

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

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №6  
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 метод градиентного спуска для задач минимизации.

## Задачи

1. Написать программную реализацию рассматриваемого метода.
2. Найти минимум функции, используя метод градиентного спуска.
3. Сравнить сходимость при различных значениях шага.

## Ход работы

В листинге ниже представлена программная реализация метода градиентного спуска:

---

```
1 set(0, defaultaxesfontsize, 12)
2 set(0, defaulttextfontsize, 12)
3
4
5 function [GX0, n] = grad(f, X0, y0, n)
6     h = 0.0001;
7     GX0 = [(f([X0(1) + h, X0(2)]) - y0) / h];
8     GX0 = [GX0, (f([X0(1), X0(2) + h]) - y0) / h];
9     n += 2;
10 end
11
12 function [Xm, ym, n, Approx] = graddesc(f, X0, h, e)
13     n = 0;
14     y0 = f(X0);
15     Approx = [];
16     [Xm, ym, n, Approx] = graddesc_step(f, X0, y0, h, e, n, Approx);
17 end
18
19 function [Xm, ym, n, Approx] = graddesc_step(f, X0, y0, h, e, n, Approx)
20     Approx = [Approx; [X0, y0]];
21
22     GX0 = grad(f, X0, y0, n);
23
24     % проверяем критерий останова
25     if sqrt(GX0(1)^2 + GX0(2)^2) <= e
26         Xm = X0;
27         ym = f(Xm); ++n;
28         return
29     end
30
31     while true
32         % перемещение точки вдоль антиградиента
33         X1 = X0 - GX0*h;
34
35         y1 = f(X1); ++n;
36         if (y1 < y0)
37             X0 = X1;
38             y0 = y1;
39             break;
40         end
41
42         h = h/2;
43     end
end
```

```

44
45     [Xm, ym, n, Approx] = graddesc_step(f, X0, y0, h, e, n, Approx);
46 end
47
48
49 % исходные данные
50 f = @(X) 3*X(1)^2 - 3*X(1)*X(2) + X(2)^2 + 7*X(1) - 7*X(2);
51 X1 = X2 = linspace(-10, 10, 50);
52 [XX1, XX2] = meshgrid(X1, X2);
53
54 YY = [];
55 for i = 1:length(X1)
56     Y = [];
57     for j = 1:length(X2)
58         Y = [Y, f([X1(i) X2(j)])];
59     end
60     YY = [YY; Y];
61 end
62
63 % вывод графика
64 surf(XX1, XX2, YY, edgcolor, none);
65 hold on
66 xlabel("x1");
67 ylabel("x2");
68 zlabel("y");
69
70 % поиск минимума
71 X0 = [1 -2];
72 [Xm ym] = fminunc(f, X0)
73 plot3(Xm(2), Xm(1), ym, b., MarkerSize, 40);
74
75 [Xm, ym, n, Approx] = graddesc(f, X0, 1, 0.8); n
76 plot3(Xm(2), Xm(1), ym, r., MarkerSize, 40);
77 plot3(Approx(:, 2), Approx(:, 1), Approx(:, 3), r, LineWidth, 3);
78
79 % количество вычислений в зависимости от масштаба (шага)
80 figure;
81 H = linspace(0.05, 1, 6);
82 for i = 1:length(H)
83     subplot(2, 3, i);
84     contour(XX1, XX2, YY, 20);
85     hold on;
86
87     [Xm, ym, n, Approx] = graddesc(f, X0, H(i), 0.8);
88     plot(Xm(2), Xm(1), r., MarkerSize, 20);
89     plot(Approx(:, 2), Approx(:, 1), r, LineWidth, 3);
90     title([Wbar , num2str(H(i)), , Вычислений , num2str(n)]);
91 end
92
93
94 pause

```

Результат поиска минимума функции из предыдущей лабораторной работы данным методом представлен на Рис. 1.

Количество вычислений функции при различном шаге отражено на Рис. 2. Каждый график представлен в виде набора уровней, где указана точка минимума, а также путь передвижения начального приближения.

