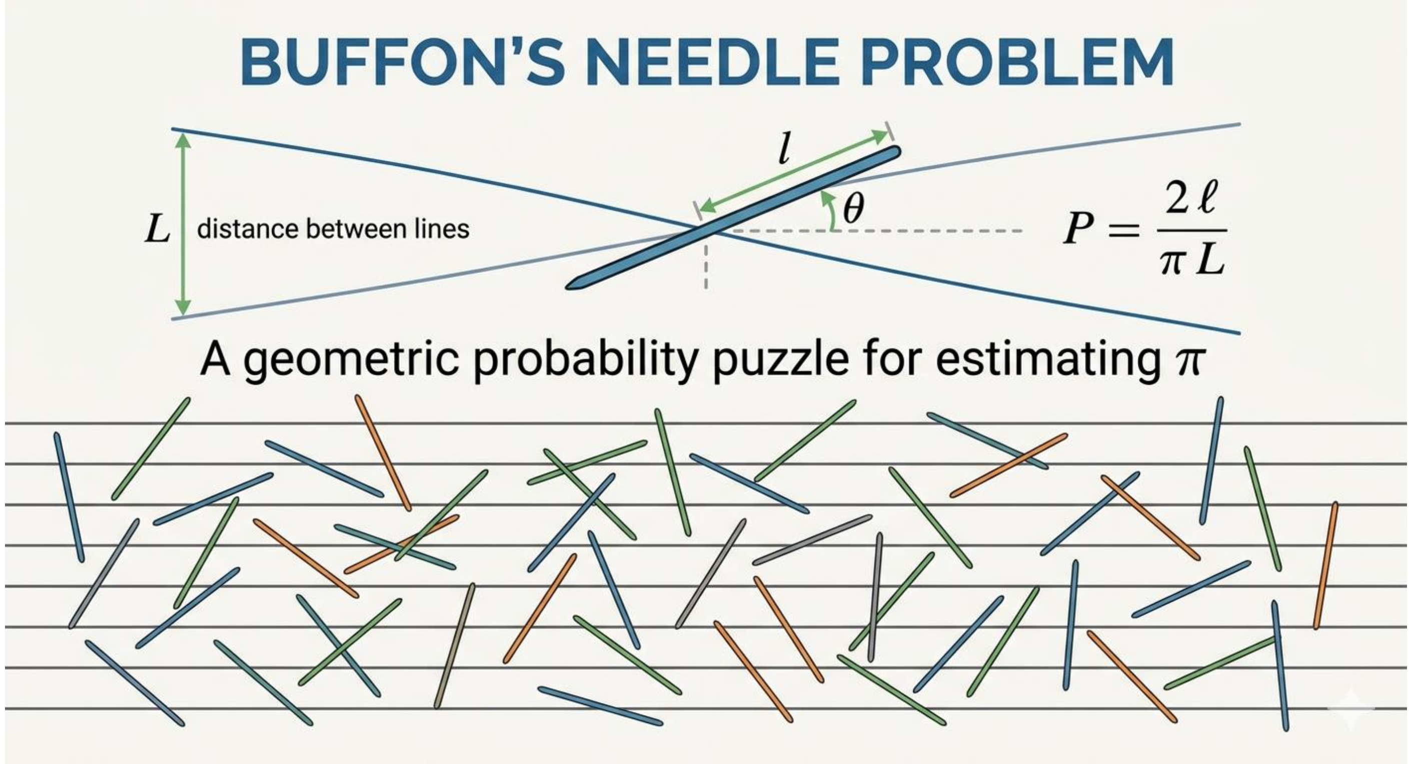
ЗАДАЧА БЮФФОНА
МЕТОД
МОНТЕ-КАРЛО

Задача Бюффона

Задача Бюффона, яка є історичним попередником методу Монте-Карло
Жорж-Луї Леклерк, граф де Бюффон - французький натураліст, біолог, математик, геолог, письменник і перекладач XVIII століття, сформулював задачу у **1733 році**, а детально описав у 1777-му



Експеримент з голкою

Уявіть, що перед вами аркуш паперу, розлінований паралельними прямими, відстань між якими дорівнює L. Є голка довжиною l (для спрощення зазвичай беруть l <=L).

