|  |  |  |
| --- | --- | --- |
|  | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования «Новосибирский государственный технический университет» | | |
|  | | |
| Кафедра прикладной математики | | |
| Практическое работа № 2 | | |
| по дисциплине «Численные методы» | | |
| **ИТЕРАЦИОННЫЕ МЕТОДЫ РЕШЕНИЯ СЛАУ** | | |
|  | | |
|  |  |  |
| Группа ПМ-01 | будник светлана |
|  | самсонов семён |
| Вариант 8 |  |
|  |  |
| Преподаватель | Задорожный александр геннадьевич |
|  |  |
|  |  |
| Новосибирск, 2022 | | |

# Цели работы

Разработать программы решения СЛАУ методами Якоби, Гаусса-Зейделя, блочной релаксации с хранением матрицы в диагональном формате. Исследовать сходимость методов для различных тестовых матриц и её зависимость от параметра релаксации.

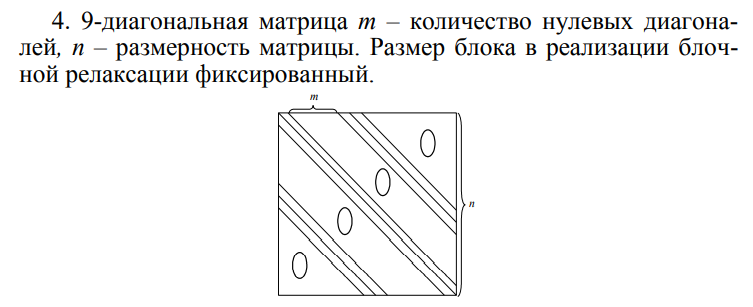
Изучить возможность оценки порядка числа обусловленности

матрицы путем вычислительного эксперимента.

# Входные данные

**Вариант 12:**

Матрица из варианта 4. Размер блока в реализации блочной релаксации переменный. Исследовать зависимость скорости сходимости от размеров блока.



