МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

Лабораторная работа №3

РЕШЕНИЕ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ МЕТОДОМ РЕЛАКСАЦИИ И МЕТОДОМ ПРОСТЫХ ИТЕРАЦИЙ

Преподаватель: Полевиков Виктор Кузьмич
доцент кафедры вычислительной математики
Студент: Болтач Антон Юрьевич
2 курс 9 группа

Минск 2017

Постановка задачи

1) Построить стандартную программу решения систем линейных алгебраических уравнений методом протой итерации и релаксации.
2) Рассмотрим СЛАУ вида:

$$\mathbf{A*x = f}$$

Задана матрица **A** с строгим диагональным преобладанием размером 20x20:

Матрица A:

```
-78.92 1.24 -0.87 0.03 3.35 -3.88 -0.47 -2.92 -0.12 -0.59 -3.55 3.24 2.06 -0.11 3.74 3.02 -1.08 -1.84 3.84 1.8

1.42 -95.15 1.61 -1.78 3.78 0.31 1.31 -2.34 -0.64 3.7 1.08 -3.84 -4.0 -3.09 -1.4 1.64 -0.55 -1.58 2.79 -2.66

-3.67 -2.21 66.06 -3.31 -1.7 -0.9 3.52 0.97 -3.56 3.21 -1.22 2.97 -0.75 0.76 -2.2 -0.94 0.75 -2.68 0.4 0.97

-0.41 1.24 0.93 -50.48 2.47 3.36 3.82 1.71 -2.48 2.83 2.5 3.5 2.52 0.48 -1.96 -2.06 1.5 -1.43 2.35 -3.53

0.8 0.52 -3.32 -2.94 -55.96 1.82 -1.89 1.54 -3.05 0.25 -0.65 2.96 -0.7 2.54 -2.97 2.52 0.15 1.73 1.76 -1.66

-2.91 -1.05 0.86 1.74 0.31 38.27 -1.27 -2.04 -0.95 -1.96 3.64 -3.69 -1.77 2.57 1.25 1.41 3.55 0.0 2.96 -0.24

2.78 1.26 -1.64 2.21 -1.41 -1.39 -36.1 -1.68 -0.28 0.5 3.92 2.91 1.14 -1.91 -1.21 1.0 1.14 1.66 2.58 -1.98

0.62 2.09 -1.66 3.25 -0.55 1.03 3.19 -62.49 3.2 0.87 2.11 -3.16 -1.0 3.28 3.03 -1.08 2.05 2.65 -0.3 1.85

-2.07 -0.96 -3.26 3.73 -2.24 -1.6 3.27 -0.93 -93.56 2.55 3.04 2.06 3.62 2.99 -1.01 -0.64 1.18 -0.98 0.44 -2.66

0.83 1.98 -1.76 -2.76 0.51 0.75 1.36 -2.33 3.11 46.0 -2.34 2.35 2.69 -3.29 -1.84 -0.12 -2.72 -0.69 -0.11 -1.12

-2.53 -0.58 1.97 -2.21 0.26 2.72 -2.84 0.99 -0.67 -3.94 -83.15 -0.39 -3.11 -0.2 -3.17 -1.14 3.46 -3.21 -3.83 2.88

-0.37 -1.12 -1.35 -3.18 -2.8 1.85 3.4 -2.66 1.9 1.66 -3.25 76.09 0.29 2.28 3.34 0.47 -1.84 -1.65 -3.97 -0.08

1.81 -2.87 3.65 0.68 -0.38 1.6 -2.19 1.02 -0.29 1.19 -1.33 -2.7 -88.23 -3.26 1.28 2.65 3.73 -1.5 1.1 0.16

2.76 -3.1 -0.79 -2.63 -3.37 -3.15 0.6 -3.42 -2.54 3.88 -3.32 1.21 -2.42 -41.26 1.56 -0.37 -0.72 1.08 1.44 -1.59

0.83 0.79 -0.12 -0.18 0.81 3.22 -1.7 0.04 -1.77 3.51 -1.4 -2.4 -2.98 1.88 48.9 -1.39 -0.66 1.97 -1.4 -0.18

3.08 1.05 -0.25 -3.01 1.64 1.66 -1.0 0.7 -0.65 3.45 2.19 1.35 -0.17 0.75 -0.7 -86.68 3.2 -1.38 -2.19 2.08

-0.71 2.49 -3.91 -1.79 2.52 0.93 1.97 -1.38 -0.81 -0.56 0.17 3.88 3.92 -0.72 3.08 3.55 84.16 -3.7 2.95 1.37

1.02 -3.99 -1.27 3.47 -0.65 2.4 -1.36 2.99 0.71 -0.17 3.14 -0.81 0.77 -3.37 2.27 -2.4 -2.03 -45.01 -1.85 1.64

3.71 -0.86 1.19 -0.24 -1.18 -2.79 -0.55 0.01 -0.75 -2.9 3.1 -2.22 -2.72 1.54 1.56 -3.11 2.3 0.76 -54.22 -0.07

2.19 3.04 2.65 2.4 2.94 3.46 0.96 -2.92 1.55 0.34 -3.13 -0.95 -2.71 2.32 0.45 0.07 3.74 3.62 -0.08 90.6
```

Вектор значений f:

1324.54049999999984
3447.6582999999996
2366.633200000001
1319.0361
-966.5789000000003
-995.2539
-1541.7934999999998
1842.8878
7920.6017
2148.5085000000004
-261.58119999999997
2438.7353000000003
-2967.8097000000002
1629.7872
1839.3933000000002
4989.893400000001
-7544.434699999999
729.0142999999999
-294.10679999999996
-9357.625499999998

Транспонированный вектор значений точного решения x*:

23.48 14.16 -18.96 90.02 3.84 46.79 82.17 73.35 92.79 80.78 -77.22 -64.71 30.78 -20.03 35.31 16.84 30.06 -74.82 35.42 67.68

Краткая теория и алгоритм решения

Рассмотрим систему линейных алгебраических уравнений $\mathbf{Ax = f}$.
Пусть **A** — квадратная матрица со строгим преобладанием главной диагонали.

Для предложенной системы необходимо:

1) Построить сходящийся алгоритм метода простых итераций и найти решение с точностью $\boldsymbol{\varepsilon} = 10^{-7}$. Определить количество выполненных итераций $\boldsymbol{k_\varepsilon}$.
2) Построить сходящийся алгоритм метода релаксации и найти решение с точностью $\boldsymbol{\varepsilon} = 10^{-7}$ для следующих значений параметра q: 0.1; 0.2; …; 1.8; 1.9.
3) Определить отрезок $[\boldsymbol{q_i}\ ;\ \boldsymbol{q_{i+1}}]$, на котором алгоритм метода релаксации сходится наиболее быстро, и уточнить на нем $\boldsymbol{q_{opt}}$.
4) Привести решение при $\boldsymbol{q} = 1$ и построить график $\boldsymbol{k_\varepsilon(q)}$.

