

# **Московский авиационный институт (национальный исследовательский университет)**

**Институт №3.**

**«Системы управления, информатика и  
электроэнергетика».**

**Кафедра №304**

**"Вычислительные машины, системы и сети"**

**Отчет по лабораторной работе по учебной дисциплине:  
«Теория оптимального планирования и управления» на тему:**

**«Методы многомерной оптимизации»**

Группа М30-207Б-18

Выполнили:  
Никита Гордеев  
Богущ Иван

Приняли:  
  
Татарникова Е.  
М. Красовская М. А.

**Москва 2019**

# **Московский авиационный институт (национальный исследовательский университет)**

**Институт №3.**

**«Системы управления, информатика и  
электроэнергетика».**

**Кафедра №304**

**"Вычислительные машины, системы и сети"**

**Отчет по лабораторной работе по учебной дисциплине:  
«Теория оптимального планирования и управления» на тему:**

**«Условия оптимальности Джона и Куна-Таккера»**

Группа М30-207Б-18

Выполнили:  
Никита Гордеев  
Богущ Иван

Приняли:  
Татарникова Е.  
М. Красовская М. А.

Москва 2019

# **Московский авиационный институт (национальный исследовательский университет)**

**Институт №3.**

**«Системы управления, информатика и  
электроэнергетика».**

**Кафедра №304**

**"Вычислительные машины, системы и сети"**

**Отчет по лабораторной работе по учебной дисциплине:  
«Теория оптимального планирования и управления» на тему:**

**«Градиентные методы»**

Группа М30-207Б-18

Выполнили:  
Никита Гордеев  
Богущ Иван

Приняли:  
Татарникова Е.  
М. Красовская М. А.

Москва 2019

## Алгоритм

Параметрами метода являются:

- коэффициент отражения  $\alpha > 0$ , обычно выбирается равным 1.
- коэффициент сжатия  $\beta > 0$ , обычно выбирается равным 0.5.
- коэффициент растяжения  $\gamma > 0$ , обычно выбирается равным 2.

Инициализация. Произвольным образом выбирается  $n+1$  точка

$x_i = (x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(n)})$ , образующие симплекс  $n$ -мерного пространства. В этих точках вычисляются значения функции:  $f_1 = f(x_1^{(1)}), f_2 = f(x_2^{(2)}), \dots, f_{n+1} = f(x_{n+1}^{(n+1)})$ .

1. Сортировка. Из вершин симплекса выбираем три точки:  $x_h$  с наибольшим (из выбранных) значением функции  $f_h$ ,  $x_g$  со следующим по величине значением  $f_g$  и  $x_l$  с наименьшим значением функции  $f_l$ . Целью дальнейших манипуляций будет уменьшение по крайней мере  $f_h$ .

2. Найдём центр тяжести всех точек, за исключением  $x_h$ . Вычислять  $f_c = f(x_c)$  не обязательно.

$$x_h: x_c = \frac{1}{n} \sum_{i \neq h} x_i$$

3. Отражение. Отразим точку  $x_h$  относительно  $x_c$  с коэффициентом  $\alpha$  (при  $\alpha=1$  это будет центральная симметрия, в общем случае — гомотетия), получим точку  $x_r$  и вычислим в ней функцию:  $f_r = f(x_r)$ . Координаты новой точки вычисляются по формуле  $x_r = (1+\alpha)x_c - \alpha x_h$

4. Далее сравниваем значение  $f_r$  со значениями  $f_h, f_g, f_l$ :

4а. Если  $f_r < f_l$ , то производим растяжение. Новая точка  $x_e = (1-\gamma)x_c + \gamma x_r$  и значение функции  $f_e = f(x_e)$ .

Если  $f_e < f_l$ , то заменяем точку  $x_h$  на  $x_e$  и заканчиваем итерацию (на шаг 8).

Если  $f_e > f_l$ , то заменяем точку  $x_h$  на  $x_r$  и заканчиваем итерацию (на шаг 8).

4б. Если  $f_l < f_r < f_g$ , то заменяем точку  $x_h$  на  $x_r$  и переходим на шаг 8.

4с. Если  $f_h > f_r > f_g$ , то меняем обозначения  $x_r, x_h$  (и соответствующие значения функции) местами и переходим на шаг 5.

4д. Если  $f_r > f_h$ , то переходим на шаг 5.

5. Сжатие. Строим точку  $x_s = \beta x_h + (1-\beta)x_c$  и вычисляем в ней значение  $f_s$ .

6. Если  $f_s < f_h$ , то заменяем точку  $x_h$  на  $x_s$  и переходим на шаг 8.
7. Если  $f_s > f_h$ , то производим сжатие симплекса — гомотетию к точке с наименьшим значением  $x_0$ :  $x_i \rightarrow x_0 + (x_i - x_0)/2$  для всех требуемых точек  $x_i$ .
8. Последний шаг — проверка сходимости. Может выполняться по-разному, например, оценкой дисперсии набора точек. Суть проверки заключается в том, чтобы проверить взаимную близость полученных вершин симплекса, что предполагает и близость их к искомому минимуму. Если требуемая точность ещё не достигнута, можно продолжить итерации с шага 1.

## Код программы

```
#include <iostream>
#include <iomanip>

using namespace std;

const int i = 2;          //количество переменных
const int j = i + 1;      //количество вершин

//-----ФУНКЦИИ
//
//нет мин, но есть max(0,1)
//double F(double x[2]) { return -(x[0] * x[0] - x[1] * x[1] + x[0] * x[1] - x[0] + 2 * x[1]); }
//
//для самопроверки мин(0.0)
//double F(double x[2]) { return (x[0]) * (x[0]) + x[1] * x[1]; }
//
//нет max и минимума, но есть локальные минимумы в (.) = (0.0) и (1.1)
double F(double x[2]) { return ((x[0] - x[1]) * (x[0] - x[1]) + 7 * (x[1] * x[1] - x[0]) * (x[1] *
x[1] - x[0])); }
//-----//

//-----УСЛОВИЕ ОСТАНОВКИ (ДИСПЕРСИЯ)
double D(double xh[2], double xg[2], double xl[2], double xc[2])
{ return sqrt(1 / (j + 1) * ((F(xh)) - F(xc)) * ((F(xh)) - F(xc)) + ((F(xg)) - F(xc)) * ((F(xg)) -
F(xc)) + ((F(xl)) - F(xc)) * ((F(xl)) - F(xc)))); }

void cопт(double x[j][i]);
void Таблица(double x[j][i], double x_наим[2], double x_2_по_вечечене[2], double
x_наиб[2], double центр_тяж[2], double отр[2], double отр_рас[2], double
отр_сжат[2], int k);

int main() {
    system("color F0");          //фон белый буквы черные
    int n;                       //индексная переменная
    int k = 0;                   //количество итераций
    bool f = 0;                 //вспомогательный флаг
    double Temp[i];             //обменная переменная

    double Eps = 0.01;          //точность
    double a = 1;               //коэффициент отражения
    double b = 0.5;             //коэффициент сжатия
    double y = 2;               //коэффициент растяжения
    //начальный набор точек

    double x_наиб[2];           //точка с наибольшим значением функции
    double x_2_по_вечечене[2]; //точка с 2 по величине значением функции
    double x_наим[2];           //точка с самым маленьким значением функции

    double центр_тяж[2];        //центр тяжести
    double отр[2];              //отраженный x
    double отр_рас[2];          //отраженный x с растяжением
```

```

double отр_сжат[2];                //отраженный х с жатеем

double x[j][i] = { { -2,-2 }, { -1,-1 }, { 0,-1 } };
setlocale(LC_ALL, "Russian");
cout << "Введите Eps : ";
cin >> Eps;
cout << "Введите коэффициент отражения : ";
cin >> a;
cout << "Введите коэффициент сжатия : ";
cin >> b;
cout << "Введите коэффициент растяжения : ";
cin >> y;
setlocale(LC_ALL, "C");

cout << "\n\n" << char(218) << setfill(char(196)) << setw(154) << char(191) << endl;
do {
    k++;
    сорт(x);           //сортировка точек
    for (n = 0; n < i; n++)
    {
        центр_тяж[n] = 0;
        отр[n] = 0;
        отр_рас[n] = 0;
        отр_сжат[n] = 0;
        x_наиб[n] = x[i][n];
        x_2_по_велечене[n] = x[i - 1][n];
        x_наим[n] = x[0][n];
    }
    f = 0;

    //поиск центра тяжести
    for (n = 0; n < i; n++)
        for (int m = 0; m < i; m++)
            центр_тяж[n] = центр_тяж[n] + (x[m][n]);
    центр_тяж[0] = центр_тяж[0] / i; центр_тяж[1] = центр_тяж[1] / i;

    for (n = 0; n < i; n++)
        отр[n] = центр_тяж[n] + a * (центр_тяж[n] - x_наиб[n]);

    if (F(отр) <= F(x_наим))
    {
        for (n = 0; n < i; n++)
            отр_рас[n] = центр_тяж[n] + y * (отр[n] - центр_тяж[n]);
        double x7 = F(отр_рас);

        if (F(отр_рас) < F(отр))
            for (n = 0; n < i; n++)
                x_наиб[n] = отр_рас[n];
        else
            for (n = 0; n < i; n++)
                x_наиб[n] = отр[n];
    }
}

