ФГБОУ ВО

«Уфимский государственный авиационный технический университет»

Кафедра ТК

**ОТЧЕТ**

**по лабораторной работе № 3**

по дисциплине «Методы оптимизации»

**Тема: «Методы безусловной многомерной оптимизации (градиентные методы)»**

**Вариант № 17**

Выполнила: студентка гр. ИВТ-221

Самсонова Е.О.

Проверил: доцент каф. ТК

Хасанов А.Ю.

Уфа 2021

**Содержание**

[Функция 3](#_Toc70948146)

[Задание 3](#_Toc70948147)

[Расчетные таблицы при : 5](#_Toc70948148)

[Расчетные таблицы при : 6](#_Toc70948149)

[Траектория движения точек при : 8](#_Toc70948150)

[Траектория движения точек при : 10](#_Toc70948151)

[Код программы 11](#_Toc70948152)

[Блок-схемы функций методов 16](#_Toc70948153)

[Вывод 19](#_Toc70948154)

Функция

Задание

Построить таблицу для и методами:

1. Градиентный с постоянным шагом;
2. Градиентный с дроблением шага;
3. Метод наискорейшего спуска (градиентный с оптимальным шагом).

**Найти точное решение** (на основе необходимых условий минимума).

В точке минимума функции должны выполняться необходимые условия первого порядка: в точке минимума частные производные первого порядка должны быть равны нулю.

Вычислим частные производные первого порядка целевой функции по :

Из первого уравнения выражаем

И подставим во второе уравнение:

Отсюда:

Ответ:

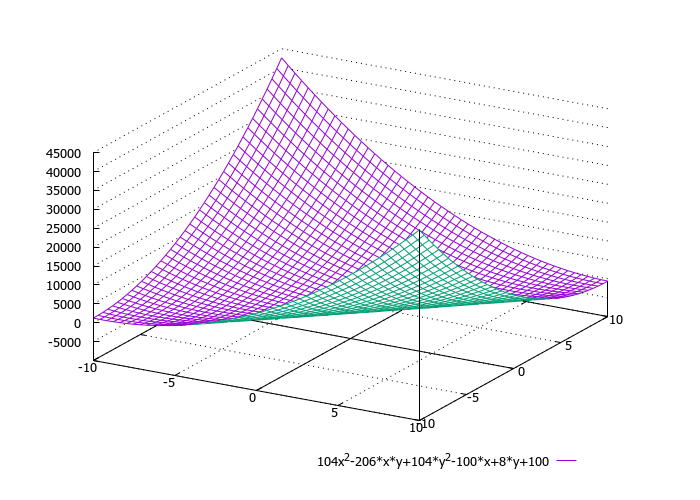


Рисунок 1− График функции

Расчетные таблицы при :

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Градиентный с постоянным шагом | | | | | | | | |
|  |  |  |  |  |  |  |  | l |
|  | 23,1043 | 22,8432 | -965,042 | 0,0371529 | 1352 | 0 | 1352 | 675 |
|  | 23,1276 | 22,8667 | -965,043 | 0,00397956 | 1800 | 0 | 1800 | 899 |
|  | 23,1302 | 22,8693 | -965,043 | 0,000376984 | 2272 | 0 | 2272 | 1135 |
|  | 23,1304 | 22,8695 | -965,043 | 4,88069e-05 | 2572 | 0 | 2572 | 1285 |
|  | 23,1304 | 22,8696 | -965,043 | 8,24227e-06 | 2758 | 0 | 2758 | 1378 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Градиентный с дробленным шагом | | | | | | | | |
|  |  |  |  |  |  |  |  | l |
|  | 23,0953 | 22,8344 | -965,041 | 0,0497028 | 3290 | 1648 | 1642 | 820 |
|  | 23,1269 | 22,866 | -965,043 | 0,00499287 | 4462 | 2234 | 2228 | 1113 |
|  | 23,1301 | 22,8692 | -965,043 | 0,000497683 | 5638 | 2822 | 2816 | 1407 |
|  | 23,1304 | 22,8695 | -965,043 | 5,04453e-05 | 6810 | 3408 | 3402 | 1700 |
|  | − | − | − | − | − | − | − | − |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Наискорейший спуск | | | | | | | | |
|  |  |  |  |  |  |  |  | l |
|  | 23,1043 | 22,8432 | -1165,04 | 0,0371529 | 1366 | 14 | 1352 | 675 |
|  | 23,1276 | 22,8667 | -1165,04 | 0,00397956 | 1814 | 14 | 1800 | 899 |
|  | 23,1302 | 22,8693 | -1165,04 | 0,000376984 | 2286 | 14 | 2272 | 1135 |
|  | 23,1304 | 22,8695 | -1165,04 | 4,88069e-05 | 2586 | 14 | 2572 | 1285 |
|  | 23,1304 | 22,8696 | -1165,04 | 8,24227e-06 | 2772 | 14 | 2758 | 1378 |

