Министерство науки и высшего образования
Федеральное государтсвенное бюджетное образовательное учреждение высшего образования
Югорский государственный университет

Отчет о лабораторной работе №2
по дисциплине «Методы оптимизации»

Выполнил

Студент группы 1162б
________________ Панчишин И. Р.
«___» _________ 2019 г.

Принял

Доцент ИЦЭ
________________ Самарин В. А.
«___» _________ 2019 г.

Ханты-Мансийск, 2019

Цель

Изучить прямые методы минимизации.

Задачи

1. Реализовать следующие три метода минимизации: дихотомии, золотого сечения и Фибоначчи.

2. Изучить зависимость числа вычислений функции (скорости работы) от заданной точности.

Ход работы

Реализовал требуемые методы на языке программирования Octave (свободная реализация Matlab). Исходный код, представленный ниже, позволяет определить минимум функции (унимодальной), визуализировать изменение отрезка поиска на каждой итерации, а также построить зависимость скорости поиска от заданной точности.

```
set(0, defaultaxesfontsize, 12)
set(0, defaulttextfontsize, 12)


% вспомогательные функции
% длина отрезка
function res = len(a, b)
    res = b - a;
end

function res = fibonacci(n)
    if n < 0
        res = -1;
    elseif n == 0
        res = 0;
    elseif n == 1
        res = 1;
    else
        res = fibonacci(n - 1) + fibonacci(n - 2);
    end
end

% анимация изменения отрезка
function shinkline(st, en, line)
    if isnan(line)
        return
    end

    X = get(line, XData);

    for d = st : len(st, en) / 30 : en
        if (st > en)
            X(2) = d;
            set(line, XData, X);
        else
            X(1) = d;
            set(line, XData, X);
        end

```

```
        pause(0.01);
    end

    pause(0.5);
end


% метод дихотомии
function [xm, ym, n] = dichotomy(f, a, b, e, line)
    %d = rand() * 2 * e;
    d = e;
    n = 0; % вычислений функции

    while (b - a) / 2 > e
        [x1 x2] = deal((a + b - d) / 2, (a + b + d) / 2);

        n = n + 2;
        if (f(x2) > f(x1))
            shinkline(b, x2, line);
            b = x2;
        else
            shinkline(a, x1, line);
            a = x1;
        end
    end

    xm = (a + b) / 2;
    ym = f(xm);
end

% метод золотого сечения
function [xm, ym, n] = gold(f, a, b, e, line)
    g = (sqrt(5) - 1) / 2;

    right = @(a, b) a + g * len(a, b);
    left = @(a, b) a + (1 - g) * len(a, b);

    [x1 x2] = deal(left(a, b), right(a, b));
    [y1 y2] = deal(f(x1), f(x2));
    n = 2;

    % точность для произвольной итерации
    %initLen = b - a;
    %1/2 * g^n * initLen

    while (b - a) / 2 > e
        ++n;
        if (y2 > y1)
            shinkline(b, x2, line);
            b = x2;
            [x2 y2] = deal(x1, y1);
            x1 = left(a, b);
            y1 = f(x1);
        else
            shinkline(a, x1, line);
            a = x1;
            [x1 y1] = deal(x2, y2);
            x2 = right(a, b);
            y2 = f(x2);
        end
```

```
    end

    xm = (a + b) / 2;
    ym = f(xm);
end

% метод Фибоначчи
function [xm, ym, n] = fib(f, a, b, e, line)
    minfib = (b - a) / e;
    k = 1;
    while minfib > fibonacci(k)
        ++k;
    end

    right = @(a, b, k) a + fibonacci(k + 1) / fibonacci(k + 2) * len(a, b);
    left = @(a, b, k) a + fibonacci(k) / fibonacci(k + 2) * len(a, b);

    [x1 x2] = deal(left(a, b, k), right(a, b, k));
    [y1 y2] = deal(f(x1), f(x2));
    n = 2;

    while k > 0
        ++n;
        if (y2 > y1)
            shinkline(b, x2, line);
            b = x2;
            [x2 y2] = deal(x1, y1);
            x1 = left(a, b, k);
            y1 = f(x1);
        else
            shinkline(a, x1, line);
            a = x1;
            [x1 y1] = deal(x2, y2);
            x2 = right(a, b, k);
            y2 = f(x2);
        end
        --k;
    end

    xm = (a + b) / 2;
    ym = f(xm);
end


% исходные данные
f = @(X) X.^4 + exp(-X)
X = -1:0.01:1;
[a b] = deal(0, 1)
e = 0.1


% поиск минимума
% построение
plot(X, f(X), Color, b);
xlabel(x);
ylabel(y);
hold on;

xm = fminbnd(f, a, b);
ym = f(xm);
```

```
plot(xm, ym, bo, LineWidth, 3);

line = plot([a b], [0 0], Color, r, LineWidth, 3);
%[xm ym] = dichotomy(f, a, b, e, line);
[xm ym] = gold(f, a, b, e, line);
%[xm ym] = fib(f, a, b, e, line);
plot(xm, ym, ro, LineWidth, 3);


% зависимость от точности
figure;
hold on;

Fm = {@dichotomy, @gold, @fib};
E = linspace(0.00001, 0.1, 20);
for i = 1:length(Fm)
    N = [];
    for e = E
        [xm ym n] = Fm{i}(f, a, b, e, NaN);
        N = [N n];
    end
    plot(E, N);
end

legend(Дихотомии, Золотого сечения, Фибоначчи)
xlabel(Погрешность)
ylabel(Вычислений)


pause
```

Результат нахождения минимума представлен на Рис. 1. Он близок к результату работы встроенной функции *fminbnd*.

