

Predstavitev 1. domače naloge

Gal Stanovnik

Univerza v Ljubljani, Fakulteta za strojništvo

22. oktober 2023

Univerza v Ljubljani
Fakulteta *za strojništvo*



Kazalo

- 1 Funkcijska datoteka
- 2 Programska datoteka
- 3 Anonimna funkcija
- 4 Vizualizacija



Funkcijska datoteka

Najprej ustvarimo funkcijsko datoteko `mcc_pi` in v njej definiramo funkcijo, ki kot rezultat vrne dve matriki z koordinatami naključno generiranih točk. V prvi matriki so točke, ki ležijo znotraj kroga, v drugi pa točke, ki ležijo izven kroga. a_x is different from a_x



Funkcijska datoteka

Najprej ustvarimo funkcijsko datoteko `mcc_pi` in v njej definiramo funkcijo, ki kot rezultat vrne dve matriki z koordinatami naključno generiranih točk. V prvi matriki so točke, ki ležijo znotraj kroga, v drugi pa točke, ki ležijo izven kroga. a_x is different from a_x

Če funkcijo pokličemo res dobimo dve matriki:



```
>> [v1,v2]=mcc_pi(10)
```

```
v1 =
```

0.3115	0.4121
0.6983	-0.4462
0.3575	-0.8057
0.5155	0.6469
0.4863	0.3897
-0.2155	-0.3658
0.3110	0.9004

```
v2 =
```

-0.9286	-0.9363
0.8680	-0.9077
-0.6576	-0.9311



Kazalo

- 1 Funkcijska datoteka
- 2 Programska datoteka
- 3 Anonimna funkcija
- 4 Vizualizacija



Programska datoteka

Najprej ustvarimo programsko datoteko `calc_pi`. Ker nas zanima vpliv števila generiranih točk na točnost dobljenega približka števila π , lahko s `for` zanko izračunamo približek pri različnem številu točk:

```
priblizki_pi = [];  
napake=[];  
for b = 1:12  
    n=2^b;  
    [koor1,koor2]=mcc_pi(n);  
    [zadnji_prib,zadnja_nap] = area_pi(koor1,koor2);  
    priblizki_pi=[priblizki_pi;zadnji_prib];  
    napake = [napake;zadnja_nap];  
end
```



Programska datoteka

Najprej ustvarimo programsko datoteko `calc_pi`. Ker nas zanima vpliv števila generiranih točk na točnost dobljenega približka števila π , lahko s `for` zanko izračunamo približek pri različnem številu točk:

```
priblizki_pi = [];  
napake=[];  
for b = 1:12  
    n=2^b;  
    [koor1,koor2]=mcc_pi(n);  
    [zadnji_prib,zadnja_nap] = area_pi(koor1,koor2);  
    priblizki_pi=[priblizki_pi;zadnji_prib];  
    napake = [napake;zadnja_nap];  
end
```

Pri tem smo klicali funkcijo `area_pi`, ki izračuna približek in napako približka. Definirali smo jo čisto na dnu programske datoteke.

Kazalo

- 1 Funkcijska datoteka
- 2 Programska datoteka
- 3 Anonimna funkcija
- 4 Vizualizacija



Anonimna funkcija

Definirali smo anonimno funkcijo, ki prejme radij željene krožnice in vektor točk x . Vrne matriko vrednosti, v prvem stolpcu so vrednosti za pozitivne y koordinate, v drugi vrstici pa so vrednosti negativnih y -ov.

```
x=linspace(-1,1,1000);  
yy=@(x)[sqrt(1-x.^2);-sqrt(1-x.^2)];
```



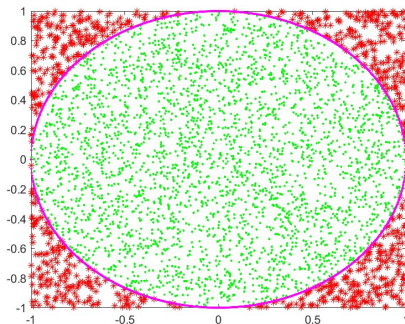
Kazalo

- 1 Funkcijska datoteka
- 2 Programska datoteka
- 3 Anonimna funkcija
- 4 Vizualizacija



Vizualizacija

Na en graf izrižemo točke krožnice, točke zunaj in točke znotraj kroga.



Slika: Točke izven krožnice so obarvane rdeče, točke znotraj pa zeleno. Točke na krožnici so vijolične.