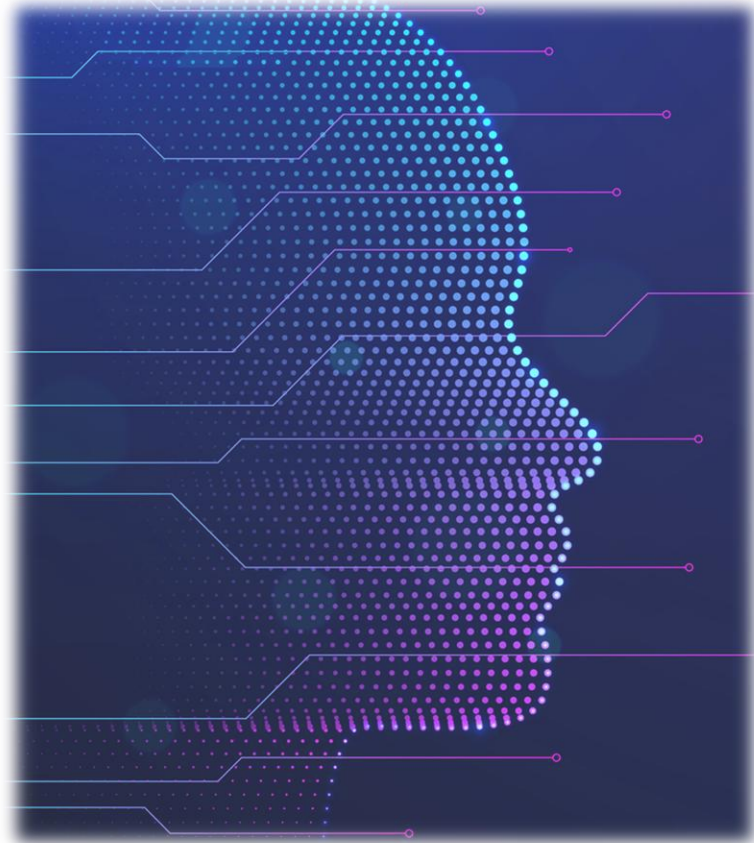


Inteligencja obliczeniowa w analizie danych



**Algorytmy
„czysto”
probabilistyczne**

Gdzie jesteśmy

Algorytmy heurystyczne

Algorytmy probabilistyczne

Algorytmy genetyczne

Strategie ewolucyjne

Metody roju cząstek

Logika rozmyta

Rozmyte systemy wnioskujące

Sterowanie rozmyte

Sztuczne sieci neuronowe

**Filary inteligencji
obliczeniowej**

Algorytmy
probabilistyczne

Obliczenia
ewolucyjne

Logika
rozmyta

Sztuczne
sieci neuronowe

Algorytm heurystyczny

Algorytm heurystyczny, heurystyka – algorytm niedający (w ogólnym przypadku) gwarancji znalezienia rozwiązania optymalnego, umożliwiający jednak znalezienie rozwiązania dość dobrego w rozsądnym czasie.

Algorytmy „czysto” probabilistyczne

Algorytm probabilistyczny (czasem stochastyczny) albo randomizowany.. to algorytm, który do swojego działania używa losowości.

W praktyce oznacza to, że implementacja takiego algorytmu korzysta przy obliczeniach z generatora liczb losowych.

Główną zaletą algorytmów probabilistycznych w porównaniu z deterministycznymi jest działanie zawsze w „średnim przypadku”, dzięki czemu „złośliwe” dane wejściowe nie wydłużają jego działania.

Symulacje stochastyczne są prostym sposobem badania zjawisk losowych.

Ściśle związane z symulacjami stochastycznymi są metody obliczeniowe nazywane „Monte Carlo” (MC).

Polegają one na wykorzystaniu „sztucznie generowanej” losowości w celu rozwiązania zadań deterministycznych.

Metody Monte Carlo są proste i skuteczne.

Dla pewnych problemów MC jest jedynym z niewielu dostępnych narzędziem obliczeniowym.

Dla innych problemów MC jest co prawda mniej efektywne od metod numerycznych, ale za to dużo łatwiejsze!

W przeciwieństwie do metod numerycznych, gdzie wypracowane algorytmy pozwalają kontrolować deterministycznie błędy, w przypadku obliczeń za pomocą metod stochastycznych dostajemy wynik losowy, i jest ważne aby zrozumieć jak taki wynik należy interpretować..

U podstaw metod MC leży prawo wielkich liczb ...

Istnieje wiele wariantów praw wielkich liczb.

Jeśli S_n jest liczbą sukcesów w n próbach Bernoulliego ..
prawdopodobieństwo sukcesu p polega na otrzymaniu w n próbach S_n sukcesów..
to dla każdego $\varepsilon > 0$

$$P\left(\left|\frac{S_n}{n} - p\right| \leq \varepsilon\right) \rightarrow 1 \quad (n \rightarrow \infty)$$

(prościej - niezależnie od wyboru szerokości przedziału wokół wartości oczekiwanej, prawdopodobieństwo dla dużych n będzie dowolnie bliskie 1)

Intuicyjne przykłady :

Liczba orłów przy dużej liczbie rzutów stabilizuje się wokół 0.5,

a liczba jedynek w rzutach kostką powinna wynosić 1/6 wszystkich rzutów.

**Największy potencjalny problem w metodzie MC
(jak i wszystkich heurystykach)
wynika z generatora liczb losowych...**

Komputer tak naprawdę nie losuje liczb ..

**Możemy generować ciągi liczbowe,
które mają z góry ustalonych wartości..**

**Musimy dbać, aby generowane liczby bardzo
dobrze imitowały ciągi liczb losowych.**

**Pierwotnie generowane są
rozkłady jednostajne.**

**Prawidłowy termin określający liczby losowe
generowane programowo, to:
liczby pseudolosowe.**

Czym są generator liczb losowych

Generator liczb losowych (ang. random number generator, RNG) – program komputerowy lub układ elektroniczny generujący losowy ciąg elementów binarnych zorganizowanych zwykle jako ciąg liczb losowych.

Oczekiwanym wynikiem działania generatora liczb losowych zwykle są **liczby z przedziału $[0,1]$ o rozkładzie jednostajnym** – ale, z generatora o takim rozkładzie można uzyskać generator o dowolnym innym rozkładzie obliczając odwrotną dystrybuantę pożądanego rozkładu z wyników pierwszego generatora.

Ze względu na sposób generowania liczb losowych można wyróżnić **dwa rodzaje generatorów**:

Generatory sprzętowe (ang. TRNG – True Random Number Generator) – działające na zasadzie obrazowania właściwości i parametrów fizycznego procesu stochastycznego, najczęściej szumu elektrycznego. Generatory takie nie produkują samych liczb, lecz stany, które dopiero później interpretowane są jako liczby.

Generatory programowe
(ang. PRNG – Pseudo Random Number Generator)
– działające na zasadzie deterministycznego obliczania ciągu liczb, które wyglądają na losowe.

Zasadniczymi zaletami generatorów sprzętowych, szczególnie ważną w kryptografii, są nieprzewidywalność i niereprodukowalność generowanych ciągów, wynikające z unikatowości realizacji fizycznego procesu stochastycznego w danym przedziale czasu.

Czym są generator liczb losowych

Tak więc liczby pochodzące z generatora programowego zwane są **liczbami pseudolosowymi**, ponieważ faktycznie nie są dziełem przypadku, lecz wynikiem obliczeń matematycznych.

Największą **zaletą generatorów pseudolosowych jest ich szybkość i dostępność.**

Należy jednak zwrócić uwagę na fakt, że znając wartości podawane na wejście generatora oraz jego stan wewnętrzny bez trudu można przewidzieć zwracane przezeń liczby.

Z tego powodu decydując się na zastosowanie w systemie kryptograficznym generatora liczb pseudolosowych należy zachować szczególną ostrożność przy doborze zarówno samego algorytmu, jak i sposobu inicjowania oraz rodzaju wartości podawanych na jego wejście.

.. jak programowo generujemy liczby pseudolosowe...

podstawowe informacje

Komputer potrafi tylko generować ciągi liczb naturalnych.

Punkt startowy to ziarno losowania.

Można „losować” ziarno –
brane jest wtedy z czasu
systemowego.

systemy zgodne z Uniksem
koduja czas systemowy
jako liczbę sekund, które
upłynęły od 1 stycznia
1970 roku 00:00:00



Stworzenie z nich rozkładu
jednostajnego jest proste – dzielimy
modulo.

Złe dobranie ziarna, albo modulo
może prowadzić szybko do
okresowości algorytmu.

***Ręczne ustawienie ziarna jest
niezbędne do replikowania
symulacji i obliczeń.***

Uzyskanie z rozkładu jednostajnego innych rozkładów uzyskuje
się najczęściej poprzez funkcję odwrotną do dystrybuanty.

.. jak programowo generujemy liczby pseudolosowe...

pierwsze próby

Ciekawostka: do tworzenia ciągu liczb losowych jako pierwszy wykorzystał komputer John von Neumann około 1946 r.

Zaproponował on aby tworzyć następną liczbę podnosząc do kwadratu poprzednią i wycięciu środkowych cyfr.

