ДОМАШНЯЯ РАБОТА №1

«Численные методы решения нелинейных уравнений»

Выполнил:
студент 2 курса 13 группы кафедры ТП.

Петров Андрей Александрович

Вариант 8.

Задание: Дана функция $f(x) = cos(nx - (2n + 1)) + nx$, где $\boldsymbol{n}$– номер студента в списке подгруппы.

Необходимо выполнить следующее:

- Отделить все корни уравнения $f(x) = 0$.
- Сузить отрезки отделённости корней до размера 10^{-2} с помощью метода бисекций.
- Решить с точностью $\varepsilon = 10^{-5}$ указанное уравнение методом простых итераций.
- Решить с точностью $\varepsilon = 10^{-5}$ указанное уравнение методом Ньютона.

Сравнить полученные в последних двух пунктах результаты.

Ход работы:

$f(x) = cos(8x – 17) + 8x$

Отделим все корни уравнения $f(x) = 0$.

x	-∞	-2	-1	0	1	2	+∞
y	-	-	-	-	+	+	+

Проанализировав функцию, можно сделать вывод, что на промежутке [0; 1] возможно имеется хотя бы один корень.

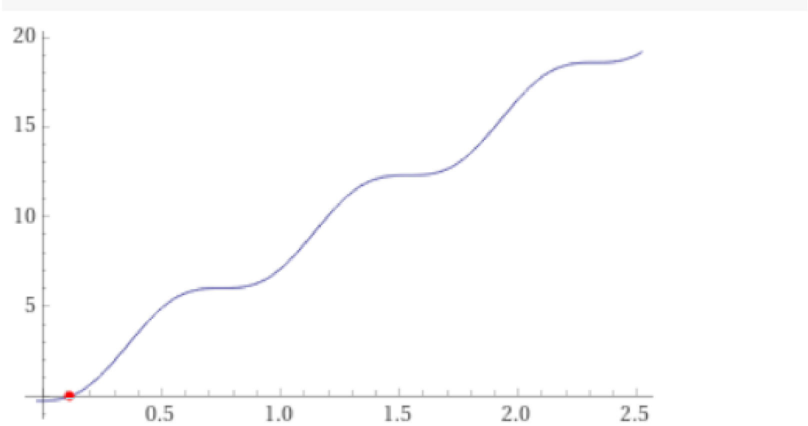
Рассмотрим [0; 1] и возьмем производную от нашей функции.

$f'(x) = -8 * \sin(8x - 17) + 8$

x	0	0.25	0.5	0.75	1
y'	+	+	+	+	+

Так как наша производная f'(x) на промежутке [0; 1] принимает положительные значения, то функция $f(x)$ на промежутке [0; 1] постоянно возрастает, что позволяем нам сделать вывод о наличии только одного корня.

График функции $f(x) = cos(8x – 17) + 8x$ (из Wolfram Alpha).



Сузить отрезки отделённости корней до размера 10^{-2} с помощью метода бисекций

Напишем программу, которая на будет сужать промежуток [0,1] до отрезка отдалённости корней размером 10^{-2}, на котором находится наш корень.

```
let a = 0.0;
let b = 1.0;

console.log(`Start segment: [a = ${a}, b = ${b}]`)

while (Math.abs(a - b) > 2e-2) {
    let x = (a + b) / 2;
    console.log(`\tSegment: [a = ${a}, b = ${b}]`);
    (f(x) * f(a)) < 0 ? b = x : a = x;
}

console.log(`Final segment: [a = ${a}, b = ${b}]`)
```

Результат работы программы:

```
Start segment: [a = 0, b = 1]
        Segment: [a = 0, b = 1]
        Segment: [a = 0, b = 0.5]
        Segment: [a = 0, b = 0.25]
        Segment: [a = 0, b = 0.125]
        Segment: [a = 0.0625, b = 0.125]
        Segment: [a = 0.09375, b = 0.125]
Final segment: [a = 0.109375, b = 0.125]
```

Решить с точностью $\varepsilon = 10^{-5}$ указанное уравнение методом простых итераций.

Выразим x из функции $f(x) = cos(8x - 17) + 8x$

$f(\text{x}) = 0 \Rightarrow \text{x} = \varphi(\text{x})$,

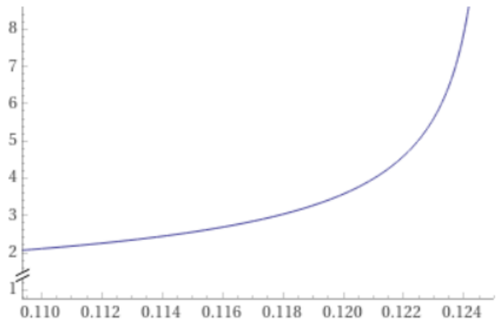
Выразим x двумя способами:

$\varphi_1(x) = -\cos(8x - 17) / 8$
$\varphi_2(x) = (\arccos(-8x) + 17) / 8$

Докажем, что производные этих функций на промежутке [0,1] меньше 1 (условие сходимости).

$\varphi'_1(x) = -1 \leq \sin(8x - 17) \leq 1$
$\varphi'_2(x) = 1 / \sqrt{1 - 64x^2}$

График $\varphi'_2(x) = 1 / \sqrt{1 - 64x^2}$ на промежутке [0.109375; 0.125]



Исходя из графика, полученного в Wolfram Alpha, можно предположить, что при $\varphi_2(x)$ метод простых итераций не сойдется.

Напишем программу, которая реализует метод простых итераций.

```
let phi = (x) => -Math.cos(8.0 * x - 17.0) / 8.0;
let xCurrent = (0.109375 + 0.125) / 2;
let xPrevious = 0.109375;
let iteration = 0;

while (Math.abs(xCurrent - xPrevious) > 1e-5) {
    xPrevious = xCurrent;
    xCurrent = phi(xCurrent);
    iteration++;
}
console.log(`Count iteration: ${iteration}; x = ${xCurrent}`)
```

Результат выполнения программы:

```
Count iteration: 3; x = 0.1172438862436553
```

Решить с точностью $\varepsilon = 10^{-5}$ указанное уравнение методом Ньютона.

Напишем программу, которая реализует решение уравнения методом Ньютона:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Код программы:

```
let f = (x) => Math.cos(8 * x - 17) + 8 * x;
let fDerivative  = (x) => -8 * Math.sin(8 * x - 17) + 8;

let xCurrent = (0.109375 + 0.125) / 2;
let xPrevious = 0.109375;
let iteration = 0;
```

```
while (Math.abs(xCurrent - xPrevious) > 1e-5) {
    iteration++;
    xPrevious = xCurrent;
    xCurrent -= f(xPrevious) / fDerivative(xPrevious);
}
console.log(`Count iteration: ${iteration}; x = ${xCurrent}`)
```

Результат выполнения программы:

```
Count iteration: 2; x = 0.11724634095943533
```

Сравнить полученные в последних двух пунктах результаты.

Решение Wolfram Alpha:	x ≈ 0.11724634095943301740
Метод простых итераций:	x ≈ 0.1172438862436553
Метод Ньютона:	x ≈ 0.11724634095943533

Исходя из полученных результатов работы методов Ньютона и простых итераций, реализованных мной, а также результата, полученного с помощью Wolfram Alpha, можно сделать вывод, что метод Ньютона точнее метода простых итераций и выполняет программу за меньшее количество итераций.