```

```

else if (F(x_наим) <= F(отр) && F(отр) <= F(x_2_по_вечечене))
{
    for (n = 0; n < i; n++)
        x_наиб[n] = отр[n];
}
else if (F(x_2_по_вечечене) <= F(отр) && F(отр) <= F(x_наиб))
{
    for (n = 0; n < i; n++)
    {
        Temp[n] = отр[n];
        отр[n] = x_наиб[n];
        x_наиб[n] = Temp[n];
        f = 1;
    }
}
else if (F(x_наиб) <= F(отр))
    f = 1;
if (f) {
    for (n = 0; n < i; n++)
        отр_сжат[n] = центр_тяж[n] + b * (x_наиб[n] - центр_тяж[n]);

    if (F(отр_сжат) < F(x_наиб))
        for (n = 0; n < i; n++)
            x_наиб[n] = отр_сжат[n];

    if (F(отр_сжат) >= F(x_наиб))
    {
        for (n = 0; n < i; n++)
        {
            x[i][n] = x_наиб[n];
            x[i - 1][n] = x_2_по_вечечене[n];
            x[0][n] = x_наим[n];
        }
        сорт(x);          //сортировка точек
        for (n = 0; n < i; n++)
        {
            x_наиб[n] = x[i][n];
            x_2_по_вечечене[n] = x[i - 1][n];
            x_наим[n] = x[0][n];
        }
        for (n = 0; n < i; n++)
        {
            x_наиб[n] = (x_наиб[n] + x_наим[n]) / 2;
            x_2_по_вечечене[n] = (x_2_по_вечечене[n] + x_наим[n]) / 2;
        }
    }
}
for (n = 0; n < i; n++)
{
    x[i][n] = x_наиб[n];
    x[i - 1][n] = x_2_по_вечечене[n];
    x[0][n] = x_наим[n];
}

```



```
}
```

```
Таблица(x, x_наим, x_2_по_вечечене, x_наиб , центр_тяж, отр , отр_рас, отр_сжат, k);
```

```
} while (fabs(D(x_наиб, x_2_по_вечечене, x_наим, центр_тяж)) > Eps);
```

```
сорт(x);      //сортировка точек
```

```
cout << char(179) << " k = " << left << setfill(char(32)) << setw(17) << k << right  
<< "x* = (" << setfill(char(32)) << setw(7) << round(x[0][0] * 100) / 100  
<< " : " << left << setfill(char(32)) << setw(7) << round(x[0][1] * 100) / 100 << right  
<< ")    "<< "f(x*) = " << left << setfill(char(32)) << setw(21)  
<< round(F(x[0]) * 100) / 100 << right<< setfill(char(32)) << setw(74)  
<< char(179) << endl;  
cout << char(192) << setfill(char(196)) << setw(154) << char(217) << "\n\n";  
system("pause");
```

```
}
```

```
void сорт(double x[j][i])
```

```
{
```

```
    double Min;                //минимальный элемент  
    int jMin;                  //индекс минимального элемента  
    int iSort;                 //граница отсортированной области  
    int k;                     //индексная переменная  
    double Temp[i];            //обменная переменная
```

```
    for (iSort = 0; iSort < j - 1; iSort++)
```

```
    {
```

```
        //первый элемент из неупорядоченных назначаем минимальным  
        Min = F(x[iSort]);      //минимум  
        jMin = iSort;           //его индекс
```

```
        //ищем минимальный элемент в оставшейся части массива
```

```
        for (k = iSort + 1; k < j; k++)
```

```
        {
```

```
            if (F(x[k]) < Min)    //очередной кандидат на минимальный  
            {
```

```
                //запоминаем минимальный элемент и его номер
```

```
                Min = F(x[k]);
```

```
                jMin = k;
```

```
            }//if
```

```
        }//for j
```

```
        //нашли минимум в неупорядоченной части массива
```

```
        //ставим его на место первого в неупорядоченной части массива
```

```
        //меняем элементы местами
```

```
        for (k = 0; k < i; k++)
```

```
        {
```

```
            Temp[k] = x[iSort][k];
```

```
            x[iSort][k] = x[jMin][k];
```

```
            x[jMin][k] = Temp[k];
```

```

    }
    } //for iSort
}

void Таблица(double x[j][i], double  x_наим[2], double  x_2_по_вечечене[2], double
x_наиб[2], double  центр_тяж[2], double отр[2], double отр_рас[2], double
отр_сжат[2], int k)
{
    cout << char(179) << " k = " << left << setfill(char(32)) << setw(17) << k << right
    << "x1 = (" << setfill(char(32)) << setw(7) << round(x[0][0] * 100) / 100 << " : "
    << left << setfill(char(32)) << setw(7) << round(x[0][1] * 100) / 100 << right << ") "
    << "x2 = (" << setfill(char(32)) << setw(7) << round(x[1][0] * 100) / 100 << " : "
    << left << setfill(char(32)) << setw(7) << round(x[1][1] * 100) / 100 << right << ") "
    << "x3 = (" << setfill(char(32)) << setw(7) << round(x[2][0] * 100) / 100 << " : "
    << left << setfill(char(32)) << setw(7) << round(x[2][1] * 100) / 100 << right << ") "
    << setfill(char(32)) << setw(45) << char(179) << endl;

    cout << char(179) << setfill(char(32)) << setw(30) << "f(x1) = " << left
    << setfill(char(32)) << setw(21) << round(F(x_наим) * 100) / 100 << right
    << "f(x2) = " << left << setfill(char(32)) << setw(21)
    << round(F(x_2_по_вечечене) * 100) / 100 << right << "f(x3) = "
    << left << setfill(char(32)) << setw(21) << round(F(x_наиб) * 100) / 100 << right
    << setfill(char(32)) << setw(45) << char(179) << endl;

    cout << char(195);
    cout << setfill(char(196)) << setw(22);
    for (int i = 0; i < 6; i++)
    {
        cout << char(194) << setfill(char(196)) << setw(22);
    }
    cout << char(180) << endl;

    cout << char(179) << "      xl      "
    << char(179) << "      xg      "
    << char(179) << "      xh      "
    << char(179) << "      xc      "
    << char(179) << "      xr      "
    << char(179) << "      xe      "
    << char(179) << "      xs      "
    << char(179) << endl;

    cout << char(179) << "      f(xl)      "
    << char(179) << "      f(xg)      "
    << char(179) << "      f(xh)      "
    << char(179) << "      f(xc)      "
    << char(179) << "      f(xr)      "
    << char(179) << "      f(xe)      "
    << char(179) << "      f(xs)      "
    << char(179) << endl;

    cout << char(195);
    cout << setfill(char(196)) << setw(22);

