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 P3230

Пономаренко Алиса Валерьевна

Преподаватель:

Селина Елена Георгиевна

Санкт-Петербург

2024

Условие задачи

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

Исследуем функцию на экстремумы с помощью первой производной.

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

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

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

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.

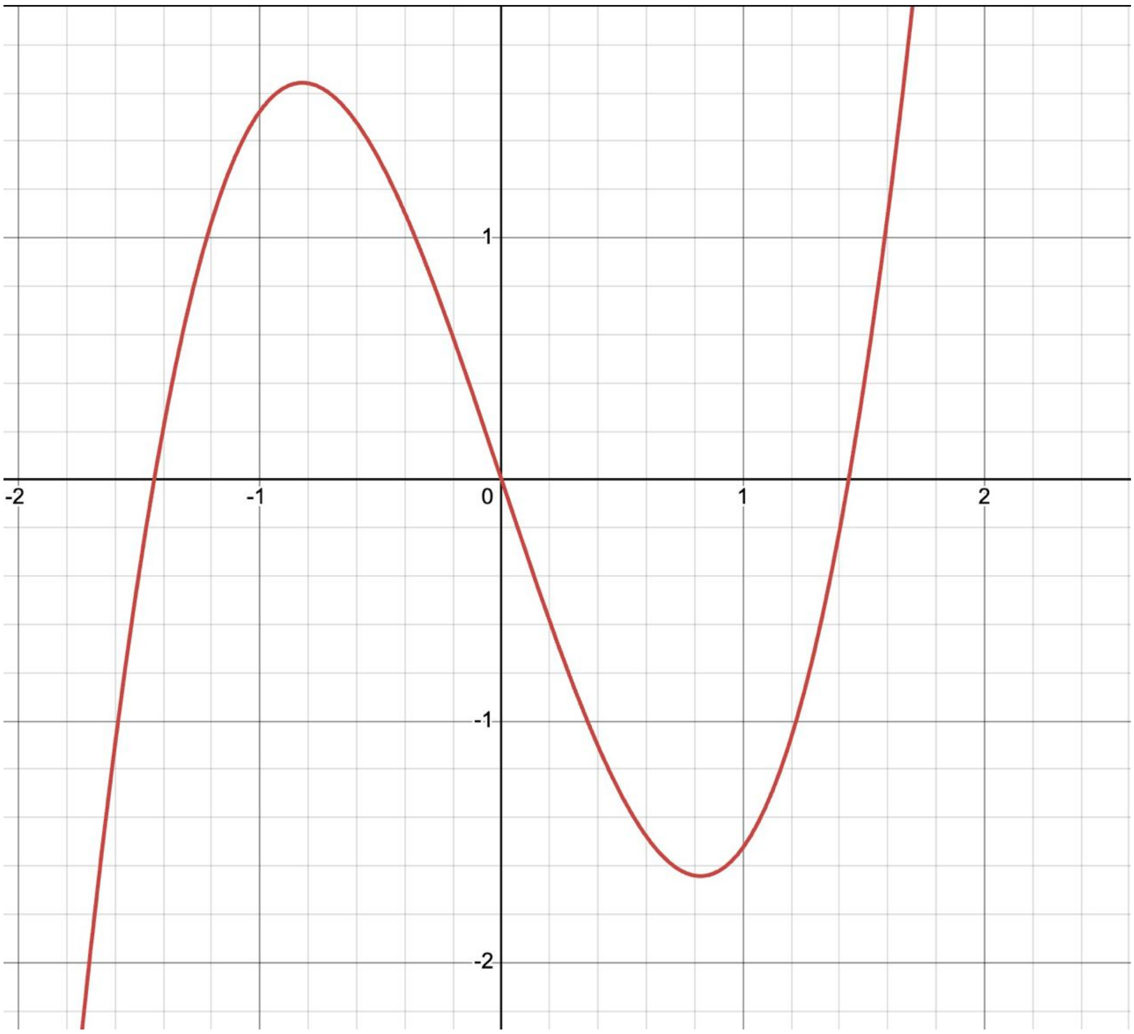


Рис. 1: График исходной функции

Метод половинного деления

Ручная проверка:

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

Метод половинного деления:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Программа на Java:

```java
package org.example.optimizationMethods;

public class Point {
    private double x;
    private double y;

    public Point() {
    }

    public Point(double x, double y) {
        this.x = x;
        this.y = y;
    }

    public double getX() {
        return x;
    }

    public void setX(double x) {
        this.x = x;
    }

    public double getY() {
        return y;
    }

    public void setY(double y) {
        this.y = y;
    }
}


package org.example.optimizationMethods;

public class DoubleDevMethod {
public static void main(String[] args) {
    Point point = getExtremumPoint(0, 1, 0.001);
    System.out.println(point.getX());
    System.out.println(point.getY());
  }

private static double getValueOfFunction(double x) {
    return Math.pow(x, 3) - 3 * Math.sin(x);
  }

```

```
private static Point getExtremumPoint(double leftSegmentEnd, //
    double rightSegmentEnd, double epsilon) {
    double endPointDifference;
    Point answerPoint = new Point();
    double x1 = (leftSegmentEnd + rightSegmentEnd - epsilon) / 2;
    double x2 = (leftSegmentEnd + rightSegmentEnd + epsilon) / 2;
    double y1 = getValueOfFunction(x1);
    double y2 = getValueOfFunction(x2);

    if (y2 < y1) {
      leftSegmentEnd = x1;
      endPointDifference = rightSegmentEnd - leftSegmentEnd;
    } else {
      rightSegmentEnd = x2;
      endPointDifference = rightSegmentEnd - leftSegmentEnd;
    }
    while (endPointDifference > 2 * epsilon) {
      x1 = (leftSegmentEnd + rightSegmentEnd - epsilon) / 2;
      x2 = (leftSegmentEnd + rightSegmentEnd + epsilon) / 2;
      y1 = getValueOfFunction(x1);
      y2 = getValueOfFunction(x2);
      if (y2 < y1) {
        leftSegmentEnd = x1;
        endPointDifference = rightSegmentEnd - leftSegmentEnd;
      } else {
        rightSegmentEnd = x2;
        endPointDifference = rightSegmentEnd - leftSegmentEnd;
      }
    }
    double resultX = (leftSegmentEnd + rightSegmentEnd) / 2;
    double resultY = getValueOfFunction(resultX);
    answerPoint.setX(resultX);
    answerPoint.setY(resultY);
    return answerPoint;
  }
}
```

Вывод программы:

0.82438232421875

-1.6421301895385456

Метод золотого сечения

Ручная проверка:

