Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Пермский государственный аграрно-технологический университет

имени академика Д.Н. Прянишникова»

Кафедра информационных технологий и программной инженерии

**ЛАБОРАТОРНАЯ РАБОТА № 1**

Градиентный спуск.

|  |  |
| --- | --- |
| Выполнил: | группа ПИНб-4  П.С. Плотников |
| Проверил: | доцент каф. ИТиПИ,  Т.А. Казаченко |

Пермь-2024 г.

**Формулировка задачи**

1) Запрограммируйте в среде RStudio метод градиентного спуска для заданной функции. Функция:

2) Графически отобразите линии уровня выбранной функции.

3) Выведите значение экстремума и число итераций, за который он достигнут.

4) Сравните полученное значение экстремума с экстремумом для данной функции в Excel.

**Программный код**

library("ggplot2")

# Определение функции

f <- function(x1, x2) {

85 \* x1^2 +168 \* x1 \* x2 + 85 \* x2^2 + 29 \* x1 - 51 \* x2 + 83

}

# Создание сетки значений x1 и x2

x1 <- seq(-50, 10, length.out = 100)

x2 <- seq(-10, 50, length.out = 100)

grid <- expand.grid(x1 = x1, x2 = x2)

# Вычислние значений z

grid$z <- with(grid, f(x1, x2))

# Построение графика линий уровня

ggplot(grid, aes(x = x1, y = x2, z = z)) +

geom\_contour(aes(color = after\_stat(level)), bins = 20) +

scale\_color\_viridis\_c() +

labs(title = "Линии уровня функции",

x = expression(x[1]),

y = expression(x[2]),

color = "Уровень") +

theme\_minimal()

# Градиент

gradient <- function(x1, x2) {

# Производная по x1

df\_x1 <- 170 \* x1 + 168 \* x2 + 29

# Производная по x2

df\_x2 <- 168 \* x1 + 170 \* x2 - 51

c(df\_x1, df\_x2)

}

# Поиск точки экстремума

gradient\_descent <- function(start, rate = 0.001, tol = 1e-6, max\_steps = 6000) {

# Начальные значения

x1 <- start[1]

x2 <- start[2]

iter <- 0

repeat {

# Счетчик проделанных шагов

iter <- iter + 1

# Получение градиента

grad <- gradient(x1, x2)

# Получение следующих точек

x1\_new <- x1 - rate \* grad[1]

x2\_new <- x2 - rate \* grad[2]

# Проверка условия сходимости

if (sqrt((x1\_new - x1)^2 + (x2\_new - x2)^2) < tol || iter >= max\_steps) {

break

}

# Присваивание новых значений

x1 <- x1\_new

x2 <- x2\_new

}

# Список с точкой экстремума, его значением и кол-вом шагов

list(

minimum\_x1 = round(x1, 2),

minimum\_x2 = round(x2, 2),

extremum = round(f(x1, x2), 2),

steps = iter

)

}

# Стартовая точка

start\_point <- c(0, 0)

result <- gradient\_descent(start\_point)

result

**Линии уровня**

В ходе выполнения написанного кода, были получены линии уровня, показанные на рисунке 1.

Изображение выглядит как текст, линия, диаграмма, рукописный текст

Автоматически созданное описание

Рисунок 1 – Линии уровня функции

**Вычисленный экстремум, количество итераций**

Для заданной функции был получен экстремум в точке , значение функции в этой точке: . Количество шагов: 5468. Большое количество шагов обусловлено тем, что функция имеет «овражный» вид и градиентный метод на ней работает плохо. Вывод результатов в консоли показан на рисунке 2.

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

Рисунок 2 – Точка экстремума, значение экстремума и количество итераций

**Сравнение с Excel**

На рисунке 3 показано значение экстремума, полученное в Excel. Для получения значений была использована функция «Поиск решений».

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Рисунок 3 – Результат поиска экстремума в Excel

Сравнивая результаты работы программы с результатами в Excel, можно сделать вывод, что написанный код программы работает корректно.