“Metoda środka kwadratu” okazała się niezbyt dobra; bowiem ma tendencje do wpadania w koleinę – krótki cykl powtarzających się elementów. Na przykład ciąg dwucyfrowych liczb zaczynających się od **43** według tego algorytmu jest
84, 05, 02, 00, . . .

przykład $X_i =$

|| **3967** ||

$X_i^2 =$

|| 15|**7370**|89| || ||

$X_{i+1} =$

|| **7370** ||

$X_{i+1}^2 =$

|| 54|**3169**|00 || ||

$X_{i+2} =$

|| **3169** ||

$X_{i+2}^2 =$

|| 10|**0425**|61| || ||

$X_{i+3} =$

|| **0425** ||

$X_{i+3}^2 =$

|| 00|**1806**|25 || ||

$X_{i+4} =$

$X_{i+4}^2 =$

|| **1806** ||

|| 03|**2616**|36 || ||

$$\mathbf{43}^2 = \mathbf{1849}, \mathbf{84}^2 = \mathbf{7056}, \mathbf{05}^2 = \mathbf{0025}, \mathbf{02}^2 = \mathbf{0004}, 00^2 = \mathbf{0000}$$

Zróbmy to inaczej ...

Generator liniowy i afiniczny

Powtórzmy nasze marzenie - generujemy ciąg liczb całkowitych z przedziału $[0;M]$ tak aby wszystkie liczby występowały z jednakowym prawdopodobieństwem oraz by częstotliwość występowania liczb z każdego z podprzedziałów tego przedziału była w przybliżeniu jednakowa w czasie.

Do osiągnięcia tych celów do dziś najczęściej wykorzystywanym jest generator liniowy. Kolejne liczby losowe są obliczane przy wykorzystaniu wzoru rekurencyjnego:

$$X_{n+1} = (a \cdot X_n) \bmod M$$

Parametry: warunek początkowy (X_0), współczynnik a oraz M , które definiuje zakres maksymalny uzyskiwanych liczb. Zarówno X_0 jak i a muszą być z przedziału $[1;M-1]$.

Co się może wydarzyć ..

...gdy któryś z wyrazów osiągnie wartość 0 (czyli wynikiem iloczynu wyrazu poprzedniego i wartości a będzie wartość równa wartości M), generator przestanie tworzyć kolejne liczby !

Aby temu zapobiec, zmodyfikowano wzór generatora liniowego, który nazwano **generatorem afinicznym**:

$$X_{n+1} = (a \cdot X_n + b) \bmod M, \quad \text{gdzie } 0 < a, b < M$$

$$X_0 = 0$$

$$X_1 = (5 \times X_0 + 1) \bmod 16 = 1 \bmod 16 = 1$$

$$X_2 = (5 \times X_1 + 1) \bmod 16 = 6 \bmod 16 = 6$$

$$X_3 = (5 \times X_2 + 1) \bmod 16 = 31 \bmod 16 = 15$$

$$X_4 = (5 \times X_3 + 1) \bmod 16 = 76 \bmod 16 = 12$$

$$X_5 = (5 \times X_4 + 1) \bmod 16 = 61 \bmod 16 = 13$$

$$X_6 = (5 \times X_5 + 1) \bmod 16 = 66 \bmod 16 = 2$$

$$X_7 = (5 \times X_6 + 1) \bmod 16 = 11 \bmod 16 = 11$$

$$X_8 = (5 \times X_7 + 1) \bmod 16 = 56 \bmod 16 = 8$$

$$X_9 = (5 \times X_8 + 1) \bmod 16 = 41 \bmod 16 = 9$$

$$X_{10} = (5 \times X_9 + 1) \bmod 16 = 46 \bmod 16 = 14$$

$$X_{11} = (5 \times X_{10} + 1) \bmod 16 = 71 \bmod 16 = 7$$

$$X_{12} = (5 \times X_{11} + 1) \bmod 16 = 36 \bmod 16 = 4$$

$$X_{13} = (5 \times X_{12} + 1) \bmod 16 = 21 \bmod 16 = 5$$

$$X_{14} = (5 \times X_{13} + 1) \bmod 16 = 26 \bmod 16 = 10$$

$$X_{15} = (5 \times X_{14} + 1) \bmod 16 = 51 \bmod 16 = 3$$

$$X_{16} = (5 \times X_{15} + 1) \bmod 16 = 16 \bmod 16 = 0$$

Okres obu
przedstawionych
generatorów zależy
od wartości
parametrów, i może
osiągnąć M

Przykładowy wynik
działania generatora
afinicznego z parametrami
 $a=5$, $b=1$ i $M=16$.

Parametry kilku generatorów liniowych o maksymalnych okresach: $X_n = (a X_{n-1} + c) \bmod m$

a	c	m	Nazwa/autor
$2^{16}+3$	1	2^{31}	RANDU (IBM360/370)
$2^2 \cdot 2^{37} + 1$	1	2^{35}	Zieliński (1966)
69069	1	2^{32}	Marsaglia (1972)
16807	1	$2^{31}-1$	Park, Miller (1980)
40692	1	$2^{31}-249$	L'Ecuyer (1988)
68909602460261	1	2^{48}	Fishman (1990)

m – liczba pierwsza \Rightarrow na ogół lepsze własności statystyczne generatora.

Wypracowano kilka reguł doboru parametrów, aby uzyskiwać maksymalne okresy generatorów.

Okres $\sim 2^{32} \approx 4 \cdot 10^9$

Inne propozycje

Generator wielomianowy (kwadratowy +)

Z uwagi na dużą przewidywalność generatorów liniowych i afinicznych prowadzono poszukiwania generatorów bez tego mankamentu.

Jedną z propozycji jest generator wielomianowy

$$X_{n+1} = \sum_{i=1}^k a_i X_n^i \bmod(M)$$

Okres generatorów zależy od wartości parametrów może osiągnąć M

Inne propozycje

Generator inwersyjny

Generatory inwersyjne pojawiły się jako efekt poszukiwań takich przekształceń, które generują ciąg wyrazów (liczb) nie cechujący się lokalną liniowością.

Generator inwersyjny także zalicza się do grupy generatorów o rozkładzie równomiernym, lecz w przeciwieństwie do liniowego, otrzymywane liczby są nieliniowe.

$$X_{n+1} = \begin{cases} (aX_n^{-1} + b) & \text{dla } X_n > 0 \\ b & \text{dla } X_n = 0 \end{cases}$$

Gdzie X^{-1} oznacza inwersję dzielenia modulo, którą liczy się ze wzoru

$$X^{-1} = c, \quad \text{gdzie } (c \cdot X) \bmod M = 1$$

Liczba M musi być liczbą pierwsza.

Maksymalny okres takiego generatora, przy odpowiednim doborze współczynników a i b może być równy $M-1$.

Przykładowy ciąg generatora inwersyjnego dla $a=5$, $b=1$ i $M=17$.

$$X_0 = 0$$

$$X_1 = b = 1$$

$$X^{-1} = c,$$

gdzie

$$(c \cdot X) \bmod M = 1$$

$$\text{bo } (1 \cdot 1) \bmod 17 = 1$$

$$X_2 = (5 \times 1 + 1) = 6$$

$$\text{bo } (3 \cdot 6) \bmod 17 = 1$$

$$X_3 = (5 \times 3 + 1) = 16$$

$$\text{bo } (16 \cdot 16) \bmod 17 = 1$$

$$X_4 = (5 \times 16 + 1) \quad 81 \bmod 17 = 13$$

$$X_5 = (5 \times 4 + 1) \quad 21 \bmod 17 = 4$$

$$X_6 = (5 \times 13 + 1) \quad 66 \bmod 17 = 15$$

$$X_7 = (5 \times 8 + 1) \quad 41 \bmod 17 = 7$$

$$X_8 = (5 \times 5 + 1) \quad 26 \bmod 17 = 9$$

$$X_9 = (5 \times 2 + 1) \quad 11 \bmod 17 = 11$$

$$X_{10} = (5 \times 14 + 1) \quad 71 \bmod 17 = 3$$

$$X_{11} = (5 \times 6 + 1) \quad 31 \bmod 17 = 14$$

$$X_{12} = (5 \times 11 + 1) \quad 56 \bmod 17 = 5$$

$$X_{13} = (5 \times 7 + 1) \quad 36 \bmod 17 = 2$$

$$X_{14} = (5 \times 9 + 1) \quad 46 \bmod 17 = 12$$

$$X_{15} = (5 \times 10 + 1) \quad \bmod 17 = 51 \bmod 17 = 0$$

$$X_{n+1} = \begin{cases} (aX_n^{-1} + b) & \text{dla } X_n > 0 \\ b & \text{dla } X_n = 0 \end{cases}$$

X^{-1} oznacza inwersję dzielenia modulo, którą liczy się ze wzoru:

$$X^{-1} = c, \quad \text{gdzie } (c \cdot X) \bmod M = 1$$

Maksymalny okres takiego generatora, przy odpowiednim doborze współczynników a i b może być równy $M-1$

Przykład konfigurowania generatora w MATLABie

W MATLABie jest zaimplementowanych kilka generatorów, np. : 'state', 'twister'...

Jeśli nic nie zostanie zadeklarowane, to jest w użyciu 'state' z ziarnem losowym.


W przeciwnym razie metodę musimy zadeklarować wpisując komendę `rand(method,s)`, gdzie w 'method' należy wpisać jedną z metod np. 'state', 'twister'.

Dobrze ocenianą alternatywą jest algorytm Mersenne Twister, który opiera się na liniowej rekurencji.

Najczęściej używana wersja algorytmu Mersenne Twister jest oparta na liczbie pierwszej Mersenne'a – okres algorytmu wynosi: $2^{31}-1$.

```
rand('twister', sum(100*clock));
```

Jest to algorytm o dużej szybkości, a jego jakość jest na tyle wysoka, że nadaje się do większości zastosowań, takich jak symulacje czy gry komputerowe.

