МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Факультет автоматики и вычислительной техники

Кафедра автоматики



**ОТЧЁТ**

**по лабораторной работе №1**

По дисциплине: «Вычислительная математика»

Тема: «Численные методы решения нелинейных уравнений»

Вариант 15

|  |  |
| --- | --- |
| Факультет: АВТФ  Группа: АВТ-019  Студент: Нерлих М. А. | Преподаватель: Балакин В.В. |

Новосибирск

2021г.**Содержание**

[Введение 3](#_heading=h.gjdgxs)

[1. Описание задания](#_heading=h.30j0zll)

[1.1. Исходные данные, цели и задачи работы соответствующие варианту](#_heading=h.1fob9te)

[2. Описание численных методов](#_heading=h.30j0zll) 4

[2.1. Метод](#_heading=h.1fob9te) итераций

2.2 Метод касательных(метод Ньютона)

2.3 Метод хорд и касательных

[3. Листинг программ](#_heading=h.30j0zll) 6

[4. Результаты вычислений по методам, различия количества итераций при разных точностях](#_heading=h.30j0zll) 9

Метод хорд и касательных

[5. Сравнения методов](#_heading=h.30j0zll) 10

[6. Вывод](#_heading=h.30j0zll)

**Введение**

Лабораторная работа №1 посвящена численным методам решения нелинейных уравнений. Необходимость отыскания корней нелинейных уравнений встречается в расчетах систем автоматического управления и регулирования, собственных колебаний машин и конструкций, в задачах кинематического анализа и синтеза, плоских и пространственных механизмов и других задачах.

**1. Описание задания**

**1.1. Цель лабораторной работы**

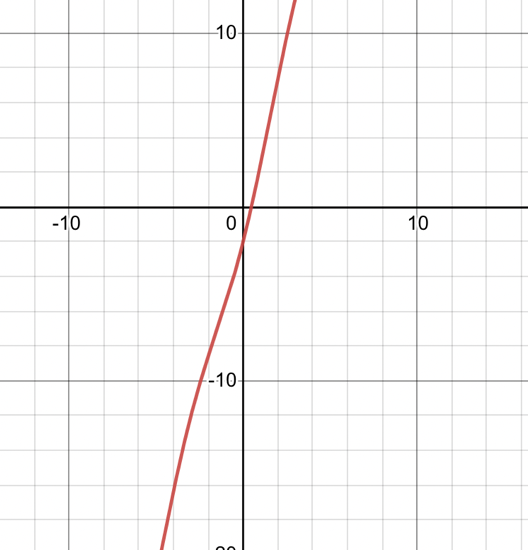
**Цель**: решить уравнение: с точностью ξ=10^-5 следующими методами:

1) Методом итераций

2) Методом касательных (Ньютона)

3) Методом хорд и касательных

Для каждого метода исследовать обусловленность задачи (численного метода) и влияние заданной точности на число потребовавшихся итераций.  Сравнить методы по скорости сходимости и выбрать наиболее быстро сходящийся вычислительный процесс.  Проанализировать результаты работы и сделать выводы.



*Рис 1. Обзорный график исходной функции*

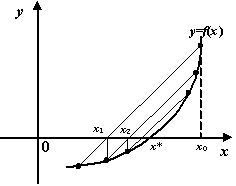
**2. Описание численных методов**

**2.1. Метод итераций**

Метод реализует стратегию постепенного уточнения значения корня.

Суть: нахождение по приближённому значению величины следующего приближения, которое является более точным. Метод позволяет получить значения корней системы с заданной точностью в виде предела последовательности некоторых векторов (итерационный процесс). Характер сходимости и сам факт сходимости метода зависит от выбора начального приближения корня x0.

Основной проблемой применения метода является обеспечение сходимости итерационного процесса: нужно найти такое эквивалентное преобразование, чтобы обеспечивалось условие сходимости.



*Рис 3. Метод итерации*

**2.2. Метод касательных (метод Ньютона)**

Метод касательных (метод Ньютона) предназначен для приближенного нахождения нулей функции. По сравнению с методом половинного деления этот метод усложняется необходимостью вычисления производных на каждом шаге, если — некоторое приближение к корню уравнения, то следующее приближение определяется как корень касательной к функции, проведенной в точке.

Тогда алгоритм последовательных вычислений в методе Ньютона состоит в следующем:

Без всяких изменений метод обобщается на комплексный случай. Если корень является корнем второй кратности и выше, то порядок сходимости падает и становится линейным.

К недостаткам метода Ньютона следует отнести его локальность, поскольку он гарантированно сходится при произвольном стартовом приближении только, если везде выполнено условие, в противной ситуации сходимость есть лишь в некоторой окрестности корня.

