0.695\,968.$$

Stąd, przyjmując poziom ufności $1-\alpha = 95\%$*, dostajemy przedział ufności* $(0.836\,037, 0.922\,308)$ *i błąd oszacowania* $0.043\,135\,7$*. Tymczasem wartość wyznaczona dokładniejszą metodą, to*

$$\frac{\sqrt{\pi}\,\mathrm{erf}(2)}{2} \approx 0.882\,081.$$

Zatem dokładna wartość ewidentnie znajduje się w wyznaczonej realizacji przedziału ufności.

To, że domyślna inicjalizacja generatora dała tak trafny wynik, to oczywiście przypadek, ale przypadek, co do którego zajścia mieliśmy 95% *pewności. Dalsze eksperymenty pokazują, że przy* $100\,000$ *tego rodzaju prób w* $95\,050$ *przypadkach dokładna wartość znajduje się w szacowanym przedziale. Jest to* 95.05% *przypadków, co świadczy o poprawności modelu.*

4.1.2. Przedział ufności dla frakcji

Jeżeli dysponujemy liczbą sukcesów w n próbach, to istotne jest dla nas pytanie o prawdopodobieństwo sukcesu w pojedynczej próbie. Jest to też wartość oczekiwana zmiennej dającej wartość 1 w przypadku

sukcesu i wartość 0 przypadku porażki. Wartość ta jest także nazywana `frakcją` wyróżnionych elementów.

Estymator

$$\frac{S_n}{n},$$

gdzie S_n jest liczbą sukcesów w ciągu n prób, jest zgodnym, nieobciążonym i efektywnym estymatorem frakcji.

Jeżeli chodzi o przedziały ufności dla frakcji, to proponowanych jest wiele rozwiązań. Opiszmy kilka z nich. Załóżmy, że S_n to liczba sukcesów w n próbach, jako $\widehat{p}$ oznaczmy stosunek liczby sukcesów do liczby prób, natomiast p to nieznana frakcja.

Metoda asymptotyczna

Metoda ta jest stosowana przy $n \geqslant 100$ [11], co na ogół zakładamy w metodzie Monte Carlo. Jest to model bazujący na przybliżaniu sumy rozkładów zero-jedynkowych rozkładem normalnym i z tego powodu może nie dawać właściwych rezultatów, gdy p jest bliskie zera lub jedynki (bo wtedy to przybliżenie jest dyskusyjne). Formuła przedziału ufności:

$$\left(\widehat{p} - \sqrt{\frac{\widehat{p}(1-\widehat{p})}{n}} z_\alpha, \widehat{p} + \sqrt{\frac{\widehat{p}(1-\widehat{p})}{n}} z_\alpha\right). \tag{4.3}$$

Wyniki uzyskane tą metodą są zbliżone do wyników, które uzyskamy, stosując dla zmiennych zero-jedynkowych formułę (4.1).

Metoda Wilsona

Metoda ta również jest oparta na przybliżaniu rozkładem normalnym, jednak etap wyprowadzenia formuły jest bardziej ścisły niż w przypadku poprzedniej metody, według testów wyniki są korzystniejsze [30]. Przedział ufności opisany jest formułą

$$\Big(A(B-C), A(B+C)\Big), \tag{4.4}$$

gdzie

$$A = \frac{n}{n+z_\alpha^2}, \quad B = \widehat{p} + \frac{z_\alpha^2}{2n}, \quad C = \frac{z_\alpha}{n}\sqrt{n\widehat{p}(1-\widehat{p}) + \frac{z_\alpha^2}{4}}.$$

Metoda ta jest często stosowana i łatwa w użyciu. Warto zauważyć, że w odróżnieniu od poprzednio podanych formuł (4.1) oraz (4.3) środek przedziału ufności różni się od średniej z próby.

W niektórych pracach [4] podaje się dla tej metody założenie, że n powinno wynosić co najmniej 100, jednak okazuje się [30], że metoda na ogół działa bardzo dobrze także dla mniejszych wartości n, co i tak w przypadku metody Monte Carlo ma niewielkie znaczenie.

Przedziały oparte o rozkład *beta*

W metodzie tej granice przedziałów wyznaczone są jako kwantyle rozkładu *beta*.

Rozkład *beta* z parametrami a, b to rozkład, w którym gęstość jest dodatnia w przedziale $(0, 1)$ i jest w tym przedziale proporcjonalna do $x^{a-1}(1-x)^{b-1}$. Oznaczmy przez $B_{a,b}(\cdot)$ dystrybuantę tego rozkładu. Wtedy przedział ufności dla frakcji może być wyznaczony jako [31]

$$(B^{-1}_{S_n, n-S_n+1}(\alpha/2), B^{-1}_{S_n, n-S_n+1}(1-\alpha/2)). \tag{4.5}$$

Nieco zmieniony wzór, według badań [6] dający lepsze praktyczne wyniki to

$$(B^{-1}_{S_n+0.5, n-S_n+0.5}(\alpha/2), B^{-1}_{S_n+0.5, n-S_n+0.5}(1-\alpha/2)). \tag{4.6}$$

W tym przypadku, jeśli $S_n = 0$, to jako dolną granicę można przyjąć 0, natomiast jeśli $S_n = n$, to jako górną granicę można przyjąć 1 – w obu sytuacjach tę drugą granicę należy obliczyć według formuły (4.6).

Na listingu 4.1 umieszczono przykładową funkcję w `C++` obliczającą, przy wykorzystaniu wspomnianej biblioteki `boost`, końce przedziałów według wzorów (4.6).

Przykład 4.2. *W przykładzie 1.1 na stronie 13 szacowanie liczby π odbywało się na podstawie frakcji punktów trafiających w koło, przemnożonej przez 4. Tabela 4.1 pokazuje, jak poszczególne metody omawiane w tym podrozdziale szacują przedział dla poszukiwanej frakcji.*

Zauważmy, że dla rzeczywiście dużego n wszystkie metody dają takie same wyniki. Zatem jeśli frakcja nie jest bardzo bliska zeru lub jedynce (co można zauważyć podczas wstępnej oceny), to usprawiedliwione jest stosowanie prostego modelu, takiego jak (4.3), do szacowania przedziału ufności bądź szacowania liczby n potrzebnej do uzyskania żądanej dokładności.

```cpp
#include <boost/math/distributions.hpp>
#include <tuple>
using boost::math::beta_distribution;
double qbeta(double p, double a, double b) {
    // oblicz wartość w punkcie *p* funkcji kwantylowej
    // rozkładu beta z parametrami *a*, *b*
    return quantile(beta_distribution<double>(a,b), p);
}
std::pair<double, double> binom_interv_beta(int k, int n,
        double alpha) {
    // przedział ufności dla frakcji przy *k* sukcesach
    // w *n* próbach
    auto interv=[k,n](double p) {
        return qbeta( p, k+0.5, n-k+0.5 );
    };
    double lewy  = (k==0)? 0 : interv(alpha/2);
    double prawy = (k==n)? 1 : interv(1-alpha/2);
    return {lewy, prawy};
}
```

Listing 4.1. Obliczenie przedziału ufności według formuły (4.6) w C++

Tab. 4.1. Przedziały ufności omawiane w przykładzie 4.2

S_n	n	formuła	przedział	przedz. $\times 4$
76	100	(4.3)	$(0.676293, 0.843707)$	$(2.70517, 3.37483)$
76	100	(4.4)	$(0.667677, 0.833087)$	$(2.67071, 3.33235)$
76	100	(4.6)	$(0.669695, 0.835505)$	$(2.67878, 3.34202)$
76	100	(4.5)	$(0.664265, 0.831220)$	$(2.65706, 3.32488)$
76	100	(4.1)	$(0.675872, 0.844128)$	$(2.70349, 3.37651)$
7854766	10000000	(4.3)	$(0.785222, 0.785731)$	$(3.14089, 3.14292)$
7854766	10000000	(4.4)	$(0.785222, 0.785731)$	$(3.14089, 3.14292)$
7854766	10000000	(4.6)	$(0.785222, 0.785731)$	$(3.14089, 3.14292)$
7854766	10000000	(4.5)	$(0.785222, 0.785731)$	$(3.14089, 3.14292)$
7854766	10000000	(4.1)	$(0.785222, 0.785731)$	$(3.14089, 3.14292)$

Omawiane modele oceniania przedziałów ufności – zwłaszcza model (4.1), ale w połączeniu z uwagami w przykładzie 4.2 dotyczy to także przedziałów dla frakcji – wskazują na istotność wariancji zmiennych losowych, których wartość oczekiwaną próbujemy szacować metodą Monte Carlo. Metodami redukcji wariancji zajmiemy się w kolejnych podrozdziałach.

4.2. Zmienne antytetyczne

Zmienne antytetyczne (ang. *antithetic*) to zmienne o tym samym rozkładzie, ujemnie skorelowane [32]. Skoro zmienne te mają ten sam rozkład, to mają także tę samą wartość oczekiwaną. Natomiast ujemna korelacja powoduje, że ich odchyłki od średniej mają przeciętnie przeciwny kierunek, stąd korzystne zmniejszenie wariancji sumy. Te intuicyjne uwagi zostaną sformułowane jako poniższe twierdzenie.

Twierdzenie 4.1. *Niech* $(Y_1, \tilde{Y}_1), (Y_2, \tilde{Y}_2), \ldots$ *będzie ciągiem niezależnych zmiennych losowych (dwuwymiarowych) o jednakowym rozkładzie, przy tym wszystkie* $Y_i, \tilde{Y}_i$, $i = 1, 2, 3 \ldots$ *mają ten sam rozkład o skończonej wariancji. Wariancja zmiennej*

$$\frac{1}{2n}(Y_1 + \tilde{Y}_1 + \ldots Y_n + \tilde{Y}_n)$$

jest mniejsza od wariancji

$$\frac{1}{2n}(Y_1 + Y_2 + \ldots + Y_{2n})$$

wtedy i tylko wtedy, gdy Y_1 *i* $\tilde{Y}_1$ *są ujemnie skorelowane.*

Dowód. Wariancja sumy zmiennych niezależnych jest równa sumie ich wariancji, stąd

$$\operatorname{Var}\left(\frac{1}{2n}(Y_1 + \tilde{Y}_1 + \ldots Y_n + \tilde{Y}_n)\right) = \frac{1}{(2n)^2} n \operatorname{Var}(Y_1 + \tilde{Y}_1) =$$
$$= \frac{2\operatorname{Var} Y_1 + 2\operatorname{Cov}(Y_1, \tilde{Y}_1)}{4n} = \frac{\operatorname{Var} Y_1 + \varrho(Y_1, \tilde{Y}_1)\operatorname{Var} Y_1}{2n},$$

gdzie $\varrho(Y_1, \tilde{Y}_1)$ jest współczynnikiem korelacji między Y_1 i $\tilde{Y}_1$.

Natomiast

$$\operatorname{Var}\left(\frac{1}{2n}(Y_1+Y_2+\ldots+Y_{2n})\right)=\frac{1}{(2n)^2}2n\operatorname{Var}Y_1=\frac{\operatorname{Var}Y_1}{2n}.$$

Porównanie liczników obu uzyskanych ułamków kończy dowód. □

W praktyce opisana wyżej własność może być stosowana w następujący sposób. Otóż często u podstaw obliczeń Monte Carlo leży jakieś przekształcenie zmiennej o rozkładzie jednostajnym na $[0,1]$. Widać to było w wielu przykładach, nawet generowanie próbek z różnych rozkładów może być sprowadzone do przekształcania zmiennych o rozkładzie jednostajnym (choćby w metodzie odwracania dystrybuanty). Powiedzmy, że $Y_i=f(X_i)$, gdzie X_i ma rozkład jednostajny na $[0,1]$. Wtedy można przyjąć np. $\tilde{Y}_i=f(1-X_i)$. A gdyby na przykład X_i miał rozkład $N(0,1)$, to za $\tilde{Y}_i$ można przyjąć $f(-X_i)$ itd.

W przypadku gdy $Y_i=f(X_i)$, gdzie X_i ma rozkład jednostajny na $[0,1]$, przyjęcie $\tilde{Y}_i=f(1-X_i)$ da najlepszy efekt wtedy, gdy przekształcenie $f(\cdot)$ jest bliskie liniowemu.

Definicja 4.1. Współczynnikiem redukcji wariancji (*variance reduction ratio*) nazywamy liczbę $\mathrm{v.r.r.}$ równą stosunkowi wariancji estymatora standardowego do wariancji estymatora zmodyfikowanego opartego na próbce tej samej wielkości.

Fakt 4.1. *Przy użyciu poprzednich oznaczeń, dla metody zmiennych antytetycznych zachodzi*

$$\mathrm{v.r.r.}=\frac{1}{1+\varrho(Y_1,\tilde{Y}_1)}. \tag{4.7}$$

Uzasadnienie. Porównujemy wariancje oparte na próbkach tej samej liczności, zatem dzielimy przez siebie oba ułamki uzyskane w dowodzie twierdzenia 4.1, stąd uzyskujemy wartość ze wzoru (4.7). □

Przykład 4.3. *Rozważmy problem obliczenia całki*

$$I=\int_0^\infty x^{0.6}e^{-3x}\,dx$$

metodą Monte Carlo.

Dokładny wynik przy użyciu funkcji *gamma Eulera* (Γ) to

$$\frac{\Gamma\left(\frac{3}{5}\right)}{5 \cdot 3^{\frac{3}{5}}} \approx 0.154\,066\,4.$$

Za względu na równość

$$\int_0^\infty x^{0.6} e^{-3x}\, dx = \int_0^\infty \left(\frac{1}{3} x^{0.6}\right) 3e^{-3x}\, dx$$

oraz na to, że $3e^{-3x}$ dla $x > 0$ jest gęstością rozkładu wykładniczego z parametrem częstotliwości 3, możemy spojrzeć na szukaną wartość jak na

$$I = E\left(\frac{1}{3} \cdot V^{0.6}\right),$$

gdzie V ma rozkład wykładniczy z parametrem częstotliwości 3 (czyli wykładniczy z wartością oczekiwaną $1/3$). V może być też – przy wykorzystaniu metody opisanej w przykładzie 3.2 na stronie 40 – przedstawiony jako

$$-1/3 \ln(1 - X),$$

gdzie X ma rozkład jednostajny w $[0, 1]$. Zatem

$$I = E\left(-\frac{1}{3}\ln(1 - X)\right)^{0.6}, \quad \text{gdzie } X \text{ ma rozkład jednostajny w } [0, 1].$$

Niech

$$Y_i = \frac{1}{3}\left(-\frac{1}{3}\ln(1 - X_i)\right)^{0.6}$$

oraz niech

$$\tilde{Y}_i = \frac{1}{3}\left(-\frac{1}{3}\ln X_i\right)^{0.6}$$

dla $i = 1, 2, 3, \ldots$, przy czym X_i mają rozkład jednostajny w $[0, 1]$. Wtedy $\varrho(Y_1, \tilde{Y}_1) \approx -0.898\,135\,12$ i stąd dosyć duży

$$\text{v. r. r.} \approx 9.816\,926.$$

Ze względu na wzór (4.1), który znajduje się na stronie 66, oznacza to, że dla dużego n spodziewamy się ponad trzykrotnie węższych

przedziałów ufności, gdy zamiast uśredniania

$$Y_1, Y_2, \dots, Y_{2n}$$

uśrednimy

$$Y_1, \tilde{Y}_1, Y_2, \tilde{Y}_2, \dots, Y_n, \tilde{Y}_n$$

lub – co na jedno wychodzi, ale jest wygodne, bo pozwala na stosowanie na przykład właśnie wzoru (4.1) – uśrednimy

$$\frac{Y_1 + \tilde{Y}_1}{2}, \frac{Y_2 + \tilde{Y}_2}{2}, \dots, \frac{Y_n + \tilde{Y}_n}{2}.$$

Czas na eksperyment w `C++`. Do generowania liczb o rozkładzie jednostajnym użyto `uniform_real_distribution<double>` oraz generatora `mt19937_64` i przyjęto poziom ufności $1 - \alpha = 0.99$.