Питання: Яка ймовірність того, що голка, кинута навмання, перетне одну з ліній?

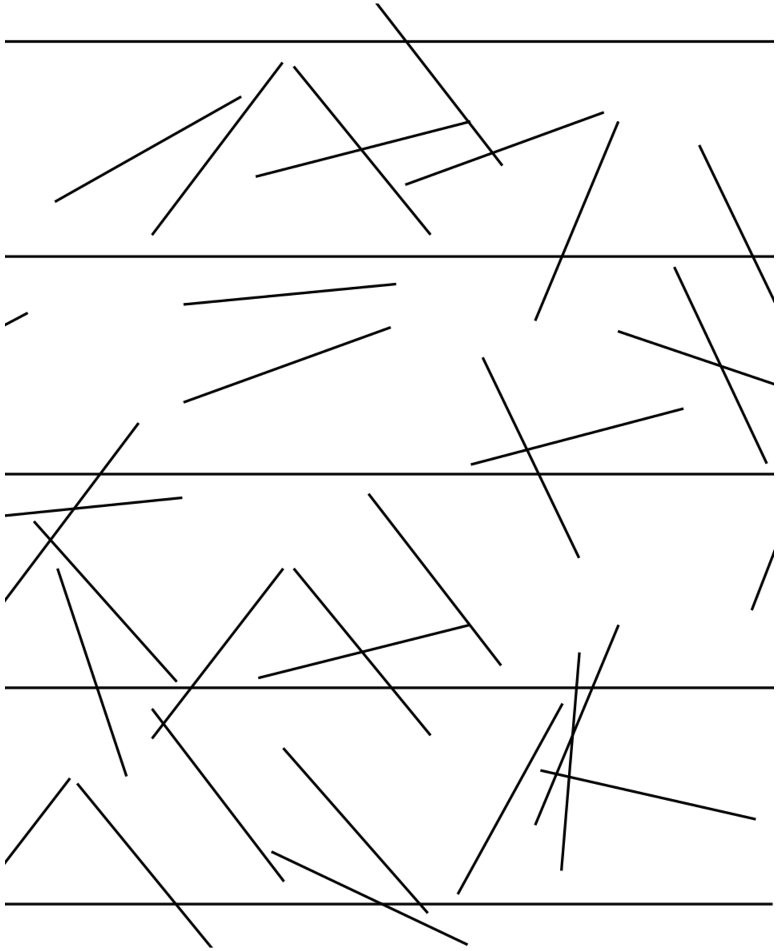
Математичний розв’язок

Бюффон довів за допомогою інтегрального числення, що ця ймовірність P безпосередньо пов'язана з числом π:

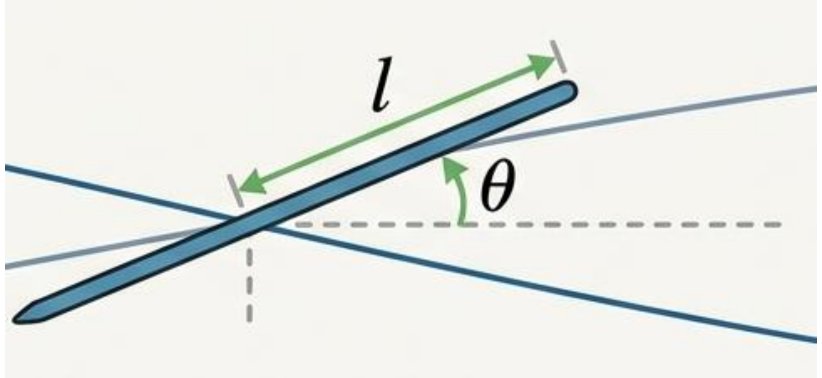
$$P = \frac{2l}{\pi L}$$

Якщо ми візьмемо голку довжиною рівною відстані між лініями (***l* = *L***), то формула спрощується до:

$$P = \frac{2}{\pi}$$



Число **π** з'являється через **можливі орієнтації голки**



1. Випадковість кута

Коли ми кидаємо голку, її положення визначається не тільки тим, куди впаде її центр, а й тим, під яким **кутом** вона ляже відносно ліній