Метод простых итераций:

Для реализации алгоритма *МПИ* выполним задачу по приведению данной системы к каноническому виду: $\boldsymbol{x^{k+1}} = \mathbf{B}\boldsymbol{x^k} + \mathbf{g}$, где ($\boldsymbol{k}$ = 0, 1, 2…). Критерий сходимости имеет вид: $\boldsymbol{\rho(B)} < 1$, где $\boldsymbol{\rho}$ - вектор собственных значений матрицы. Пусть $\boldsymbol{A = A^T > 0}$, тогда справедливы следующие формулы:

$$\boldsymbol{B = E - \frac{A}{\|A\|}}\ ,\ \ \boldsymbol{g = \frac{f}{\|A\|}}.$$

При любом значении вектора начального приближения $\boldsymbol{x^0}$ алгоритм продолжает свою работу, пока не выполнено условие: $\boldsymbol{\left\|x^{k+1} - x^k\right\| \le \varepsilon}$**.**

Априорную оценку необходимого числа итераций для заданного ε можно выполнить при условии, что $\|\boldsymbol{B}\| < 1$, с помощью формулы:

$$\boldsymbol{k_\varepsilon = \left\lfloor \log_{\|B\|} \frac{\varepsilon(1 - \|B\|)}{\|g\|} \right\rfloor}$$

Метод релаксации(при q = 1, метод Зейделя):

Пусть $\boldsymbol{A = A^T > 0}$, тогда матрица **B** и вектор **g** строятся следующим образом:

$$\boldsymbol{b_{ii} = 0,\ \ b_{ij} = \frac{-a_{ij}}{a_{ii}}\ (i \ne j),\ \ g_i = \frac{f_i}{a_{ii}};\ \forall i = \overline{1,n}}.$$

Для построения алгоритма необходимо представить матрицу **B** в виде суммы нижнетреугольной матрицы **L** и верхнетреугольной матрицы **F.** Тогда при любом значении вектора начального приближения $\boldsymbol{x^0}$ алгоритм имеет вид:

$$\boldsymbol{x^{k+1} = (1 - q)x^k + q\left[Lx^{k+1} + Rx^k + g\right], k = 0, 1, \ldots\ \ ,\ \ \forall q \in (0; 2)}.$$

Построенный алгоритм сходится при $0 < \boldsymbol{q} < 2$ и продолжает работу до тех пор, пока не будет выполнено условие:

$$\boldsymbol{\frac{\left\|x^{k+1} - x^k\right\|}{q} \le \varepsilon}.$$

Листинг программы на языке программирования Java

```
import java.util.Random;

public class main {
    private static int k = 0;
    private static double ke = 0;
    private static double q = 0.1;
    private static double qOptimal = q;
    private static int remIteration = Integer.MAX_VALUE;

    public static void output(double[][] arr) {
        for (int i = 0; i < arr.length; i++) {
            for (int j = 0; j < arr.length; j++)
                System.out.print(arr[i][j] + " ");
            System.out.println();
        }
        System.out.println();
    }
    public static void output(double[] vector) {
        for (int i = 0; i < vector.length; i++)
            System.out.println(vector[i]);
        System.out.println();
    }
    public static void output(double[] a, double[] b) {
        for (int i = 0; i < a.length; i++)
            System.out.println("X = " + a[i] + " Answer = " + b[i]);
    }

    public static double[][] transposeMatrix(double [][] arr){
        double[][] temp = new double[arr[0].length][arr.length];
        for (int i = 0; i < arr.length; i++)
            for (int j = 0; j < arr[0].length; j++)
                temp[j][i] = arr[i][j];
        return temp;
    }

    public static double[][] multiplicar(double[][] arr, double[][] arrT) {
        double[][] C = new double[arr.length][arr.length];
        for (int i = 0; i < arr.length; i++)
            for (int j = 0; j < arr.length; j++)
                C[i][j] = 0;

        for (int i = 0; i < arr.length; i++)
            for (int j = 0; j < arr.length; j++)
                for (int k = 0; k < arr.length; k++) {
                    C[i][j] += arr[i][k] * arrT[k][j];
                    C[i][j] = Math.rint(C[i][j] * 100.0) / 100.0;
                }

        return C;
    }
    public static double[] multiplicar(double[][] arr, double[] x) {
        int n = arr.length;
        double[] f = new double[n];
        for (int j = 0; j < f.length; j++)
            for (int i = 0; i < f.length; i++)
                f[j] += arr[j][i] * x[i];

        return f;
    }

    public static double norma(double[] a) {
        double result = 0;
        for (int i = 0; i < a.length; i++)
            if (Math.abs(a[i]) > result)
                result = Math.abs(a[i]);

        return result;
    }
    public static double norma(double[][] a) {
        double result = 0;
        for (int i = 0; i < a.length; i++) {
            double sum = 0;
            for (int j = 0; j < a.length; j++)
                sum += Math.abs(a[i][j]);
            if (result < sum)
                result = sum;
        }
```

```
    return result;
}

public static void main(String[] args) {
    int n = 20;
    Random r = new Random();

    System.out.println("Create matrix with diagonal prevalence:");
    double[][] arr = new double[n][n];
    for (int i = 0; i < arr.length; i++) {
        double sum = 0;
        double tmp = -100 + (200) * r.nextDouble();
        int rememberPos = 0;
        for (int j = 0; j < arr.length; j++) {
            if (i != j) {
                while(Math.abs(tmp) > (99.99 / n - 1))
                    tmp = -100 + (200) * r.nextDouble();
                arr[i][j] = Math.rint(tmp * 100.0) / 100.0;
                sum += Math.abs(arr[i][j]);
                tmp = -100 + (200) * r.nextDouble();
            }
            else
                rememberPos = j;
        }
        tmp = -100 + (200) * r.nextDouble();
        while(Math.abs(tmp) < sum)
            tmp = -100 + (200) * r.nextDouble();
        arr[rememberPos][rememberPos] = Math.rint(tmp * 100.0) / 100.0;
    }
    output(arr);

    double[] x = new double[n];
    for (int i = 0; i < x.length; i++) {
        x[i] = -100 + (200) * r.nextDouble();
        x[i] = Math.rint(x[i] * 100.0) / 100.0;
    }

    System.out.println("Calculate vector f:");
    double[] f = multiplicar(arr, x);
    output(f);

    System.out.println("Transpose matrix, ArrT");
    double[][] arrT = transposeMatrix(arr);
    System.out.println("Calculate ArrT*Arr:");
    double[][] arrT_arr = multiplicar(arrT, arr);
    output(arrT_arr);
    System.out.println("Calculate ArrT*f");
    double[] arrT_f = multiplicar(arrT, f);
    output(arrT_f);

    commonStep(arr, f);
    commonStep(arrT_arr, arrT_f);

    mthdSmpl(arr, f, 1e-7, x);

    for (q = 0.1; q < 2; q += 0.1) {
        q = Math.rint(q * 10.0) / 10.0;
        mthdRelax(arrT_arr, arrT_f, 1e-7, x);
    }
    System.out.println("qOptimal = " + qOptimal);
    double tmp = qOptimal;
    for (q = tmp + 0.01; q < tmp + 0.10; q += 0.01) {
        q = Math.rint(q * 100.0) / 100.0;
        mthdRelax(arrT_arr, arrT_f, 1e-7, x);
    }
    System.out.println("qNextOptimal = " + qOptimal);
}

public static double[][] calculateB(double[][] a) {
    int n = a.length;
    double[][] B = new double[n][n];
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++) {
            if (j != i)
                B[i][j] = -(a[i][j] / a[i][i]);
            else
                B[i][j] = 0;
        }
```

```
        return B;
    }
    public static double[] calculateG(double[][] a, double[] f) {
        int n = a.length;
        double[] g = new double[n];
        for (int i = 0; i < n; i++)
            g[i] = f[i] / a[i][i];

        return g;
    }

    private static void commonStep(double[][] arr, double[] f) {
        /** Finding b[i][j] for matrix B */
        System.out.println("Matrix B:");
        double[][] B = calculateB(arr);
        output(B);

        /** Finding g[i] of vector G*/
        System.out.println("Vector g:");
        double[] g = calculateG(arr, f);
        output(g);
    }

    private static void mthdSmpl(double[][] arr, double[] f, double eps, double[] answer) {
        int n = arr.length;
        double[] x = new double[n];

        double[][] B = calculateB(arr);
        double[] g = calculateG(arr, f);

        while(true) {
            k++;
            double[] curX = new double[n];

            for (int i = 0; i < n; i++) {
                for (int j = 0; j < n; j++)
                    if (j != i)
                        curX[i] += B[i][j] * x[j];
                curX[i] += g[i];
            }

            double[] error = new double[n];
            for (int i = 0; i < n; i++)
                error[i] = curX[i] - x[i];

            double[] test = new double[n];
            for (int i = 0; i < n; i++)
                test[i] = curX[i] - answer[i];

            /** Method evaluation
            double rightGrade = (Math.pow(norma(B), k + 1) * norma(g)) / (1 - norma(B));
            System.out.println("Norma(test) = " + norma(test));
            System.out.println("RightGrade = " + rightGrade);
            if (norma(test) <= rightGrade)
                System.out.println("Good");
            else
                System.out.println("Bad"); */

            if (norma(error) <= eps)
                break;

            x = curX;
        }

        /** Calculation k(e) */
        ke = Math.log10((eps * (1 - norma(B))) / norma(g)) / Math.log10(norma(B));

        System.out.println("\nIterations: " + k);
        System.out.println("Norma ||B|| = " + norma(B));
        System.out.println("Norma ||g|| = " + norma(g));
        System.out.println("K(e) iterations: " + ke);
        System.out.println("Calculate vector answers:");
        output(answer, x);
    }

    private static void mthdRelax(double[][] arrT_arr, double[] arrT_f, double eps, double[]
answer) {
        int n = arrT_arr.length;
        double[] x = new double[n];
```

```
        double[][] B = calculateB(arrT_arr);
        double[] g = calculateG(arrT_arr, arrT_f);

        k = 0;
        while (true) {
            k++;
            double[] curX = new double[n];

            for (int i = 0; i < n; i++) {
                for (int j = 0; j < n; j++) {
                    if (j < i)
                        curX[i] += B[i][j] * curX[j];
                    else
                        curX[i] += B[i][j] * x[j];
                }
                curX[i] += g[i];
                curX[i] = (1 - q) * x[i] + q * curX[i];
            }

            double[] error = new double[n];
            for (int i = 0; i < n; i++)
                error[i] = curX[i] - x[i];

            if ((norma(error) / q) <= eps)
                break;

            x = curX;
        }

        if (remIteration >= k) {
            remIteration = k;
            qOptimal = q;
        }

        System.out.println("\nIterations = " + k);
        System.out.println("q  = " + q);
        System.out.println("Calculate vector answers:");
        output(answer, x);
    }
}
```