Średnia z $100\,000\,000$ realizacji zmiennych Y_i dała przedział ufności dla szukanej wielkości

$$(0.154\,029, 0.154\,078),$$

natomiast średnia z $50\,000\,000$ realizacji $(Y_i + \tilde{Y}_i)/2$ dała przedział

$$(0.154\,059, 0.154\,075).$$

Stosunek długości przedziałów to $3.132\,77$, a więc metoda potwierdziła swoją skuteczność.

koniec przykładu 4.3

4.3. Zmienne kontrolne

Metoda zmiennych kontrolnych wykorzystuje błędy w ocenie znanych wielkości, by zredukować błędy oszacowania wielkości nieznanych.

Niech $Y_1, Y_2, \dots$ będzie ciągiem niezależnych zmiennych losowych o jednakowym rozkładzie, których realizacja dla ustalonego n pozwoli oszacować EY_1 metodą Monte Carlo jako realizację estymatora

$$\overline{Y} = (Y_1 + Y_2 + \cdots + Y_n)/n.$$

Niech

$$C_1, C_2, \ldots$$

będzie ciągiem zmiennych losowych takich, że

$$(Y_1, C_1), (Y_2, C_2), \ldots$$

jest ciągiem niezależnych zmiennych losowych o jednakowym rozkładzie.

Załóżmy, że zmienne C_i mają wartość oczekiwaną i niech dla $b \in \mathbb{R}$

$$Y_i^{(b)} = Y_i - b(C_i - EC_1)$$

oraz

$$\overline{Y^{(b)}} = (Y_1^{(b)} + Y_2^{(b)} + \cdots + Y_n^{(b)})/n.$$

Twierdzenie 4.2. *Dla dowolnego $b \in \mathbb{R}$ estymator $\overline{Y^{(b)}}$ jest zgodnym i nieobciążonym estymatorem EY_1.*

Uzasadnienie. Estymator `średnia z próby` jest nieobciążonym i zgodnym estymatorem wartości oczekiwanej (o ile ona istnieje), a ponadto

$$EY_1 = EY_1^{(b)}.$$

□

Fakt 4.2. *Wprost z podstawowych własności wariancji i kowariancji, jeśli założymy istnienie $\operatorname{Var} Y_1$ oraz $\operatorname{Var} C_1$, to*

$$\operatorname{Var} Y_1^{(b)} = \operatorname{Var} Y_1 - 2b \operatorname{Cov}(Y_1, C_1) + b^2 \operatorname{Var} C_1. \tag{4.8}$$

Wynika z tego, że wariancja estymatora $\overline{Y^{(b)}}$ jest minimalizowana dla

$$b^* = \frac{\sigma_{Y_1}}{\sigma_{C_1}} \varrho(Y_1, C_1), \tag{4.9}$$

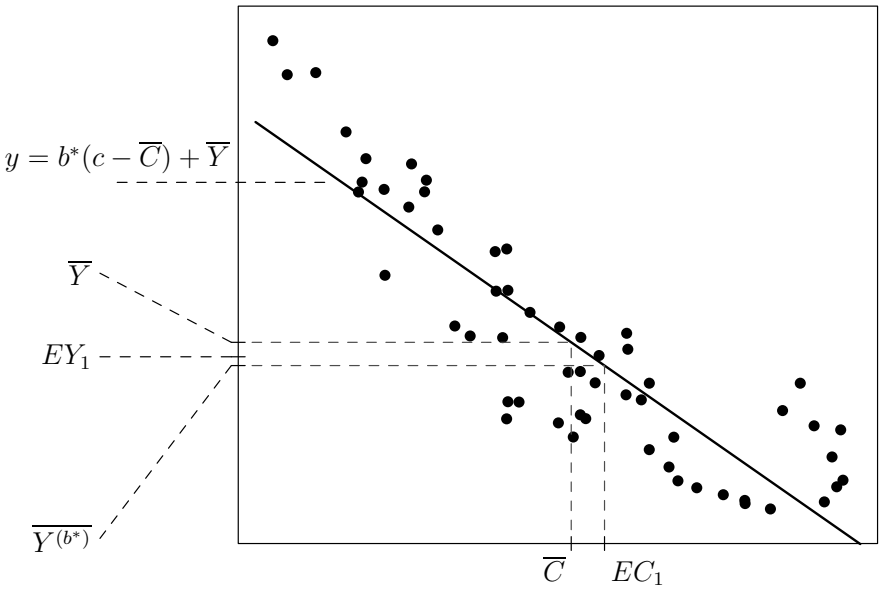
gdzie $\varrho(Y_1, C_1)$ to współczynnik korelacji między Y_1 i C_1. Wstawiwszy (4.9) do (4.8) otrzymamy, że

$$\text{v. r. r.} = \frac{\operatorname{Var} \overline{Y}}{\operatorname{Var} \overline{Y^{(b^*)}}} = \frac{\operatorname{Var} Y_1}{\operatorname{Var} Y_1^{(b^*)}} = \frac{1}{1 - \varrho_{CY}^2}. \tag{4.10}$$

Stąd wniosek, że korzyści z metody (czyli znaczny współczynnik v. r. r.) są duże, gdy korelacja między Y_1 i C_1 jest silna, niezależnie od jej kierunku.

By ułatwić interpretację tej metody zauważmy jeszcze tylko, że wzór (4.9) jest współczynnikiem kierunkowym prostej regresji drugiego rodzaju objaśniającej Y_1 za pomocą C_1.

Rozważmy rysunek 4.1.



Rys. 4.1. Interpretacja metody zmiennych kontrolnych

Dla danych z rysunku średnia $\overline{C}$ jest zaniżona w stosunku do wartości oczekiwanej EC_1. Korelacja między Y_1 i C_1 jest ujemna (na co wskazuje nachylenie linii regresji), zatem spodziewamy się, że średnia $\overline{Y}$ jest zawyżona w stosunku do EY_1, w związku z tym korygujemy tę średnią o właściwą wielkość (wynikającą z nachylenia linii), uzyskując $\overline{Y}^{(b*)}$, co jest na ogół bliższe szukanej wartości EY_1.

Przykład 4.4 (Pierwotne zmienne kontrolne)**.** *Wykonując obliczenia metodą Monte Carlo używamy zwykle zmiennych losowych U_i o znanym rozkładzie – najczęściej jednostajnym na $[0, 1]$. Wtedy można przyjąć jako C_i wartość $f(U_i)$, jeśli tylko jesteśmy w stanie podać dokładną wartość $E(f(U_1))$. Tą dokładną wartością oczekiwaną może być na przykład moment ustalonego rzędu, najlepiej tak dobrany, by zwiększać* v. r. r.

Przykład 4.5. *Rozważając wycenę opcji na akcje przeprowadza się symulację, w której będą występowały między innymi zmienne losowe odpowiadające funkcji dyskonta, wartości papieru podstawowego, wartości opcji. Ponieważ – przy założeniu braku możliwości arbitrażu – zakłada się, że wartość oczekiwana wartości obecnej papieru podstawowego w ustalonym odległym momencie czasowym jest równa jego aktualnej (dzisiejszej) cenie, więc wartość obecna papieru podstawowego jest naturalnym kandydatem na zmienną kontrolną.*

Oczywiście zwykle nie mamy możliwości dokładnego poznania b^*, więc wielkość ta jest estymowana np. wzorem

$$\widehat{b} = \frac{\sum_{i=1}^{n}(C_i - \overline{C})(Y_i - \overline{Y})}{\sum_{i=1}^{n}(C_i - \overline{C})^2}.$$

Jest to standardowy estymator współczynnika kierunkowego prostej regresji [4].

Niestety zastępując b^* przez $\widehat{b}$ sprawiamy, że estymator $\overline{Y^{(b)}}$ jest na ogół obciążony (choć oczywiście jest asymptotycznie nieobciążony). Wynika to z tego, że $\widehat{b}$ zależy od C_i, nie może więc być potraktowane jak stała. Istnieje kilka sposobów radzenia sobie z tym problemem, w praktyce jednak estymator $Y^{(\widehat{b})}$ ma bardzo dobre właściwości.

Jedna z prostszych metod pozbycia się obciążenia, to szacowanie $\widehat{b}$ na podstawie części realizacji (Y_i, C_i), następnie budowa estymatora i kontynuacja obliczeń na podstawie pozostałych realizacji (Y_i, C_i). Ze względu na niezależność obu zbiorów obserwacji, nieobciążoność jest jasna, oczywiście kosztem powiększenia wariancji, gdyż z powodu małej wielkości próby użytej do budowania $\widehat{b}$ jego wartość zapewne nieco będzie się różnić od optymalnej wartości b^*.

Przypadek wielowymiarowy

Metoda zmiennych kontrolnych ma naturalne uogólnienia na przypadek wielokrotny.

Przykład 4.6. *W przypadku pierwotnych zmiennych kontrolnych, omawianych w przykładzie 4.4 na poprzedniej stronie, można rozważać wiele funkcji obliczanych na podstawie zmiennych wykorzystywanych do generowania próbek przy symulacji – na przykład mogą to być momenty różnego rzędu. Stąd dostaniemy wiele zmiennych kontrolnych o znanych wartościach oczekiwanych.*

Przykład 4.7. *W przykładzie 4.5, który znajduje się na poprzedniej stronie, powiedziane zostało, że wartość obecna papieru wartościowego w dowolnie ustalonym momencie jest właściwym kandydatem na zmienną kontrolną. Ale takich momentów możemy sobie wybrać dowolnie dużo, możemy więc mieć wiele zmiennych kontrolnych tego rodzaju.*

Załóżmy, że mamy zmienne $Y_1, Y_2, \dots$ (ich realizacjami będą wyniki symulacji) oraz $C_i = [C_i^{(1)}, \dots, C_i^{(d)}]^T$, załóżmy ponadto, że znamy wektor wartości oczekiwanych EC_1, że pary

$$(C_i, Y_i), \qquad i = 1, 2, \dots, n$$

są niezależne i o jednakowym rozkładzie oraz istnieje macierz kowariancji

$$\begin{bmatrix} \Sigma_C & \Sigma_{CY} \\ \Sigma_{CY}^T & \sigma_Y^2 \end{bmatrix},$$

która jest nieosobliwa.

Wtedy również rozważamy estymator

$$\overline{Y^{(b)}} = \overline{Y} - b^T(\overline{C} - EC),$$

gdzie $b \in \mathbb{R}^d$. W zapisach macierzowych b traktowane jest jak macierz kolumnowa.

Okazuje się [12], że optymalnym wyborem b minimalizującym wariancję jest

$$b^* = \Sigma_C^{-1}\Sigma_{CY},$$

a $\mathrm{v.r.r.}$ wynosi $1/(1-R^2)$, gdzie

$$R^2 = \Sigma_{CY}^T\Sigma_C^{-1}\Sigma_{CY}/\sigma_Y^2.$$

W praktyce nie znamy b^*, lecz estymujemy formułą

$$\hat{b} = S_C^{-1}S_{CY},$$

gdzie w $d \times d$ macierzy S_C element w i-tym wierszu i j-tej kolumnie wynosi

$$\frac{1}{n-1}\left(\sum_{k=1}^{n} C_k^{(i)}C_k^{(j)} - n \cdot \overline{C^{(i)}} \cdot \overline{C^{(j)}}\right).$$

Dokładniejsze przyjrzenie się wszystkim wzorom dla przypadku wielu zmiennych kontrolnych pokazuje, że są one uogólnieniem podstawowego przypadku jednej zmiennej kontrolnej.

Warto jeszcze dodać, że w przypadku wysokiej korelacji między jakąś parą zmiennych kontrolnych, lepiej jest zmniejszyć liczbę zmiennych kontrolnych, gdyż macierz S_C w takiej sytuacji może być trudna do odwrócenia.

4.4. Losowanie istotnościowe

Opisane wcześniej metody redukcji wariancji mogą być bardzo skuteczne, jednak nie w każdej sytuacji da się je zastosować. Metoda omawiana w tym podrozdziale nie tylko pozwala na znaczące przyśpieszenie zbieżności rozwiązania osiąganego metodą Monte Carlo (zwłaszcza gdy gęstość rozkładu zmiennych losowych jest mocno skupiona w jakimś obszarze), ale w dodatku umożliwia generowanie zmiennych losowych o całkiem innym rozkładzie. Generowanie liczb z „oryginalnego" rozkładu może być bardzo trudne lub nawet niemożliwe (na przykład dystrybuanta ma trudną postać lub gęstość jest znana tylko z dokładnością do stałej normującej).

Angielski termin dla omawianej metody to *importance sampling*. Tłumaczenia na język polski, z którymi można się spotkać, to zarówno losowanie istotnościowe, jak i losowanie istotne.

Główne twierdzenie sformułujemy w terminach gęstości, jednak można również dokonać analogicznego sformułowania dla rozkładów dyskretnych. Sytuacja jest pod tym względem nieco zbliżona do podrozdziału 3.3.

Twierdzenie 4.3. *Niech X będzie d-wymiarową zmienną losową, niech $f(\cdot)$ będzie jej gęstością, niech $h : \mathbb{R}^d \to \mathbb{R}$ będzie funkcją taką, że istnieje*

$$Eh(X).$$

Niech Y będzie d-wymiarową zmienną losową o gęstości $g(\cdot)$ takiej, że

$$f(x) > 0 \implies g(x) > 0, \quad \text{dla } x \in \mathbb{R}^d.$$

Wtedy

$$Eh(X) = E\left(h(Y)\frac{f(Y)}{g(Y)}\right). \tag{4.11}$$

Iloraz funkcji $f(\cdot)$ i $g(\cdot)$ we wzorze (4.11) traktujemy jako zerowy wszędzie tam, gdzie mianownik jest równy zero.

Dowód.

$$\begin{aligned} Eh(X) &= \int_{R^d} h(x)f(x)\,dx = \int_{R^d} \frac{h(x)f(x)}{g(x)}g(x)\,dx = \\ &= E\left(h(Y)\frac{f(Y)}{g(Y)}\right). \end{aligned}$$

□

Oznaczmy przez $\hat{\alpha}(n)$ podstawowy estymator wielkości $Eh(X)$ bazujący na n-elementowej próbie, tzn.

$$\hat{\alpha}(n) = \frac{1}{n}\sum_{i=1}^{n} h(X_i),$$

natomiast przez $\hat{\alpha_g}(n)$ – estymator istotnościowy

$$\hat{\alpha_g}(n) = \frac{1}{n}\sum_{i=1}^{n} h(Y_i)\frac{f(Y_i)}{g(Y_i)}, \tag{4.12}$$

gdzie $X_1, X_2 \ldots$ są niezależne o gęstości $f(\cdot)$ oraz $Y_1, Y_2, \ldots$ są niezależne o gęstości $g(\cdot)$.

Wartości oczekiwane estymatorów $\hat{\alpha}(n)$ i $\hat{\alpha}_g(n)$ są jednakowe, zatem porównując wariancję wystarczy porównać drugie momenty.

$$\begin{aligned} E\left(h(Y)\frac{f(Y)}{g(Y)}\right)^2 &= \int_{\mathbb{R}^d}\left(h(y)\frac{f(y)}{g(y)}\right)^2 g(y)\,dy = \\ &= \int_{\mathbb{R}^d} h(y)^2\frac{f(y)}{g(y)}f(y)\,dy = E\left(h(X)^2\frac{f(X)}{g(X)}\right), \end{aligned}$$

co oczywiście może być dużo większe bądź dużo mniejsze niż

$$Eh(X)^2,$$

wszystko zależy od doboru funkcji $g(\cdot)$.