```

```

for (int i = 0; i < 6; i++)
{
    cout << char(197) << setfill(char(196)) << setw(22);
}
cout << char(180) << endl;
/*
    double* arrays[] = { x_наим, x_2_по_велечение, x_наиб, центр_тяж, отр,
отр_рас, отр_сжат };
    int arrays_size = sizeof(arrays)/sizeof(arrays[0]);*/

    /*for (int i = 0; i < arrays_size; ++i) {
        cout << char(179) << " ("
            << fixed << setprecision(2) << arrays[i][0] << "; " << left
            << arrays[i][1] << right << ") ";
    }
    cout << endl;*/

    cout << char(179) << " (" << setfill(char(32)) << setw(7) << round(x_наим[0] * 100) /
100 << "; " << left << setfill(char(32)) << setw(7) << round(x_наим[1] * 100) / 100 << right
<< ") ";
    cout << char(179) << " (" << setfill(char(32)) << setw(7) << round(x_2_по_велечение[0]
* 100) / 100 << "; " << left << setfill(char(32)) << setw(7) << round(x_2_по_велечение[1] *
100) / 100 << right << ") ";
    cout << char(179) << " (" << setfill(char(32)) << setw(7) << round(x_наиб[0] * 100) /
100 << "; " << left << setfill(char(32)) << setw(7) << round(x_наиб[1] * 100) / 100 << right
<< ") ";
    cout << char(179) << " (" << setfill(char(32)) << setw(7) << round(центр_тяж[0] * 100)
/ 100 << "; " << left << setfill(char(32)) << setw(7) << round(центр_тяж[1] * 100) / 100 <<
right << ") ";
    cout << char(179) << " (" << setfill(char(32)) << setw(7) << round(отр[0] * 100) / 100
<< "; " << left << setfill(char(32)) << setw(7) << round(отр[1] * 100) / 100 << right << ") ";
    cout << char(179) << " (" << setfill(char(32)) << setw(7) << round(отр_рас[0] * 100) /
100 << "; " << left << setfill(char(32)) << setw(7) << round(отр_рас[1] * 100) / 100 << right
<< ") ";
    cout << char(179) << " (" << setfill(char(32)) << setw(7) << round(отр_сжат[0] * 100) /
100 << "; " << left << setfill(char(32)) << setw(7) << round(отр_сжат[1] * 100) / 100 << right
<< ") ";
    cout << char(179) << endl;

    cout << char(179) << setfill(char(32)) << setw(13) << round(F(x_наим) * 100) / 100 <<
" ";
    cout << char(179) << setfill(char(32)) << setw(13) << round(F(x_2_по_велечение) *
100) / 100 << " ";
    cout << char(179) << setfill(char(32)) << setw(13) << round(F(x_наиб) * 100) / 100 <<
" ";
    cout << char(179) << setfill(char(32)) << setw(13) << round(F(центр_тяж) * 100) / 100
<< " ";
    cout << char(179) << setfill(char(32)) << setw(13) << round(F(отр) * 100) / 100 << "
";
    cout << char(179) << setfill(char(32)) << setw(13) << round(F(отр_рас) * 100) / 100 <<
" ";

```

```

    cout << char(179) << setfill(char(32)) << setw(13) << round(F(отр_сжат) * 100) / 100
<< "
    ";
    cout << char(179) << endl;

    cout << char(195);
    cout << setfill(char(196)) << setw(22);
    for (int i = 0; i < 6; i++)
    {
        cout << char(193) << setfill(char(196)) << setw(22);
    }
    cout << char(180) << endl;
}

```

# Результаты работы программ

Введите Eps : 0.01  
Введите коэффициент отражения : 1  
Введите коэффициент сжатия : 0.5  
Введите коэффициент растяжения : 2

k = 1						
x1 = ( -1 : -1 ) f(x1) = 2		x2 = ( 0 : -1 ) f(x2) = 3		x3 = ( 1 : 0 ) f(x3) = 2		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( -1 ; -1 ) 2	( 0 ; -1 ) 3	( 1 ; 0 ) 2	( -0.5 ; -1 ) 2.25	( 1 ; 0 ) 2	( 2.5 ; 1 ) 5.25	( 0 ; 0 ) -0
k = 2						
x1 = ( -1 : -1 ) f(x1) = 2		x2 = ( 1 : 0 ) f(x2) = 2		x3 = ( 0 : 0.5 ) f(x3) = -0.75		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( -1 ; -1 ) 2	( 1 ; 0 ) 2	( 0 ; 0.5 ) -0.75	( 0 ; -0.5 ) 1.25	( 0 ; 0 ) -0	( 0 ; 0.5 ) -0.75	( 0 ; 0 ) -0
k = 3						
x1 = ( 0 : 0.5 ) f(x1) = -0.75		x2 = ( -0.13 : 0.06 ) f(x2) = -0.22		x3 = ( 0.5 : 0.25 ) f(x3) = 0.19		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( 0 ; 0.5 ) -0.75	( -0.13 ; 0.06 ) -0.22	( 0.5 ; 0.25 ) 0.19	( 0.5 ; 0.25 ) 0.19	( 2 ; 1.5 ) 2.25	( 0 ; 0 ) -0	( -0.25 ; -0.38 ) 0.61
k = 4						
x1 = ( 0 : 0.5 ) f(x1) = -0.75		x2 = ( -0.13 : 0.06 ) f(x2) = -0.22		x3 = ( -0.63 : 0.31 ) f(x3) = -0.57		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( 0 ; 0.5 ) -0.75	( -0.13 ; 0.06 ) -0.22	( -0.63 ; 0.31 ) -0.57	( -0.06 ; 0.28 ) -0.52	( -0.63 ; 0.31 ) -0.57	( 0 ; 0 ) -0	( 0 ; 0 ) -0
k = 5						
x1 = ( 0 : 0.5 ) f(x1) = -0.75		x2 = ( -0.63 : 0.31 ) f(x2) = -0.57		x3 = ( -0.5 : 0.75 ) f(x3) = -0.81		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( 0 ; 0.5 ) -0.75	( -0.63 ; 0.31 ) -0.57	( -0.5 ; 0.75 ) -0.81	( -0.31 ; 0.41 ) -0.74	( -0.5 ; 0.75 ) -0.81	( -0.69 ; 1.09 ) -0.45	( 0 ; 0 ) -0
k = 6						
x1 = ( -0.5 : 0.75 ) f(x1) = -0.81		x2 = ( 0 : 0.5 ) f(x2) = -0.75		x3 = ( 0.13 : 0.94 ) f(x3) = -0.97		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( -0.5 ; 0.75 ) -0.81	( 0 ; 0.5 ) -0.75	( 0.13 ; 0.94 ) -0.97	( -0.25 ; 0.63 ) -0.89	( 0.13 ; 0.94 ) -0.97	( 0.5 ; 1.25 ) -0.81	( 0 ; 0 ) -0
k = 7						
x1 = ( 0.13 : 0.94 ) f(x1) = -0.97		x2 = ( -0.08 : 0.98 ) f(x2) = -1		x3 = ( -0.19 : 0.84 ) f(x3) = -0.97		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( 0.13 ; 0.94 ) -0.97	( -0.08 ; 0.98 ) -1	( -0.19 ; 0.84 ) -0.97	( -0.19 ; 0.84 ) -0.97	( 0 ; 0.5 ) -0.75	( 0 ; 0 ) -0	( -0.28 ; 1.02 ) -0.92
k = 8						
x1 = ( -0.08 : 0.98 ) f(x1) = -1		x2 = ( -0.08 : 0.94 ) f(x2) = -0.99		x3 = ( 0.02 : 0.96 ) f(x3) = -1		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( -0.08 ; 0.98 ) -1	( -0.08 ; 0.94 ) -0.99	( 0.02 ; 0.96 ) -1	( 0.02 ; 0.96 ) -1	( 0.23 ; 1.07 ) -0.96	( 0 ; 0 ) -0	( -0.08 ; 0.9 ) -0.99
k = 8						
x* = ( 0.02 : 0.96 )		f(x*) = -1				

Для продолжения нажмите любую клавишу . . .

Введите Eps : 0.1  
Введите коэффициент отражения : 1  
Введите коэффициент сжатия : 0.5  
Введите коэффициент растяжения : 2

Введите коэффициент отражения : 1  
Введите коэффициент сжатия : 0.5  
Введите коэффициент растяжения : 2