Результаты

Матрица B(Метод простой итерации):

0.0 0.015712113532691332 -0.011023821591485048 3.8013177901672577E-4 0.042448048656867715 -0.0491637100861632 -
0.005955397871262037 -0.036999493157627975 -0.001520527116066903 -0.00747592498732894 -0.04498226051697922
0.041054232133806386 0.026102382159148504 -0.001393816523061328 0.04738976178408515 0.03826659908768373 -
0.01368474404460213 -0.02331474911302585 0.0486568677141409 0.02280790674100355

0.01492380451918024 0.0 0.01692065160273253 -0.018707304256437202 0.0397267472411981 0.00325801366263794
0.013767735155018392 -0.024592748292170254 -0.006726221755123489 0.03888596952180767 0.011350499211770887 -
0.040357330530740934 -0.04203888596952181 -0.032475039411455596 -0.01471361008933263 0.017235943247503938 -
0.005780346820809248 -0.016605359957961113 0.02932212296374146 -0.027955859169732002

0.05555555555555555 0.03345443536179231 0.0 0.050105964274901604 0.025734181047532546 0.013623978201634877 -
0.05328489252194974 -0.014683620950650922 0.05389040266424463 -0.04859218891916439 0.018468059339993945 -
0.0449591280653951 0.011353315168029064 -0.011504692703602786 0.03330305782621859 0.01422948834392976 -
0.011353315168029064 0.04056917953375719 -0.006055101422948834 -0.014683620950650922

-0.00812202852614897 0.024564183835182253 0.01842313787638669 0.0 0.048930269413629164 0.06656101426307448
0.07567353407290016 0.033874801901743266 -0.049128367670364506 0.05606180665610143 0.04952456418383518
0.06933438985736926 0.04992076069730587 0.009508716323296355 -0.03882725832012678 -0.040808240887480196
0.02971473851030111 -0.028328050713153724 0.046553090332805074 -0.06992868462757527

0.014295925661186563 0.009292351679771266 -0.05932809149392423 -0.05253752680486061 0.0 0.03252323087919943 -
0.03377412437455325 0.027519656897784132 -0.05450321658327376 0.004467476769120801 -0.011615439599714082
0.052894924946390275 -0.01250893495353824 0.045389563974267334 -0.05307362401715511 0.04503216583273767
0.00268048606147248 0.03091493924231594 0.031451036454610434 -0.029664045746962114

0.07603867258949569 0.027436634439508754 -0.02247191011235955 -0.045466422785471645 -0.008100339691664488 0.0
0.03318526260778678 0.053305461196759864 0.024823621635746012 0.05121505095374967 -0.09511366605696367
0.09642017245884504 0.04625032662660047 -0.06715442905670237 -0.03266266004703423 -0.036843480533054605 -
0.0927619545335772 -0.0 -0.07734517899137705 0.006271230729030571

0.07700831024930747 0.03490304709141274 -0.04542936288088642 0.06121883656509695 -0.03905817174515235 -
0.03850415512465374 0.0 -0.04653739612188365 -0.00775623268698061 0.013850415512465374 0.10858725761772853
0.08060941828254847 0.03157894736842105 -0.052908587257617726 -0.0335180055401662 0.027700831024930747
0.03157894736842105 0.045983379501385035 0.07146814404432134 -0.05484764542936288

0.009921587453992638 0.03344535125620099 -0.026564250280044804 0.052008321331413025 -0.00880140822531605
0.016482637221955514 0.05104816770683309 0.0 0.05120819331092975 0.013922227556409025 0.0337654024643943 -
0.05056809089454313 -0.016002560409665547 0.052488398143702986 0.0484877580412866 -0.01728276524243879
0.032805248839814366 0.04240678508561369 -0.004800768122899664 0.02960473675788126

-0.022124839675074816 -0.0102607952116289 -0.034843950406156475 0.03986746472851646 -0.023941855493800773 -
0.017101325352714837 0.034950833689610945 -0.009940145361265498 0.0 0.027255237280889265 0.032492518170158184
0.02201795639162035 0.038691748610517315 0.031958101752885854 -0.010795211628901239 -0.006840530141085934
0.01261222744762719 -0.010474561778537837 0.00470286447199658 -0.028430953398888414

-0.018043478260869564 -0.043043478260869565 0.03826086956521739 0.06 -0.011086956521739131 -0.016304347826086956
-0.02956521739130435 0.05065217391304348 -0.06760869565217391 0.0 0.0508695652173913 -0.051086956521739134 -
0.058478260869565216 0.07152173913043479 0.04 0.002608695652173913 0.0591304347826087 0.015 0.002391304347826087
0.024347826086956525

-0.030426939266386045 -0.006975345760673481 0.023692122669873722 -0.026578472639807574 0.003126879134095009
0.03271196632591702 -0.034155141310883944 0.01190619362597715 -0.008057726999398678 -0.047384245339747444 0.0 -
0.004690318701142514 -0.03740228502705953 -0.0024052916416115455 -0.03812387251954299 -0.013710162357185807
0.04161154539987973 -0.0386049308478653 -0.04606133493686109 0.03463619963920625

