Министерство образования Республики Беларусь

Учреждение образования

“Брестский государственный технический университет”

**Лабораторная работа №2**

**По дисциплине ЯП за 2 семестр**  
**Тема: «Итерационные методы поиска экстремумов»**

**Вариант 9**

**Выполнил:**

Студент группы ПО-6(1)  
 1-го курса

Мартынович Даниил

**Проверила:**

Анфилец С.В

Градиентный спуск — метод нахождения локального экстремума (минимума или максимума) функции с помощью движения вдоль градиента. Для минимизации функции в направлении градиента используются методы одномерной оптимизации, например, метод золотого сечения. Также можно искать не наилучшую точку в направлении градиента, а какую-либо лучше текущей.

#include<iostream>

#include<math.h>

using namespace std;

double Function(double x, double y, double z);

double dX(double x);

double dY(double y);

double dZ(double z);

int main()

{

setlocale(LC\_ALL, "ru");

int a,i=0;

double x[20], y[20], z[20], ot, dot;

cout << "Введите диапозон поиска\nОт ";

cin >> ot;

cout << "до ";

cin >> dot;

cout << "Введите стартовую точку поиска x0 y0 z0: "<<endl;

cin >> x[0] >> y[0] >> z[0];

cout << Function(x[0], y[0], z[0])<< endl;

if (Function(x[0], y[0], z[0]) > dot || Function(x[0], y[0], z[0]) < ot) {

cout<< "вы вышли за пределы диапазона"<< endl;

cout<< "введите x0,y0,z0"<< endl;

cin>> x[0] >> y[0] >> z[0];

cout << Function(x[0], y[0], z[0]) << endl;

}

cout << "градиент(" << dX(x[0])<< ","<< dY(y[0]) << "," << dZ(z[0]) <<")"<< endl;

cout << "\t1 - Поиск минимума\t2 - Поиск максимума" << endl;

cin >> a;

if (a == 2)//условие максимума

{

for (;; i++)

{

if (i > 1) {

if (Function(x[i], y[i], z[i]) > dot) { break; }

}

x[i+1] = x[i] + 0.1 \* dX(x[i]);

y[i+1] = y[i] + 0.1 \* dY(y[i]);

z[i+1] = z[i] + 0.1 \* dZ(z[i]);

}

cout << Function(x[i - 1], y[i - 1], z[i - 1]) << endl;

cout << "Amax(" << x[i] << "," << y[i] << "," << z[i] << ")";

}

else if (a == 1)//условие минимума

{

for (;; i++)

{

x[i+1] = x[i] - 0.1 \* dX(x[i]);

y[i+1] = y[i] - 0.1 \* dY(y[i]);

z[i+1] = z[i] - 0.1 \* dZ(z[i]);

if (i > 1) {

if ((x[i + 1] - x[i], y[i + 1] - y[i]) < 0.00000001) { break; }

}

}

cout << Function(x[i], y[i], z[i]) << endl;

cout <<"Amin("<<x[i]<<"," << y[i]<<"," <<z[i]<<")";

}

system("pause");

return 0;

}

double Function(double x, double y, double z)

{

double result = x \* y \* pow(z, 2) + (2 \* pow(x, 2) + 3) \* (z + 1);

return result;

}

double dX(double x)

{

double result = 1 + 4 \* x;

return result;

}

double dY(double y)

{

double result = 2 \* y;

return result;

}

double dZ(double z)

{

double result = 2 \* z + 1;

return result;

}



