Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра теоретической и прикладной информатики

Лабораторная работа № 3
по дисциплине «Методы оптимизации»

Метод штрафных функций

Факультет:	ПМИ
Группа:	ПМ-71
Вариант:	4
Студенты:	Бурдуков Вадим Баштовой Павел Востриков Вячеслав
Преподаватель:	Чимитова Екатерина Владимировна

Новосибирск

2020

1. Цель работы

Реализовать метод штрафных функций. Исследовать сходимость метода штрафных функций в зависимости от изменения каждого из параметров в отдельности.

2. Задание

- Применяя методы поиска минимума 0-го порядка, реализовать программу для решения задачи нелинейного программирования с использованием метода штрафных функций.
- Исследовать сходимость метода штрафных функций в зависимости от
  - выбора штрафных функций,
  - начальной величины коэффициента штрафа,
  - стратегии изменения коэффициента штрафа,
  - начальной точки,
  - задаваемой точности ε.
  - Сформулировать выводы.
- Применяя методы поиска минимума 0-го порядка, реализовать программу для решения задачи нелинейного программирования с использованием метода штрафных функций.

Функция:

$$f(x,y) = 2(x-y)^2 + 14(y-3)^2 \to \min$$

при ограничении:

а) $y - x \geq -1$

б) $x = -y$

3. Метод штрафных функций

$[\ f(x,y) \to \ \min\ ] \to \ F(x,y) = \ f(x,y) + I(x,y)$, где $I(x,y) -$ штраф

$I(x,y) = r \cdot h\big(q(x,y)\big), r := \ \beta r,\ \ \beta > 1$ на каждой итерации

$f(x,y) = 2(x-y)^2 + \ 14(y-3)^2 \to min$

1) Ограничение: $y - x \geq -1 \to -(y-x) - 1 \leq 0$:

$F(x,y) = 2(x-y)^2 + \ 14(y-3)^2 + r \cdot \left(\frac{x-y-1\ +\ |x-y-1|}{2}\right)^k$

2) Ограничение: $x = -y \to x + y = 0$:

$F(x,y) = 2(x-y)^2 + \ 14(y-3)^2 + r \cdot |x+y|^k$

Будем использовать метод Гаусса.

4. Исследования

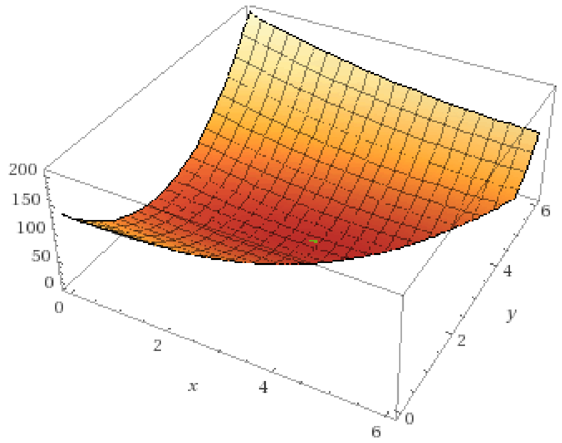
Исследования проводим на ограничение неравенством (внутри области)

Функция и минимальное значение (f = 0):

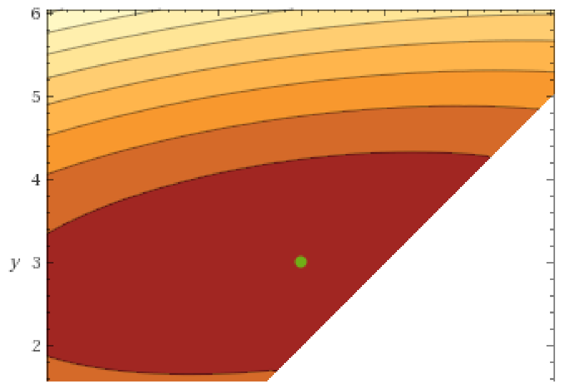
Global minimum:

$\min\{2(x-y)^2 + 14(y-3)^2 \mid y - x \geq -1\} = 0$ at $(x, y) = (3, 3)$

3D plot:



Contour plot:



Экспоненциальный вывод, а точнее степень f в точке можно считать мерой близости к истинному ответу, т.к. минимальное значение f при ограничении = 0.

