

# WYDZIAŁ NAUK ŚCISŁYCH I TECHNICZNYCH

## Symulacje Komputerowe

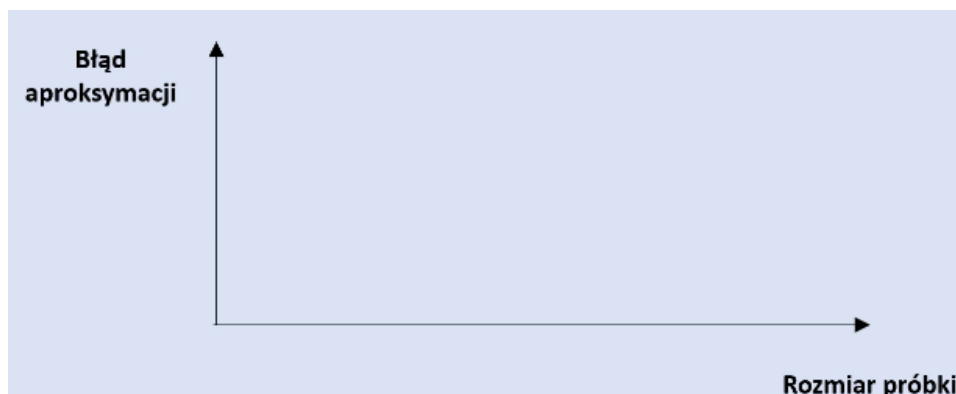
Sprawozdanie “Symulacja Monte Carlo”

Adam Talarczyk, Mateusz Wrzoł

Uniwersytet Śląski, Sosnowiec, 2021

## 1 Zadanie 1

Należy zmodyfikować kod dla aproksymacji stałej  $\pi$ , aby sprawdzić jak rozmiar próbki wpływa na błąd aproksymacji. Błąd aproksymacji obliczamy jako wartość bezwzględną różnicy, pomiędzy aproksymacją  $\pi$  i wartością rzeczywistą  $\pi$  (3.14159265). Należy przygotować wykres [Rysunek 1].



Rysunek 1: Przykład wykresu

### 1.1 Rozwiązanie

Opis rozwiązania

### 1.2 Kod źródłowy

```
1 # Title      : Monte Carlo Simulation
2 # Objective  :
3 # Created by : Adam Talarczyk , Mateusz Wrzol
4 # Created on: 16.04.2021
5
6 library("xlsx")
7 source('pi/approximation.R')
8 source('pi/avarage_absolute_difference.R')
9
10 calculate_and_make_plot <- function(steps , sequences) {
11   diff.vector <- NULL
12   dataset <- data.frame()
13
14   for (runs in steps)
15   {
16     difference <- avarage_absolute_difference(runs ,
17       sequences)
18     dataset <- rbind(dataset , difference)
19     diff.vector <- append(diff.vector , difference[1, 'avg_
20       diff '])
21   }
22 }
```

```

19 }
20
21 export_dataset(dataset)
22 draw_plot(steps, diff.vector)
23 }
24
25 export_dataset <- function(dataset){
26   write.xlsx(dataset, file = "pi/export/data.xlsx",
27     sheetName="PI")
28 }
29 draw_plot <- function(steps, results) {
30   plot(steps, results, xlab = 'Rozmiar probki', ylab = 'Bład
    aproksymacji', col = 'black')
31 }
32
33 # calculate_and_make_plot(seq(0, 1000000, by = 1000))
34 sequence <- c(1,500,1000,10000,50000,100000,500000,1000000)
35 calculate_and_make_plot(sequence, 10)

```

Listing 1: Plik main.R - wywołanie głównej funkcji

```

1 # Title      : Avarage absolute difference
2 # Objective  :
3 # Created by: Adam Talarczyk, Mateusz Wrzol
4 # Created on: 10.04.2021
5
6 avarage_absolute_difference <- function(runs, sequences =
7   100) {
8   pi.vector <- NULL
9   for (i in seq(1, sequences, by = 1))
10    {
11      mc.pi <- approximation(runs)
12      pi.vector <- append(pi.vector, mc.pi)
13    }
14
15   avarage_pi <- mean(pi.vector, trim = 0, na.rm = FALSE)
16   avarage_difference <- abs(3.14159265 - avarage_pi)
17
18   dataset <- data.frame(
19     step = runs,
20     pi_value = pi.vector,
21     avg_pi = avarage_pi,
22     avg_diff = avarage_difference
23   )
24 }

```

Listing 2: Plik avarage\_absolute\_difference.R - odpowiedzialny za wyliczanie błędu aproksymacji PI

```

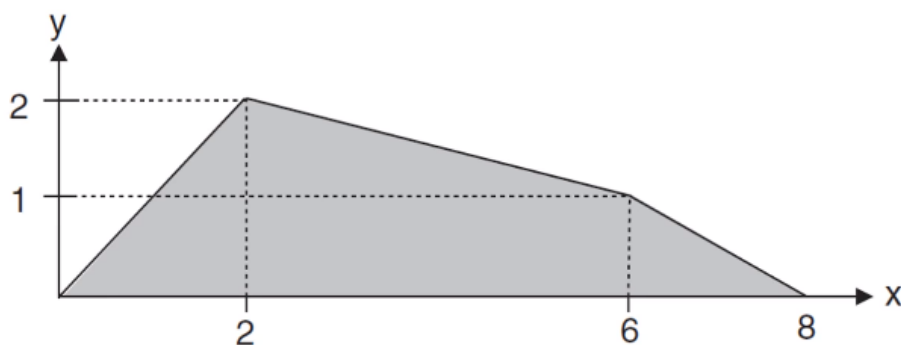
1 # Title      : PI approximation
2 # Objective  :
3 # Created by: Adam Talarczyk, Mateusz Wrzol
4 # Created on: 10.04.2021
5
6 approximation <- function(runs = 1000) {
7   xs <- runif(runs, min = -0.5, max = 0.5)
8   ys <- runif(runs, min = -0.5, max = 0.5)
9   in.circle <- xs^2 + ys^2 <= 0.5^2
10  mc.pi <- (sum(in.circle) / runs) * 4
11 }

```

Listing 3: Plik approximation.R - odpowiedzialny za obliczanie przybliżenia

## 2 Zadanie 2

Zaprogramować symulację Monte Carlo (np. w języku R), która pozwoli obliczyć pole powierzchni szarego obszaru, przedstawionego na poniższym rysunku [Rysunek 2]. Obliczyć błąd uzyskanego wyniku.



Rysunek 2: Figura

### 2.1 Rozwiązanie

Opis rozwiązania

### 2.2 Kod źródłowy

Listingi