Министерство образования и науки Российской Федерации

Санкт-Петербургский политехнический университет Петра Великого

Институт прикладной математики и информатики

Лабораторная работа по дисциплине

Численные методы

«Решение нелинейных уравнений»

Выполнил

студент гр.5030102/20001 Дрекалов Н.С.

Преподаватель Добрецова С.Б.

Санкт-Петербург

2023

Оглавление

[**Формулировка задачи** 3](#_Toc147173981)

[**Формализация** 3](#_Toc147173982)

[**Теорема о верхней границе** 3](#_Toc147173983)

[**Метод половинного деления** 3](#_Toc147173984)

[**Метод простых итераций** 4](#_Toc147173985)

[**Предварительный анализ задачи** 5](#_Toc147173986)

[**Тестовый пример к методам** 7](#_Toc147173987)

[**Контрольные тесты** 8](#_Toc147173988)

[**Модульная структура программы** 8](#_Toc147173989)

[**Численный анализ методов** 10](#_Toc147173990)

[**Вывод** 12](#_Toc147173991)

## **Формулировка задачи**

Найти решение нелинейного уравнения с помощью метода половинного деления и метода простых итераций.

*Функции:*

## **Формализация**

* Пусть f (x) : R− > R - алгебраическая и трансцендентная функция. Требуется: найти x∗ такое, что f (x∗) = 0. Корень x∗ ищется с точностью до заданного ε, что |x − x∗| < ε,

где x - точный корень.

## **Теорема о верхней границе**

*Условия применимости:*

* Функция f(x) – многочлен.

*Алгоритм метода:*

* Посчитать границу по формуле , где m ­ номер первого отрицательного коэффициента в ряду и a′ ­ наибольший по модулю отрицательный коэффициент. Проделать то же самое для -x, 1/x, -1/x. Таким образом пара значений (x, 1/x) дает диапазон положительных значений корня, а пара (-x, -1/x) – отрицательных.

## **Метод половинного деления**

*Условия применимости:*

* Если функция f(x) непрерывна на отрезке [a, b] и на его концах принимает разные знаки, то методом половинного деления можно найти корень уравнения.

*Алгоритм метода:*

Допустим, что единственный корень уравнения лежит на отрезке [a, b].

1. Расчет середины отрезка [а, b].

x= (a+ b) / 2

1. k-итерация (k=1,2,3…). Сужение отрезка [а, b].

Если f(а) \* f (x) < 0, то b**=** x

Если f(а) \* f(x) > 0, то а= x

1. Расчет середины отрезка [а, b].

x= (a+ b) / 2

1. Проверка условия итерационного процесса. Если условие выполнено, то сk – корень уравнения:

| b – a | < ℇ

Иначе перейти к 2 пункту.

**Метод простых итераций**

*Условия применимости:*

* ∃q : |ϕ ′ (x)|≤ q < 1 для ∀x ∈ [a, b]
* Корень x∗ ∈ [α, β] ⊂ [a, b], где ,

*Алгоритм метода:*

1. Находим M и m – границы производной f’ на отрезке [a; b].
2. Находим и , x1 = a, x2 = b.
3. Проверка условия остановки итерационного процесса, если условие выполнено, то – корень уравнения:

Иначе перейти к 4 пункту.

1. Получаем новую пару так, что , а .
2. Переходим к пункту 3.

## **Предварительный анализ задачи**

*Полученные отрезки:*

* “x”: ; m = 4, a’ = -1, a0 = 2;
* “-x”: ; m = 2, a’ = -8, a0 = 2;
* “1/x”: ; m = 2, a’ = -8, a0 = 1;
* “-1/x”: ; m = 2, a’ = -8, a0 = 1;

Таким образом, положительные корни находятся в интервале (0.26; 1.85), а отрицательные – (-3, -0.26)

*Выделение корня*

*Найдем производную полинома:*

Производная положительна для любого x > 0, в частности, для x из интервала (0.26; 1.85), то есть на данном интервале один корень.

*Найдем производную трансцендентной функции:*

Функция всюду непрерывна и при этом f(-6) > 0, f(-4) < 0, и на этом участке производная знакопостоянна. То есть, на интервале (-6; -4) есть корень и он единственный.

*Проверка условий для методов (полином)*

*Метод половинного деления:*

* Функция непрерывна на этом отрезке
* f (0.26) ∗ f (1.85) < 0

*Метод простых итераций:*

* Функция непрерывна и имеет производную в каждой точке
* f (0.26) ∗ f (1.85) < 0

