Računanje približkov za π z Monte Carlo metodo

Miha Brvar
Mentor: Bor Grošelj Simić

Povzetek

Spoznali smo, kako lahko z Monte Carlo metodo in nekaj preprostimi programi dobimo dobre približke za π. To smo dosegli na dva načina: z integriranjem in s simulacijo problema Buffonove igle. V programskem jeziku Python smo napisali programe, ki simulirajo ta dva problema, in analizirali hitrost konvergenc.

1 Uvod

Problem, ki smo ga reševali, lepo povzame naslednja uganka:

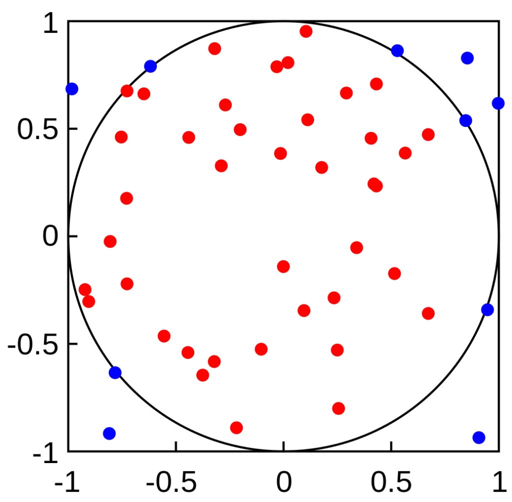
Okrog Marsa kroži sonda, ki na marsovsko postajo na površju pošilja podatke o površju. Del sondinega sistema za obračanje antene odpove, zato se antena začne med pošiljanjem sporočila naključno obračati. Zanima nas, kolikšen delež sporočil bo kljub tej napaki vseeno usmerjen dovolj blizu marsovske postaje, da bo postaja prejela sporočilo.

Območje, na katero satelit pošilja signale, ima obliko kvadrata, postaja pa je okrogla. S pomočjo Monte Carlo metode in programskega jezika Python smo računali približke tega razmerja.

Spoznali smo dve metodi računanja tega približka: Monte Carlo integriranje in simulacija problema Buffonove igle.

2 Monte Carlo integriranje

Monte Carlo integriranje je metoda za izračunavanje približkov ploščin poljubnih likov ali pa prostornine teles v višjih dimenzijah. Pri tem si pomagamo z računalnikom, ker postopek zahteva veliko ponovitev. Izberemo si nek pravokotnik z znanimi stranicami, ki obdaja lik, katerega ploščino želimo izračunati. Računalnik znotraj tega pravokotika izbere naključne točke in preveri, ali se nahajajo znotraj njega ali ne. Po veliko ponovitvah bo iz razmerja števila točk, ki so v liku, in točk, ki niso v liku, mogoče izračunati približek ploščine lika.



Slika 1: Računanje ploščine kroga z Monte Carlo integriranjem.

Program 1: Monte Carlo integracija

```
def integriranje(n)
    a = 0
    b = 0
    for i in range(0,n):

        x = random.random()
        y = random.random()


        d= x**2 + y**2
        if d>1:
            a += 1
        else:
            b += 1

    return(b<<2)/(a+b)
```

Zadali smo si, da bomo s pomočjo te metode izračunali približek za π. Za pravokotnik smo si izbrali enotski kvadrat, ki smo mu včrtali četrtino kroga z enotskim polmerom. Računalniku smo zadali nalogo, naj znotraj tega kvadrata izbere naključne točke in s pomočjo Pitagorovega izreka preveri, ali se nahajajo znotraj četrtine kroga. Po veliko ponovitvah se je vrednost razmerja med točkami znotraj kroga in vsemi točkami približala $\frac{\pi}{4}$.

3 Buffonova igla

Ta problem si je prvič zastavil Georges-Louis Leclerc de Buffon v 18. stoletju. V njem si je zamislil tla iz enako širokih desk, na katera bi metal iglo in se vprašal, kakšna je verjetnost, da igla po pristanku leži tako, da prečka mejo med deskama (igla a na Sliki 2). Buffonova igla je prvi znani problem geometrijske verjetnosti in izkaže se, da je njegova rešitev $\frac{2l}{\pi t}$, pri čemer je t širina desk in l dolžina igle.

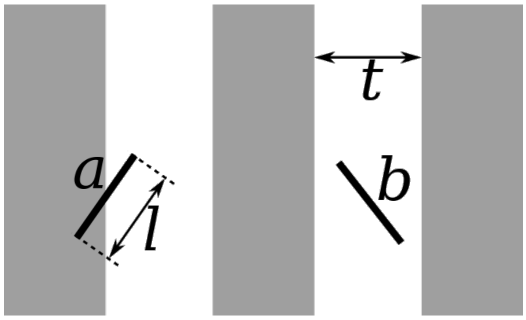
Da je bil problem čim bolj preprost, smo obravnavali primer, ko je $l = t = 1$. Točna rešitev problema zahteva poznavanje kotnih funkcij in integriranja, mi pa smo se ga lotili z Monte Carlo metodo. Računalniku smo določili območje, v katerem naj izbira naključne lokacije in orientacije palic. Nato smo prešteli, koliko palic je sekalo mejo med črnimi in belimi polji, ki so predstavljala deske.