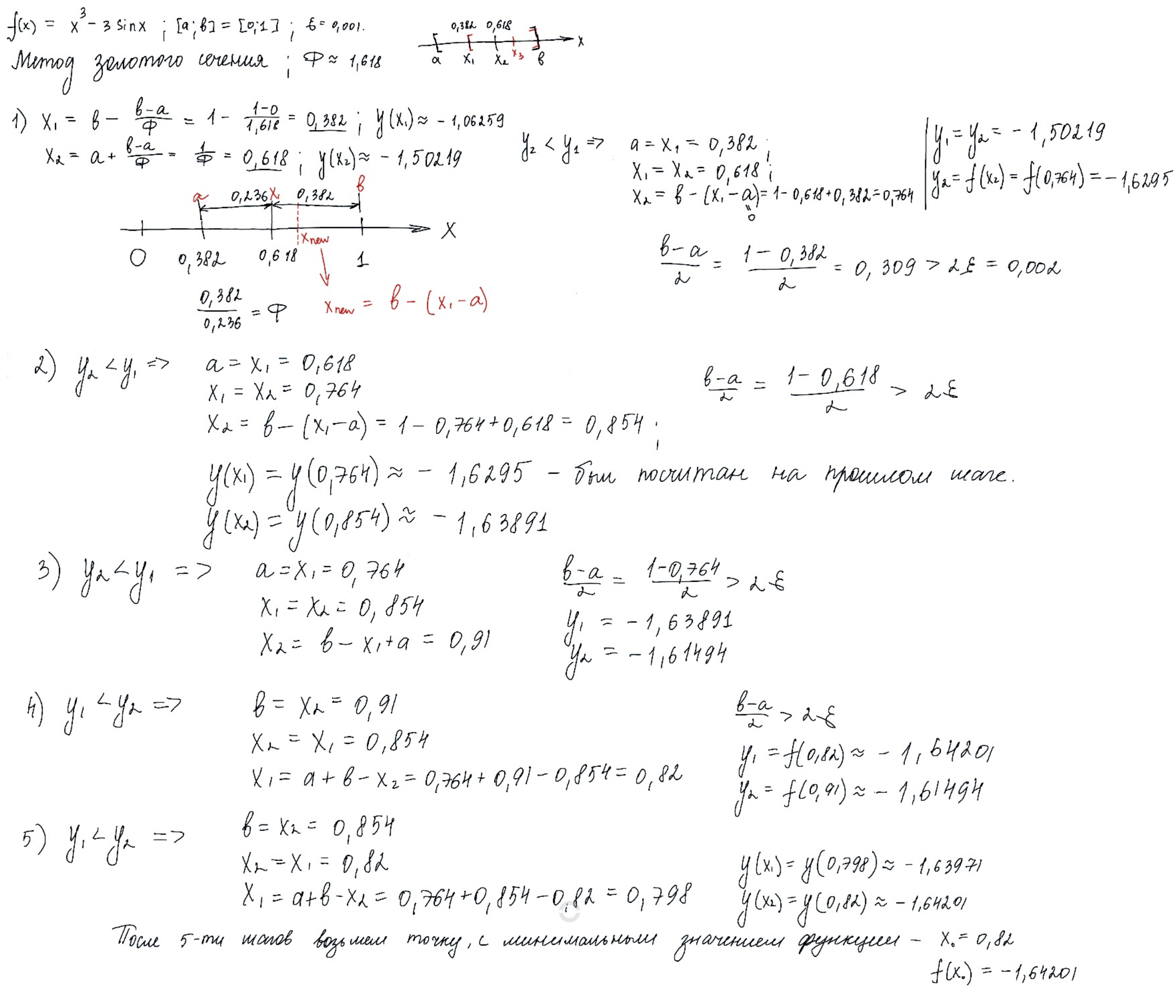


Рис. 2: Метод золотого сечения. Первые пять операций.

Программа на Java:

```java
package org.example.optimizationMethods;

public class GoldSectionMethod {
    public static void main(String[] args) {
        Point point = getExtremumPoint(0, 1, 0.001);
        System.out.println(point.getX());
        System.out.println(point.getY());
    }

    private static double getValueOfFunction(double x) {
        return Math.pow(x, 3) - 3 * Math.sin(x);
    }

```

```java
    private static Point getExtremumPoint(double leftSegmentEnd, double rightSegmentEnd, double epsilon) {
        Point answerPoint = new Point();
        double goldSection = 1.61803398875;
        double x1 = rightSegmentEnd - (rightSegmentEnd - leftSegmentEnd) / goldSection;
        double x2 = leftSegmentEnd + (rightSegmentEnd - leftSegmentEnd) / goldSection;
        double y1 = getValueOfFunction(x1);
        double y2 = getValueOfFunction(x2);
        if (y1 < y2) {
            rightSegmentEnd = x2;
            x2 = x1;
            x1 = leftSegmentEnd + rightSegmentEnd - x2;
        } else {
            leftSegmentEnd = x1;
            x1 = x2;
            x2 = rightSegmentEnd - x1 + leftSegmentEnd;
        }
        double difference = rightSegmentEnd - leftSegmentEnd;
        while (difference > 2 * epsilon) {
            if (y1 < y2) {
                rightSegmentEnd = x2;
                x2 = x1;
                x1 = leftSegmentEnd + rightSegmentEnd - x2;
                y1 = getValueOfFunction(x1);
                y2 = getValueOfFunction(x2);
                difference = rightSegmentEnd - leftSegmentEnd;
            } else {
                leftSegmentEnd = x1;
                x1 = x2;
                x2 = rightSegmentEnd - x1 + leftSegmentEnd;
                y1 = getValueOfFunction(x1);
                y2 = getValueOfFunction(x2);
                difference = rightSegmentEnd - leftSegmentEnd;
            }
        }
        if (y1 < y2) {
            answerPoint.setX(x1);
            answerPoint.setY(y1);
        } else {
            answerPoint.setX(x2);
            answerPoint.setY(y2);
        }
        return answerPoint;
    }
}
```

