

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

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №2
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 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.

```
1  addpath('./code');
2
3  set(0, defaultaxesfontsize, 14);
4  set(0, defaulttextfontsize, 14);
5
6
7  % целевая функция
8  f = @(X) X.^4 + exp(-X)
9  % область определения
10 X = -1:0.01:1;
11 % отрезок поиска
12 [a b] = deal(-0.5, 1)
13 e = 0.1
14
15 % минимум
16 [targetxm, targetym] = fminbnd(f, a, b)
17
18 M = {@dichotomy, @gold, @fibonacci};
19 Name = {Дихотомии, Золотого сечения, Фибоначчи};
20
21
22 % поиск минимума
23
24 for i = 1:length(M)
25     subplot(1, 3, i);
26
27     plot(X, f(X), Color, b);
28     xlabel(x);
29     ylabel(y);
30     hold on;
31
32     plot(targetxm, targetym, bo, LineWidth, 3);
33
34
35     [xm, ym, info] = M{i}(f, a, b, e);
36     title([Name{i} , Шаров num2str(info.nstep) , Вычислений num2str(info.ncalc)]);
37
38     segment = plot([info.Approx(1, 1) info.Approx(1, 2)], [0 0], Color, r, LineWidth, 3);
39     for j = 1:length(info.Approx) - 1
```

```

40         st = info.Approx(j, :);
41         en = info.Approx(j+1, :);
42         if st(1) != en(1)
43             shrinkseg(st(1), en(1), segment, 1);
44         end
45         if st(2) != en(2)
46             shrinkseg(st(2), en(2), segment, 1);
47         end
48     end
49
50     plot(xm, ym, ro, LineWidth, 3);
51 end
52
53 % зависимость от точности
54 figure;
55 hold on;
56
57 E = linspace(0.0001, 0.2, 100);
58 for i = 1:length(M)
59     N = [];
60     for e = E
61         [xm ym info] = M{i}(f, a, b, e);
62         N = [N info.ncalc];
63     end
64     plot(E, N);
65 end
66
67 legend(Name);
68 xlabel(Погрешность);
69 ylabel(Вычислений);
70
71
72 pause

```

```

1  % Copyright © 2019 Panchishin Ivan
2
3  % метод дихотомии
4  % dichotomy method
5
6  function [xm, ym, info] = dichotomy(f, a, b, e)
7      info.nstep = 0;
8      info.ncalc = 1;
9      info.Approx = [a, b];
10
11      d = e; %rand() * 2 * e;
12
13      while ((b - a) / 2 > e)
14          [x1, x2] = deal((a + b - d) / 2, (a + b + d) / 2);
15
16          if (f(x2) > f(x1))
17              b = x2;
18          else
19              a = x1;
20          end
21
22          ++info.nstep;
23          info.ncalc += 2;
24          info.Approx = [info.Approx; [a, b]];
25      end
26

```

```

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

```

```

1  % Copyright © 2019 Panchishin Ivan
2
3  % метод Фибоначчи
4  % fibonacci method
5
6  function [xm, ym, info] = fibonacci(f, a, b, e)
7      info.nstep = 0;
8      info.ncalc = 3;
9      info.Approx = [a, b];
10
11     k = 0;
12     while (b - a) / e > fibnum(++k) end
13
14     right = @(a, b, k) a + fibnum(k - 1) / fibnum(k) * len(a, b);
15     left = @(a, b, k) a + fibnum(k - 2) / fibnum(k) * len(a, b);
16
17     [x1, x2] = deal(left(a, b, k), right(a, b, k));
18     [y1, y2] = deal(f(x1), f(x2));
19
20     while (k > 3) % последние числа 1/3 и 2/3
21         if (y2 > y1)
22             b = x2;
23             [x2, y2] = deal(x1, y1);
24             x1 = left(a, b, k);
25             y1 = f(x1);
26         else
27             a = x1;
28             [x1, y1] = deal(x2, y2);
29             x2 = right(a, b, k);
30             y2 = f(x2);
31         end
32
33         --k;
34
35         ++info.nstep;
36         ++info.ncalc;
37         info.Approx = [info.Approx; [a, b]];
38     end
39
40     xm = (a + b) / 2;
41     ym = f(xm);
42 end
43
44 function res = fibnum(n)
45     if (n < 0)
46         res = -1;
47     elseif (n == 0)
48         res = 0;
49     elseif (n == 1)
50         res = 1;
51     else
52         res = fibnum(n - 1) + fibnum(n - 2);
53     end
54 end

