|  |  |
| --- | --- |
| |  | | --- | | МИНОБРНАУКИ РОССИИ  Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | |

Институт Информационных технологий

Кафедра МОСИТ

**ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №4**

по дисциплине «Теория принятия решений»

**Студент группы** ИКБО-11-17 Алиев Ю. А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись студента)*

**Руководитель работы** Петренко А. А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись руководителя)*

Москва, 2020

# Постановка задачи

Найти: *Argmin* *f*(*x*), *x* ∈ [-5; 5] при Δ*x* ≤ 0.1 перечисленными ниже методами в соответствии с вариантом.

1. Метод половинного деления (дихотомии)
2. Метод золотого сечения
3. Метод Фибоначчи

Вариант 1: *f*(*x*) = 2x2 - 5

# Метод половинного деления (дихотомии)

Для решения данным методом была написана следующая программа на языке программирования JavaScript:

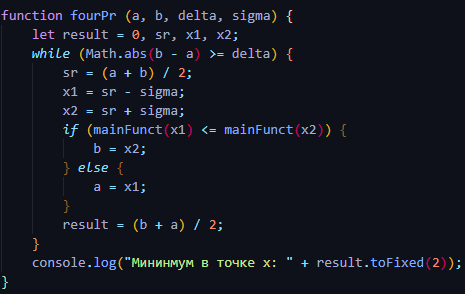


Рисунок 1. Алгоритм метода дихотомии

Алгоритм написан на основе лекций.

Для варианта №1 алгоритм выдает следующий ответ. В выводе предоставлены предыдущие два способа

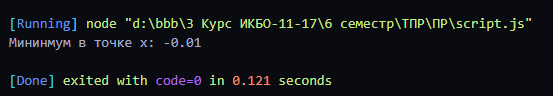


Рисунок 2. Ответ, найденный методом дихотомии

Ответ совпадает с решением из предыдущих номеров с погрешностью в 0.01.

# Метод золотого сечения

Метод золотого сечения также описан на языке программирования JavaScript.

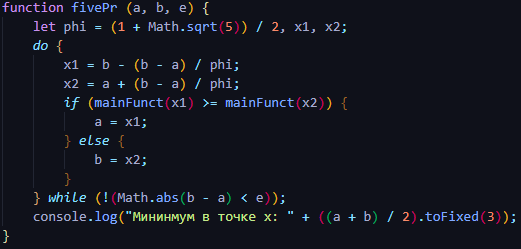


Рисунок 3. Алгоритм метода золотого сечения

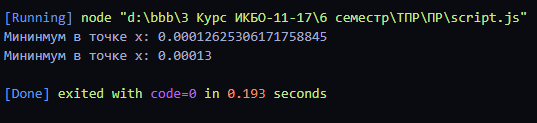


Рисунок 4. Ответ, найденный методом золотого сечения

Ответ совпадает с решением из предыдущих номеров.

# Метод Фибоначчи

Для данного метода также был написан алгоритм на языке JavaScript. Однако особенностью этого алгоритма является длительная компиляция из-за использования рекурсии.

Тем не менее, ответ найден верно.

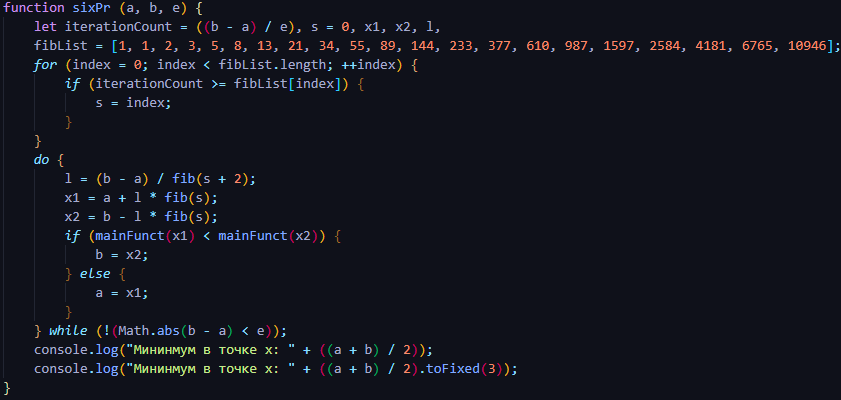


Рисунок 5. Алгоритм метода Фибоначчи

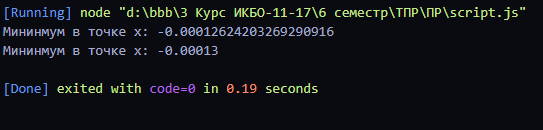


Рисунок 6. Ответ, найденный методом Фибоначчи

Как видно, ответ во всех трех случаях получен одинаковый c погрешностью в 0.00001.

