



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 ИТ, ИУ7

Отчет по лабораторной работе №1
по дисциплине
«Методы вычислений»

Тема: «Метод поразрядного поиска»

Вариант 16

Студент ИУ7-21М
Спасенов Иван

Москва 2021

Цель: изучение метода поразрядного поиска для решения задачи одномерной минимизации.

Задачи:

- реализовать метод поразрядного поиска в виде программы на ЭВМ
- провести решение задачи

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

для данных индивидуального варианта;

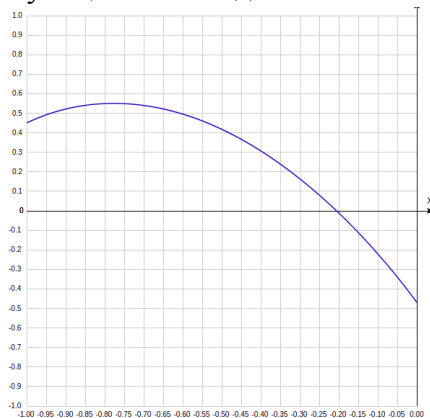
- организовать вывод на экран графика целевой функции, найденной точки минимума (x , $f(x)$) и последовательности точек (x_i , $f(x_i)$), приближающих точку искомого минимума (для последовательности точек следует предусмотреть возможность ”отключения” вывода ее на экран).

Данные варианта:

$$\ln(2x^5 - 7x + \sqrt{11}) + \operatorname{sh}\left(\frac{-4x^2 - 4x + 3 - 4\sqrt{2}}{3x^2 + 3x + 3\sqrt{2}}\right) - 1.0$$

Интервал: $[-1, 0]$.

Функция имеет один локальный максимум:

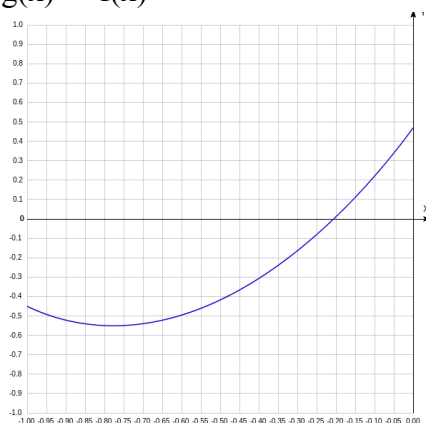


И два локальных минимума на краях интервала $[-1; 0]$

С учетом того, что для данной функции не заработает, например, корректно метод парабол, т.к. не выполняются условия:

1. $x_1 < x_2 < x_3$
2. $f(x_1) \geq f(x_2) \leq f(x_3)$

Было решено, для дальнейших корректных сравнений методов минимизации, взять функцию $g(x) = -f(x)$



И минимизировать уже её.

Краткое описание алгоритма:

1. Выбрать начальный шаг $sh=(b-a)/4$. Положить $x_0=a$. Вычислить $F(x_0)$.
2. Положить $x_1=x_0+sh$. Вычислить $F(x_1)$.
3. Сравнить $F(x_0)$ и $F(x_1)$. Если $F(x_0)>F(x_1)$, то перейти к шагу 4, иначе -- к шагу 5.
4. Положить $x_0=x_1$ и $F(x_0)=F(x_1)$. Проверить условие принадлежности x_0 интервалу $[a,b]$. Если $a < x_0 < b$, то перейти к шагу 2, иначе -- к шагу 5.
5. Проверка на окончание поиска: если $|sh| \leq \epsilon$, то вычисления завершить, полагая $x_m=x_0$, $F_m=F(x_0)$, иначе -- перейти к шагу 6.
6. Изменить направление поиска: положить $x_0=x_1$, $F(x_0)=F(x_1)$, $sh=-sh/4$. Перейти к шагу 2.

Текст программы:

```
% Точка входа
% Вариант 16
function lab1()
    a = -1;
    b = 0;
    eps = 10.^-6;
    % флаг отвечающий за вывод и отрисовку последовательности точек, приближающих точку искомого минимума
    additional_points = true;

    [x_opt, n, points] = bitwise_search(a, b, eps);

    fprintf('x точки минимума = %f\n', x_opt);
    fprintf('y точки минимума = %f\n', func(x_opt));
    fprintf('Количество вычислений значений функции = %d\n', n);
    if additional_points
        fprintf('Точки, приближающие точку искомого минимума\n');
        for i = 1:length(points)
            fprintf('Итерация %d = %3d\n', i, points(i));
        end
    end

    end
    my_plot(a, b, x_opt, func(x_opt), points, n, additional_points);
end

function y = func(x)
    y = log(2 * x.^5 - 7 * x + sqrt(11)) + sinh((-4 * x.^2 - 4 * x + 3 - 4 * sqrt(2)) / (3 * x.^2 + 3 * x + 3 * sqrt(2))) - 1.0;
    y = -y;
end

function my_plot(a, b, x_opt, y_opt, points, n, withPoints)
    x = a:(b - a)/10:b;
    y = arrayfun(@func, x);
    points_y = arrayfun(@func, points);
    plot(x_opt, y_opt, '-*r');
    hold on;
    if withPoints
        plot(points, points_y, 'og');
        hold on;
    end
    plot(x, y, 'b');
    hold off;
    title('лабораторная 1')
    xlabel('X');
    ylabel('Y');
    min_legend = strcat('Минимальное значение функции y=', num2str(y_opt), ', достигается при x=', num2str(x_opt));
    legend(min_legend);
end

function [x_res, n, points] = bitwise_search(a, b, eps)
```

```

n = 1;
x1 = a;
y1 = func(x1);
x0 = x1;
points = [];
y0 = y1;
delta = (b - a) / 4.;
while abs(delta) > eps
    x0 = x1;
    y0 = y1;
    x1 = x0 + delta;
    y1 = func(x1);
    n = n + 1;
    while y0 > y1 && a <= x0 && x0 <= b
        n = n + 1;
        x0 = x1;
        y0 = y1;
        x1 = x0 + delta;
        y1 = func(x1);
        n = n + 1;
    end

    points = [ points, x0 ];
    delta = -delta / 4;

end
x_res = x0;

end

```

Результаты работы:

№	eps	N	x*	f(x*)
1	10 ⁻²	18	-0.78125	-0.55048
2	10 ⁻⁴	47	-0.77661	-0.55052
3	10 ⁻⁶	74	-0.77665	-0.55052