Метод Ньютона (метод касательных) обычно применяется в том случае, если уравнение имеет корень, и выполняются следующие условия:

‒ функция определена и непрерывна;

‒ функция принимает значения разных знаков на концах отрезка;

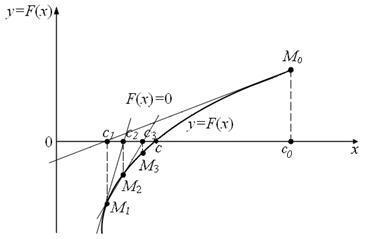
‒ производные сохраняют знак на отрезке (то есть функция либо возрастает, либо убывает на отрезке, сохраняя при этом направление выпуклости);

Приближенное решение *ξ* и погрешность приближения находятся по следующей схеме:

*, n=0,1,2,…;* **(3)**

если **(4)**

если **(5)**



*Рис 3. Метод касательных (метод Ньютона)*

Смысл метода заключается в следующем: на отрезке выбирается такое число, при котором имеет тот же знак, что и производная, то есть выполняется условие. Таким образом, выбирается точка с абсциссой, в которой касательная к кривой на отрезке пересекает ось. За точку сначала удобно выбирать один из концов отрезка.

***2.3.* Метод хорд и касательных**

Метод даёт приближения к корню с разных сторон.

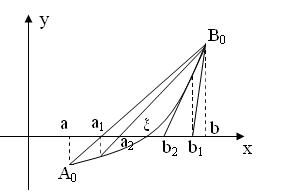
Идея метода хорд состоит в том чтобы заменить функцию на отрезке хордой, а идея метода касательных или метода Ньютона является замена дуги кривой функции ее касательной. Стоит отметить, что начальное приближение метода хорд определяется тот конец промежутка для которого производная в данной точке умноженная на двойную производную этой же точки меньше нуля, а для метода касательных больше нуля. Процесс сужения так же производится до указанной точности. К плюсам, как уже отмечалось относится быстрота нахождения и меньшая затратность на приведенные итерации. Приближенное решение *ξ* и погрешность приближения находятся по следующей схеме:

если

*,, n=0,1,2,…;* **(6)**

если

*,, n=0,1,2,… .* **(7)**



*Рис 4. Метод хорд и касательных*

**3. Ход работы**

Для работы программы пере используются несколько модулей.

*/\* math.js \*/*

export let eps = 10 \*\* -5;

export const setEps = (newEps) => eps = newEps;

export const derivative = (f) => (x) => (f(x + eps) - f(x)) / eps;

export const secondDerivative = (f) => derivative(derivative(f));

*/\* solve.js \*/*

export const solve = (method) => (f, a, b) => {

const methodName = method.name;

const begin = performance.now();

const { result, iterations } = method(f, a, b);

const end = performance.now();

const time = end - begin;

console.log(`method: ${methodName}`);

console.log(`result: ${result}`);

console.log(`iterations: ${iterations}`);

console.log(`execution time: ${time}`);

console.log('\n');

};

*/\* index.js \*/*

*/\**

*\* Лабораторная работа 1*

*\* Численные методы решения нелинейных уравнений*

*\* Вариант 15*

*\* Функция 4x-cos(x)-1=0*

*\* Методы 4, 5, x*

*\* Метод итераций, метод хорд и касательных, метод Ньютона*

*\*/*

import { solve } from './solve.js';

import { newton, iterative, chordsAndTangents } from './methods.js';

import { setEps, eps } from './math.js';

const f = (x) => 4 \* x - Math.cos(x) - 1;

setEps(10 \*\* -5);

const a = -5;

const b = 10;

console.log(`a = ${a}; b = ${b}; eps=${eps};`);

console.log('\n');

solve(newton)(f, a, b);

solve(iterative)(f, a, b);

solve(chordsAndTangents)(f, a, b);

**3.1. Метод итераций**

Листинг программы на языке JavaScript:

*/\* index.js \*/*

*/\**

*\* Лабораторная работа 1*

*/\* methods.js \*/*

import { derivative, secondDerivative, eps } from './math.js';

export const iterative = (f, a, b) => {

const iter = (f) => (x, L) => x + L \* f(x);

const L = -1/4;

let x = 0;

let iterations = 0;

let checker = 0;

x = (b + a) / 2;

while (Math.abs(checker - x) > eps) {

x = iter(f)(x, L);

checker = iter(f)(x, L);

iterations++;

}

return { result: x, iterations };

};

**3.2. Метод касательных (метод Ньютона)**

*/\* methods.js \*/*

import { derivative, secondDerivative, eps } from './math.js';

export const newton = (f, a, b) => {

let x = 0;

let iterations = 0;

let checker = 0;

if (f(a) \* f(b) > 0) {

return { result: 'n/a', iterations };