Исследование сходимости от выбора штрафных функций

$x_0 = (0,\ 10),\ h(x-y-1) = \left(\frac{x-y-1+|x-y-1|}{2}\right)^k$

$r_0 = 1,\ \beta = 1.1,\ \varepsilon = 1e-7$

k	Количество итераций	Количество вычислений f	Лучшая точка	Значение f в лучшей точке
1	1	681	(3,00000044137978, 2,99999947895362)	5,65337886331658E-12
2	1	695	(3,00000039023144, 2,99999942780527)	6,43622350231373E-12
4	1	778	(3,00000015813157, 2,9999991957054)	1,09089854420594E-11
8	1	859	(2,99999978283927, 2,99999882041311)	2,13324816110725E-11
16	1	869	(2,99999956912423, 2,99999888403512)	1,83739807537154E-11

При увеличении k увеличивается количество вычислений функции, для достижения той же точности: $h(x - y - 1) \leq eps$.

Исследование сходимости от начальной величины коэффициента штрафа

Выберем $h\big(q(x,y)\big) = \left(\frac{x-y-1+|x-y-1|}{2}\right)^1$

Не удалось найти точку в области, где итераций будет больше 1, поэтому выберем:

$x_0 = (100, 100),\ \beta = 1.1,\ \varepsilon = 1e-7$

r0	Количество итераций	Количество вычислений f	Лучшая точка	Значение f в лучшей точке
1.00E-04	1	815	(3,0000002686436, 2,9999993474644)	7,65838012693641E-12
1.00E-03	1	815	(3,00000009557536, 2,99999917439617)	1,1239845904932E-11
1.00E-02	1	815	(3,0000004978891, 2,99999957670991)	4,20558523845309E-12
1.00E-01	1	815	(3,00000054933647, 2,99999962815727)	3,63288043343193E-12
1.00E+00	1	815	(3,00000023977685, 2,99999931859765)	8,19747044284396E-12
1.00E+01	1	890	(2,99999978615871, 2,99999886497951)	1,97329433502017E-11
1.00E+02	1	1172	(2,99999965257011, 2,99999907138287)	1,27481740859401E-11
1.00E+03	1	4162	(2,99999986985115, 2,99999909559079)	1,26503425283706E-11
1.00E+04	1	9373	(2,99999969982053, 2,99999921760002)	9,03516948439358E-12

Но если взять прямую x = -y (буква б), точку $\boldsymbol{x_0} = (\mathbf{1}, \mathbf{1})$ и те же параметры, то:

r0	Количество итераций	Количество вычислений f	Лучшая точка	Значение f в лучшей точке	Абсолютная разница между истинным и полученным
1.00E-07	311	46760	(-1,90057117500327, 1,91825313871625)	45,6156581398142	0,202523678367584
1.00E-06	287	44398	(-1,90064447802317, 1,91823786921207)	45,6166797334575	0,201502084724304
1.00E-05	263	41373	(-1,90071568103404, 1,91822059808585)	45,6176957522475	0,200486065934342
1.00E-04	239	37355	(-1,90078846162046, 1,9182051212475)	45,6187071832697	0,199474634912129
1.00E-03	215	30364	(-1,90086279489575, 1,9181916005498)	45,6197140447605	0,198467773421271
1.00E-02	191	23186	(-1,90093087188578, 1,91817291781136)	45,6207143151455	0,197467503036314
1.00E-01	166	19629	(-1,90055067021684, 1,91825924254624)	45,6153561652049	0,202825652976969
1.00E+00	142	15983	(-1,9006229079984, 1,91824198321252)	45,6163789698267	0,201802848355072
1.00E+01	118	13536	(-1,90069720791513, 1,9182279298039)	45,6173972780716	0,200784540110249
1.00E+02	94	11251	(-1,90075986473651, 1,91820266381895)	45,6184075319345	0,199774286247283
1.00E+03	70	9231	(-1,88531459843146, 1,9027213999692)	45,6206310319787	0,197550786203088
1.00E+04	46	614207	(-1,87037380210321, 1,88764749203604)	45,6329365330809	0,185245285100905
1.00E+05	21	3356105	(-1,86455682281639, 1,88228067576946)	45,6341208013147	0,184061016867169
1.00E+06	1	10840854	(-1,86271744741902, 1,87823887162188)	45,6644206188732	0,15376119930859