W Python także najczęściej używamy generatora Mersenne Twister  `random()`

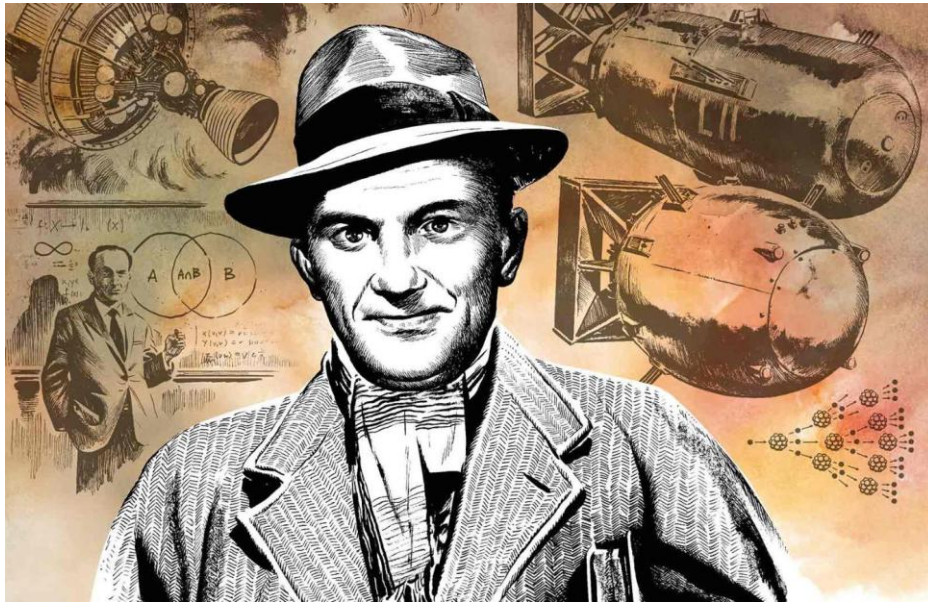
Bardziej zaawansowany moduł `secrets` opiera się na systemowych funkcjach losowości . Te funkcje generują liczby losowe, które są trudniejsze do przewidzenia, ponieważ korzystają z danych, takich jak czas, aktywność systemu czy inne czynniki, które są zmienne i trudne do odtworzenia.

Metoda Monte Carlo

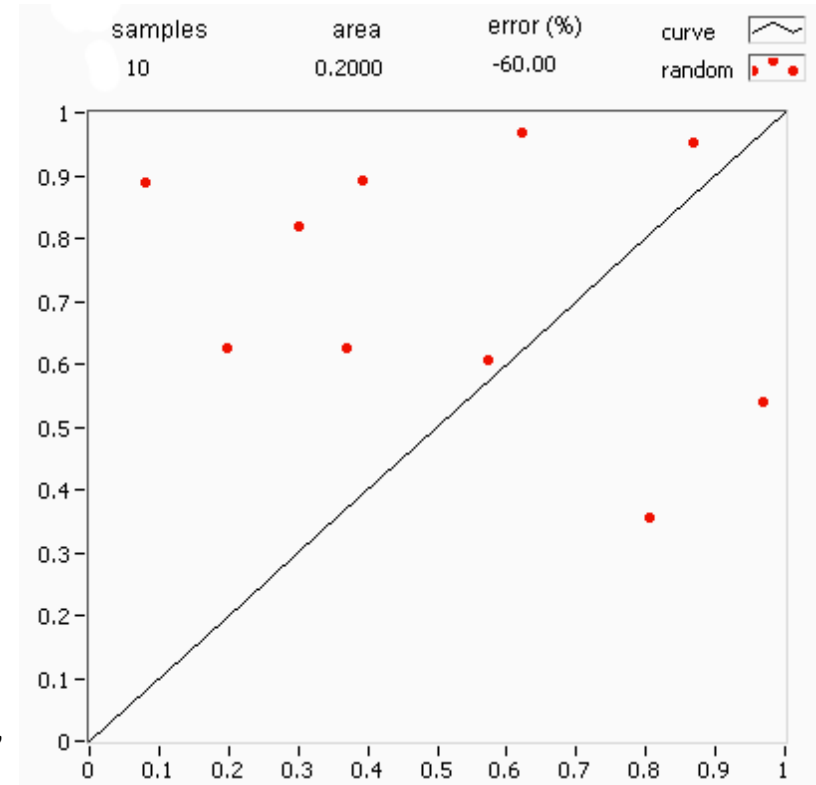
Metoda Monte Carlo jest często wybierana w trakcie symulowania wielu procesów.

Metodę opracował i po raz pierwszy zastosował polski matematyk Stanisław Ulam we współpracy z Johnem von Neumannem (współtwórcy bomby wodorowej).

Często za pomocą Monte Carlo realizuje się całkowanie numeryczne.

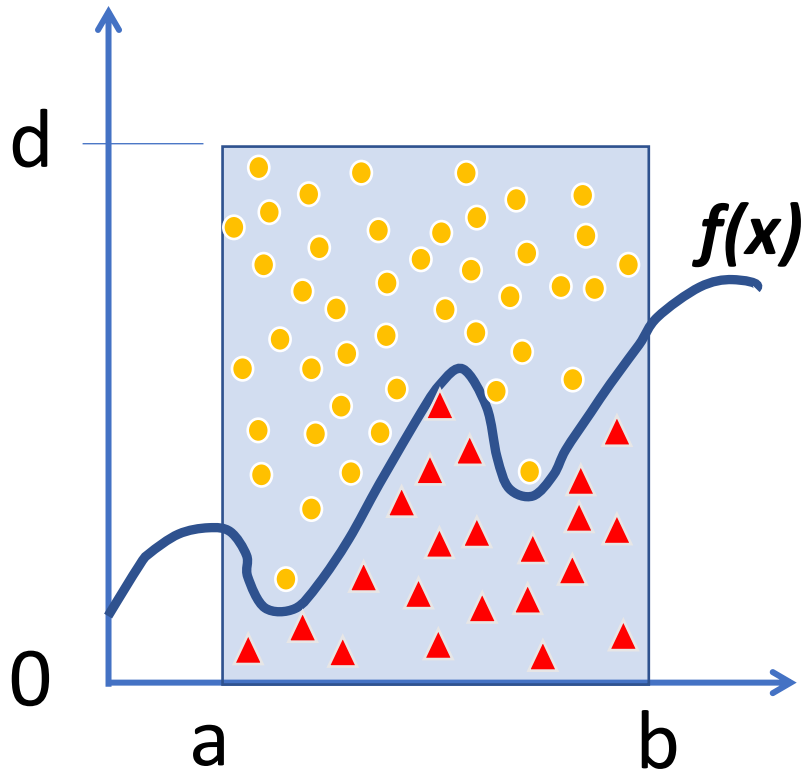


Zamiast rozwiązywać całkę numerycznie, stosujemy najprostszy algorytm heurystyczny ..



Schemat typowego Monte Carlo (metoda akcepracji/odrzuceń)

- Losujemy N punktów z obszaru ograniczonego prostokątem $[a,b] \times [0,d]$ (lub rozwinięciem wielowymiarowym)
- Sprawdzamy, ile (k) punktów leży pod wykresem



$k = 0$ i $n = 0$.

Wylosuj x i y z rozkładu jednostajnego $[a, b]$

$$x = a + (b-a) \cdot \text{rand}()$$

Analogicznie dla y z zakresu $[0, d]$

$$y = d \cdot \text{rand}()$$

Sprawdzamy czy $y < f(x)$

Jeżeli tak $k++$;

Powtórz N razy .. (N powinno być bardzo duże)

Podziel k/n

Pomnóż przez długość przedziału całkowania $[a, b]$ i wysokość analizowanego zakresu $[0, d]$

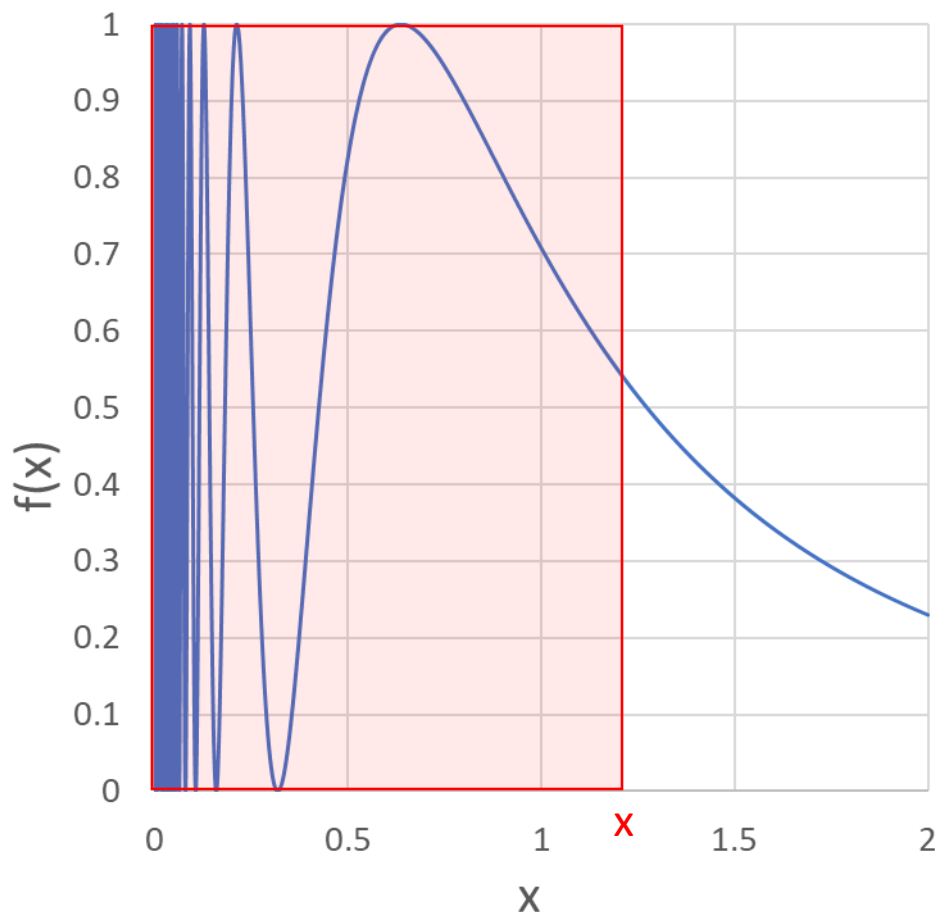
$$\int_a^b f(x) dx \approx \frac{k}{N} (b-a)d$$

● ▲

Konkretny przykład

Funkcja $f(x)$, całki z $f(x)$ nie da się rozwiązać analitycznie

$$f(x) = \sin^2\left(\frac{1}{x}\right) \qquad I(x) = \int_0^x \sin^2\left(\frac{1}{x}\right) dx$$



Ale całka, to pole powierzchni pod wykresem funkcji ..