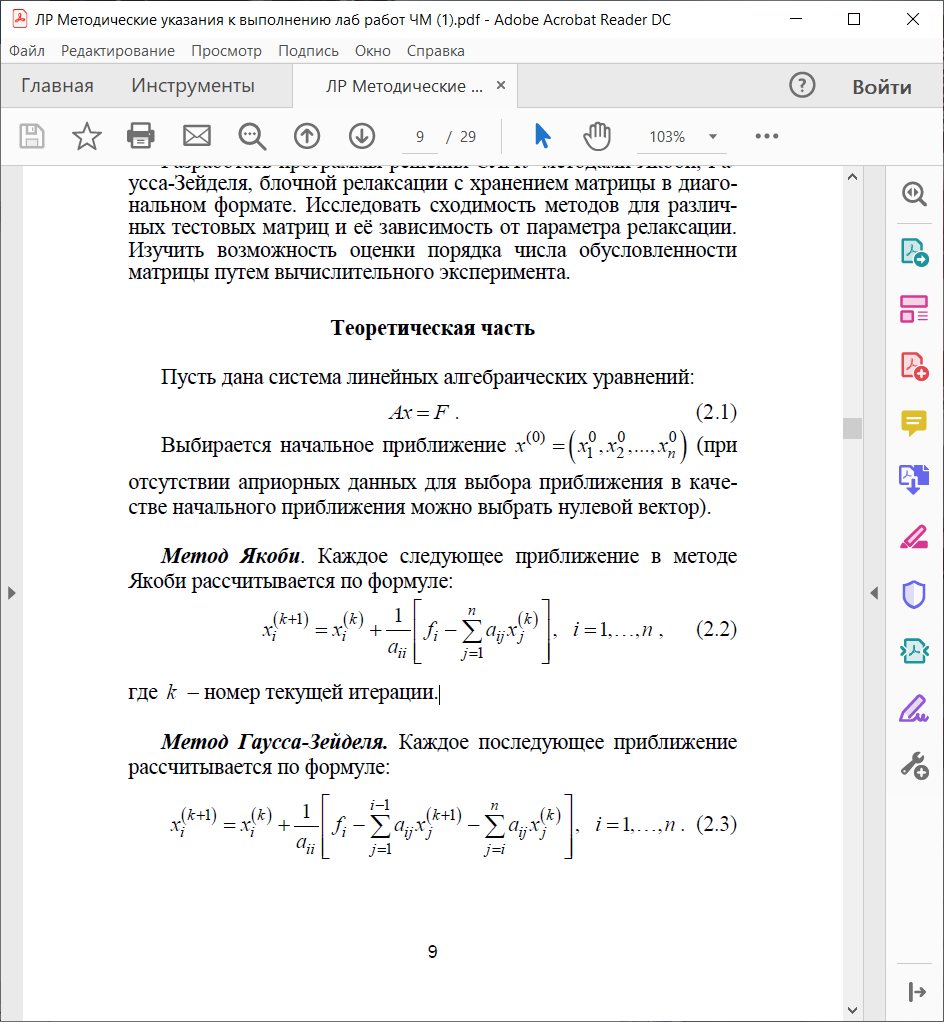
# Анализ

**Входные данные:** система линейных алгебраических уравнений **Ax = F**.

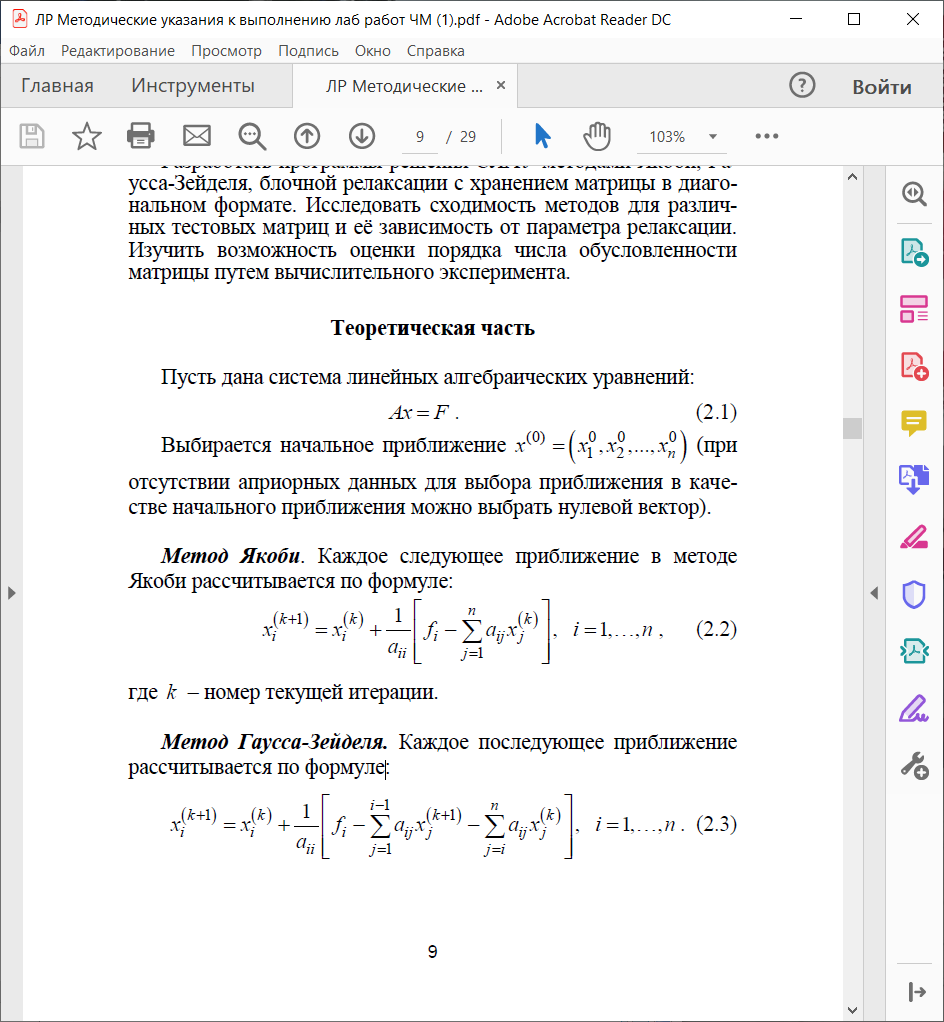
**Выходные данные:** вектор неизвестных **.**

Мы выбираем нулевое начальное приближение вектора, и находили следующее приближение.

**Метод Якоби**



**Метод Гаусса-Зейделя**.



Для ускорения вычислений используется ***метод релаксации***:

**Метод Якоби** с параметром релаксации

**Методе Гаусса-Зейделя** с параметром релаксации

# Текст программы

Файл matrix\_IO.h:

#pragma once

#include <fstream>

#include <iostream>

#include <vector>

/// <summary>

/// Функция считывания вектора данных с файла

/// </summary>

/// <typeparam name="T"> - тип данных, который необходимо считать </typeparam>

/// <param name="filePath"> - путь до файла</param>

/// <param name="count"> - число элементов, необходимое считать </param>

/// <returns> считанный вектор </returns>

template<typename T>

std::vector<T> GetVectorFromFile(const char\* filePath, const size\_t count)

{

   std::vector<T> arr(count);

   auto fin = std::ifstream(filePath);

   for (int i = 0; i < count; i++)

   {

      fin >> arr[i];

   }

   fin.close();

   return arr;

}

/// <summary>

/// Функция считывания вектора данных с файла

/// </summary>

/// <typeparam name="T"> - тип данных, который необходимо считать </typeparam>

/// <param name="file"> - предварительно открытый файл</param>

/// <param name="count"> - число элементов, необходимое считать </param>

/// <returns> считанный вектор </returns>

template<typename T>

std::vector<T> GetVectorFromFile(std::istream& file, const size\_t count)

{

   std::vector<T> arr(count);

   for (int i = 0; i < count; i++)

   {

      file >> arr[i];

   }

   return arr;

}

/// <summary>

/// Вывод массива в произвольный поток

/// </summary>

/// <typeparam name="T"> - тип данных для вывода</typeparam>

/// <param name="\_arr"> - указатель на начало массива </param>

/// <param name="\_size"> - длина массива </param>

/// <param name="\_out">  - поток вывода (cout по умолчанию)</param>

template <typename T>

void PrintArray(const T\* \_arr, const size\_t \_size, const int \_precision = 7, std::ostream& \_out = std::cout)

{

   auto prevPrec = \_out.precision();

   auto prevFlags = \_out.flags();

   \_out.precision(\_precision);

   \_out.setf(std::ios\_base::fixed);

   for (int i = 0; i < \_size; i++)

   {

      \_out << \_arr[i] << std::endl;

   }

   \_out.precision(prevPrec);

   \_out.flags(prevFlags);

}

/// <summary>

/// Вывод массива в произвольный поток

/// </summary>

/// <typeparam name="T"> - тип данных для вывода</typeparam>

/// <param name="\_arr"> - указатель на начало массива </param>

/// <param name="\_size"> - длина массива </param>

/// <param name="\_out">  - поток вывода (cout по умолчанию)</param>

template <typename T>

void PrintArray(const T\* \_arr, const size\_t \_size, const int \_precision, std::ostream&& \_out)

{

   auto prevPrec = \_out.precision();

   auto prevFlags = \_out.flags();

   \_out.precision(\_precision);

   \_out.setf(std::ios\_base::fixed);

   for (int i = 0; i < \_size; i++)

   {

      \_out << \_arr[i] << std::endl;

   }

   \_out.precision(prevPrec);

   \_out.flags(prevFlags);

}

Файл main.cpp:

#include <iostream>

#include <fstream>

#include <vector>

#include <iomanip>

#include <format>

#include "matrix\_IO.h"

using namespace std;

vector<double> matrixA[9];  // Вектор диагоналей матрицы А (от верхней к нижней)

vector<double> vectorX;

vector<double> vectorB;

vector<double> nextVectorX;      // Вектор Х для следующих операций

vector<int64\_t> diagsShift;

size\_t n;                        // Размер матрицы

int64\_t m;                       // Число нулевых диагоналей

size\_t maxIterations;            // Максимальное число итераций

double maxDif;                   // Максимальная невязка

double w;                        // Коэф. релаксации

void ReadData()

{

   ifstream matrixParams("./iofiles/matrixParams.txt");

   matrixParams >> n >> m;

   matrixParams.close();

   ifstream solversParams("./iofiles/solversParams.txt");

   solversParams >> maxIterations >> maxDif >> w;

   solversParams.close();

   diagsShift = { 4 + m, 3 + m, 2 + m, 1, 0, -1, -2 - m, -3 - m, -4 - m };

   vectorB.resize(n);

   vectorX.resize(n);

   nextVectorX.resize(n);

   ifstream matrixAFile("./iofiles/matrixA.txt");