Ogólna wskazówka: dobrze byłoby, gdyby gęstość $g(\cdot)$ była bliska funkcji proporcjonalnej do iloczynu funkcji $h(\cdot)$ oraz $f(\cdot)$. Łatwo można zauważyć, że gdyby gęstość $g(\cdot)$ była z prawdopodobieństwem 1 proporcjonalna do iloczynu $(hf)(\cdot)$, to wariancja estymatora $\alpha_g(n)$ wyniosłaby 0. Widać to natychmiast, gdy wyobrazimy sobie taką sytuację patrząc na wzór (4.12). Wszystkie składniki sumy byłyby wtedy identyczne.

W szczególnym przypadku, gdy $h(\cdot)$ jest indykatorem pewnego zbioru $A \subset \mathbb{R}^d$, optymalnym doborem gęstości $g(\cdot)$ jest oryginalna gęstość warunkowana zbiorem A. W praktyce znaczy to, że należy wybrać taką gęstość $g(\cdot)$, by zdarzeniu przynależności do zbioru A dać wysokie prawdopodobieństwo, no i oczywiście, żeby na zbiorze A miała ona wartości w przybliżeniu proporcjonalne do $f(\cdot)$.

4.5. Losowanie warstwowe

Załóżmy, że chcemy oszacować EY dla pewnej zmiennej losowej Y. Niech X będzie inną zmienną (być może wielowymiarową), dla której istnieją rozłączne niepuste zbiory $A_1, A_2, \ldots, A_k$, dla których potrafimy obliczać $p_i = \Pr(X \in A_i)$. Wtedy oczywiście

$$EY = \sum_{i=1}^{k} p_i E(Y|X \in A_i). \tag{4.13}$$

Rzecz jasna szacowanie każdej z wartości $E(Y|X \in A_i)$ może się odbywać metodą Monte Carlo na podstawie próbki n_i elementowej, przy czym liczbę n_i można dobierać według rożnych kryteriów. Ze względu na charakter tego rozdziału, najistotniejsza będzie dla nas minimalizacja wariancji estymatora.

Jeżeli uzyskanie pojedynczej realizacji Y wymaga całego ciągu wielu próbek losowych – taka sytuacja ma miejsce choćby przy badaniu metodą symulacyjną własności procesu stochastycznego – nie

stosujemy tej metody (podziału na warstwy) „wewnątrz" trajektorii; stosujemy ją „między" trajektoriami, czyli całe trajektorie będą zaliczane do poszczególnych warstw. Na przykład znając rozkład końcowych wartości trajektorii można, uwzględniając narzucone prawdopodobieństwa na warstwy, wylosować jej zakończenia, następnie – dla każdej trajektorii osobno – wylosować uzupełnienia. Do tej koncepcji nawiążemy w podrozdziale 5.3.7.

W wielu wypadkach zbiory A_i dobiera się tak, by p_i były na zbliżonym poziomie. To może być wygodne, jednak nie ma to znaczenia dla minimalizacji wariancji estymatora warstwowego. Samo generowanie realizacji par (X, Y) jest bardzo zależne od konkretnego problemu, może być tak, że Y obliczany jest na podstawie X, związek ten może być jednak bardziej subtelny (jak choćby w omawianej w poprzednim akapicie sytuacji z trajektorią procesu stochastycznego, gdzie X mógłby być końcową wartością trajektorii, a Y byłby wartością obliczaną na podstawie całej trajektorii).

Niech $q_i = \frac{n_i}{n}$, gdzie n_i jest założoną liczbą próbek w i-tej warstwie, n jest liczbą wszystkich próbek, k jest liczbą warstw. Niech $\hat{Y}$ będzie warstwowym estymatorem EY, to znaczy

$$\hat{Y} = \sum_{i=1}^{k} p_i \frac{1}{n_i} \sum_{j=1}^{n_i} Y_{ij} = \sum_{i=1}^{k} \frac{1}{n} \frac{p_i}{q_i} \sum_{j=1}^{n_i} Y_{ij}, \tag{4.14}$$

gdzie $(X_{ij}, Y_{ij}), j = 1, 2, \ldots n_i$ to ciąg niezależnych zmiennych losowych mających rozkład taki jak $(X, Y)|X \in A_i$.

W dalszym ciągu tego podrozdziału będziemy zakładali, że dla każdej warstwy istnieje warunkowa wariancja $Y|X \in A_i$, dla tej wariancji oraz dla warunkowych wartości oczekiwanych będziemy używać oznaczeń σ_i^2 oraz μ_i, to znaczy

$$\sigma_i^2 = \mathrm{Var}(Y|X \in A_i) \quad \text{oraz} \quad \mu_i = E(Y|X \in A_i), \quad \text{dla } i = 1, 2, \ldots, k.$$

Lemat 4.1. *Wartość oczekiwana estymatora warstwowego $E\hat{Y}$ jest równa wartości oczekiwanej EY.*

Dowód. Teza wynika natychmiast z liniowości wartości oczekiwanej oraz równości (4.13) i (4.14):

$$E\widehat{Y} = \sum_{i=1}^{k} \frac{1}{n} \frac{p_i}{q_i} n_i \mu_i = \sum_{i=1}^{k} p_i \mu_i = EY.$$

□

Następnie zauważymy, że w sytuacji, gdy liczebności n_i próbek w poszczególnych warstwach dobierzemy proporcjonalnie do prawdopodobieństw p_i dla poszczególnych warstw, to wariancja estymatora warstwowego zmniejszy się w stosunku do wariancji estymatora standardowego.

Lemat 4.2. *Załóżmy, że przy oznaczeniach użytych w tym podrozdziale zachodzi równość $p_i = q_i$. Wtedy*

$$\operatorname{Var} \widehat{Y} \leqslant \operatorname{Var} \overline{Y},$$

przy czym nierówność stanie się ostra, jeśli nie wszystkie μ_i są identyczne.

Dowód. Na podstawie wzoru (4.14) możemy zapisać, że

$$\operatorname{Var} \widehat{Y} = \sum_{i=1}^{k} p_i^2 \operatorname{Var} \left(\frac{1}{n_i} \sum_{j=1}^{n_i} Y_{ij} \right) = \sum_{i=1}^{k} \frac{p_i^2}{n_i} \sigma_i^2. \tag{4.15}$$

Jeśli $n_i = p_i \cdot n$, to

$$\operatorname{Var} \widehat{Y} = \sum_{i=1}^{k} \frac{p_i^2}{n p_i} \sigma_i^2 = \frac{1}{n} \sum_{i=1}^{k} p_i \sigma_i^2.$$

Zauważmy, że

$$EY^2 = \sum_{i=1}^{k} p_i E(Y^2 | X \in A_i) = \sum_{i=1}^{k} p_i (\mu_i^2 + \sigma_i^2).$$

Zatem

$$\begin{aligned}\operatorname{Var}\overline{Y} &= \frac{1}{n}\operatorname{Var} Y = \frac{1}{n}\left(EY^2 - (EY)^2\right) = \\ &= \frac{1}{n}\left(\sum_{i=1}^{k} p_i(\mu_i^2+\sigma_i^2) - \Big(\sum_{i=1}^{k} p_i\mu_i\Big)^2\right) \geqslant \\ &\geqslant \frac{1}{n}\left(\sum_{i=1}^{k} p_i(\mu_i^2+\sigma_i^2) - \sum_{i=1}^{k} p_i\mu_i^2\right) = \operatorname{Var}\hat{Y}.\end{aligned}$$

Skorzystaliśmy tu z nierówności

$$\Big(\sum_{i=1}^{k} p_i\mu_i\Big)^2 \leqslant \sum_{i=1}^{k} p_i\mu_i^2,$$

która jest po prostu nierównością Jensena [9] dla funkcji wypukłej

$$f(x) = x^2.$$

Nierówność ta w przypadku, gdy nie wszystkie liczby μ_i są identyczne jest nierównością ostrą. □

Twierdzenie 4.4. *Zachodzi równość*

$$E(\hat{Y}) = EY,$$

a ponadto wariancja estymatora $\hat{Y}$ osiąga minimum dla

$$q_i^* = \frac{p_i\sigma_i}{\sum_{j=1}^{k} p_j\sigma_j}, \quad i = 1, 2, \ldots, k.$$

Minimum to wynosi

$$\frac{1}{n}\left(\sum_{i=1}^{k} p_i\sigma_i\right)^2 \leqslant \operatorname{Var}\overline{Y}.$$

Dowód. Wypowiedź na temat wartości oczekiwanej była uzasadniona w lemacie 4.1.

Wzór (4.15) zapiszmy – korzystając z $n_i = q_i n$ – w postaci

$$\frac{1}{n}\sum_{i=1}^{k}\frac{p_i^2}{q_i}\sigma_i^2. \tag{4.16}$$

Chcemy znaleźć układ liczb $q_i, i = 1, 2, \dots, k$, minimalizujący wariancję, ale oczywiście przy spełnieniu warunku

$$q_1 + q_2 + \dots q_k = 1.$$

Skorzystamy z metody mnożników Lagrange'a: w tym celu definiujemy funkcję

$$F(q_1, q_2, \dots, q_k, \lambda) = \frac{1}{n} \sum_{i=1}^{k} \frac{p_i^2}{q_i} \sigma_i^2 - \lambda(q_1 +_2 + \dots q_k - 1),$$

dla której szukamy punktu ekstremalnego, rozwiązując układ

$$\begin{cases} q_1 + q_2 + \dots q_k = 1 \\ \frac{-p_i^2 \sigma_i^2}{q_i^2} - \lambda n = 0 \qquad \text{dla } i = 1, 2, \dots k. \end{cases}$$

Stąd obliczamy

$$q_i = \frac{p_i \sigma_i}{\sum_{j=1}^{k} p_j \sigma_j}, \quad \text{dla } i = 1, 2, \dots, k.$$

Jest to jedyny punkt, w którym może być poszukiwana ekstremalna wartość. Aby przekonać się, że jest to ekstremum minimum zauważmy, że gdyby takie właśnie wartości wstawić do wzoru (4.16), to otrzymalibyśmy

$$\begin{aligned} \operatorname{Var} \hat{Y} &= \frac{1}{n} \sum_{i=1}^{k} \left(\frac{p_i^2 \sigma_i^2}{p_i \sigma_i} \sum_{j=1}^{k} p_j \sigma_j \right) = \frac{1}{n} \left(\sum_{i=1}^{k} p_i \sigma_i \right)^2 \leqslant \\ &\leqslant \frac{1}{n} \sum_{k=1}^{n} p_i \sigma_i^2 \leqslant \operatorname{Var} \overline{Y}. \end{aligned}$$

Pierwsza z powyższych nierówności to znów nierówność Jensena dla funkcji kwadratowej, natomiast druga została udowodniona przy okazji uzasadniania lematu 4.2 na stronie 83. □

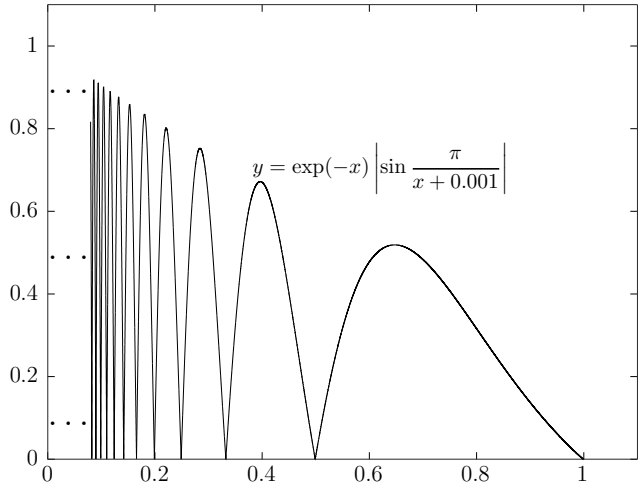
Oczywiście wielkości σ_i nie są na ogół znane – w praktyce estymuje się je za pomocą pomocniczych losowań, następnie dopiero przeprowadza się zoptymalizowane losowanie warstwowe. Wiadomo też, że trudno się spodziewać, by liczby $n \cdot q_i$ dla optymalnych q_i były

całkowite, jednak jeśli n jest dostatecznie duże (co przeważnie ma miejsce w próbach Monte Carlo), a liczba warstw jest względnie niewielka, to faktyczna wariancja warstwowego estymatora z liczbami n_i tylko w przybliżeniu równymi nq_i (dla optymalnych q_i) będzie bardzo bliska tej optymalnej wariancji, o której mówi twierdzenie 4.4.

Przykład 4.8. *Rozważmy obliczenie całki*

$$\int_0^1 \exp(-x) \left|\sin \frac{\pi}{x + 0.001}\right| dx.$$

Widać ze wzoru funkcji podcałkowej (zob. także rysunek 4.2), że najwięcej kłopotu w rachunkach – ze względu na duże wahanie wartości – przysparza lewa część przedziału $[0, 1]$. Dla konkretnych rachunków przyjmiemy $k = 10$ warstw.



Rys. 4.2. Wykres funkcji podcałkowej z przykładu 4.8. Zbyt duże wahanie funkcji uniemożliwiło dokładny rysunek w skrajnie lewej części przedziału

Używając terminologii tego podrozdziału, zmienną losową X jest zmienna o rozkładzie jednostajnym w $[0, 1]$, zmienna losowa Y (której wartość oczekiwana to interesująca nas wielkość) to

$$Y = \exp(-X) \left|\sin \frac{\pi}{X + 0.001}\right|,$$

natomiast zbiory A_i to przedziały lewostronnie domknięte postaci

$$A_i = [(i-1)\cdot 0.1, i\cdot 0.1), \quad i = 1, 2, \ldots, 10.$$

Dla każdej warstwy A_i, na podstawie $m = 1000$ próbek, oszacowana została wariancja σ_i^2. Każde z p_i jest w tym wypadku oczywiście równe 0.1. Na tej podstawie już można (korzystając z twierdzenia 4.4) dokonać ustalenia przybliżeń optymalnych wielkości próbek. Dla dziesięciu warstw (od przedziału $[0, 0.1)$ zaczynając) jest to około: 20%, 18%, 15%, 14%, 14%, 9%, 1%, 3%, 3%, 3%.

Po zapoznaniu się z tymi wynikami zauważymy, że rysunek 4.2 na poprzedniej stronie zdaje się je potwierdzać: na przykład przedział $[0.6, 0.7)$ na oko cechuje się najmniejszą zmiennością funkcji podcałkowej, zatem trzeba mu poświęcić najmniej „uwagi".

Na zakończenie warto dodać, że standardowe pakiety do obliczeń numerycznych mają kłopoty z radzeniem sobie z tą całką. Na przykład standardowa konfiguracja funkcji `quad_quags` w programie `Maxima` bądź funkcji `integrate` w języku `R` jest niewystarczająca, trzeba zwiększać liczbę podziałów przedziału całkowania (w przypadku `Maxima` także tzw. tolerancję). Po „dostrojeniu" obie te funkcje dają wynik w przybliżeniu zgodny i potwierdzany metodą Monte Carlo. Na przykład funkcja `romberg` w programie `Maxima` liczy bez żadnych technicznych błędów i ostrzeżeń, ale daje wynik znacząco inny niż pozostałe z wymienionych, zgodne metody.

koniec przykładu 4.8

4.5.1. Przedział ufności dla losowania warstwowego

Przy obliczaniu przedziału ufności można przyjąć poniższą procedurę.