Prawdopodobieństwo P , że punkt (a,b) wylosowany z zakresu analizowanego prostokąta, leży poniżej krzywej $f(x)$ wynosi:

$$P(\downarrow) = \frac{I(x)}{\text{pole } \square} = \frac{I(x)}{x \cdot 1}$$

$$I(x) = x \cdot P(\downarrow)$$

$$P(\downarrow) = \lim_{N \rightarrow \infty} \frac{k}{N} \quad \begin{array}{l} \text{ile pkt } \downarrow \\ \text{ile prób} \end{array}$$

Monte Carlo – całkowanie numeryczne

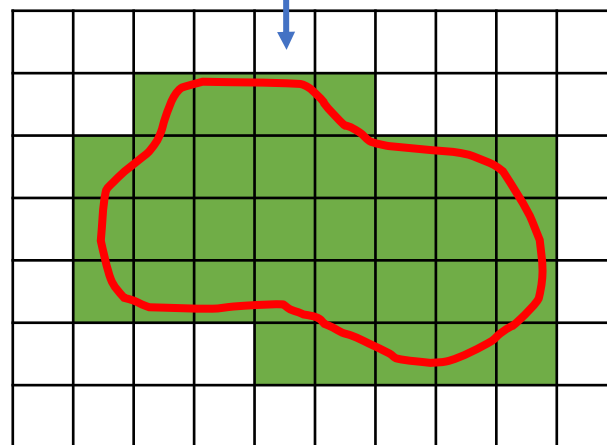
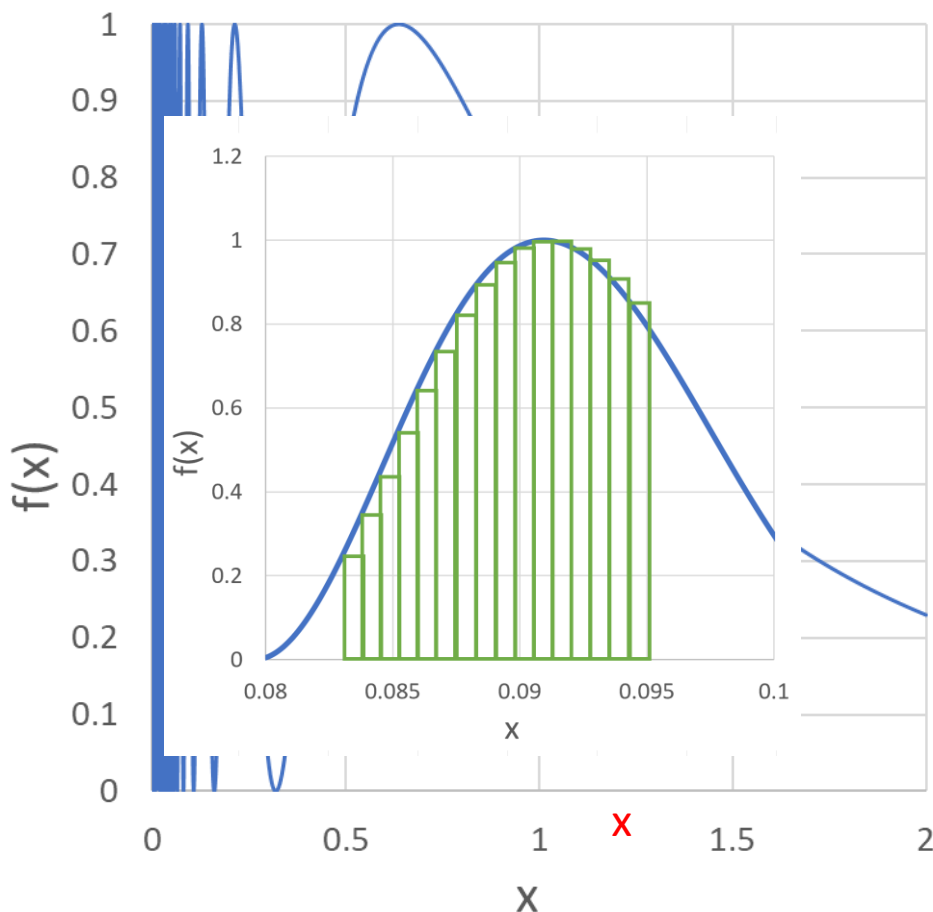
Po dość czasochłonnnych obliczeniach dochodzi się do wniosku, że błąd powierzchni pod wykresem jest zależny od $\left(\frac{1}{\sqrt{N}}\right)$.. natomiast całkowanie (liczby prób)

(liczby prób)

numeryczne metodą prostokątów generuje błąd na poziomie $\frac{1}{\sqrt{N^d}}$..

.... gdzie d jest liczbą wymiarów ..
(x ; x,y ; x,y,z ; ...),
a N liczbą prostokątów..

Błąd metody MC nie zależy od wymiaru przestrzeni .. Więc już dla $N=2$ oszacowanie jest podobne ...



Przykład finansowy

Firma produkuje rowery.

Jej działalność obarczona jest dużym ryzykiem, ze względu na niekorzystne warunki w branży.

Postanowiono wprowadzić na rynek nowy produkt - hulajnogę.

Oceńmy opłacalności produkcji.

Z przeprowadzonych rozmów z przedstawicielami działu produkcji i marketingu wiesz, że koszty produkcji są stałe i będą wahać się w przedziale 60-80 tys. zł w zależności od aktualnych cen materiałów (pakiet elementów z Chin).

Każda wartość z tego przedziału może wystąpić z jednakowym prawdopodobieństwem.

Analogicznie przedstawia się sytuacja z przychodami.

W pierwszym okresie sprzedaży liczba klientów będzie wahać się w przedziale od 125 do 175 osób, a cena jednostkowa sprzedaży wynosi 500 zł.

$N = 1000$

Przychód:

$(125 + \text{rand}()) \cdot 500 \times 500$

Koszty:

$60\,000 + \text{rand}() \cdot 20\,000$

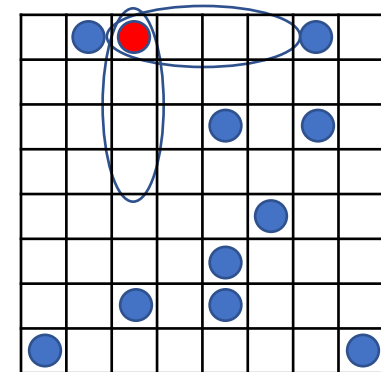
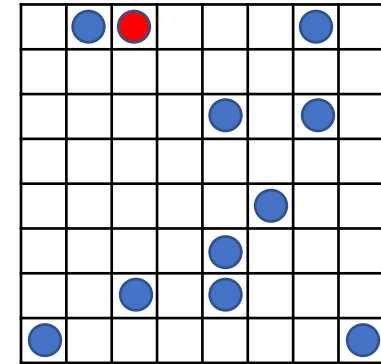
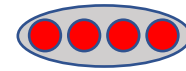
Wynik_finansowy:

Przychód-koszty

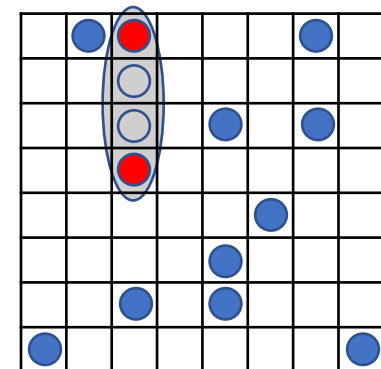
$\text{If}(\text{Wynik_finansowy} > 0) \text{ sukces}++$

$\text{Prawdopodobienstwo_zysku} = \text{sukces} / N$

Przykład: gra w statki.



Losowe próbkowanie
jest efektywniejsze
niż systematyczne
przeszukiwanie
obszaru.



Przykład Naukowy

modelowanie i symulacja rozpraszania cząsteczek w fizyce – zwłaszcza w kontekście analizy cząsteczek w gazach i cieczech.