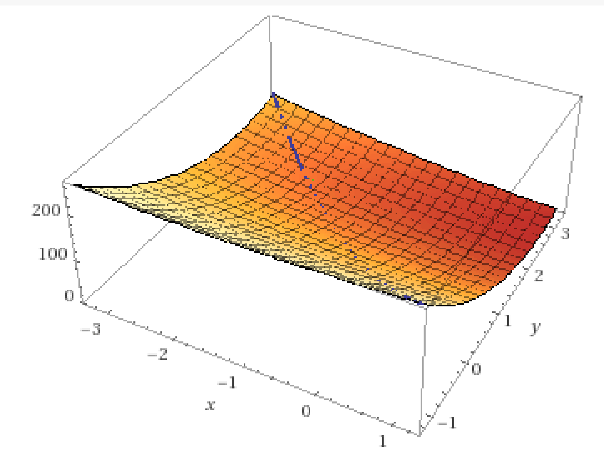
$$F(x, y) = 2(x - y)^2 + 14(y - 3)^2 + r \cdot |x + y|^4,$$

Истинное значение: $fun\left(-\frac{21}{11}, \frac{21}{11}\right) = \frac{504}{11} \cong 45{,}81(81)$

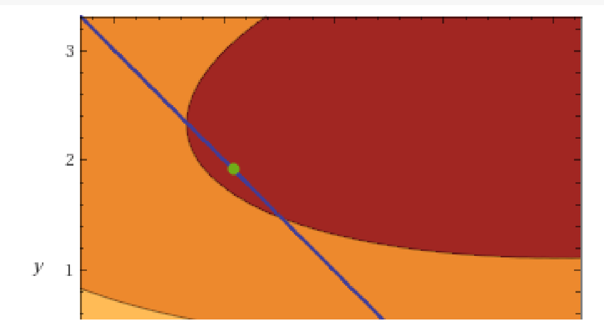
Global minimum: Approximate form

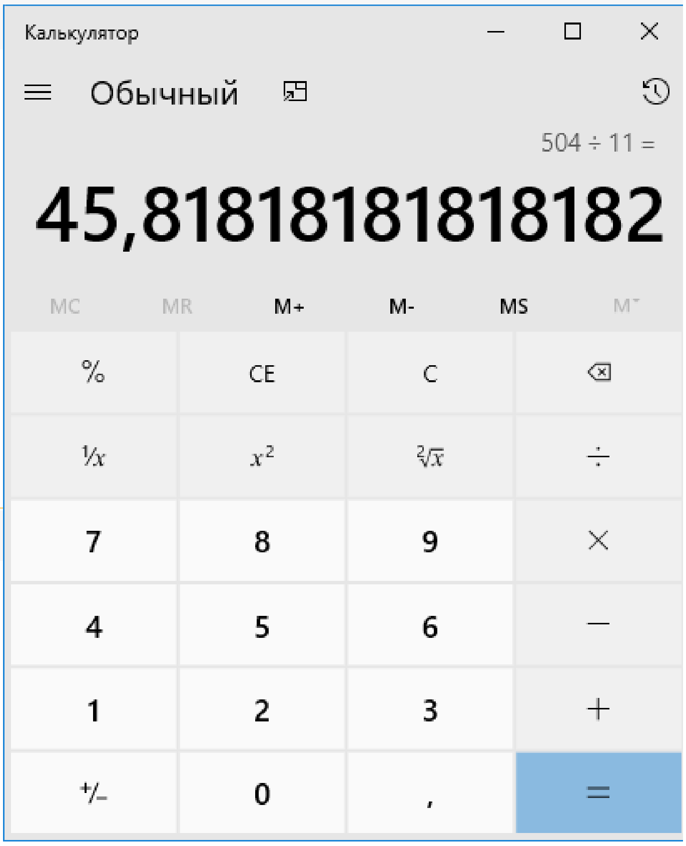
$$\min\{2(x-y)^2 + 14(y-3)^2 \mid x = -y\} = \frac{504}{11} \text{ at } (x, y) = \left(-\frac{21}{11}, \frac{21}{11}\right)$$

3D plot:



Contour plot:





Как видим, в случае прямой – мы попадаем в цикл и можем судить о количестве итераций.

При уменьшении r0 (1 – стандартное значение) увеличивается количество итераций и количество вычислений функции.