Введите коэффициент сжатия : 0.5  
Введите коэффициент растяжения : 2

Введите коэффициент растяжения : 2

k = 1						
x1 = (      -1 : -1      ) f(x1) = 2		x2 = (      0 : -1      ) f(x2) = 3		x3 = (      1 : 0      ) f(x3) = 2		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
(      -1 ; -1      ) 2	(      0 ; -1      ) 3	(      1 ; 0      ) 2	(    -0.5 ; -1    ) 2.25	(      1 ; 0      ) 2	(    2.5 ; 1    ) 5.25	(      0 ; 0      ) -0
k = 2						
x1 = (      -1 : -1      ) f(x1) = 2		x2 = (      1 : 0      ) f(x2) = 2		x3 = (      0 : 0.5      ) f(x3) = -0.75		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
(    -1 ; -1    ) 2	(      1 ; 0      ) 2	(      0 ; 0.5      ) -0.75	(      0 ; -0.5      ) 1.25	(      0 ; 0      ) -0	(      0 ; 0.5      ) -0.75	(      0 ; 0      ) -0
k = 3						
x1 = (      0 : 0.5      ) f(x1) = -0.75		x2 = (    -0.13 : 0.06    ) f(x2) = -0.22		x3 = (    0.5 : 0.25    ) f(x3) = 0.19		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
(      0 ; 0.5      ) -0.75	(    -0.13 ; 0.06    ) -0.22	(    0.5 ; 0.25    ) 0.19	(    0.5 ; 0.25    ) 0.19	(      2 ; 1.5      ) 2.25	(      0 ; 0      ) -0	(    -0.25 ; -0.38    ) 0.61
k = 4						
x1 = (      0 : 0.5      ) f(x1) = -0.75		x2 = (    -0.13 : 0.06    ) f(x2) = -0.22		x3 = (    -0.63 : 0.31    ) f(x3) = -0.57		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
(      0 ; 0.5      ) -0.75	(    -0.13 ; 0.06    ) -0.22	(    -0.63 ; 0.31    ) -0.57	(    -0.06 ; 0.28    ) -0.52	(    -0.63 ; 0.31    ) -0.57	(      0 ; 0      ) -0	(      0 ; 0      ) -0
k = 5						
x1 = (      0 : 0.5      ) f(x1) = -0.75		x2 = (    -0.63 : 0.31    ) f(x2) = -0.57		x3 = (    -0.5 : 0.75    ) f(x3) = -0.81		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
(      0 ; 0.5      ) -0.75	(    -0.63 ; 0.31    ) -0.57	(    -0.5 ; 0.75    ) -0.81	(    -0.31 ; 0.41    ) -0.74	(    -0.5 ; 0.75    ) -0.81	(    -0.69 ; 1.09    ) -0.45	(      0 ; 0      ) -0
k = 6						
x1 = (    -0.5 : 0.75    ) f(x1) = -0.81		x2 = (      0 : 0.5      ) f(x2) = -0.75		x3 = (    0.13 : 0.94    ) f(x3) = -0.97		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
(    -0.5 ; 0.75    ) -0.81	(      0 ; 0.5      ) -0.75	(    0.13 ; 0.94    ) -0.97	(    -0.25 ; 0.63    ) -0.89	(    0.13 ; 0.94    ) -0.97	(    0.5 ; 1.25    ) -0.81	(      0 ; 0      ) -0
k = 7						
x1 = (    0.13 : 0.94    ) f(x1) = -0.97		x2 = (    -0.08 : 0.98    ) f(x2) = -1		x3 = (    -0.19 : 0.84    ) f(x3) = -0.97		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
(    0.13 ; 0.94    ) -0.97	(    -0.08 ; 0.98    ) -1	(    -0.19 ; 0.84    ) -0.97	(    -0.19 ; 0.84    ) -0.97	(      0 ; 0.5      ) -0.75	(      0 ; 0      ) -0	(    -0.28 ; 1.02    ) -0.92
k = 7						
x* = (    -0.08 : 0.98    )		f(x*) = -1				

## Другой начальный набор точек

Введите Eps : 0.01  
 Введите коэффициент отражения : 1  
 Введите коэффициент сжатия : 0.5  
 Введите коэффициент растяжения : 2

k = 1						
x1 = ( -10 : -10 ) f(x1) = 110		x2 = ( 0 : -10 ) f(x2) = 120		x3 = ( 10 : 0 ) f(x3) = 110		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( -10 ; -10 110 )	( 0 ; -10 120 )	( 10 ; 0 110 )	( -5 ; -10 90 )	( 10 ; 0 110 )	( 10 ; 0 110 )	( 0 ; 0 -0 )
k = 2						
x1 = ( -10 : -10 ) f(x1) = 110		x2 = ( 10 : 0 ) f(x2) = 110		x3 = ( 0 : 0 ) f(x3) = -0		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( -10 ; -10 110 )	( 10 ; 0 110 )	( 0 ; 0 -0 )	( 0 ; -5 35 )	( 0 ; 0 -0 )	( 0 ; 0 -0 )	( 0 ; 0 -0 )
k = 3						
x1 = ( 0 : 0 ) f(x1) = -0		x2 = ( -1.25 : -2.5 ) f(x2) = 8.44		x3 = ( 5 : 0 ) f(x3) = 30		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( 0 ; 0 -0 )	( -1.25 ; -2.5 8.44 )	( 5 ; 0 30 )	( 5 ; 0 30 )	( 20 ; 10 300 )	( 0 ; 0 -0 )	( -2.5 ; -5 26.25 )
k = 4						
x1 = ( 0 : 0 ) f(x1) = -0		x2 = ( -0.63 : -1.25 ) f(x2) = 3.05		x3 = ( -1.72 : -0.94 ) f(x3) = 2.38		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( 0 ; 0 -0 )	( -0.63 ; -1.25 3.05 )	( -1.72 ; -0.94 2.38 )	( -0.63 ; -1.25 3.05 )	( 5 ; 0 30 )	( 0 ; 0 -0 )	( -3.44 ; -1.88 9.2 )
k = 5						
x1 = ( 0 : 0 ) f(x1) = -0		x2 = ( -1.72 : -0.94 ) f(x2) = 2.38		x3 = ( -1.09 : 0.31 ) f(x3) = -0.08		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( 0 ; 0 -0 )	( -1.72 ; -0.94 2.38 )	( -1.09 ; 0.31 -0.08 )	( -0.86 ; -0.47 0.63 )	( -1.09 ; 0.31 -0.08 )	( -1.09 ; 0.31 -0.08 )	( 0 ; 0 -0 )
k = 6						
x1 = ( -1.09 : 0.31 ) f(x1) = -0.08		x2 = ( 0 : 0 ) f(x2) = -0		x3 = ( 0.63 : 1.25 ) f(x3) = -0.7		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( -1.09 ; 0.31 -0.08 )	( 0 ; 0 -0 )	( 0.63 ; 1.25 -0.7 )	( -0.55 ; 0.16 -0.45 )	( 0.63 ; 1.25 -0.7 )	( 0.63 ; 1.25 -0.7 )	( 0 ; 0 -0 )
k = 7						
x1 = ( 0.63 : 1.25 ) f(x1) = -0.7		x2 = ( -1.09 : 0.31 ) f(x2) = -0.08		x3 = ( -0.47 : 1.56 ) f(x3) = -0.2		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( 0.63 ; 1.25 -0.7 )	( -1.09 ; 0.31 -0.08 )	( -0.47 ; 1.56 -0.2 )	( -0.23 ; 0.78 -0.95 )	( -0.47 ; 1.56 -0.2 )	( 0 ; 0 -0 )	( 0 ; 0 -0 )
k = 8						
x1 = ( -0.51 : 0.86 ) f(x1) = -0.79		x2 = ( 0.06 : 1.05 ) f(x2) = -1		x3 = ( -0.49 : 1.21 ) f(x3) = -0.61		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( -0.51 ; 0.86 -0.79 )	( 0.06 ; 1.05 -1 )	( -0.49 ; 1.21 -0.61 )	( 0.08 ; 1.41 -0.86 )	( 1.25 ; 2.5 0.94 )	( 0 ; 0 -0 )	( -0.51 ; 0.86 -0.79 )
k = 9						
x1 = ( 0.06 : 1.05 ) f(x1) = -1		x2 = ( -0.51 : 0.86 ) f(x2) = -0.79		x3 = ( 0.04 : 0.7 ) f(x3) = -0.9		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( 0.06 ; 1.05 -1 )	( -0.51 ; 0.86 -0.79 )	( 0.04 ; 0.7 -0.9 )	( -0.22 ; 0.96 -0.96 )	( 0.04 ; 0.7 -0.9 )	( 0 ; 0 -0 )	( 0 ; 0 -0 )
k = 10						
x1 = ( 0.06 : 1.05 ) f(x1) = -1		x2 = ( -0.09 : 0.96 ) f(x2) = -0.99		x3 = ( 0.05 : 0.88 ) f(x3) = -0.98		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( 0.06 ; 1.05 -1 )	( -0.09 ; 0.96 -0.99 )	( 0.05 ; 0.88 -0.98 )	( 0.05 ; 0.88 -0.98 )	( 0.61 ; 0.9 -0.56 )	( 0 ; 0 -0 )	( -0.23 ; 0.87 -0.96 )
k = 11						
x1 = ( 0.06 : 1.05 ) f(x1) = -1		x2 = ( 0.04 : 1 ) f(x2) = -1		x3 = ( -0.01 : 1.01 ) f(x3) = -1		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( 0.06 ; 1.05 -1 )	( 0.04 ; 1 -1 )	( -0.01 ; 1.01 -1 )	( -0.01 ; 1.01 -1 )	( -0.08 ; 1.14 -0.96 )	( 0 ; 0 -0 )	( 0.02 ; 0.94 -1 )
k = 11						
x* = ( -0.01 : 1.01 )		f(x*) = -1				