```

```

1  % Copyright © 2019 Panchishin Ivan
2
3  % метод золотого сечения
4  % golden-section method
5
6  function [xm, ym, info] = gold(f, a, b, e)
7      info.nstep = 0;
8      info.ncalc = 3;
9      info.Approx = [a, b];
10
11     g = (sqrt(5) - 1) / 2;
12     right = @(a, b) a + g * len(a, b);
13     left = @(a, b) a + (1 - g) * len(a, b);
14
15     [x1 x2] = deal(left(a, b), right(a, b));
16     [y1 y2] = deal(f(x1), f(x2));
17
18     % точность для произвольной итерации
19     %initLen = b - a;
20     %1/2 * g^n * initLen
21
22     while ((b - a) / 2 > e)
23         if (y2 > y1)
24             b = x2;
25             [x2, y2] = deal(x1, y1);
26             x1 = left(a, b);
27             y1 = f(x1);
28         else
29             a = x1;
30             [x1, y1] = deal(x2, y2);
31             x2 = right(a, b);
32             y2 = f(x2);
33         end
34
35         ++info.nstep;
36         ++info.ncalc;
37         info.Approx = [info.Approx; [a, b]];
38     end
39
40     xm = (a + b) / 2;
41     ym = f(xm);
42 end

```

```

1  % длина отрезка
2
3  function res = len(a, b)
4      res = b - a;
5  end

```

```

1  % Copyright © 2019 Panchishin Ivan
2
3  % анимация изменения отрезка
4
5  function shrinkseg(st, en, line, timeout)
6      X = get(line, XData);
7
8      for d = st : len(st, en) / 30 : en
9          if (st > en)

```

```

10         X(2) = d;
11         set(line, XData, X);
12     else
13         X(1) = d;
14         set(line, XData, X);
15     end
16
17     pause(0.01);
18 end
19
20 pause(timeout);
21 end

```

Найденный каждым методом минимум отмечен красной точкой на Рис. 1. Синей точкой отмечен минимум, найденный при помощи встроенной функции *fminbnd*. Минимумы не совпадают, так как заданная точность не достаточно велика. Отрезок, выделенный красным, является конечным отрезком поиска минимума.