Расчетные таблицы при :

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Градиентный с постоянным шагом | | | | | | | | |
|  |  |  |  |  |  |  |  | l |
|  | 23,1043 | 22,8432 | -965,042 | 0,0371398 | 1368 | 0 | 1368 | 683 |
|  | 23,1278 | 22,8669 | -965,043 | 0,00377696 | 1826 | 0 | 1826 | 912 |
|  | 23,1302 | 22,8693 | -965,043 | 0,000375174 | 2288 | 0 | 2288 | 1143 |
|  | 23,1304 | 22,8695 | -965,043 | 4,57748e-05 | 2584 | 0 | 2584 | 1291 |
|  | 23,1304 | 22,8696 | -965,043 | 7,54619e-06 | 2790 | 0 | 2790 | 1394 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Градиентный с дробленным шагом | | | | | | | | |
|  |  |  |  |  |  |  |  | l |
|  | 23,0953 | 22,8344 | -965,041 | 0,0497394 | 3334 | 1670 | 1664 | 831 |
|  | 23,1269 | 22,866 | -965,043 | 0,00499655 | 4506 | 2256 | 2250 | 1124 |
|  | 23,1301 | 22,8692 | -965,043 | 0,000498049 | 5682 | 2844 | 2838 | 1418 |
|  | 23,1304 | 22,8695 | -965,043 | 5,00945e-05 | 6858 | 3432 | 3426 | 1712 |
|  | − | − | − | − | − | − | − | − |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Наискорейший спуск | | | | | | | | |
|  |  |  |  |  |  |  |  | l |
|  | 23,1043 | 22,8432 | -1165,04 | 0,0371398 | 1382 | 14 | 1368 | 683 |
|  | 23,1278 | 22,8669 | -1165,04 | 0,00377696 | 1840 | 14 | 1826 | 912 |
|  | 23,1302 | 22,8693 | -1165,04 | 0,000375174 | 2302 | 14 | 2288 | 1143 |
|  | 23,1304 | 22,8695 | -1165,04 | 4,57748e-05 | 2598 | 14 | 2584 | 1291 |
|  | 23,1304 | 22,8696 | -1165,04 | 7,54619e-06 | 2804 | 14 | 2790 | 1394 |

Траектория движения точек при :