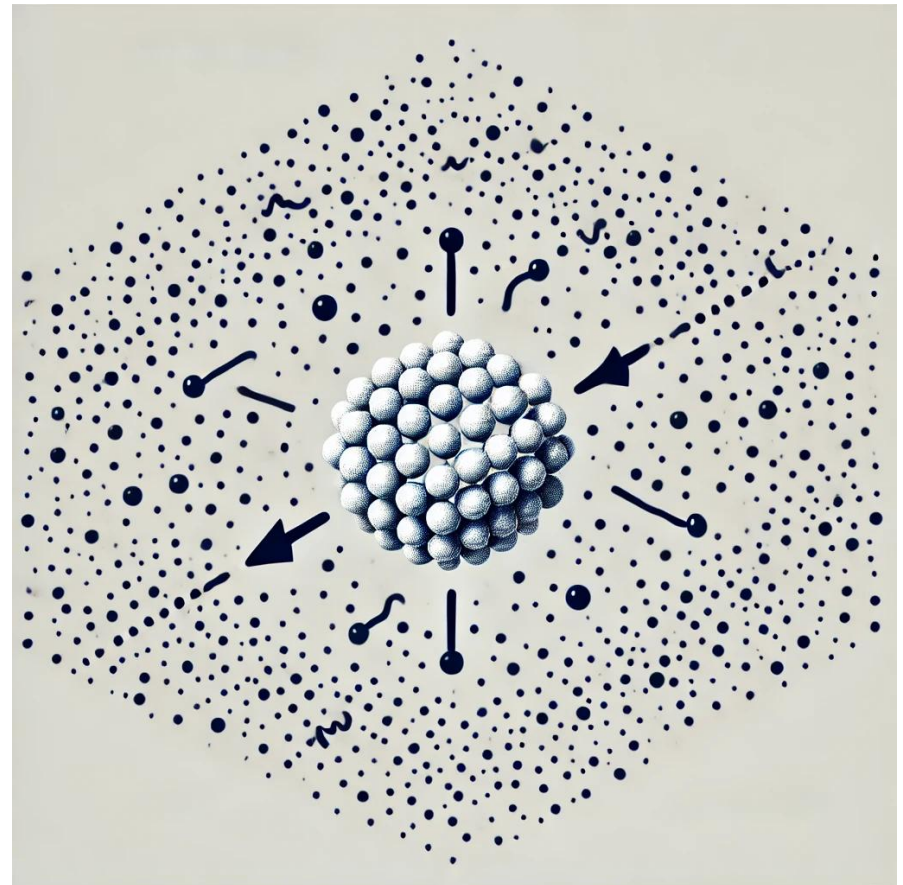
Kroki w symulacji Monte Carlo

1.Reprezentacja układu:

1. Układ składa się z N cząsteczek, które są rozproszone w przestrzeni. eni.
2. Każda cząsteczka ma swoje właściwości, takie jak masa, prędkość, kierunek ruchu i energia kinetyczna. Możemy je generować losowo, aby symulować różne stany systemu.

2.Losowe generowanie trajektorii:

1. W każdym kroku symulacji, cząsteczka porusza się po losowej trajektorii. Wybieramy kierunek, w którym cząsteczka porusza się, oraz jej prędkość, która jest również losowa, ale zgodna z rozkładem prędkości cząsteczek w gazie (np. rozkład Maxwella).
2. Następnie obliczamy potencjalne zderzenia cząsteczek w przestrzeni.



Przykład Naukowy

modelowanie i symulacja rozpraszania cząsteczek w fizyce – zwłaszcza w kontekście analizy cząsteczek w gazach i cieczech.

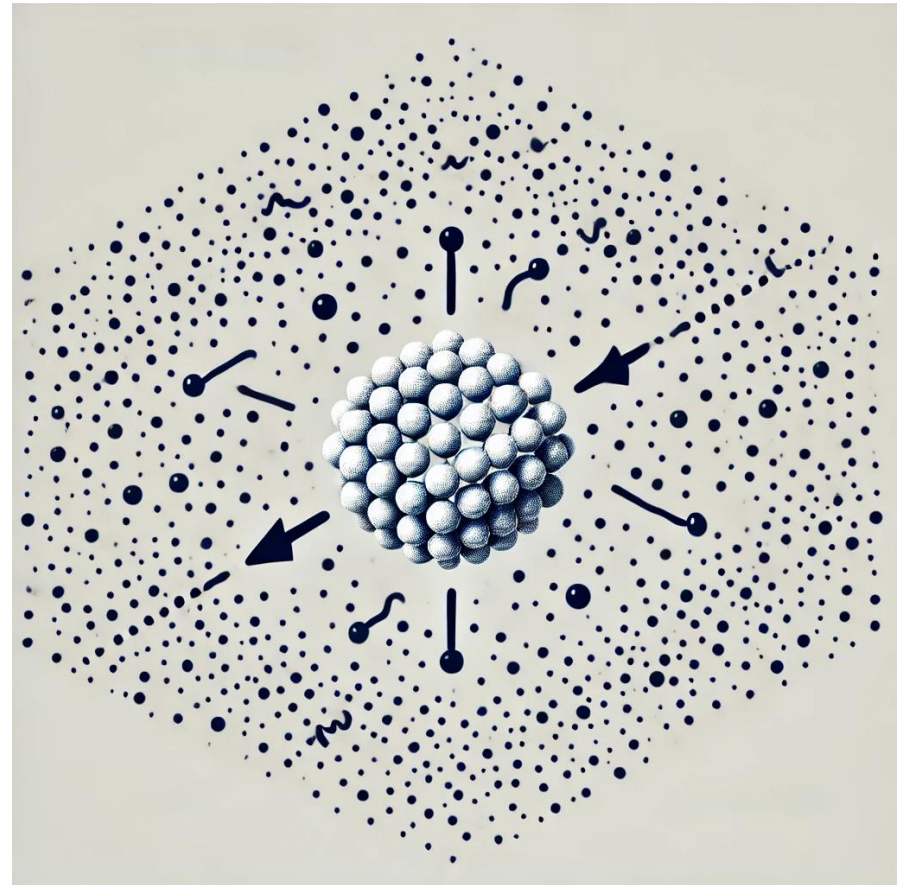
Kroki w symulacji Monte Carlo

3. Symulowanie zderzeń:

1. Zderzenia między cząsteczkami są modelowane jako procesy, w których zachodzi wymiana energii. Po każdym zderzeniu nowe trajektorie cząsteczek są generowane na podstawie zasad fizyki
2. Dzięki metodzie Monte Carlo możemy wybrać odpowiednią liczbę zderzeń w jednostce czasu i uzyskać statystyczną reprezentację układu.

4. Analiza wyników:

1. Po przeprowadzeniu wielu iteracji, możemy analizować średnie wartości takich wielkości jak średnia energia kinetyczna cząsteczek, średnia odległość między zderzeniami czy średnia czas trwania między zderzeniami.



CHEMIA

**CZASOPISMO TECHNICZNE
TECHNICAL TRANSACTIONS**

CHEMISTRY

WYDAWNICTWO

POLITECHNIKI KRAKOWSKIEJ

2-Ch/2012

ZESZYT 17

ROK 109

ISSUE 17

YEAR 109

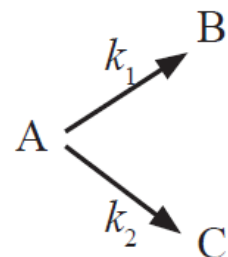
MONIKA GWADERA, KRZYSZTOF KUPIEC*

**ZASTOSOWANIE METODY MONTE CARLO
DO WYZNACZANIA KRZYWYCH KINETYCZNYCH
ZŁOŻONYCH REAKCJI CHEMICZNYCH**

**APPLICATION OF MONTE CARLO METHOD
FOR DETERMINATION OF MULTIPLE REACTIONS
KINETIC CURVES**

2. Reakcje równoległe

Rozważając dwie nieodwracalne jednocząsteczkowe reakcje równoległe I rzędu:



można stwierdzić, że prawdopodobieństwo utworzenia danego produktu: B lub C jest tym większe, im większe jest stężenie substratu C_A oraz im większą wartość ma stała szybkości reakcji tworzenia tego produktu, odpowiednio k_1 i k_2 . W celu uwzględnienia w algorytmie Monte Carlo wpływu wartości stałych szybkości reakcji na przebieg procesu, wprowadzono wielkość Ω :

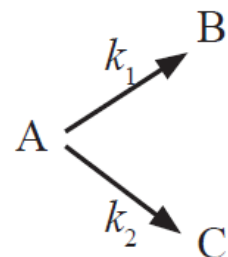
$$\Omega = \frac{k_2}{k_{12}} \tag{3}$$

gdzie:

$$k_{12} = k_1 + k_2 \tag{4}$$

2. Reakcje równoległe

Rozważając dwie nieodwracalne jednocząsteczkowe reakcje równoległe I rzędu:



można stwierdzić, że prawdopodobieństwo utworzenia danego produktu: B lub C jest tym większe, im większe jest stężenie substratu C_A oraz im większą wartość ma stała szybkości reakcji tworzenia tego produktu, odpowiednio k_1 i k_2 . W celu uwzględnienia w algorytmie Monte Carlo wpływu wartości stałych szybkości reakcji na przebieg procesu, wprowadzono wielkość Ω :

$$\Omega = \frac{k_2}{k_{12}} \tag{3}$$

gdzie:

$$k_{12} = k_1 + k_2 \tag{4}$$

Przykłady:

Wartości Ω znajdują się w przedziale od 0 do 1.

Algorytm Monte Carlo dla dwóch nieodwracalnych, jednocząsteczkowych reakcji równoległych I rzędu składa się z czterech kroków [12]:

I. Ustalenie składu początkowego mieszaniny reakcyjnej

Przykładowe liczby molekuł w mieszaninie początkowej: $A = 40000$, $B = 0$, $C = 0$.

II. Losowanie molekuł

W układzie znajdują się molekuły A, B oraz C, przy czym jedynie molekuły A są zdolne do reakcji. Jeśli wylosowano A, reakcja zajdzie. Jeżeli wylosowano B lub C, reakcja nie zajdzie.