   // Данные читаются с верхней диагонали к нижней

   // причём нижние диагонали должны быть прижатыми в правый край,

   // а верхние диагонали в левый край. Пустые элементы заполнить нулями

   for (int i = 0; i < 9; i++)

   {

      matrixA[i] = GetVectorFromFile<double>(matrixAFile, n);

   }

   matrixAFile.close();

   vectorB = GetVectorFromFile<double>("./iofiles/vectorB.txt", n);

   vectorX = GetVectorFromFile<double>("./iofiles/initialX.txt", n);

   for (auto& elem : nextVectorX)

   {

      elem = 0;

   }

}

double Norm(vector<double> X)

{

   double norma = 0;

   for (size\_t i = 0; i < n; i++)

   {

      norma += X[i] \* X[i];

   }

   norma = sqrt(norma);

   return norma;

}

size\_t Iterations(vector<double>& vectorXPrev, vector<double>& vectorXNext, double& resultDif, bool debugOutput = false)

{

   size\_t k;

   double sum = 0;

   double norm = Norm(vectorB);     // Норма

   double dif = norm;                  // Невязка, делаем её по умолчанию равной norm, чтобы зашло в цикл

   // Итерационный цикл

   for (k = 1; (k <= maxIterations) && (dif > maxDif); k++)

   {

      dif = 0;

      // Пробегаемся по всем строкам матрицы А

      for (size\_t i = 0; i < n; i++)

      {

         sum = 0;

         int64\_t indX;

         // Пробегаемся по всем столбцам матрицы А

         for (size\_t j = 0; j < 4; j++)

         {

            indX = i + diagsShift[j];

            if (indX < n)

            {

               sum += vectorXPrev[indX] \* matrixA[j][i];

            }

         }

         sum += vectorXPrev[i] \* matrixA[4][i];

         for (size\_t j = 5; j < 9; j++)

         {

            indX = i + diagsShift[j];

            if (indX >= 0)

            {

               sum += vectorXPrev[indX] \* matrixA[j][i];

            }

         }

         vectorXNext[i] = vectorXPrev[i] + (vectorB[i] - sum) \* w / matrixA[4][i];

         dif += (vectorB[i] - sum) \* (vectorB[i] - sum);       // В данный момент sum = скалярное произведение i-ой строки А на вектор Х

      }

      dif = sqrt(dif) / norm;                               // относительная невязка

      // Выводим на то же место, что и раньше (со сдвигом каретки)

      if (debugOutput)

      {

         cout << format("\rИтерация: {0:<10} относительная невязка: {1:<15.3e}", k, dif);

      }

      if (isinf(dif))

      {

         break;

      }

      swap(vectorXNext, vectorXPrev);

   }

   if (debugOutput)

   {

      cout << endl;

      if (isinf(dif))

      {

         cout << "Выход по переполнению метода" << endl << endl;

      }

      else if (k > maxIterations)

      {

         cout << "Выход по числу итераций" << endl << endl;

      }

      else

      {

         cout << "Выход по относительной невязке" << endl << endl;

      }

   }

   resultDif = dif;

   return k - 1;

}

void RelaxationTester() {

   double w1, w2, step;

   cout << "Тестирование на решение матрицы с разными коэф. релаксации" << endl << endl;

   cout << "Для тестирования метода Якоби диапазон значений должен быть таким:   0 < w1 <= w2 <= 1" << endl;

   cout << "Для тестирования метода Зейделя диапазон значений должен быть таким: 0 < w1 <= w2 < 2" << endl << endl;

   cout << "Введите диапазон коэф. релаксации:" << endl;

   cout << "  начало диапазона w1: ";

   cin >> w1;

   cout << "  конец диапазона w2:  ";

   cin >> w2;

   cout << "  шаг: ";

   cin >> step;

   if (w1 > w2 || w1 < 0 || w2 < 0 || w1 > 2 || w2 > 2)

   {

      cout << "Неправильно введены данные." << endl;

      return;

   }

   int methodNum = 0;

   cout << endl << "Выберите метод для исследования: " << endl;

   cout << "  1) Якоби" << endl;

   cout << "  2) Зейдель" << endl;

   cout << " > ";

   cin >> methodNum;

   auto& vec1 = vectorX;

   auto& vec2 = methodNum == 1 ? nextVectorX : vectorX;

   string methodName;

   switch (methodNum)

   {

      case 1:

         methodName = "Якоби";

         break;

      case 2:

         methodName = "Зейделя";

         break;

      default:

         cout << "Неверно введено значение." << endl;

         return;

         break;

   }

   cout << endl << "Начало исследования для метода " << methodName << endl << endl;

   size\_t minIter = maxIterations;

   double minIterRelax = INFINITY;

   double minDif = INFINITY;

   vector<double> bestVec;

   auto defaultVec1 = vector<double>(vec1);

   auto defaultVec2 = vector<double>(vec2);

   for (size\_t i = 0; i <= (w2 - w1) / step; i++)

   {

      cout << endl << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl << endl;

      w = w2 - i \* step;

      if (w2 - i \* step < w1) w = w1;

      cout << "Текущее начение коэф.релаксации: " << w << endl;

      size\_t currIter = 0;

      double dif;

      currIter = Iterations(vec1, vec2, dif);

      if (isinf(dif) || isnan(dif))

      {

         cout << "Решение не было получено, метод разошёлся." << endl;

      }

      else

      {

         if (currIter < minIter)

         {

            minIter = currIter;

            minIterRelax = w;

            minDif = dif;

            bestVec = vector<double>(vec1);

         }

         if (currIter == minIter && dif < minDif)

         {

            minIterRelax = w;

            minDif = dif;

            bestVec = vector<double>(vec1);

         }

         cout << format("Число итераций: {0}\nПолученная относительная невязка: {1:<15.3e}\n", currIter, dif);

         cout << "Полученный вектор решения: " << endl;

         PrintArray(vec1.data(), vec1.size(), 15);

      }

      vec1 = vector<double>(defaultVec1);

      vec2 = vector<double>(defaultVec2);

   }

   cout << endl << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl << endl;

   cout << "Исследование завершено. Получены следующие результаты: " << endl;

   cout << "Наименьшее значение итераций метода на этом диапазоне: " << minIter << endl;

   cout << "Значение релаксации при этом:   " << minIterRelax << endl;

   cout << "Значение относительной невязки: " << minDif << endl;

   cout << "Наиболее точный полученный вектор решения: " << endl;