Program 2: Simulacija Buffonove igle

```
def preveri(koordinate):
    x = koordinate[0]
    x = math.floor(x)
    if x % 2 == 1:
        return 'bela'
    else:
        return 'crna'

def krajisca(palica):
    kot = palica[1]
    a = math.sin(kot) / 2
    b = math.cos(kot) / 2
    return ((palica[0][0] + b, palica[0][1] + a),
                              (palica[0][0] - b, palica[0][1] - a))

def ali_seka(krajisce1, krajisce2):
    if krajisce1 == 'bela' and krajisce2 == 'bela':
        return 0
    if krajisce1 == 'bela' and krajisce2 == 'crna':
        return 1
    if krajisce1 == 'crna' and krajisce2 == 'bela':
        return 1
    if krajisce1 == 'crna' and krajisce2 == 'crna':
        return 0
```

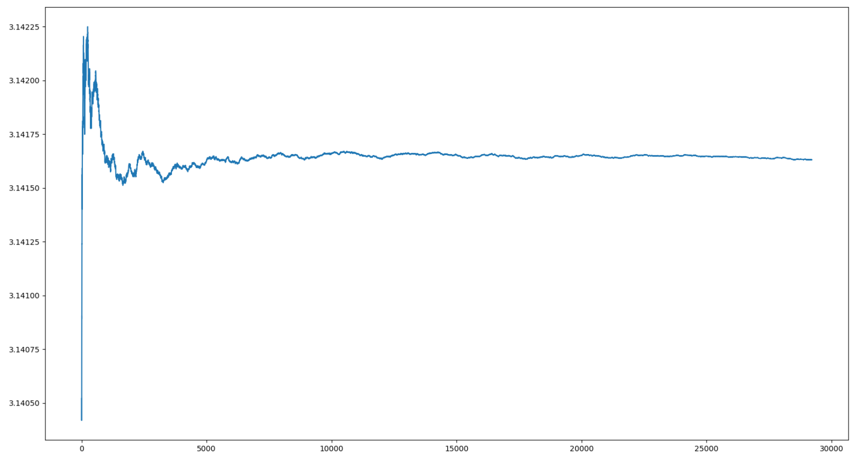


Slika 2: Problem Buffonove igle

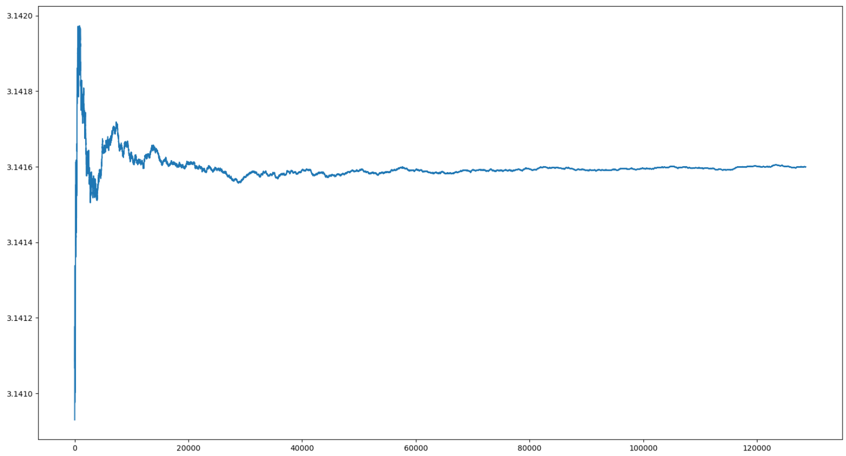
4 Hitrost konvergence

Z metodo Monte Carlo integriranja smo uspeli doseči tri decimalke natančnosti že po nekaj tisoč ponovitvah, četrte pa tudi po več kot milijardi ponovitev ne. Zahtevano število ponovitev eksperimenta narašča s kvadratom željene natančnosti, kar pomeni, da za dvakrat večjo natančnost približka potrebujemo štirikrat več ponovitev. To je posledica centralnega limitnega izreka.

Tudi približek izračuna s simulacijo Buffonove igle se je izboljševal precej počasi, vendar hitreje kot pri integriranju, izračunati nam je uspelo 5 decimalk.



Slika 3: Konvergenca Monte Carlo integriranja.



Slika 4: Konvergenca simulacije problema Buffonove igle.

5 Zaključek

Ugotovili smo, da je rešitev uganke iz uvoda pravzaprav primer Monte Carlo integriranja in lahko z njegovo pomočjo računamo približke za π. Spoznali smo, da Monte Carlo metoda omogoča približno reševanje problemov, ki sicer zahtevajo veliko matematičnega znanja, vendar pa je za dobre približke pogosto potrebno ogromno računske moči.

Literatura

[1] *Buffons needle problem*, v: Wikipedia: The Free Encyclopedia, [ogled 28. 7. 2021], dostopno na https://en.wikipedia.org/wiki/Buffon%27s_needle_problem.

[2] *Monte Carlo integration*, v: Wikipedia: The Free Encyclopedia, [ogled 28. 7. 2021], dostopno na https://en.wikipedia.org/wiki/Monte_Carlo_integration.

[3] *Monte Carlo integration*, [ogled 28. 7. 2021], dostopno na https://cs.dartmouth.edu/wjarosz/publications/dissertation/appendixA.pdf

[4] *Slika 1*, v: Wikipedia: The Free Encyclopedia, [ogled 28. 7. 2021], dostopno na https://en.wikipedia.org/wiki/Monte_Carlo_integration#/media/File:MonteCarloIntegrationCircle.svg.

[5] *Slika 2*, v: Wikipedia: The Free Encyclopedia, [ogled 28. 7. 2021], dostopno na https://en.wikipedia.org/wiki/Buffon%27s_needle_problem#/media/File:Buffon_needle.svg.