III. Losowanie reakcji

Jeśli w drugim kroku algorytmu wylosowano molekułę A, należy określić, która z dwóch możliwych reakcji będzie zachodzić. W tym celu generowana jest liczba losowa q z przedziału $[0, 1]$. Reakcja tworzenia produktu B (reakcja 1) zajdzie, gdy $\Omega < q$, natomiast reakcja tworzenia produktu C (reakcja 2) zajdzie, gdy $\Omega > q$. W ten sposób liczba przemian chemicznych zachodzących w kierunku B jest proporcjonalna do liczby molekuł A i do stałej k_1 , a liczba przemian w kierunku C jest proporcjonalna do liczby molekuł A i do stałej k_2 .

Przykłady:

Na rys. 1 przedstawiono ideę losowania reakcji dla przykładowych wartości stałych szybkości reakcji: $k_1 = 0,05 \text{ 1/s}$, $k_2 = 0,1 \text{ 1/s}$. Wartość Ω jest równa $2/3$ i dzieli odcinek $[0, 1]$ w stosunku $2:1$. Przedział odpowiadający reakcji 1 jest dwa razy krótszy niż dla reakcji 2, a zatem prawdopodobieństwo znalezienia się w tym przedziale, czyli wylosowania liczby q większej od Ω , jest również dwa razy mniejsze. Wynika to ze stosunku stałych szybkości reakcji: $k_2/k_1 = 2$.

IV. Zmiana liczby molekul

Liczba molekul w mieszaninie jest zmieniana zgodnie ze schematem reakcji, która zaszła. Jeżeli zaszła reakcja 1, to liczba molekul A maleje o 1, a liczba molekul B rośnie o 1. Liczba molekul C pozostaje bez zmian. W przypadku gdy zaszła reakcja 2, liczba molekul A maleje o 1, liczba molekul C – rośnie o 1, a liczba molekul B pozostaje bez zmian.

Przedstawiona powyżej procedura jest powtarzana wiele razy. Liczba powtórzeń jest proporcjonalna do czasu, a liczba molekul danego składnika – do jego stężenia.



Rys. 1. Losowanie reakcji – podział odcinka $[0, 1]$ dla $k_2/k_1 = 2$

Mariusz KRAJEWSKI

UNIwersytet ZIELONOGÓRSKI, INSTYTUT METROLOGII ELEKTRYCZNEJ

Zastosowanie metody Monte Carlo do porównania właściwości algorytmów cyfrowego przetwarzania sygnałów

Uniwersytet Jagielloński
Wydział Matematyki i Informatyki
Matematyka

Praca magisterska

Zastosowanie metody Monte Carlo w wycenie opcji finansowych

Marek Śmieja

Tomasz Wiśniewski

Wykorzystanie symulacji Monte Carlo w analizie ryzyka projektów inwestycyjnych

Studia i Prace Wydziału Nauk Ekonomicznych i Zarządzania 34/2, 65-80

2013

Pomiary Automatyka Robotyka, ISSN 1427-9126, R. 24, Nr 2/2020, 31–38, DOI: 10.14313/PAR_236/31

Zastosowanie metody Monte Carlo do parametrycznej identyfikacji akcelerometrów w dziedzinie częstotliwości

Krzysztof Tomczyk

Politechnika Krakowska, Wydział Inżynierii Elektrycznej i Komputerowej, ul. Warszawska 24, 31-155 Kraków

Zalety metody MC

Można skupić się na istocie problemu, uwalniając się od często skomplikowanych teorii i równań

W prosty sposób zastępujemy rozwiązania analityczne

Rosnąca moc obliczeniowa komputerów sprzyja metodzie

Często pozwala rozwiązać trudne problemy

Wady metod MC

Wynik jest zawsze pewnym przybliżeniem (jak we wszystkich heurystykach)

Pomimo rosnącej mocy obliczeniowej zawsze będziemy daleko od nieskończoności..

Na wynik może mieć wpływ słaby generator liczb pseudolosowych

Plan na najbliższy czas ..

Marzec

2025

Pn	Wt	Śr	Cz	Pt	So	N
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

Wykład kolejny za tydzień ..
odwołany .. 2024-03-19

! Najbliższe obligatoryjne
ćwiczenia:
● 2024-03-13 i 2024-03-14
Na tych ćwiczeniach
zrobimy kolejny wykład
oraz określe pierwszy
(„na rozgrzewkę”)
projekt ...

Ćwiczenia 20 i 21 marca
będą ćwiczeniami nieobligatoryjnymi ...
(to czas na zrobienie projektów..)

! Kolejny (trzeci) wykład: 2024-03-26 – będzie to jeden z najważniejszych wykładów..

! Kolejne (drugie) obligatoryjne laborki: 2024-03-27 i 2024-03-28

Inteligencja obliczeniowa w analizie danych

moja propozycja:

4 mini-projekty w ramach ćwiczeń:

- **Algorytm Monte Carlo**
- Algorytm genetyczny
- System ekspercki na fuzzy logic
- Sieci neuronowe

Obecność w sensie ścisłym obowiązkowa na 4 ćwiczeniach w czasie omawiania koncepcji kolejnych mini-projektów i streszczenia zagadnienia niezbędnego do realizacji

zaliczenie mini projektów: m-plik lub .py z programem + krótkie sprawozdanie .pdf z opisem i własnymi wnioskami ..

(minimum na zaliczenie ćwiczeń to 2z4 mini projektów)

2 projekty – 3.0

3 projekty – 4.0

4 projekty – 5.0

Na koniec prosty egzamin – zrobimy termin zerowy pod koniec semestru..

Projekty są kompaktowe i przekrojowe.
Liczę na Państwa dojrzałość – każdy pracuje dla siebie..
Apeluję o samodzielność..

Algorytm „czysto” probabilistyczny - Monte Carlo

Klasyka gatunku – przybliżymy wartość liczby π ...

Wygenerujemy N par liczb x_n, y_n będących współrzędnymi punktów na kwadracie $(0,0 ; 1,1)$

Skorzystamy z równania okręgu $x^2 + y^2 = r^2$ dla $r=1$ - ograniczymy się do 1 ćwiartki..

Przekształcimy równanie na postać $y=f(x)$...

Wielokrotnie, dla kolejnych $n < N$ sprawdzimy, czy $y_n \leq f(x_n)$

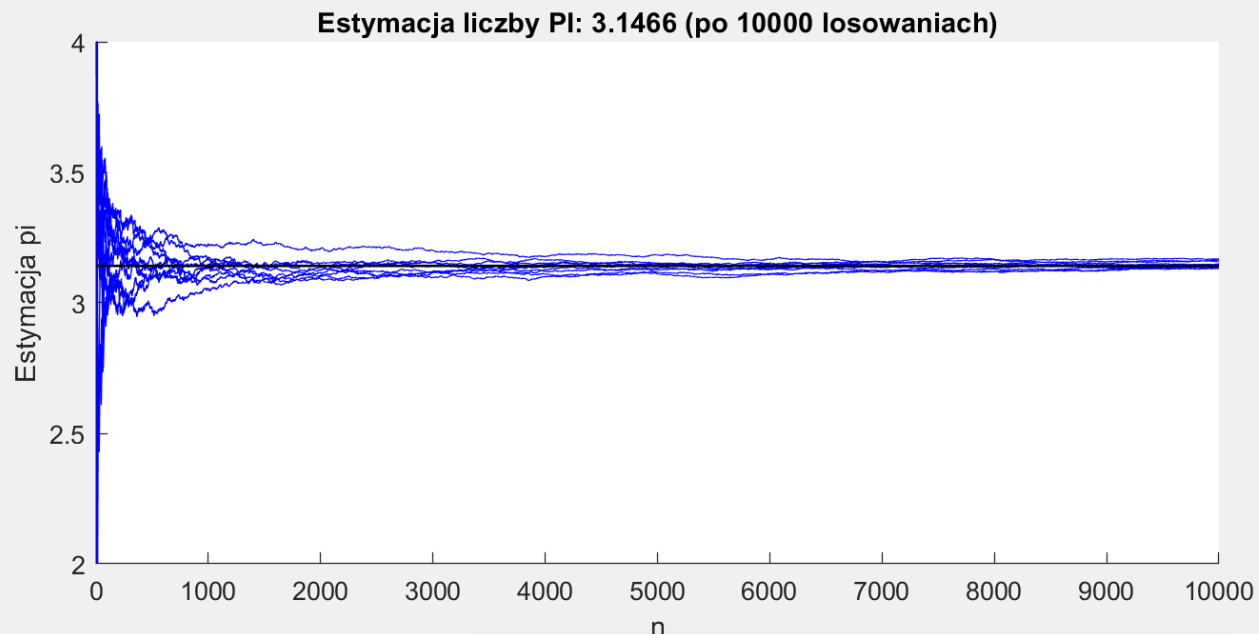
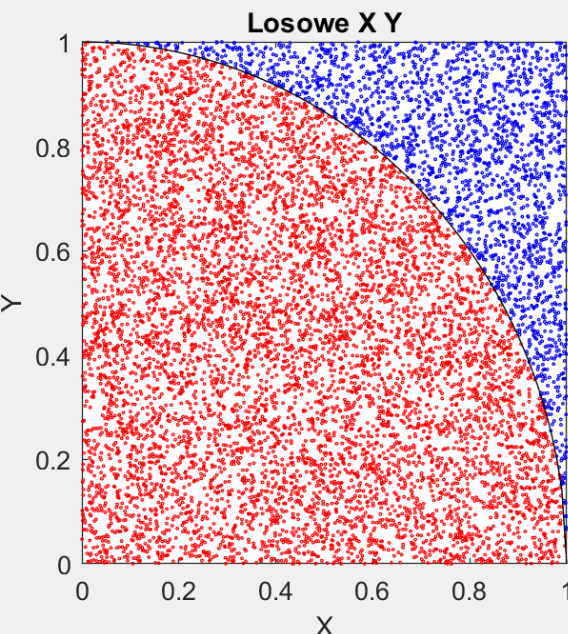
Statystycznie obliczymy stosunek pola  do pola  ()

W ten sposób przybliżymy wartość liczby π ..

