Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии

ОТЧЕТ
по Лабораторной работе №4
по курсу
«Методы вычислений»
Тема
«Метод Ньютона»
Вариант 9

Студент ИУ7-21М Карпухин А.С.
(Группа) (И.О.Фамилия)

Преподаватель Власов П.А.
(И.О.Фамилия)

2021 г.

1. Постановка задачи

Общий вид задачи оптимизации:

$$\begin{cases} f(x) \to extr \\ x \in G \end{cases}$$

где $f: G \to \mathrm{R}$, $G \subseteq R^n$.

Если в данной задаче $G = [a, b] \subseteq \mathrm{R}$, то соответствующая задача называется задачей одномерной оптимизации.

Задача одномерной минимизации имеет вид:

$$\begin{cases} f(x) \to min \\ x \in [a, b] \end{cases}$$

2. Исходные данные варианта

Целевая функция f:

$$f = tg\frac{x^4 + 2x^2 - 2x + \sqrt{2} + 1}{8} + sin\frac{4x^3 - 7x - 9}{20x + 28}$$

a = 0, b = 1.

3. Метод Ньютона

Если предположить, что унимодальная на отрезке [a, b] функция f(x) дважды непрерывно дифференцируема на данном отрезке, то для ее минимизации можно использовать метод Ньютона, основанный на линейной аппроксимации производной функции f(x). Производная заменяется касательной к ней в некоторой начальной точке x_0, затем вычисляется точка пересечения этой касательной с осью абсцисс. Полученная точка берется в качестве следующего приближения искомой точки минимума функции f.

Очередное приближение точки минимума функции f вычисляется по формуле:

$$x_k = x_{k-1} - \frac{f'(x_{k-1})}{f''(x_{k-1})} \qquad (1)$$

В реализуемой модификации метода Ньютона вторая производная полагается равной константе:

$$f''(x_k) = f''(x_0) = const$$

Алгоритм метода Ньютона:

Шаг 1. Выбрать $x_0 \in [a, b]$. Шаг 2. Вычислить вторую производную $f''(x_0)$. Шаг 3. Вычислить очередное приближение x_k искомой точки минимума по формуле (1) с использованием ранее полученной второй производной $f''(x_0)$. Шаг 4. Проверка на окончание поиска: если разность между текущим и предыдущим приближением точки минимума меньше заданной точности ε, то закончить поиск, полагая $x^* = x_0$, $f^* = f(x_0)$. Иначе перейти к шагу 3.

Для аппроксимации производных в данном методе могут быть использованы их разностные аналоги:

$$f'(x_k) = \frac{f(x_{k+1}) - f(x_{k-1})}{x_{k+1} - x_{k-1}}$$

$$f''(x_k) = \frac{f(x_{k+1}) - f(x_k) + f(x_{k-1})}{(x_k - x_{k-1})^2}$$

4. Текст программы

Код разработанной программной реализации модифицированного метода Ньютона приведен в листинге 1.

