

# Prva domača naloga pri predmetu Napredna računalniška orodja

Tina Jeglič  
23211172

October 21, 2023

Univerza v Ljubljani  
Fakulteta *za strojništvo*



# Kazalo

Približek števila  $\pi$  z metodo Monte Carlo

Funkcijska in programska datoteka

Vizualizacija

## Definicija naloge

Z metodo Monte Carlo smo v programu Matlab izračunali približno vrednost števila  $\pi$ .

Postopek domače naloge:

- definicija funkcijske datoteke s funkcijo `mcc_pi.m` z enim vhodnim parametrom (število naključnih točk),
- programska datoteka `calc_pi.m`, ki vključuje prejšnjo funkcijsko in novo funkcijo, ki primerja število točk znotraj in zunaj kroga in tako oceni vrednosti  $\pi$  in odstopanje od prave vrednosti,
- vključitev anonimne funkcije za izračun točk na loku krožnice,
- vizualizacija točk znotraj in zunaj kroga ter krožnice, ki jih ločuje

## Funkcijska datoteka

Funkcijska datoteka ima enako ime kot prva funkcija v datoteki, gledano z vrha. Spremenljivke v tovrstni datoteki so lokalne in niso dostopne iz drugih funkcij. Ta del programa za izračun razvijemo v obliki funkcijske datoteke z imenom `mcc_pi.m`, ki ima en vhodni parameter (število naključnih točk). Funkcija ob klicu vrne koordinate točk znotraj kroga in koordinate točk znotraj kvadrata.

```
function [znotraj_kroga, znotraj_kvadrata] = mcc_pi(n)
    % Funkcija za Monte Carlo simulacijo za računanje približka števila  $\pi$ .

    % Generiraj naključne točke v kvadratu [0, 1] x [0, 1]
    x = rand(1, n);
    y = rand(1, n);

    % Preveri, katere točke so znotraj kroga
    notraje_tocke = (x.^2 + y.^2 <= 1);

    % Koordinate točk znotraj kroga in kvadrata
    znotraj_kroga = [x(notraje_tocke); y(notraje_tocke)];
    znotraj_kvadrata = [x; y];

end
```

# Programska datoteka

Programska datoteka je običajno krovna datoteka, ki definira začetne vrednosti in kliče funkcije potrebne za rešitev problema. Ustvarite programsko datoteko `mcc_pi.m`.

```
function calc_pi()
    n=5000;
    r=1;
    [pi_priblizek, napaka] = area_pi(n);
    disp('Koordinate točk znotraj kroga in znotraj kvadrata v vektorski obliki: odstrani %komentar')
    %[znotraj_kroga, znotraj_kvadrata] = mcc_pi(n)
    disp(['Približek števila pi:', num2str(pi_priblizek)])
    disp(['Odstopanje:', num2str(napaka)])
    figure(n)
end
```

Figure: Prvi del kode

## Koordinate naključnih točk

Funkcijo smo zakomentirali, saj nam vrne zelo dolg output. Fotografija spodaj prikazuje koordinate le nekaj naključnih točk. Opazimo, da so vse pozitivno predznačene, saj smo se pri računanju približka  $\pi$  omejili na prvi kvadrant.

Columns 1.574 through 1.586

0.2876	0.6426	0.8549	0.0503	0.5389	0.4863	0.1828	0.5764	0.4348	0.4473	0.6683	0.2134	0.4971
0.8579	0.6342	0.9928	0.8738	0.1946	0.1218	0.1563	0.2062	0.8023	0.9262	0.2724	0.1112	0.2720

Columns 1.587 through 1.599

0.4440	0.3449	0.8087	0.7068	0.8838	0.6071	0.3774	0.2045	0.5915	0.4819	0.5948	0.0368	0.4068
0.0250	0.1879	0.7173	0.3688	0.5900	0.5096	0.9906	0.5408	0.3477	0.0333	0.2212	0.9451	0.8126

## Vizualizacija rezultatov

Na prejšnji strani smo prikazali koordinate točk, ki smo jih naključno generirali z ukazom `rand`. Kot rezultat dobimo še našo ocenjeno vrednost števila  $\pi$  in napako pri računanju. S spreminjanjem števila naključnih točk  $n$  ugotovimo, da je z večjim številom točk tudi rezultat bolj točen. Spodaj prikazani približek je generiran na podlagi 5000 točk.

```
Približek števila pi:3.112
```

```
Odstopanje:0.029593
```

## Vizualizacija rezultatov

Na prejšnji strani smo prikazali koordinate točk, ki smo jih naključno generirali z ukazom `rand`. Kot rezultat dobimo še našo ocenjeno vrednost števila  $\pi$  in napako pri računanju. S spreminjanjem števila naključnih točk  $n$  ugotovimo, da je z večjim številom točk tudi rezultat bolj točen. Spodaj prikazani približek je generiran na podlagi 5000 točk.

```
Približek števila pi:3.112  
Odstopanje:0.029593
```