   PrintArray(bestVec.data(), bestVec.size(), 15);

}

void main()

{

   setlocale(LC\_ALL, "ru-RU");

   ReadData();

   int operationNum;

   double dif;

   cout << "Выберите метод решения СЛАУ:" << endl;

   cout << "  1) Якоби" << endl;

   cout << "  2) Зейдель" << endl;

   cout << "  4) Исследование на зависимость скорости сходимости от релаксации" << endl;

   cout << "> ";

   cin >> operationNum;

   switch (operationNum)

   {

      case 1:

         Iterations(vectorX, nextVectorX, dif, true);

         cout << "Полученный вектор решения системы: " << endl;

         PrintArray<double>(vectorX.data(), vectorX.size(), 15, cout);

         PrintArray<double>(vectorX.data(), vectorX.size(), 15, ofstream("./iofiles/allFloatOutput.txt"));

         break;

      case 2:

         Iterations(vectorX, vectorX, dif, true);

         cout << "Полученный вектор решения системы: " << endl;

         PrintArray<double>(vectorX.data(), vectorX.size(), 15, cout);

         PrintArray<double>(vectorX.data(), vectorX.size(), 15, ofstream("./iofiles/allFloatOutput.txt"));

         break;

      case 4:

         RelaxationTester();

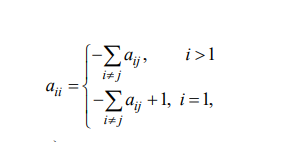
         break;

    }

}

# Исследование

Матрица А строится по правилу:



****

Матрица А:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 16 | -4 | 0 | 0 | 0 | 0 | -4 | -3 | -4 | 0 | 0 | 0 |
| -4 | 10 | -1 | 0 | 0 | 0 | 0 | -2 | -2 | -1 | 0 | 0 |
| 0 | -3 | 7 | -1 | 0 | 0 | 0 | 0 | 0 | -3 | 0 | 0 |
| 0 | 0 | -3 | 14 | -1 | 0 | 0 | 0 | 0 | -3 | -3 | -4 |
| 0 | 0 | 0 | -3 | 9 | -4 | 0 | 0 | 0 | 0 | -2 | 0 |
| 0 | 0 | 0 | 0 | -3 | 8 | -4 | 0 | 0 | 0 | 0 | -1 |
| -4 | 0 | 0 | 0 | 0 | -4 | 9 | -1 | 0 | 0 | 0 | 0 |
| -3 | -2 | 0 | 0 | 0 | 0 | -4 | 9 | 0 | 0 | 0 | 0 |
| 0 | -3 | -1 | 0 | 0 | 0 | 0 | 0 | 7 | -3 | 0 | 0 |
| 0 | -2 | -1 | -1 | 0 | 0 | 0 | 0 | 0 | 5 | -1 | 0 |
| 0 | 0 | -4 | -3 | -4 | 0 | 0 | 0 | 0 | -1 | 14 | -2 |
| 0 | 0 | 0 | -4 | -1 | -1 | 0 | 0 | 0 | 0 | -4 | 10 |

|  |
| --- |
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |
| 6 |
| 7 |
| 8 |
| 9 |
| 10 |
| 11 |
| 12 |

|  |
| --- |
| -80 |
| -31 |
| -19 |
| -69 |
| -13 |
| -7 |
| 27 |
| 37 |
| 24 |
| 28 |
| 76 |
| 49 |

Х

|  |
| --- |
| 112 |
| 71 |
| 61 |
| 181 |
| 103 |
| 103 |
| 99 |
| 107 |
| 102 |
| 72 |
| 232 |
| 191 |

Матрица B (матрица А с обратным знаком внедиагональных элементов):

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 16 | 4 | 0 | 0 | 0 | 0 | 4 | 3 | 4 | 0 | 0 | 0 |
| 4 | 10 | 1 | 0 | 0 | 0 | 0 | 2 | 2 | 1 | 0 | 0 |
| 0 | 3 | 7 | 1 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 |
| 0 | 0 | 3 | 14 | 1 | 0 | 0 | 0 | 0 | 3 | 3 | 4 |
| 0 | 0 | 0 | 3 | 9 | 4 | 0 | 0 | 0 | 0 | 2 | 0 |
| 0 | 0 | 0 | 0 | 3 | 8 | 4 | 0 | 0 | 0 | 0 | 1 |
| 4 | 0 | 0 | 0 | 0 | 4 | 9 | 1 | 0 | 0 | 0 | 0 |
| 3 | 2 | 0 | 0 | 0 | 0 | 4 | 9 | 0 | 0 | 0 | 0 |
| 0 | 3 | 1 | 0 | 0 | 0 | 0 | 0 | 7 | 3 | 0 | 0 |
| 0 | 2 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 5 | 1 | 0 |
| 0 | 0 | 4 | 3 | 4 | 0 | 0 | 0 | 0 | 1 | 14 | 2 |
| 0 | 0 | 0 | 4 | 1 | 1 | 0 | 0 | 0 | 0 | 4 | 10 |

|  |
| --- |
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |
| 6 |
| 7 |
| 8 |
| 9 |
| 10 |
| 11 |
| 12 |

Х

Параметры тестирования для матриц:

Максимальное число итераций: 100000

Начальное приближение: нулевой вектор

Минимальная относительная невязка: 10-13

**Матрица А:**

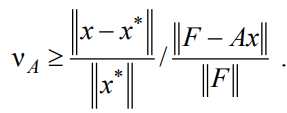
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | w | Метод Якоби | | | | Метод Зейделя | | | |
| x | x\* - x | число итераций | Относит. Невязка | x | x\* - x | число итераций | Относит. Невязка |
| 1 | 0,01 | 0,997739303193777 | 2,3E-03 | 10000 | 4,33E-06 | 0,997824809322430 | 2,2E-03 | 100000 | 4,18E-06 |
| 1,997624577511460 | 2,4E-03 | 1,99771446327847 | 2,3E-03 |
| 2,997546516289700 | 2,5E-03 | 2,99763937580243 | 2,4E-03 |
| 3,997484323381420 | 2,5E-03 | 3,99757954641693 | 2,4E-03 |
| 4,997484854621770 | 2,5E-03 | 4,99758004252199 | 2,4E-03 |
| 5,997529608609110 | 2,5E-03 | 5,99762312529180 | 2,4E-03 |
| 6,997616668493630 | 2,4E-03 | 6,99770693012416 | 2,3E-03 |
| 7,997640980740560 | 2,4E-03 | 7,99773035144624 | 2,3E-03 |
| 8,997555470542520 | 2,4E-03 | 8,99764801665748 | 2,4E-03 |
| 9,997533653054690 | 2,5E-03 | 9,99762706092980 | 2,4E-03 |
| 10,997484056617200 | 2,5E-03 | 10,9975793662801 | 2,4E-03 |
| 11,997469140013600 | 2,5E-03 | 11,9975650582640 | 2,4E-03 |
| 2 | 0,11 | 0,999999999947803 | 5,2E-11 | 29660 | 9,99E-14 | 0,999999999950786 | 4,9E-11 | 28052 | 1,00E-13 |
| 1,999999999945150 | 5,5E-11 | 1,99999999994829 | 5,2E-11 |
| 2,999999999943350 | 5,7E-11 | 2,99999999994660 | 5,3E-11 |
| 3,999999999941910 | 5,8E-11 | 3,99999999994525 | 5,5E-11 |
| 4,999999999941920 | 5,8E-11 | 4,99999999994526 | 5,5E-11 |
| 5,999999999942960 | 5,7E-11 | 5,99999999994624 | 5,4E-11 |
| 6,999999999944970 | 5,5E-11 | 6,99999999994814 | 5,2E-11 |
| 7,999999999945530 | 5,4E-11 | 7,99999999994868 | 5,1E-11 |
| 8,999999999943550 | 5,6E-11 | 8,99999999994680 | 5,3E-11 |
| 9,999999999943050 | 5,7E-11 | 9,99999999994634 | 5,4E-11 |
| 10,999999999941900 | 5,8E-11 | 10,9999999999452 | 5,5E-11 |
| 11,999999999941500 | 5,9E-11 | 11,9999999999449 | 5,5E-11 |
|  |  |  |  |  |  |  |  |  |  |
| 3 | 0,21 | 0,999999999947842 | 5,2E-11 | 15530 | 9,99E-14 | 0,999999999953572 | 4,6E-11 | 13920 | 9,99E-14 |
| 1,999999999945190 | 5,5E-11 | 1,99999999995123 | 4,9E-11 |
| 2,999999999943390 | 5,7E-11 | 2,99999999994964 | 5,0E-11 |
| 3,999999999941950 | 5,8E-11 | 3,99999999994837 | 5,2E-11 |
| 4,999999999941970 | 5,8E-11 | 4,99999999994837 | 5,2E-11 |
| 5,999999999943000 | 5,7E-11 | 5,99999999994930 | 5,1E-11 |
| 6,999999999945010 | 5,5E-11 | 6,99999999995111 | 4,9E-11 |
| 7,999999999945570 | 5,4E-11 | 7,99999999995162 | 4,8E-11 |
| 8,999999999943590 | 5,6E-11 | 8,99999999994983 | 5,0E-11 |
| 9,999999999943090 | 5,7E-11 | 9,99999999994940 | 5,1E-11 |
| 10,999999999941900 | 5,8E-11 | 10,9999999999484 | 5,2E-11 |
| 11,999999999941600 | 5,8E-11 | 11,9999999999481 | 5,2E-11 |
| 4 | 0,31 | 0,999999999947951 | 5,2E-11 | 10517 | 9,98E-14 | 0,999999999956228 | 4,4E-11 | 8903 | 1,00E-13 |
| 1,999999999945310 | 5,5E-11 | 1,99999999995403 | 4,6E-11 |
| 2,999999999943510 | 5,6E-11 | 2,99999999995254 | 4,7E-11 |
| 3,999999999942080 | 5,8E-11 | 3,99999999995134 | 4,9E-11 |
| 4,999999999942090 | 5,8E-11 | 4,99999999995134 | 4,9E-11 |
| 5,999999999943120 | 5,7E-11 | 5,99999999995222 | 4,8E-11 |
| 6,999999999945120 | 5,5E-11 | 6,99999999995394 | 4,6E-11 |
| 7,999999999945680 | 5,4E-11 | 7,99999999995443 | 4,6E-11 |
| 8,999999999943710 | 5,6E-11 | 8,99999999995273 | 4,7E-11 |
| 9,999999999943210 | 5,7E-11 | 9,99999999995233 | 4,8E-11 |
| 10,999999999942000 | 5,8E-11 | 10,9999999999513 | 4,9E-11 |
| 11,999999999941700 | 5,8E-11 | 11,9999999999511 | 4,9E-11 |
|  |  |  |  |  |  |  |  |  |  |
| 5 | 0,41 | 0,999999999947988 | 5,2E-11 | 7949 | 9,99E-14 | 0,999999999959057 | 4,1E-11 | 6333 | 9,99E-14 |
| 1,999999999945340 | 5,5E-11 | 1,99999999995701 | 4,3E-11 |
| 2,999999999943550 | 5,6E-11 | 2,99999999995562 | 4,4E-11 |
| 3,999999999942120 | 5,8E-11 | 3,99999999995451 | 4,5E-11 |
| 4,999999999942130 | 5,8E-11 | 4,99999999995450 | 4,5E-11 |
| 5,999999999943160 | 5,7E-11 | 5,99999999995533 | 4,5E-11 |
| 6,999999999945160 | 5,5E-11 | 6,99999999995695 | 4,3E-11 |
| 7,999999999945720 | 5,4E-11 | 7,99999999995741 | 4,3E-11 |
| 8,999999999943750 | 5,6E-11 | 8,99999999995581 | 4,4E-11 |
