

Университет ИТМО

Факультет программной инженерии и компьютерной техники

Лабораторная работа №2

по курсу «Методы оптимизации»

Использование численных методов решения задач одномерной
безусловной оптимизации.

Выполнил:

Студент группы Р3230

Пономаренко Алиса Валерьевна

Преподаватель:

Селина Елена Георгиевна

Санкт-Петербург

2024

Условие задачи

$$f(x) = x^3 - 3\sin(x); \quad [a; b] = [0; 1]; \quad e = 0.001.$$

Исследуем функцию на экстремумы с помощью первой производной.

$$f'(x) = 3x^2 - 3\cos(x)$$

$$f'(a) = f'(0) = -3$$

$$f'(b) = f'(1) \approx 1.37909$$

Производные в концах отрезка имеют разные знаки ($f'(a) < 0$ и $f'(b) > 0$), а значит, что на отрезке $[a; b]$ есть точка локального **минимума**.

Построим график, чтобы убедиться в правильности алгоритмов.

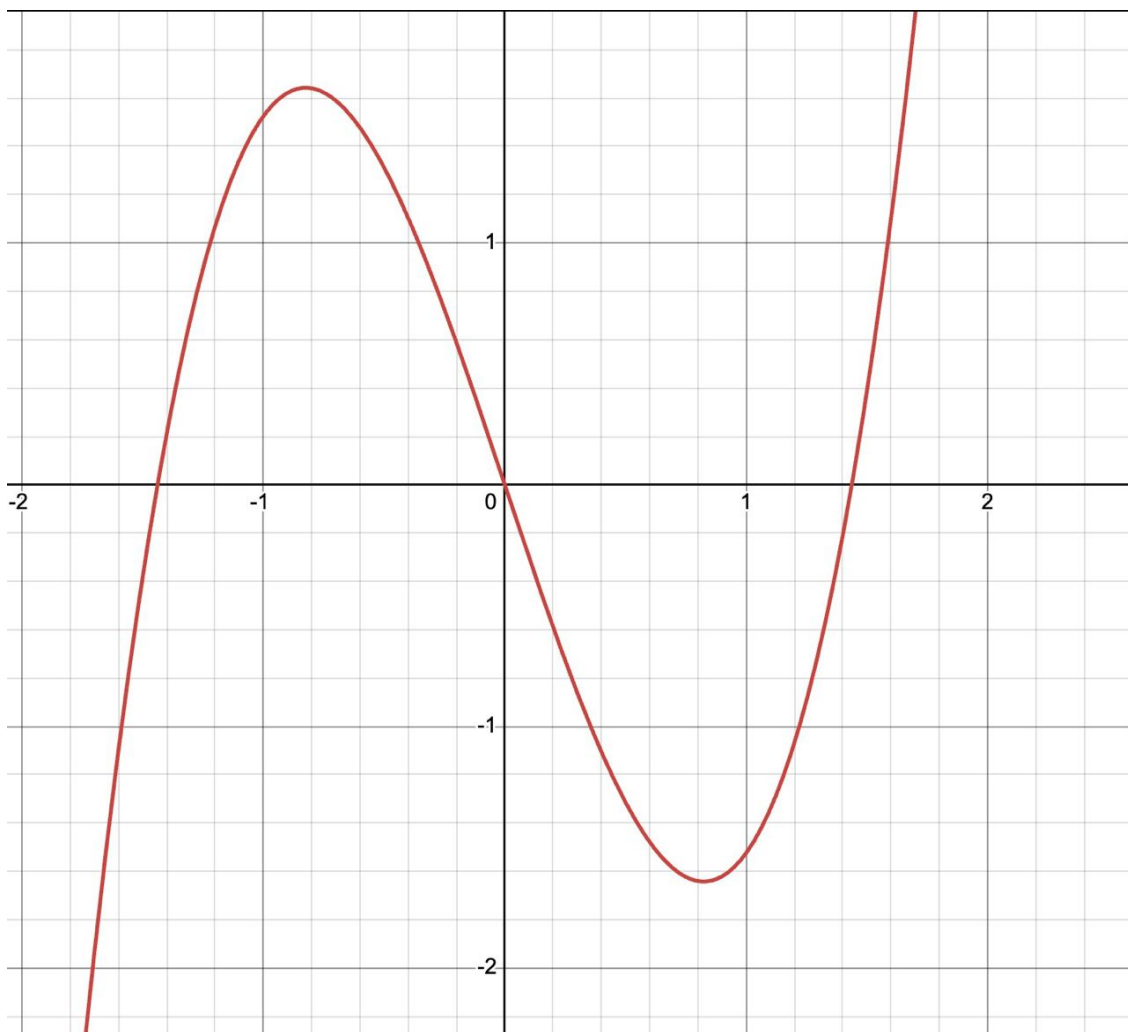


Рис. 1: График исходной функции

Метод половинного деления

Ручная проверка:

$$f(x) = x^3 - 3 \sin x ; [a; b] = [0; 1] ; \varepsilon = 0,001.$$

Метод половинного деления:

$$1) \quad x_1 = \frac{0+1-0,001}{2} = 0,4995 ; y_1 = f(0,4995) = 0,4995^3 - 3 \sin 0,4995 \approx -1,31233$$

$$x_2 = \frac{0+1+0,001}{2} = 0,5005 ; y_2 = f(0,5005) = 0,5005^3 - 3 \sin 0,5005 \approx -1,31422 \quad y_2 < y_1$$

