

Міністерство освіти і науки України Національний технічний університет України

«Київський політехнічний інститут»

Лабораторна робота №1

з дисципліни «Алгоритми та методи обчислень»

Виконав студент групи: КВ-22

ПІБ: Крутогуз Максим Ігорович

Перевірив:

ОБЧИСЛЕННЯ ЗНАЧЕНЬ ФУНКЦІЇ

Загальне завдання:

3

1.2. Завдання для лабораторної роботи

Для заданого варіанта (табл. 1.1) виконати 3 завдання.

1. Побудувати таблицю залежності довжини ряду n, що забезпечує точність функції не меншу за задане значення eps у точці x = (b + a)/2, від eps:

eps	n	Абсолютна похибка	Залишковий член
10-2	4	0.005	0.001
	•••		

Значення eps змінюється від 10^{-2} до 10^{-14} з кроком 10^{-3} .

2. Для n (довжина ряду фіксована й дорівнює n), отриманого в п.1 при $eps = 10^{-8}$, у точках $x_i = a + h \cdot i$, h = (b - a)/10, i = 0, ..., 10 обчислити абсолютну похибку та залишковий член ряду. Результати подати у вигляді таблиці:

x_i	Абсолютна	Залишковий
	похибка	член
0	0.005	0.001

П р и м і т к а. Точним значенням функції вважати результат, що його дає бібліотечна функція Сі, при обчисленні якої отримують всі члени ряду відмінні від нуля.

Індивідуальне завдання:

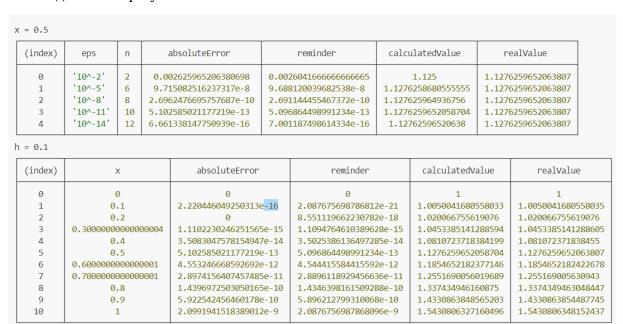
$$\frac{\text{Бирик-}}{\text{при в }} = \frac{\text{Функ-}}{f(x)}$$
 Інтервал $[a, b]$

12 chx $[0; 1]$

Код програми:

```
// my task is chx on the interval [0, 1]
interface Task1Data {
    eps: string,
    n: number,
    absoluteError: number,
    reminder: number,
    calculatedValue: number,
    realValue: number
}
interface Task2Data {
    x: number,
    absoluteError: number,
    reminder: number,
    calculatedValue: number,
    realValue: number
}
class Calculation {
    public static readonly a: number = 0
    public static readonly b: number = 1
    public static firstTableGeneration() : void {
        const data : Task1Data[] = []
         const x = (this.b + this.a) / 2
console.log("x = " + x)
         for (let epsDegree: number = 2; epsDegree <= 14; epsDegree += 3) {</pre>
             const eps: number = 10**-epsDegree
             let res: number = 1
             let uk: number = x^{**}2 / 2
             let k = 2
             while (eps < uk^*2/3 \mid \mid ((k = k - 2) \&\& false)) {
                  res += uk
                  uk *= x**2 / ((k + 1) * (k + 2))
                  k += 2
             const task1Data: Task1Data = {
                  eps: '10^-' + epsDegree,
                  n: k.
                  absoluteError: Math.abs(Math.cosh(x) - res),
                  reminder: uk,
                  calculatedValue: res,
                  realValue: Math.cosh(x)
             data.push(task1Data)
        console.table(data)
    public static secondTableGeneration(n: number) : void {
        const data: Task2Data[] = []
const h: number = (this.b - this.a) / 10
console.log('h = ' + h)
for (let i = 0; i <= 10; i++) {</pre>
             const x : number = this.a + h * i
             let res: number = 1
             let uk: number = x^**2 / 2
             for (let k = 2; k \le n; k += 2) {
                  res += uk
                  uk *= x**2 / ((k + 1) * (k + 2))
             const task2Data : Task2Data = {
                  absoluteError: Math.abs(Math.cosh(x) - res),
                  reminder: uk,
                  calculatedValue: res,
                  realValue: Math.cosh(x)
             data.push(task2Data)
         console.table(data)
    }
}
Calculation.firstTableGeneration()
```

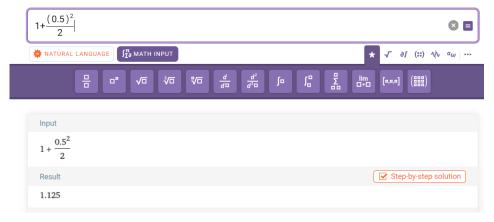
Налагодження та результати:



Малюнок 1: результат програми

FROM THE MAKERS OF **WOLFRAM LANGUAGE** AND **MATHEMATICA**





Малюнок 2: перевірка результату степеневого ряду для n=2 в точці x=0.5

Висновки:

Було перевірено на практиці теорію наближення чисел за допомогою ряду Тейлора. Для того, аби прорахувати функцію, ми використовуємо наближення функії в деякому недалекому околі точки, якщо ми не можемо зробити потрібно робити деякі перетворення.

Щодо наближення в першій таблиці ми бачимо, що похибка зменшується і з збільшенням члені ряду.

Щодо наближення в другій таблиці ми бачимо, що біля точки 0 менше похибка, навіть з такою самою кількістю членів. Це тому, що ми використовували ряд Маклорена, який має найбільшу точність приближення в точці 0.

Блоксхеми

