

Wykonał: Radosław Smoter

Grupa: 14

Nr: 27

Numer zadania: 3

Przykład: 62

Prowadzący: Prof. dr hab. inż.
Volodymyr Samotyy

Politechnika Krakowska

Wydział Inżynierii Elektrycznej i Komputerowej

Sprawozdanie: Wstęp do Programowania

Spis treści

Polecenie.....	1
Kod programu.....	2
Wyniki.....	4
Opis Programu.....	5
Funkcja doSum().....	5
Funkcja countFunc().....	5
Wnioski.....	6

Polecenie

Obliczyć wartość funkcji podanej w postaci rzędu. Wyniki obliczeń zapisać do pliku tekstowego. Narysować wykres $y(x)$.

Suma: $\sum_{k=1}^{\infty} \frac{1}{2^k} \left(\sin\left(\frac{x}{2^k}\right) + \cos\left(\frac{x}{2^k}\right) \right)$.

Dziedzina: $[3; 4]$.

Błąd: 0.000016 .

Krok: 0.01 .

Kod programu

```

/**
 * @file main.c
 * @author Radoslaw Smoter (radoslaw.smoter@student.pk.edu.pl)
 * @version 0.1
 * @date 2021-12-16
 *
 * @copyright Copyright (c) 2021
 */

/*
 * Calculate given function.
 * Results save to a file.
 * Draw a plot for the results.
 */

#include <stdio.h>
#include <math.h>

int main(void)
{
    /* Function error */
    double error = .000016;
    /* Domain step */
    double step = .01;
    /* Domain */
    double domain[] = { 3., 4. };
    /* Sum iterator */
    int k = 1;
    /* Total sum */
    double sum = .0;

    for (double x = domain[0]; x < domain[1]; x += step) {
        /* Partial sum; for each domain step */
        double elem;
        /* Local copy of sum iterator */
        int l = k;
        /* Iterate, while element value is significant;
        When drops below error, stop the loop */
        do {
            double countFunc(double, int);
            elem = countFunc(x, l);
            sum += elem;
            l++;
        } while (elem > error);

        printf("THE_SUM = %.6f\n", sum);

        /* Save to a file */
        FILE *file = fopen("results.txt", "w");
        if (file != NULL) {
            fprintf(file, "THE_SUM = %.6f\n", sum);
            fclose(file);
        }
        /* File did not open correctly. */
        else {
            fprintf(stderr, "Error: File did not open correctly.\n");
            return -1;
        }
    }

    return 0;
}

/* Calculate a mathematical function */
double countFunc(double x, int k) {
    /* Partial values*/

```

```
double sinVal = sin(x / pow(2, k));  
double cosVal = cos(x / pow(2, k));  
double fractVal = 1 / pow(2, k);  
  
return fractVal * (sinVal + cosVal);  
}
```

Wyniki

THE_SUM = 106.783308

Opis Programu

Program składa się z funkcji `main()`, w której odbywa się pętla `for()` działająca na całej dziedzinie po kroku (`step`). W niej zadeklarowany jest `elem`, będący częściową sumą (sumą dla każdej iteracji). W pętli `do...while()` warunkiem zakończenia działania jest zmniejszenie dokładności obliczania `elem` przez funkcję `countFunc()`, aż `elem` stanie się mniejszy niż `error`. W tej pętli następuje również dodanie wartości `elem` do `sum`, które jest sumą wszystkich elementów.

Dalej, program wypisuje uzyskaną sumę na standardowe wyjście (`stdout`), a następnie zapisują ją jeszcze do pliku tekstowego („`results.txt`”), zgodnie z poleceniem. W razie niepowodzenia zapisu, tj. jeśli utworzenie pliku nie jest możliwe, zwraca błąd na `stderr` oraz kończy program z kodem `-1`.

Wnioski

Wynik sumy pokrywa się z oczekiwaniami, co łatwo zweryfikować programem napisanym w języku Matlab.

```
% sum.m
clc;clear;close all;

domain = [ 3.0, 4.0 ];
step = 0.01;
error = 0.000016;

sum = 0;

for x = domain(1):step:domain(2)
    elem = 1;
    k = 1;
    while elem > error
        elem = count_func(x, k);
        sum = sum + elem;
        k = k + 1;
    end
end

% count_func.m
function out = count_func(x, k)
    sin_val = sin(x / 2^k);
    cos_val = cos(x / 2^k);
    fract_val = 1 / (2^k);
    out = fract_val * (sin_val + cos_val);
end
```