МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АДЫГЕЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Инженерно-физический факультет
Кафедра автоматизированных систем обработки информации и управления

Отчет по практике

Реализация метода касательных (Ньютона) для нахождения корня функции.

1 курс, группа 1УТС

Выполнил:
________________ Д. А. Коржаков
«___» __________ 2021 г.

Руководитель:
________________ С. В. Теплоухов
«___» __________ 2021 г.

Майкоп, 2021 г.

1. План работы

1) Текстовая формулировка задачи

2) Краткая теория, необходимая для решения задачи

3) Графическая иллюстрация метода Ньютона для локализации корня

4) Программа поиска корня

5) Примеры работы программы

2. Ход работы

2.1. Текстовая формулировка задачи

Требуется реализовать метод касательных (Ньютона) для нахождения корня кубической функции $f(x) = 2x^3+3x^2+x-10$ путем написания консольной программы на C++. Должен осуществляться контроль вводимых данных - границ отрезка поиска корня [a;b] и погрешности измерения ε.

2.2. Краткая теория, необходимая для решения задачи

Метод Ньютона, алгоритм Ньютона (также известный как метод касательных) — это итерационный численный метод нахождения корня (нуля) заданной функции, наиболее эффективный для решения уравнения $f(x) = 0$. Поиск решения осуществляется путём построения последовательных приближений и основан на принципах простой итерации. Метод обладает квадратичной сходимостью.

Алгоритм поиска корня согласно методу Ньютона выглядит следующим образом:

- Задается начальное приближение x_0 к корню.
- Новое приближение вычисляется с помощью расчетной формулы метода касательных:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \tag{1}$$

- Новое приближение вычисляется пока не выполнено условие остановки:

$$|f'(x_{i+1})| < \varepsilon \tag{2}$$

Условие остановки (2) будет использовано как завершение алгоритма программы.

2.3. Графическая иллюстрация метода Ньютона для локализации корня

Для нахождения корня кубического уравнения $f(x) = 0$ необходимо локализовать корень для задания пользователем отрезка поиска корня. Для этого пользователь может воспользоваться теоремой Штурма или графическим методом. График функции $f(x) = 2x^3 + 3x^2 + x - 10$ представлен на рис. 1:

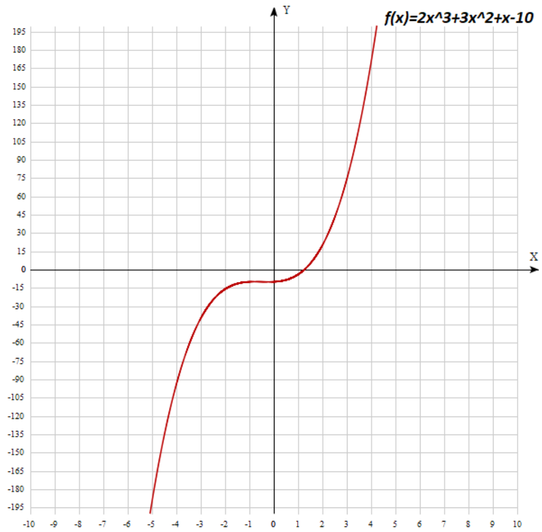


Рис. 1. График функции

Ниже на рис. 2 представлена геометрическая интерпретация метода касательных на локализованном отрезке графика функции $f(x) = 2x^3 + 3x^2 + x - 10$:

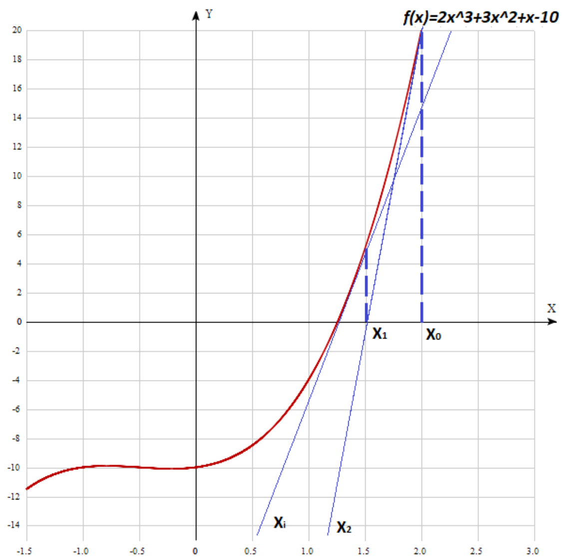


Рис. 2. Графическая иллюстрация

2.4. Программа поиска корня

На основании положений вышеприведенной теории, написана программа, осуществляющая поиск корня функции $f(x) = 2x^3 + 3x^2 + x - 10$ при задании пользователем отрезка поиска. Код программы:

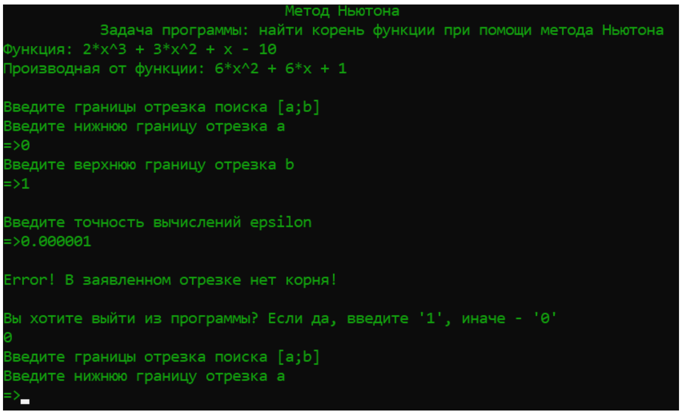
2.5. Код приложения

```
#include <cmath>
#include <iostream>

using namespace std;

float f(float x)
{
    return 2 * x * x * x + 3 * x * x + x - 10;
}

float df(float x)
{
    return 6 * x * x + 6 * x + 1;
}

int main()
{
    setlocale(LC_ALL, "Russian");
    int exit = 0, i = 0;
    float a, b, epsilon, xi = 0;

    printf("Метод Ньютона\n");
    printf("Задача программы: найти корень функции при помощи метода Ньютона\n");
    printf("Функция: 2*x^3 + 3*x^2 + x - 10\n");
    printf("Производная от функции: 6*x^2 + 6*x + 1\n\n");

    do
    {
        cout << "Введите границы отрезка поиска [a;b]\n";
        cout << "Введите нижнюю границу отрезка a\n=>";
        cin >> a;
        cout << "Введите верхнюю границу отрезка b\n=>";
        cin >> b;
        cout << "\nВведите точность вычислений epsilon\n=>";
        cin >> epsilon;

        if (a > b)
        {
            xi = a;
            a = b;
            b = xi;
        }

        if (f(a) * f(b) > 0)
```

```
        {
            cout << "\nError! В заявленном отрезке нет корня!\n";
        }
        else
        {
            int iterator = 0;
            xi = a;
            float eps = epsilon;
            float last_xi = 0;

            printf("+----+------------+------------+------------+------------+-------"
                "-----+\n");
            printf("|  i |     x_i     |    f(xi)    |    f’(xi)    |   x_i + 1   |   "
                "epsilon  |\n");
            printf("+----+------------+------------+------------+------------+-------"
                "-----+\n");

            while (1)
            {
                float fxi = f(xi);
                float _fxi = df(xi);
                float xi_1 = xi - (fxi / _fxi);
                eps = abs(xi - last_xi);
                printf("|%3d |%12.8f |%12.8f |%12.8f |%12.8f |%12.8f|\n", iterator, xi, fxi,
                    _fxi, xi_1, eps);
                iterator++;
                last_xi = xi;
                xi = xi_1;
                eps = abs(xi - last_xi);
                if (eps <= epsilon)
                {
                    printf("+----+------------+------------+------------+------------+---"
                        "---------+\n");
                    break;
                }
            }
            cout << "Приблизительное значение корня функции x при погрешности epsilon = ";
            printf("%1.8f", epsilon);
            cout << ":\n x = ";
            printf("%1.8f", xi);
        }
        cout << "\nВы хотите выйти из программы? Если да, введите ’1’, иначе - ’0’\n";
        cin >> exit;

    }
    while (exit != 1);
    return 0;
}
```