При увеличении начального коэффициента уменьшается количество итераций и (до определенного момента) количество вычислений функции.

Точность при степени r0 = -2 и -1, показывает, что r0 не всегда убывает при увеличении r0.

Повторим исследование с параметром k для функции h:

k	Количество итераций	Количество вычислений f	Лучшая точка	Значение f в лучшей точке	Абсолютная разница между истинным и полученным
2	107	13724	(-1,91490633789692, 1,91521347463387)	45,8166038046584	1.58E-03
4	142	15983	(-1,9006229079984, 1,91824198321252)	45,6163789698267	2.02E-01
8	156	17903	(-1,8066890793287, 1,93891251664823)	44,0651630752387	1.75E+00
16	159	18818	(-1,6172585320689, 1,980700318644)	40,7575890778851	5.06E+00
32	156	143399	(-1,42024359378826, 2,02418059684992)	37,3148673726878	8.50E+00
64	151	1499000	(-1,27565581713589, 2,05291291930829)	34,876715408371	1.09E+01

При увеличении k всё также увеличивается количество вычислений функции, а количество итераций увеличивается до k = 16, затем следует уменьшение. Степень в научном представлении точности увеличивается при увеличении k -> k = 2 оптимальнее.

Остальные исследования проведем на x = -y

Исследование сходимости от стратегии изменения коэффициента штрафа

Зафиксируем $h\big(q(x,y)\big) = |x+y|^2,\ x_0 = (1,1),\ r_0 = 1.5,\ \varepsilon = 1e-7$

a) $\boldsymbol{r_{k+1} = \beta r_k}$

beta	Количество итераций	Количество вычислений f	Лучшая точка	Значение f в лучшей точке	Абсолютная разница между истинным и полученным
1,5	25	4814	(-1,92887010072166, 1,92915821380111)	45,8246908642231	6.51E-03
2	15	3174	(-1,93551218556754, 1,93580335239596)	45,8314545628843	1.33E-02
2,5	12	2576	(-1,93860102315476, 1,93879959618437)	45,8359294558425	1.77E-02
3	10	2282	(-1,94056433631145, 1,94080364616363)	45,8382824991424	2.01E-02
3,5	9	2201	(-1,93745226518977, 1,93766310989959)	45,8343753168775	1.62E-02
4	8	2051	(-1,93924968156659, 1,93953811288721)	45,8361458470652	1.80E-02
4,5	8	1902	(-1,94640347337381, 1,94652767954089)	45,8479456417319	2.98E-02
5	7	1754	(-1,9454473046835, 1,94574498446426)	45,8451826642884	2.70E-02

С уменьшением $\boldsymbol{\beta}$ происходит уменьшение количества итераций и вычислений функции.

Порядок точности может поменяться, но в целом – сохраняется.

b) $\boldsymbol{r_{k+1} = r_k + \beta}$

beta	Количество итераций	Количество вычислений f	Лучшая точка	Значение f в лучшей точке	Абсолютная разница между истинным и полученным
1,5	15167	2215299	(-1,90877234326409, 1,9090885530367)	45,8156274297292	2.55E-03
2	11891	1736927	(-1,90879548291639, 1,90911168817626)	45,8157304698954	2.45E-03
2,5	9655	1410387	(-1,90881156865335, 1,90912778743604)	45,8157659444403	2.42E-03
3	8049	1175907	(-1,90882287976915, 1,90913909817944)	45,8157668369442	2.41E-03
3,5	6900	1008302	(-1,9088301351896, 1,90914632067075)	45,8157670851402	2.41E-03
4	6037	882294	(-1,90883505204377, 1,90915127354531)	45,8157668352163	2.41E-03
4,5	5367	784401	(-1,90883892764754, 1,90915510896698)	45,8157671360282	2.41E-03
5	4830	705995	(-1,90884148027978, 1,90915769351242)	45,8157669369372	2.41E-03

С уменьшением $\boldsymbol{\beta}$ происходит уменьшение количества итераций и вычислений функции.

