

# Predstavitve domače naloge

Maj Voršič

Fakulteta za Strojništvo UL

23. oktober 2023

Univerza v Ljubljani  
Fakulteta za *strojništvo*



## 1 Uvod

## 2 MATLAB

- mcc \_\_pi
- calc\_pi
- Izris točk

## 3 GitHub



V tej nalogi smo uporabili metodo Monte Carlo za izračun približne vrednosti števila  $\pi$ . Razvili smo funkcijo `mcc_pi`, ki je generirala naključne točke znotraj kroga in kvadrata.

Glavna programska datoteka `mcc_pi` je klicala to funkcijo z naraščajočim številom točk. S funkcijo `area_pi` smo primerjali točke znotraj in zunaj kroga ter ocenili vrednost  $\pi$ . Uporabili smo tudi anonimno funkcijo za izris loka krožnice, ki je ločevala notranje in zunanje točke.



Na GitHubu smo ustvarili zasebni repozitorij za predmet "Napredna računalniška orodja" in povabili asistenta in sošolca k sodelovanju. Opravili smo pull request in spremljali potek dela preko statističnih podatkov na GitHubu.



Nazadnje smo pripravili predstavitev v LaTeX orodju Beamer. Predstavitev vključuje naslovnico, kazalo, logotip fakultete, besedilo, sliko s podnapisom in uporabili smo funkcijo "*pause*". Predstavitev je bila naložena na GitHub repozitorij.



## 1 Uvod

## 2 MATLAB

- mcc\_pi
- calc\_pi
- Izris točk

## 3 GitHub



# Generator naključnih števil

S pomočjo metode Monte Carlo smo izračunali približno vrednost števila  $\pi$ . To lahko storimo tako, da primerjamo ploščino kroga in njemu očrtanega kvadrata. Ploščino lahko ocenimo tako, da generiramo veliko število naključnih točk, potem pa preverimo, ali so znotraj kroga ali znotraj kvadrata



Napisali smo kodo za izračun približne vrednosti števila  $\pi$  z "Monte Carlo"metodo :

```
function [points_in_circle , points_in_square] = mcc_pi(num_points)
    % V kvadratu naključno generirane točke
    x = 2 * rand(1, num_points) - 1;
    y = 2 * rand(1, num_points) - 1;

    % Preverimo , ce so točke znotraj kroga
    r = sqrt(x.^2 + y.^2);
    inside_circle = r <= 1;

    % Locimo točke znotraj kvadrata in kroga
    points_in_circle = [x(inside_circle); y(inside_circle)];
    points_in_square = [x; y];
end
```





Napisali smo kodo za izračun približne vrednosti števila  $\pi$  z "Monte Carlo"metodo :

```
function [points_in_circle, points_in_square] = mcc_pi(num_points)
    % V kvadratu naključno generirane točke
    x = 2 * rand(1, num_points) - 1;
    y = 2 * rand(1, num_points) - 1;

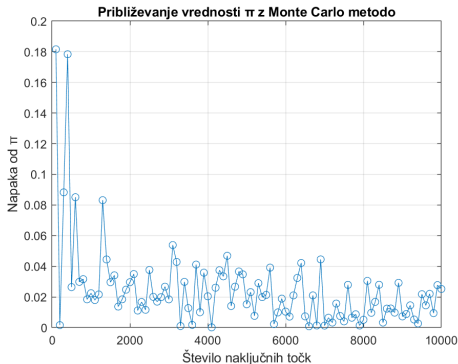
    % Preverimo, ce so točke znotraj kroga
    r = sqrt(x.^2 + y.^2);
    inside_circle = r <= 1;

    % Locimo točke znotraj kvadrata in kroga
    points_in_circle = [x(inside_circle); y(inside_circle)];
    points_in_square = [x; y];
end
```

S klicom `mcc_pi(1000)` nam funkcija za 1000 točk vrne koordinate točk znotraj kroga.



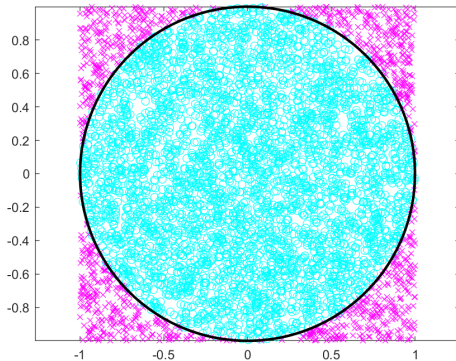
Izdelali smo programsko datoteko calc\_pi, v kateri smo z naraščajočim številom naključnih števil funkcije mcc\_pi spremljali, kako se približujemo številu  $\pi$ .



Slika: Približevanje vrednosti  $\pi$  z Monte Carlo metodo

# Izris točk

Izrisali smo 5000 naključnih točk in jih obarvali z modro, če se točka nahaja znotraj kroga, ter zunanje roza barve. S pomočjo anonimne funkcije smo izrisali črno krožico.



Slika: Izris točk



## 1 Uvod

## 2 MATLAB

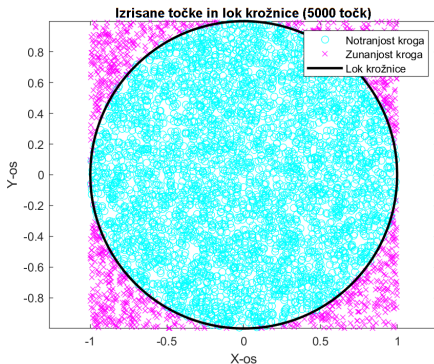
- mcc\_pi
- calc\_pi
- Izris točk

## 3 GitHub



# Uporaba GitHuba

Našo Matlab kodo smo naložili na Github. V naš repozitorij smo povabili sošolca, ki je naši kodi s pomočjo "pull requesta" dodal legendo našem izrisu točk ter imenoval graf ter osi. Rezultat je viden na sliki 3.



Slika: Dodelan izris

