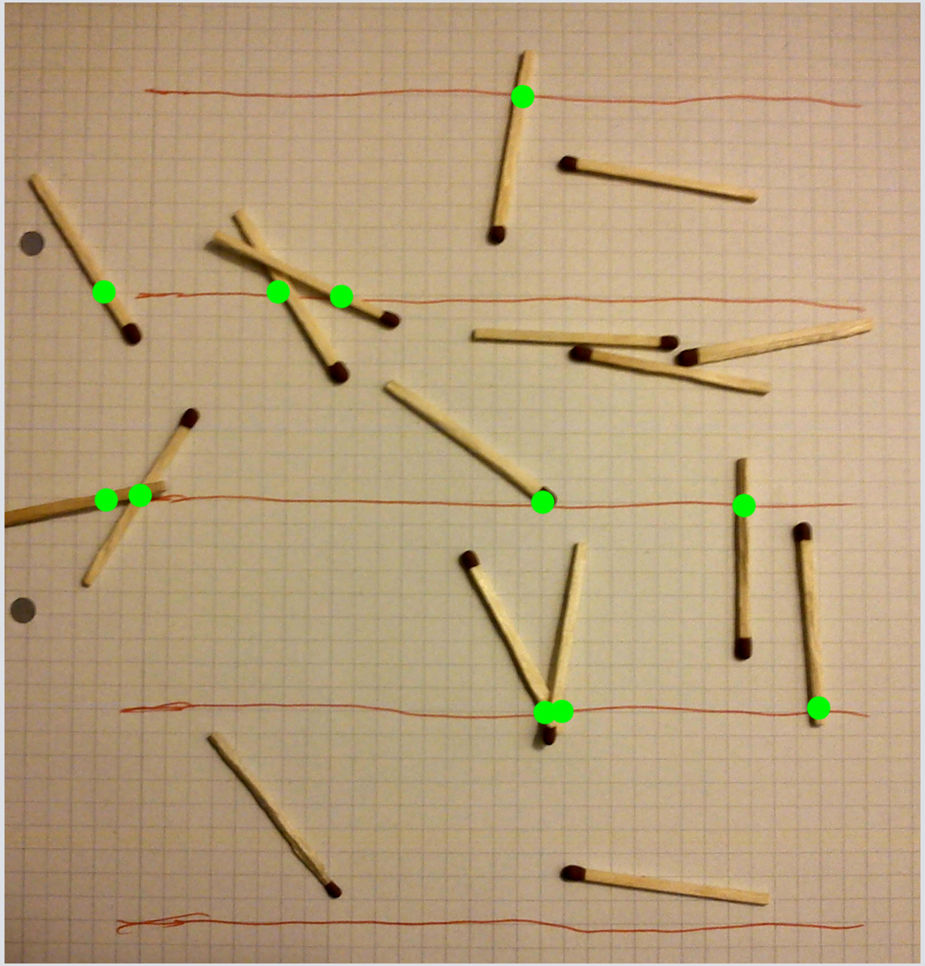
π közelítése gyufák segítségével

Sebe Zsolt, Február 25, 2025

Bevezetés

A probléma lényege

Véletlenszerűen ledobott gyufák és párhuzamos vonalak metszése alapján lehet π-t becsülni.



Matematikai háttér

- L: a gyufa hossza
- d: a párhuzamos vonalak távolsága
- p: annak a valsége, hogy van metszés $\left(\frac{N_{\text{metszés}}}{N}\right)$

$$p = \frac{2}{\pi} \cdot \frac{L}{d}$$

Matematikai háttér

- L: a gyufa hossza
- d: a párhuzamos vonalak távolsága
- p: annak a valsége, hogy van metszés $\left(\frac{N_{\text{metszés}}}{N}\right)$

$$p = \frac{2}{\pi} \cdot \frac{L}{d} \Rightarrow \pi \approx \frac{2 \cdot L \cdot N}{d \cdot N_{\text{metszés}}}$$

Matematikai háttér

Ha $d = 2L$ (azaz a gyufa pont fele olyan hosszú mint a vonalak közötti távolság), akkor:

$$\pi \approx \frac{2 \cdot L \cdot N}{2 \cdot L \cdot N_{\text{metszés}}}$$

Matematikai háttér

Ha $d = 2L$ (azaz a gyufa pont fele olyan hosszú mint a vonalak közötti távolság), akkor:

$$\pi \approx \frac{2 \cdot L \cdot N}{2 \cdot L \cdot N_{\text{metszés}}} \Rightarrow \pi \approx \frac{N}{N_{\text{metszés}}}$$

