

Predstavitev 1. domače naloge

Matija Mačus

Fakulteta za strojništvo

23. oktober 2023



- 1 Uvod
- 2 Oblikovanje programa
- 3 Grafični prikaz in komentar

Uporaba generatorja naključnih števil v Matlabu:

- Naloga od nas zahteva, da s pomočjo metode Monte Carlo izračunamo približno vrednost števila π s pomočjo primerjave ploščine kroga in njemu očrtanega kvadrata. Ploščino ocenimo tako, da generiramo točke in preverimo ali se nahajajo znotraj kroga ali znotra očrtanega kvadrata.
- Kasneje pa z variacijo št. generiranih točk opazujemo numerično napako od točne vrednosti števila π .

Funckija `calcpi()`

Najprej oblikujemo funkcijo `calcpi()`, kateri predhodno določimo št. točk in radij izbranega kroga.

Nato v funkcijo vplejemo podfunkcijo `areapi(nabor točk, R)` in s pomočjo le te določimo ocenjeno vrednost π , napako in koordinate točk znotraj kroga.

Rezultat funkcije calcpi()

Funkcija nam vrne sledeč rezultat:

```
Command Window
>> calc_pi
Ocena n: 3.24
Napaka: 0.098407
Koordinate točk znotraj kroga:
      0      0
      0      0
    -0.1691  0.2434
    -0.5277  0.1854
    -0.2367 -0.6517
    -0.0409  0.5328
      0      0
    -0.5451  0.4318
    -0.1935 -0.7803
    0.6316 -0.5702
      0      0
    0.1314  0.5150
    0.0580 -0.6649
    -0.4096 -0.0265
    -0.7878 -0.2198
    -0.3269 -0.6357
    -0.6067 -0.3261
    -0.0546 -0.1625
    -0.0153 -0.5800
    -0.5981 -0.5055
    -0.0847 -0.4814
```

Slika: Rezultat funkcije calcpi()

Anonimna funkcija

Navodilo od nas zahteva tudi, da oblikujemo anonimno funkcijo, ki ob klicu ovrednoti, ali predhodno naključno izbrani koordinati ležita na krožnici izbranega kroga. Funkcijo definiramo znotraj funkcije `areapi()`:

```
function [Ocena_Pi, error, koordinate_v_krogu] = area_pi(nabor_tock, R)

% Podamo začetno stanje:
V_Krogu = 0;
koordinate_v_krogu = zeros(nabor_tock, 2);

% Anonimna funkcija, ki preverja ali je točka T(x,y) na krožnici:
Na_kroznici = @(x, y) x^2 + y^2 <= R^2;

% Zapišemo for zanko:
for i = 1:nabor_tock
    x = 2 * rand() - 1;
    y = 2 * rand() - 1;
    if Na_kroznici(x, y)
        V_Krogu = V_Krogu + 1;
        koordinate_v_krogu(i, :) = [x, y];
    end
end
Ocena_Pi = 4 * V_Krogu / nabor_tock;
```

Slika: Prikaz anonimne funkcije znotraj kode

Rezultat anonimne funkcije

Funkcija nam vrne sledeč rezultat:

```
x = 0.5; % Določite vrednost za x
y = 0.5; % Določite vrednost za y

% Uporaba funkcije za preverjanje
rezultat = na_loku(x, y);

if rezultat
    disp(['Točka (' , num2str(x), ', ' , num2str(y), ') leži na loku krožnice z radijem R.']);
else
    disp(['Točka (' , num2str(x), ', ' , num2str(y), ') ne leži na loku krožnice z radijem R.']);
end
Točka (0.5, 0.5) leži na loku krožnice z radijem R.
```

Slika: Rezultat anonimne funkcije

Grafična predstavitev

Navodilo od nas zahteva tudi, da oblikujemo funkcijo, ki omogoča izris in grafično vizualizacijo obravnavanih naključno generiranih točk s pomočjo orodja scatter.

```
%Zapišemo funkcijo za izris grafa:

function izris_grafa(St_Tock, R)

    x = 2 * rand(St_Tock, 1) - 1;
    y = 2 * rand(St_Tock, 1) - 1;
    razdalja = sqrt(x.^2 + y.^2);

    tocke_v = razdalja <= R;
    tocke_izven = razdalja > R;

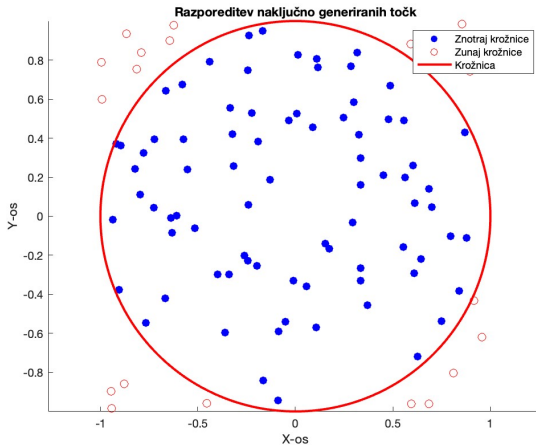
    % Določimo barvo in obliko točk na grafu
    scatter(x(tocke_v), y(tocke_v), 50, 'blue', 'o', 'filled');
    hold on;
    scatter(x(tocke_izven), y(tocke_izven), 50, 'r', 'o');

    Plot_kroznice(R);
    axis equal;
    title('Razporeditev naključno generiranih točk');
    xlabel('X-os');
    ylabel('Y-os');
    legend('Znotraj krožnice', 'Zunaj krožnice', 'Krožnica');
end
```

Slika: Prikaz kode za grafični izris

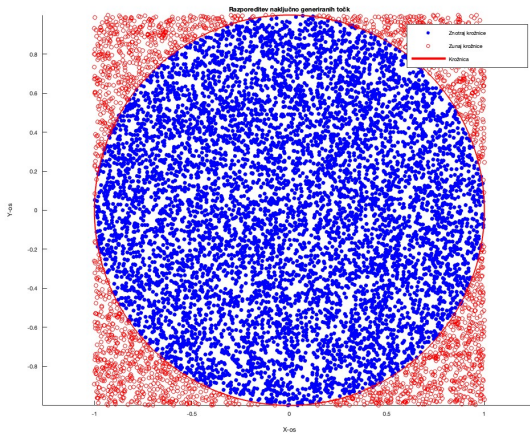
Kot pričakovano, z večajem števila naključno generiranih točk izboljšujemo odstopanje rezultata od točne vrednosti števila π . Kodo `areapi()` lahko nadgradimo tudi tako, da določimo sprejemljivo mejo odstopanja in število generiranih točk iterativno povečujemo, dokler ne dosežemo želene natančnosti.

Grafični prikaz za 100 točk



Slika: Grafični prikaz za 100 točk

Grafični prikaz za 10000 točk



Slika: Grafični prikaz za 10000 točk