ФГБОУ ВО

«Уфимский государственный авиационный технический университет»

Кафедра ТК

**ОТЧЕТ**

**по лабораторной работе № 3**

по дисциплине «Методы оптимизации»

**Тема: «МЕТОДЫ БЕЗУСЛОВНОЙ МНОГОМЕРНОЙ ОПТИМИЗАЦИИ»**

Вариант № 13

Выполнил: студент гр. ИВТ-221

Хуснутдинов Х.В.

Проверил: доцент каф. ТК

Хасанов А.Ю.

Уфа 2021

**Функция**

**Целевая функция**

**Задание**

Построить таблицу для и методами:

1. Градиентный с постоянным шагом;
2. Градиентный с дроблением шага;
3. Метод наискорейшего спуска.

**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:

**График функции**

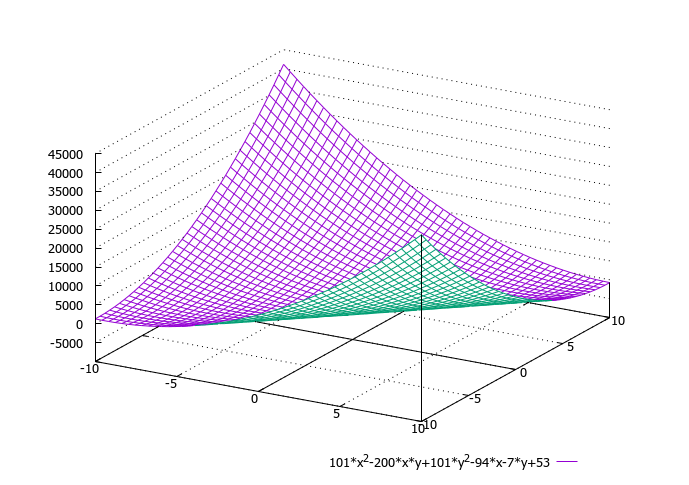


Рисунок 1− График функции

**Расчетные таблицы при :**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Градиентный с дробленным шагом | | | | | | | | |
|  |  |  |  |  |  |  |  | l |
|  | 25,3229 | 25,1065 | -1226,83 | 0,0498649 | 3338 | 1672 | 1666 | 832 |
|  | 25,3547 | 25,1383 | -1226,83 | 0,00497001 | 4514 | 2260 | 2254 | 1126 |
|  | 25,3579 | 25,1414 | -1226,83 | 0,000499262 | 5686 | 2846 | 2840 | 1419 |
|  | 25,3582 | 25,1418 | -1226,83 | 4,9783e-05 | 6862 | 3434 | 3428 | 1713 |
|  | − | − | − | − | − | − | − | − |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Градиентный с постоянным шагом | | | | | | | | |
|  |  |  |  |  |  |  |  | l |
|  | 25,3309 | 25,1147 | -1226,83 | 0,0384154 | 1302 | 0 | 1302 | 650 |
|  | 25,3554 | 25,139 | -1226,83 | 0,00393039 | 1738 | 0 | 1738 | 868 |
|  | 25,3579 | 25,1415 | -1226,83 | 0,000402137 | 2174 | 0 | 2174 | 1086 |
|  | 25,3582 | 25,1418 | -1226,83 | 3,74379e-05 | 2450 | 0 | 2450 | 1224 |
|  | 25,3582 | 25,1418 | -1226,83 | 4,53222e-06 | 2642 | 0 | 2642 | 1320 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Наискорейший спуск | | | | | | | | |
|  |  |  |  |  |  |  |  | l |
|  | 25,3309 | 25,1147 | -1332,83 | 0,0384154 | 1316 | 14 | 1302 | 650 |
|  | 25,3554 | 25,139 | -1332,83 | 0,00393039 | 1752 | 14 | 1738 | 868 |
|  | 25,3579 | 25,1415 | -1332,83 | 0,000402137 | 2188 | 14 | 2174 | 1086 |
|  | 25,3582 | 25,1418 | -1332,83 | 3,74379e-05 | 2464 | 14 | 2450 | 1224 |
|  | 25,3582 | 25,1418 | -1332,83 | 4,53222e-06 | 2656 | 14 | 2642 | 1320 |