C:\ПРОГРАММЫ\ЗС\_ТОПУ2\Debug\ЗС\_ТОПУ2.exe

Введите Eps : 0.1

Введите коэффициент отражения : 1


Введите коэффициент сжатия : 0.5

Введите коэффициент растяжения : 2

k = 1						
x1 = ( -10 : -10 ) f(x1) = 110		x2 = ( 0 : -10 ) f(x2) = 120		x3 = ( 10 : 0 ) f(x3) = 110		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( -10 ; -10 ) 110	( 0 ; -10 ) 120	( 10 ; 0 ) 110	( -5 ; -10 ) 90	( 10 ; 0 ) 110	( 25 ; 10 ) 480	( 0 ; 0 ) -0
k = 2						
x1 = ( -10 : -10 ) f(x1) = 110		x2 = ( 10 : 0 ) f(x2) = 110		x3 = ( 0 : 0 ) f(x3) = -0		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( -10 ; -10 ) 110	( 10 ; 0 ) 110	( 0 ; 0 ) -0	( 0 ; -5 ) 35	( 0 ; 0 ) -0	( 0 ; 5 ) 15	( 0 ; 0 ) -0
k = 3						
x1 = ( 0 : 0 ) f(x1) = -0		x2 = ( -1.25 : -2.5 ) f(x2) = 8.44		x3 = ( 5 : 0 ) f(x3) = 30		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( 0 ; 0 ) -0	( -1.25 ; -2.5 ) 8.44	( 5 ; 0 ) 30	( 5 ; 0 ) 30	( 20 ; 10 ) 300	( 0 ; 0 ) -0	( -2.5 ; -5 ) 26.25
k = 4						
x1 = ( 0 : 0 ) f(x1) = -0		x2 = ( -0.63 : -1.25 ) f(x2) = 3.05		x3 = ( -1.72 : -0.94 ) f(x3) = 2.38		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( 0 ; 0 ) -0	( -0.63 ; -1.25 ) 3.05	( -1.72 ; -0.94 ) 2.38	( -0.63 ; -1.25 ) 3.05	( 5 ; 0 ) 30	( 0 ; 0 ) -0	( -3.44 ; -1.88 ) 9.2
k = 5						
x1 = ( 0 : 0 ) f(x1) = -0		x2 = ( -1.72 : -0.94 ) f(x2) = 2.38		x3 = ( -1.09 : 0.31 ) f(x3) = -0.08		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( 0 ; 0 ) -0	( -1.72 ; -0.94 ) 2.38	( -1.09 ; 0.31 ) -0.08	( -0.86 ; -0.47 ) 0.63	( -1.09 ; 0.31 ) -0.08	( -1.33 ; 1.09 ) 0.9	( 0 ; 0 ) -0
k = 6						
x1 = ( -1.09 : 0.31 ) f(x1) = -0.08		x2 = ( 0 : 0 ) f(x2) = -0		x3 = ( 0.63 : 1.25 ) f(x3) = -0.7		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( -1.09 ; 0.31 ) -0.08	( 0 ; 0 ) -0	( 0.63 ; 1.25 ) -0.7	( -0.55 ; 0.16 ) -0.45	( 0.63 ; 1.25 ) -0.7	( 1.8 ; 2.34 ) 1.62	( 0 ; 0 ) -0
k = 7						
x1 = ( 0.63 : 1.25 ) f(x1) = -0.7		x2 = ( -1.09 : 0.31 ) f(x2) = -0.08		x3 = ( -0.47 : 1.56 ) f(x3) = -0.2		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( 0.63 ; 1.25 ) -0.7	( -1.09 ; 0.31 ) -0.08	( -0.47 ; 1.56 ) -0.2	( -0.23 ; 0.78 ) -0.95	( -0.47 ; 1.56 ) -0.2	( 0 ; 0 ) -0	( 0 ; 0 ) -0
k = 8						
x1 = ( -0.51 : 0.86 ) f(x1) = -0.79		x2 = ( 0.06 : 1.05 ) f(x2) = -1		x3 = ( -0.49 : 1.21 ) f(x3) = -0.61		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( -0.51 ; 0.86 ) -0.79	( 0.06 ; 1.05 ) -1	( -0.49 ; 1.21 ) -0.61	( 0.08 ; 1.41 ) -0.86	( 1.25 ; 2.5 ) 0.94	( 0 ; 0 ) -0	( -0.51 ; 0.86 ) -0.79
k = 9						
x1 = ( 0.06 : 1.05 ) f(x1) = -1		x2 = ( -0.51 : 0.86 ) f(x2) = -0.79		x3 = ( 0.04 : 0.7 ) f(x3) = -0.9		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( 0.06 ; 1.05 ) -1	( -0.51 ; 0.86 ) -0.79	( 0.04 ; 0.7 ) -0.9	( -0.22 ; 0.96 ) -0.96	( 0.04 ; 0.7 ) -0.9	( 0 ; 0 ) -0	( 0 ; 0 ) -0
k = 10						
x1 = ( 0.06 : 1.05 ) f(x1) = -1		x2 = ( -0.09 : 0.96 ) f(x2) = -0.99		x3 = ( 0.05 : 0.88 ) f(x3) = -0.98		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( 0.06 ; 1.05 ) -1	( -0.09 ; 0.96 ) -0.99	( 0.05 ; 0.88 ) -0.98	( 0.05 ; 0.88 ) -0.98	( 0.61 ; 0.9 ) -0.56	( 0 ; 0 ) -0	( -0.23 ; 0.87 ) -0.96
k = 10						
x* = ( 0.06 : 1.05 )		f(x*) = -1				



## Изменение коэффициентов

 C:\ПРОГРАММЫ\ЗС\_ТОПУ2\Debug\ЗС\_ТОПУ2.exe

Введите Eps : 0.1  
Введите коэффициент отражения : 1.5  
Введите коэффициент сжатия : 0.5  
Введите коэффициент растяжения : 2

k = 1						
x1 = ( 0.63 : -0.25 ) f(x1) = 1.73		x2 = ( -0.19 : -0.63 ) f(x2) = 1.37		x3 = ( 0.31 : -0.63 ) f(x3) = 2.25		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( 0.63 ; -0.25 ) 1.73	( -0.19 ; -0.63 ) 1.37	( 0.31 ; -0.63 ) 2.25	( -0.5 ; -1 ) 2.25	( -2 ; -2 ) 6	( 0 ; 0 ) -0	( 0.63 ; -0.25 ) 1.73
k = 2						
x1 = ( -0.19 : -0.63 ) f(x1) = 1.37		x2 = ( 0.63 : -0.25 ) f(x2) = 1.73		x3 = ( -0.06 : 0.13 ) f(x3) = -0.29		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( -0.19 ; -0.63 ) 1.37	( 0.63 ; -0.25 ) 1.73	( -0.06 ; 0.13 ) -0.29	( 0.22 ; -0.44 ) 1.43	( 0.08 ; -0.16 ) 0.43	( -0.06 ; 0.13 ) -0.29	( 0 ; 0 ) -0
k = 3						
x1 = ( -0.06 : 0.13 ) f(x1) = -0.29		x2 = ( -0.19 : -0.63 ) f(x2) = 1.37		x3 = ( -1.25 : -0.25 ) f(x3) = 0.56		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( -0.06 ; 0.13 ) -0.29	( -0.19 ; -0.63 ) 1.37	( -1.25 ; -0.25 ) 0.56	( -0.13 ; -0.25 ) 0.42	( -1.25 ; -0.25 ) 0.56	( 0 ; 0 ) -0	( 0 ; 0 ) -0
k = 4						
x1 = ( -0.06 : 0.13 ) f(x1) = -0.29		x2 = ( -0.54 : 0.24 ) f(x2) = -0.54		x3 = ( -0.66 : -0.06 ) f(x3) = -0.14		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( -0.06 ; 0.13 ) -0.29	( -0.54 ; 0.24 ) -0.54	( -0.66 ; -0.06 ) -0.14	( -0.66 ; -0.06 ) -0.14	( -0.19 ; -0.63 ) 1.37	( 0 ; 0 ) -0	( -1.01 ; 0.36 ) -0.22
k = 5						
x1 = ( -0.54 : 0.24 ) f(x1) = -0.54		x2 = ( -0.06 : 0.13 ) f(x2) = -0.29		x3 = ( 0.24 : 0.55 ) f(x3) = -0.64		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( -0.54 ; 0.24 ) -0.54	( -0.06 ; 0.13 ) -0.29	( 0.24 ; 0.55 ) -0.64	( -0.3 ; 0.18 ) -0.49	( 0.24 ; 0.55 ) -0.64	( 0.77 ; 0.92 ) -0.34	( 0 ; 0 ) -0
k = 6						
x1 = ( 0.24 : 0.55 ) f(x1) = -0.64		x2 = ( -0.54 : 0.24 ) f(x2) = -0.54		x3 = ( -0.28 : 0.81 ) f(x3) = -0.94		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( 0.24 ; 0.55 ) -0.64	( -0.54 ; 0.24 ) -0.54	( -0.28 ; 0.81 ) -0.94	( -0.15 ; 0.4 ) -0.7	( -0.28 ; 0.81 ) -0.94	( -0.41 ; 1.21 ) -0.7	( 0 ; 0 ) -0
k = 7						
x1 = ( -0.28 : 0.81 ) f(x1) = -0.94		x2 = ( 0.04 : 0.91 ) f(x2) = -0.99		x3 = ( -0.02 : 0.68 ) f(x3) = -0.9		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( -0.28 ; 0.81 ) -0.94	( 0.04 ; 0.91 ) -0.99	( -0.02 ; 0.68 ) -0.9	( -0.02 ; 0.68 ) -0.9	( -0.54 ; 0.24 ) -0.54	( 0 ; 0 ) -0	( 0.37 ; 1.01 ) -0.87
k = 7						
x* = ( 0.04 : 0.91 )		f(x*) = -0.99				