0.004862662636351689 0.01471941122355106 0.017742147456958866 0.041792614009725323 0.03679852805887764 -
0.024313313181758445 -0.04468392692863714 0.034958601655933765 -0.0249704297542384 -0.0218162702063346
0.04271257721119726 0.0 -0.0038112761203837556 -0.029964515705086077 -0.04389538704166119 -0.006176895781311604
0.02418188986726245 0.02168484689183861 0.05217505585490866 0.0010513865159679327

0.020514564207185764 -0.03252861838376969 0.0413691488155956 0.007707129094412331 -0.004306925082171596
0.01813442139861725 -0.02482148928935736 0.011560693641618497 -0.0032868638784993762 0.01348747591522158 -
0.01507423778760059 -0.030601836110166612 0.0 -0.036948883599682646 0.0145075371188938 0.030035135441459818
0.042275869885526465 -0.017001020061203673 0.01246741471154936 0.001813442139861725

0.06689287445467765 -0.07513330101793506 -0.019146873485215706 -0.06374212312166748 -0.08167716917111004 -
0.0763451284537082 0.014541929229277752 -0.08288899660688319 -0.06156083373727582 0.09403780901599612 -
0.08046534173533688 0.02932622394571013 -0.05865244789142026 0.0 0.03780901599612216 -0.00896752302472128 -
0.0174503150751333 0.026175472612699955 0.0349006301502666 -0.03853611245758604

-0.016973415132924333 -0.016155419222903887 0.00245398773006135 0.0036809815950920245 -0.016564417177914112 -
0.06584867075664622 0.034764826175869123 -8.179959100204499E-4 0.03619631901840491 -0.07177914110429448
0.028629856850715747 0.049079754601226995 0.06094069529652352 -0.03844580777096115 0.0 0.028425357873210634
0.013496932515337425 -0.040286298568507156 0.028629856850715747 0.0036809815950920245

0.035532994923857864 0.012113520996769727 -0.002884171665897554 -0.034725426857406545 0.018920166128287952
0.019150899861559757 -0.011536686663590217 0.008075680664513151 -0.007498846331333641 0.03980156898938625
0.025265343793262573 0.015574526995846792 -0.001961236732810337 0.008652514997692661 -0.008075680664513151 0.0
0.03691739732348869 -0.015920627595754498 -0.025265343793262573 0.02399630826026765

0.008436311787072243 -0.029586501901140688 0.046459125475285176 0.021269011406844108 -0.029942965779467683 -0.011050380228136883 -0.023407794676806086 0.016397338403041823 0.009624524714828898 0.006653992395437263 -0.002019961977186312 -0.046102661596958174 -0.04657794676806084 0.008555133079847909 -0.036596958174904944 -0.04218155893536121 0.0 0.0439638783269962 -0.0350522813688213 -0.01627851711026616

0.022661630748722506 -0.08864696734059099 -0.0282159520106643 0.07709397911575207 -0.014441235281048657 0.053321484114641195 -0.030215507664963344 0.06642968229282382 0.015774272383914687 -0.003776938458120418 0.0697622750499889 -0.017996000888691403 0.017107309486780718 -0.07487225061097534 0.05043323705843146 -0.053321484114641195 -0.04510108864696734 0.0 -0.041101977338369254 0.03643634747833815

0.0684249354481741 -0.015861305791220952 0.021947620804131315 -0.004426410918480265 -0.021763187015861305 -0.05145702692733309 -0.01014385835485061 1.8443378827001107E-4 -0.01383253412025083 -0.05348579859830321 0.05717447436370343 -0.04094430099594246 -0.050165990409443015 0.028402803393581706 0.02877167097012173 -0.05735890815197344 0.04241977130210254 0.014016967908520842 0.0 -0.0012910365178900776

-0.024172185430463577 -0.033554083885209716 -0.02924944812362031 -0.026490066225165563 -0.03245033112582782 -0.03818984547461369 -0.010596026490066225 0.032229580573951436 -0.017108167770419427 -0.003752759381898455 0.03454746136865342 0.010485651214128035 0.02991169977924945 -0.025607064017660042 -0.004966887417218543 -7.72626931567329E-4 -0.04128035320088301 -0.03995584988962473 8.830022075055189E-4 0.0

Транспонированный вектор g(Метод простой итерации):

-6.4817600101368456 -36.233928533893845 35.8255101422949 -26.129875198098258 17.272675125089354 -26.006111836948 42.708961218836556 -29.490923347735638 -84.65799166310389 46.706706521739136 3.1458953698135894 32.050667630437644 33.637194831689904 -39.500416868637906 37.61540490797547 -57.56683664051685 -89.64394843155893 -16.196718506998444 5.424323127997049 -103.28504966887417

Количество итераций k: 14

Норма ||B|| = 0.9682501211827437

Норма ||g|| = 103.28504966887417

k(epselen): 509

Сравнение векторов точного значения и полученного в результаце метода Якоби:

X = 23.48 Answer = 23.48000000905633

X = 14.16 Answer = 14.159999981567832

X = -18.96 Answer = -18.960000018655165

X = 90.02 Answer = 90.02000001560626

X = 3.84 Answer = 3.8399999942871084

X = 46.79 Answer = 46.79000001891209

X = 82.17 Answer = 82.16999996978463

X = 73.35 Answer = 73.34999999934368

X = 92.79 Answer = 92.7900000120903

X = 80.78 Answer = 80.77999998573114

X = -77.22 Answer = -77.21999999797576

X = -64.71 Answer = -64.71000001445177

X = 30.78 Answer = 30.779999989451355

X = -20.03 Answer = -20.030000006720734

X = 35.31 Answer = 35.30999999890122

X = 16.84 Answer = 16.83999997887005

X = 30.06 Answer = 30.06000001128668

X = -74.82 Answer = -74.81999998559644

X = 35.42 Answer = 35.4200000144816

X = 67.68 Answer = 67.67999998983032

Матрица A^T*A:

6313.94 -223.81 -166.56 19.19 -302.93 192.82 -80.96 177.39 212.33 105.57 491.35 -296.99 -344.28 -124.73 -236.47 -508.46 31.52 136.34 -498.85 45.4

-223.81 9126.91 -313.27 108.24 -342.2 -52.5 -153.58 72.1 172.66 -268.92 -60.85 302.77 651.09 440.99 165.69 -230.72 268.2 345.81 -202.13 527.19

-166.56 -313.27 4444.12 -259.63 91.28 -11.56 262.36 176.85 68.51 116.95 -255.15 33.43 -424.36 65.19 -163.11 -54.14 -241.86 -115.96 -51.01 310.76

19.19 108.24 -259.63 2663.52 37.1 -106.28 -274.53 -277.7 -190.34 -292.81 115.3 -458.61 -170.06 94.1 117.76 347.15 -206.56 -39.53 -92.16 391.67

-302.93 -342.2 91.28 37.1 3213.25 -61.53 149.57 -59.22 380.9 7.93 10.42 -378.41 72.1 -18.99 202.39 -257.48 216.13 -80.12 -1.07 363.94

192.82 -52.5 -11.56 -106.28 -61.53 1564.06 -1.41 -114.84 119.8 -26.46 -84.67 -18.67 -223.12 232.85 184.44 -99.03 238.46 -105.7 225.9 310.06

-80.96 -153.58 262.36 -274.53 149.57 -1.41 1395.66 -163.48 -286.73 93.07 94.5 189.42 187.91 68.46 -28.63 41.09 113.03 -12.54 -49.19 132.17

