

# Wyznaczanie liczby $\pi$

Kacper Kingsford

Analiza Numeryczna (M) - Zadanie P1.4  
22 października 2020

## Streszczenie

Liczba  $\pi$  (zwana inaczej ludolfiną) jest jedną z pierwszych odkrytych przez człowieka liczb niewymiernych. Oznacza stosunek długości obwodu koła do długości jego średnicy. Jest też pierwszą literą greckiego słowa "perimetron", oznaczającego obwód. Datę 14 marca na Dzień Liczby Pi wybrano nieprzypadkowo - w Stanach Zjednoczonych zapisuje się ją jako 3.14, tak jak przybliżoną wartość ludolfiny.

Ze względu na częstotliwość występowania liczby  $\pi$  w matematyce, istotne jest to, żeby móc ją dokładnie aproksymować. Niniejsze sprawozdanie jest podsumowaniem eksperymentu numerycznego polegającego na obliczaniu liczby  $\pi$  przy pomocy szeregu  $\pi = 4 \sum_{k=0}^{\infty} (-1)^k (2k+1)^{-1}$ , sumując składniki w porządku naturalnym oraz odwrotnym.

## **Spis treści**

# 1 Definicja liczby $\pi$

$\pi$  (czyt. pi), ludolfina, stała Archimedesa – stosunek obwodu koła (czyli długości okręgu) do długości jego średnicy; stosunek ten jest niezależny od wyboru koła, bowiem każde dwa koła są podobne. Liczba  $\pi$  nazywana jest czasami stałą Archimedesa w uznaniu zasług Archimedesa z Syrakuz, który jako pierwszy badał własności i znaczenie w matematyce tej liczby; określenie ludolfina pochodzi od Ludolpha van Ceulena, który zyskał sławę przedstawiając tę liczbę z dokładnością do 35 miejsc po przecinku.

Liczba  $\pi$  z dokładnością do 100 miejsc po przecinku:

$\pi \approx 3.141592653589793238462643383279502884197169399375105820974944592307816406286208998628034825$

## 1.1 Obliczanie $\pi$ jako sumy szeregu $\sum_{k=0}^{\infty} (-1)^k (2k+1)^{-1}$

### Twierdzenie 1

Liczbę  $\pi$  możemy zdefiniować następująco:

$$\pi = 4 \sum_{k=0}^{\infty} (-1)^k (2k+1)^{-1}$$

Dowód:

Zauważmy, że:

$$\begin{aligned} \frac{\pi}{4} &= \arctg(1) = \int_0^1 \frac{1}{1+x^2} dx = \int_0^1 \left( \sum_{k=0}^n (-1)^k x^{2k} + \frac{(-1)^{n+1} x^{2n+2}}{1+x^2} \right) dx \\ &= \left( \sum_{k=0}^n \frac{(-1)^k}{2k+1} \right) + (-1)^{n+1} \left( \int_0^1 \frac{x^{2n+2}}{1+x^2} dx \right). \end{aligned}$$

Oszacujmy całkę znajdująca się w poprzednim wierszu:

$$0 \leq \int_0^1 \frac{x^{2n+2}}{1+x^2} dx \leq \int_0^1 x^{2n+2} dx = \frac{1}{2n+3} \rightarrow 0 \text{ gdy } n \rightarrow \infty.$$

Z twierdzenia o trzech ciągach otrzymujemy, że gdy  $n \rightarrow \infty$  to:

$$\frac{\pi}{4} = \sum_{k=0}^{\infty} \frac{(-1)^k}{2k+1} \iff \pi = 4 \sum_{k=0}^{\infty} \frac{(-1)^k}{2k+1}.$$

Wiedząc, że  $\pi = 4 \sum_{k=0}^{\infty} \frac{(-1)^k}{2k+1}$ , wiemy, że dla dużych wartości  $n$  zachodzi

$$\pi \approx 4 \sum_{k=0}^n \frac{(-1)^k}{2k+1}.$$

Wykorzystując tę obserwację można sformułować algorytm<sup>1</sup>, który będzie wyznaczał przybliżaną wartość liczby  $\pi$ .

<sup>1</sup>Wszystkie programy w niniejszym sprawozdaniu napisane są w języku Julia w wersji 1.5.2.