$$\Rightarrow a = x_1 = 0,4995 ; b - a = 1 - 0,4995 = 0,5005 > 0,002 = 2\varepsilon$$

$$2) \quad x_1 = \frac{0,4995+1-0,001}{2} = 0,74925 ; y_1 \approx -1,62266$$

$$x_2 = \frac{0,4995+1+0,001}{2} = 0,75025 ; y_2 \approx -1,62317 \quad y_2 < y_1$$

$$\Rightarrow a = x_1 = 0,74925 ; b - a = 1 - 0,74925 = 0,25075 > 0,002 = 2\varepsilon$$

$$3) \quad x_1 = \frac{0,74925+1-0,001}{2} = 0,874125 ; y_1 \approx -1,63303$$

$$x_2 = \frac{0,74925+1+0,001}{2} = 0,875125 ; y_2 \approx -1,63266 \quad y_1 < y_2$$

$$\Rightarrow b = x_2 = 0,875125 ; b - a = 0,875125 - 0,74925 = 0,125875 > 0,002 = 2\varepsilon$$

$$4) \quad x_1 = \frac{0,74925+0,875125-0,001}{2} = 0,8116875 ; y_1 \approx -1,64158$$

$$x_2 = \frac{0,74925+0,875125+0,001}{2} = 0,8126875 ; y_2 \approx -1,64166 \quad y_2 < y_1$$

$$\Rightarrow a = x_1 = 0,8116875 ; b - a = 0,875125 - 0,8116875 = 0,0634375 > 0,002 = 2\varepsilon$$

$$5) \quad x_1 = \frac{0,8116875+0,875125-0,001}{2} = 0,84290625 ; y_1 \approx -1,64086$$

$$x_2 = \frac{0,8116875+0,875125+0,001}{2} = 0,84390625 ; y_2 \approx -1,64072 \quad y_1 < y_2$$

$$\Rightarrow b = 0,84390625 ; b - a = 0,84390625 - 0,8116875 = 0,03221875 > 0,002 = 2\varepsilon$$