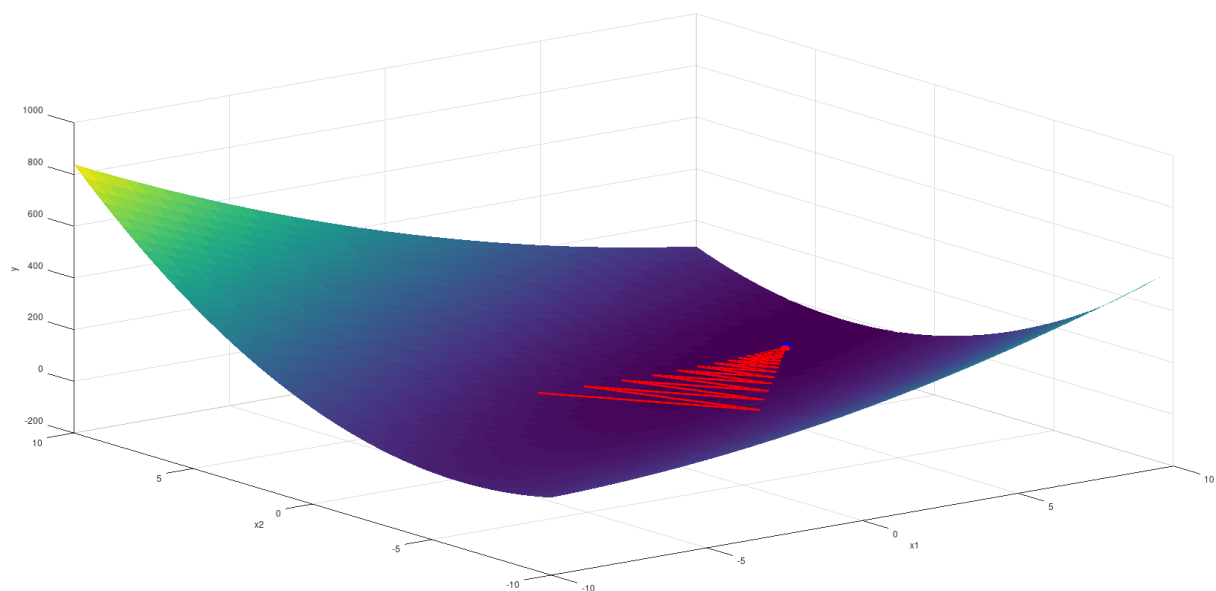


Рис. 1: Минимум функции

## Вывод

Выполнил все поставленные задачи. Наилучшую сходимость обеспечивает наименьший шаг, но наилучшую скорость работы алгоритма обеспечило среднее значение – 0,43 для 19 вычислений.

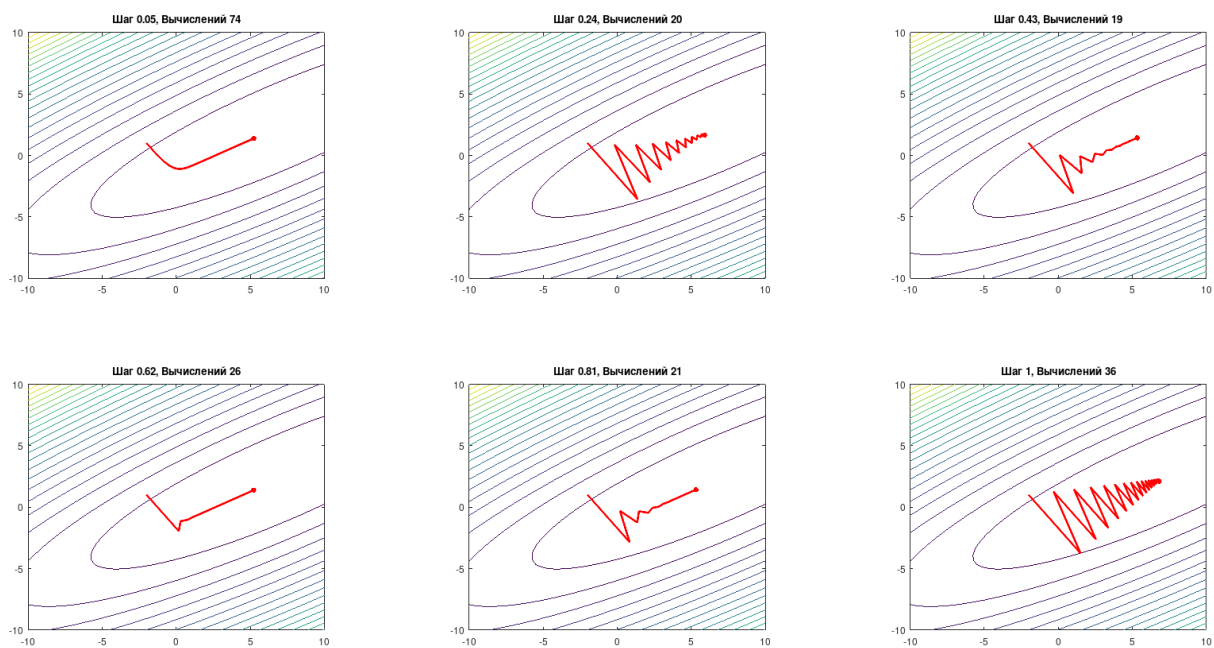


Рис. 2: Скорость работы при заданном шаге