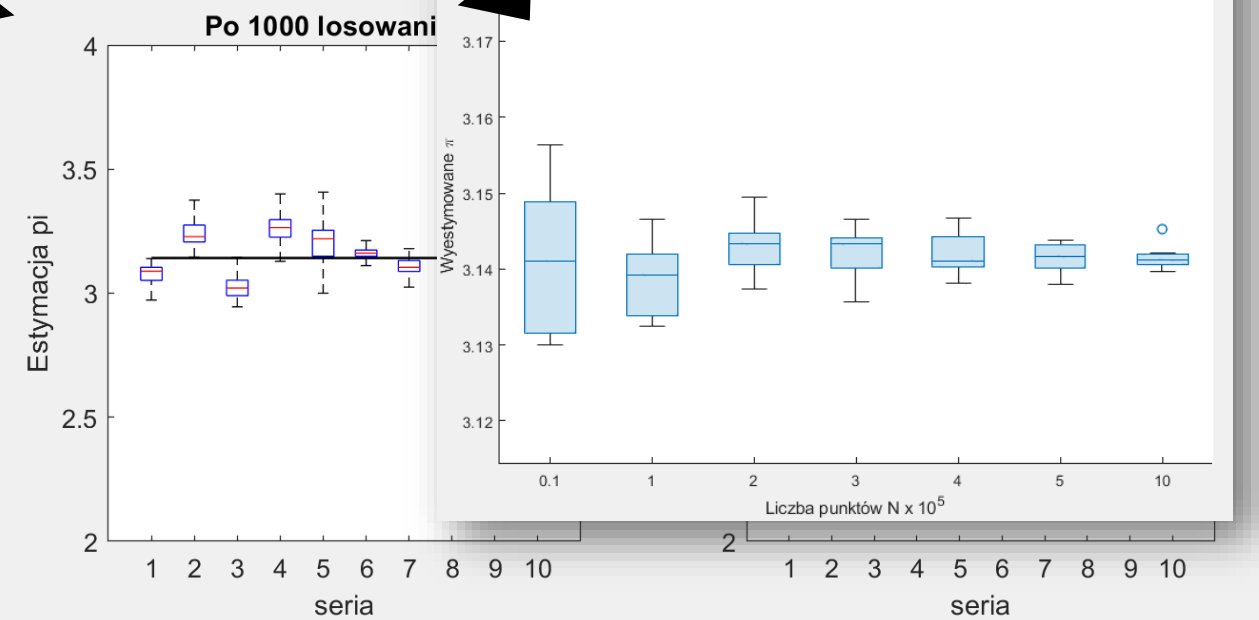
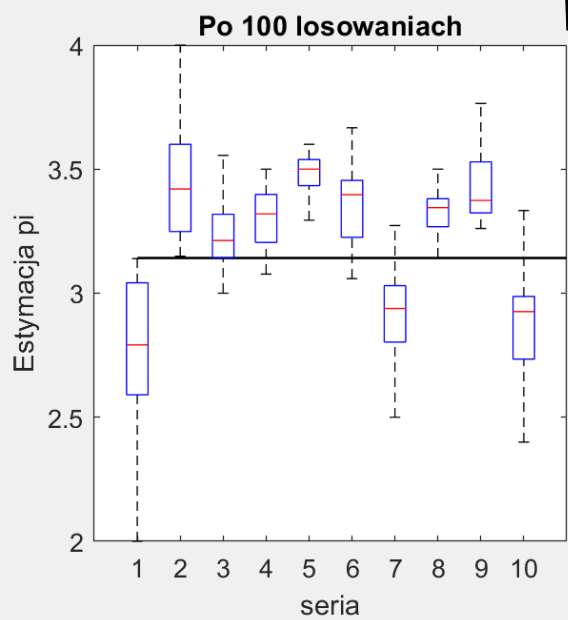
Sprawdzimy, jaki wpływ na wynik ma ilość losowań N (powiedzmy do 100 000).

Przykładowa projekcja wyników może wyglądać tak ↓ ...

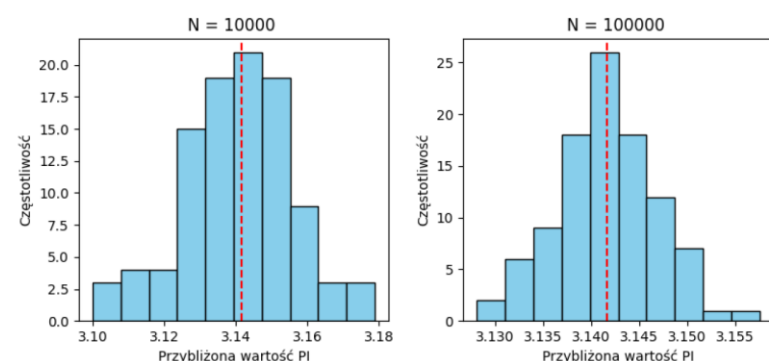
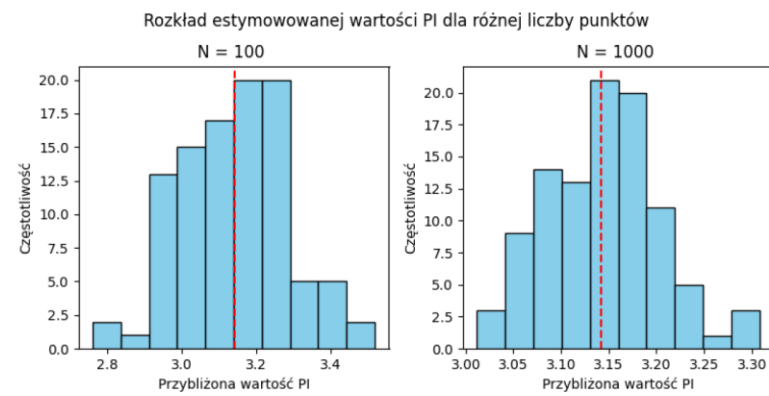
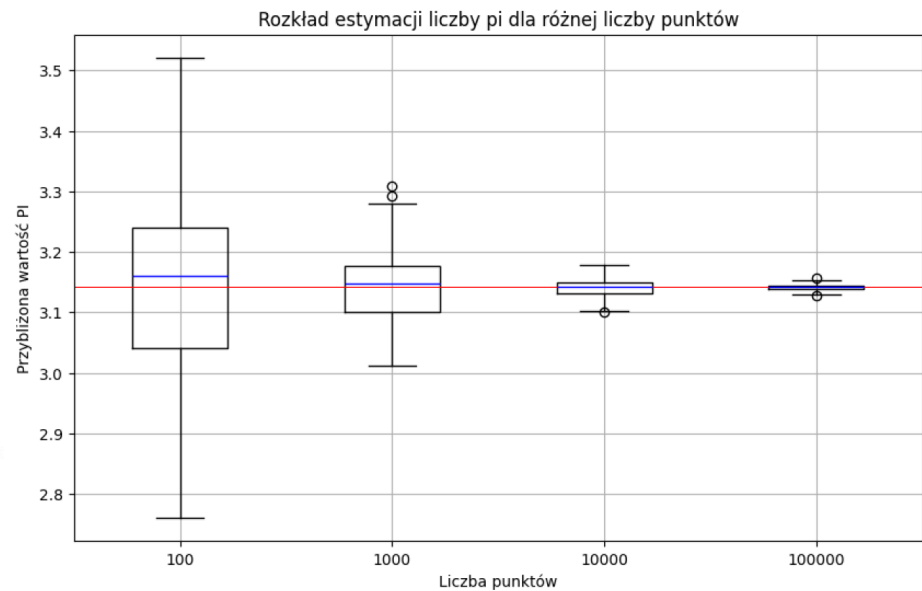
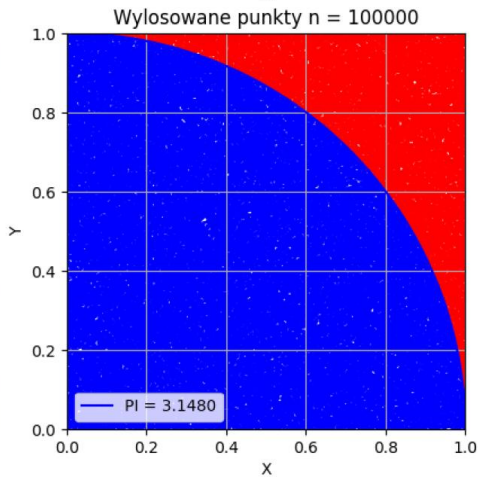
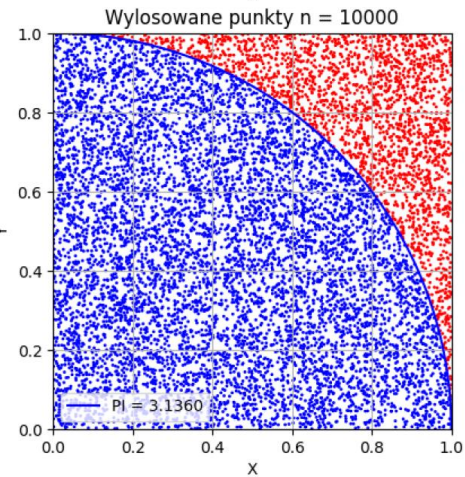
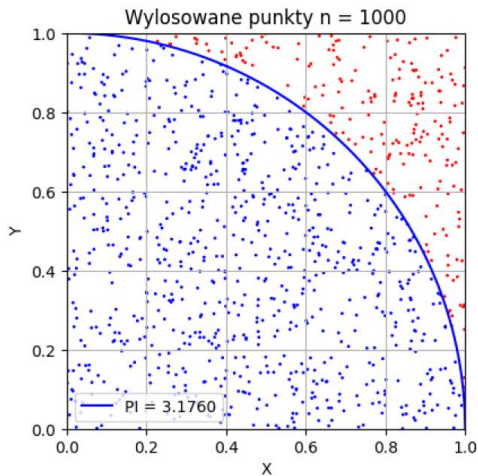
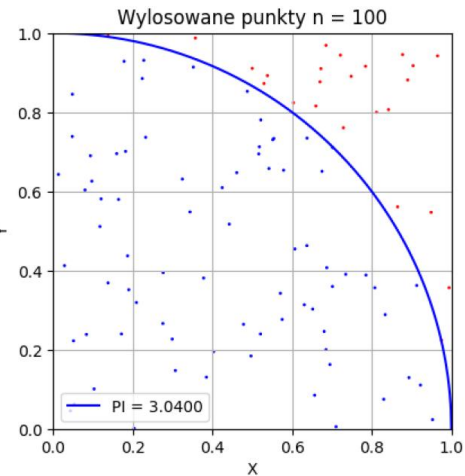
Przykładowa wizualizacja wyników ...



