Отчет о методе Стефенсона на NodeJS

В алгоритме решения нелинейных уравнений методом Стефенсона использую следующие функции:

1. ReadFile - функция для чтения ‘функции’ из файла

```
async function readFile(path){
    const expression = await fs.readFile(path,{encode:'utf-8'},(err)=>{
        if(err) return 'File not be reading';
    });

    return(expression.toString());
}
```

2. WriteFile – функция для записи ответа в файл

```
async function writeFile(path, data){
    try {
        await fs.writeFile(path, data);
        console.log(`Ответ записан в файл ${path}`);
    } catch (err) {
        console.log('Ошибка при записи в файл');
    }
}
```

3. Func - функция для расчета значения ‘функции’ в точке x_i

```
function Func(expression, xValue){
    const modifiedExpression = expression.replace(/x/g, xValue);
    const result = eval(modifiedExpression);
    return(result);
}
```

4. SteffensonMetod - метод для решения нелинейных уранений

```
function SteffencensMethod(expression,x,accuracy){
    let nextX = x - ( Func(expression,x)**2 ) / ( Func(expression, x +
Func(expression,x))- Func(expression,x) );
    if(Func(expression,nextX) < accuracy){
        return(nextX);
    }else{
        return SteffencensMethod(expression,nextX,accuracy);
    }
}
```

5. CalcFirstX - функция для нахождения x_0 для уравнения

```
function calcFirstX(expression){
    for(let i = 0; i < 10; i++){
        if(Func(expression, i) < 0 && Func(expression, i+1) > 0){
            return ( (i+i+1)/2 ) + 0.1;
        }
    }
}
```

6. Main - главная функция запускающая остальные

```
async function Main(){
    const expression = await readFile('./input.txt');
    const firstX = calcFirstX(expression);
    const answer = SteffencensMethod(expression, firstX, 0.0001);
    await writeFile('output.txt', answer.toString());
}
```

Алгоритм :

1. Находим x_0 как среднее значение на [a,b] где a,b ближайшие целые точки где меняется знак функции
2. Находим x_i по формуле

$$x_1 = x_0 - \frac{(f(x_0))^2}{f(x_0 + f(x_0)) - f(x_0)}$$

3. После подставляем найденный x_i в ‘функцию’ и проверяем что бы разница между новым значением и старым при x_i и x_i-1 соответственно отличались на нужную нам погрешность.
4. Если разница больше то продолжаем процесс поиска x_n
5. Если разница меньше то получаем ответ