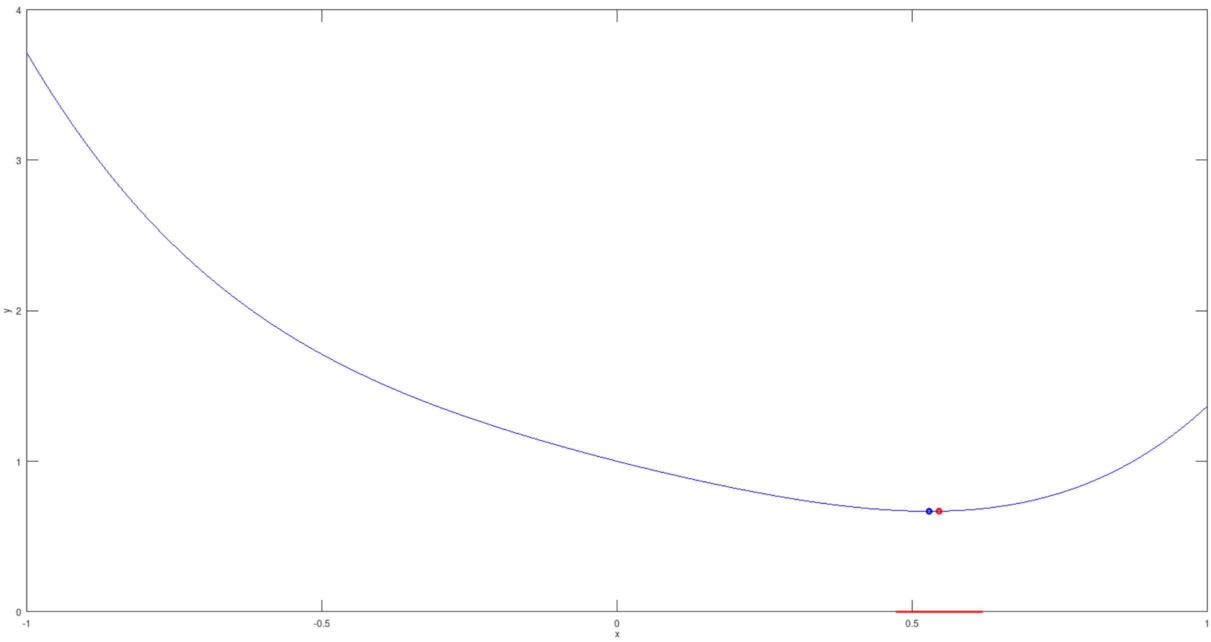

Рис. 1: Минимум функции

Зависимость числа вычислений от точности представлена на Рис. 2. Наблюдается экспоненциальный рост количества вычислений с увеличением точности. Метод золотого сечения показал себя лучше остальных, рассматриваемых, методов.

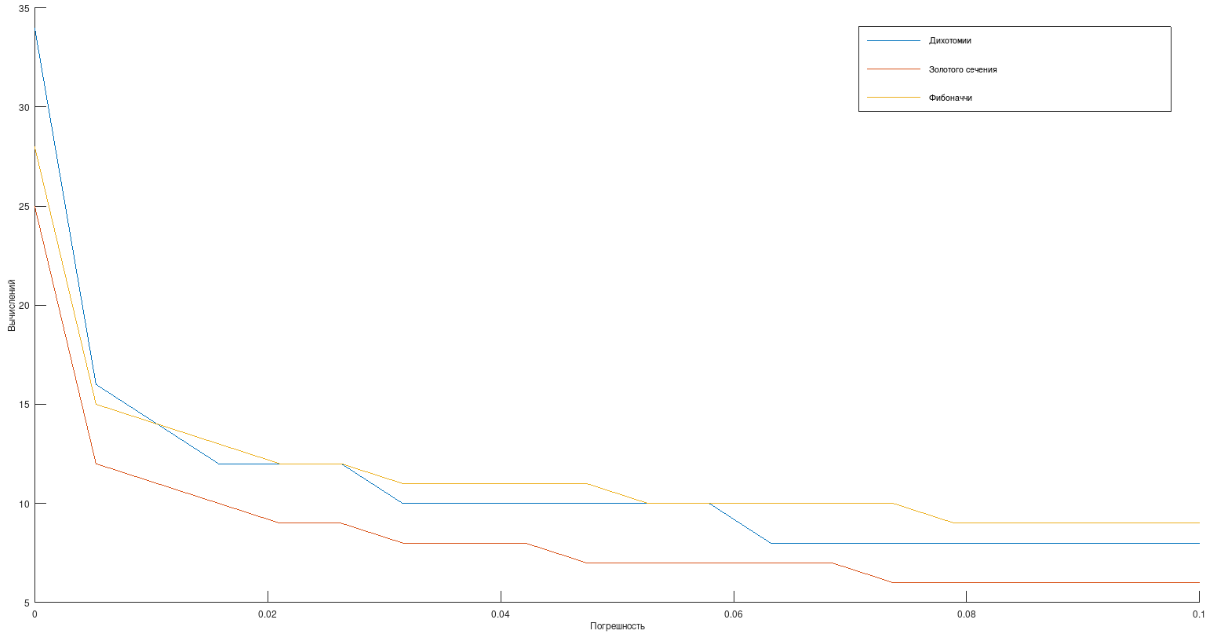

Рис. 2: Зависимость скорости от точности

Вывод

Реализовал прямые методы минимизации: дихотомии, золотого сечения и Фибоначчи. Сравнил их работу.