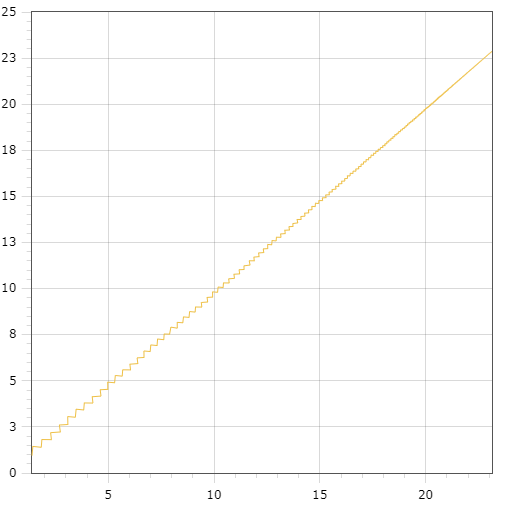


Рисунок 1 − Градиентный с постоянным шагом

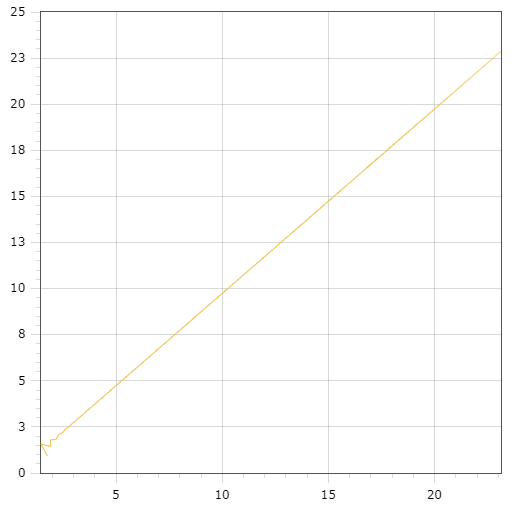


Рисунок 2 − Градиентный с дробленным шагом

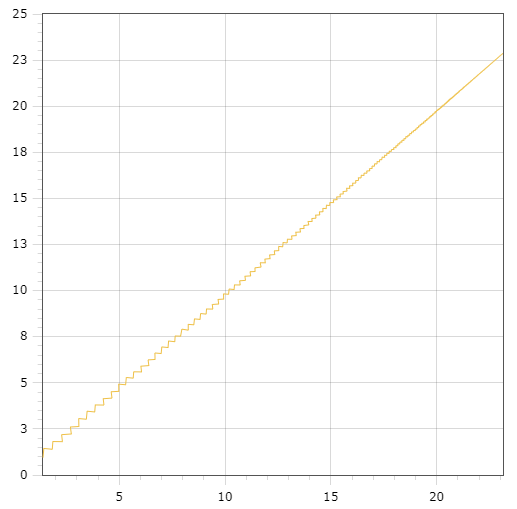


Рисунок 3 − Наискорейший спуск

Траектория движения точек при :