**Расчетные таблицы при :**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Градиентный с дробленным шагом | | | | | | | | |
|  |  |  |  |  |  |  |  | l |
|  | 25,3229 | 25,1065 | -1226,83 | 0,0499057 | 3378 | 1692 | 1686 | 842 |
|  | 25,3547 | 25,1383 | -1226,83 | 0,00497408 | 4554 | 2280 | 2274 | 1136 |
|  | 25,3579 | 25,1414 | -1226,83 | 0,00049967 | 5726 | 2866 | 2860 | 1429 |
|  | 25,3582 | 25,1418 | -1226,83 | 4,98237e-05 | 6902 | 3454 | 3448 | 1723 |
|  | − | − | − | − | − | − | − | − |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Градиентный с постоянным шагом | | | | | | | | |
|  |  |  |  |  |  |  |  | l |
|  | 25,3298 | 25,1132 | -1226,83 | 0,0402867 | 1308 | 0 | 1308 | 653 |
|  | 25,3554 | 25,139 | -1226,83 | 0,00391342 | 1754 | 0 | 1754 | 876 |
|  | 25,3579 | 25,1415 | -1226,83 | 0,0004004 | 2190 | 0 | 2190 | 1094 |
|  | 25,3582 | 25,1418 | -1226,83 | 4,6009e-05 | 2478 | 0 | 2478 | 1238 |
|  | 25,3582 | 25,1418 | -1226,83 | 4,59392e-06 | 2648 | 0 | 2648 | 1323 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Наискорейший спуск | | | | | | | | |
|  |  |  |  |  |  |  |  | l |
|  | 25,3298 | 25,1132 | -1332,83 | 0,0402867 | 1322 | 14 | 1308 | 653 |
|  | 25,3554 | 25,139 | -1332,83 | 0,00391342 | 1768 | 14 | 1754 | 876 |
|  | 25,3579 | 25,1415 | -1332,83 | 0,0004004 | 2204 | 14 | 2190 | 1094 |
|  | 25,3582 | 25,1418 | -1332,83 | 4,6009e-05 | 2492 | 14 | 2478 | 1238 |
|  | 25,3582 | 25,1418 | -1332,83 | 4,59392e-06 | 2662 | 14 | 2648 | 1323 |

**Траектория движения точек при :**

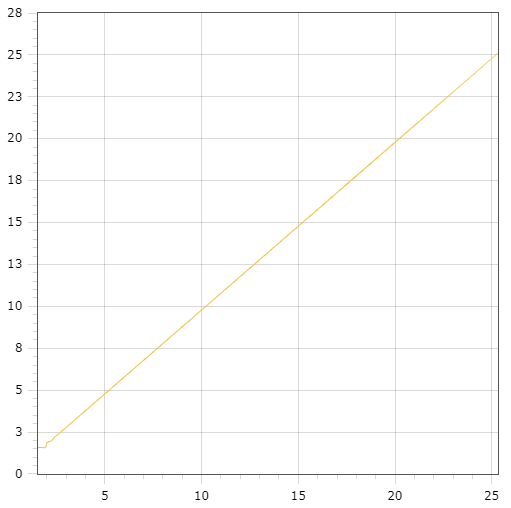


Рисунок 2 − Градиентный с дробленным шагом

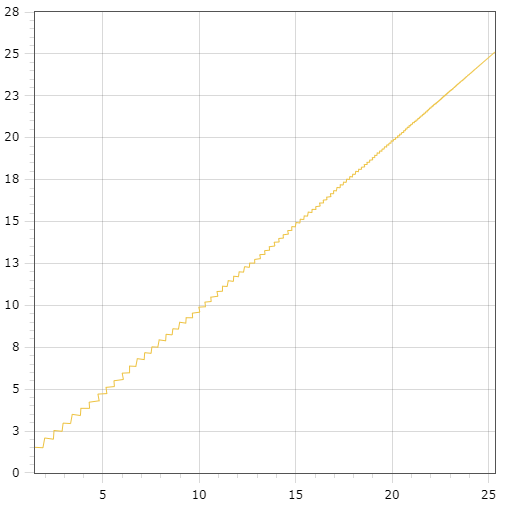


Рисунок 3 − Градиентный с постоянным шагом

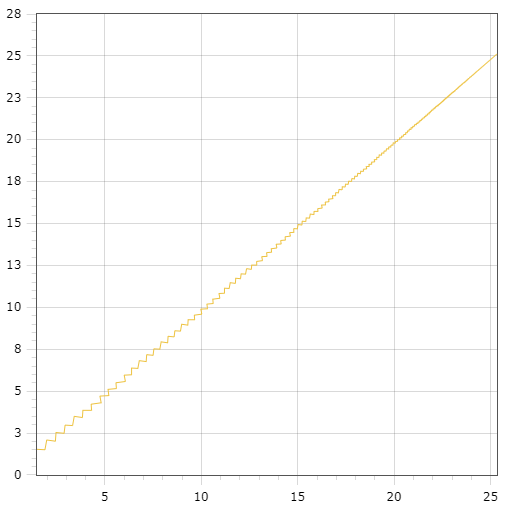


Рисунок 4 − Наискорейший спуск

Траектория движения точек при :

