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 №3
по курсу
«Методы вычислений»
Тема
«Метод парабол»
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 парабол

Метод парабол заключается в замене целевой функции на рассматриваемом отрезке полиномом второй степени и поиске точки минимума для данного полинома.

Изначально выбирается три точки x_1, x_2, x_3, через которые проводится парабола. Точка ее минимума вычисляется по формуле:

$$\tilde{x}^* = \frac{1}{2}\frac{f_1 r_{23} + f_2 r_{31} + f_3 r_{12}}{f_1 s_{23} + f_2 s_{31} + f_3 s_{12}} \qquad (1)$$

где $r_{ij} = x^2{}_i - x^2{}_j, s_{ij} = x_i - x_j, i, j = 1,2,3.$

Затем из полученных четырех точек выбирается три методом исключения отрезков на основе свойства унимодальных функций для следующей итерации.

Начальные точки x_1, x_2, x_3 выбираются так, чтобы удовлетворять условию:

$$x_1 < x_2 < x_3, f(x_1) \geq \mathrm{f}(x_2) \leq f(x_3) \qquad (2)$$

Алгоритм метода парабол:

Шаг 1. Выбрать x_1, x_2, x_3 согласно условию (2).

Шаг 2. Найти $\tilde{x}*$ по формуле (1). На первой итерации перейти к шагу 4. На остальных – к шагу 3.

Шаг 3. Проверка на окончание поиска: если длина отрезка $[x_1, x_3]$ или разность между предыдущим и текущим найденным значением $\tilde{x}*$ меньше точности ε, то завершить поиск, полагая $x^* = \tilde{x}*$, $f^* = f(\tilde{x}*)$. Иначе перейти к шагу 4.

Шаг 4. Вычислить значение $f(\tilde{x}*)$.

Шаг 5. Выбрать новую тройку x_1, x_2, x_3 для следующей итерации методом исключения отрезков и присвоить $f(x_1)$, $f(x_2)$, $f(x_3)$ соответствующие значения выбранных точек. Перейти к шагу 2.

Для выбора начальных точек предварительно используется метод золотого сечения, до тех пор, пока пробные точки метода, выбранные на очередной итерации, а также одна из крайних точек текущего отрезка не будут удовлетворять условию (2).

Текст программы

Код разработанной программной реализации метода парабол приведен в листинге 1.

Листинг 1. Код программы, реализующей метод парабол

```
function lab3()
    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];

    fprintf("\nРезультаты вычисления точки минимума\n");
    fprintf("для различных значений точности:\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 парабол');

    for i = 1:length(eps)
        % Вычисление точки минимума и минимума функции
        [X0, F0, A_S, B_S, N] = ParabolaMethod(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([A_S, B_S], [f(A_S), f(B_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

% Метод парабол
function [X, F, A_S, B_S, N] =  ParabolaMethod(a, b, f, eps)
    [X0, F0, A_S, B_S, N] = GoldenRatio(a, b, f);

    [x1, x2, x3] = deal(X0(1), X0(2), X0(3));
    [f1, f2, f3] = deal(F0(1), F0(2), F0(3));

    x_curr = x2;
    f_s = f2;
    is_first_iter = true;
```

```
    % Повторять, пока отрезок или разность между старым
    % и новым значением f(x*) не станет меньше точности
    while true
        A_S = [A_S, x1];
        B_S = [B_S, x3];

        r12 = x1^2 - x2^2;
        r23 = x2^2 - x3^2;
        r31 = x3^2 - x1^2;

        s12 = x1 - x2;
        s23 = x2 - x3;
        s31 = x3 - x1;

        % Вычисление точки минимума аппроксимирующей параболы
        x_prev = x_curr;
        x_curr = 0.5 * (f1*r23 + f2*r31 + f3*r12) / ...
            (f1*s23 + f2*s31 + f3*s12);

        % Проверка условия окончания поиска (на всех итерациях кроме
первой)
        if is_first_iter
            is_first_iter = false;
        elseif abs(x_prev - x_curr) <= eps
            break;
        end

        f_s = f(x_curr);
        N = N + 1;

        % Выбор тройки точек для следующей итерации
        if x_curr >= x2 && x_curr <= x3
            if f_s <= f2
                x1 = x2; f1 = f2;
                x2 = x_curr; f2 = f_s;
            else
                x3 = x_curr;
                f3 = f_s;
            end
        elseif x_curr >= x1 && x_curr <= x2
            if f_s <= f2
                x3 = x2; f3 = f2;
                x2 = x_curr; f2 = f_s;
            else
                x1 = x_curr;
                f1 = f_s;
            end
        end
    end
```

```
    X = x_curr;
    F = f(x_curr);
    N = N + 1;
end

% Метод золотого сечения для выбора начальных точек метода парабол
function [X, F,  A_S, B_S, N] =  GoldenRatio(a, b, f)
    tau = (5^0.5 - 1) / 2;

    A_S = [];
    B_S = [];

    l = b - a;

    % Вычисление пробных точек
    x1 = b - tau * l;
    x2 = a + tau * l;

    f1 = f(x1);
    f2 = f(x2);

    N = 2;

    while true
        A_S = [A_S, a];
        B_S = [B_S, b];

        % Проверка условия выбора начальных точек для метода парабол
        if (x1 < x2) && (x2 < b) && (f2 <= f1) && (f2 <= f(b))
            X = [x1, x2, b];
            F = [f1, f2, f(b)];
            N = N + 1;
            break;
        elseif (a < x1) && (x1 < x2) && (f1 <= f(a)) && (f1 <= f2)
            X = [a, x1, x2];
            F = [f(a), f1, f2];
            N = N + 1;
            break;
        end

        % Выбор новых точек для следующей итерации
        if (f1 >= f2)
            a = x1;
            l = b - a;

            x1 = x2;
            f1 = f2;

            x2 = a + tau * l;
            f2 = f(x2);
            N = N + 1;
```

```
        else
            b = x2;
            l = b - a;

            x2 = x1;
            f2 = f1;

            x1 = b - tau * l;
            f1 = f(x1);
            N = N + 1;
        end
    end
end
```

4. Результаты расчётов

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

Таблица 1. Результаты расчетов для задачи индивидуального варианта.

№	ε	N	x^*	$f(x^*)$
1	10^{-2}	6	0.38360	-0.06533
2	10^{-4}	7	0.38369	-0.06533
3	10^{-6}	11	0.8379	-0.06533