}

if (f(a) \* secondDerivative(f)(a) > 0) {

x = a;

} else {

x = b;

}

while (Math.abs(checker - x) > eps) {

x -= f(x) / derivative(f)(x);

checker = x - f(x) / derivative(f)(x);

iterations++;

}

return { result: x, iterations };

};

**3.3. Метод Хорд и касательных**

/\* methods.js \*/

import { derivative, secondDerivative, eps } from './math.js';

export const chordsAndTangents = (f, a, b) => {

let iterations = 0;

if (f(a) \* f(b) >= 0)

return { result: 'n/a', iterations };

while (true) {

if (f(a) \* secondDerivative(f)(a) < 0)

a = a - f(a) \* (a - b) / (f(a) - f(b));

else if (f(a) \* secondDerivative(f)(a) > 0)

a = a - f(a) / derivative(f)(a);

if (f(b) \* secondDerivative(f)(b) < 0)

b = b - f(b) \* (b - a) / (f(b) - f(a))

else if (f(b) \* secondDerivative(f)(b) > 0)

b = b - f(b) / derivative(f)(b);

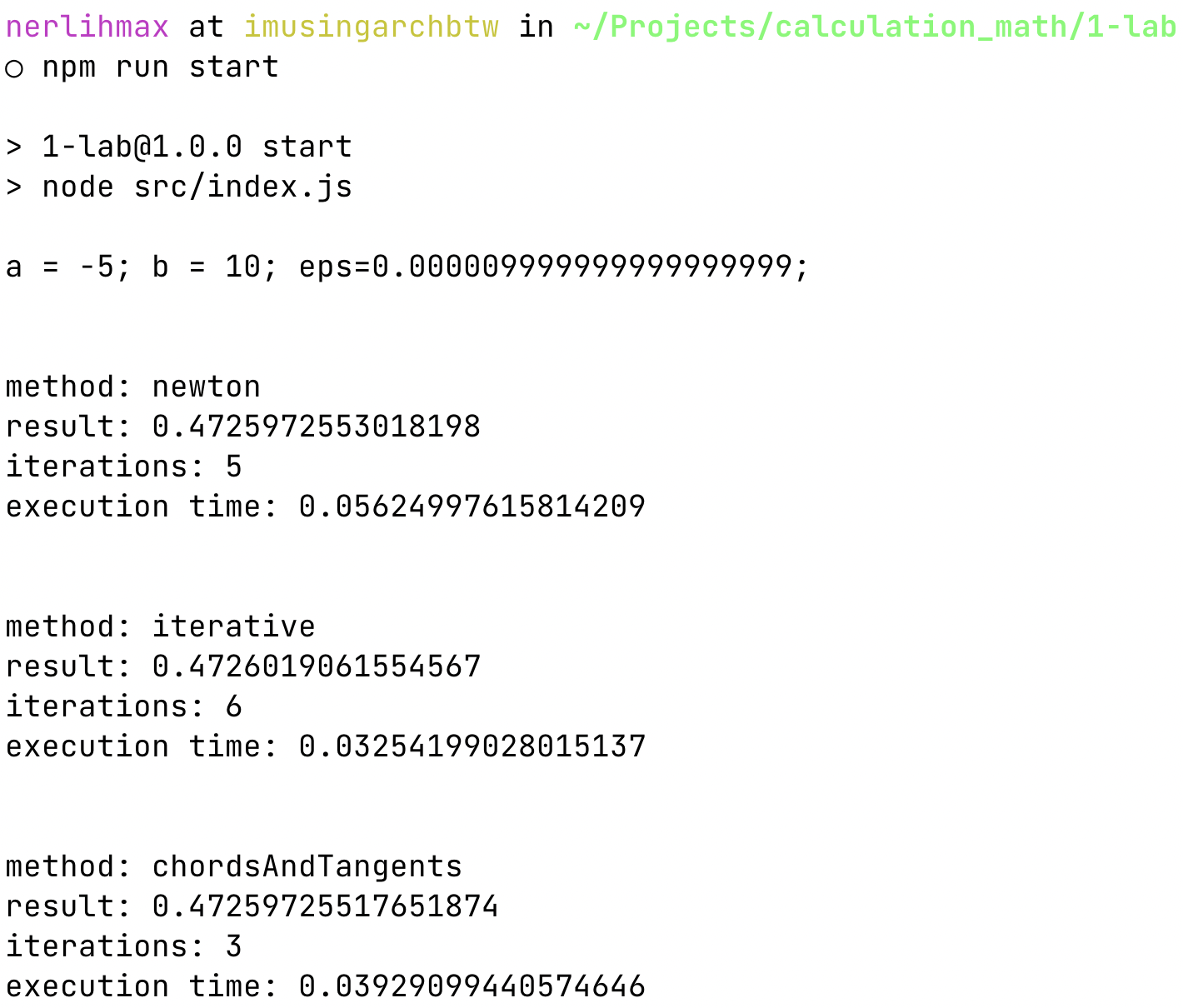
iterations++;

if (Math.abs(a - b) <= 2 \* eps)

return { result: (a + b) / 2, iterations };