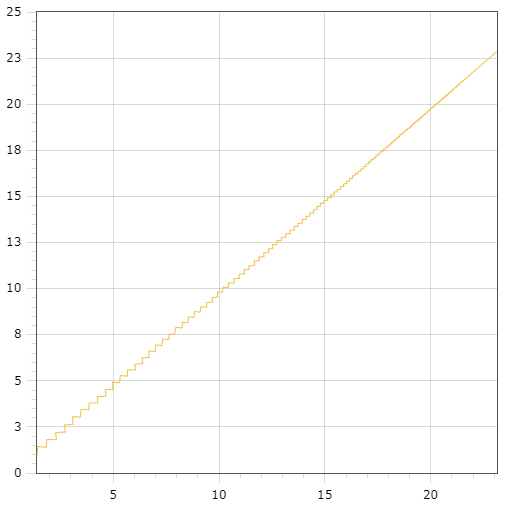


Рисунок 4 − Градиентный с постоянным шагом

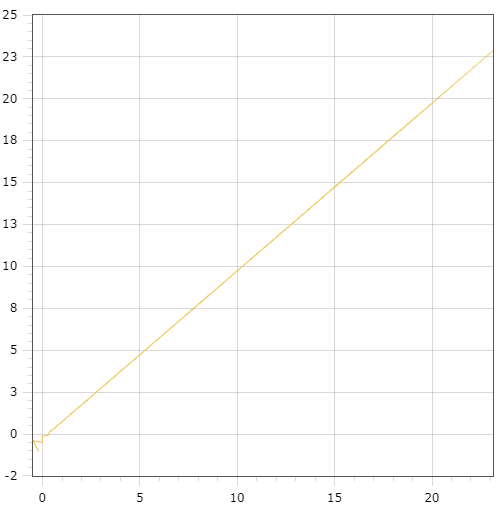


Рисунок 5 − Градиентный с дробленным шагом

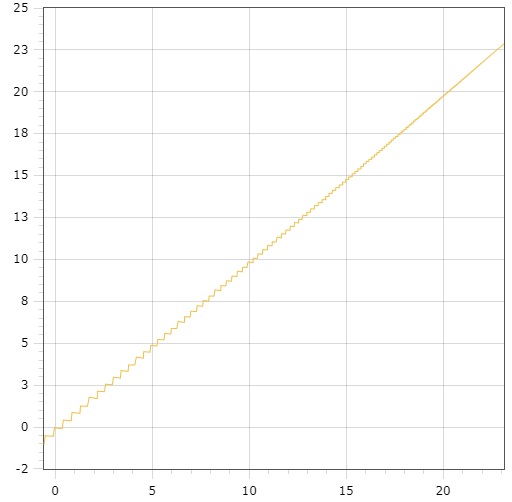


Рисунок 6 − Наискорейший спуск

Код программы

#include <cstdlib>

#include <iostream>

#include <Windows.h>

#include <cstring>

#include <conio.h>

#include <iomanip>

#include <fstream>

#include <cmath>

#include <malloc.h>

using namespace std;

void findAB(double x1, double x2, double z1, double z2, int h, double& a, double& b, int& N0)

{

double y1, y2;

int i;

y1 = 104 \* pow(x1, 2) - 206 \* x1 \* x2 + 104 \* pow(x2, 2) - 100 \* x1 + 8 \* x2 + 100;

i = 1;

while (1)

{

y2 = 104 \* pow((x1 - (i - h) \* z1), 2) - 206 \* (x1 - (i - h) \* z1) \* (x2 - (i - h) \* z2) + 104 \* pow((x2 - (i - h) \* z2), 2) - 100 \* (x1 - (i - h)\*z1) + 8 \* (x2 - (i - h) \* z2) + 100;

N0 = N0 + 1;

if (y1 > y2)

{

i = i + 1;

y1 = y2;

}

else

break;

}

b = i \* h;

if (i = 1)

a = 0;

else

a = (i - 2) \* h;

}

void find(double a, double b, double x1, double x2, double z1, double z2, double& lam, int& N0)

{

double t, e, y0, y1, y, xk, egar, lam0, lam1;

int j, i;

e = 0.001;

t = (1 + sqrt(5)) / 2;

lam0 = b - (b - a) / t;

lam1 = a + (b - a) / t;

y0 = 104 \* pow((x1 - lam0 \* z1), 2) - 206 \* (x1 - lam0 \* z1) \* (x2 - lam0 \* z2) + 104 \* pow((x2 - lam0 \* z2), 2) - 100 \* (x1 -

lam0 \* z1) + 8 \* (x2 - lam0 \* z2) + 100;

y1 = 104 \* pow((x1 - lam1 \* z1), 2) - 206 \* (x1 - lam1 \* z1) \* (x2 - lam1 \* z2) + 104 \* pow((x2 - lam1 \* z2), 2) - 100 \* (x1 -

lam1 \* z1) + 8 \* (x2 - lam1 \* z2) + 100;

N0 = 2;

do

{

if (y0 < y1)

{

b = lam1;

lam1 = lam0;

y1 = y0;

lam0 = a + b - lam1;

y0 = 104 \* pow((x1 - lam0 \* z1), 2) - 206 \* (x1 - lam0 \* z1) \* (x2 - lam0 \* z2) + 104 \* pow((x2 - lam0 \* z2), 2) - 100 \* (x1 -

lam0 \* z1) + 8 \* (x2 - lam0 \* z2) + 100;

}

else

{

a = lam0;

lam0 = lam1;

y0 = y1;

lam1 = a + b - lam0;

y1 = 104 \* pow((x1 - lam1 \* z1), 2) - 206 \* (x1 - lam1 \* z1) \* (x2 - lam1 \* z2) + 104 \* pow((x2 - lam1 \* z2), 2) - 100 \* (x1 -

lam1 \* z1) + 8 \* (x2 - lam1 \* z2) + 100;

}

N0 = N0 + 1;

} while ((b - a) > 2 \* e \* t);

if (y0 < y1)

b = lam1;

else

a = lam0;

lam = (a + b) / 2;

egar = (b - a) / 2;

y = 104 \* pow((x1 - lam \* z1), 2) - 206 \* (x1 - lam \* z1) \* (x2 - lam \* z2) + 104 \* pow((x2 - lam \* z2), 2) - 100\* (x1 - lam \* z1) + 8 \* (x2 -

lam \* z2) + 100;

}

void spusk(double a, double b)