| 9,999999999943250 | 5,7E-11 | 9,99999999995545 | 4,5E-11 |
| 10,999999999942100 | 5,8E-11 | 10,9999999999545 | 4,5E-11 |
| 11,999999999941700 | 5,8E-11 | 11,9999999999543 | 4,6E-11 |
| 6 | 0,51 | 0,999999999948010 | 5,2E-11 | 6388 | 9,98E-14 | 0,999999999961762 | 3,8E-11 | 4769 | 9,99E-14 |
| 1,999999999945370 | 5,5E-11 | 1,99999999995987 | 4,0E-11 |
| 2,999999999943570 | 5,6E-11 | 2,99999999995857 | 4,1E-11 |
| 3,999999999942140 | 5,8E-11 | 3,99999999995753 | 4,2E-11 |
| 4,999999999942150 | 5,8E-11 | 4,99999999995752 | 4,2E-11 |
| 5,999999999943180 | 5,7E-11 | 5,99999999995831 | 4,2E-11 |
| 6,999999999945190 | 5,5E-11 | 6,99999999995982 | 4,0E-11 |
| 7,999999999945740 | 5,4E-11 | 7,99999999996027 | 4,0E-11 |
| 8,999999999943780 | 5,6E-11 | 8,99999999995875 | 4,1E-11 |
| 9,999999999943280 | 5,7E-11 | 9,99999999995843 | 4,2E-11 |
| 10,999999999942100 | 5,8E-11 | 10,9999999999576 | 4,2E-11 |
| 11,999999999941700 | 5,8E-11 | 11,9999999999574 | 4,3E-11 |
|  |  |  |  |  |  |  |  |  |  |
| 7 | 0,61 | 0,999999999948086 | 5,2E-11 | 5339 | 9,98E-14 | 0,999999999964557 | 3,5E-11 | 3717 | 9,96E-14 |
| 1,999999999945450 | 5,5E-11 | 1,99999999996281 | 3,7E-11 |
| 2,999999999943650 | 5,6E-11 | 2,99999999996162 | 3,8E-11 |
| 3,999999999942220 | 5,8E-11 | 3,99999999996066 | 3,9E-11 |
| 4,999999999942240 | 5,8E-11 | 4,99999999996065 | 3,9E-11 |
| 5,999999999943270 | 5,7E-11 | 5,99999999996138 | 3,9E-11 |
| 6,999999999945270 | 5,5E-11 | 6,99999999996280 | 3,7E-11 |
| 7,999999999945820 | 5,4E-11 | 7,99999999996322 | 3,7E-11 |
| 8,999999999943860 | 5,6E-11 | 8,99999999996180 | 3,8E-11 |
| 9,999999999943360 | 5,7E-11 | 9,99999999996151 | 3,8E-11 |
| 10,999999999942200 | 5,8E-11 | 10,9999999999607 | 3,9E-11 |
| 11,999999999941800 | 5,8E-11 | 11,9999999999606 | 3,9E-11 |
| 8 | 0,71 | 0,999999999948014 | 5,2E-11 | 4585 | 1,00E-13 | 0,999999999967282 | 3,3E-11 | 2960 | 9,99E-14 |
| 1,999999999945370 | 5,5E-11 | 1,99999999996569 | 3,4E-11 |
| 2,999999999943580 | 5,6E-11 | 2,99999999996459 | 3,5E-11 |
| 3,999999999942150 | 5,8E-11 | 3,99999999996371 | 3,6E-11 |
| 4,999999999942160 | 5,8E-11 | 4,99999999996369 | 3,6E-11 |
| 5,999999999943190 | 5,7E-11 | 5,99999999996438 | 3,6E-11 |
| 6,999999999945190 | 5,5E-11 | 6,99999999996570 | 3,4E-11 |
| 7,999999999945750 | 5,4E-11 | 7,99999999996610 | 3,4E-11 |
| 8,999999999943780 | 5,6E-11 | 8,99999999996477 | 3,5E-11 |
| 9,999999999943280 | 5,7E-11 | 9,99999999996451 | 3,5E-11 |
| 10,999999999942100 | 5,8E-11 | 10,9999999999638 | 3,6E-11 |
| 11,999999999941800 | 5,8E-11 | 11,9999999999637 | 3,6E-11 |
|  |  |  |  |  |  |  |  |  |  |
| 9 | 0,81 | 0,999999999948217 | 5,2E-11 | 4018 | 9,97E-14 | 0,999999999970085 | 3,0E-11 | 2389 | 9,97E-14 |
| 1,999999999945590 | 5,4E-11 | 1,99999999996864 | 3,1E-11 |
| 2,999999999943800 | 5,6E-11 | 2,99999999996765 | 3,2E-11 |
| 3,999999999942370 | 5,8E-11 | 3,99999999996685 | 3,3E-11 |
| 4,999999999942380 | 5,8E-11 | 4,99999999996682 | 3,3E-11 |
| 5,999999999943410 | 5,7E-11 | 5,99999999996746 | 3,3E-11 |
| 6,999999999945400 | 5,5E-11 | 6,99999999996868 | 3,1E-11 |
| 7,999999999945960 | 5,4E-11 | 7,99999999996906 | 3,1E-11 |
| 8,999999999944000 | 5,6E-11 | 8,99999999996782 | 3,2E-11 |
| 9,999999999943500 | 5,7E-11 | 9,99999999996760 | 3,2E-11 |
| 10,999999999942300 | 5,8E-11 | 10,9999999999669 | 3,3E-11 |
| 11,999999999942000 | 5,8E-11 | 11,9999999999668 | 3,3E-11 |
| 10 | 0,91 | 0,999999999948193 | 5,2E-11 | 3575 | 1,00E-13 | 0,999999999972743 | 2,7E-11 | 1942 | 1,00E-13 |
| 1,999999999945560 | 5,4E-11 | 1,99999999997144 | 2,9E-11 |
| 2,999999999943770 | 5,6E-11 | 2,99999999997055 | 2,9E-11 |
| 3,999999999942350 | 5,8E-11 | 3,99999999996982 | 3,0E-11 |
| 4,999999999942360 | 5,8E-11 | 4,99999999996979 | 3,0E-11 |
| 5,999999999943380 | 5,7E-11 | 5,99999999997038 | 3,0E-11 |
| 6,999999999945380 | 5,5E-11 | 6,99999999997151 | 2,8E-11 |
| 7,999999999945940 | 5,4E-11 | 7,99999999997187 | 2,8E-11 |
| 8,999999999943980 | 5,6E-11 | 8,99999999997071 | 2,9E-11 |
| 9,999999999943480 | 5,7E-11 | 9,99999999997053 | 2,9E-11 |
| 10,999999999942300 | 5,8E-11 | 10,9999999999699 | 3,0E-11 |