Program 1: Funkcja w języku Julia obliczająca  $4 \sum_{k=0}^n \frac{(-1)^k}{2k+1}$

```

1 function suma(n)
2     suma = 0.0 # zaczynamy sumowanie od 0
3     for k in 0:n
4         if k%2==0 # jesli k jest parzyste to dodajemy kolejny składnik do sumy
5             suma+=1/(2*k + 1)
6         else # w przeciwnym przypadku odejmujemy go
7             suma-=1/(2*k + 1)
8         end
9     end
10    4.0*suma
11 end

```

## 1.2 Błąd sumowania $n$ liczb

Na początku wprowadźmy definicje i twierdzenie:

### Definicja 1

Precyją arytmetyki danego komputera nazywamy liczbę

$$u = \frac{1}{2}2^{-t},$$

gdzie  $t \in \mathbb{N}$  jest liczbą bitów przeznaczonych na mantysę.

### Twierdzenie 2

Jeśli  $|\epsilon_k| \leq u$  i  $\rho_k = \pm 1$  dla  $k = 1, 2, \dots, n$  oraz  $nu < 1$ , to zachodzi równość:

$$\prod_{k=1}^n (1 + \epsilon_k)^{\rho_k} = 1 + \alpha_n$$

gdzie

$$|\alpha_n| \leq \frac{nu}{1 - nu}$$

Chcemy dodać do siebie  $n$  wyrazów ciągu. Dla uogólnienia nazwijmy je  $a_1, a_2, \dots, a_n$ .

$$\sum_{k=0}^n a_k = a_1 + a_2 + \dots + a_n$$

W komputerze powyższa suma obliczy się do:

$$\begin{aligned}
& fl\left(\sum_{k=0}^n a_k\right) = fl\left(a_1 + a_2 + \dots + a_n\right) = ((\dots((a_1 + a_2)(1 + \epsilon_2) + a_3)(1 + \epsilon_3)\dots) + a_n)(1 + \epsilon_n) \\
& = a_1 \prod_{k=1}^n (1 + \epsilon_k) + a_2 \prod_{k=2}^n (1 + \epsilon_k) + \dots + a_n \prod_{k=n}^n (1 + \epsilon_k) = \sum_{i=1}^n \left( a_i \prod_{k=i}^n (1 + \epsilon_k) \right) \text{ gdzie } |\epsilon_k| \leq u \text{ dla } k = 2, 3, \dots, n
\end{aligned}$$

Oczywiście  $\epsilon_1 = 0$ . Podstawmy:

$$\prod_{k=i}^n (1 + \epsilon_k) = 1 + \alpha_i$$

Otrzymujemy:

$$fl\left(\sum_{i=1}^n a_i\right) = \sum_{i=1}^n a_i(1 + \alpha_i)$$

Wtedy z Twierdzenia 2 otrzymujemy, że:

$$|\alpha_i| \leq \frac{(n+1-i)u}{1-(n+1-i)u} \text{ gdzie } i = 2, 3, \dots n$$

Wywnioskować można stąd, że największe zaburzenie będzie występowało dla pierwszego składnika, analogicznie najmniejsze dla ostatniego.

Spójrzmy teraz na poniższe sumy:

$$4 \sum_{k=0}^n \frac{(-1)^k}{2k+1} \quad \text{oraz} \quad 4 \sum_{k=n}^0 \frac{(-1)^k}{2k+1}.$$

Różnią się one tylko kolejnością sumowania składników.

Korzystając z powyższych obserwacji możemy się spodziewać, że w komputerze dokładniejszym wynikiem będzie suma druga. Będzie tak, z uwagi na to, że największy błąd będzie występował podczas sumowania dwóch najmniejszych liczb sumy, analogicznie z kolejnymi błędami.

## 2 Opis eksperymentu i wyniki

Obliczenia zostały wykonane w języku Julia (wersja 1.5.2.) przy użyciu 32-bitowej arytmetyki (32 bity przewinaczone na reprezentację mantysy).

Funkcje *suma\_rosnaca(n)* oraz *suma\_malejaca(n)* zostały uruchomione dla różnych wartości  $n$ , a następnie została porównana precyzja, z jaką przybliżają liczbę  $\pi$  poprzez porównanie wyniku do wbudowanej stałej *MathConstants.pi*.