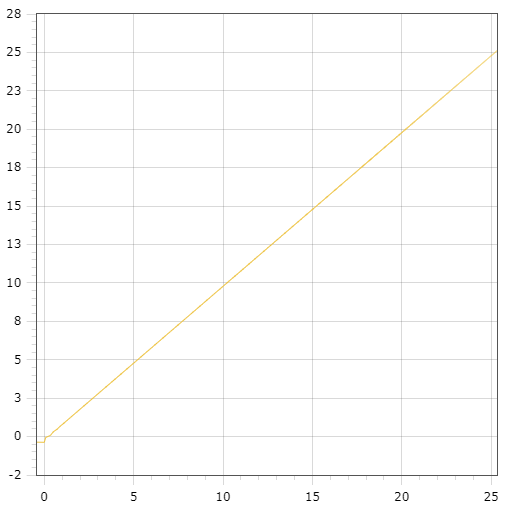


Рисунок 5 − Градиентный с дробленным шагом

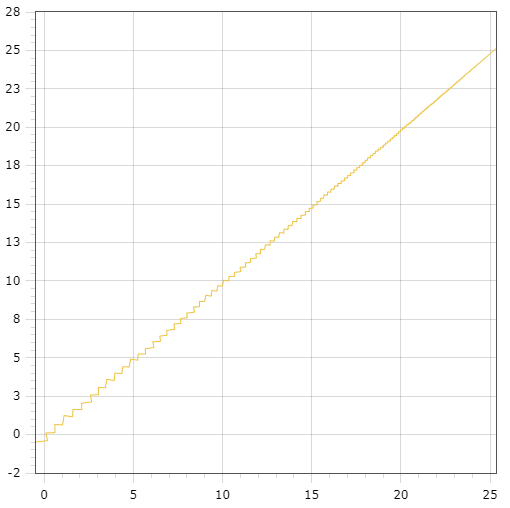


Рисунок 6 − Градиентный с постоянным шагом

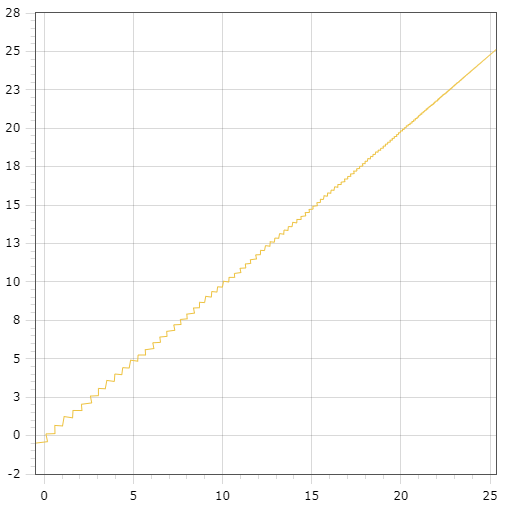


Рисунок 7 − Наискорейший спуск

**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 = 101 \* pow(x1, 2) - 200 \* x1 \* x2 + 101 \* pow(x2, 2) - 94 \* x1 - 7 \* x2 + 53;

i = 1;

while (1)

{

y2 = 101 \* pow((x1 - (i - h) \* z1), 2) - 200 \* (x1 - (i - h) \* z1) \* (x2 - (i - h) \* z2) + 101 \* pow((x2 - (i - h) \* z2), 2) - 94 \* (x1 - (i - h) \* z1) - 7 \* (x2 - (i - h) \* z2) + 53;

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 = 101 \* pow((x1 - lam0 \* z1), 2) - 200 \* (x1 - lam0 \* z1) \* (x2 - lam0 \* z2) + 101 \* pow((x2 - lam0 \* z2), 2) - 94 \* (x1 -

lam0 \* z1) - 7 \* (x2 - lam0 \* z2) + 53;

y1 = 101 \* pow((x1 - lam1 \* z1), 2) - 200 \* (x1 - lam1 \* z1) \* (x2 - lam1 \* z2) + 101 \* pow((x2 - lam1 \* z2), 2) - 94 \* (x1 -

lam1 \* z1) - 7 \* (x2 - lam1 \* z2) + 53;

N0 = 2;

do

{

if (y0 < y1)