tak, lub tak ...



Przykładowa wizualizacja wyników ...



Ważne, żeby znalazły się wykresy :

- 1) Jak losowane są punkty – z zaznaczeniem tych leżących pod wykresem oraz nad wykresem ..
(oraz ćwiartka okręgu)
- 1) Jakie jest „dążenie” chwilowej estymowanej wartości π wraz ze wzrostem ilości losowań ..
(w tle linia 3.1416 ..)
- 2) Proces można uruchomić kilkakrotnie (np. 10 razy) – sprawdzić na wykresie, jakie są „ścieżki”
dążenia do π ...

Dodatkowo można..

- 3) Dokonać wybranej analizy statystycznej zobrazowanej graficznie (wykresem), potwierdzającej
poprawę jakości estymacji wraz ze wzrostem N ... (mogą to być wykresy „pudełkowe”
statystyka np. dla 10 serii po $N=100$, $N=1000$, .. $N=100\ 000$..

Oczekuję m-pliku lub .py z programem oraz krótkiego opisu,
jak pracuje algorytm wraz z wnioskami ..

prace przesyłamy na maila nskoczylas@agh.edu.pl

Jeśli ktoś ma inny pomysł na Monte Carlo – jestem otwarty na propozycję..

Estimating the value of π using Buffon's needle problem and a Monte Carlo method

Opis algorytmu:
Do estymacji wartości liczby π postanowiłem użyć metody Monte Carlo oraz problemu **Igły Buffona**. Zagadnienie polega na tym, że na planszy z zaznaczonymi poziomymi liniami odległymi od siebie o t upuszczamy n igieł o długości l ($l \leq t$) i chcemy wyznaczyć prawdopodobieństwo p , że igła spadnie tak, że przetnie jedną z linii. Po użyciu odpowiednich wzorów i obliczenia całek otrzymujemy wzór na prawdopodobieństwo:

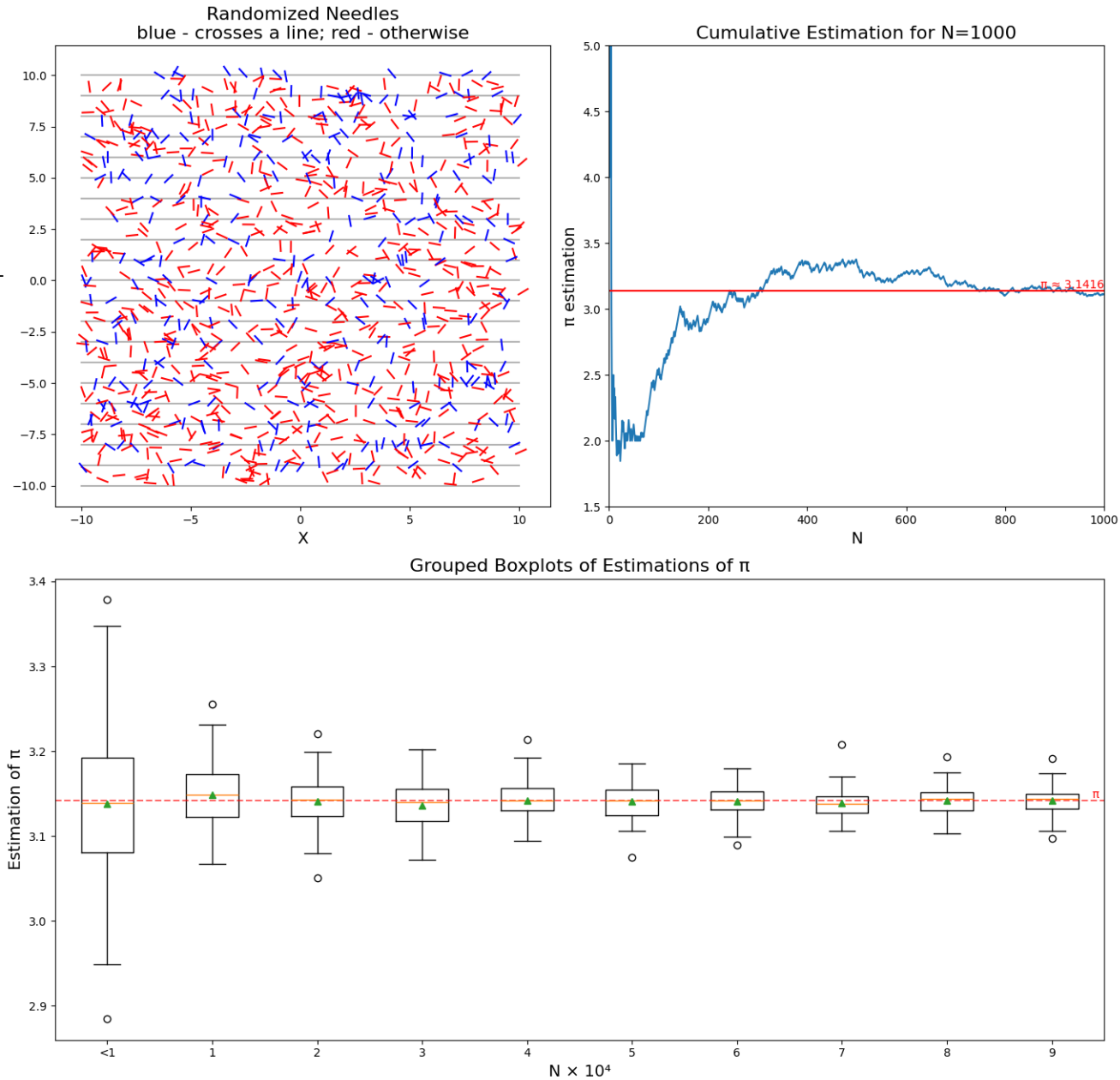
$$p = \frac{2l}{\pi t}$$

Stosując metodę Monte Carlo, tzn. wielokrotnie (n -krotnie) losując jak upadanie igła na planszy (losujemy najpierw współrzędną początku igły, a następnie także kąt z przedziału $[0, 180]$ stopni i korzystając z trygonometrii obliczamy współrzędne końca igły) możemy estymować prawdopodobieństwo p jest w przybliżeniu równe liczbie igieł, które spadły i przecięły jakąkolwiek linię, podzielonej przez liczbę losowań n . Korzystając z tej wiedzy możemy wyprowadzić wzór do estymowania liczby π :

$$\pi \approx \frac{2ln}{h}$$

Wnioski:

Dla $N=1000$ przeprowadziłem osobną analizę, w tym wizualizację jak igły zostały losowo wygenerowane na planszy, a także jak zmieniała się skumulowana estymacja liczby π po każdym kolejnym losowaniu. Można zauważyć, że od losowania nr 200 estymowana wartość zaczęła już zbliżać się do prawdziwej z błędem nieprzekraczającym 0,3. Od losowania o numerze około 750 szacowana wartość była już bardzo bliska π , zatem dobre przybliżenie zostało uzyskane w krótkim czasie. Dodatkowo przeprowadziłem osobne losowania o liczbie randomizowanych igieł od 100 do 100,000 z krokiem 10 otrzymane wartości estymowane podzieliłem na 10 grup zależnych od N . Można zauważyć, że pomimo znacznej różnicy liczbie losowanych wektorów średnia szacowana wartość utrzymywała się na poziomie bardzo bliskim π . ogółem wraz ze wzrostem N malał zasięg wąsów wykresu pudełkowego dla danej grupy, co oznaczało „pewniejszy” wynik. Mimo to, nawet dla losowań o $N > 90,000$, zdarzają się wyniki znacznie odstające od reszty wartości w tej grupie. Pomimo dużej liczby losowań o wysokim N cały program wykonuje się dość szybko, co jest wielkim plusem metody Monte Carlo.



Marzec

2025

Na kiedy ??

6 kwietnia/
montecarlo

Pn	Wt	Śr	Cz	Pt	So	N
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

6 kwietnia/
montecarlo

Kwiecień

6 kwietnia/
montecarlo

Pn	Wt	Śr	Cz	Pt	So	N
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				