177.39 72.1 176.85 -277.7 -59.22 -114.84 -163.48 3979.17 -127.48 -177.94 -177.19 -8.19 -20.65 -72.8 -215.65 -20.86 -240.37 -304.12 -10.92 -364.84

212.33 172.66 68.51 -190.34 380.9 119.8 -286.73 -127.48 8821.16 -122.37 -235.33 -76.65 -296.34 -184.02 36.53 112.27 -193.88 76.81 -29.08 405.34

105.57 -268.92 116.95 -292.81 7.93 -26.46 93.07 -177.94 -122.37 2227.6 205.59 253.0 16.6 -306.73 79.96 -303.77 -179.91 -24.87 166.51 -56.37

491.35 -60.85 -255.15 115.3 10.42 -84.67 94.5 -177.19 -235.33 205.59 7043.05 -235.13 386.65 157.6 179.31 -123.2 -235.35 128.1 164.61 -536.13

-296.99 302.77 33.43 -458.61 -378.41 -18.67 189.42 -8.19 -76.65 253.0 -235.13 5924.63 342.03 119.27 112.94 -62.09 159.43 -123.48 -153.48 -105.41

-344.28 651.09 -424.36 -170.06 72.1 -223.12 187.91 -20.65 -296.34 16.6 386.65 342.03 7893.16 370.09 -246.26 -199.51 -32.25 66.11 73.04 -268.22

-124.73 440.99 65.19 94.1 -18.99 232.85 68.46 -72.8 -184.02 -306.73 157.6 119.27 370.09 1799.57 36.95 -62.84 1.91 135.39 -154.89 276.91

-236.47 165.69 -163.11 117.76 202.39 184.44 -28.63 -215.65 36.53 79.96 179.31 112.94 -246.26 36.95 2484.31 7.13 216.66 -1.85 -144.96 59.29

-508.46 -230.72 -54.14 347.15 -257.48 -99.03 41.09 -20.86 112.27 -303.77 -123.2 -62.09 -199.51 -62.84 7.13 7579.92 21.58 209.35 402.39 -174.1

31.52 268.2 -241.86 -206.56 216.13 238.46 113.03 -240.37 -193.88 -179.91 -235.35 159.43 -32.25 1.91 216.66 21.58 7177.97 -217.89 130.1 464.58

136.34 345.81 -115.96 -39.53 -80.12 -105.7 -12.54 -304.12 76.81 -24.87 128.1 -123.48 66.11 135.39 -1.85 209.35 -217.89 2104.81 40.13 240.2

-498.85 -202.13 -51.01 -92.16 -1.07 225.9 -49.19 -10.92 -29.08 166.51 164.61 -153.48 73.04 -154.89 -144.96 402.39 130.1 40.13 3039.43 -38.34

45.4 527.19 310.76 391.67 363.94 310.06 132.17 -364.84 405.34 -56.37 -536.13 -105.41 -268.22 276.91 59.29 -174.1 464.58 240.2 -38.34 8270.36

Транспонированный вектор A^T*f:

-60037.182597000006 -374213.934452 120921.46846800014 -88419.703499 4037.7513170000275 -74181.47844600001 94815.52486300003 -101280.416482 -750735.885736 178669.167523 66316.91815599999 189582.693212 272345.008816 -51631.41319499999 54337.597237999995 -480817.8074620001 -671826.1926069999 -62441.351489999994 -20309.740370000007 -876667.5672919998

Матрица B(Метод Релаксации):

0.0 0.03544696338577814 0.026379724862763982 -0.003039306676971907 0.047977966214439796 -0.030538776104936063 0.012822421499095652 -0.02809497714580753 -0.033628764289809536 -0.016720146216150297 -0.07781987158572937 0.04703719072401702 0.054526967313595 0.019754701501756435 0.037452050542133755 0.0805297484613411 -0.004992128528304039 -0.021593489960310047 0.07900771942717226 -0.00719043893353437

0.02452199046557347 0.0 0.03432377442091573 -0.011859435449675739 0.03749352190390833 0.005752220631078865 0.016827162752782707 -0.007899716333348307 -0.01891768407927765 0.029464517563994826 0.006667097626688551 -0.03317333029469996 -0.0713373967750312 -0.04831755763998988 -0.018154008311684896 0.025279092266714584 -0.0293856299667686 -0.03788905555111204 0.022146597260189922 -0.057762156085685086

0.03747873594772418 0.07049089583539599 0.0 0.058421014734075585 -0.020539499383454993 0.002601189886861741 -0.05903530957759917 -0.03979415497331305 -0.015415875358901201 -0.026315671043986213 0.057412941144703564 -0.007522299127836332 0.0954879706218554 -0.014668820823920146 0.03670242927733725 0.012182389314419954 0.054422472840517364 0.02609290478204909 0.01147808790041673 -0.06992610460563621

-0.007204751606896138 -0.040637952784285454 0.09747627200096114 0.0 -0.013928936144650688 0.0399020844596624 0.10307037304018742 0.10426052742235838 0.07146182495344507 0.10993347149636572 -0.04328858052501952 0.17218192467111193 0.06384784045173304 -0.035329188442361985 -0.04421217036102601 -0.13033504535351714 0.07755151078272361 0.01484126268997417 0.03460082897819427 -0.147049768727098

0.09427526647475298 0.1064965377732825 -0.028407375709950985 -0.011545942581498483 0.0 0.019148836847428616 -0.04654788765268809 0.01842993853575041 -0.11854041857931999 -0.0024679063253715086 -0.0032428226873103556 0.11776550221738116 -0.022438341243289504 0.005909904302497471 -0.06298607329028243 0.0801307087839415 -0.06726211779351124 0.024934256593791334 3.329961876604684E-4 -0.11326227339920641

-0.12328171553520964 0.03356648721916039 0.00739102080482846 0.06795135736480698 0.03933992302085598 0.0 9.014999424574505E-4 0.07342429318568341 -0.07659552702581743 0.016917509558456837 0.05413475186373925 0.011936882216794753 0.14265437387312507 -0.1488753628377428 -0.11792386481337033 0.06331598532025626 -0.1524621817577331 0.06758052760124292 -0.14443179929158728 -0.1982404767080547

0.05800839746069959 0.11004112749523523 -0.187982746514194 0.19670263531232532 -0.10716793488385422 0.0010102747087399509 0.0 0.11713454566298381 0.20544401931702563 -0.06668529584569306 -0.06770990069214565 -0.1357207342762564 -0.13463880887895333 -0.04905206139030995 0.020513592135620422 -0.029441267930584814 -0.08098677328289125 0.008984996345814882 0.0352449737041973 -0.09470071507387184

-0.04457964851966616 -0.018119356549230112 -0.04444394182706444 0.06978842321388631 0.014882500621989007 0.028860289959966527 0.04108394464172176 0.0 0.032036831801606866 0.044717868299167914 0.04452938678166552 0.0020582181711261394 0.005189524448565907 0.018295272632232348 0.05419471899918828 0.005242299273466577 0.06040706981606717 0.07642799880377063 0.0027442908948348524 0.09168746246076442

-0.024070530406431808 -0.019573389440844514 -0.007766552244829479 0.021577660987897283 -0.04318026200635744 -0.013580980279237652 0.03250479528769459 0.01445161407343252 0.0 0.013872325181722132 0.02667789723800498 0.00868933337565581 0.03359422116819103 0.020861201928091093 -0.004141178711189911 -0.012727351051335652 0.02197896875240898 -0.008707471579701536 0.0032966185853107754 -0.04595087267434215

