



# Predstavitev pri predmetu - NRO

## 1.DN

Gašper Kostelec

Univerza v Ljubljani  
Fakulteta za strojništvo  
RRP, 3.letnik

23. oktober 2023

# Kazalo

- 1 Uvod in koda
- 2 Grafi
- 3 GitHub in Beamer
- 4 Zaključek

# Kazalo

- 1 Uvod in koda
- 2 Grafi
- 3 GitHub in Beamer
- 4 Zaključek

# Uvod in kratek opis funkcij v kodi

- S pomočjo metode Monte Carlo smo v Matlab-u zapisali kodo(funkcije). Kodo smo nato nadgradili s sošolcem v Github-u in na koncu izdelali predstavitev v okolju Beamer.

# Uvod in kratek opis funkcij v kodi

- S pomočjo metode Monte Carlo smo v Matlab-u zapisali kodo(funkcije). Kodo smo nato nadgradili s sošolcem v Github-u in na koncu izdelali predstavitev v okolju Beamer.
- `calc-pi()` - v tej funkciji je zapisano začetno število točk, radij in kliče naslednjo funkcijo

# Uvod in kratek opis funkcij v kodi

- S pomočjo metode Monte Carlo smo v Matlab-u zapisali kodo(funkcije). Kodo smo nato nadgradili s sošolcem v Github-u in na koncu izdelali predstavitev v okolju Beamer.
- `calc-pi()` - v tej funkciji je zapisano začetno število točk, radij in kliče naslednjo funkcijo
- `raziskuje-priblizke` - ta funkcija pokliče vse potrebne funkcije in se izvede tolikokrat kot je zahtevano število-približkov. V tej funkciji je uporabljena tudi 'for' zanka.

# Uvod in kratak opis funkcij v kodi

- S pomočjo metode Monte Carlo smo v Matlab-u zapisali kodo(funkcije). Kodo smo nato nadgradili s sošolcem v Github-u in na koncu izdelali predstavitev v okolju Beamer.
- `calc-pi()` - v tej funkciji je zapisano začetno število točk, radij in kliče naslednjo funkcijo
- `raziskuje-priblizke` - ta funkcija pokliče vse potrebne funkcije in se izvede tolikokrat kot je zahtevano število-približkov. V tej funkciji je uporabljena tudi 'for' zanka.
- `izrisi-graf()` - pokliče svoji dve potrebni funkciji, in nariše graf za vsak približek posebej

```
function calc_pi()  
% na podlagi začetnega števila točk, radija in naknadnih povečanj začetnih  
% točk, izpiše izračunano vrednost  $\pi$  in napako oz. odstopanje od števila  $\pi$ , ki je shranjeno v Matlabu kot \pi.  
n = 100;  
r = 1;  
raziskuje_priblizke_pi(n, r);  
end
```

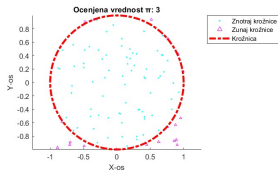
Slika 1: Začetna funkcija celotne datoteke

# Kazalo

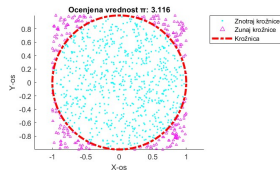
- 1 Uvod in koda
- 2 Grafi**
- 3 GitHub in Beamer
- 4 Zaključek



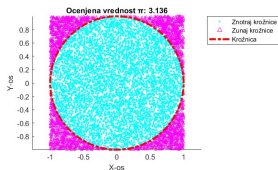
# Prikaz grafov v odvisnosti od $n$



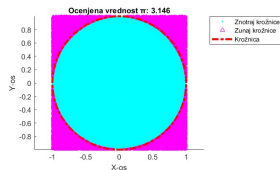
Slika 2: graf  $n=100$



Slika 3: graf  $n*10$



Slika 4: graf  $n*100$



Slika 5: graf  $n*1000$

# Kazalo

- 1 Uvod in koda
- 2 Grafi
- 3 GitHub in Beamer**
- 4 Zaključek

# Opis dela

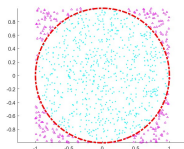
- 1 k sodelavanju na projektu sem povabil asistenta in enega izmed sošolcev

# Opis dela

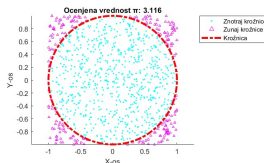
- 1 k sodelavanju na projektu sem povabil asistenta in enega izmed sošolcev
- 2 sošolec je nato dopolnil nekaj podatkov glede izrisa grafa, katere sem po pregledu sprejel

# Opis dela

- 1 k sodelavanju na projektu sem povabil asistenta in enega izmed sošolcev
- 2 sošolec je nato dopolnil nekaj podatkov glede izrisa grafa, katere sem po pregledu sprejel
- 3 sledila je priprava predstavitve v okolju Beamer



(a) Graf brez naslova, legende in poimenovanja osi



(b) Ustrezno opremljen graf

**Slika 6:** Primerjava grafov pred in po sodelovanju s sošolcem

# Kazalo

- 1 Uvod in koda
- 2 Grafi
- 3 GitHub in Beamer
- 4 Zaključek**

# Tabela

približek	število točk	ocenjena vrednost	odstopanje
1	100	3	0.14159

# Tabela

približek	število točk	ocenjena vrednost	odstopanje
1	100	3	0.14159
2	1000	3.116	0.025593



# Tabela

približek	število točk	ocenjena vrednost	odstopanje
1	100	3	0.14159
2	1000	3.116	0.025593
3	10000	3.136	0.0055927

# Tabela

približek	število točk	ocenjena vrednost	odstopanje
1	100	3	0.14159
2	1000	3.116	0.025593
3	10000	3.136	0.0055927
4	100000	3.146	0.0044473

**Tabela 1:** Prikaz rezultata enega izmed poiskusov iskanja vrednosti  $\pi$

S pomočjo zgornje tabele lahko opazimo, da z naraščajnem številu točk, dobimo tudi boljši približek vrednosti  $\pi$ .