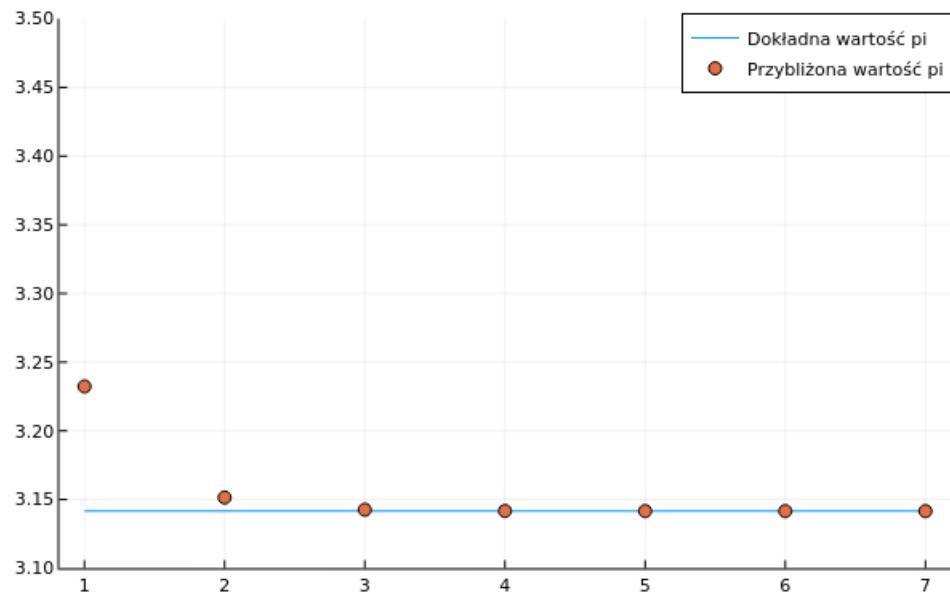
Obliczona została liczba cyfr znaczących oraz iloraz błędów każdego przybliżenia przy użyciu wzoru

$$\lfloor -\log_{10}(\delta_\pi) \rfloor$$

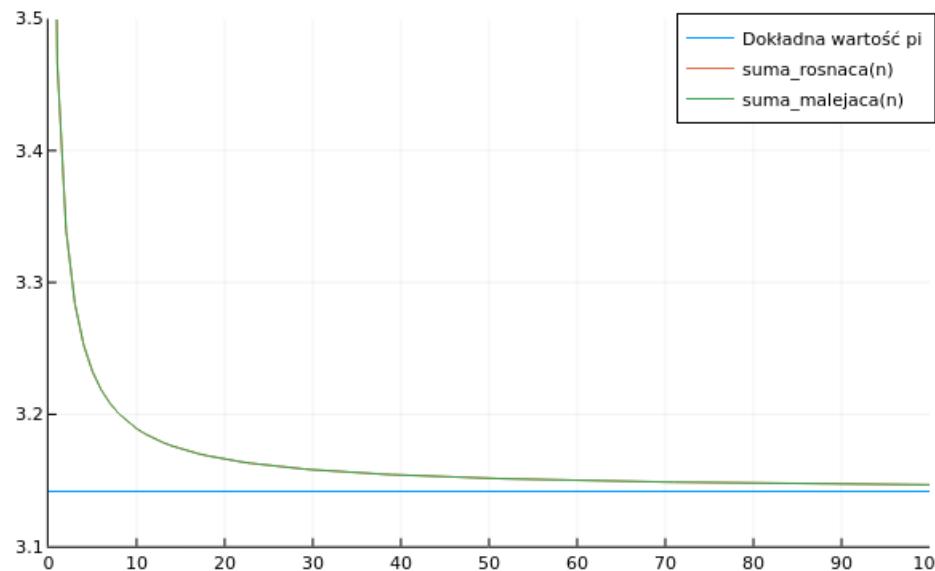
gdzie

$$\delta_\pi = \left| \frac{\Delta\pi}{\pi} \right| \text{ i } \Delta\pi = \pi - \tilde{\pi} \text{ gdzie } \tilde{\pi} \text{ oznacza otrzymane przybliżenie } \pi.$$

## 2.1 Wyniki obliczeń oraz ich analiza



Rysunek 1: Wykres przedstawiający wyniki funkcji  $suma\_rosnaca(n)$  dla  $n = 10^1, 10^2, \dots, 10^7$