*Проверка условий для методов (трансцендентное уравнение)*

*Метод половинного деления:*

* Функция непрерывна на этом отрезке
* f (-6) ∗ f (-4) < 0

*Метод простых итераций:*

* Функция непрерывна и имеет производную в каждой точке
* f (-6) ∗ f (-4) < 0

**Тестовый пример к методам**

*Посчитаем каждый шаг МПД для полинома для e = 0.001 на отрезке (0.26, 0.5)*

a = 0.260, b = 0.50000, x = (a + b) / 2 = 0.380, (x – x\*) = 0.074 a = 0.260, b = 0.380, x = (a + b) / 2 = 0.320, (x – x\*) = 0.014 a = 0.260, b = 0.320, x = (a + b) / 2 = 0.290, (x – x\*) = 0.016 a = 0.290, b = 0.320, x = (a + b) / 2 = 0.305, (x – x\*) = 0.001 a = 0.305, b = 0.320, x = (a + b) / 2 = 0.313, (x – x\*) = 0.006 a = 0.305, b = 0.313, x = (a + b) / 2 = 0.309, (x – x\*) = 0.002 a = 0.305, b = 0.309, x = (a + b) / 2 = 0.307, (x – x\*) < 0.001 = e

Найден корень x\* = 0.307. Получаем, что точность в среднем увеличивается с каждой итерацией.

*Посчитаем каждый шаг МПИ для полинома для e = 0.001 на отрезке (0.26, 0.5)*

x = 0.260, x’ = 0.297, x0 = (x + x’) / 2 = 0.278, (x - x0) = 0.028

x = 0.297, x’ = 0.290, x0 = (x + x’) / 2 = 0.293, (x - x0) = 0.013

x = 0.290, x’ = 0.304, x0 = (x + x’) / 2 = 0.297, (x - x0) = 0.010

x = 0.304, x’ = 0.301, x0 = (x + x’) / 2 = 0.302, (x - x0) = 0.004

x = 0.301, x’ = 0.306, x0 = (x + x’) / 2 = 0.303, (x - x0) = 0.003

x = 0.306, x’ = 0.305, x0 = (x + x’) / 2 = 0.305, (x - x0) < 0.001 = e

Найден корень x\* = 0.307. Получаем, что точность увеличивается с каждой итерацией.

**Контрольные тесты**

*Для каждого из уравнений построим графики зависимостей*

* Зависимость числа итераций от заданной точности ε = [10−2, 10−15]
* Зависимости фактической ошибки (разности точного и найденного значений корня) от заданной точности ε = [10−2, 10−15]. На график нанести линию заданной точности.

**Модульная структура программы**

/// Класс, который имеет два поля - для хранения числа итераций и для хранения корня

class Result {

public:

explicit Result(unsigned iterNumber, double root) : iterNumber(iterNumber), root(root) {};

const unsigned iterNumber = 0;

const double root;

};

#define MAX\_ITER\_NUM ULLONG\_MAX

/\*\*

\* **\brief** Функция находит корень в заданном промежутке методом ПД с точностью эпсилон

\* **@param** f функция, с у которой ищется корень

\* **@param** a левая граница

\* **@param** b правая граница

\* **@param** epsilon точность

\* **@return** тип Result в котором содержатся два поля - корень и число итераций

\*/

Result findRoot(double(&f)(double), double a, double b, double epsilon);

/\*\*

\* **\brief** Функция находит корень в заданном промежутке методом простых итераций с точностью эпсилон для полинома

\* **@param** f полином

\* **@param** epsilon точность

\* **@return** тип Result в котором содержатся два поля - корень и число итераций

\*/

Result findRootForPolynom(double(&f)(double), double epsilon);

/\*\*

\* **\brief** Функция находит корень в заданном промежутке методом простых итераций с точностью эпсилон для трансцендентой функции

\* **@param** f трансцендентная функция

\* **@param** epsilon точность

\* **@return** тип Result в котором содержатся два поля - корень и число итераций

\*/

Result findRootForComplex(double(&f)(double), double epsilon);

/\*\*

\* **\brief** функция, считающая значения полинома в точке x

\* **\param** x переменная

\* **\return** значение полинома в точке x

\*/

double polyVar11(const double x);

/\*\*

\* **\brief** функция, считающая значения транс фукнции в точке x

\* **\param** x переменная

\* **\return** значение функции в точке x

\*/

double complexVar11(const double x);

/\*\*

\* **\brief** функция, считающая корень для полинома для различной точности и выводящая всё на экран

\*/

void testFirstFunc();

/\*\*

\* **\brief** функция, считающая корень трансцедентной фукнции для различной точности и выводящая всё на экран

\*/

void testSecondFunc();

## **Численный анализ методов**

*Рассмотрим зависимости количества итераций от точности приближения.*



На графике видно, что быстрее всего работает МПИ для трансцендентного уравнения (это связано с удачно выбранным интервалом корня), остальные три выполняются примерно одинаково по скорости, с логарифмической сложностью.

*Рассмотрим зависимость ln разности с истинным корнем от точности*



Из графика видно, что во всех случаях достигается нужная точность (значение погрешности не превышает значение заданной точности). То есть результаты получены верно.

## **Вывод**

В лабораторной работе я нашел корни уравнений методом половинного деления и методом простых итераций.

В ходе исследования сравнены эффективность методов. Метод половинного деления стабильно сходится к корню, метод простых итераций же сходится к корню с адекватной скоростью только при ряде условий.

Можно сделать вывод, что для наиболее эффективного решения уравнения, нужно его исследовать (узнать поведение функции и её производных на промежутке) и на основе полученной предварительной точности выбрать один из методов.