(a) Środek przedziału ufności $\hat{y}$ wyznaczamy jako realizację estymatora warstwowego

$$\hat{y} = \sum_{i=1}^{k} \frac{p_i}{n_i} \sum_{j=1}^{n_i} y_{ij}.$$

(b) Oszacowanie $\hat{s}^2$ wariancji estymatora warstwowego można obliczyć ze wzoru wynikającego z (4.15), tzn.

$$\hat{s}^2 = \sum_{i=1}^{k} \frac{p_i^2}{n_i} \hat{s_i}^2,$$

gdzie $\hat{s_i}^2$ jest wartością nieobciążonego estymatora wariancji dla próbki

$$y_{i1}, y_{i2}, \dots y_{in_i}.$$

(c) Przedział ufności na poziomie ufności $1-\alpha$ będzie dany wzorem

$$\Big(\hat{y} - z(1-\alpha/2)\hat{s}, \hat{y} + z(1-\alpha/2)\hat{s}\Big),$$

gdzie $z(1-\alpha/2)$ jest kwantylem rzędu $1-\alpha/2$ rozkładu normalnego.

Opisane wyżej postępowanie wynika z tego, że rozkład średnich obliczanych wewnątrz warstw jest w dobrym przybliżeniu normalny, z kolei suma kilku zmiennych o rozkładzie normalnym (czyli w przybliżeniu średnich z poszczególnych warstw) także ma rozkład normalny.

4.6. Metoda quasi-Monte Carlo

Można zauważyć, że losując bardziej „równomiernie" w wariancie metody zmiennych antytetycznych, który wraz z $f(X)$ – dla X z rozkładu jednostajnego w $[0,1]$ – wprowadzał także do liczenia średniej wartość $f(1-X)$, szybciej uzyskujemy zadowalający wynik. Oczywiście to nie może być aż tak proste: gdy pozbędziemy się losowania na rzecz tej równomierności, regularności, to dużo tracimy. Przestrzeń, z której trzeba losować punkty dla uzyskania pojedynczego wyniku może być ogromna i wtedy nie będzie możliwe regularne jej próbkowanie, a przede wszystkim ta nadmierna regularność zabierze nam możliwość korzystania z praw statystyki, choćby przy wyliczaniu przedziałów ufności.

Mimo tych zastrzeżeń są sytuacje, gdy takie regularne przeglądanie przestrzeni ma sens i bywa stosowane. Część idei jest zbliżona do tych stosowanych w metodzie Monte Carlo, zatem zostanie to

omówione właśnie w tym rozdziale, dotyczącym dokładności obliczeń. Nazwa tej metody to „metoda quasi-Monte Carlo".

Załóżmy, że naszym celem jest obliczenie wartości oczekiwanej zmiennej losowej $f(U_1, U_2, \dots, U_d)$, to znaczy

$$Ef(U_1, U_2, \dots, U_d) = \int_{[0,1)^d} f(x)\, dx,$$

gdzie U_i są niezależnymi zmiennymi losowymi o rozkładzie jednostajnym na $[0, 1]$, zaś

$$f\colon \mathbb{R}^d \to \mathbb{R}.$$

W metodzie quasi-Monte Carlo

$$\int_{[0,1)^d} f(x)\, dx \approx \frac{1}{n} \sum_{i=1}^{n} f(x_i), \tag{4.17}$$

gdzie $x_i \in [0, 1)^d$, a ciąg $x_1, x_2, \dots$ jest deterministycznym ciągiem w kostce d-wymiarowej, który powinien w sposób „równomierny" wypełniać tę kostkę. Dlatego w tej metodzie ważne jest ścisłe określenie wymiaru problemu, to znaczy liczby elementów z $[0, 1)$ potrzebnych do wygenerowania jednej z uśrednianych w (4.17) wartości. Zauważmy, że w zwykłej metodzie Monte Carlo nie przejmujemy się tym zbytnio; w szczególności przy stosowaniu metody akceptacji/odrzucania nawet nie wiadomo dokładnie, ile liczb potrzeba do otrzymania jednej z uśrednianych wartości.

Oczywiście pojęcie „równomiernego" wypełniania musi być jakoś uściślone, do tego będą służyły pojęcia rozbieżności i niskiej rozbieżności.

4.6.1. Ciągi o niskiej rozbieżności

Jako miarę rozbieżności ciągów n-elementowych w kostce $[0, 1)^d$ przyjmuje się [26]

$$D_n(x_1, x_2, \dots, x_n) = \sup_{A \in \mathcal{A}} \left| \frac{|\{x_1, \dots, x_n\} \cap A|}{n} - \mu(A) \right|,$$

gdzie $\mu(A)$ jest d-wymiarową miarą Lebesgue'a zbioru A, zaś $\mathcal{A}$ jest zbiorem tych podzbiorów kostki d-wymiarowej, które są iloczynami kartezjańskimi przedziałów lewostronnie domkniętych i prawostronnie otwartych.

Jest też druga, podobna miara, mianowicie

$$D_n^*(x_1, x_2, \dots, x_n) = \sup_{A \in \mathcal{A}^*} \left| \frac{|\{x_1, \dots, x_n\} \cap A|}{n} - \mu(A) \right|,$$

gdzie $\mathcal{A}^*$ to zbiór tych elementów z $\mathcal{A}$, które są zbudowane na bazie przedziałów o lewym końcu równym 0.

Relacja między tymi miarami może być przedstawiona poniższą nierównością:

$$D_n^*(x_1, x_2, \dots, x_n) \leqslant D_n(x_1, x_2, \dots, x_n) \leqslant 2^d D_n^*(x_1, x_2, \dots, x_n).$$

Na przykład dla $d = 1$ wiadomo, że rozbieżność D_n^* ciągu n-elementowego jest od dołu ograniczona przez $\frac{1}{2n}$ (dla miary D_n to ograniczenie wynosi $\frac{1}{n}$) i ograniczenie dolne jest realizowane przez ciąg

$$x_i = \frac{2i-1}{2n}, i = 1, 2, \dots, n,$$

zarówno dla miary D_n^*, jak i dla D_n.

Gdyby jednak chcieć wskazać nieskończony ciąg i mierzyć rozbieżności jego początkowych fragmentów, to

$$D_n^*(x_1, x_2, \dots, x_n) \geqslant \frac{c \ln n}{n}.$$

Przy zwiększaniu wymiaru d przyjmuje się, że ma miejsce ograniczenie (w sytuacji, gdy mierzona jest rozbieżność początkowego fragmentu ciągu nieskończonego)

$$D_n^*(x_1, x_2, \dots, x_n) \geqslant \frac{c_d (\ln n)^d}{n},$$

przy tym c_d jest pewną stałą zależną tylko od wymiaru d. Ta hipoteza jest co prawda udowodniona tylko dla $d \leqslant 2$, jednakże stała się podstawą do przyjęcia, że ciągiem niskiej rozbieżności nazywamy taki

ciąg nieskończony $x_1, x_2, \ldots$, gdzie $x_i \in [0,1)^d$, dla którego

$$D_n^*(x_1, x_2, \ldots, x_n) = O\left(\frac{(\ln n)^d}{n}\right),$$

lub innymi słowy: istnieje taka stała $k > 0$, że dla dostatecznie dużych $n \in \mathbb{N}$ zachodzi

$$D_n^*(x_1, x_2, \ldots, x_n) \leqslant k \cdot \frac{(\ln n)^d}{n}.$$

Przy pewnych założeniach odnośnie funkcji $f(\cdot)$ (skończone wahanie $f(\cdot)$ w sensie Hardy'ego i Krause'a [26]) można wykazać, że dla ciągów o niskiej rozbieżności, dla dowolnego $\varepsilon > 0$ istnieje $k > 0$, że

$$\left| \int_{[0,1)^d} f(x)\, dx - \frac{1}{n} \sum_{i=1}^{n} f(x_i) \right| < k \cdot \frac{1}{n^{1-\varepsilon}} \tag{4.18}$$

dla prawie wszystkich n.

W świetle formuły (4.1), która znajduje się na stronie 66, widać, że błąd rozwiązania otrzymywanego metodą Monte Carlo jest rzędu

$$O\left(\frac{1}{\sqrt{n}}\right),$$

zaś w przypadku metody quasi-Monte Carlo dla błędu mamy korzystniejsze oszacowanie

$$O\left(\frac{1}{n^{1-\varepsilon}}\right),$$

przy dowolnie małym $\varepsilon > 0$.

Jednak należy zwrócić uwagę na fakt, że nierówność (4.1) daje konkretne stałe mnożące czynnik $\frac{1}{\sqrt{n}}$. Natomiast stała k w nierówności (4.18) nie jest sprecyzowana i dla ε bliskiego zeru może być bardzo duża. Dlatego oszacowanie (4.18) nie ma może bardzo praktycznego znaczenia, raczej pokazuje ono, jak ważne mogą być ciągi o niskiej rozbieżności, które rzeczywiście w wielu przypadkach są skutecznym narzędziem obliczeniowym [12].

4.6.2. Ciąg Haltona

W podrozdziale tym zaprezentowany zostanie przykładowy ciąg o niskiej rozbieżności, budowany na bazie liczb pierwszych, zwany ciągiem Haltona. Jego konstrukcja została opisana w 1960 roku w artykule *On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals* [14].

Niech $d > 0$. Ciąg d-wymiarowy nieskończony $x_1, x_2, ...$, gdzie

$$x_k = (x_k^{(1)}, x_k^{(2)}, \dots, x_k^{(d)}), \qquad k \in \mathbb{N},$$

definiujemy poprzez:

$$x_k^{(i)} = a_0 \cdot p_i^{-1} + a_1 \cdot p_i^{-2} + \dots a_{m-1} p_i^{-m},$$

gdzie p_i jest i-tą liczbą pierwszą, zaś współczynniki $a_0, a_1, \dots a_{m-1}$ to cyfry zapisu liczby k przy podstawie p_i, to znaczy

$$k = a_0 + a_1 p_i + a_2 p_i^2 + \dots a_{m-1} p_i^{m-1}.$$

Przykład 4.9. *Jeśli przyjmiemy $d = 3$, to pięć pierwszych wyrazów ciągu Haltona przedstawia się następująco*

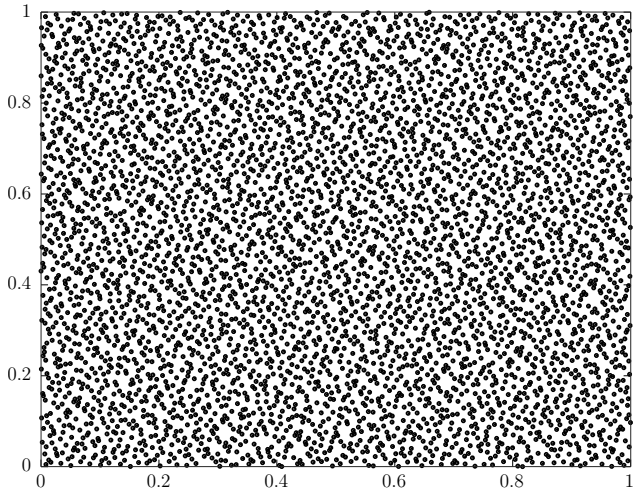
$$(\tfrac{1}{2}, \tfrac{1}{3}, \tfrac{1}{5}), (\tfrac{1}{4}, \tfrac{2}{3}, \tfrac{2}{5}), (\tfrac{3}{4}, \tfrac{1}{9}, \tfrac{3}{5}), (\tfrac{1}{8}, \tfrac{4}{9}, \tfrac{4}{5}), (\tfrac{5}{8}, \tfrac{7}{9}, \tfrac{1}{25}).$$

..... *koniec przykładu 4.9* .....

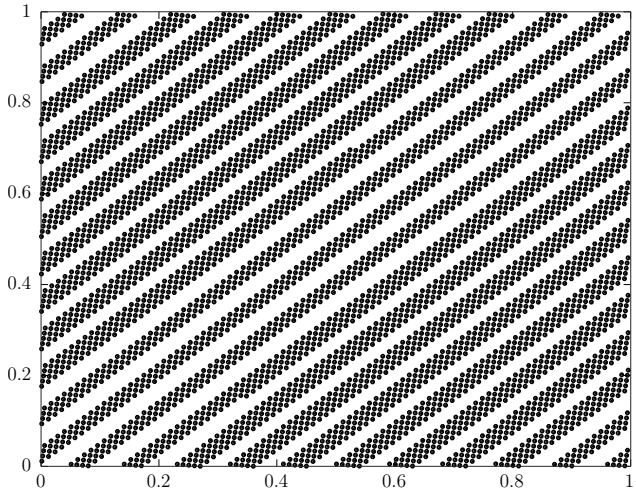
Na rysunku 4.3 pokazane jest 5000 elementów ciągu Haltona dla $d = 2$, pierwsza współrzędna to odcięta, druga to rzędna punktu na wykresie. Kwadrat wydaje się dość solidnie wypełniony.

Znacznie gorzej przedstawia się sytuacja tych współrzędnych, do użycia których wykorzystane były większe liczby pierwsze. Widać to dobrze na rysunku 4.4, gdzie rzutowane są współrzędne 14 i 15. Jak łatwo się domyślić, jeszcze więcej luk widać na rysunku z rzutowanymi współrzędnymi 39 i 40 (rys. 4.5).

Zatem ciąg ten – choć rzeczywiście `jest` ciągiem o niskiej rozbieżności – ma pewne słabości, które są przezwyciężone w innych konstrukcjach (jak ciągi Sobola [12]).



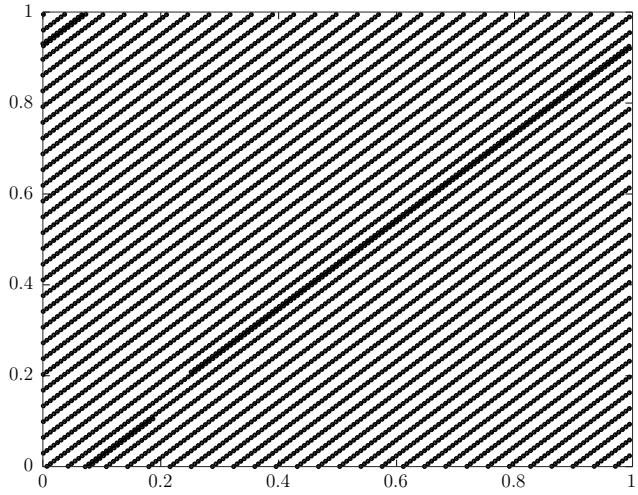
Rys. 4.3. 5000 elementów ciągu Haltona, $d = 2$



Rys. 4.4. 5000 elementów ciągu Haltona, rzutowane współrzędne 14 i 15

Zadania

Zadanie 4.1. Szacowanie przedziału ufności przy założeniach zadania 1.1, ze strony 22, zostało omówione w przykładzie 4.1. Dokonaj analogicznych obliczeń dla części (b) tego zadania.



Rys. 4.5. 5000 elementów ciągu Haltona, rzutowane współrzędne 39 i 40

Zadanie 4.2. Oceń, korzystając z modelu (4.3), długość uzyskiwanego przedziału ufności dla szukanej wartości liczby π w eksperymencie Lazzariniego ($t = 2.5$, $h = 3$, $n = 3408$ – przykład 1.2 na stronie 17). Przyjmij poziom ufności $1 - \alpha = 0.95$. Przeprowadź symulację komputerową eksperymentu.

Zadanie 4.3. Program rozwiązujący zadanie 1.3 wzbogać o obliczanie przedziału ufności – przy wykorzystaniu modelu (4.3) – przyjmując poziom ufności 99%. Na tej podstawie podaj ocenę błędu popełnianego w sytuacji, gdy jako wynik podamy środkową wartość przedziału.

Zadanie 4.4. Dla obliczenia całki

$$\int_0^{\pi/2} \sin x^2 \, dx,$$

przyjmując poziom ufności $1 - \alpha = 0.95$, wyznacz przedział ufności, korzystając z $100\,000\,000$ próbek:

(a) prostą metodą Monte Carlo,

(b) z użyciem „naturalnej” metody zmiennych antytetycznych.