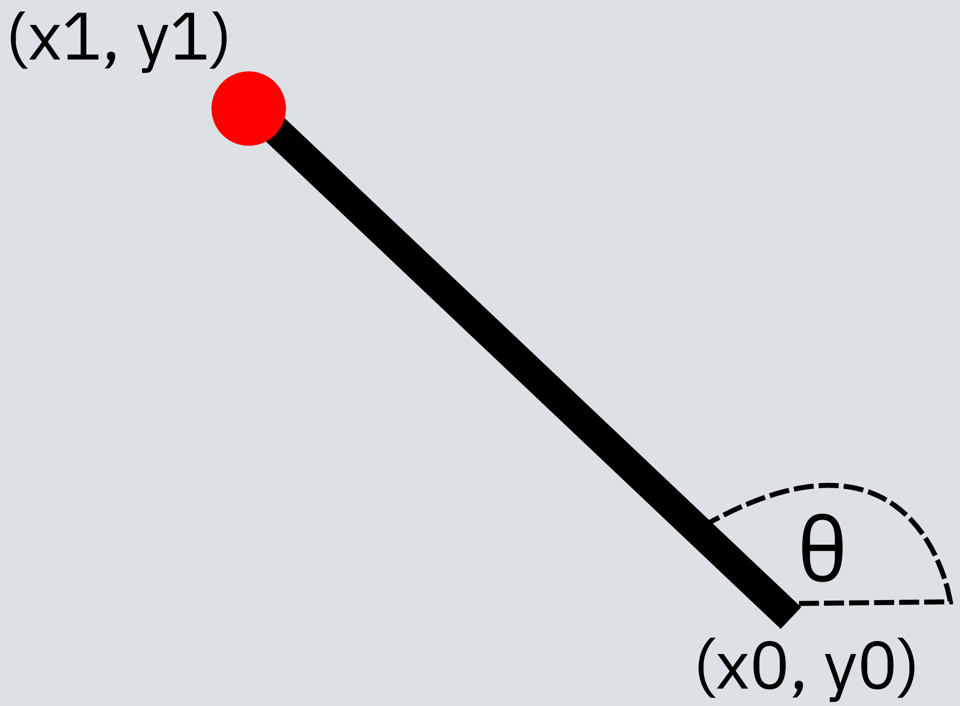
Implementáció

Áttekintés

- Egy C++ kód generálja a random (x_0, y_0, x_1, y_1, c) párost, majd ezt átadja a Python kódnak, vagy kiírja egy file-be (opcionális).
- A Python kód a C++ kóddól kapott párosokból, vagy a file-ból kiolvasva megcsinálja a plot-ot.
- **Figyelem:** Ha túl sok minta van, a plot le fog fagyni.
- Ha csak az eredmény érdekel, akkor a C++ kód ki tudja számítani azt a párosok generálása nélkül.

Áttekintés

- (x_0, y_0) a gyufa egyik végének koordinátái. Értékük random
- (x_1, y_1) a gyufa másik koordinátái, Értékük az előző koordinátától és a θ szögtől függ.
- θ egy egyenletes eloszlású szög. (ez nincs benne a kimenetben, mert csak a másik véghez kell)
- $c \in \{0, 1\}$ megmondja, hogy a gyufánk átmegy-e a vonalon.



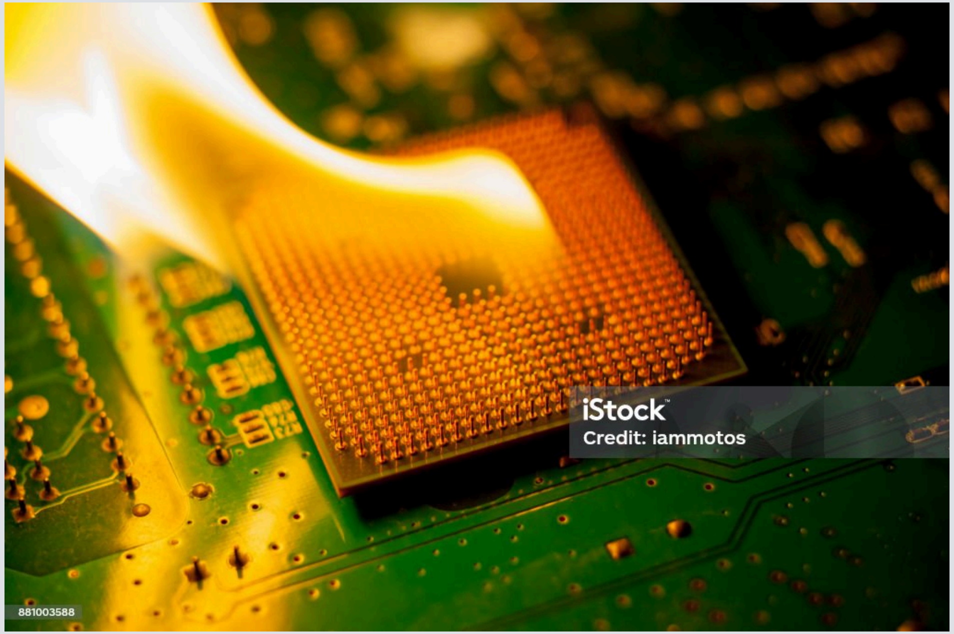
A szimuláció paraméterei

- $L = 1, d = 2$
- **Gyufa:**

$$
\begin{aligned}
x_0 &\leftarrow \mathrm{uniform\_random}(0, 8) \\
y_0 &\leftarrow \mathrm{uniform\_random}(0, 8) \\
\theta &\leftarrow \mathrm{uniform\_random}(0, 360°) \\
x_1 &\leftarrow x_0 + \sin\theta \\
y_1 &\leftarrow y_0 + \cos\theta \\
c &\leftarrow 1 \text{ ha van metszés, különben } 0
\end{aligned}
$$

Problémák a módszerrel

- **Nem hatékony:** 10 milliárd próba után is csak 3 tizedesjegyik jó (≈ 3.141623, err = −0.00003034), és úgy tűnik hogy innentől már nem javul
- **Függőség:** a szimulációnak alapból tudni kell π értékét, bár ez helyettesíthető lenne.
- **Nem realisztikus:** A valóságban ha random dobunk gyufákat nem lesz egyenletes az eloszlás.



Források

- *Buffon's needle problem*, Wikipédia
  https://en.wikipedia.org/wiki/Buffon%27s_needle_problem
- *why it works 8*, @tylerschannel
  https://youtu.be/puDKR0lZRZM

Vége