Кут повороту **Θ** (тета) може бути будь-яким від 0 до 180 градусів (або від 0 до **π** радіан). Цей повний діапазон можливих поворотів і приносить із собою геометрію кола

2. Геометрія перетину

Щоб голка перетнула лінію, її вертикальна проекція має бути достатньо великою

Якщо голка лежить паралельно лініям ***Θ* = *0***), вона ніколи їх не перетне

Якщо голка лежить перпендикулярно лініям (***Θ* = 90°**), ймовірність перетину найбільша

Математично, довжина цієї проекції залежить від **синуса** кута нахилу (***sin(Θ)***).

Щоб знайти загальну ймовірність перетину, математики мають «додати» (інтегрувати) всі можливі положення голки для всіх можливих кутів від 0 до $\boldsymbol{\pi}$. Коли обчислюється середнє значення синуса на півколі, у результаті розрахунків з'являється знаменник, що містить довжину цього діапазону кутів, тобто $\boldsymbol{\pi}$

Число π у формулі $P = \frac{2l}{\pi L}$ - це математичне відображення того факту, що **голка може обертатися**, а простір усіх можливих обертань описується колом

Тому, рахуючи кількість перетинів, ми опосередковано вимірюємо «міру» всіх можливих поворотів голки, яка містить у собі $\boldsymbol{\pi}$

Знамениті експерименти (майже «комп'ютерні» обчислення вручну)

Протягом XIX століття математики-ентузіасти витрачали тижні, кидаючи голки, щоб перевірити теорію:

1850 рік, Рудольф Вольф: кинув голку **5000 разів**. Він отримав значення **$\pi \approx 3.1596$**

1901 рік, Маріо Лаццаріні: провів «найуспішніший» експеримент, зробивши **3408 кидків**. Він отримав **$\pi \approx 3.1415929$**, що збігається до 6-го знаку!

(*Але, пізніше історики запідозрили його в шахрайстві: він обрав таку кількість кидків, щоб дріб* $\frac{n}{m}$ *точно наближався до відомого раціонального наближення* $\boldsymbol{\pi = \frac{355}{133}}$). *$n$ – кількість кидків, m – кількість випадків, коли голка перетнула лінію*

Пояснення задачі Бюффона

https://www.youtube.com/watch?v=sJVivjuMfWA

Метод Монте-Карло

Метод Монте-Карло - це група чисельних методів для вивчення математичних процесів за допомогою моделювання випадкових величин.

Основна ідея: Замість розв'язання складної аналітичної задачі проводиться велика кількість випадкових експериментів і обчислюється середній результат

Закон великих чисел: Точність методу зростає пропорційно $\sqrt{n}$, де n - кількість ітерацій.

Метод отримав свою назву на честь відомого казино в Монако, оскільки гра в рулетку - це найпростіша модель генерації випадкових чисел.

Класичний приклад обчислення числа $\boldsymbol{\pi}$

Найпростіша візуалізація методу - метод «потрапляння в мішень».

- Вписуємо чверть кола радіусом $\boldsymbol{r}$ у квадрат зі стороною $\boldsymbol{r}$
- Генеруємо велику кількість випадкових точок ($\boldsymbol{x}$, $\boldsymbol{y}$) у межах квадрата
- Рахуємо кількість точок, що потрапили всередину кола (де $\boldsymbol{x^2 + y^2 \leq r^2}$).
- Відношення точок у колі до загальної кількості точок наближається до $\frac{\pi}{4}$