# Приложение

Исходный код программы:

|  |
| --- |
| import java.text.DecimalFormat;  public class pr1 {  public static void main(String[] args) {  findMin\_2EX();  findMin\_3EX();  findMin\_4EX(-5, 5, 0.1, 0.04);  findMin\_5EX(-5, 5, 0.001);  findMin\_6EX(-5, 5, 0.001);  findMin\_7EX(0, 1, 0.001);  findMin\_8EX(-5, 5, 0.001);  findMin\_9EX(-5, 5, 0.1, 0.001);  findMin\_10EX(-5, 0.1, 0.1);  findMin\_11EX(-5, 5, 0.1, 1);  }  //Основная функция  public static double func\_result(double x) {  return 3 \* (x \* x) + x + 1;  }  //Первая производная от искомой функции  public static double deractive(double x) {  return 6 \* x + 1;  }  //Вторая производная от искомой функции  public static double deractive\_1(double x) {  return 6;  }  //Метод полного перебора  public static void findMin\_2EX() {  DecimalFormat df = new DecimalFormat("###.###");  double x = -5;  double min = 999;  double argmin = 0;  double range = 0.001;  while (x <= 5) {  double y = func\_result(x);  x = x + range;  if (y <= min) {  min = y;  argmin = x;  }  }  System.out.println("Минимум в точке x по второму способу составляет: " + df.format(argmin));  }  //Метод поразрядного поиска  public static void findMin\_3EX() {  DecimalFormat df = new DecimalFormat("###.###");  double x = -5;  double argmin = 0;  double min = 999;  double range = 0.3;  double newRange = 0.001;  double newX = 0;  while (x <= 5) {  double y = func\_result(x);  x = x + range;  double nextY = func\_result(x + range);  if (nextY > y) {  newX = x;  x = -5;  break;  }  }  for (double i = x; i <= newX; i += newRange) {  if (func\_result(i) <= min) {  min = func\_result(i);  argmin = i;  }  }  System.out.println("Минимум в точке x по третьему способу составляет: " + df.format(argmin));  }  //Метод дихотомии  public static void findMin\_4EX(double a, double b, double delta, double sigma) {  DecimalFormat df = new DecimalFormat("###.###");  double result = 0;  while (Math.abs(b - a) >= delta) {  double sr = (a + b) / 2;  double x1 = sr - sigma;  double x2 = sr + sigma;  if (func\_result(x1) <= func\_result(x2)) {  b = x2;  } else {  a = x1;  }  result = (b + a) / 2;  }  System.out.println("Минимум в точке x по четвертому способу составляет: " + df.format(result));  }  //Метод золотого сечения  public static void findMin\_5EX(double a, double b, double e) {  DecimalFormat df = new DecimalFormat("###.###");  double PHI = (1 + Math.sqrt(5)) / 2;  double x1, x2;  do {  x1 = b - (b - a) / PHI;  x2 = a + (b - a) / PHI;  if (func\_result(x1) >= func\_result(x2)) {  a = x1;  } else {  b = x2;  }  } while (!(Math.abs(b - a) < e));  System.out.println("Минимум в точке x по пятому способу составляет: " + df.format((a + b) / 2));  }  //Числа Фибоначчи  public static int fib(int i) {  if (i == 1) return 1;  if (i == 2) return 1;  return fib(i - 1) + fib(i - 2);  }  //Метод Фибоначчи  public static void findMin\_6EX(double a, double b, double e) {  DecimalFormat df = new DecimalFormat("###.###");  int iterationCount = (int) ((b - a) / e);  int S = 0;  int[] fibList = {1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946};  for (int index = 0; index < fibList.length; ++index) {  if (iterationCount >= fibList[index]) {  S = index;  }  }  double x1, x2, l;  do {  l = (b - a) / fib(S + 2);  x1 = a + l \* fib(S);  x2 = b - l \* fib(S);  if (func\_result(x1) < func\_result(x2)) {  b = x2;  } else {  a = x1;  }  } while (!