}

};

****

**4. Результаты вычислений по методам, различия**

**количества итераций при разных точностях**

|  |  |  |
| --- | --- | --- |
| **Метод** | **Количество итераций** | **Время работы** |
| Метод итераций | 6 | 0.032 мс |
| Метод касательных (метод Ньютона) | 5 | 0.056 мс |
| Метод хорд и касательных | 3 | 0.039 мс |

**4.1. Решение заданного нелинейного уравнения** **методом итераций**

*Таблица 1 – Результаты вычислений корня методом итераций*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№** | **Точность (ε)** | **a** | **b** | **x** | **Кол-во итераций** | **Время работы программы** |
| 1 |  | -5 | 10 | 0.49969 | 2 | 0.030 мс |
| 2 |  | 0.46943 | 3 | 0.030 мс |
| 3 |  | 0.47295 | 4 | 0.031 мс |
| 4 |  | 0.47255 | 5 | 0.033 мс |
| 5 |  | 0.47260 | 6 | 0.032 мс |

Вывод: для метода **итераций** с увеличением точности, количество итераций и времени также увеличивается.

**4.2. Решение заданного нелинейного уравнения** **методом касательных (методом Ньютона)**

*Таблица 2 – Результаты вычислений корня методом касательных*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№** | **Точность (ε)** | **a** | **b** | **x** | **Кол-во итераций** | **Время работы программы** |
| 1 |  | -5 | 10 | 0.48651 | 2 | 0.052 мс |
| 2 |  | 0.47262 | 3 | 0.055 мс |
| 3 |  | 0.47261 | 4 | 0.055 мс |
| 4 |  | 0.47261 | 4 | 0.055 мс |
| 5 |  | 0.47259 | 5 | 0.056 мс |

Вывод: для метода касательных (метода Ньютона) с увеличением точности, количество итераций и времени также увеличивается

**4.3. Решение заданного нелинейного уравнения** **методом хорд и касательных**

*Таблица 3 – Результаты вычислений корня методом хорд и касательных*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№** | **Точность (ε)** | **a** | **b** | **x** | **Кол-во итераций** | **Время работы программы** |
| 1 |  | -5 | 10 | 0.47252 | 2 | 0.036 мс |
| 2 |  | 0.47248 | 2 | 0.035 мс |
| 3 |  | 0.47248 | 2 | 0.037 мс |
| 4 |  | 0.47259 | 3 | 0.039 мс |
| 5 |  | 0.47259 | 3 | 0.039 мс |

Вывод: для метода хорд и касательных с увеличением точности, количество итераций и времени также увеличивается.

**5. Сравнение количества итераций и скорости при увеличении точности вычислений**

Наиболее трудоёмким получился метод итераций, имея рекордные 6 итераций, а наименее – метод хорд и касательных. Однако самым быстрым оказался метод итераций, а самым долгим метод Ньютона.

*Таблица 4 – Сравнение результатов трех методов при разной точности*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Точность (ε)** | **x** | **Трудоёмкость** | | |
| **Метод итераций** | **Метод касательных** | **Метод Хорд и касательных** |
| 1 |  | 0.4 | 2 | 2 | 2 |
| 2 |  | 0.47 | 3 | 3 | 2 |
| 3 |  | 0.472 | 4 | 4 | 2 |
| 4 |  | 0.4725 | 5 | 4 | 3 |
| 5 |  | 0.47259 | 6 | 5 | 3 |

*Таблица 5 – Сравнение результатов трех методов при разной скорости выполнения*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Точность (ε)** | **x** | **Время** | | |
| **Метод итераций** | **Метод касательных** | **Метод Хорд и касательных** |
| 1 |  | 0.4 | 0.030 мс | 0.052 мс | 0.036 мс |
| 2 |  | 0.47 | 0.030 мс | 0.055 мс | 0.035 мс |
| 3 |  | 0.472 | 0.031 мс | 0.055 мс | 0.037 мс |
| 4 |  | 0.4725 | 0.033 мс | 0.055 мс | 0.039 мс |
| 5 |  | 0.47259 | 0.032 мс | 0.056 мс | 0.039 мс |

**6. Вывод**

В ходе лабораторной работы №1 были выведены 3 программы с различными методами решения нелинейных уравнений. Были применены 3 различных метода, данные в методических указаниях. метод итераций является самым быстрым по скорости выполнения, а метод Ньютона – самым медленным.