Wyznacz stosunek długości przedziałów uzyskanych w (a) i (b).

Zadanie 4.5. Używając próbek wielkości $n = 100\,000\,000$, oszacuj (licząc 95% przedziały ufności) całkę

$$\int_0^5 \exp(-x^2)\,dx,$$

traktując szukaną wielkość jako

$$E\left(\frac{1}{5}\exp\left(-X^2\right)\right),$$

gdzie X ma rozkład jednostajny w $[0, 5]$.

(a) Użyj prostej metody Monte Carlo.

(b) Użyj metody zmiennych kontrolnych. Niech zmienną kontrolną będzie X. Na podstawie długości przedziałów w (a) i (b) podaj oszacowanie $\mathrm{v.\,r.\,r.}$

(c) Użyj metody zmiennych kontrolnych. Niech zmienną kontrolną będzie X^2. Na podstawie długości przedziałów w (a) i (c) podaj oszacowanie $\mathrm{v.\,r.\,r.}$

Zadanie 4.6. Rozważmy obliczenie całki

$$I = \int_0^2 x^2 e^{\sin x}\,dx$$

metodą Monte Carlo, używając koncepcji metody losowania istotnościowego. Obliczenie tej całki inną metodą numeryczną dało wynik w przybliżeniu $6.772\,267$.

Przyjmijmy na przykład – używając oznaczeń z twierdzenia 4.3 na stronie 79 – że

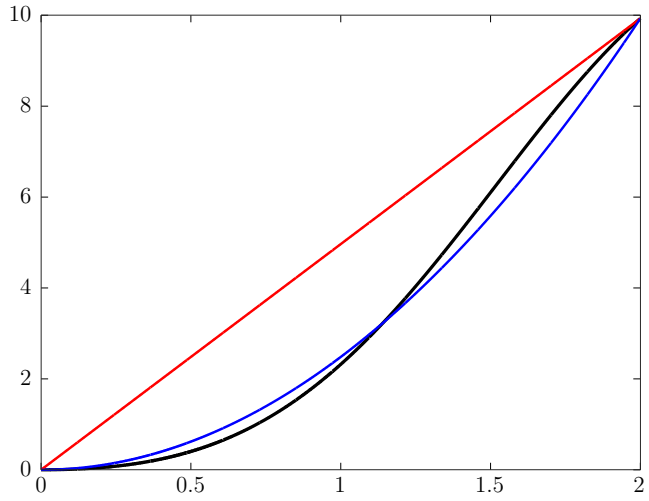
$$I = Eh(X),$$

gdzie X ma rozkład jednostajny na $[0, 2]$, oraz $h(x) = 2x^2 e^{\sin x}$.

Iloczyn $h(\cdot)$ i gęstości zmiennej losowej X jest funkcją ściśle rosnącą w przedziale $[0, 2]$ od wartości 0, co mogłoby sugerować przyjęcie jako gęstości $g(\cdot)$ na przykład funkcji rosnącej liniowo od zera w przedziale $[0, 2]$.

Wykonaj próbę tej metody dla $n = 500\,000\,000$, przy szacowaniu przedziałów ufności przyjmij poziom ufności $1 - \alpha = 0.95$, porównaj z wynikami podstawowej metody Monte Carlo i oszacuj $\mathrm{v.\,r.\,r.}$

Zadanie 4.7. W zadaniu 4.6 liniowa (w przedziale $[0, 2]$) funkcja gęstości $g(\cdot)$ dała całkiem niezłą redukcję wariancji, ale prawdziwie imponujący wynik dostaniemy poprzez jeszcze lepsze dopasowanie funkcji gęstości $g(\cdot)$ do iloczynu $h(\cdot)$ i $f(\cdot)$. Iloczyn ten pokazany jest na rysunku 4.6, widać, że styczna w lewym końcu przedziału jest pozioma, zatem jeszcze lepszym kandydatem na gęstość $g(\cdot)$ będzie parabola z wierzchołkiem w $(0, 0)$. Oczywiście z odpowiednim czynnikiem



Rys. 4.6. Kształt wykresu $h(\cdot)f(\cdot)$ z zadań 4.6 i 4.7 (linia czarna) wraz z przeskalowanymi wykresami linii $y = x$ oraz $y = x^2$

normującym. Dla takiej właśnie funkcji $g(\cdot)$ wykonaj polecenia analogiczne do tych z zadania 4.6.

Zadanie 4.8. Dla całki z przykładu 4.8, używając opisanego tam podziału na warstwy, wykonaj samodzielne obliczenia, wyznaczając bliskie optymalnym liczby n_i za pomocą 1000 próbek w każdej warstwie. Następnie wykonaj szacowanie przedziału ufności (biorąc $n = 100\,000\,000$, $1 - \alpha = 0.99$) dla metody klasycznej oraz warstwowej i wyznacz $\mathrm{v.r.r.}$

Zadanie 4.9. Napisz wartości $9.$ współrzędnej d-wymiarowego ciągu Haltona dla wyrazów o numerach $i, i+1, i+2, i+3, i+4$, gdzie $i = 1585$. Końcowe wyniki należy zaokrąglać do ośmiu cyfr znaczących.

Zadanie 4.2. Długość przedziału wynosi około 0.2. Wskazówka: przedział dla frakcji trzeba odwrócić i przeskalować czynnikiem $\frac{2t}{h}$.

Zadanie 4.3. Końce przedziału dla frakcji należy pomnożyć przez pole prostokąta, z którego losujemy jednostajnie punkty, w tym wypadku pole to szczęśliwie wynosi 1. Oszacowanie błędu to około $0.000\,376$.

Zadanie 4.4. Stosunek długości przedziałów to około 3.47.

Zadanie 4.5. Przybliżone wartości $\mathrm{v.\,r.\,r.}$: (b) 2.58, (c) 1.58.

Zadanie 4.6. $\mathrm{v.\,r.\,r.} \approx 5.22$.

Zadanie 4.7. $\mathrm{v.\,r.\,r.} \approx 93.4$.

Zadanie 4.8. $\mathrm{v.\,r.\,r.} \approx 2.55$.

Zadanie 4.9. Podstawą jest liczba 23, a żądane, właściwie zaokrąglone liczby to:

$$9.547\,957\,6 \cdot 10^{-1}, 9.982\,740\,2 \cdot 10^{-1}, 2.465\,685\,9 \cdot 10^{-4},$$
$$4.372\,482\,9 \cdot 10^{-2}, 8.720\,309\,0 \cdot 10^{-2}.$$

ROZDZIAŁ 5

Wybrane aspekty stosowania metody Monte Carlo

5.1. Całkowanie funkcji, obliczanie miar obszarów

Numeryczne obliczenia związane z wyznaczaniem wartości całek, to prawdopodobnie najczęstsze skojarzenie z metodą Monte Carlo. Właściwie każdą wartość oczekiwaną rozkładu można przestawić jako całkę, więc bardzo wiele przykładów stosowania metody Monte Carlo pasuje do takich skojarzeń.

W sytuacji gdy mówimy o klasycznym zagadnieniu z analizy matematycznej, to znaczy obliczaniu

$$\int_A h(x)\,dx,$$

gdzie A jest podzbiorem $\mathbb{R}^d$ a $h\colon \mathbb{R}^d \to \mathbb{R}$, ogólna metoda polega na znalezieniu jakiegoś rozkładu o dodatniej gęstości $f(\cdot)$ na A i obliczeniu

$$E\Big(h^*(X)/f(X)\Big),$$

gdzie X jest zmienną losową o rozkładzie z gęstością $f(\cdot)$, natomiast funkcja h^* ma być zgodna z $h(\cdot)$ w zbiorze A, a poza nim ma się zerować, to znaczy

$$h^*(x) = \begin{cases} h(x), & \text{dla } x \in A, \\ 0, & \text{dla } x \in \mathbb{R}^d \setminus A. \end{cases}$$

Dobór gęstości może być dokonany stosownie do sytuacji, na przykład metodą losowania istotnościowego, jak w podrozdziale 4.4, lub w pewien uniwersalny, prosty sposób, który w przypadku ograniczonych

zbiorów A może sprowadzać się do losowania jednostajnego z nadzbioru A.

Tego rodzaju postępowanie było istotą przykładów 4.1 (str. 67), 4.3 (str. 72), 4.8 (str. 86) oraz kilku zaproponowanych zadań: 1.1, 1.2, 4.1, 4.4, 4.5, 4.6, 4.7, 4.8.

Oczywiście dla funkcji o nieskomplikowanym przebiegu, mających małe wahanie, ciągłych poza być może kilkoma punktami, jest wiele skutecznych metod numerycznych obliczania całek oznaczonych, metod, w których nie ma mowy o losowaniu. Ale nawet wtedy metoda Monte Carlo – ze względu na swoją prostotę – może posłużyć na przykład do testowania, zgrubnego (lub dokładnego, ale kosztownego czasowo) sprawdzenia. Stosunkowo prosta funkcja, która sprawia pewien kłopot standardowym metodom numerycznym, a z którą łatwo sobie radzi uniwersalna metoda Monte Carlo omówiona była w przykładzie 4.8 na stronie 86.

Liczenie długości krzywych, pól powierzchni, objętości i innych tego rodzaju miar często daje się sprowadzić do całek oznaczonych. W szczególności łatwo liczy się pole podzbioru $\mathbb{R}^2$ lub objętość podzbioru $\mathbb{R}^3$, gdy są to obszary normalne, a więc takie, które leżą pomiędzy wykresami dwu funkcji o wspólnej dziedzinie. Wtedy miara takiego obszaru jest całką wartości bezwzględnej różnicy tych ograniczających funkcji.

W sytuacji gdy obszar ten nie jest normalny, można dzielić go na części, w których jest normalny, być może uprzednio dokonując zamiany zmiennych. Gdyby jednak w miarę łatwo dawało się stwierdzać, czy dany punkt z przestrzeni należy czy też nie do tego obszaru i gdyby obszar ten był podzbiorem pewnego ograniczonego zbioru o znanej mierze, to wystarczyłoby z tego ograniczającego nadzbioru losować jednostajnie punkty i szacować frakcję punktów trafiających w wyróżniony obszar. Oszacowanie frakcji pomnożone przez miarę ograniczającego nadzbioru będzie oszacowaniem miary interesującego nas obszaru.

Tego rodzaju postępowanie omawiane było w przykładach 1.1 (str. 13), 4.2 (str. 69) oraz zadaniach 1.3 i 4.3.

5.2. Badanie własności rozkładów zmiennych losowych

Metoda Monte Carlo jest metodą obliczeniową ściśle opartą na statystyce i rachunku prawdopodobieństwa. Jednocześnie te dziedziny matematyki potrzebują metody Monte Carlo, by obliczać wartości, wobec których metody analityczne bądź też klasyczne (nie używające liczb pseudolosowych) metody numeryczne wydają się bezradne. Na przykład dla wielu testów statystycznych ich wartości krytyczne są wyznaczane właśnie metodą Monte Carlo.

Przykład 5.1. *Test Kołmogorowa–Lillieforsa jest testem służącym do sprawdzania hipotezy o normalności rozkładu. Mając próbkę m uporządkowanych niemalejąco wartości $x_1, x_2, \dots, x_m$ należy obliczyć średnią z próby $\bar{x}$, odchylenie standardowe próby $\hat{s}$ (pierwiastek z wartości nieobciążonego estymatora wariancji), obliczyć liczby*

$$F_i = \Phi\left(\frac{x_i - \bar{x}}{\hat{s}}\right), \quad d_i = \max\{|F_i - (i-1)/m|, |F_i - i/m|\},$$

dla $i = 1, 2, \dots m$ oraz obliczyć statystykę testową

$$k_m = \max\{d_1, d_2, \dots, d_m\}.$$

Duże wartości statystyki testowej świadczą oczywiście przeciw hipotezie zerowej, którą jest normalność rozkładu. Według hipotezy alternatywnej badany rozkład nie jest rozkładem normalnym. Hipoteza zerowa nie precyzuje parametrów rozkładu, jedynie mówi o rodzaju rozkładu.

Skąd wziąć wartości krytyczne dla takiego testu? Analitycznie wydaje się to bardzo trudne, natomiast obliczenie ich metodą Monte Carlo jest niezwykle proste i wynika wprost z koncepcji testów istotności. Mianowicie ustalmy liczbę m (liczebność próby) oraz poziom istotności testu α (mała liczba dodatnia będąca prawdopodobieństwem mylnego odrzucenia hipotezy zerowej) i jakiekolwiek konkretne parametry dla rozkładu normalnego, może to być nawet $\mu = 0$, $\sigma = 1$. Następnie n-krotnie (n to liczba prób dla podstawowej metody Monte Carlo) wykonujemy eksperyment polegający na wygenerowaniu m próbek z rozkładu $N(\mu, \sigma)$ i obliczeniu statystyki testowej. Mając n takich wartości, obliczamy ich kwantyl rzędu $1 - \alpha$ i ta wartość będzie przybliżeniem szukanej wartości krytycznej.

W taki sposób zostały te wartości wygenerowane i opublikowane w oryginalnej pracy *On the Kolmogorov–Smirnov test for normality with mean and variance unknown* [22] wyjaśniającej koncepcję tego testu. Jej autor pisze, że liczba n za każdym razem była równa co najmniej 1000, bez sprecyzowania, jak dużo więcej niż 1000 mogło to być. Można się domyślać, że niewiele więcej, biorąc pod uwagę fakt, że praca jest z roku 1967. W przytoczonej publikacji podane są wartości krytyczne dla

$$\alpha \in \{0.01, 0.05, 0.10, 0.15, 0.20\}, \quad m \in \{4, 5, 6 \ldots, 30\}. \tag{5.1}$$

Dla wartości $m > 30$ podane zostały proste formuły aproksymujące wartości krytyczne, osobne dla każdego z pięciu wskazanych wyżej poziomów istotności.

W pracy z 2017 roku [25] przeliczono ponownie wartości krytyczne tą metodą, tym razem w każdym przypadku biorąc $n = 10^5$ i podając wyniki zaokrąglone do czterech cyfr po kropce. Jednak nie ma tam żadnego uzasadnienia, dlaczego taka liczba n miałaby dać pewność akurat tylu cyfr dokładności.

Poniżej opisany został przykład procedury, która w takim przypadku da racjonalne podstawy do oceny dokładności obliczeń. Można wygenerować wyniki testu Kołmogorowa–Lillieforsa (przy ustalonym m) dla dużej liczby próbek, powiedzmy $n = 10^7$, następnie dla uzyskanego rozkładu empirycznego należy wyznaczyć przedział ufności – na wysokim poziomie ufności, powiedzmy 99% – dla kwantyli rzędu $1-\alpha$, biorąc α jak w formule (5.1). To, jak wyznaczać przedziały ufności dla kwantyli, stanowi osobne zagadnienie statystyczne, dla uproszczenia użyto (dla wygenerowanych w `C++` danych) funkcji `QuantileCI` z pakietu `DescTools` środowiska `R` [28].

Przyjmijmy, że $m = 25$ oraz poziom istotności $\alpha \in \{0.01, 0.05\}$. W tabeli 5.1 pokazano w pierwszym wierszu oszacowania wartości krytycznych z pracy [22], w wierszu drugim – poprawione oszacowania z pracy [25], natomiast w trzecim i czwartym pokazano 99% przedziały ufności dla wartości krytycznych, uzyskane opisanym wyżej sposobem na podstawie $n = 10^5$ prób oraz $n = 10^7$ prób. Jasno widać, że cztery cyfry w pracy [25] były przejawem nieuzasadnionego optymizmu.

..... *koniec przykładu 5.1* .....

