

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

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №3
по дисциплине «Методы оптимизации»

Выполнил

Студент группы 11626

_____ 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 золотого сечения или просто выбрать случайную точку, если отрезок поиска небольшой (рассматриваются унимодальные функции).

```
1  addpath('./code');
2
3  set(0, defaultaxesfontsize, 14);
4  set(0, defaulttextfontsize, 14);
5
6
7  % исходные данные
8  f = @(X) X.^4 + exp(-X)
9  X = linspace(0, 1, 100);
10 [a b] = deal(0, 1)
11 e = 0.01
12
13 % вывод функции
14 plot(X, f(X), Color, b);
15 xlabel(x);
16 ylabel(y);
17 hold on;
18
19 % минимум
20 xm = fminbnd(f, a, b)
21 ym = f(xm)
22 plot(xm, ym, bo, LineWidth, 3);
23
24
25 [xm ym info] = parab(f, a, b, e)
26 title(['Парабол, Шаров', num2str(info.nstep), ', Вычислений', num2str(info.ncalc)]);
27
28 % строим параболы
29
30 % интерполяционный квадратный многочлен Ньютона
31 g = @(a0, a1, a2, x1, x2, X) a0 + a1 * (X - x1) + a2 * (X - x1) .* (X - x2);
32
33 for Coef = info.Approx % итерация по строкам матрицы
34     plot(X, g(Coef(1), Coef(2), Coef(3), Coef(4), Coef(5), X), Color, r);
35     pause(1);
36 end
37
38 plot(xm, ym, ro, LineWidth, 3);
39
40
41 pause
```

```

1  % Copyright © 2019 Panchishin Ivan
2
3  % метод парабол
4  % parabola method
5
6  % Approx - позволяет восстановить параболу
7
8  function [xm, ym, info] = parab(f, a, b, e)
9      % точки пересечений
10     [x1, x3, mlncalc] = minloc(f, a, (b - a) / 4) %или a b
11     x2 = x1 + (x3 - x1) * rand() % (a + b) / 2
12
13     [y1 y2 y3] = deal(f(x1), f(x2), f(x3));
14
15     info.nstep = 0;
16     info.ncalc = 4 + mlncalc;
17     info.Approx = [];
18
19     x42 = NaN;
20     x41 = NaN;
21     while true
22         a0 = y1;
23         a1 = (y2 - y1) / (x2 - x1);
24         a2 = 1 / (x3 - x2) * ((y3 - y1) / (x3 - x1) - (y2 - y1) / (x2 - x1));
25
26         ++info.nstep;
27         ++info.ncalc;
28         info.Approx = [info.Approx; [a0, a1, a2, x1, x2]];
29
30         x42 = x41;
31         % минимум параболы
32         x41 = 1/2 * (x1 + x2 - a1/a2);
33
34         if (x41 > x2)
35             [x1 y1] = deal(x2, y2);
36             [x2 y2] = deal(x41, f(x41));
37         else
38             [x1 y1] = deal(x41, f(x41));
39         end
40
41         if (!isnan(x42) && abs(x41 - x42) <= e)
42             break
43         end
44     end
45
46     xm = x41;
47     ym = f(xm);
48 end

```

```

1  % Copyright © 2019 Panchishin Ivan
2
3  % грубая локализация минимума
4  % rough minimum localization
5
6  function [a, b, ncalc] = minloc(f, x0, h)
7      ncalc = 2;
8

```

```

9      % определяем направление убывания
10     y0 = f(x0);
11     while (f(x0 + h) > y0)
12         ++ncalc;
13         if f(x0 - h) > y0
14             h = h / 2;
15         else
16             h = -h;
17         break
18     end
19
20     ++ncalc;
21 end
22
23 x1 = x0 + h;
24 y1 = f(x1);
25
26 while y1 <= y0 % движение к локальному экстр.
27     x0 = x1;
28     y0 = y1;
29
30     x1 = x1 + h;
31     y1 = f(x1);
32     ++ncalc;
33 end
34
35 x0 = x0 - h; % на случай, если перепрыгнули экстремум
36
37 if x1 > x0, [a b] = deal(x0, x1);
38 else [a b] = deal(x1, x0); end
39 end

```

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

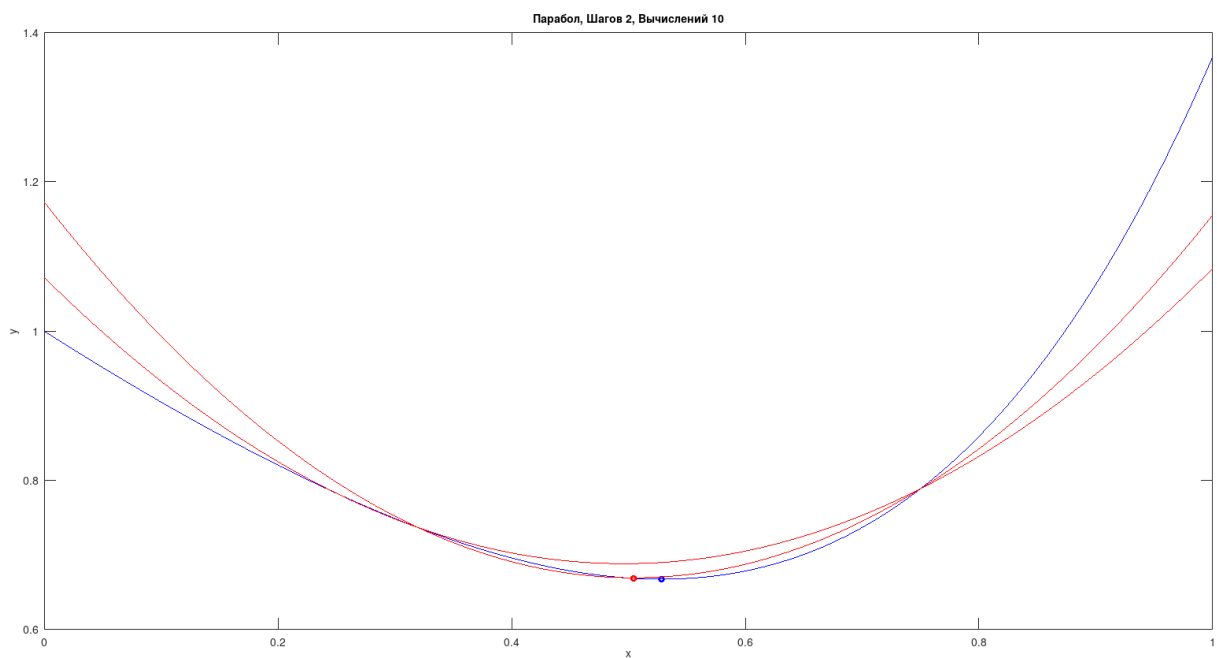


Рис. 1: Минимум функции

Вывод

Реализовал метод парабол, поставленную задачу выполнил.