|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Информатики и систем управления

КАФЕДРА Теоретической информатики и компьютерных технологий

**Лабораторная работа № 6**

«Метод наискорейшего спуска поиска минимума функции многих переменных»

по курсу «Численные методы»

Выполнила:

студент группы ИУ9-61Б

Яровикова Анастасия

Проверила:

Домрачева А. Б.

Москва, 2023

1. **Цель**

Целью данной работы является изучение метода наискорейшего спуска для поиска минимума функции многих переменных и сравнение полученного результата со значением минимума функции, найденным аналитически.

**Постановка задачи**

**Дано:** функция многих переменных и точка ;

**Задание:**

* Найти минимум функции двух переменных с точностью , начиная итерации из точки ;
* Найти минимум аналитичности;
* Сравнить полученные результаты.

**Индивидуальный вариант:**

, .

1. **Основные теоретические сведения**

Метод наискорейшего спуска является итерационным.  
Пусть для заданной функции на -том шаге имеется некоторое приближение к минимуму .

Рассмотрим функцию одной переменной :

где вектор – градиент функции в точке .

Функция представляет собой ограничение функции на прямую градиентного спуска, проходящую через точку -го приближения .

Для следующего приближения к точке минимума полагаем

где точка – это минимум функции .

Процесс поиска минимума продолжается до тех пор, пока не станет меньше допустимой погрешности ε.

В двумерном случае итерация имеет следующий вид:

где ;

где все производные берутся в точке .

1. **Реализация**

Листинг 1. Метод наискорейшего спуска поиска минимума функции многих переменных

from sympy import \*

import math

eps = 0.001

x, y = symbols('x y')

def f():

return 3\*x\*\*2 - 3\*x\*y - 4\*y\*\*2 - 2\*x + y

def analytical\_min():

return 1/3, 0.0

print('f: ', f())

fx = diff(f(), x)

fy = diff(f(), y)

print('df/dx: ', fx)

print('df/dy: ', fy)

print('d^2f/dx^2: ',diff(fx, x))

print('d^2f/dy^2: ',diff(fy, y))

print()

k = 0

xk, yk = 0.0, 0.0

while ( max(fx.subs({x: xk, y: yk}), fy.subs({x: xk, y:yk})) >= eps):

phi1 = - (fx.subs({x: xk, y: yk}))\*\*2 - (fy.subs({x: xk, y: yk}))\*\*2

phi2 = diff(fx, x).subs({x: xk, y: yk}) \* (fx.subs({x: xk, y: yk}))\*\*2 + 2 \* diff(fx , y).subs({x: xk, y: yk}) \* fx.subs({x: xk, y: yk}) \* fy.subs({x: xk, y: yk}) + diff(fy, y).subs({x: xk, y: yk}) \* (fy.subs({x: xk, y: yk}))\*\*2

t\_star = - phi1 / phi2

xk = xk - t\_star \* fx.subs({x: xk, y: yk})

yk = yk - t\_star \* fy.subs({x: xk, y: yk})

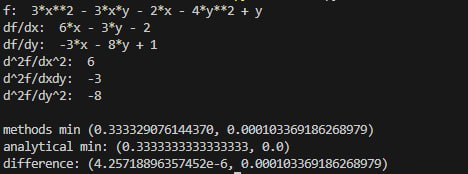
k +=1

print(f'methods min {xk, yk}')

print (f'analytical min: {analytical\_min()}')

print(f'difference: {math.fabs(xk -analytical\_min()[0]), math.fabs(yk -analytical\_min()[1])}')

1. **Результаты**



1. **Вывод**

В ходе выполнения лабораторной работы был изучен и реализован метод наискорейшего спуска, получено приближенное значение минимума функции двух переменных и был найден ее минимум аналитичности. В результате тестирования для приведенной функции после 4 итераций вычислительная погрешность составила около 0.0001.