Выйдя из цикла после 5-ти итераций: $x_m = \frac{a+b}{2} = \frac{0,8116875+0,84390625}{2} = 0,82779688$

$$y_m = f(x_m) \approx -1,64208$$

Программа на Java:

```
1 package org.example.optimizationMethods;
2
3 public class Point {
4     private double x;
5     private double y;
6
7     public Point() {
8     }
9
10    public Point(double x, double y) {
11        this.x = x;
12        this.y = y;
13    }
14
15    public double getX() {
16        return x;
17    }
18
19    public void setX(double x) {
20        this.x = x;
21    }
22
23    public double getY() {
24        return y;
25    }
26
27    public void setY(double y) {
28        this.y = y;
29    }
30 }
31
32
33 package org.example.optimizationMethods;
34
35 public class DoubleDevMethod {
36     public static void main(String[] args) {
37         Point point = getExtremumPoint(0, 1, 0.001);
38         System.out.println(point.getX());
39         System.out.println(point.getY());
40     }
41
42     private static double getValueOfFunction(double x) {
43         return Math.pow(x, 3) - 3 * Math.sin(x);
44     }
45 }
```

```

46 private static Point getExtremumPoint(double leftSegmentEnd, //
47     double rightSegmentEnd, double epsilon) {
48     double endPointDifference;
49     Point answerPoint = new Point();
50     double x1 = (leftSegmentEnd + rightSegmentEnd - epsilon) / 2;
51     double x2 = (leftSegmentEnd + rightSegmentEnd + epsilon) / 2;
52     double y1 = getValueOfFunction(x1);
53     double y2 = getValueOfFunction(x2);
54
55     if (y2 < y1) {
56         leftSegmentEnd = x1;
57         endPointDifference = rightSegmentEnd - leftSegmentEnd;
58     } else {
59         rightSegmentEnd = x2;
60         endPointDifference = rightSegmentEnd - leftSegmentEnd;
61     }
62     while (endPointDifference > 2 * epsilon) {
63         x1 = (leftSegmentEnd + rightSegmentEnd - epsilon) / 2;
64         x2 = (leftSegmentEnd + rightSegmentEnd + epsilon) / 2;
65         y1 = getValueOfFunction(x1);
66         y2 = getValueOfFunction(x2);
67         if (y2 < y1) {
68             leftSegmentEnd = x1;
69             endPointDifference = rightSegmentEnd - leftSegmentEnd;
70         } else {
71             rightSegmentEnd = x2;
72             endPointDifference = rightSegmentEnd - leftSegmentEnd;
73         }
74     }
75     double resultX = (leftSegmentEnd + rightSegmentEnd) / 2;
76     double resultY = getValueOfFunction(resultX);
77     answerPoint.setX(resultX);
78     answerPoint.setY(resultY);
79     return answerPoint;
80 }
81 }

```

Вывод программы:

0.82438232421875

-1.6421301895385456

Метод золотого сечения

Ручная проверка:

$f(x) = x^3 - 3 \sin x$; $[a, b] = [0, 1]$; $\varepsilon = 0,001$.
 Метод золотого сечения; $\Phi \approx 0,618$

1) $x_1 = b - \frac{b-a}{\Phi} = 1 - \frac{1-0}{0,618} = 0,382$; $y(x_1) \approx -1,06259$
 $x_2 = a + \frac{b-a}{\Phi} = \frac{1}{\Phi} = 0,618$; $y(x_2) \approx -1,50219$ $y_2 < y_1 \Rightarrow$

$a = x_1 = 0,382$;
 $x_1 = x_2 = 0,618$;
 $x_2 = b - (x_1 - a) = 1 - 0,618 + 0,382 = 0,764$

$y_1 = y_2 = -1,50219$
 $y_2 = f(x_2) = f(0,764) = -1,6295$

$\frac{b-a}{2} = \frac{1-0,382}{2} = 0,309 > 2\varepsilon = 0,002$

2) $y_2 < y_1 \Rightarrow$ $a = x_1 = 0,618$
 $x_1 = x_2 = 0,764$
 $x_2 = b - (x_1 - a) = 1 - 0,764 + 0,618 = 0,854$

$y(x_1) = y(0,764) \approx -1,6295$ — вычисл. на прошлом шаге.
 $y(x_2) = y(0,854) \approx -1,63891$

3) $y_2 < y_1 \Rightarrow$ $a = x_1 = 0,764$
 $x_1 = x_2 = 0,854$
 $x_2 = b - x_1 + a = 0,91$