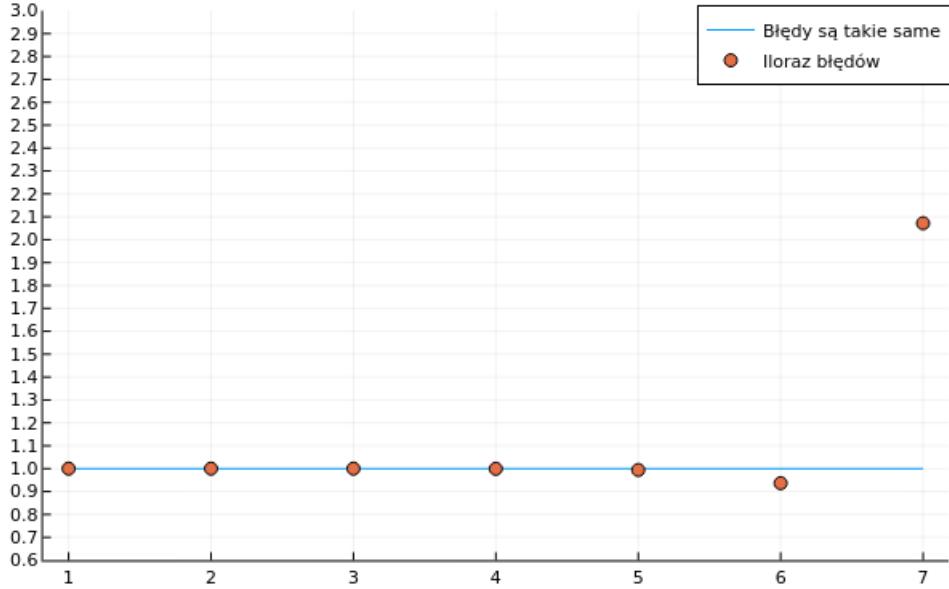


Rysunek 2: Wykres przedstawiający wyniki funkcji  $suma\_rosnaca(n)$  oraz  $suma\_malejaca(n)$

Zauważmy, że powyższe wykresy są tak do siebie podobne, że nie możemy określić różnic między nimi. Spójrzmy więc na wykres ilorazów błędów wspomnianych wyżej funkcji.

$n$	$\lfloor -\log_{10}(\delta_\pi) \rfloor$	$\delta_{\pi_1} =  \frac{\Delta\pi}{\pi} $	$\delta_{\pi_2} =  \frac{\Delta\pi}{\pi} $
$10^1$	1	0.028	0.028
$10^2$	2	0.0031	0.0031
$10^3$	3	0.00031	0.00031
$10^4$	4	$3.17 \cdot 10^{-5}$	$3.17 \cdot 10^{-5}$
$10^5$	5	$3.13 \cdot 10^{-6}$	$3.15 \cdot 10^{-6}$
$10^6$	6	$2.72 \cdot 10^{-7}$	$2.90 \cdot 10^{-7}$
$10^7$	8	$8.24 \cdot 10^{-9}$	$4.01 \cdot 10^{-9}$

Tablica 1: Wyniki programu dla wybranych wartości  $n$ .



Rysunek 3: Wykres przedstawiający iloraz błędów funkcji  $suma\_rosnaca(n)$  oraz  $suma\_rosnaca(n)$ ,  $|\frac{\delta_{\pi_1}}{\delta_{\pi_2}}|$  dla  $n = 10^1, 10^2, \dots, 10^7$

Jak widać błędy funkcji dla  $n = 10^1, 10^2, \dots, 10^6$  są bardzo podobne. Zauważmy, że dla  $n = 10^7$  iloraz błędów funkcji jest równy  $|\frac{\delta_{\pi_1}}{\delta_{\pi_2}}| \approx 2.1$ , wtedy funkcja  $suma\_rosnaca(n)$  przybliża liczbę  $\pi$  z dokładnością dwukrotnie mniejszą niż funkcja  $suma\_malejaca(n)$ .

Powyższe obserwacje potwierdzają naszą hipotezę - dokładniejszy wynik otrzymamy sumując sumę  $4 \sum_{k=n}^0 \frac{(-1)^k}{2k+1}$  w porządku odwrotnym.