Tab. 5.1. Tabela wartości krytycznych testu Kołmogorowa–Lillieforsa dla rozmiaru próby równego 25 (przykład 5.1); dwa ostatnie wiersze to 99% przedziały ufności dla szacowanej wielkości

obliczenia	$\alpha = 0.01$	$\alpha = 0.05$
praca [22], $n \geqslant 10^3$	0.203	0.180
praca [25], $n = 10^5$	0.2010	0.1726
skrypt, $n = 10^5$	$(0.199\,182, 0.201\,895)$	$(0.172\,088, 0.173\,448)$
skrypt, $n = 10^7$	$(0.201\,189, 0.201\,450)$	$(0.172\,925, 0.173\,063)$

5.3. Badanie własności procesów stochastycznych

Procesem stochastycznym $\{S(t)\}_{t\in T}$ nazywamy rodzinę zmiennych losowych, to znaczy przyporządkowanie każdemu elementowi zbioru indeksów T zmiennej losowej $S(t)$ określonej w pewnej przestrzeni probabilistycznej $(\Omega, \mathcal{F}, \Pr)$. Przyjmiemy, że T jest zbiorem liczb rzeczywistych dodatnich, wygodną interpretacją T jest w takim przypadku czas. Zmienne losowe, które są wartościami procesu stochastycznego, będziemy traktowali jako rzeczywiste jednowymiarowe zmienne losowe, chociaż oczywiście w innych obszarach zastosowań używa się bardziej ogólnego podejścia.

Dla każdego $t \in T$ wartość $S(t)$ to zmienna losowa, a więc formalnie funkcja, której dziedziną jest Ω. W sytuacji, gdy warto będzie podkreślić, że chodzi o konkretną realizację wartości procesu, może być użyty zapis $S(t, \omega)$. Na przykład `trajektorią` procesu $\{S(t)\}_{t\in T}$ nazywamy każdą funkcję $f\colon T \to \mathbb{R}$, dla której istnieje $\omega \in \Omega$, że

$$f(t) = S(t, \omega), \quad \text{dla } t \in T.$$

5.3.1. Zagadnienia inżynierii finansowej prowadzące do stochastycznych równań różniczkowych i symulacji trajektorii

Typowymi przykładami zagadnień, w modelowaniu których może pojawić się proces stochastyczny, a także określające go równanie, są ceny akcji, ceny instrumentów finansowych pochodnych, wielkości stóp procentowych.

Cena akcji w każdej przyszłej chwili jest modelowana zmienną losową, mamy więc zależną od czasu rodzinę zmiennych losowych,

która stanowi proces stochastyczny. Proces ten ma właściwości, które w pewnym uproszczeniu wyrażone są odpowiednim równaniem. W równaniu tym, obok wartości procesu, istotną rolę odgrywa także przyrost wartości procesu w krótkim czasie, znajdzie się tam także pewien czynnik nieprzewidywalny, losowy – stąd w naturalny sposób dostaniemy stochastyczne równanie różniczkowe. Następnie na bazie ceny instrumentu podstawowego obliczana jest wartość instrumentu pochodnego (opcja zakupu lub sprzedaży akcji). Rozwiązanie analityczne może być trudne – przybliżone rozwiązanie można uzyskać metodą Monte Carlo.

Przykład 5.2. *Cena $S(t)$ papieru bazowego (akcji) w chwili $t \geqslant 0$ może być modelowana równaniem*

$$dS(t) = \mu S(t)\, dt + \sigma S(t) dW(t), \tag{5.2}$$

gdzie lewa strona oznacza przyrost ceny, μ i σ są pewnymi stałymi parametrami, natomiast $dW(t)$ jest przyrostem procesu Wienera, na temat którego więcej szczegółów pojawi się w podrozdziale 5.3.2. Składnik związany z $dW(t)$ koncentruje w sobie całą losowość prawej strony równania (5.2), gdyby tej części nie było, to równanie sprowadziłoby się do prostego równania różniczkowego, którego rozwiązaniem byłaby funkcja wykładnicza (zakładając jedną wartość w chwili $t = 0$).

Rozwiązawszy równanie (5.2) można następnie np. szacować wartość opcji związanych z tymi akcjami bądź rozważać różnego typu strategie portfelowe.

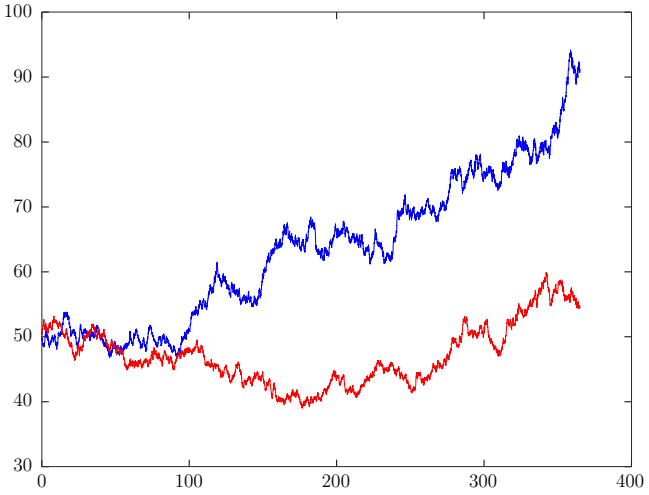
Rysunek 5.1 pokazuje przykładowe dwie trajektorie procesu (5.2), dla konkretnych wartości parametrów równania i wspólnej wartości początkowej.

Przykład 5.3. *Model Vasička zakłada, że intensywność oprocentowania $R(t)$ podlega prawu*

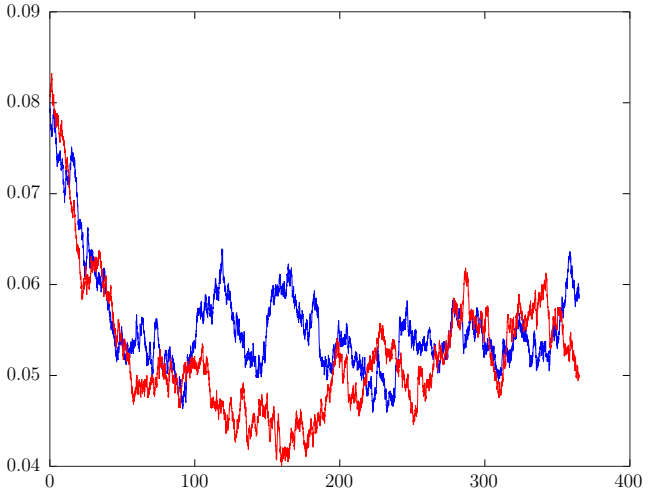
$$dR(t) = a(b - R(t))\, dt + \sigma dW(t), \tag{5.3}$$

gdzie parametr b jest długoterminową średnią, natomiast mnożnik a determinuje szybkość zbliżania się procesu do średniej, σ wyznacza zmienność części losowej prawej strony równania.

Na rysunku 5.2 można znaleźć przykłady dwu trajektorii, które pokazują, jak realizowana jest „skłonność" do zbliżania się do owej długoterminowej średniej.



Rys. 5.1. Dwie trajektorie procesu (5.2), rysunek wygenerowany dla $\mu = 0.2, \sigma = 0.3$



Rys. 5.2. Dwie trajektorie rozwiązania równania (5.3). Rysunek wygenerowany dla $a = 10.5$, $b = 0.05, \sigma = 0.025$

5.3.2. Proces Wienera

U podstaw modelowania, którego przykłady pojawiły się w poprzednim podrozdziale, leży proces zwany procesu Wienera.

Proces stochastyczny o wartościach rzeczywistych $\{W(t)\}_{t\in[0,\infty)}$ nazywany jest procesem Wienera (lub jednowymiarowym ruchem Browna), jeżeli

- $W(0) = 0$ z prawdopodobieństwem 1,
- $W(t) - W(s)$ ma rozkład $N(0, \sqrt{t-s})$, tzn. rozkład normalny ze średnią 0 i wariancją $t - s$,
- dla dowolnego układu $0 < t_1 < t_2 < \ldots < t_n$ zmienne losowe $W(t_1), W(t_2) - W(t_1), \ldots, W(t_n) - W(t_{n-1})$ są niezależne.

Powszechność tego procesu w rozważaniach z wielu działów matematyki, fizyki, techniki czy finansów może być tłumaczona centralnymi twierdzeniami granicznymi.

Własności trajektorii procesu Wienera

Poniższe twierdzenie – którego dowód można znaleźć na przykład w [8] – implikuje, że z prawdopodobieństwem 1 trajektorie procesu Wienera są ciągłe.

Twierdzenie 5.1. *Niech $\{W(t)\}_{t\in[0,\infty)}$ będzie procesem Wienera oraz niech $u > 0$. Wtedy dla dowolnego $0 < \gamma < 0.5$ i dla prawie wszystkich $\omega \in \Omega$ istnieje takie $K > 0$, że dla dowolnych $s, t \in [0, u]$ zachodzi*

$$|W(t,\omega) - W(s,\omega)| \leqslant K|t-s|^{\gamma}.$$

Jednak kolejne twierdzenie [8] pokazuje, że trajektorie, gdy im się dobrze przyjrzymy (pomyślmy na przykład o powiększaniu, skalowaniu wykresu trajektorii) mają własności mocno odbiegające od funkcji, którymi zajmujemy się w klasycznej analizie rzeczywistej.

Twierdzenie 5.2. *Niech $\{W(t)\}_{t\in[0,\infty)}$ będzie procesem Wienera. Wtedy dla prawie wszystkich $\omega \in \Omega$ trajektoria $t \mapsto W(t,\omega)$ jest w każdym punkcie nieróżniczkowalna.*

Ścisły dowód nie jest w tym skrypcie potrzebny, a jako nieformalne uzasadnienie wskażmy, jaki jest dla dowolnego $t > 0$ rozkład ilorazu różnicowego (co wyprowadza się natychmiast z określenia procesu Wienera i z ogólnych własności wariancji), mianowicie

$$\frac{W(t_0 + \Delta t) - W(t_0)}{\Delta t} \sim N\left(0, \frac{1}{\sqrt{\Delta t}}\right).$$

Zatem przy $\Delta t \to 0$ zmienność ilorazu różnicowego rośnie do nieskończoności. To daje intuicyjne wyjaśnienie braku różniczkowalności trajektorii oraz coraz bardziej stromych przebiegów trajektorii w miarę kolejnego przybliżania (skalowania) rysunku.

Ilustracją tego faktu niech będą trzy kolejne wykresy (wszystkie umieszczone na rys. 5.3), z których pierwszy pokazuje przykładową trajektorię $W(\cdot, \omega)$ w przedziale $[0, 1]$ z zaznaczonym punktem

$$\Big(0.5, W(0.5, \omega)\Big),$$

drugi i trzeci to 10- i 100-krotne przybliżenie otoczenia wyróżnionego punktu (skala przybliżenia dotyczy osi czasu).

5.3.3. Stochastyczna całka Itô

Problemem w określeniu całki typu

$$\int_0^T F(t, W(t)) dW(t) \tag{5.4}$$

jest to, że próbując sumowania wartości na trajektoriach, napotyka się problemy nieznane w klasycznej analizie – trajektoria nie ma nigdzie pochodnej i ma nieograniczone wahanie. Konstrukcja takiej całki (której wynikiem dla ustalonego T jest zmienna losowa) polega na tworzeniu sum Riemanna na trajektoriach przy właściwym wyborze podziałów i punktów reprezentujących podprzedziały (w przypadku całki Itô – lewe końce).

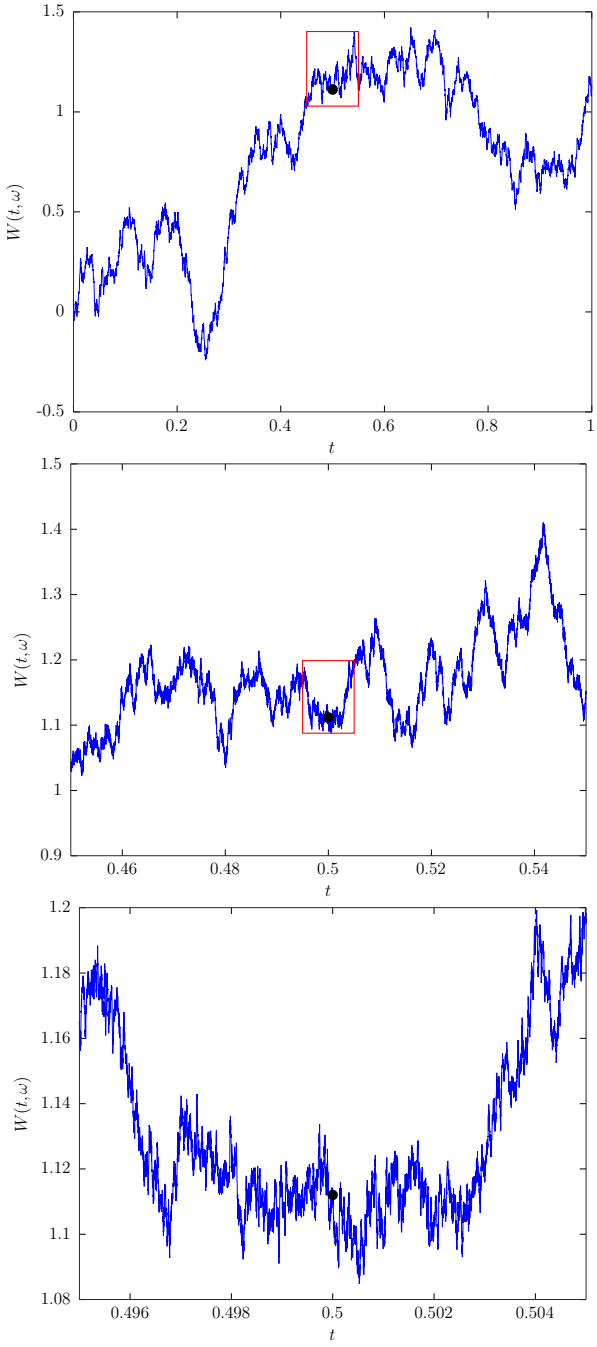
Na przykład, zgodnie z intuicją, mamy [8]

$$\int_0^T dW(t) = W(T), \tag{5.5}$$

ale (to już może mniej intuicyjne)

$$\int_0^T W(t)\, dW(t) = \frac{W^2(T)}{2} - \frac{T}{2}. \tag{5.6}$$

Gdyby przy określaniu sum Riemanna brać pod uwagę środki przedziałów, to składnik $-\frac{T}{2}$ we wzorze (5.6) zniknie.



Rys. 5.3. Wybrana trajektoria procesu Wienera w trzech skalach; ramka otaczająca wyróżniony punkt na pierwszych dwu wykresach zakreśla fragment trajektorii powiększany na kolejnym wykresie

Tab. 5.2. Porównanie wartości obliczonych analitycznie i pochodzących z obliczeń metodą Monte Carlo dla całek (5.5) i (5.6), $T = 10$

całka	wartość oczekiwana		odch. standardowe	
	teoret.	metoda MC	teoret.	metoda MC
całka (5.5)	0	$-0.004\,273\,87$	$\sqrt{10} \approx 3.1623$	3.1621
całka (5.6)	0	$-0.000\,413\,01$	$5\sqrt{2} \approx 7.0711$	7.0684

5.3.4. Obliczanie całek stochastycznych metodą Monte Carlo

Wynik całki (5.4) dla ustalonego T jet zmienną losową, której własności (dystrybuantę, momenty i.t.p.) można łatwo wyliczać metodą Monte Carlo, nawet podstawową, szacującą wartości za pomocą średniej z próbek.