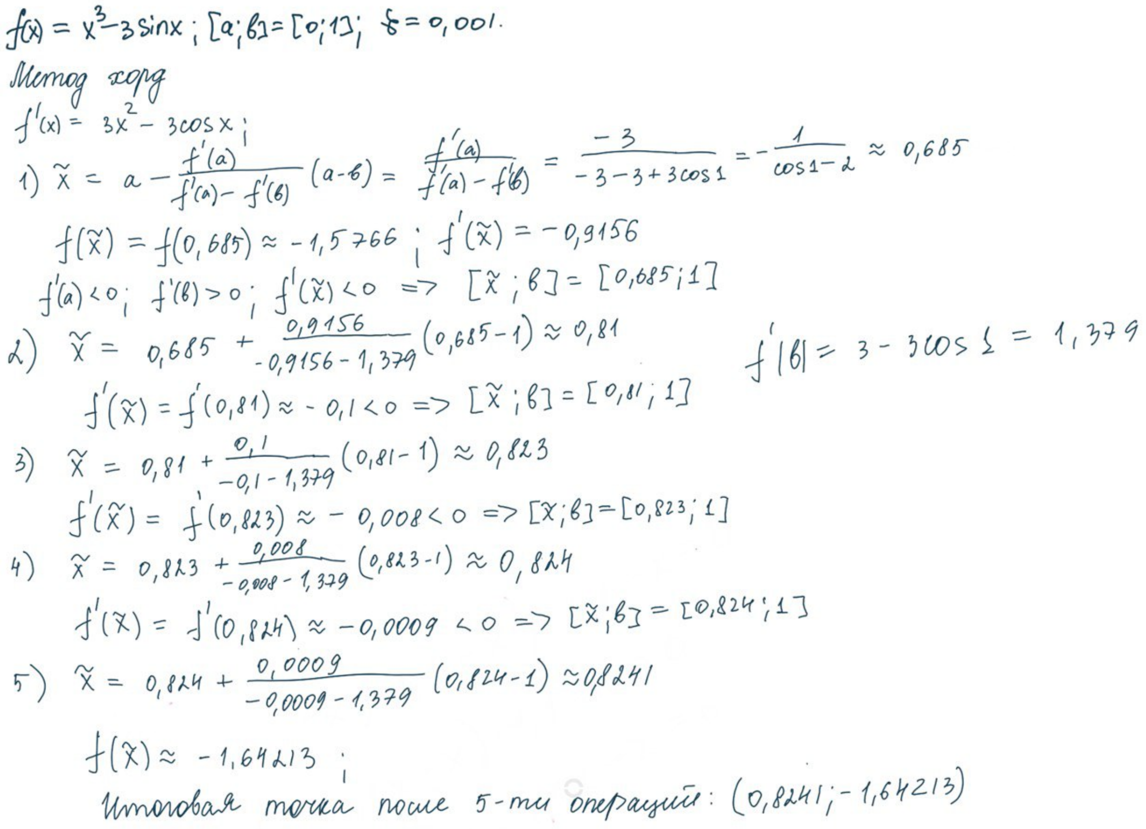
Вывод программы:

0.8239519738271477

-1.6421302967101874

Метод хорд

Ручная проверка:



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

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) $\tilde{x} = 0{,}685 + \frac{0{,}9156}{-0{,}9156 - 1{,}379}(0{,}685 - 1) \approx 0{,}81$ $\quad f'(b) = 3 - 3\cos 1 = 1{,}379$

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

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:

```java
package org.example.optimizationMethods;

public class SegmentationMethod {
    public static void main(String[] args) {
        Point point = getExtremumPoint(0, 1, 0.001);
        System.out.println(point.getX());
        System.out.println(point.getY());
    }

    private static double getValueOfFunction(double x) {
```

```java
        return Math.pow(x, 3) - 3 * Math.sin(x);
    }

    private static double getDerivativeOfFunction(double x) {
        return 3 * Math.pow(x, 2) - 3 * Math.cos(x);
    }

    private static Point getExtremumPoint(double leftSegmentEnd, double rightSegmentEnd, double epsilon) {
        double endPointDifference;
        Point answerPoint = new Point();
        double x = leftSegmentEnd - ((leftSegmentEnd - rightSegmentEnd) * getDerivativeOfFunction(leftSegmentEnd) / (getDerivativeOfFunction(leftSegmentEnd) - getDerivativeOfFunction(rightSegmentEnd)));
        double derX = getDerivativeOfFunction(x);
        double derA = getDerivativeOfFunction(leftSegmentEnd);
        if (derA * derX < 0) {
            leftSegmentEnd = x;
            endPointDifference = rightSegmentEnd - leftSegmentEnd;
        } else {
            rightSegmentEnd = x;
            endPointDifference = rightSegmentEnd - leftSegmentEnd;
        }
        while (endPointDifference > 2 * epsilon) {
            x = leftSegmentEnd - ((leftSegmentEnd - rightSegmentEnd) * getDerivativeOfFunction(leftSegmentEnd) / (getDerivativeOfFunction(leftSegmentEnd) - getDerivativeOfFunction(rightSegmentEnd)));
            derX = getDerivativeOfFunction(x);
            derA = getDerivativeOfFunction(leftSegmentEnd);
            if (derA * derX < 0) {
                leftSegmentEnd = x;
                endPointDifference = rightSegmentEnd - leftSegmentEnd;
            } else {
                rightSegmentEnd = x;
                endPointDifference = rightSegmentEnd - leftSegmentEnd;
            }
        }
        double resultX = (leftSegmentEnd + rightSegmentEnd) / 2;
        double resultY = getValueOfFunction(resultX);
        answerPoint.setX(resultX);
        answerPoint.setY(resultY);
        return answerPoint;
    }
```

```
}
```