Листинг 1. Код программы, реализующей модифицированный метод Ньютона

```
function lab4()
    f = @(x) (tan((x.^4 + 2.*x.^2 - 2.*x + 2^0.5 + 1) ./ 8) + ...
           sin((4.*x.^3 - 7.*x - 9) ./ (20.*x + 28)));

    a = 0; b = 1;
    X = a:0.01:b;
    Y = f(X);
    eps = [0.01, 0.0001, 0.000001];
```

```matlab
    NeutonMethodTable(f, a, b, X, Y, eps);
    FminbndTable(f, a, b, eps);
end

% Вывод таблицы для встроенной функции fminbnd
function FminbndTable(f, a, b, eps)
    fprintf("\nРезультаты вычисления точки минимума\n");
    fprintf("для различных значений точности функцией fminbnd:\n\n");
    fprintf(" # |    Eps    | N  |    x*    |   f(x*)\n");
    fprintf("---|----------|----|---------|--------\n");
    for i = 1:length(eps)
        options = optimset('TolX', eps(i));
        [x, fval, ~, output] = fminbnd(f, a, b, options);
        fprintf("%2i | %5f | %2i | %5.5f | %5.5f\n", ...
            i, eps(i), output.funcCount, x, fval);
    end
end

% Вывод таблицы и графиков для результатов работы метода Ньютона
function NeutonMethodTable(f, a, b, X, Y, eps)
    fprintf("\nРезультаты вычисления точки минимума\n");
    fprintf("для различных значений точности методом Ньютона:\n\n");
    fprintf("Eps - точность\n");
    fprintf("N - число обращений к целевой функции\n");
    fprintf("x* - найденная точка минимума функции\n");
    fprintf("f(x*) - найденный минимум функции\n\n");
    fprintf(" # |    Eps    | N  |    x*    |   f(x*)\n");
    fprintf("---|----------|----|---------|--------\n");

    figure('Units', 'normalized', 'OuterPosition', [0 0 1 1]);
    title('Метод Ньютона');

    for i = 1:length(eps)
        % Вычисление точки минимума и минимума функции
        [X0, F0, X_S, N] = NeutonMethod(a, b, f, eps(i));

        % Вывод строки таблицы результатов вычислений
        fprintf("%2i | %5f | %2i | %5.5f | %5.5f\n", ...
            i, eps(i), N, X0, F0);

        % Вывод графика для данной точности
        subplot(2, 2, i);
        % Целевая функция
        plot(X, Y, '-b','LineWidth',1.5);
        hold on;
        % Последовательность приближений
        plot(X_S, f(X_S), '*g','LineWidth', 2);
        % Точка минимума
        plot(X0, F0, '*r','LineWidth', 4);
        title(sprintf("Точность Eps = %2.0e", eps(i)));
        legend('Целевая функция', 'Последовательность приближений', ...
            'Точка минимума');
    end
end

% Разностная аппроксимация первой производной
function df = df(f_prev, f_next, delta)
    df = (f_next - f_prev) / (2*delta);
end
```

```
% Разностная аппроксимация второй производной
function d2f = d2f(f_prev, f_m, f_next, delta)
    d2f = (f_next - 2*f_m + f_prev) / (delta^2);
end

% Метод Ньютона
function [X, F, X0, N] =  NeutonMethod(a, b, f, eps)
    x0_prev = (a + b) / 3;
    x_prev = x0_prev - eps;
    x_next = x0_prev + eps;

    f_prev = f(x_prev);
    f_next = f(x_next);

    d2ff = d2f(f_prev, f(x0_prev), f_next, eps);

    X0 = [x0_prev];
    N = 3; % Число обращений к целевой функции

    while true
        % Очередное приближение точки минимума
        dff = df(f_prev, f_next, eps);
        x0 = x0_prev - dff / d2ff;
        X0 = [X0, x0];

        % Проверка условия завершения поиска
        if abs(x0_prev - x0) <= eps
            break;
        end

        % Вычисление точек для разностной аппроксимации производной
        x0_prev = x0;
        x_prev = x0 - eps;
        x_next = x0 + eps;
        f_prev = f(x_prev);
        f_next = f(x_next);
        N = N + 2;
    end

    X = x0;
    F = f_next;
end
```

5. Результаты расчётов

Результаты расчетов модифицированным методом Ньютона для задачи одномерной минимизации в соответствии с индивидуальным вариантом для различных значений ε приведены ниже в таблице 1.

Таблица 1. Результаты расчетов методом Ньютона.

№	ε	N	x^*	$f(x^*)$
1	10^{-2}	5	0.38361	-0.06526
2	10^{-4}	9	0.38379	-0.06533
3	10^{-6}	11	0.38379	-0.06533

В таблице 2 приведены результаты расчётов с использованием встроенной функции fminbnd пакета MATLAB.

Таблица 2. Результаты расчетов функцией fminbnd.

№	ε	N	x^*	$f(x^*)$
1	10^{-2}	6	0.38530	-0.06533
2	10^{-4}	8	0.38379	-0.06533
3	10^{-6}	9	0.38379	-0.06533

В таблице 3 приведены сводные данные, обобщающие результаты вычислений из работ 1 – 4 всеми использованными методами для точности 10^{-6}.

Таблица 3. Результаты вычислений из работ 1 – 4 для точности 10^{-6}.

№	Метод	N	x^*	$f(x^*)$
1	Поразрядного поиска	50	0.38379	-0.06533
2	Золотого сечения	30	0.38379	-0.06533
3	Парабол	11	0.38379	-0.06533
	Ньютона	11	0.38379	-0.06533
	Функция fminbnd	9	0.38379	-0.06533