{

b = lam1;

lam1 = lam0;

y1 = y0;

lam0 = a + b - lam1;

y0 = 101 \* pow((x1 - lam0 \* z1), 2) - 200 \* (x1 - lam0 \* z1) \* (x2 - lam0 \* z2) + 101 \* pow((x2 - lam0 \* z2), 2) - 94 \* (x1 -

lam0 \* z1) - 7 \* (x2 - lam0 \* z2) + 53;

}

else

{

a = lam0;

lam0 = lam1;

y0 = y1;

lam1 = a + b - lam0;

y1 = 101 \* pow((x1 - lam1 \* z1), 2) - 200 \* (x1 - lam1 \* z1) \* (x2 - lam1 \* z2) + 101 \* pow((x2 - lam1 \* z2), 2) - 94 \* (x1 -

lam1 \* z1) - 7 \* (x2 - lam1 \* z2) + 53;

}

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 = 101 \* pow((x1 - lam \* z1), 2) - 200 \* (x1 - lam \* z1) \* (x2 - lam \* z2) + 101 \* pow((x2 - lam \* z2), 2) - 94 \* (x1 - lam \* z1) - 7 \* (x2 -

lam \* z2) + 53;

}

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 = 202 \* x1 - 200 \* x2 - 94;

z2 = -200 \* x1 + 202 \* x2 - 7;

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);

x1l = x1 - lam \* z1;

x2l = x2 - lam \* z2;

z1 = 202 \* x1l - 200 \* x2l - 94;

z2 = -200 \* x1l + 202 \* x2l - 7;

N1 = N1 + 2;

l = l + 1;

x1 = x1l;

x2 = x2l;

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

fout << right;

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

xk = x1;

yk = x2;

y = 101 \* pow(xk, 2) - 200 \* xk \* yk + 101 \* pow(yk, 2) - 94 \* xk - 7 \* yk - 53;

N = N0 + N1;

egar = sqrt((pow((x1 - 25.35821), 2)) + (pow((x2 - 25.14179), 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= ";

cin >> e1;

x1 = a;

x2 = b;

z1 = 202 \* x1 - 200 \* x2 - 94;

z2 = -200 \* x1 + 202 \* x2 - 7;

N0 = 0;

N1 = 2;

l = 0;

lam = 1;

fout << "x1; x2;";

do

{

y = 101 \* pow(x1, 2) - 200 \* x1 \* x2 + 101 \* pow(x2, 2) - 94 \* x1 - 7 \* x2 + 53;

N0 = N0 + 1;

do

{

y1 = 101 \* pow((x1 - lam \* z1), 2) - 200 \* (x1 - lam \* z1) \* (x2 - lam \* z2) + 101 \* pow((x2 -

lam \* z2), 2) - 94 \* (x1 - lam \* z1) - 7 \* (x2 - lam \* z2) + 53;

N0 = N0 + 1;

lam = lam / 2;

} while (y <= y1);

lam = lam \* 2;

x1l = x1 - lam \* z1;

x2l = x2 - lam \* z2;

z1 = 202 \* x1l - 200 \* x2l - 94;

z2 = -200 \* x1l + 202 \* x2l - 7;

x1 = x1l;

x2 = x2l;

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 = 101 \* pow(x1, 2) - 200 \* x1 \* x2 + 101 \* pow(x2, 2) - 94 \* x1 - 7 \* x2 + 53;

N = N0 + N1;

egar = sqrt((pow((x1 - 25.35821), 2)) + (pow((x2 - 25.14179), 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 = 202 \* x1 - 200 \* x2 - 94;

z2 = -200 \* x1 + 202 \* x2 - 7;

N0 = 0;

N = 2;

l = 0;

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 << "лямбда: " << lam << endl;

x1l = x1 - lam \* z1;

x2l = x2 - lam \* z2;

z1 = 202 \* x1l - 200 \* x2l - 94;

z2 = -200 \* x1l + 202 \* x2l - 7;

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 = 101 \* pow(x1, 2) - 200 \* x1 \* x2 + 101 \* pow(x2, 2) - 94 \* x1 - 7 \* x2 + 53;

N0 = 0;

N1 = N;

egar = sqrt((pow((x1 - 25.35821), 2)) + (pow((x2 - 25.14179), 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 8 – Блок-схема градиентного метода с дроблением шага



Рисунок 9 – Блок-схема градиентного метода с постоянным шагом



Рисунок 10 – Блок-схема наискорейшего спуска

**Вывод:** в ходе выполнения данной лабораторной работы были сравнены градиентные методы (градиентный метод с дроблением шага; градиентный метод с постоянным шагом; метод наискорейшего спуска). Лучшим методом оказался метод наискорейшего спуска, для заданной целевой функции и для заданных условий окончания поиска по критерию минимального числа экспериментов.