МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «Стратегічного управління»

Звіт з лабораторної роботи №2

з дисципліни «Дослідження операцій»

Виконав:

Студент групи КН-27

Перевірив:

Лисицький В. Л.

Харків – 2019

**Ціль роботи:** Розв’язати задачу мінімізації методом циклічного покоординатного спуску

**Задача:**

Задана функція:

Характер екстремуму: мінімум.

Початкова точка: (-1, 3)

Точність = 0,001

Межі одновимірних функцій: a = -10, b = 10

**Алгоритм досліджуваного метода:**

Для зупинки алгоритму використовується критерій |f(𝑥k)-f(xk+1)|<=ε, хоча можна застосовувати і будь-який інший критерій.

**Початковий етап.**

1. Инициализация некоторым значением x_0 \in \mathbb{R}^n

2. повторювати:

* + для i=1\dots n
    1. 1. фіксуємо значення всіх змінних крімx_i, отримуючи одновимірну функцію f(x_i)
    2. проводимо одновимірну оптимізацію по змінної x_i, будь-яким методом одновимірної оптимізації (в даній лабораторній роботі використовується метод золотого перетину)
    3. 3. якщо виконано критерій зупинки (варіанти описані нижче), то повертаємо поточне значення x=(x_1,\dots,x_n)

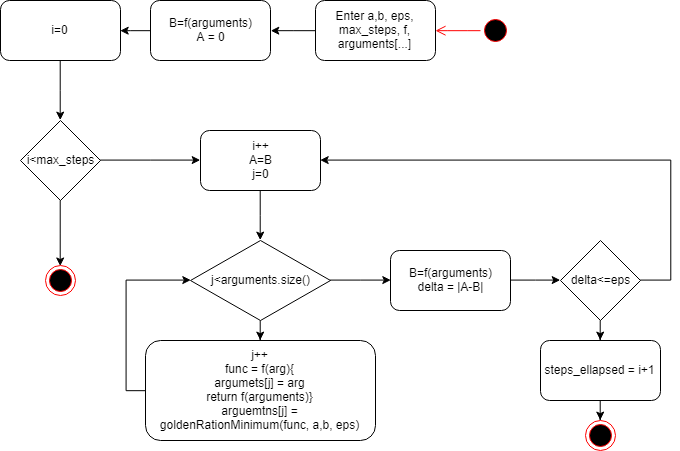


Рисунок 1 – Блок схема алгоритму циклічного по координатного спуску

**Код розробленої програми:**

|  |
| --- |
| function testFunction(arguments){  return Math.pow(arguments[0],3) + Math.pow(arguments[1],3) + 3 \* arguments[0] \* arguments[1]  }  function draw(data, x, y){  Plotly.newPlot('chart', [{  z: data,  type: 'surface',  x: x,  y: y,  }]);  }  document.addEventListener("DOMContentLoaded", () => {  document.getElementById("run").addEventListener("click", () =>{  document.getElementById("answer").innerHTML = "";  //eq = document.getElementById("equation").value;  var start = parseFloat(document.getElementById("start").value);  var end = parseFloat(document.getElementById("end").value);  var eps = parseFloat(document.getElementById("epsilon").value);  var arguments = []  arguments.push(-1)  arguments.push(3)    var graph\_arr = []    var x = []  var y = []  for (let i = start; i < end; i+=1){  x.push(i)  y.push(i)  var arr = []  for (let j = start; j < end; j+=1){  arr.push(testFunction([j, i]))  }  graph\_arr.push(arr)  }  draw(graph\_arr, x ,y)    var A = 0.0  var B = testFunction(arguments)  var steps\_ellapsed = 0  var max\_steps\_count = 100  var delta = 0.0    var n = arguments.length;    function testForStop(arr1, arr2){  function getNorm(arr){  var square\_sum = 0  for (var i = 0; i < arr.length; i++){  square\_sum += arr[i] \* arr[i]  }  return Math.sqrt(square\_sum)  }  function substract\_vectors(arr1, arr2){  var result\_arr = []  for (var i = 0; i < arr1.length; i++){  result\_arr.push(arr2[i]-arr1[i])  }  return result\_arr  }  if (getNorm(substract\_vectors(arr2, arr1)) <= eps)  return true  return false    }  for (var i = 0; i < max\_steps\_count; i++){  A = B  for (var j = 0; j < n; j++){  var func = function(arg, arguments){  arguments[j] = arg  return testFunction(arguments)  }  arguments[j] = goldenRatioMinimum(func, start, end, eps, arguments)  }  B = testFunction(arguments)  delta = Math.abs(A-B)  if (delta <= eps){  steps\_ellapsed = i+1  break  }  }    document.getElementById("answer").innerHTML += "Понадобилось " + steps\_ellapsed + " итераций<br>";  document.getElementById("answer").innerHTML += arguments;  document.getElementById("answer").innerHTML += "<br>";  document.getElementById("answer").innerHTML += "Function(args) = " + testFunction(arguments);    });  document.getElementById("clean").addEventListener("click", () =>{  document.getElementById("answer").innerHTML = "";  document.getElementById("chart").innerHTML = "";  });    });  function goldenRatioMinimum(fn, a, b, ε, arguments) {  ε = ε || 1e-8;  var φ = (-1 + Math.sqrt(5)) / 2;  var x1 = a + (1 - φ) \* (b - a);  var x2 = b - (1 - φ) \* (b - a);  var A = fn(x1, arguments);  var B = fn(x2, arguments);  while(b - a > ε) {  if(A < B) {  b = x2;  x2 = x1;  B = A;  x1 = a + (1 - φ) \* (b - a);  A = fn(x1, arguments);  } else {  a = x1;  x1 = x2;  A = B;  x2 = b - (1 - φ) \* (b - a);  B = fn(x2, arguments);  }  }  var x = (a + b) / 2;  x = Math.abs(x) < 1e-6 ? 0 : x;  return x  } |