Точность не меняется.

c) $\boldsymbol{r_{k+1} = r_k{}^{\beta}}$

C:\WINDOWS\system32\cmd.exe

```
1,5     9       2392    (-1,94365535718471, 1,94387048625584)    45,8429561447288        2.48E-002
2       6       1550    (-1,96345421118504, 1,96346962125087)    45,8830971905684        6.49E-002
2,5     5       1240    (-2,00667226767326, 2,0066730309307)     46,027663920983 2.09E-001
3       4       1329    (-1,95157635900673, 1,95169725786946)    45,8570612449384        3.89E-002
3,5     4       1826    (-1,95005656166626, 1,9500567397584)     45,8551005492025        3.69E-002
```

Начиная с $\boldsymbol{\beta}$ = 4, нельзя достичь нужной точности, следовательно степенной метод – не универсален. Количество итераций также убывает с увеличением параметра, в то время как количество вычислений сначала уменьшалось, а затем (с 3) начало расти (точность решения сохраняется, падая в минимуме (2.5)).

Исследование сходимости от начальной точки

Выберем $h\big(q(x,y)\big) = \ |x+y|^2,\ r_0 = 1.5,\ \varepsilon = 1e-7,\ r_k = \ r_{k-1}{}^{1.5},\ \varepsilon_{\text{Фибоначчи}} = 1e-6$

Точка не на прямой:

Точка	Количество итераций	Количество вычислений f	Лучшая точка	Значение f в лучшей точке	Абсолютная разница между истинным и полученным
(0, 0)	9	213523	(-1,94114194204069, 1,94135718655372)	45,8392531571624	2.11E-02
(10, 20)	9	213632	(-1,94114277727616, 1,94135823555529)	45,8392543250258	2.11E-02
(20, 40)	9	213649	(-1,94114182527961, 1,94135688798982)	45,8392530079892	2.11E-02
(30, 60)	9	213659	(-1,94114223414583, 1,94135743471809)	45,8392535751317	2.11E-02
(40, 80)	9	213663	(-1,94114193342887, 1,94135696267704)	45,8392531645952	2.11E-02
(50, 100)	9	213724	(-1,94114201242768, 1,94135704554306)	45,839253276235	2.11E-02
(60, 120)	9	213731	(-1,94114258736669, 1,94135781978151)	45,8392540732637	2.11E-02
(70, 140)	9	213734	(-1,94114199378667, 1,94135701362594)	45,8392532511093	2.11E-02
(80, 160)	9	213734	(-1,94114208976288, 1,94135722409965)	45,8392533763083	2.11E-02
(90, 180)	9	213738	(-1,94114268702267, 1,94135787734488)	45,839254218214	2.11E-02

Точка на прямой:

Точка	Количество итераций	Количество вычислений f	Лучшая точка	Значение f в лучшей точке	Абсолютная разница между истинным и полученным
(0, 0)	9	213523	(-1,94114194204069, 1,94135718655372)	45,839253157162	2.11E-02
(10, -10)	9	213574	(-1,94114265896984, 1,94135788085566)	45,839254175697	2.11E-02
(20, -20)	9	213639	(-1,94114194735319, 1,94135701198703)	45,839253180903	2.11E-02
(30, -30)	9	213597	(-1,94114239674812, 1,9413576126226)	45,839253804378	2.11E-02
(40, -40)	9	213600	(-1,94114214500201, 1,94135723689058)	45,839253458576	2.11E-02
(50, -50)	9	213655	(-1,94114268087571, 1,94135784968342)	45,839254211400	2.11E-02
(60, -60)	9	213615	(-1,94114203527875, 1,94135727613295)	45,839253289682	2.11E-02
(70, -70)	9	213615	(-1,94114208823827, 1,94135725625064)	45,839253371104	2.11E-02
(80, -80)	9	213615	(-1,94114180924316, 1,94135712221565)	45,839252963242	2.11E-02
(90, -90)	9	213673	(-1,94114197775419, 1,94135710233333)	45,839253218375	2.11E-02

Координаты точки не определяет результат.