C:\ПРОГРАММЫ\3C\_ТОПУ2\Debug\3C\_ТОПУ2.exe

Введите Eps : 0.1

Введите коэффициент отражения : 1

Введите коэффициент сжатия : 0.1  
Введите коэффициент растяжения : 2

$k = 1$	$x1 = ( \quad -1 : -1 \quad )$ $f(x1) = 2$	$x2 = ( \quad \quad \quad 0 : -1 \quad )$ $f(x2) = 3$	$x3 = ( \quad \quad \quad 1 : 0 \quad )$ $f(x3) = 2$			
	$x1$ $f(x1)$	$xg$ $f(xg)$	$xh$ $f(xh)$	$xc$ $f(xc)$	$xr$ $f(xr)$	$xe$ $f(xe)$
	$( \quad -1 ; -1 \quad )$ $2$	$( \quad \quad \quad 0 ; \quad -1 \quad )$ $3$	$( \quad \quad \quad 1 ; \quad 0 \quad )$ $2$	$( \quad -0.5 ; -1 \quad )$ $2.25$	$( \quad \quad \quad 1 ; \quad 0 \quad )$ $2$	$( \quad 2.5 ; 1 \quad )$ $5.25$
$k = 2$	$x1 = ( \quad -1 : -1 \quad )$ $f(x1) = 2$	$x2 = ( \quad \quad \quad 1 : 0 \quad )$ $f(x2) = 2$	$x3 = ( \quad \quad \quad 0 : 0.5 \quad )$ $f(x3) = -0.75$			
	$x1$ $f(x1)$	$xg$ $f(xg)$	$xh$ $f(xh)$	$xc$ $f(xc)$	$xr$ $f(xr)$	$xe$ $f(xe)$
	$( \quad -1 ; -1 \quad )$ $2$	$( \quad \quad \quad 1 ; \quad 0 \quad )$ $2$	$( \quad \quad \quad 0 ; 0.5 \quad )$ $-0.75$	$( \quad \quad \quad 0 ; -0.5 \quad )$ $1.25$	$( \quad \quad \quad 0 ; \quad 0 \quad )$ $-0$	$( \quad \quad \quad 0 ; 0.5 \quad )$ $-0.75$
$k = 3$	$x1 = ( \quad \quad \quad 0 : 0.5 \quad )$ $f(x1) = -0.75$	$x2 = ( \quad 0.18 : 0.31 \quad )$ $f(x2) = -0.38$	$x3 = ( \quad \quad \quad 0.5 : 0.25 \quad )$ $f(x3) = 0.19$			
	$x1$ $f(x1)$	$xg$ $f(xg)$	$xh$ $f(xh)$	$xc$ $f(xc)$	$xr$ $f(xr)$	$xe$ $f(xe)$
	$( \quad \quad \quad 0 ; 0.5 \quad )$ $-0.75$	$( \quad 0.18 ; 0.31 \quad )$ $-0.38$	$( \quad 0.5 ; 0.25 \quad )$ $0.19$	$( \quad 0.5 ; 0.25 \quad )$ $0.19$	$( \quad \quad \quad 2 ; 1.5 \quad )$ $2.25$	$( \quad \quad \quad 0 ; \quad 0 \quad )$ $-0$
$k = 4$	$x1 = ( \quad \quad \quad 0 : 0.5 \quad )$ $f(x1) = -0.75$	$x2 = ( \quad 0.18 : 0.31 \quad )$ $f(x2) = -0.38$	$x3 = ( \quad -0.32 : 0.56 \quad )$ $f(x3) = -0.85$			
	$x1$ $f(x1)$	$xg$ $f(xg)$	$xh$ $f(xh)$	$xc$ $f(xc)$	$xr$ $f(xr)$	$xe$ $f(xe)$
	$( \quad \quad \quad 0 ; 0.5 \quad )$ $-0.75$	$( \quad 0.18 ; 0.31 \quad )$ $-0.38$	$( \quad -0.32 ; 0.56 \quad )$ $-0.85$	$( \quad 0.09 ; 0.41 \quad )$ $-0.59$	$( \quad -0.32 ; 0.56 \quad )$ $-0.85$	$( \quad -0.74 ; 0.72 \quad )$ $-0.58$
$k = 5$	$x1 = ( \quad -0.32 : 0.56 \quad )$ $f(x1) = -0.85$	$x2 = ( \quad \quad \quad 0 : 0.5 \quad )$ $f(x2) = -0.75$	$x3 = ( \quad -0.5 : 0.75 \quad )$ $f(x3) = -0.81$			
	$x1$ $f(x1)$	$xg$ $f(xg)$	$xh$ $f(xh)$	$xc$ $f(xc)$	$xr$ $f(xr)$	$xe$ $f(xe)$
	$( \quad -0.32 ; 0.56 \quad )$ $-0.85$	$( \quad \quad \quad 0 ; 0.5 \quad )$ $-0.75$	$( \quad -0.5 ; 0.75 \quad )$ $-0.81$	$( \quad -0.16 ; 0.53 \quad )$ $-0.83$	$( \quad -0.5 ; 0.75 \quad )$ $-0.81$	$( \quad \quad \quad 0 ; \quad 0 \quad )$ $-0$
$k = 5$	$x^* = ( \quad -0.32 : 0.56 \quad ) \quad f(x^*) = -0.85$					

Введите Eps : 0.1  
 Введите коэффициент отражения : 1  
 Введите коэффициент сжатия : 0.5  
 Введите коэффициент растяжения : 5