(Math.abs(b - a) < e));  System.out.println("Минимум в точке x по шестому способу составляет: " + df.format((a + b) / 2));  }  //Метод парабол  public static void findMin\_7EX(double a, double b, double h) {  DecimalFormat df = new DecimalFormat("###.###");  if (a == 0) a += 0.1;  while ((func\_result(a + h) - 2 \* func\_result(a) + func\_result(a - h)) / (h \* h) <= 0) {  a += 0.1;  }  double x1;  x1 = a - 0.5 \* h \* (func\_result(a + h) - func\_result(a - h)) / (func\_result(a + h) - 2 \*  func\_result(a) + func\_result(a - h));  while (Math.abs(x1 - a) > b) {  a = x1;  x1 = a - 0.5 \* h \* (func\_result(a + h) - func\_result(a - h)) / (func\_result(a + h) - 2 \*  func\_result(a) + func\_result(a - h));  }  System.out.println("Минимум в точке x по седьмому способу составляет: " + df.format(x1));  }  //Метод средней точки  public static void findMin\_8EX(double a, double b, double e) {  DecimalFormat df = new DecimalFormat("###.###");  double x = (a + b) / 2;  while (Math.abs(deractive(x)) > e) {  x = (a + b) / 2;  if (deractive(x) > 0) {  b = x;  } else {  a = x;  }  }  x = (a + b) / 2;  System.out.println("Минимум в точке x по восьмому способу составляет: " + df.format(x));  }  //Метод хорд  public static void findMin\_9EX(double a, double b, double e, double accuracy) {  DecimalFormat df = new DecimalFormat("###.###");  double x = a - (deractive(a) / (deractive(a) - deractive(b))) \* (a - b);  double y = deractive(x);  if (Math.abs(y) < accuracy) {  //TODO  } else {  if (y > 0.0) {  findMin\_9EX(a, x, e, accuracy);  } else {  findMin\_9EX(x, b, e, accuracy);  }  }  System.out.println("Минимум в точке x по девятому способу составляет: " + df.format(x));  }  //Метод Ньютона №1  public static double r\_method(double x, double prev, double accuracy) {  if (Math.abs(x - prev) > accuracy) {  return r\_method(x - deractive(x) / deractive\_1(x), x, accuracy);  } else {  return x;  }  }  //Метод Ньютона №2  public static void findMin\_10EX(double x, double accuracy, double xprev) {  DecimalFormat df = new DecimalFormat("###.###");  r\_method(x, xprev, accuracy);  System.out.println("Минимум в точке x по десятому способу составляет: " + df.format(r\_method(x, xprev, accuracy)));  }  //Метод ломанных  public static void findMin\_11EX(double a, double b, double e, double iteration) {  DecimalFormat df = new DecimalFormat("###.###");  double L = get\_max(a, b);  double x0, p0, d, x1, x2, ld;  while (true) {  x0 = (1 / (2 \* L)) \* (func\_result(a) - func\_result(b) + L \* a + L \* b);  p0 = 0.5 \* (func\_result(a) + func\_result(b) + L \* a - L \* b);  d = ((1 / (2 \* L)) \* (func\_result(x0) - p0));  x1 = x0 - d;  x2 = x0 + d;  ld = 2 \* L \* d;  if (ld > e) {  if (Math.abs(deractive(x1)) < Math.abs(deractive(x2))) {  b = x0;  } else {  a = x0;  }  iteration += 0.1;  } else {  break;  }  }  double final\_x = x0;  System.out.println("Минимум в точке x по одиннадцатому способу составляет: " + df.format(final\_x));  }  //Поиск максимума на концах отрезка  public static double get\_max(double a, double b) {  if (Math.abs(deractive(a)) > Math.abs(deractive(b))) {  return Math.abs(deractive(a));  } else {  return Math.abs(deractive(b));  }  }  } |

Листинг 1. Исходный код файла pr1