Исследование сходимости от ε

Зафиксируем $h(q(x,y)) = |x+y|^2,\ r_0 = 1.5,\ \varepsilon = 1e-7,\ r_k = {r_{k-1}}^{1.5},$

$x_0 = (1,1), \varepsilon_{\text{Фибоначчи}} = 1e-8$

eps	Количество итераций	Количество вычислений f	Лучшая точка	Значение f в лучшей точке	Абсолютная разница между истинным и полученным
1.00E-08	10	2538	(-1,94386917001563, 1,94387034155766)	45,844783069256	2.66E-02
1.00E-07	9	2392	(-1,94365535718471, 1,94387048625584)	45,8429561447288	2.48E-02
1.00E-06	9	2392	(-1,94365535718471, 1,94387048625584)	45,8429561447288	2.48E-02
1.00E-05	9	2392	(-1,94365535718471, 1,94387048625584)	45,8429561447288	2.48E-02
1.00E-04	8	2246	(-1,9370093445273, 1,94389395289496)	45,7862157152881	3.20E-02
1.00E-03	8	2246	(-1,9370093445273, 1,94389395289496)	45,7862157152881	3.20E-02
1.00E-02	7	2093	(-1,87418687878222, 1,94461889925734)	45,2628890903363	5.55E-01

При увеличении ε увеличивается количество вычислений и уменьшается количество итераций. Точность ухудшается.

5. Код

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Lab
{
   class Program
   {
      public class Iteration
      {
         public int number_iter;
         public ValueTuple<double, double> point;
         public double value_point, value_strafe;

         public Iteration(int k, ValueTuple<double, double> xk, double fk, double strafe)
         {
            number_iter = k;
            point = xk;
            value_point = fk;
            value_strafe = strafe;
         }
      }

      public static double Fun(ValueTuple<double, double> point) => 2 * Math.Pow(point.Item1
- point.Item2, 2) + 14 * Math.Pow(point.Item2 - 3, 2);

      public static int k = 1;

      // x + y = 0 -> |x + y|^k
      public static double Fun_strafe(ValueTuple<double, double> point) =>
Math.Pow(Math.Abs(point.Item1 + point.Item2), k);

      // (((x - y - 1) + |x - y - 1|) / 2)^k
      //public static double Fun_strafe(ValueTuple<double, double> point)
      //{
      //   return Math.Pow(((point.Item1 - point.Item2 - 1) + Math.Abs(point.Item1 -
point.Item2 - 1)) / 2, k);
      //}
```

```
        public static double Fun_min(double value)
        {
            count_fun += 1;
            // минимизация по x -> value = x, current_value = y
            if (current_index == 0)
                return Fun((value, current_value)) + r * Fun_strafe((value, current_value));
            // минимизация по y -> value = y, current_value = x
            else
                return Fun((current_value, value)) + r * Fun_strafe((current_value, value));
        }

        public static int Number_fib(int n)
        {
            int f1 = 1, f2 = 1, i = 2, sum = 1;
            while (i < n)
            {
                sum = f1 + f2;
                f1 = f2;
                f2 = sum;
                i++;
            }
            return sum;
        }

        // -6, или -8
        public static double Fibonachi(ValueTuple<double, double> interval, double eps=1e-8)
        {
            double ak, bk;
            (ak, bk) = interval;

            if (bk - ak < eps)
                return (ak + bk) / 2;
            else
            {
                int n = 1, f = Number_fib(n);
                double s = (bk - ak) / eps, yk, zk;
                List<int> numbers_fib = new List<int>() { 0 };

                while (f < s)
                {
                    n++;
                    numbers_fib.Add(f);
                    f = Number_fib(n);
                }
                numbers_fib.Add(f);
                n -= 2;

                yk = ak + (double)numbers_fib[n] / numbers_fib[n + 2] * (bk - ak);
                zk = ak + (double)numbers_fib[n + 1] / numbers_fib[n + 2] * (bk - ak);

                double f1 = Fun_min(yk), f2 = Fun_min(zk);

                int k = 1;
                while (k < n)
                {
                    k++;
                    if (f1 <= f2)
                    {
                        bk = zk;
                        zk = yk;
                        f2 = f1;
                        yk = ak + (double)numbers_fib[n - k + 1] / numbers_fib[n - k + 3] * (bk -
ak);
                        f1 = Fun_min(yk);
```

```
                }
                else
                {
                   ak = yk;
                   yk = zk;
                   f1 = f2;
                   zk = ak + (double)numbers_fib[n - k + 2] / numbers_fib[n - k + 3] * (bk -
ak);
                   f2 = Fun_min(zk);
                }
             }
             return yk;
          }
       }