W najprostszym przypadku należy wygenerować n trajektorii procesu Wienera obliczonych w równo oddalonych punktach

$$0 = t_0 < t_1 < ... < t_{m-1}, \quad \text{gdzie } t_{i+1} - t_i = T/m.$$

Jeżeli $w_j^{(i)}$ oznacza wartość i-tej trajektorii procesu Wienera w punkcie t_j, to na przykład wartość oczekiwaną całki (5.4) można aproksymować wielkością

$$\frac{1}{n} \sum_{i=1}^{n} \sum_{j=0}^{m-1} F(t_j, w_j^{(i)})(w_{j+1}^{(i)} - w_j^{(i)}),$$

$w_0^{(i)} = 0$, natomiast przyrosty $w_{j+1}^{(i)} - w_j^{(i)}$ należy generować jako liczby pseudolosowe z rozkładu $N(0, 1)$ pomnożone przez $\sqrt{t_{j+1} - t_j}$.

Przykład obliczeń dla $T = 10, n = 10^6, m = 10^4$ przedstawiono w tabeli 5.2.

W związku z subtelnościami związanymi z definicją całki Itô, trzeba także mocno uważać przy symulacjach związanych z tą całką. Na przykład gdyby przy obliczaniu całki (5.6) brać „prawego” reprezentanta przedziału do sumy całkowej (zamiast „lewego”), to zamiast liczby $-0.000\,413\,01$ w tabeli 5.2 pojawiłaby się wartość $9.999\,31$, a więc różnica między obliczonymi różnymi technikami wartościami oczekiwanymi zmiennej losowej znacznie przekroczyłaby nawet jej odchylenie standardowe. Czyli wynik ten byłby kompletnie bezużyteczny!

5.3.5. Stochastyczne równania różniczkowe

Przez rozwiązanie równania w postaci

$$\begin{cases} dX = f(X,t)\,dt + g(X,t)\,dW, \\ X(0) = X_0, \end{cases}$$

dla $0 \leqslant t \leqslant T$, rozumiemy proces stochastyczny $X(\cdot)$, spełniający dla dowolnego $0 \leqslant t \leqslant T$ równość

$$X(t) = X_0 + \int_0^t f(X(s),s)\,ds + \int_0^t g(X(s),s)\,dW(s)$$

prawie pewnie.

Szczegółowe założenia odnośnie do funkcji i procesów występujących powyżej zostały pominięte – nie są one potrzebne do wykonywania obliczeń metodą Monte Carlo w zakresie niniejszego skryptu.

Niektóre równania tego rodzaju dają się rozwiązać analitycznie. Na przykład wspomniane wcześniej równanie (5.3) ma rozwiązanie

$$R(t) = R(0)e^{-at} + b(1-e^{-at}) + \sigma \int_0^t e^{-a(t-s)} dW(s)\,ds,$$

skąd można wywnioskować, że dla $0 \leqslant s < t$ warunkowy rozkład $R(t)|R(s)$ jest normalny ze średnią

$$R(s)e^{-a(t-s)} + b(1-e^{-a(t-s)})$$

i wariancją

$$\frac{\sigma^2}{2a}(1-e^{-2a(t-s)}).$$

5.3.6. Wycena opcji europejskich w modelu Blacka–Scholesa

Europejska opcja kupna z ceną wykonania K i o chwili wykupu T, jest to opcja, która po upływie czasu T daje prawo zakupu akcji po cenie K. Zatem jej wartość w chwili T wynosi

$$\max(S(T) - K, 0),$$

gdzie $S(t)$ jest wartością papieru bazowego w chwili t.

W modelu Blacka–Scholesa – przy założeniu braku możliwości arbitrażu, stałej intensywności oprocentowania, możliwości swobodnego lokowania przy ustalonej intensywności oraz możliwości swobodnego zakupu i sprzedaży akcji – cena takiej opcji w chwili 0 wynosi

$$\begin{aligned} S(0) \cdot \Phi \left(\frac{\ln \frac{S(0)}{K} + \left(r + \frac{\sigma^2}{2} \right) T}{\sigma \sqrt{T}} \right) - \\ - K e^{-rT} \Phi \left(\frac{\ln \frac{S(0)}{K} + \left(r - \frac{\sigma^2}{2} \right) T}{\sigma \sqrt{T}} \right), \end{aligned} \tag{5.7}$$

gdzie r jest obowiązującą intensywnością oprocentowania, zaś σ jest zmiennością akcji (szacowaną poprzez odchylenie standardowe).

Europejska opcja sprzedaży z ceną wykonania K i o chwili wykupu T, jest to opcja, która po upływie czasu T daje prawo sprzedaży akcji po cenie K. Zatem jej wartość w chwili T wynosi

$$\max(K - S(T), 0),$$

gdzie $S(t)$ jest wartością papieru bazowego w chwili t.

W modelu Blacka–Scholesa – przy założeniu braku możliwości arbitrażu, stałej intensywności oprocentowania, możliwości swobodnego lokowania przy ustalonej intensywności oraz możliwości swobodnego zakupu i sprzedaży akcji – cena takiej opcji w chwili 0 wynosi

$$\begin{aligned} K e^{-rT} \Phi \left(\frac{-\ln \frac{S(0)}{K} - \left(r - \frac{\sigma^2}{2} \right) T}{\sigma \sqrt{T}} \right) - \\ - S(0) \cdot \Phi \left(\frac{-\ln \frac{S(0)}{K} - \left(r + \frac{\sigma^2}{2} \right) T}{\sigma \sqrt{T}} \right). \end{aligned}$$

Dzięki temu prostemu modelowi i powyższym bezpośrednim wzorom można łatwo testować skuteczność metody Monte Carlo (a także poprawność wyprowadzeń tych wzorów). Przykładowy program porównujący obliczenia podstawową metodą Monte Carlo z wynikami według formuły (5.7) przedstawiony jest na listingu 5.1.

```cpp
#include <iostream>
#include <random>
#include <cmath>
double rnorm() {
    static std::mt19937_64 gen;
    static std::normal_distribution<double> norm(0,1);
    return norm(gen);
}
int main() {
    // parametry dla opcji
    const double T = 3./12.;  // czas wykupu
    const double K = 12;      // cena wykupu
    const double r=0.05;      // oprocentowanie
    const double sigma = 2.5; // zmienność
    const double S0 = 10;     // cena początkowa akcji
    // parametry dla metody Monte Carlo
    const int n = 10'000'000; // ile wartości dla uśredniania
    const int m = 500;        // liczba podprzedziałów przy
                              // symulacji
    // ------------------------------------------------
    double Sum = 0;
    for (int i=0; i<n; ++i) {
        if (i%1000==0)
            std::cout << i << "\n";
        // jedna ścieżka
        double S = S0; //w S liczę końcową cenę akcji;
        double dt = T/m; double sdt = sqrt(dt);
        for (int j=0; j<m; ++j) {
            double dW = rnorm()*sdt;
            S += S*(r*dt + sigma*dW);
        }
        Sum += std::max( 0., S-K);
    }
    double sr = Sum/n;
    std::cout << "oszacowanie ceny = " << sr*exp(-T*r) << "\n";
}
```

Listing 5.1. Metoda Monte Carlo dla europejskich opcji kupna

Na przykład dla $K = 12$, $T = 3/12$, $S_0 = 10$, $r = 0.05$, $\sigma = 2.5$ cena opcji kupna wyznaczona po uśrednieniu 10^7 trajektorii (z których każda była obliczona w 500 punktach) podstawową metodą Monte Carlo wyniosła $4.230\,32$. Tymczasem cena ze wzoru (5.7) to $4.229\,14$.

Po co robić symulacje rozwoju cen akcji, skoro równanie (5.2) można rozwiązać analitycznie? Chodzi właśnie o to, by program, taki jak opisany wyżej, mógł zostać dopasowany do zupełnie innej sytuacji niż tylko ta wymagająca końcowej ceny akcji. Modyfikacja programu w ten sposób, by obliczał oszacowania cen opcji w innych, trudniejszych analitycznie przypadkach, jest stosunkowo prosta. Może to dotyczyć na przykład opcji amerykańskich (dowolny moment realizacji prawa zakupu lub sprzedaży), opcji azjatyckich (zależnych od średniej wartości ceny aktywa bazowego w ustalonym okresie w przyszłości), opcji z progiem (przekroczenie progu w wyznaczonym odcinku czasowym powoduje uzyskanie premii).

5.3.7. Most Browna

Czasem, przy generowaniu trajektorii procesu Wienera, pojawia się potrzeba generowania trajektorii o znanym początku i końcu. Taka potrzeba może wystąpić na przykład przy losowaniu warstwowym trajektorii procesu stochastycznego. Kanonicznym przypadkiem są trajektorie procesu Wienera uwarunkowane zależnością $W(1) = 0$. Taki proces, tzn. $(W(t)|W(1) = 0)$, nazywany jest mostem Browna.

Kilka trajektorii tego procesu zostało przedstawionych na rysunku 5.4.

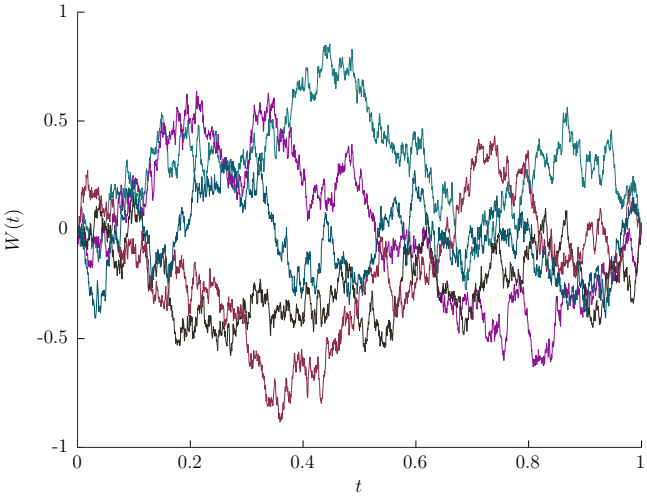
Można udowodnić [12], że jeżeli $W(t)$ jest procesem Wienera, to

$$B(t) = W(t) - t \cdot W(1)$$

jest mostem Browna. Podobnie, jeżeli $B(t)$ jest mostem Browna, to

$$W(t) = B(t) + t \cdot Z$$

jest procesem Wienera, gdzie Z jest niezależną od $B(t)$ zmienną losową o rozkładzie $N(0, 1)$.



Rys. 5.4. Pięć przykładowych trajektorii mostu Browna

Twierdzenie 5.3. *Niech $W(t)$ będzie procesem Wienera i niech $0 \leqslant t_a < t < t_b$. Wtedy $W(t)$, przy warunku $W(t_a) = w_a$, $W(t_b) = w_b$, ma rozkład normalny ze średnią*

$$\frac{(t_b - t)w_a + (t - t_a)w_b}{t_b - t_a}$$

i wariancją

$$\frac{(t - t_a)(t_b - t)}{t_b - t_a}.$$

Dowód można znaleźć np. w [12]. Dla nas istotne jest praktyczne znaczenie: gdy znamy dwa punkty trajektorii procesu Wienera i chcemy je uzupełnić punktami „pomiędzy", to właśnie takiego uzupełnienia można dokonać, losując z rozkładu normalnego o parametrach podanych w twierdzeniu 5.3.

5.3.8. Losowanie warstwowe końcowe przy symulacji trajektorii

Jeżeli do jakiejś symulacji potrzebujemy obliczania wielu trajektorii procesu Wienera, to jakość losowania może być podniesiona poprzez zapewnienie „sprawiedliwego" losowania. Mianowicie, jeśli końcową chwilą symulacji jest T, to wiadomo, że $W(T) \sim N(0, \sqrt{T})$. Jeżeli zbiór liczb rzeczywistych podzielimy na kilka podzbiorów o jednakowych

prawdopodobieństwach dla tego rozkładu, to sensowne wydaje się, by liczba wartości końcowych trajektorii, które należą do każdego z tych zbiorów, była podobna.

Przykład 5.4. *Na rysunku 5.5 przedstawiono jedenaście przypadkowo wylosowanych trajektorii procesu Wienera. Rzuca się w oczy brak symetrii osiowej względem osi odciętych, który jest oczywiście dziełem przypadku.*

Poniżej przedstawiony jest algorytm, który losuje k trajektorii procesu Wienera, i-ta trajektoria kończy się w i-tym podzbiorze $\mathbb{R}$, $i = 0, 1, \ldots k-1$, przy czym każdy z tych podzbiorów jest jednakowo prawdopodobny według rozkładu $N(0, \sqrt{T})$, każda trajektoria reprezentowana jest przez $m+1$ punktów odpowiadających chwilom $0 = t_0 < t_1 < t_2 < \ldots < t_m = T$.

Trajektorie procesu Wienera z użyciem mostu Browna

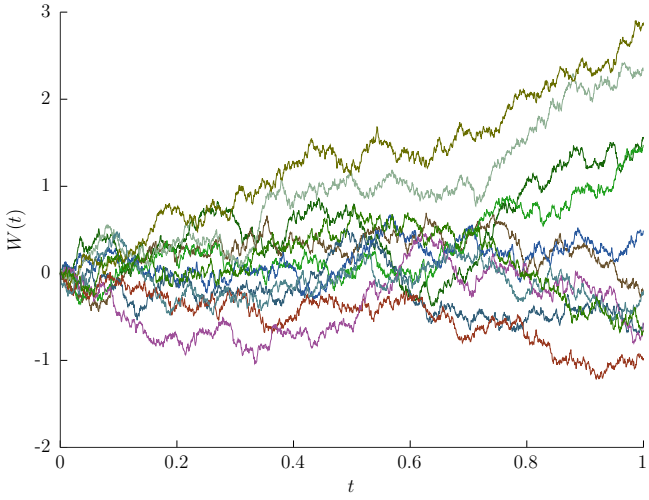
Każde spośród m zakończeń trajektorii wylosowane jest z jednego z m rozłącznych zbiorów o jednakowym prawdopodobieństwie względem rozkładu $W(T)$.

---

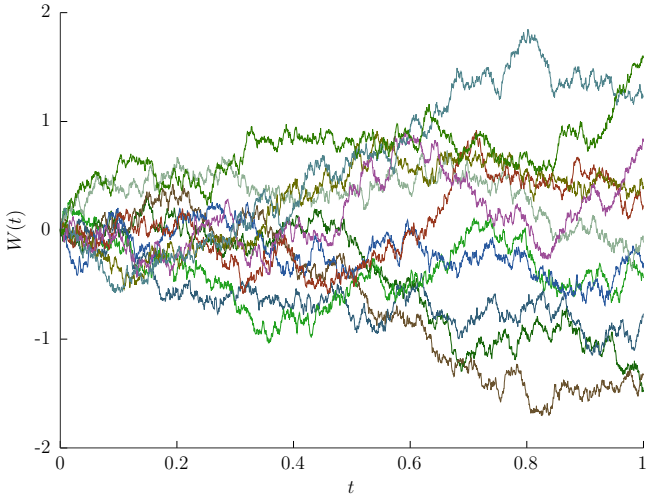
dla $i = 0, 1, \ldots \ k-1$
  wygeneruj U z rozkł. jednostajnego na $[0,1)$
  $v = (i+U)/k$ // $v \in [\ i/k,\ (i+1)/k)$
  $W(t_m) = \sqrt{t_m}\ \Phi^{-1}(v)$
  dla $j = 1, 2, \ldots, \ m-1$
    generuj Z z rozkładu $N(0,1)$
    $W(t_j) = (t_m - t_j)/(t_m - t_{j-1})\ W(t_{j-1}) +$
    $+(t_j - t_{j-1})/(t_m - t_{j-1})\ W(t_m) +$
    $+\sqrt{((t_m - t_j)(t_j - t_{j-1}))/(t_m - t_{j-1})}Z$

Rysunek 5.6 pokazuje 11 trajektorii wylosowanych przy użyciu tego właśnie algorytmu. Rzeczywiście ten rysunek wygląda na bardziej „wyrównany”, a więc można spodziewać się mniejszej wariancji wyników symulacji.