{

ofstream fout;

const int L = 10;

char file[L];

fout.open("rrr3.txt");

double e1, y1, x, y, xk, yk, z1, z2, y2, z, lam, x1, x2, x1l, x2l, egar;

int N, l, j, i, k, N0, N1;

cout << "Введите e1\n";

cout << "e1= ";

cin >> e1;

int h = 1;

x1 = a;

x2 = b;

z1 = 208 \* x1 - 206 \* x2 - 100;

z2 = -206 \* x1 + 208 \* x2 + 8;

N0 = 0;

N1 = 2;

l = 0;

fout << "x1; x2;";

fout << "1; 1;";

do

{

findAB(x1, x2, z1, z2, h, a, b, N0);

find(a, b, x1, x2, z1, z2, lam, N0);

/\*cout << a << " " << b << endl;\*/

x1l = x1 - lam \* z1;

x2l = x2 - lam \* z2;

z1 = 208 \* x1l - 206 \* x2l - 100;

z2 = -206 \* x1l + 208 \* x2l + 8;

N1 = N1 + 2;

l = l + 1;

x1 = x1l;

x2 = x2l;

/\*cout << x1 << "; " << x2 << ";" << endl;

cout << right;\*/

fout << x1 << "; " << x2 << ";" << endl;

fout << right;

} while ((sqrt(pow(z1, 2) + pow(z2, 2))) > e1);

xk = x1;

yk = x2;

y = 104 \* pow(xk, 2) - 206 \* xk \* yk + 104 \* pow(yk, 2) - 100 \* xk + 8 \* yk - 100;

N = N0 + N1;

egar = sqrt((pow((x1 - 23.13043), 2)) + (pow((x2 - 22.86957), 2)));

cout << "Корень X= " << xk << endl;

cout << "Корень Y= " << yk << endl;

cout << "Корень Y1= " << y << endl;

cout << "Число l= " << l << endl;

cout << "Число N0= " << N0 << endl;

cout << "Число N1= " << N1 << endl;

cout << "Число N= " << N << endl;

cout << "Число Ефакт.= " << egar << endl;

system("pause");

}

void droblshag(double a, double b)

{

ofstream fout;

const int L = 10;

char file[L];

fout.open("rrr2.txt");

double e1, y1, x, y, xk, yk, z1, z2, y2, z, lam, x1, x2, x1l, x2l, egar;

int N, l, j, i, k, N0, N1;

cout << "Введите e1\n";

cout << "e1= ";

/\*cout << a; cout << b;\*/

cin >> e1;

x1 = a;

x2 = b;

z1 = 208 \* x1 - 206 \* x2 - 100;

z2 = -206 \* x1 + 208 \* x2 + 8;

N0 = 0;

N1 = 2;

l = 0;

lam = 1;

fout << "x1; x2;";

do

{

y = 104 \* pow(x1, 2) - 206 \* x1 \* x2 + 104 \* pow(x2, 2) - 100 \* x1 + 8 \* x2 + 100;

N0 = N0 + 1;

do

{

y1 = 104 \* pow((x1 - lam \* z1), 2) - 206 \* (x1 - lam \* z1) \* (x2 - lam \* z2) + 104 \* pow((x2 -

lam \* z2), 2) - 100 \* (x1 - lam \* z1) + 8 \* (x2 - lam \* z2) + 100;

N0 = N0 + 1;

lam = lam / 2;

} while (y <= y1);

lam = lam \* 2;

x1l = x1 - lam \* z1;

x2l = x2 - lam \* z2;

z1 = 208 \* x1l - 206 \* x2l - 100;

z2 = - 206 \* x1l + 208 \* x2l + 8;

x1 = x1l;

x2 = x2l;

/\*cout << x1 << "; " << x2 << ";" << endl;

cout << right;\*/

fout << x1 << "; " << x2 << ";" << endl;

fout << right;

N1 = N1 + 2;

l = l + 1;

} while ((sqrt(pow(z1, 2) + pow(z2, 2))) > e1);

xk = x1;

yk = x2;

y = 104 \* pow(x1, 2) - 206 \* x1 \* x2 + 104 \* pow(x2, 2) - 100 \* x1 + 8 \* x2 + 100;