-0.04739181181540671 0.1207218531154606 -0.05250044891362902 0.13144639971269528 -0.0035598850781109713 0.01187825462381038 -0.0417803914526845 0.07987969114742324 0.05493356078290537 0.0 -0.09229215298976477 -0.11357514814149758 -0.007451966241695099 0.13769527742862275 -0.03589513377626145 0.1363664930867301 0.08076405099658826 0.011164481953672114 -0.07474860836775005 0.02530526126773209

-0.06976380971312145 0.008639722847345965 0.03622720270337425 -0.01637074846834823 -0.0014794726716408375 0.012021780336643926 -0.013417482482731202 0.025158134614975045 0.033413080980541104 -0.02919047855687522 0.0 0.03338468419221786 -0.05489809102590497 -0.022376669198713624 -0.025459140571201397 0.017492421607116237 0.033415920659373426 -0.01818814292103563 -0.023371976629443212 0.07612185061869502

0.050128024872439295 -0.05110361322141636 -0.005642546454377741 0.07740736552324787 0.0638706552139121 0.003151251639342879 -0.031971616792947403 0.001382364805903491 0.01293751677319934 -0.04270308863169514 0.03968686652162245 0.0 -0.05773018737035055 -0.0201312149450683 -0.01906279379471798 0.010479979340482021 -0.02690969731443145 0.020841807842852635 0.025905415190484466 0.01779182835046239

0.043617511871037705 -0.08248787557834886 0.053763004930851524 0.02154523663526395 -0.00913449112902817 0.028267512631189536 -0.023806688322547623 0.002616189206857583 0.03754389876804727 -0.0021030867231881783 -0.04898545069401861 -0.043332454935665815 0.0 -0.046887431649681496 0.031199164846525344 0.025276315189353818 0.00408581607366378 -0.008375606221082557 -0.009253581582027985 0.03398132053575501

0.06931100207271737 -0.24505298487972127 -0.036225320493228935 -0.05229026934212062 0.010552520879987997 -0.12939202142734096 -0.03804242124507521 0.040454108481470574 0.10225776157637659 0.17044627327639383 -0.087576476602744 -0.06627694393660707 -0.2056546841745528 0.0 -0.020532682807559585 0.03491945298043422 -0.0010613646593352856 -0.07523463938607555 0.08607056130075517 -0.15387564807148377

0.09518538346663662 -0.06669457515366439 0.06565605741634499 -0.0474014917623002 -0.08146728870390571 -0.07424194243069504 0.011524326674207325 0.08680478684222179 -0.014704284086929571 -0.032185999331806416 -0.07217698274369946 -0.04546131521428485 0.09912611550088354 -0.014873345113935057 0.0 -0.00287001219654552 -0.0872113383595445 7.446735713336903E-4 0.05835020589217932 -0.023865781645607834

0.06707986363972179 0.030438315971672524 0.0071425555942542926 -0.04579863639721791 0.03396869623953815 0.013064781686350252 -0.005420901539857941 0.002752007936759227 -0.014811501968358504 0.04007562085087969 0.016253469693611543 0.008191379328541727 0.02632085826763343 0.008290324963851861 -9.406431730150186E-4 0.0 -0.0028469957466569565 -0.027619025002902406 -0.053086312256593736 0.02296858014332605

-0.004391213671832008 -0.03736432445384976 0.033694763282655124 0.028776938326574224 -0.030110184355744033 -0.03322109175714025 -0.015746791920278294 0.033487183702355955 0.027010422166712872 0.025064189457464994 0.0327878216264487 -0.022211015092010695 0.004492913734663143 -2.660919452157086E-4 -0.030184021387662525 -0.0030064210354738175 0.0 0.03035537902777526 -0.018124901608672087 -0.06472303450697063

-0.06477544291408727 -0.16429511452340118 0.05509285873784332 0.018780792565599745 0.038065193532907965 0.05021830949111797 0.0059577824126643255 0.14448810106375398 -0.036492605033233406 0.011815793349518485 -0.06086060024420256 0.05866562777637887 -0.03140901078957246 -0.06432409576161269 8.789391916610051E-4 -0.09946265933742238 0.10352003268703588 0.0 -0.01906585392505737 -0.11411956423620184

0.1641261683934159 0.06650260081660048 0.016782752029163364 0.030321474750199872 3.5203969165271124E-4 -0.07432314611621259 0.016183955544296136 0.0035927789092033706 0.00956758339557088 -0.05478329818419901 -0.05415818097472224 0.050496310163418795 -0.024030821568517786 0.05096021293466209 0.04769315299250189 -0.13238995469545276 -0.04280407839627825 -0.013203133482264769 0.0 0.012614207269126121

-0.005489482924564347 -0.0637445044713894 -0.03757514787748054 -0.04735827702784401 -0.04400533954991076 -0.03749050827291678 -0.015981166478847352 0.044114161898635б -0.04901167591253575 0.006815906441799389 0.0648254731353895 0.012745515310095327 0.03243147819441959 -0.033482218428218365 -0.007168974506551105 0.021051078792217022 -0.056174096411764414 -0.02904347573745278 0.00463583205567835 0.0

Транспонированный вектор g(Метод Релаксации):

-9.508671700554647 -41.0011640798474 27.209316685418067 -33.1965607538145 1.2565942011981723 -47.428793298211076 67.935976429073 -25.452648789069077 -85.10625424955448 80.20702438633508 9.415937435628027 31.999077277737175 34.50392603418656 -28.690972396183525 21.87230950968277 -63.43309790367182 -93.59556986264917 -29.66602757018448 -6.6820885396274985 -106.00113747067839

Вектора полученных значений:

Шаг q: 0.1, Итераций: 313

23.480000093288673 14.159999927980119 -18.960000172012144 90.02000009104606 3.840000060090386 46.79000008192024 82.17000009858073 73.3499999078971 92.78999988401247 80.77999999777896 -77.21999989413878 -64.70999985747746 30.779999957135956 -20.029999987369656 35.309999961801935 16.840000171104997 30.060000039394303 -74.81999986483461 35.42000008125571 67.68000014244123

Шаг q: 0.2, Итераций: 149

23.4800000806768 14.15999993749097 -18.960000151250124 90.02000007460862 3.8400000502141 46.790000073449 82.17000008215939 73.34999991894897 92.78999990105686 80.78000000218377 -77.21999991110205 -64.70999988081991 30.77999996277406 -20.029999988275037 35.30999996507471 16.840000144204613 30.06000003146731 -74.81999988561758 35.420000067513 67.68000011576873

Шаг q: 0.3, Итераций: 94

23.480000071057287 14.159999944813018 -18.960000135959188 90.02000006181083 3.840000042593672 46.79000006723257 82.17000006908746 73.3499999273644 92.7899999142131 80.78000000593897 -77.21999992423022 -64.7099998989548 30.779999967060327 -20.02999998895207 35.309999967383185 16.840000123556546 30.06000002506949 -74.81999990161385 35.42000005679706 67.68000009521472

Шаг q: 0.4, Итераций: 67

23.48000005199924 14.159999959377203 -18.960000102047143 90.02000004260076 3.840000302257515 46.79000005128437 82.17000004764782 73.34999994632236 92.78999993837442 80.78000000799418 -77.21999994695143 -64.70999992942093 30.779999976150393 -20.029999991518515 35.30999997452908 16.84000008832146 30.060000015644533 -74.81999992973613 35.4200000392421 67.68000006465603