$\frac{b-a}{2} = \frac{1-0,764}{2} > 2\varepsilon$
 $y_1 = -1,63891$
 $y_2 = -1,61494$

4) $y_1 < y_2 \Rightarrow$ $b = x_2 = 0,91$
 $x_2 = x_1 = 0,854$
 $x_1 = a + b - x_2 = 0,764 + 0,91 - 0,854 = 0,82$

$\frac{b-a}{2} > 2\varepsilon$
 $y_1 = f(0,82) \approx -1,64201$
 $y_2 = f(0,91) \approx -1,61494$

5) $y_1 < y_2 \Rightarrow$ $b = x_2 = 0,854$
 $x_2 = x_1 = 0,82$
 $x_1 = a + b - x_2 = 0,764 + 0,854 - 0,82 = 0,798$

$y(x_1) = y(0,798) \approx -1,63971$
 $y(x_2) = y(0,82) \approx -1,64201$

После 5-ти шагов получили точку, с минимальным значением функции — $x_* = 0,82$
 $f(x_*) = -1,64201$

Рис. 2: Метод золотого сечения. Первые пять операций.

Программа на Java:

```

1 package org.example.optimizationMethods;
2
3 public class GoldSectionMethod {
4     public static void main(String[] args) {
5         Point point = getExtremumPoint(0, 1, 0.001);
6         System.out.println(point.getX());
7         System.out.println(point.getY());
8     }
9
10    private static double getValueOfFunction(double x) {
11        return Math.pow(x, 3) - 3 * Math.sin(x);
12    }
13

```

```

14     private static Point getExtremumPoint(double leftSegmentEnd,
15     double rightSegmentEnd, double epsilon) {
16         Point answerPoint = new Point();
17         double goldSection = 1.61803398875;
18         double x1 = rightSegmentEnd - (rightSegmentEnd -
19         leftSegmentEnd) / goldSection;
20         double x2 = leftSegmentEnd + (rightSegmentEnd -
21         leftSegmentEnd) / goldSection;
22         double y1 = getValueOfFunction(x1);
23         double y2 = getValueOfFunction(x2);
24         if (y1 < y2) {
25             rightSegmentEnd = x2;
26             x2 = x1;
27             x1 = leftSegmentEnd + rightSegmentEnd - x2;
28         } else {
29             leftSegmentEnd = x1;
30             x1 = x2;
31             x2 = rightSegmentEnd - x1 + leftSegmentEnd;
32         }
33         double difference = rightSegmentEnd - leftSegmentEnd;
34         while (difference > 2 * epsilon) {
35             if (y1 < y2) {
36                 rightSegmentEnd = x2;
37                 x2 = x1;
38                 x1 = leftSegmentEnd + rightSegmentEnd - x2;
39                 y1 = getValueOfFunction(x1);
40                 y2 = getValueOfFunction(x2);
41                 difference = rightSegmentEnd - leftSegmentEnd;
42             } else {
43                 leftSegmentEnd = x1;
44                 x1 = x2;
45                 x2 = rightSegmentEnd - x1 + leftSegmentEnd;
46                 y1 = getValueOfFunction(x1);
47                 y2 = getValueOfFunction(x2);
48                 difference = rightSegmentEnd - leftSegmentEnd;
49             }
50         }
51         if (y1 < y2) {
52             answerPoint.setX(x1);
53             answerPoint.setY(y1);
54         } else {
55             answerPoint.setX(x2);
56             answerPoint.setY(y2);
57         }
58         return answerPoint;
59     }

```

Вывод программы:

0.8239519738271477

-1.6421302967101874

Метод хорд

Ручная проверка:

$$f(x) = x^3 - 3 \sin x; [a; b] = [0; 1]; \varepsilon = 0,001.$$

Метод хорд

$$f'(x) = 3x^2 - 3 \cos x;$$

$$1) \tilde{x} = a - \frac{f'(a)}{f'(a) - f'(b)} (a - b) = \frac{f'(a)}{f'(a) - f'(b)} = \frac{-3}{-3 - 3 + 3 \cos 1} = -\frac{1}{\cos 1 - 2} \approx 0,685$$

$$f(\tilde{x}) = f(0,685) \approx -1,5766; f'(\tilde{x}) = -0,9156$$

$$f'(a) < 0; f'(b) > 0; f'(\tilde{x}) < 0 \Rightarrow [\tilde{x}; b] = [0,685; 1]$$

$$2) \tilde{x} = 0,685 + \frac{0,9156}{-0,9156 - 1,379} (0,685 - 1) \approx 0,81$$

$$f'(b) = 3 - 3 \cos 1 = 1,379$$

$$f'(\tilde{x}) = f'(0,81) \approx -0,1 < 0 \Rightarrow [\tilde{x}; b] = [0,81; 1]$$

$$3) \tilde{x} = 0,81 + \frac{0,1}{-0,1 - 1,379} (0,81 - 1) \approx 0,823$$

$$f'(\tilde{x}) = f'(0,823) \approx -0,008 < 0 \Rightarrow [\tilde{x}; b] = [0,823; 1]$$

$$4) \tilde{x} = 0,823 + \frac{0,008}{-0,008 - 1,379} (0,823 - 1) \approx 0,824$$

$$f'(\tilde{x}) = f'(0,824) \approx -0,0009 < 0 \Rightarrow [\tilde{x}; b] = [0,824; 1]$$

$$5) \tilde{x} = 0,824 + \frac{0,0009}{-0,0009 - 1,379} (0,824 - 1) \approx 0,8241$$

$$f(\tilde{x}) \approx -1,64213;$$

Итоговая точка после 5-ти операций: (0,8241; -1,64213)

Рис. 3: Метод хорд. Первые пять операций.

Программа на Java:

```
1 package org.example.optimizationMethods;
2
3 public class SegmentationMethod {
4     public static void main(String[] args) {
5         Point point = getExtremumPoint(0, 1, 0.001);
6         System.out.println(point.getX());
7         System.out.println(point.getY());
8     }
9
10    private static double getValueOfFunction(double x) {
```



```

11         return Math.pow(x, 3) - 3 * Math.sin(x);
12     }
13
14     private static double getDerivativeOfFunction(double x) {
15         return 3 * Math.pow(x, 2) - 3 * Math.cos(x);
16     }
17
18     private static Point getExtremumPoint(double leftSegmentEnd,
19     double rightSegmentEnd, double epsilon) {
20         double endPointDifference;
21         Point answerPoint = new Point();
22         double x = leftSegmentEnd - ((leftSegmentEnd -
23     rightSegmentEnd) * getDerivativeOfFunction(leftSegmentEnd) / (
24     getDerivativeOfFunction(leftSegmentEnd) - getDerivativeOfFunction
25     (rightSegmentEnd)));
26         double derX = getDerivativeOfFunction(x);
27         double derA = getDerivativeOfFunction(leftSegmentEnd);
28         if (derA * derX < 0) {
29             leftSegmentEnd = x;
30             endPointDifference = rightSegmentEnd - leftSegmentEnd;
31         } else {
32             rightSegmentEnd = x;
33             endPointDifference = rightSegmentEnd - leftSegmentEnd;
34         }
35         while (endPointDifference > 2 * epsilon) {
36             x = leftSegmentEnd - ((leftSegmentEnd - rightSegmentEnd)
37     * getDerivativeOfFunction(leftSegmentEnd) / (
38     getDerivativeOfFunction(leftSegmentEnd) - getDerivativeOfFunction
39     (rightSegmentEnd)));
40             derX = getDerivativeOfFunction(x);
41             derA = getDerivativeOfFunction(leftSegmentEnd);
42             if (derA * derX < 0) {
43                 leftSegmentEnd = x;
44                 endPointDifference = rightSegmentEnd - leftSegmentEnd
45             ;
46             } else {
47                 rightSegmentEnd = x;
48                 endPointDifference = rightSegmentEnd - leftSegmentEnd
49             ;
50             }
51         }
52         double resultX = (leftSegmentEnd + rightSegmentEnd) / 2;
53         double resultY = getValueOfFunction(resultX);
54         answerPoint.setX(resultX);
55         answerPoint.setY(resultY);
56         return answerPoint;
57     }
58 }