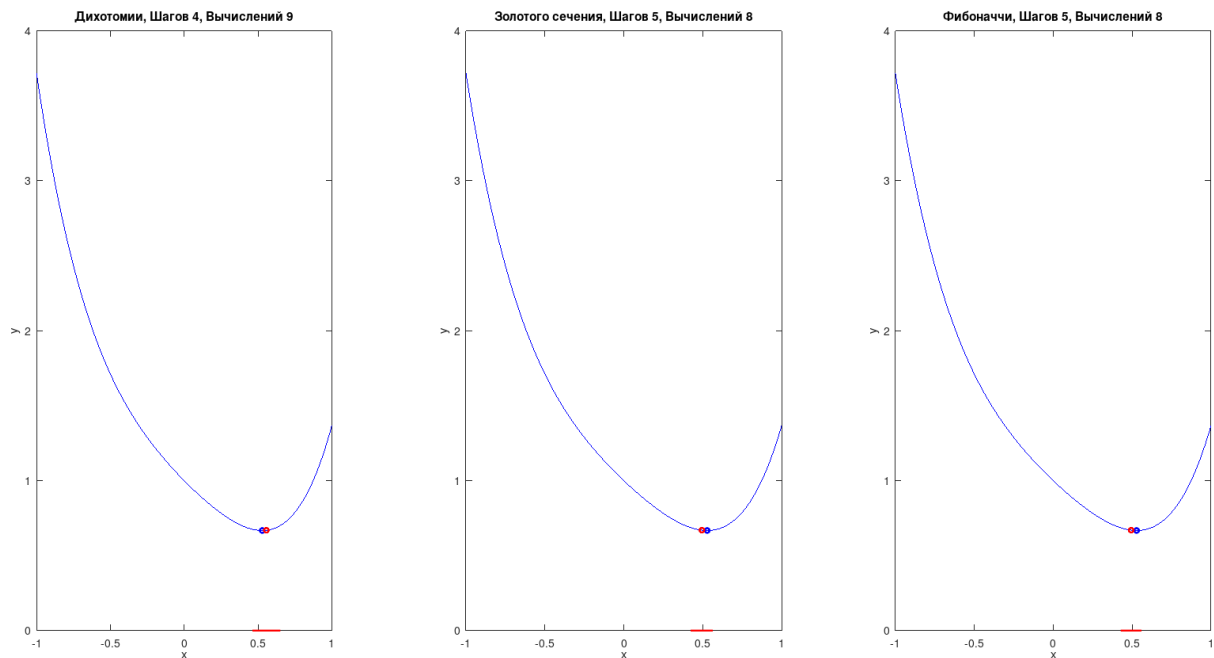


Рис. 1: Минимум функции

Зависимость числа вычислений от точности представлена на Рис. 2. Наблюдается экспоненциальный рост количества вычислений с увеличением точности.

Вывод

Реализовал прямые методы минимизации: дихотомии, золотого сечения и Фибоначчи. Сравнил их работу. Метод золотого сечения показал себя лучше остальных рассмотренных методов, однако его преимущество становится ощутимым только при относительно высокой точности.

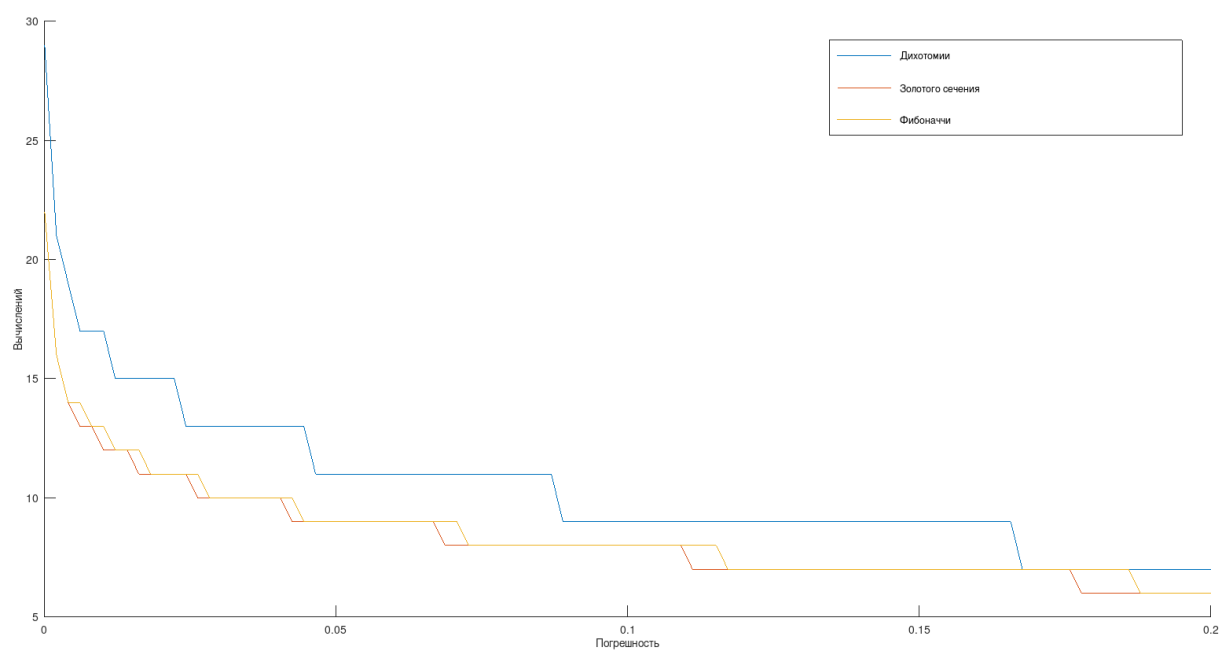


Рис. 2: Зависимость скорости от точности