Шаг q: 0.5, Итераций: 50

23.480000046437556 14.159999963685063 -18.96000009402988 90.02000003516176 3.8400000258720453 46.790000048076294 82.17000003924595 73.34999995153683 92.78999994613656 80.78000001099846 -77.21999995501776 -64.7099999403126 30.7799999788444 -20.029999991980887 35.30999997558402 16.84000007675533 30.060000011336086 -74.81999993895596 35.42000003289848 67.68000005284142

Шаг q: 0.6, Итераций: 39

23.480000032644362 14.159999974156616 -18.960000068710357 90.02000002268917 3.8400000175642597 46.79000003604734 82.1700000246691 73.34999996597881 92.78999996290054 80.78000001170416 -77.21999997090359 -64.70999996086022 30.77999998568577 -20.02999999411413 35.30999998136393 16.840000053077233 30.060000005439388 -74.81999995823375 35.4200000215689 67.68000003331701

Шаг q: 0.7, Итераций: 31

23.48000002108519 14.159999982702534 -18.96000004662045 90.02000001285509 3.840000108666985 46.79000002582723 82.17000001352278 73.34999997808211 92.78999997612924 80.78000001152037 -77.21999998354923 -64.70999997671285 30.77999999135933 -20.029999995953382 35.309999986757504 16.840000034186957 30.060000001634535 -74.81999997369593 35.420000013293176 67.68000001847092

Шаг q: 0.8, Итераций: 24

23.480000022357043 14.159999980131856 -18.960000053148136 90.02000001000323 3.840000010513947 46.790000032883725 82.17000001077025 73.34999997605043 92.78999997347394 80.78000001909774 -77.21999998589945 -64.70999997722807 30.77999999149055 -20.029999995078914 35.30999998420064 16.84000003682065 30.059999999836215 -74.81999997262398 35.42000001454999 67.68000001529815

Шаг q: 0.9, Итераций: 19

23.480000007506085 14.159999989408016 -18.96000002305196 90.01999999828077 3.840000002159986 46.79000002096097 82.17000000124598 73.3499999878339 92.78999998580768 80.78000001425856 -

77.21999999713407 -64.70999999078171 30.779999996584547 -20.029999996970727 35.309999993052756 16.84000001552425 30.060000000070122 -74.81999998934916 35.42000000863726 67.68000000236002

Шаг q: 1.0, Итераций: 14

23.47999998815048 14.160000004174435 -18.959999987706723 90.01999998626832 3.8399999898558734 46.79000000425853 82.16999999384916 73.34999999464347 92.78999999880207 80.77999999467366 -77.21999999986086 -64.70999999951944 30.779999996939324 -20.029999998776503 35.31000000537466 16.839999991254956 30.060000004228904 -74.82000000526432 35.42000000312401 67.67999999654242

Шаг q: 1.1, Итераций: 13 - Оптимальный

23.480000002078683 14.160000000728003 -18.960000002585367 90.02000000354516 3.84000000688523 46.78999999629114 82.17000000099762 73.35000000032055 92.79000000122778 80.77999999843048 -77.21999999803862 -64.70999999985327 30.780000000155663 -20.030000000038985 35.309999999979595 16.840000001237012 30.05999999943953 -74.81999999889057 35.41999999874016 67.68000000125512

Шаг q: 1.2, Итераций: 16

23.47999999484042 14.160000006250929 -18.959999988315946 90.01999999695525 3.839999998870984 46.790000001887975 82.17000000136485 73.35000000002798 92.79000000135126 80.77999999252448 -77.22000000038918 -64.71000000049978 30.77999999846344 -20.03000000231608 35.309999998324685 16.83999999607407 30.06000000116654 -74.82000000336691 35.4200000022812 67.6800000010642

Шаг q: 1.3, Итераций: 20

23.479999997597222 14.159999999168651 -18.960000005317916 90.020000009107 3.840000027958416 46.79000000193079 82.17000001170544 73.34999999591848 92.79000000079391 80.78000000084215 -77.22000001201694 -64.70999999505568 30.780000004139477 -20.02999999967284 35.30999999319882 16.83999999600476 30.059999999329058 -74.82000000622179 35.42000000186509 67.68000000086008

Шаг q: 1.4, Итераций: 25

23.480000000769557 14.160000000028486 -18.95999999498236 90.02000001109553 3.8399999873987993 46.790000021772116 82.1699999938539 73.34999999746327 92.7899999918091 80.77999999031829 -77.21999999173646 -64.71000000878047 30.779999986988138 -20.030000007000446 35.309999985471904 16.8399999994558 30.06000000751682 -74.81999998994418 35.42000000570539 67.67999999137155

Шаг q: 1.5, Итераций: 31

23.48000002807305 14.16000002278801 -18.960000061291097 90.02000000090618 3.840000206176603 46.789999995779866 82.17000002660998 73.34999997152872 92.78999997908926 80.7799999972977 -77.22000002061567 -64.7099999929168 30.779999993765383 -20.030000021741444 35.30999996204159 16.840000057282353 30.060000002993547 -74.82000002714288 35.41999999523351 67.67999998645108

Шаг q: 1.6, Итераций: 42

23.479999960021885 14.160000010803094 -18.96000000936012 90.01999999127392 3.8400000458336265 46.78999997288046 82.17000000412622 73.34999998712539 92.79000000068235 80.77999994944462 -77.21999999246565 -64.70999998327409 30.78000001234771 -20.03000002973706 35.310000005636724 16.840000044463714 30.059999994943265 -74.8199999957985 35.41999998561007 67.68000001420434

Шаг q: 1.7, Итераций: 59

23.47999994074579 14.159999991234516 -18.959999957191787 90.01999997941112 3.8399999623390033 46.790000043381525 82.17000000162452 73.35000001233689 92.78999998986656 80.78000004219655 -77.22000001557637 -64.71000002825014 30.779999983171482 -20.029999982562884 35.310000025605746 16.83999993622072 30.06000004150409 -74.8200000272164 35.42000004013309 67.67999995510296

Шаг q: 1.8, Итераций: 93

23.47999995144888 14.160000023211373 -18.959999915325213 90.0199999908585 3.8400000390142477 46.79000000771866 82.16999995123804 73.35000002449308 92.78999999547172 80.78000003138386 -77.22000002512851 -64.7100000241526 30.780000024506617 -20.029999969512232 35.31000008412422 16.83999997911522 30.05999997117187 -74.82000001332321 35.420000028069026 67.68000002829457

Шаг q: 1.9, Итераций: 197

23.47999999064517 14.160000038469523 -18.95999991751242 90.01999997202276 3.840000040225416 46.78999997159764 82.16999993267565 73.34999999252354 92.78999997897863 80.78000000942599 -77.22000001783148 -64.71000003135069 30.780000035625715 -20.029999994811252 35.31000004797477 16.839999978842474 30.05999997213201 -74.8200000601401 35.41999998624895 67.68000004089633

Шаг q: 1.11, Итераций: 13 - Оптимальный

23.48000000588829 14.159999996913275 -18.960000008998655 90.02000000611139 3.8400000031490378 46.78999999734156 82.17000000199506 73.34999999967991 92.78999999983276 80.78000000325584 -77.21999999916503 -64.70999999929506 30.780000001389105 -20.029999999375693 35.3099999989437 16.840000004011035 30.059999998814867 -74.81999999750674 35.41999999886402 67.68000000070073

Шаг q: 1.12, Итераций: 13

23.4800000075123 14.15999999139935 -18.960000013832612 90.02000000605177 3.840000003685205 46.789999999707824 82.17000000125392 73.34999999824562 92.78999999785202 80.78000001047744 -77.22000000272672 -64.70999999884376 30.780000003314495 -20.029999997969774 35.30999999917762 16.840000004863555 30.059999998917935 -74.81999999750258 35.419999999112804 67.67999999897971

