Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №2

По дисциплине «Теоретические интеллектуальные и информационные технологии»

Тема: «Итерационные методы поиска экстремумов»

Выполнил:

Студент 1 курса

Группы ИИ-21(1)

Кирилович А.А.

Проверил:

Анфилец С. В.

Брест 2022

**Цель:** научиться находить минимум/максимум функции методом градиентного спуска.

**Ход работы:**

import random

import numpy as np

global\_epsilon = 0.000000001

epsilon = 0.0001

def function(x, y, z):

    return x \* y + (z \*\* 2) \* np.sin(x) + (2 \* (x \*\* 2) + 3) \* (y \*\* 2)

def derivative\_x(x, y, z):

    return (function(x + global\_epsilon, y, z) -

            function(x, y, z)) / global\_epsilon

def derivative\_y(x, y, z):

    return (function(x, y + global\_epsilon, z) -

            function(x, y, z)) / global\_epsilon

def derivative\_z(x, y, z):

    return (function(x, y + global\_epsilon, z) -

            function(x, y, z)) / global\_epsilon

def abs\_gradient(x, y, z):

    return (derivative\_x(x, y, z) \*\* 2 + derivative\_y(x, y, z) \*\* 2 + derivative\_z(x, y, z) \*\* 2) \*\* 0.5

if \_\_name\_\_=="\_\_main\_\_":

    xmin, xmax = map(int, input("Введите xmin, xmax: ").split(' '))

    ymin, ymax = map(int, input("Введите ymin, ymax: ").split(' '))

    zmin, zmax = map(int, input("Введите zmin, zmax: ").split(' '))

    print("Введите 1 для поиска максимума,\n0 - для поиска минимума: ")

    choice=int(input())

    x = random.uniform(xmin,xmax)

    y = random.uniform(ymin,ymax)

    z = random.uniform(zmax,zmin)

    iter = 100

    alf = 0.1

    if(choice == 1):

        for i in range(iter):

            if (x < xmin or x > xmax or y < ymin or \

                 y > ymax or z < zmin or z > zmax):

                break

            if(abs\_gradient(x, y, z) < epsilon):

                break

            x\_next = x + alf \* derivative\_x(x, y, z)

            y\_next = y + alf \* derivative\_y(x, y, z)

            z\_next = z + alf \* derivative\_z(x, y, z)

            if (x\_next < xmin or x\_next > xmax or y\_next < ymin or \

                 y\_next > ymax or z\_next < zmin or z\_next > zmax):

                break

            if(abs\_gradient(x\_next, y\_next, z\_next) < epsilon):

                break

            x = x\_next

            y = y\_next

            z = z\_next

        print(f"Максимум: x = {x}, y = {y}, z = {z}; function = {function(x, y, z)}")

    elif(choice == 0):

        for i in range(iter):

            if (x < xmin or x > xmax or y < ymin or \

                 y > ymax or z < zmin or z > zmax):

                break

            if(abs\_gradient(x, y, z) < epsilon):

                break

            x\_next = x - alf \* derivative\_x(x, y, z)

            y\_next = y - alf \* derivative\_y(x, y, z)

            z\_next = z - alf \* derivative\_z(x, y, z)

            if (x\_next < xmin or x\_next > xmax or y\_next < ymin or \

                 y\_next > ymax or z\_next < zmin or z\_next > zmax):

                break

            if(abs\_gradient(x\_next, y\_next, z\_next) < epsilon):

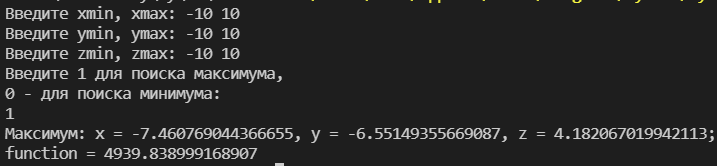
                break

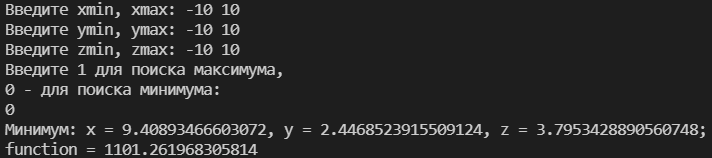
            x = x\_next

            y = y\_next

            z = z\_next

        print(f"Минимум: x = {x}, y = {y}, z = {z}; function = {function(x, y, z)}")





**Вывод:** с помощью метода градиентного спуска можно довольно точно находить локальные максимумы и минимумы функции.