

WYDZIAŁ NAUK ŚCISŁYCH I TECHNICZNYCH

Symulacje Komputerowe

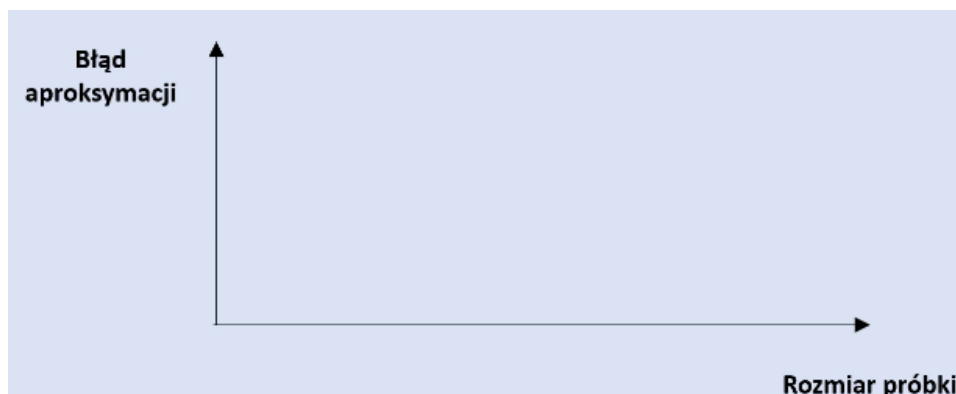
Sprawozdanie “Symulacja Monte Carlo”

Adam Talarczyk, Mateusz Wrzoł

Uniwersytet Śląski, Sosnowiec, 2021

1 Zadanie 1

Należy zmodyfikować kod dla aproksymacji stałej π , aby sprawdzić jak rozmiar próbki wpływa na błąd aproksymacji. Błąd aproksymacji obliczamy jako wartość bezwzględną różnicy, pomiędzy aproksymacją π i wartością rzeczywistą π (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/avarage_absolute_difference.R')
8
9 calculate_and_make_plot <- function(steps , sequences) {
10   diff.vector <- NULL
11   dataset <- data.frame()
12
13   for (runs in steps)
14   {
15     difference <- avarage_absolute_difference(runs ,
16       sequences)
17     dataset <- rbind(dataset , difference)
18     diff.vector <- append(diff.vector , difference[1, 'avg_diff'])
19   }
20 }
```

```

19
20 export_dataset(dataset)
21 draw_plot(steps, diff.vector)
22 }
23
24 export_dataset <- function(dataset){
25   write.xlsx(dataset, file = "pi/export/data.xlsx",
26     sheetName="PI")
27 }
28 draw_plot <- function(steps, results) {
29   plot(steps, results, xlab = 'Rozmiar probki', ylab = 'Bład
    aproksymacji', col = 'black')
30 }
31
32 # calculate_and_make_plot(seq(0, 1000000, by = 1000))
33 sequence <- c(1,500,1000,10000,50000,100000,500000,1000000)
34 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 source('pi/approximation.R')
7
8 avarage_absolute_difference <- function(runs, sequences =
9   100) {
10   pi.vector <- NULL
11   for (i in seq(1, sequences, by = 1))
12   {
13     mc.pi <- approximation(runs)
14     pi.vector <- append(pi.vector, mc.pi)
15   }
16
17   avarage_pi <- mean(pi.vector, trim = 0, na.rm = FALSE)
18   avarage_difference <- abs(3.14159265 - avarage_pi)
19
20   dataset <- data.frame(
21     step = runs,
22     pi_value = pi.vector,
23     avg_pi = avarage_pi,
24     avg_diff = avarage_difference
25   )
26 }

```

Listing 2: Plik avarage_absolute_difference.R - odpowiedzialny za wyliczanie błędu aproksymacji PI

```

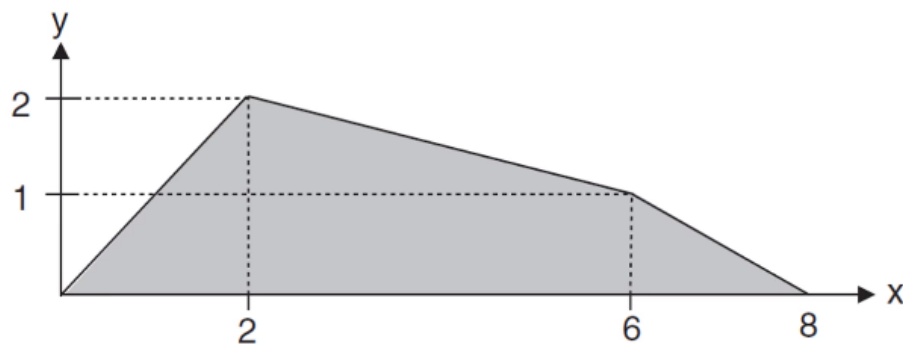
1 # Title      : PI approximation
2 # Objective  :
3 # Created by: Adam Talarczyk, Mateusz Wrzol
4 # Created on: 10.04.2021
5
6 approximation <- function(runs = 1000) {
7   xs <- runif(runs, min = -0.5, max = 0.5)
8   ys <- runif(runs, min = -0.5, max = 0.5)
9   in.circle <- xs^2 + ys^2 <= 0.5^2
10 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

```

1 # Title      : Monte Carlo Simulation
2 # Objective  :
3 # Created by: Adam Talarczyk, Mateusz Wrzol
4 # Created on: 15.04.2021
5
6 figure <- function(runs) {
7   xs <- runif(runs, min = 0, max = 8)
8   ys <- runif(runs, min = -0, max = 2)

```

```

9
10 in.r_triangle <- xs <= 2 & ys <= 2 & ys <= xs
11 in.mid_square <- xs <= 6 & xs >= 2 & ys <= 1
12 in.l_triangle <- xs >= 6 &
13   xs <= 8 &
14   ys <= 1 &
15   ys <= (-0.5 * xs + 4)
16 in.mid_triangle <- xs >= 2 &
17   xs <= 6 &
18   ys >= 1 &
19   ys <= (-0.25 * xs + 2.5)
20
21 in_all_figures <- in.r_triangle +
22   in.mid_square +
23   in.mid_triangle +
24   in.l_triangle
25
26 mc_figure_field <- (sum(in_all_figures) / runs) * 16
27 mc_exact_figure_field <- (2 * 2 / 2) + (1 * 4 / 2) + (1 *
28   2 / 2) + (1 * 4)
29
30 mc.error <- abs(mc_exact_figure_field - mc_figure_field)
31 plot(xs, ys,
32       col = ifelse(in_all_figures, "blue", "grey"),
33       xlab = '', ylab = '', asp = 1, pch = '.',
34       main = paste("Figure field =", mc_figure_field, ",
35       Result error =", mc.error))
36 }
37
38 figure(1000000)

```

Listing 4: Plik figure.R - odpowiedzialny obliczenie pola i wygenerowanie wykresu