N = N0 + N1;

egar = sqrt((pow((x1 - 23.13043), 2)) + (pow((x2 - 22.86957), 2)));

cout << "Корень X= " << xk << endl;

cout << "Корень Y= " << yk << endl;

cout << "Корень Y1= " << y << endl;

cout << "Число l= " << l << endl;

cout << "Число N0= " << N0 << endl;

cout << "Число N1= " << N1 << endl;

cout << "Число N= " << N << endl;

cout << "Число Ефакт.= " << egar << endl;

system("pause");

}

void postshag(double a, double b)

{

ofstream fout;

const int L = 10;

char file[L];

fout.open("rrr1.txt");

double e1, y1, x, y, xk, yk, z1, z2, y2, z, lam, x1, x2, x1l, x2l, egar;

int N, l, j, i, k, N0, N1;

cout << "Введите e1\n";

cout << "e1= ";

cin >> e1;

int h = 1;

x1 = a;

x2 = b;

z1 = 208 \* x1 - 206 \* x2 - 100;

z2 = -206 \* x1 + 208 \* x2 + 8;

N0 = 0;

N = 2;

l = 0;

//lam = 1;

fout << "x1; x2;";

cout << "x1; x2;";

do

{

findAB(x1, x2, z1, z2, h, a, b, N0);

find(a, b, x1, x2, z1, z2, lam, N0);

cout << /\*a << " " << b << \*/"лямбда: " << lam << endl;

x1l = x1 - lam \* z1;

x2l = x2 - lam \* z2;

z1 = 208 \* x1l - 206 \* x2l - 100;

z2 = -206 \* x1l + 208 \* x2l + 8;

N = N + 2;

l = l + 1;

x1 = x1l;

x2 = x2l;

fout << x1 << "; " << x2 << ";" << endl;

fout << right;

cout << x1 << "; " << x2 << ";" << endl;

cout << right;

} while ((sqrt(pow(z1, 2) + pow(z2, 2))) > e1);

xk = x1;

yk = x2;

y = 104 \* pow(x1, 2) - 206 \* x1 \* x2 + 104 \* pow(x2, 2) - 100 \* x1 + 8 \* x2 + 100;

N0 = 0;

N1 = N;

egar = sqrt((pow((x1 - 23.13043), 2)) + (pow((x2 - 22.86957), 2)));

cout << "Корень X= " << xk << endl;

cout << "Корень Y= " << yk << endl;

cout << "Корень Y1= " << y << endl;

cout << "Число l= " << l << endl;

cout << "Число N0= " << N0 << endl;

cout << "Число N1= " << N1 << endl;

cout << "Число N= " << N << endl;

cout << "Число Ефакт.= " << egar << endl;

system("pause");

}

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

setlocale(LC\_ALL, "Russian");

double x10, x20;

int j;

cout << " Введите значение для (x1[0];x2[0]) " << endl;

cout << " x1[0]= ";

cin >> x10;

cout << " x2[0]= ";

cin >> x20;

while (1)

{

cout << "1 Градиентный с дробленным шагом\n";

cout << "2 Градиентный с постоянным шагом\n";

cout << "3 Наискорейший спуск\n";

cout << "0 Выход из программы...\n\n";

cout << " Ваш выбор: \n";

cin >> j;

switch (j)

{

case 1: droblshag(x10, x20); break;

case 2: postshag(x10, x20); break;

case 3: spusk(x10, x20); break;

case 0: cout << " Конец работы.\n"; system("pause"); return 0;

default: cout << " Ошибка!\n";

system("pause");

}

system("cls");

}

}

Блок-схемы функций методов



Рисунок 7 − Блок-схема градиентного метода с постоянным шагом



Рисунок 8 − Блок-схема градиентного метода с дроблением шага



Рисунок 9 − Блок-схема наискорейшего спуска

Вывод

В лабораторной работе были сравнены градиентные методы: градиентный метод с дроблением шага, градиентный метод с постоянным шагом и метод наискорейшего спуска. Для заданной целевой функции для заданных условий окончания поиска по критерию минимального числа экспериментов лучшим оказался **метод наискорейшего спуска**.