..... *koniec przykładu 5.4* .....



Rys. 5.5. Jedenaście trajektorii procesu Wienera



Rys. 5.6. Jedenaście trajektorii procesu Wienera z losowaniem warstwowym końcowym, jak w przykładzie 5.4

5.4. Próbkowanie Monte Carlo łańcuchami Markowa

Grupa metod znana pod nazwą „próbkowanie Monte Carlo łańcuchami Markowa" opiera się na konstruowaniu łańcuchów Markowa, na podstawie których można otrzymywać próbki z interesującego nas rozkładu. Łańcuchy Markowa to procesy stochastyczne z czasem dyskretnym,

dla których charakterystyczne jest to, że prawdopodobieństwa uzyskiwania poszczególnych stanów w kolejnym kroku zależą tylko od stanu procesu w kroku bezpośrednio poprzedzającym rozważany moment.

Metody te szczególnie przydatne okazały się w przybliżaniu numerycznym całek wielowymiarowych, umożliwiając koncentrację obliczeń na najważniejszych punktach (o największej gęstości interesującego nas rozkładu).

W grupie tych metod wyróżnia się – ze względu na bogatą historię i czytelny opis – algorytm Metropolisa–Hastingsa.

5.4.1. Algorytm Metropolisa–Hastingsa

Algorytm, znany jako algorytm Metropolisa, został opisany w 1953 roku [24], później go ulepszono [15] i jego zmodyfikowana wersja to właśnie algorytm Metropolisa–Hastingsa.

Załóżmy, że mamy pewien wielowymiarowy rozkład o danej funkcji gęstości $f\colon \mathbb{R}^d \to \mathbb{R}$. Jak się okaże w algorytmie – wystarczy znajomość $f(\cdot)$ z dokładnością do stałej, gdyż w algorytmie ta funkcja występuje jednocześnie w liczniku i mianowniku pewnego ułamka.

Potrzebna jest także funkcja błądzenia losowego $q(y|x)$, która oblicza gęstość prawdopodobieństwa położenia kolejnego punktu y, gdy w poprzednim kroku przyjęty został punkt x. To może być na przykład gęstość rozkładu wielowymiarowego normalnego z wartością oczekiwaną x i wariancją (macierzą wariancji i kowariancji) przyjętą stosownie do rozważanego zagadnienia. Najprościej jest przyjąć, że $q(y|x)$ jest gęstością d-wymiarowego rozkładu normalnego z nieskorelowanymi współrzędnymi i średnią x. Wariancje dla poszczególnych współrzędnych można dopracować eksperymentalnie, umieszczając na początku algorytmu dodatkową pętlę, która wypróbuje różne wariancje i dobierze właściwe wartości (być może różne dla różnych współrzędnych). Zbyt małe wariancje $q(y|x)$ są niekorzystne, bo oznaczają, że stany łańcucha będą się po badanej przestrzeni przemieszczały zbyt wolno. Za duża wariancja z kolei może oznaczać wielką liczbę przejść w regiony z niską gęstością $f(\cdot)$, co będzie powodować niewielką frakcję akceptowanych próbek. Można się spotkać z zaleceniami dobierania funkcji $q(y|x)$ tak, by frakcja akceptowanych punktów wynosiła

około 30%. W opisie algorytmu, dla zachowania przejrzystości zakładamy, że funkcja $q(y|x)$ jest dana.

Oto przebieg algorytmu [12]:

Algorytm Metropolisa–Hastingsa

Wejście: funkcja gęstości $f(\cdot)$, funkcja błądzenia losowego $q(y|x)$, punkt początkowy x_0.

---

dla t = 1, 2, …, n
    wylosuj y z rozkładu $q(\cdot|x_{t-1})$;
    wylosuj u rozkładu jednostajnego na $[0,1]$
    jeżeli $u < \frac{f(y)q(x_{t-1}|y)}{f(x_{t-1})q(y|x_{t-1})}$
        $x_t = y$
    w przeciwnym razie
        $x_t = x_{t-1}$

W praktyce – oprócz wspomnianej fazy dobrania właściwych parametrów dla $q(y|x)$ – początkową część otrzymywanych puntów odrzuca się, by szybciej uniezależnić uzyskany zbiór punktów od stanu początkowego x_0.

Zauważmy jeszcze, że na przykład dla funkcji błądzenia losowego $q(y|x)$, będącej gęstością rozkładu normalnego o średniej x, mamy oczywiście $q(x|y) = q(y|x)$. Zatem funkcja $q(\cdot|\cdot)$ znika z nierówności w opisanym wyżej algorytmie, dając czytelną interpretację: jeżeli gęstość w punkcie y jest większa niż w punkcie x_{t-1}, to na pewno y zostanie zaakceptowany, jeśli natomiast gęstość w nowo wygenerowanym punkcie jest mniejsza niż w poprzednim, to prawdopodobieństwo jego przyjęcia jest równe stosunkowi $\frac{f(y)}{f(x_{t-1})}$.

Jak uzasadnić poprawność tego algorytmu? Po pierwsze, z ogólnych własności łańcuchów Markowa wynika, że jeśli łańcuch Markowa jest odwracalny względem pewnej miary, to miara ta jest miarą stacjonarną. Odwracalność łańcucha względem miary oznacza, że równie prawdopodobne jest bycie w chwili t w stanie ze zbioru A (to zależy od tej właśnie miary) i przejście do stanu ze zbioru B (to już związane jest z „działaniem” łańcucha), co bycie w chwili t w stanie ze zbioru B i przejście do stanu ze zbioru A w chwili następnej. I tak ma być dla wszystkich możliwych par mierzalnych zbiorów A, B.

Z algorytmu Metropolisa–Hastingsa wynika, że dla ustalonego x, gęstość przejścia od stanu x do stanu y (dla $x \neq y$) pomnożona przez $f(x)$ wynosi

$$f(x)q(y|x) \cdot \min\left(\frac{f(y)q(x|y)}{f(x)q(y|x)}, 1\right)$$

i łatwo sprawdzić, że wartość ta jest symetryczna ze względu na x i y.

Zatem miara zadana rozkładem $f(\cdot)$ jest dla łańcucha tworzonego algorytmem Metropolisa–Hastingsa miarą stacjonarną.

Stacjonarność rozkładu $f(\cdot)$ implikuje, że wychodząc z dowolnego stanu – o ile stany łańcucha komunikują się ze sobą – rozkład łańcucha w chwili n zbliża się do właśnie tego rozkładu stacjonarnego, gdy $n \to \infty$.

Komunikowanie się ze sobą stanów łańcucha łatwo zapewnimy dobierając funkcję $q(\cdot|\cdot)$, która się nigdzie nie zeruje, na przykład opartą o rozkład normalny.

Zadania

Zadanie 5.1. Napisz funkcję przyjmującą jako parametry: wielkość próby n, funkcję $h\colon \mathbb{R}^2 \to \mathbb{R}$, funkcję $\psi\colon \mathbb{R}^2 \to \{0, 1\}$ oraz liczby rzeczywiste x_1, x_2, y_1, y_2. Jej wynikiem jest oszacowanie podstawową metodą Monte Carlo całki $\iint_D h(x, y)\, dxdy$, gdzie obszar D to zbiór tych punktów, dla których funkcja $\psi(\cdot, \cdot)$ daje wynik 1 i który jest zawarty w prostokącie $[x_1, x_2] \times [y_1, y_2]$. Wykonaj test dla

$$D = \{(x, y) \in \mathbb{R}^2 \colon x < y < 3x, \tfrac{1}{x} < y < \tfrac{5}{x}, x > 0, y > 0\},$$
$$h(x, y) = \tfrac{y}{x}, \qquad [x_1, x_2] \times [y_1, y_2] = [0.5, 2.5] \times [1, 4].$$

Zadanie 5.2. Powróćmy do przykładu 5.1. Dla $m = 30$ i $\alpha = 0.05$ praca [25] podaje (na podstawie 10^5 prób) oszacowanie wartości krytycznej testu Kołmogorowa–Lillieforsa równe 0.1590. W oryginalnej pracy Lillieforsa [22] oszacowanie to wynosiło 0.161, na podstawie dużo mniejszej mniejszej liczby prób. Zweryfikuj te wartości, obliczając oszacowanie punktowe na podstawie 10^7 prób.

Zadanie 5.3. Zmodyfikuj program z listingu 5.1, tak by działał dla europejskiej opcji sprzedaży.

Zadanie 5.4. Zmodyfikuj rozwiązanie zadania 5.3, tak aby dotyczyło azjatyckiej opcji sprzedaży. Opcja ta w momencie wykupu T będzie miała wartość $\max(K - A, 0)$, gdzie K jest ustaloną z góry ceną wykonania, natomiast A to średnia z całego okresu od nabycia opcji do wykonania.

Zadanie 5.5. Napisz funkcję w C++ realizującą algorytm Metropolisa–Hastingsa, która przyjmuje następujące parametry:

- punkt początkowy x_0,
- funkcję $f(\cdot)$ proporcjonalną do gęstości rozkładu,
- funkcję $h(\cdot)$ do liczenia wartości oczekiwanej $Ef(X)$, gdzie X pochodzi z rozkładu $c \cdot f(X)$ dla nieznanej wartości stałej normalizującej c,
- n – liczbę próbek, które posłużą do oszacowania średniej,
- m – liczbę prób, na podstawie których oceniona zostanie jedna para odchyleń standardowych dla funkcji błądzenia losowego,
- k – liczbę próbek do pominięcia po fazie „dostrojenia" funkcji błądzenia losowego,
- S – wektor proponowanych odchyleń standardowych, które trzeba wypróbować w fazie ustalania funkcji błądzenia losowego.

Funkcja błądzenia losowego ma być budowana (dla każdej współrzędnej) na podstawie rozkładu normalnego, parametry mają być tak dobierane, by wyznaczyć parę parametrów dającą frakcję akceptacji jak najbliższą 30%.

Wypróbuj działanie, przyjmując funkcję $h(\cdot)$ z zadania 5.1, funkcję $f(\cdot)$ będącą funkcją charakterystyczną zbioru D z zadania 5.1, $n = 10^8$, $m = 10^4$, $k = 10^5$ i wektor możliwych odchyleń standardowych S postaci

$$S = [2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 2^0, 2^1, 2^2].$$

Odpowiedzi, wskazówki

Zadanie 5.1. Dla przykładu testowego wynik dokładny to 4.

Zadanie 5.5. Dla danych testowych wynik dokładny to $\frac{2}{\ln 3}$. Dla funkcji błądzenia losowego najlepszy wybór z proponowanych to $\sigma_x = 0.25$, $\sigma_y = 2$.

Bibliografia

1. Aho A.V., Ullman J.D. i Hopcroft J.E., Projektowanie i analiza algorytmów, Helion 2003.
2. Badger L., Lazzarini's lucky approximation of π, Mathematics Magazine, 67, http://www.jstor.org/stable/2690682 (dostęp 20. 04. 2025), 83–91, 1994.
3. Bartos J., Królikowska K., Wasilewski M., Dyczka W. i Krysicki W., Rachunek prawdopodobieństwa i statystyka matematyczna w zadaniach. Część 1: rachunek prawdopodobieństwa, Wydawnictwo Naukowe PWN 2004.
4. Bartos J., Królikowska K., Wasilewski M., Dyczka W. i Krysicki W., Rachunek prawdopodobieństwa i statystyka matematyczna w zadaniach. Część 2: statystyka matematyczna, Wydawnictwo Naukowe PWN 2004.
5. Box G.E.P. i Muller M.E., A note on the generation of random normal deviates, The Annals of Mathematical Statistics, 29, DOI: 10.1214/aoms/1177706645, 610–611, 1958.
6. Brown L.D., Cai T.T. i DasGupta A., Interval estimation for a binomial proportion, Statistical Science, 16, 101–133, 2001.
7. Cornfeld I., Fomin S. i Sinai Y.G., Ergodic Theory, Springer-Verlag 1982.
8. Evans L.C., An introduction to stochastic differentail equations, American Mathematical Society 2013.
9. Fichtenholz G.M., Rachuek różniczkowy i całkowy, Tom I, Wydawnictwo Naukowe PWN 2007.
10. Fortuna Z., Macukow B. i Wąsowski J., Metody numeryczne, Wydawnictwo Naukowe PWN 2017.
11. Gajek L. i Kałuszka M., Wnioskowanie statystyczne, WNT 1996.
12. Glasserman P., Monte Carlo Methods in Financial Engineering, DOI: 10.1007/978-0-387-21617-1, Springer 2013.
13. Gutterman Z. i Pinkas B., Analysis of the linux random number generator, http://www.pinkas.net/PAPERS/gpr06.pdf (dostęp 20. 04. 2025).
14. Halton J.H., On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals, Numerische Mathematik, 2, 84–90, 1960.
15. Hastings W.K., Monte carlo sampling methods using markov chains and their applications, Biometrica, 57, 97–109, 1970.
16. Jakubowski J. i Sztencel R., Rachunek prawdopodobieństwa dla (prawie) każdego, Script 2017.
17. Klein A., Linear Feedback Shift Registers, w: Klein A., Stream Ciphers, DOI: 10.1007%2F978-1-4471-5079-4_2, Springer 2013.

18. Knuth D.E., Sztuka programowania. Tom 2: Algorytmy seminumeryczne, WNT 2002.
19. Laplace P.S. de, Théorie analityque des probabilités, 1812.
20. Lazzarini M., Un'applicazione del calcolo della probabilità alla ricerca sperimentale di un valore approssimato di π, Periodico di Matematica per l'Insegnamento Secondario, 4, 140–143, 1901.
21. Lehmer D., Mathematical methods in large-scale computing units, Proceedings of a Second Symposium on large-scale digital calculating machinery, 141–146, 1949.
22. Lilliefors H.W., On the Kolmogorov–Smirnov test for normality with mean and variance unknown, Journal of the American Statistical Association, 62, DOI: 10.1080/01621459.1967.10482916, 399–402, 1967.
23. Marsaglia G. i Bray T.A., A convenient method for generating normal variables, SIAM Review, 6, DOI: 10.1137/1006063, 260–264, 1964.
24. Metropolis N., Rosenbluth A.W., Rosenbluth M.N., Teller A.H. i Teller E., Equation of state calculations by fast computing machines, Journal of Chemical Physics, 21, 1087–1092, 1953.
25. Molin P. i Abdi H., New tables and numerical approximation for the Kolmogorov–Smirnov/Lillierfors/van Soest test of normality, Technical report, Univ. of Bourgogne, https://personal.utdallas.edu/~herve/MolinAbdi1998-LillieforsTechReport.pdf, 1998.
26. Niederreiter H., Random number generation and quasi-Monte Carlo methods, Society for Industrial i Applied Mathematics 1992.
27. Sierpiński W., Teoria liczb, Wydawnictwo Naukowe PWN 1959.
28. Signorell A., Desctools: tools for descriptive statistics, https://CRAN.R-project.org/package=DescTools, R package version 0.99.55, 2024.
29. Thomson W.E., A modified congruence method of generating pseudo--random numbers, The Computer Journal, 1, DOI: 10.1007/978-1-4471-5079-4_2, 1958.
30. Wallis S., Binomial confidence intervals and contingency tests: mathematical fundamentals and the evaluation of alternative methods, Journal of Quantitative Linguistics, 20, DOI: 10.1080/09296174.2013.799918, 178–208, 2013.
31. Zieliński R., Przedział ufności dla frakcji, Matematyka Stosowana, 37, DOI: 10.14708/ma.v37i51/10.265, 2009.
32. Zieliński R., Metody Monte Carlo, WNT 1970.

Indeks terminów