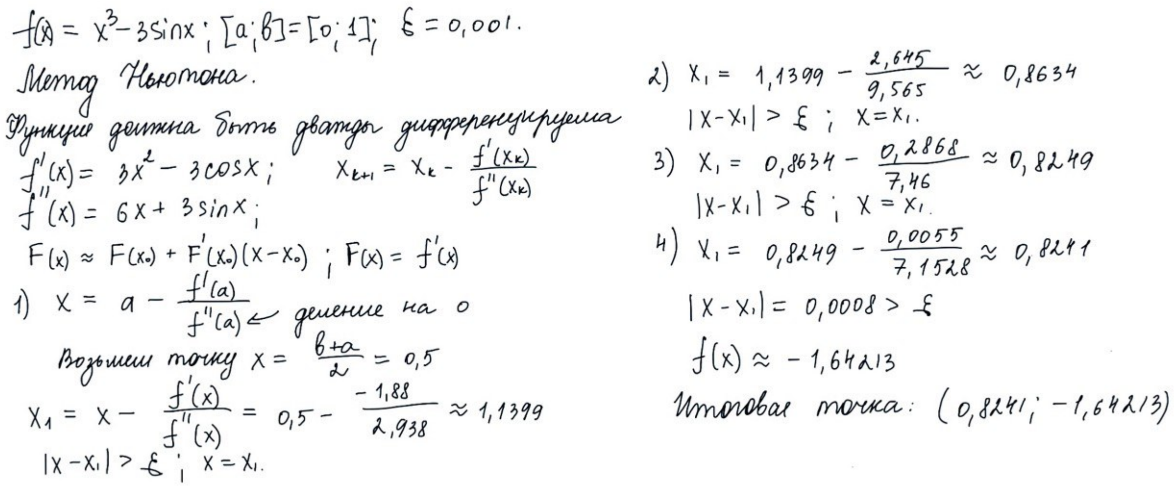
Вывод программы:

0.8354286327834163

-1.6416725013479152

Метод Ньютона

Ручная проверка:



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

Метод Ньютона.

Функция должна быть дважды дифференцируема

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

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

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

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

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

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

$|x - x_1| > \varepsilon$; $x = x_1$.

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

$|x - x_1| > \varepsilon$; $x = x_1$.

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

$|x - x_1| > \varepsilon$; $x = x_1$.

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

$|x - x_1| = 0{,}0008 > \varepsilon$

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

Итоговая точка: $(0{,}8241; -1{,}64213)$

Рис. 4: Метод Ньютона. Первые пять операций.

Программа на Java:

```
package org.example.optimizationMethods;

public class NewtonMethod {
    public static void main(String[] args) {
        Point point = getExtremumPoint(0, 1, 0.001);
        System.out.println(point.getX());
        System.out.println(point.getY());
    }

    private static double getValueOfFunction(double x) {
        return Math.pow(x, 3) - 3 * Math.sin(x);
    }

    private static double getDerivativeOfFunction(double x) {
```

```java
        return 3 * Math.pow(x, 2) - 3 * Math.cos(x);
    }

    private static double getDoubleDerivativeOfFunction(double x) {
        return 6 * x + 3 * Math.sin(x);
    }


    private static Point getExtremumPoint(double leftSegmentEnd,
   double rightSegmentEnd, double epsilon) {
        double endPointDifference;
        Point answerPoint = new Point();
        double x = (leftSegmentEnd + rightSegmentEnd) / 2;
        double x1 = x - (getDerivativeOfFunction(x) /
   getDoubleDerivativeOfFunction(x));
        endPointDifference = Math.abs(x - x1);
        while (endPointDifference > 2 * epsilon) {
            x = x1;
            x1 = x - (getDerivativeOfFunction(x) /
   getDoubleDerivativeOfFunction(x));
            endPointDifference = Math.abs(x - x1);
        }
        double resultY = getValueOfFunction(x1);
        answerPoint.setX(x1);
        answerPoint.setY(resultY);
        return answerPoint;
    }

}
```

Вывод программы:

0.8241323124099124

-1.6421304129142102

Итогова 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.