Реалізація обчислення числа π методом Монте-Карло

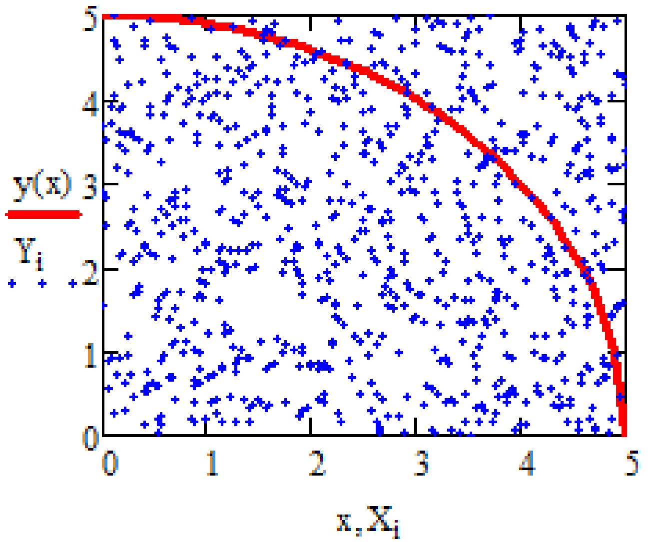
$a := 5$ Сторона квадрата

$N := 800$ Кількість спроб (кидків)

$i := 1..N$

$x := 0, 0.1..a$

$y(x) := \sqrt{a^2 - x^2}$ $\quad X_i := rnd(a)$ $\quad Y_i := rnd(a)$



$$n := \sum_{i=1}^{N} \left[\left[(X_i)^2 + (Y_i)^2 \right] \le a^2 \right]$$

$n = 629$ Кількість вдалих кидків (ті, що потрапили в виділений сектор)

$$Pi := \frac{n \cdot 4}{N} = 3.145$$

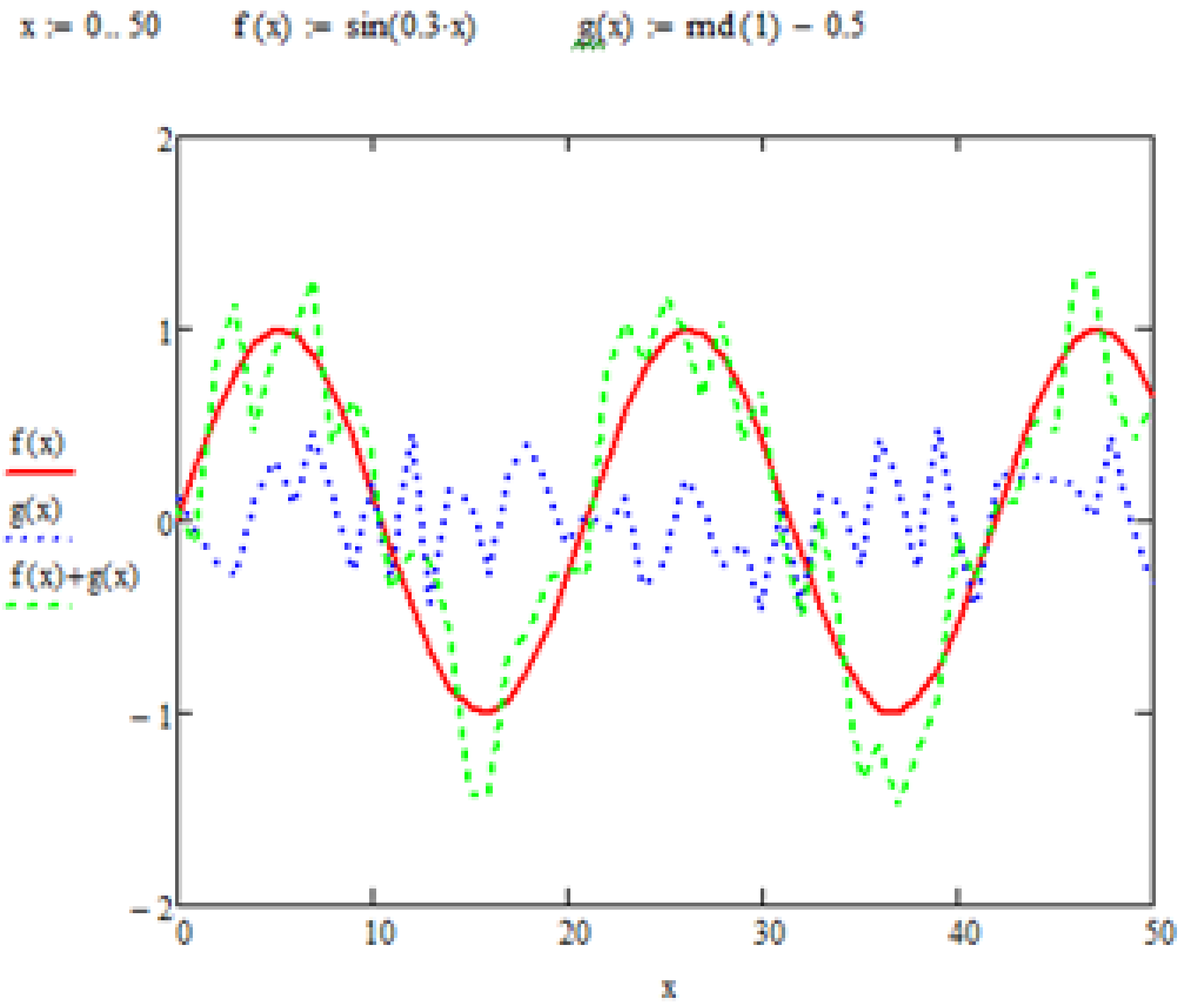
Моделювання сигналів та шумів за методом Монте-Карло