k = 1						
x1 = ( -1 : -1 ) f(x1) = 2		x2 = ( 0 : -1 ) f(x2) = 3		x3 = ( 1 : 0 ) f(x3) = 2		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( -1 ; -1 ) 2	( 0 ; -1 ) 3	( 1 ; 0 ) 2	( -0.5 ; -1 ) 2.25	( 1 ; 0 ) 2	( 7 ; 4 ) 36	( 0 ; 0 ) -0
k = 2						
x1 = ( -1 : -1 ) f(x1) = 2		x2 = ( 1 : 0 ) f(x2) = 2		x3 = ( 0 : 0 ) f(x3) = -0		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( -1 ; -1 ) 2	( 1 ; 0 ) 2	( 0 ; 0 ) -0	( 0 ; -0.5 ) 1.25	( 0 ; 0 ) -0	( 0 ; 2 ) -0	( 0 ; 0 ) -0
k = 3						
x1 = ( 0 : 0 ) f(x1) = -0		x2 = ( -0.13 : -0.25 ) f(x2) = 0.42		x3 = ( 0.5 : 0 ) f(x3) = 0.75		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( 0 ; 0 ) -0	( -0.13 ; -0.25 ) 0.42	( 0.5 ; 0 ) 0.75	( 0.5 ; 0 ) 0.75	( 2 ; 1 ) 3	( 0 ; 0 ) -0	( -0.25 ; -0.5 ) 0.94
k = 4						
x1 = ( 0 : 0 ) f(x1) = -0		x2 = ( -0.13 : -0.25 ) f(x2) = 0.42		x3 = ( -0.63 : -0.25 ) f(x3) = 0.17		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( 0 ; 0 ) -0	( -0.13 ; -0.25 ) 0.42	( -0.63 ; -0.25 ) 0.17	( -0.06 ; -0.13 ) 0.2	( -0.63 ; -0.25 ) 0.17	( 0 ; 0 ) -0	( 0 ; 0 ) -0
k = 5						
x1 = ( 0 : 0 ) f(x1) = -0		x2 = ( -0.63 : -0.25 ) f(x2) = 0.17		x3 = ( -0.5 : 0 ) f(x3) = -0.25		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( 0 ; 0 ) -0	( -0.63 ; -0.25 ) 0.17	( -0.5 ; 0 ) -0.25	( -0.31 ; -0.13 ) 0.01	( -0.5 ; 0 ) -0.25	( -1.25 ; 0.5 ) 0.19	( 0 ; 0 ) -0
k = 6						
x1 = ( -0.5 : 0 ) f(x1) = -0.25		x2 = ( 0 : 0 ) f(x2) = -0		x3 = ( 0.13 : 0.25 ) f(x3) = -0.33		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( -0.5 ; 0 ) -0.25	( 0 ; 0 ) -0	( 0.13 ; 0.25 ) -0.33	( -0.25 ; 0 ) -0.19	( 0.13 ; 0.25 ) -0.33	( 1.63 ; 1.25 ) 1.3	( 0 ; 0 ) -0
k = 7						
x1 = ( 0.13 : 0.25 ) f(x1) = -0.33		x2 = ( -0.5 : 0 ) f(x2) = -0.25		x3 = ( -0.38 : 0.25 ) f(x3) = -0.58		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( 0.13 ; 0.25 ) -0.33	( -0.5 ; 0 ) -0.25	( -0.38 ; 0.25 ) -0.58	( -0.19 ; 0.13 ) -0.36	( -0.38 ; 0.25 ) -0.58	( -1.13 ; 0.75 ) 0.05	( 0 ; 0 ) -0
k = 8						
x1 = ( -0.38 : 0.25 ) f(x1) = -0.58		x2 = ( 0.13 : 0.25 ) f(x2) = -0.33		x3 = ( 0.25 : 0.5 ) f(x3) = -0.56		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( -0.38 ; 0.25 ) -0.58	( 0.13 ; 0.25 ) -0.33	( 0.25 ; 0.5 ) -0.56	( -0.13 ; 0.25 ) -0.52	( 0.25 ; 0.5 ) -0.56	( 0 ; 0 ) -0	( 0 ; 0 ) -0
k = 9						
x1 = ( -0.38 : 0.25 ) f(x1) = -0.58		x2 = ( 0.25 : 0.5 ) f(x2) = -0.56		x3 = ( -0.25 : 0.5 ) f(x3) = -0.81		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( -0.38 ; 0.25 ) -0.58	( 0.25 ; 0.5 ) -0.56	( -0.25 ; 0.5 ) -0.81	( -0.06 ; 0.38 ) -0.64	( -0.25 ; 0.5 ) -0.81	( -1 ; 1 ) -0	( 0 ; 0 ) -0
k = 10						
x1 = ( -0.25 : 0.5 ) f(x1) = -0.81		x2 = ( -0.14 : 0.47 ) f(x2) = -0.77		x3 = ( -0.31 : 0.38 ) f(x3) = -0.71		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( -0.25 ; 0.5 ) -0.81	( -0.14 ; 0.47 ) -0.77	( -0.31 ; 0.38 ) -0.71	( -0.31 ; 0.38 ) -0.71	( -0.88 ; 0.25 ) -0.33	( 0 ; 0 ) -0	( -0.03 ; 0.44 ) -0.7
k = 11						
x1 = ( -0.25 : 0.5 ) f(x1) = -0.81		x2 = ( -0.14 : 0.47 ) f(x2) = -0.77		x3 = ( -0.08 : 0.59 ) f(x3) = -0.86		
x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)	xs f(xs)
( -0.25 ; 0.5 ) -0.81	( -0.14 ; 0.47 ) -0.77	( -0.08 ; 0.59 ) -0.86	( -0.2 ; 0.48 ) -0.8	( -0.08 ; 0.59 ) -0.86	( 0.39 ; 1.03 ) -0.86	( 0 ; 0 ) -0
k = 11						
x* = ( -0.08 : 0.59 )		f(x*) = -0.86				

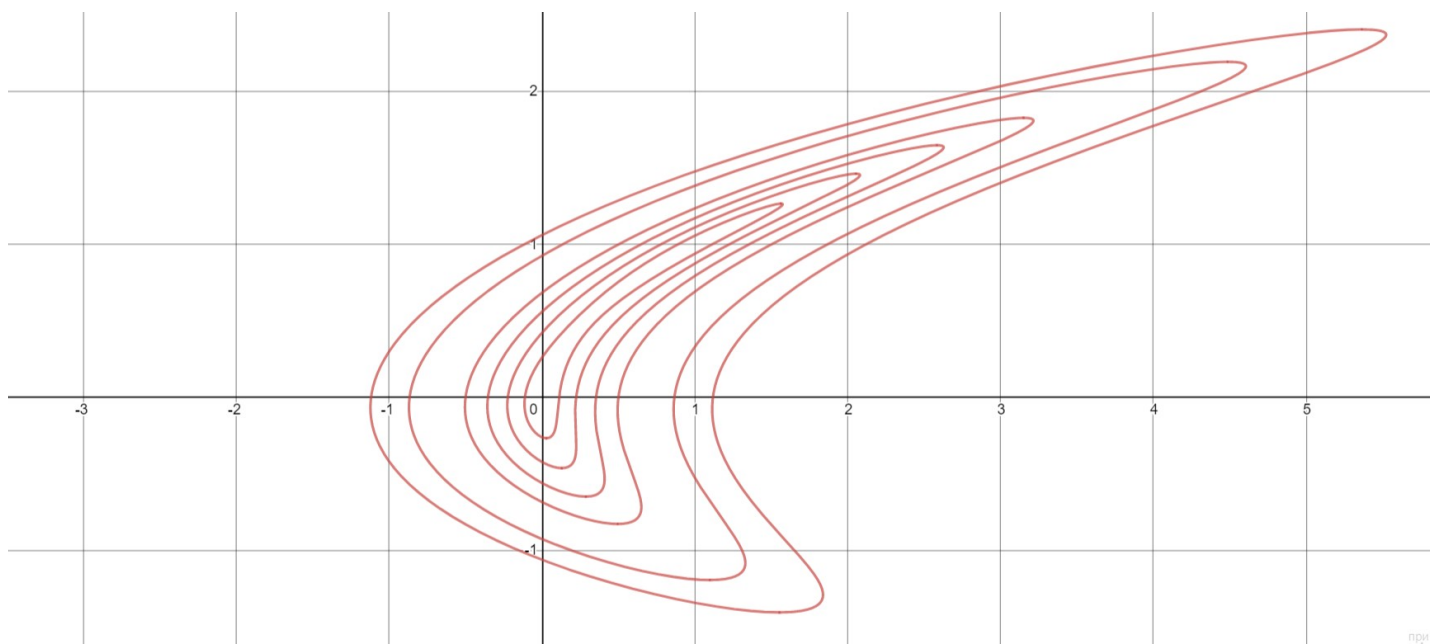
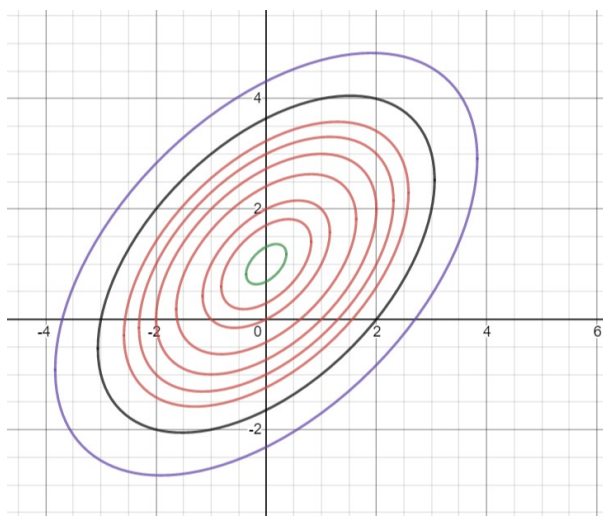
## Вторая функция