| 11,999999999942000 | 5,8E-11 | 11,9999999999699 | 3,0E-11 |
|  |  |  |  |  |  |  |  |  |  |
| 11 | 1,01 | 0,999999999948282 | 5,2E-11 | 3220 | 9,97E-14 | 0,999999999975686 | 2,4E-11 | 1583 | 9,94E-14 |
| 1,999999999945650 | 5,4E-11 | 1,99999999997454 | 2,5E-11 |
| 2,999999999943870 | 5,6E-11 | 2,99999999997375 | 2,6E-11 |
| 3,999999999942450 | 5,8E-11 | 3,99999999997311 | 2,7E-11 |
| 4,999999999942460 | 5,8E-11 | 4,99999999997308 | 2,7E-11 |
| 5,999999999943480 | 5,7E-11 | 5,99999999997361 | 2,6E-11 |
| 6,999999999945470 | 5,5E-11 | 6,99999999997464 | 2,5E-11 |
| 7,999999999946030 | 5,4E-11 | 7,99999999997497 | 2,5E-11 |
| 8,999999999944070 | 5,6E-11 | 8,99999999997391 | 2,6E-11 |
| 9,999999999943570 | 5,6E-11 | 9,99999999997376 | 2,6E-11 |
| 10,999999999942400 | 5,8E-11 | 10,9999999999732 | 2,7E-11 |
| 11,999999999942100 | 5,8E-11 | 11,9999999999732 | 2,7E-11 |
| 12 | 1,15 | 0,999999999948537 | 5,1E-11 | 2827 | 9,95E-14 | 0,999999999979775 | 2,0E-11 | 1183 | 9,81E-14 |
| 1,999999999945920 | 5,4E-11 | 1,99999999997885 | 2,1E-11 |
| 2,999999999944140 | 5,6E-11 | 2,99999999997821 | 2,2E-11 |
| 3,999999999942720 | 5,7E-11 | 3,99999999997769 | 2,2E-11 |
| 4,999999999942740 | 5,7E-11 | 4,99999999997765 | 2,2E-11 |
| 5,999999999943760 | 5,6E-11 | 5,99999999997811 | 2,2E-11 |
| 6,999999999945740 | 5,4E-11 | 6,99999999997899 | 2,1E-11 |
| 7,999999999946290 | 5,4E-11 | 7,99999999997928 | 2,1E-11 |
| 8,999999999944350 | 5,6E-11 | 8,99999999997836 | 2,2E-11 |
| 9,999999999943850 | 5,6E-11 | 9,99999999997826 | 2,2E-11 |
| 10,999999999942700 | 5,7E-11 | 10,9999999999778 | 2,2E-11 |
| 11,999999999942300 | 5,8E-11 | 11,9999999999778 | 2,2E-11 |
|  |  |  |  |  |  |  |  |  |  |
| 13 | 1,16 | 0,999999999948292 | 5,2E-11 | 2802 | 9,99E-14 | 0,999999999980057 | 2,0E-11 | 1158 | 9,79E-14 |
| 1,999999999945660 | 5,4E-11 | 1,99999999997915 | 2,1E-11 |
| 2,999999999943880 | 5,6E-11 | 2,99999999997852 | 2,1E-11 |
| 3,999999999942460 | 5,8E-11 | 3,99999999997800 | 2,2E-11 |
| 4,999999999942470 | 5,8E-11 | 4,99999999997796 | 2,2E-11 |
| 5,999999999943490 | 5,7E-11 | 5,99999999997842 | 2,2E-11 |
| 6,999999999945480 | 5,5E-11 | 6,99999999997929 | 2,1E-11 |
| 7,999999999946040 | 5,4E-11 | 7,99999999997958 | 2,0E-11 |
| 8,999999999944080 | 5,6E-11 | 8,99999999997867 | 2,1E-11 |
| 9,999999999943590 | 5,6E-11 | 9,99999999997857 | 2,1E-11 |
| 10,999999999942400 | 5,8E-11 | 10,9999999999782 | 2,2E-11 |
| 11,999999999942100 | 5,8E-11 | 11,9999999999781 | 2,2E-11 |
| 14 | 1,17 | Решение не было получено, метод разошёлся | | | | 0,999999999980156 | 2,0E-11 | 1133 | 9,91E-14 |
| 1,99999999997926 | 2,1E-11 |
| 2,99999999997863 | 2,1E-11 |
| 3,99999999997812 | 2,2E-11 |
| 4,99999999997808 | 2,2E-11 |
| 5,99999999997853 | 2,1E-11 |
| 6,99999999997940 | 2,1E-11 |
| 7,99999999997969 | 2,0E-11 |
| 8,99999999997878 | 2,1E-11 |
| 9,99999999997869 | 2,1E-11 |
| 10,9999999999783 | 2,2E-11 |
| 11,9999999999783 | 2,2E-11 |
|  |  |  | | | |  |  |  |  |
| 15 | 1,21 | - | | | | 0,999999999981538 | 1,8E-11 | 1039 | 9,72E-14 |
| 1,99999999998071 | 1,9E-11 |
| 2,99999999998013 | 2,0E-11 |
| 3,99999999997965 | 2,0E-11 |
| 4,99999999997962 | 2,0E-11 |
| 5,99999999998004 | 2,0E-11 |
| 6,99999999998086 | 1,9E-11 |
| 7,99999999998114 | 1,9E-11 |
| 8,99999999998028 | 2,0E-11 |
| 9,99999999998020 | 2,0E-11 |
| 10,9999999999798 | 2,0E-11 |
| 11,9999999999798 | 2,0E-11 |
| 16 | 1,41 |  | | | | 0,999999999986736 | 1,3E-11 | 641 | 9,99E-14 |
| 1,99999999998619 | 1,4E-11 |
| 2,99999999998580 | 1,4E-11 |
| 3,99999999998547 | 1,5E-11 |
| 4,99999999998543 | 1,5E-11 |
| 5,99999999998576 | 1,4E-11 |
| 6,99999999998640 | 1,4E-11 |
| 7,99999999998664 | 1,3E-11 |
| 8,99999999998594 | 1,4E-11 |
| 9,99999999998593 | 1,4E-11 |
| 10,9999999999857 | 1,4E-11 |
| 11,9999999999857 | 1,4E-11 |
|  |  |  | | | |  |  |  |  |
| 17 | 1,61 |  | | | | 0,999999999992890 | 7,1E-12 | 328 | 9,53E-14 |
| 1,99999999999267 | 7,3E-12 |
| 2,99999999999250 | 7,5E-12 |
| 3,99999999999234 | 7,7E-12 |
| 4,99999999999229 | 7,7E-12 |
| 5,99999999999251 | 7,5E-12 |
| 6,99999999999292 | 7,1E-12 |
| 7,99999999999309 | 6,9E-12 |
| 8,99999999999262 | 7,4E-12 |
| 9,99999999999268 | 7,3E-12 |