```

Вывод программы:

0.8354286327834163

-1.6416725013479152

Метод Ньютона

Ручная проверка:

$$f(x) = x^3 - 3\sin x; [a; b] = [0; 1]; \xi = 0,001.$$

Метод Ньютона.

Функция должна быть дважды дифференцируема

$$f'(x) = 3x^2 - 3\cos x; \quad x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$$

$$f''(x) = 6x + 3\sin x;$$

$$F(x) \approx F(x_0) + F'(x_0)(x - x_0); \quad F(x) = f'(x)$$

$$1) \quad x = a - \frac{f'(a)}{f''(a)} \leftarrow \text{деление на 0}$$

$$\text{Возьмем точку } x = \frac{b+a}{2} = 0,5$$

$$x_1 = x - \frac{f'(x)}{f''(x)} = 0,5 - \frac{-1,88}{2,938} \approx 1,1399$$

$$|x - x_1| > \xi; \quad x = x_1.$$

$$2) \quad x_1 = 1,1399 - \frac{2,645}{9,565} \approx 0,8634$$

$$|x - x_1| > \xi; \quad x = x_1.$$

$$3) \quad x_1 = 0,8634 - \frac{0,2868}{7,46} \approx 0,8249$$

$$|x - x_1| > \xi; \quad x = x_1.$$

$$4) \quad x_1 = 0,8249 - \frac{0,0055}{7,1528} \approx 0,8241$$

$$|x - x_1| = 0,0008 > \xi$$

$$f(x) \approx -1,64213$$

Итоговая точка: $(0,8241; -1,64213)$

Рис. 4: Метод Ньютона. Первые пять операций.

Программа на Java:

```
1 package org.example.optimizationMethods;
2
3 public class NewtonMethod {
4     public static void main(String[] args) {
5         Point point = getExtremumPoint(0, 1, 0.001);
6         System.out.println(point.getX());
7         System.out.println(point.getY());
8     }
9
10    private static double getValueOfFunction(double x) {
11        return Math.pow(x, 3) - 3 * Math.sin(x);
12    }
13
14    private static double getDerivativeOfFunction(double x) {
```

```

15         return 3 * Math.pow(x, 2) - 3 * Math.cos(x);
16     }
17
18     private static double getDoubleDerivativeOfFunction(double x) {
19         return 6 * x + 3 * Math.sin(x);
20     }
21
22
23     private static Point getExtremumPoint(double leftSegmentEnd,
24     double rightSegmentEnd, double epsilon) {
25         double endPointDifference;
26         Point answerPoint = new Point();
27         double x = (leftSegmentEnd + rightSegmentEnd) / 2;
28         double x1 = x - (getDerivativeOfFunction(x) /
29         getDoubleDerivativeOfFunction(x));
30         endPointDifference = Math.abs(x - x1);
31         while (endPointDifference > 2 * epsilon) {
32             x = x1;
33             x1 = x - (getDerivativeOfFunction(x) /
34             getDoubleDerivativeOfFunction(x));
35             endPointDifference = Math.abs(x - x1);
36         }
37         double resultY = getValueOfFunction(x1);
38         answerPoint.setX(x1);
39         answerPoint.setY(resultY);
40         return answerPoint;
41     }
42 }

```

Вывод программы:

0.8241323124099124

-1.6421304129142102

Итоговая таблица

Вычислительный метод	X	Y
Метод половинного деления	0.82438	-1.64213
Метод золотого сечения	0.82395	-1.64213
Метод хорд	0.83542	-1.64167
Метод Ньютона	0.82413	-1.64213

Таблица 1: Полученные значения программой разными способами.