2.6. Примеры работы программы

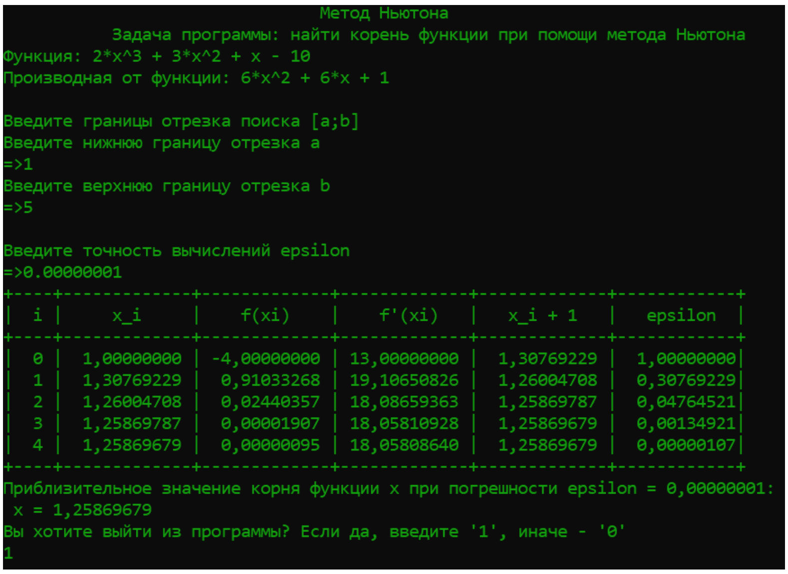
Расмотрены основные варианты работы программы. Если пользователем введен неверный отрезок поиска, то программа выдаст ошибку и предложит пользователю продолжить работу и ввести отрезок заново (рис. 3):



```
                                Метод Ньютона
           Задача программы: найти корень функции при помощи метода Ньютона
Функция: 2*x^3 + 3*x^2 + x - 10
Производная от функции: 6*x^2 + 6*x + 1

Введите границы отрезка поиска [a;b]
Введите нижнюю границу отрезка a
=>0
Введите верхнюю границу отрезка b
=>1

Введите точность вычислений epsilon
=>0.000001

Error! В заявленном отрезке нет корня!

Вы хотите выйти из программы? Если да, введите '1', иначе - '0'
0
Введите границы отрезка поиска [a;b]
Введите нижнюю границу отрезка a
=>
```

Рис. 3. Работа программы с неверно заданным отрезком

Если пользователем задан отрезок поиска, содержащий корень функции, то программа проведет поиск корня и выведет таблицу приближений найденных в ходе ее работы. За начальное приближение берется нижняя граница отрезка a. На рис. 4 представлена работа программы при введенном отрезке [1;5] и погрешности ε= 0.00000001:



```
                                Метод Ньютона
           Задача программы: найти корень функции при помощи метода Ньютона
Функция: 2*x^3 + 3*x^2 + x - 10
Производная от функции: 6*x^2 + 6*x + 1

Введите границы отрезка поиска [a;b]
Введите нижнюю границу отрезка a
=>1
Введите верхнюю границу отрезка b
=>5

Введите точность вычислений epsilon
=>0.00000001
+----+------------+------------+------------+------------+------------+
| i  |    x_i     |   f(xi)    |   f'(xi)   |  x_i + 1   |  epsilon   |
+----+------------+------------+------------+------------+------------+
| 0  | 1,00000000 | -4,00000000| 13,00000000| 1,30769229 | 1,00000000|
| 1  | 1,30769229 | 0,91033268 | 19,10650826| 1,26004708 | 0,30769229|
| 2  | 1,26004708 | 0,02440357 | 18,08659363| 1,25869787 | 0,04764521|
| 3  | 1,25869787 | 0,00001907 | 18,05810928| 1,25869679 | 0,00134921|
| 4  | 1,25869679 | 0,00000095 | 18,05808640| 1,25869679 | 0,00000107|
+----+------------+------------+------------+------------+------------+
Приблизительное значение корня функции x при погрешности epsilon = 0,00000001:
 x = 1,25869679
Вы хотите выйти из программы? Если да, введите '1', иначе - '0'
1
```

Рис. 4. Работа программы с верно заданным отрезком

Значит, корень функции $f(x) = 2x^3+3x^2+x-10$ при погрешности $\varepsilon = 0.00000001$: $x \approx 1,25869679$

3. Заключение

В ходе работы была написана на языке программирования C++ программа, позволяющая реализовать метод касательных (Ньютона) для нахождения корня функции. Для иллюстрации работы программы была использована конкретная кубическая функция, однако код программы позволяет реализовать метод Ньютона и для других функций.

Для создания отчета о создание и работы программы был использован набор макрорасширений *LaTeX* системы компьютерной вёрстки *TeX*.

Список литературы

[1] Кнут Д.Э. Всё про TeX. — Москва: Изд. Вильямс, 2003 г. 550 с.

[2] Иванов А.П. Практикум по численным методам. Метод Ньютона. - Санкт-Петербург: 2013 г.

[3] Львовский С.М. Набор и верстка в системе LaTeX. — 3-е издание, исправленное и дополненное, 2003 г.

[4] Воронцов К.В. LaTeX в примерах. 2005 г.