Шаг q: 1.13, Итераций: 13

23.480000004698244 14.159999986989058 -18.960000013475593 90.0200000023849 3.8399999998217362 46.79000000055447 82.16999999809664 73.34999999618537 92.78999999718218 80.78000001750605 -77.22000000885171 -64.70999999862902 30.78000000577836 -20.029999995826394 35.31000000189148 16.840000001285873 30.06000000031994 -74.81999999993445 35.4199999982373 67.67999999687156

Шаг q: 1.14, Итераций: 14

23.479999995901622 14.16000000417105 -18.95999999204566 90.01999999634653 3.8399999983688327 46.790000001457315 82.16999999966528 73.35000000057708 92.79000000080288 80.77999999518042 -77.21999999967483 -64.71000000057228 30.779999998427986 -20.030000001180362 35.3099999999216 16.839999997156724 30.06000000076962 -74.82000000178448 35.420000001022586 67.68000000023775

Шаг q: 1.15, Итераций: 14

23.47999999492366 14.160000007943651 -18.95999998806236 90.01999999647384 3.8399999989661193 46.79000000123779 82.17000000070033 73.35000000152681 92.79000000159006 80.77999998984014 -77.21999999690817 -64.71000000091003 30.779999996837848 -20.030000002605508 35.30999999893326 16.839999997364277 30.06000000058595 -74.82000000163438 35.42000000179106 67.68000000141987

Шаг q: 1.16, Итераций: 14

23.479999995997485 14.160000011164069 -18.95999998587283 90.0199999986528 3.84000002666388 46.79000000291399 82.17000000337244 73.35000000279494 92.79000000122498 80.77999998343171 -77.21999999199844 -64.71000000124329 30.77999999454503 -20.03000000486639 35.30999999618391 16.840000000582236 30.05999999959887 -74.82000000007184 35.42000000406102 67.68000000275161

Шаг q: 1.17, Итераций: 15

23.480000004105843 14.159999995009585 -18.96000000907469 90.02000000298898 3.840000001087064 46.7899999984447 82.16999999945253 73.34999999957016 92.78999999898444 80.78000000596805 -77.220000000353 -64.70999999946808 30.78000000162752 -20.02999999831678 35.310000000860455 16.840000002836984 30.05999999918917 -74.81999999790429 35.41999999849425 67.67999999934375

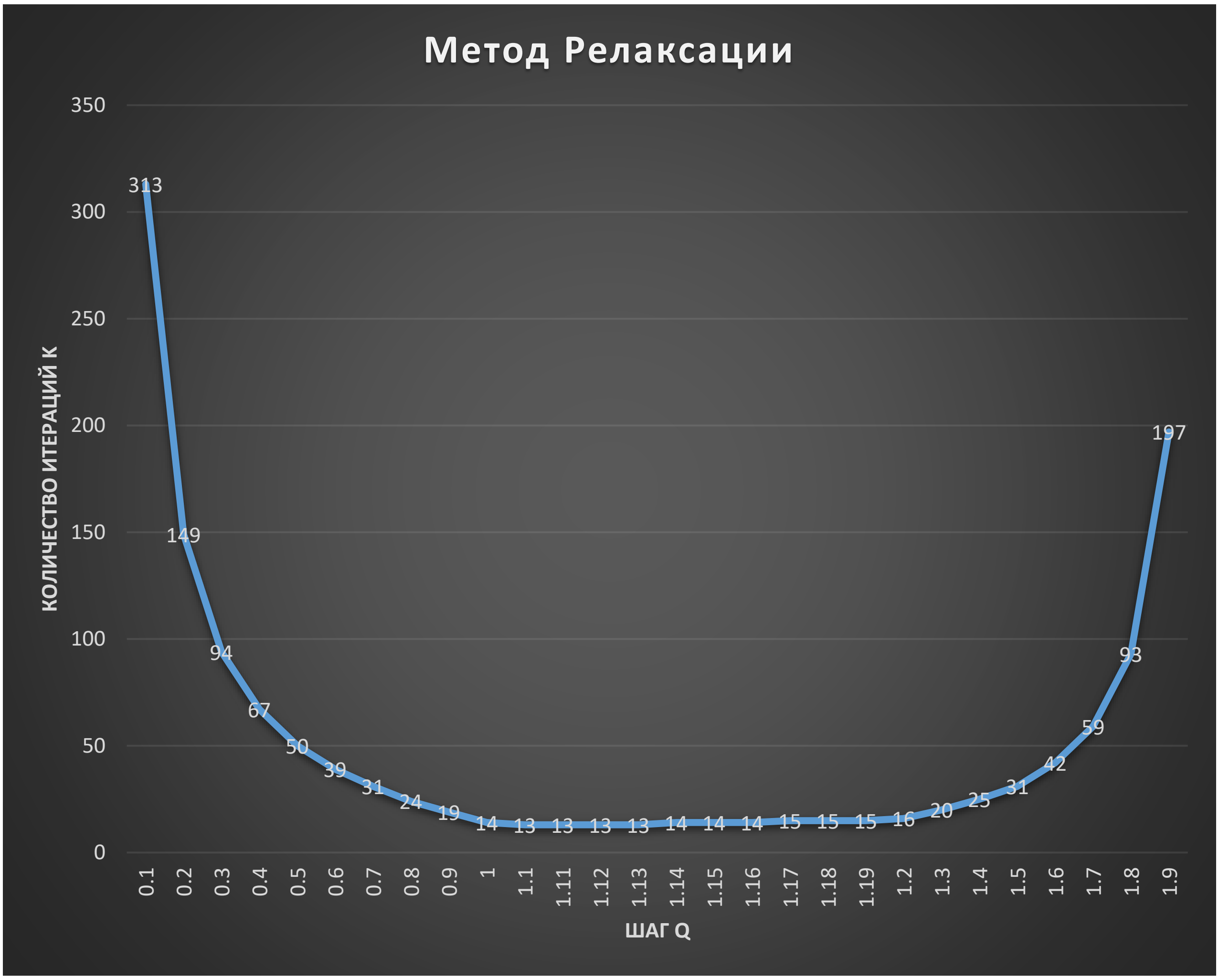
Шаг q: 1.18, Итераций: 15

23.480000005692997 14.159999991081262 -18.960000014518464 90.02000000305671 3.840000000280945 46.78999999795006 82.16999999816784 73.34999999896112 92.7899999983907 80.78000001177996 -77.22000000280562 -64.70999999911918 30.780000003170848 -20.029999996572403 35.31000000221569 16.840000002759265 30.05999999920756 -74.81999999759071 35.419999997035646 67.67999999812747

Шаг q: 1.19, Итераций: 15

23.480000006425726 14.159999986742735 -18.960000020396066 90.02000000144226 3.839999996719483 46.78999999572275 82.16999999521869 73.34999999824583 92.78999999865235 80.78000002008788 -77.2200000074503 -64.7099999987143 30.780000005535506 -20.02999999367588 35.31000000525555 16.84000000022241 30.05999999980771 -74.81999999817681 35.41999999374747 67.67999999652903

График результатов



q	k	q	k	q	k
0.1	313	1.1	13	1.2	16
0.2	149	1.11	13	1.3	20
0.3	94	1.12	13	1.4	25
0.4	67	1.13	13	1.5	31
0.5	50	1.14	14	1.6	42
0.6	39	1.15	14	1.7	59
0.7	31	1.16	14	1.8	93
0.8	24	1.17	15	1.9	197
0.9	19	1.18	15		
1.0	14	1.19	15		