       // value - значение переменной по которой минимизируем
       public static ValueTuple<double, double> Find_interval(double value, double dx=1e-2)
       {
          double h = 0, x0 = value, x1, x2;
          if (Fun_min(x0) > Fun_min(x0 + dx))
          {
             x1 = x0 + dx;
             h = dx;
          }
          else
          {
             x1 = x0 - dx;
             h = -dx;
          }

          h *= 2;
          x2 = x1 + h;

          while (Fun_min(x1) > Fun_min(x2))
          {
             h *= 2;
             x0 = x1;
             x1 = x2;
             x2 = x1 + h;
          }
          return x0 > x2 ? ValueTuple.Create(x2, x0) : ValueTuple.Create(x0, x2);
       }

       public static int count_fun = 0, current_index = 0;
       public static double current_value, r;

       public static ValueTuple<ValueTuple<double, double>, double>
Gauss_method(ValueTuple<double, double> point, double eps_fun=1e-6, double eps_var=1e-6)
       {
          ValueTuple<double, double> xk = point, previous_xk;
          double fk = double.MaxValue, previous_fk, a, b;
          bool flag = true;

          while (flag)
          {
             previous_xk = xk;

             current_index = 0;
             current_value = xk.Item2;

             (a, b) = Find_interval(xk.Item1);
             xk.Item1 = Fibonachi(ValueTuple.Create(a, b));

             current_index = 1;
             current_value = xk.Item1;
```

```
            (a, b) = Find_interval(xk.Item2);
            xk.Item2 = Fibonachi(ValueTuple.Create(a, b));

            previous_fk = fk;
            // т.к. current_index = 1
            fk = Fun_min(xk.Item2);

            if (Math.Abs(fk - previous_fk) < eps_fun && Math.Abs(xk.Item1 -
previous_xk.Item1) < eps_var && Math.Abs(xk.Item2 - previous_xk.Item2) < eps_var)
               flag = false;
         }

         return ValueTuple.Create(xk, fk);
      }

      public static ValueTuple<int, int, ValueTuple<double, double>, double,
List<Iteration>> Strafe(ValueTuple<double, double> x0, double eps=1e-7, double r0=1, double
beta=1.2)
      {
         ValueTuple<double, double> xk;
         int p = 1;
         double fk;
         r = r0;

         (xk, fk) = Gauss_method(x0);

         List<Iteration> history = new List<Iteration>();

         while (Fun_strafe(xk) > eps)
         {
            history.Add(new Iteration(p, xk, fk, Fun_strafe(xk)));
            r = Math.Pow(r, beta);
            // r += beta;
            // r *= beta;
            (xk, fk) = Gauss_method(xk);
            p += 1;
         }

         history.Add(new Iteration(p, xk, fk, Fun_strafe(xk)));
         return ValueTuple.Create(p, count_fun, xk, fk, history);
      }

      static void Main(string[] args)
      {
         ValueTuple<double, double> x0 = (1, 1);
         int[] power_k = { 1, 2, 4, 8, 16, 32 };
         string path = "file.txt";

         ValueTuple<double, double> x_best;
         double f_best;
         int num1, num2;
         List<Iteration> hist = new List<Iteration>();
         // (num1, num2, x_best, f_best, hist) = Strafe(x0, eps: 1e-7, r0: 1.5, beta: 1.5);

         using (var stream = File.CreateText(path))
         {
            k = 2;
            stream.WriteLine($"beta;Количество итераций;Количество вычислений f;Лучшая
точка;Значение f в лучшей точке;Абсолютная разница между истинным и полученным");
            for (double eps = 1e-8; eps < 1e-1; eps *= 10, count_fun = 0)
            {
               (num1, num2, x_best, f_best, hist) = Strafe(x0, eps: eps, r0: 1.5, beta:
1.5);
```

```
                Console.WriteLine($"{eps.ToString("E2",
System.Globalization.CultureInfo.InvariantCulture)}\t{num1}\t{num2}\t{x_best}\t{f_best}\t{Ma
th.Abs(f_best - Fun(((double)-21 / 11, (double)21 / 11))).ToString("E2",
System.Globalization.CultureInfo.InvariantCulture)}");
                stream.WriteLine($"{eps.ToString("E2",
System.Globalization.CultureInfo.InvariantCulture)};{num1};{num2};{x_best};{f_best};{Math.Ab
s(f_best - Fun(((double)-21 / 11, (double)21 / 11))).ToString("E2",
System.Globalization.CultureInfo.InvariantCulture)}");
            }

        }
    }
  }
}
```