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

Отчет о лабораторной работе №3
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 метод парабол (полиномиальной интерполяции).

Ход работы

Реализовал метод парабол на языке Octave. Кроме самого метода в реализован алгоритм грубой локализации минимума, который используется для подбора выпуклой тройки точек, лежащих на уменьшенном отрезке поиска. Кроме грубой локализации можно использовать, например, метод золтого сечения или просто выбрать случайную точку, если отрезок поиска небольшой (рассматриваются унимодальные функции).

```
addpath(../code);

set(0, defaultaxesfontsize, 14);
set(0, defaulttextfontsize, 14);


% исходные данные
f = @(X) X.^4 + exp(-X)
X = linspace(0, 1, 100);
[a b] = deal(0, 1)
e = 0.01

% вывод функции
plot(X, f(X), Color, b);
xlabel(x);
ylabel(y);
hold on;

% минимум
xm = fminbnd(f, a, b)
ym = f(xm)
plot(xm, ym, bo, LineWidth, 3);


[xm ym n Approx] = parab(f, a, b, e)

% строим параболы

% интерполяционный квадратный многочлен Ньютона
g = @(a0, a1, a2, x1, x2, X) a0 + a1 * (X - x1) + a2 * (X - x1) .* (X - x2);

for Coef = Approx % итерация по строкам матрицы
    plot(X, g(Coef(1), Coef(2), Coef(3), Coef(4), Coef(5), X), Color, r);
    pause(1);
end

plot(xm, ym, ro, LineWidth, 3);


pause
```

```matlab
% метод парабол

function [xm, ym, n, Approx] = parab(f, a, b, e)
    Approx = [];

    % точки пересечений
    [x1 x3 n] = minloc(f, a, (b-a)/4); %или a b
    x2 = x1 + (x3 - x1) * rand(); %(a + b) / 2

    [y1 y2 y3] = deal(f(x1), f(x2), f(x3));
    n += 3;

    x42 = NaN;
    x41 = NaN;
    while true
        a0 = y1;
        a1 = (y2 - y1) / (x2 - x1);
        a2 = 1 / (x3 - x2) * ((y3 - y1) / (x3 - x1) - (y2 - y1) / (x2 - x1));

        Approx = [Approx; [a0, a1, a2, x1, x2]];

        x42 = x41;
        % минимум параболы
        x41 = 1/2 * (x1 + x2 - a1/a2);

        ++n;
        if (x41 > x2)
            [x1 y1] = deal(x2, y2);
            [x2 y2] = deal(x41, f(x41));
        else
            [x1 y1] = deal(x41, f(x41));
        end

        if (!isnan(x42) && abs(x41 - x42) <= e)
            break;
        end
    end

    xm = x41;
    ym = f(xm); ++n;
end
```

```matlab
% грубая локализация минимума

function [a, b, n] = minloc(f, x0, h)
    % направление убывания
    n = 2;
    while f(x0 + h) > f(x0)
        n += 2;
        if f(x0 - h) > f(x0)
            h = h / 2;
        else
            h = -h;
            break
        end

        n += 2;
    end

```

```
    x1 = x0 + h;

    n += 2;
    while f(x1) <= f(x0) % движение к локальному экстр.
        x0 = x1;
        x1 = x1 + h;

        n += 2;
    end

    x0 = x0 - h; % на случай, если перепрыгнули экстремум

    if x1 > x0, [a b] = deal(x0, x1);
    else [a b] = deal(x1, x0); end
end
```

Результат работы метода представлен на Рис. 1. Здесь изображена исходная функция со своим минимумом и аппроксимирующие параболы.

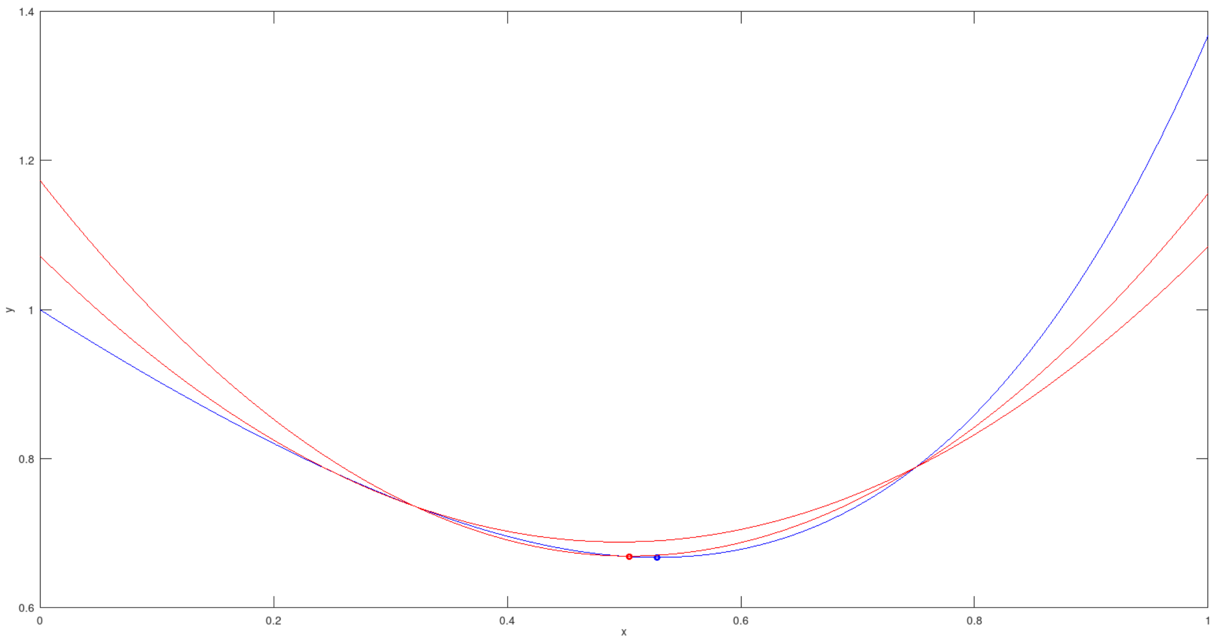

Рис. 1: Минимум функции

Вывод

Реализовал метод парабол, поставленную задачу выполнил.