Для моделювання різних фізичних, економічних та інших явищ широко розповсюджені статистичні методи, які називають методами Монте-Карло, основна ідея яких полягає в генерації випадкових послідовностей, що моделюють відповідний ефект, наприклад шум в фізичному експерименті

Для генерації одного псевдовипадкового числа в MathCAD використовується, наприклад, функція:

`rnd(x)` - випадкове число, яке має рівномірну щільність розподілу на інтервалі (**`0,x`**)

Приклад візуалізації сигналу та шуму у фізичному експерименті



Що таке детермінований алгоритм?

У сучасному програмуванні та ІТ-технологіях метод Монте-Карло використовується там, де детерміновані алгоритми занадто повільні або неможливі (задачу не можна розв'язати гарантовано точно за прийнятний час). Головна характеристика - **передбачуваність**

Принцип: Для одних і тих самих вхідних даних алгоритм завжди виконає **абсолютно однакову** послідовність кроків і видає **абсолютно однаковий** результат. У ньому немає місця випадковості.

Приклад: Алгоритм сортування бульбашкою, пошук найкоротшого шляху Дейкстри, обчислення факторіала. Якщо ви дасте їм масив [3, 1, 2], вони завжди повернуть [1, 2, 3].

Проблема «повільності детермінованих алгоритмів» відбувається, коли задача має **величезну обчислювальну складність**. Спроба знайти ідеально точне, гарантоване рішення аналітичним шляхом призводить до експоненціального зростання часу роботи.

Класичний приклад: Гра в шахи або Ґо

Детермінований підхід: Комп'ютер намагається прорахувати *абсолютно всі* можливі ходи, відповіді суперника, відповіді на відповіді і так до кінця партії.

Проблема: Кількість можливих позицій у Ґо (10^{170}) перевищує кількість атомів у Всесвіті. Прорахувати все неможливо навіть на суперкомп'ютерах. Детермінований алгоритм тут «занадто повільний» (фактично, він буде працювати вічно).

Використання методу Монте-Карло

Монте-Карло відмовляється від пошуку єдиного, ідеально точного рішення. Замість цього він використовує **випадковість** (стохастичність), щоб знайти **наближене**, але прийнятне рішення

Принцип: Проводяться тисячі випадкових експериментів і статистично узагальнюється результат. Приймається той факт, що відповідь буде з певною похибкою, але ми отримаємо її швидко. Замість того, щоб перебирати *все*, перебирається *випадкова вибірка*

Приклад: Гра в Ґо (AlphaGo)

Підхід Монте-Карло (Monte Carlo Tree Search, MCTS): Замість повного перебору, програма робить **тисячі випадкових "розігрувань"** партії з поточної позиції до самого кінця. Вона не знає ідеального ходу, але вона бачить, що хід А призводить до перемоги у 80% випадкових розігрувань, а хід Б - лише у 20%.

Результат: Програма робить хід А. Вона не гарантує перемогу, але вона робить статистично найкращий вибір за обмежений час. *Це зробило неможливе можливим))*