**Результати роботи програми:**

Результати роботи програми показані на рисунку 2.

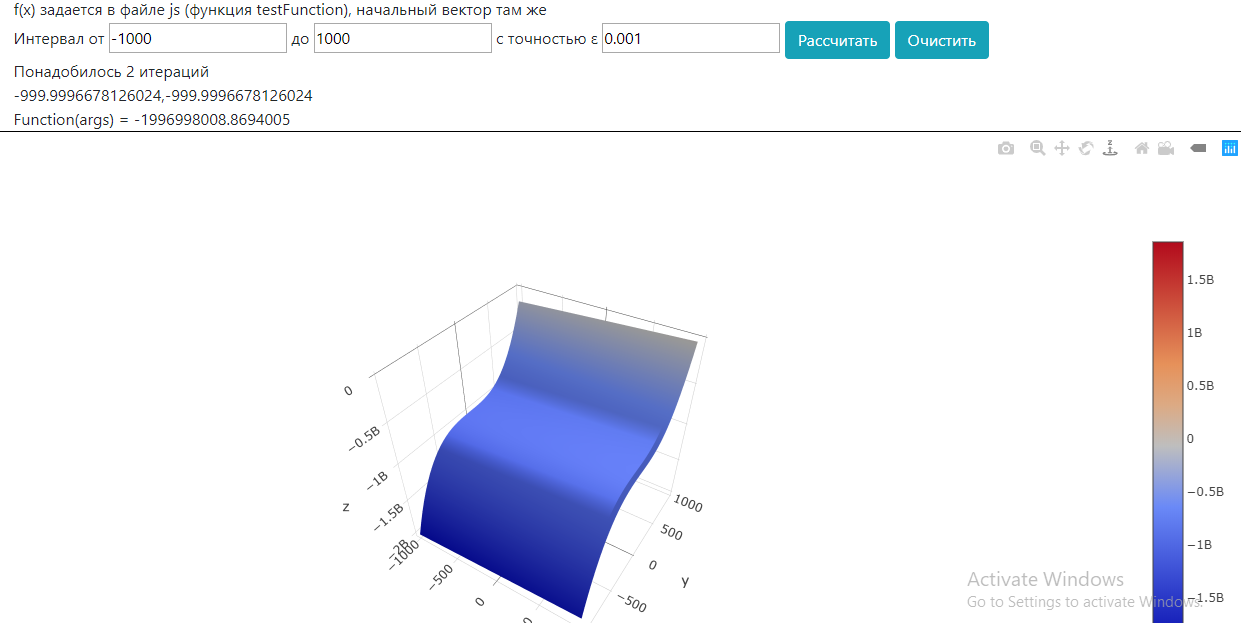


Рисунок 2 – Інтерфейс програми

**Висновки:**

У цій лабі був розглянутий простий метод циклічного по координатного спуску для пошуку мінімуму функції багатьох змінних. Для пошуку мінімуму одномірної функції був використаний метод золотого поділу.