Введите Eps : 0.01  
 Введите коэффициент отражения : 1  
 Введите коэффициент сжатия : 0.5  
 Введите коэффициент растяжения : 2

k = 1	x1 = ( 0 : -1 ) f(x1) = 8		x2 = ( -1 : -1 ) f(x2) = 28		x3 = ( 1 : 0 ) f(x3) = 8	
	x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)
	( 0 ; -1 8 )	( -1 ; -1 28 )	( 1 ; 0 8 )	( -0.5 ; -1 16 )	( 1 ; 0 8 )	( 2.5 ; 1 18 )
k = 2	x1 = ( -0.25 : -0.75 ) f(x1) = 4.87		x2 = ( 0.38 : -0.38 ) f(x2) = 0.95		x3 = ( -0.13 : -0.88 ) f(x3) = 6.11	
	x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)
	( -0.25 ; -0.75 4.87 )	( 0.38 ; -0.38 0.95 )	( -0.13 ; -0.88 6.11 )	( 0.5 ; -0.5 1.44 )	( 2 ; 0 32 )	( 0 ; 0 0 )
k = 3	x1 = ( 0.38 : -0.38 ) f(x1) = 0.95		x2 = ( -0.25 : -0.75 ) f(x2) = 4.87		x3 = ( 0.25 : -0.25 ) f(x3) = 0.5	
	x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)
	( 0.38 ; -0.38 0.95 )	( -0.25 ; -0.75 4.87 )	( 0.25 ; -0.25 0.5 )	( 0.06 ; -0.56 0.84 )	( 0.25 ; -0.25 0.5 )	( 0.44 ; 0.06 1.46 )
k = 4	x1 = ( 0.25 : -0.25 ) f(x1) = 0.5		x2 = ( 0.14 : -0.39 ) f(x2) = 0.28		x3 = ( 0.31 : -0.31 ) f(x3) = 0.71	
	x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)
	( 0.25 ; -0.25 0.5 )	( 0.14 ; -0.39 0.28 )	( 0.31 ; -0.31 0.71 )	( 0.31 ; -0.31 0.71 )	( 0.88 ; 0.13 5.73 )	( 0 ; 0 0 )
k = 5	x1 = ( 0.14 : -0.39 ) f(x1) = 0.28		x2 = ( 0.25 : -0.25 ) f(x2) = 0.5		x3 = ( 0.08 : -0.33 ) f(x3) = 0.17	
	x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)
	( 0.14 ; -0.39 0.28 )	( 0.25 ; -0.25 0.5 )	( 0.08 ; -0.33 0.17 )	( 0.2 ; -0.32 0.33 )	( 0.08 ; -0.33 0.17 )	( -0.04 ; -0.34 0.25 )
k = 6	x1 = ( 0.08 : -0.33 ) f(x1) = 0.17		x2 = ( 0.11 : -0.36 ) f(x2) = 0.22		x3 = ( 0.13 : -0.32 ) f(x3) = 0.2	
	x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)
	( 0.08 ; -0.33 0.17 )	( 0.11 ; -0.36 0.22 )	( 0.13 ; -0.32 0.2 )	( 0.11 ; -0.36 0.22 )	( -0.03 ; -0.47 0.63 )	( 0 ; 0 0 )
k = 7	x1 = ( 0.08 : -0.33 ) f(x1) = 0.17		x2 = ( 0.13 : -0.32 ) f(x2) = 0.2		x3 = ( 0.09 : -0.25 ) f(x3) = 0.12	
	x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)
	( 0.08 ; -0.33 0.17 )	( 0.13 ; -0.32 0.2 )	( 0.09 ; -0.25 0.12 )	( 0.1 ; -0.32 0.18 )	( 0.1 ; -0.29 0.15 )	( 0.09 ; -0.25 0.12 )
k = 8	x1 = ( 0.09 : -0.25 ) f(x1) = 0.12		x2 = ( 0.08 : -0.33 ) f(x2) = 0.17		x3 = ( -0 : -0.23 ) f(x3) = 0.07	
	x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)
	( 0.09 ; -0.25 0.12 )	( 0.08 ; -0.33 0.17 )	( -0 ; -0.23 0.07 )	( 0.08 ; -0.29 0.14 )	( 0.04 ; -0.26 0.1 )	( -0 ; -0.23 0.07 )
k = 9	x1 = ( -0 : -0.23 ) f(x1) = 0.07		x2 = ( 0.09 : -0.25 ) f(x2) = 0.12		x3 = ( -0.02 : -0.06 ) f(x3) = 0.01	
	x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)
	( -0 ; -0.23 0.07 )	( 0.09 ; -0.25 0.12 )	( -0.02 ; -0.06 0.01 )	( 0.04 ; -0.24 0.08 )	( 0.01 ; -0.15 0.03 )	( -0.02 ; -0.06 0.01 )
k = 10	x1 = ( -0.02 : -0.06 ) f(x1) = 0.01		x2 = ( -0.04 : -0.08 ) f(x2) = 0.02		x3 = ( -0.01 : -0.15 ) f(x3) = 0.03	
	x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)
	( -0.02 ; -0.06 0.01 )	( -0.04 ; -0.08 0.02 )	( -0.01 ; -0.15 0.03 )	( -0.01 ; -0.15 0.03 )	( 0.09 ; -0.25 0.12 )	( 0 ; 0 0 )
k = 11	x1 = ( -0.02 : -0.06 ) f(x1) = 0.01		x2 = ( -0.03 : -0.05 ) f(x2) = 0.01		x3 = ( -0.03 : -0.07 ) f(x3) = 0.01	
	x1 f(x1)	xg f(xg)	xh f(xh)	xc f(xc)	xr f(xr)	xe f(xe)
	( -0.02 ; -0.06 0.01 )	( -0.03 ; -0.05 0.01 )	( -0.03 ; -0.07 0.01 )	( -0.03 ; -0.07 0.01 )	( -0.01 ; -0.15 0.03 )	( 0 ; 0 0 )
k = 11	x* = ( -0.02 : -0.06 ) f(x*) = 0.01					



## Графики ф-ий



### Сводная таблица

№	F	x1	x2	x3	Eps	a	b	y	k	x*	f(x*)
1	1	(-2;-2)	(-1;-1)	(0;-1)	0.01	1	0.5	2	8	(0.02;0.96)	1
2	1	(-2;-2)	(-1;-1)	(0;-1)	0.1	1	0.5	2	7	(-0.08;0.96)	1
3	1	(-20;-20)	(-10;-10)	(0;-10)	0.01	1	0.5	2	11	(-0.01;1.01)	1
4	1	(-20;-20)	(-10;-10)	(0;-10)	0.1	1	0.5	2	10	(0.06;1.05)	1
5	1	(-2;-2)	(-1;-1)	(0;-1)	0.1	1.5	0.5	2	7	(0.04;0.91)	0.99
6	1	(-2;-2)	(-1;-1)	(0;-1)	0.1	1	0.1	2	5	(-0.32;0.56)	0.82
7	1	(-2;-2)	(-1;-1)	(0;-1)	0.1	1	0.5	5	11	(-0.08;0.59)	0.86
8	2	(-2;-2)	(-1;-1)	(0;-1)	0.01	1	0.5	2	11	(-0.02;-0.06)	0.01
9	2	(-2;-2)	(1;1)	(0;-1)	0.01	1	0.5	2	6	(1;1)	0

## Вывод

По результатам исследований можно сделать вывод:

- 1) Алгоритм Нелдера и Мида работает эффективно как с любыми функциями.
- 2) При повышении точности количество итераций почти не меняется (в тесте №1 точность в 10 раз выше, однако итераций больше на 1).
- 3) Если у функции несколько минимумов то достаточно изменить начальные точки, однако их приходится угадывать.
- 4) Вводимая точность не соответствует абсолютной, так как она используется при подсчёте дисперсии точек а не сравнении с точным значением.
- 5) Любые изменения коэффициентов сжатия, растяжения и отражения приводят к потере точности результатов, однако изменение коэффициента отражения сказывается на точности меньше чем изменение остальных параметров (в тесте №5 ответ точнее чем в тестах 6,7).
- 6) Отдаление начальных точек от минимума сказывается на числе итераций хуже, чем повышение точности (в тесте №2 7 итераций, однако при тех-же параметрах, но более отдалёнными начальными точками 10 итераций, а в 1 (где поднялась только точность) 8 итераций).