| 10,9999999999926 | 7,4E-12 |
| 11,9999999999927 | 7,3E-12 |
| 18 | 1,62 |  | | | | 0,999999999992952 | 7,0E-12 | 313 | 9,86E-14 |
| 1,99999999999274 | 7,3E-12 |
| 2,99999999999258 | 7,4E-12 |
| 3,99999999999243 | 7,6E-12 |
| 4,99999999999237 | 7,6E-12 |
| 5,99999999999259 | 7,4E-12 |
| 6,99999999999300 | 7,0E-12 |
| 7,99999999999317 | 6,8E-12 |
| 8,99999999999270 | 7,3E-12 |
| 9,99999999999276 | 7,2E-12 |
| 10,9999999999927 | 7,3E-12 |
| 11,9999999999928 | 7,2E-12 |
|  |  |  | | | |  |  |  |  |
| 19 | 1,63 |  | | | | 0,999999999993641 | 6,4E-12 | 299 | 9,73E-14 |
| 1,99999999999340 | 6,6E-12 |
| 2,99999999999340 | 6,6E-12 |
| 3,99999999999319 | 6,8E-12 |
| 4,99999999999314 | 6,9E-12 |
| 5,99999999999331 | 6,7E-12 |
| 6,99999999999370 | 6,3E-12 |
| 7,99999999999377 | 6,2E-12 |
| 8,99999999999358 | 6,4E-12 |
| 9,99999999999341 | 6,6E-12 |
| 10,9999999999934 | 6,6E-12 |
| 11,9999999999936 | 6,4E-12 |
| 20 | 1,64 |  | | | | 0,999999999999907 | 9,3E-14 | 322 | 9,17E-14 |
| 2,00000000000022 | -2,2E-13 |
| 2,99999999999940 | 6,0E-13 |
| 3,99999999999984 | 1,6E-13 |
| 4,99999999999978 | 2,2E-13 |
| 5,99999999999999 | 9,8E-15 |
| 6,99999999999990 | 1,0E-13 |
| 8,00000000000043 | -4,3E-13 |
| 8,99999999999903 | 9,7E-13 |
| 10,00000000000030 | -3,0E-13 |
| 10,9999999999996 | 4,0E-13 |
| 11,9999999999993 | 7,0E-13 |
|  |  |  | | | |  |  |  |  |
| 21 | 1,81 |  | | | | Решение не было получено, метод разошёлся | | | |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Матрица В:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | w | Метод Якоби | | | | Метод Зейделя | | | |
| x | x\* - x | число итераций | Относит. Невязка | x | x\* - x | число итераций | Относит. Невязка |
| 1 | 0,01 | 1,000000000002460 | -2,5E-12 | 8998 | 9,99E-14 | 1,000000000002450 | -2,5E-12 | 8974 | 9,99E-14 |
| 1,999999999996340 | 3,7E-12 | 1,99999999999636 | 3,6E-12 |
| 3,000000000001660 | -1,7E-12 | 3,00000000000165 | -1,6E-12 |
| 3,999999999999900 | 1,0E-13 | 3,99999999999991 | 9,0E-14 |
| 4,999999999992890 | 7,1E-12 | 4,99999999999289 | 7,1E-12 |
| 6,000000000010210 | -1,0E-11 | 6,00000000001019 | -1,0E-11 |
| 6,999999999991230 | 8,8E-12 | 6,99999999999126 | 8,7E-12 |
| 8,000000000005470 | -5,5E-12 | 8,00000000000543 | -5,4E-12 |
| 9,000000000001330 | -1,3E-12 | 9,00000000000133 | -1,3E-12 |
| 10,000000000000900 | -9,0E-13 | 10,00000000000090 | -9,0E-13 |
| 11,000000000002400 | -2,4E-12 | 11,0000000000024 | -2,4E-12 |
| 11,999999999998200 | 1,8E-12 | 11,9999999999982 | 1,8E-12 |
| 2 | 0,11 | 1,000000000002390 | -2,4E-12 | 807 | 9,98E-14 | 1,000000000002310 | -2,3E-12 | 783 | 9,86E-14 |
| 1,999999999996450 | 3,6E-12 | 1,99999999999666 | 3,3E-12 |
| 3,000000000001620 | -1,6E-12 | 3,00000000000150 | -1,5E-12 |
| 3,999999999999900 | 1,0E-13 | 3,99999999999996 | 4,0E-14 |
| 4,999999999993080 | 6,9E-12 | 4,99999999999325 | 6,8E-12 |
| 6,000000000009930 | -9,9E-12 | 6,00000000000949 | -9,5E-12 |
| 6,999999999991470 | 8,5E-12 | 6,99999999999198 | 8,0E-12 |
| 8,000000000005310 | -5,3E-12 | 8,00000000000487 | -4,9E-12 |
| 9,000000000001270 | -1,3E-12 | 9,00000000000120 | -1,2E-12 |
| 10,000000000000800 | -8,0E-13 | 10,00000000000070 | -7,0E-13 |
| 11,000000000002400 | -2,4E-12 | 11,0000000000023 | -2,3E-12 |
| 11,999999999998200 | 1,8E-12 | 11,9999999999983 | 1,7E-12 |
|  |  |  |  |  |  |  |  |  |  |
| 3 | 0,21 | 1,000000000002310 | -2,3E-12 | 417 | 9,92E-14 | 1,000000000002200 | -2,2E-12 | 392 | 9,91E-14 |
| 1,999999999996570 | 3,4E-12 | 1,99999999999691 | 3,1E-12 |
| 3,000000000001560 | -1,6E-12 | 3,00000000000138 | -1,4E-12 |
| 3,999999999999900 | 1,0E-13 | 4,00000000000001 | -9,8E-15 |
| 4,999999999993330 | 6,7E-12 | 4,99999999999351 | 6,5E-12 |
| 6,000000000009580 | -9,6E-12 | 6,00000000000894 | -8,9E-12 |
| 6,999999999991770 | 8,2E-12 | 6,99999999999257 | 7,4E-12 |
| 8,000000000005120 | -5,1E-12 | 8,00000000000440 | -4,4E-12 |
| 9,000000000001220 | -1,2E-12 | 9,00000000000112 | -1,1E-12 |
| 10,000000000000800 | -8,0E-13 | 10,00000000000060 | -6,0E-13 |
| 11,000000000002300 | -2,3E-12 | 11,0000000000022 | -2,2E-12 |
| 11,999999999998300 | 1,7E-12 | 11,9999999999984 | 1,6E-12 |
| 4 | 0,31 | 1,000000000002130 | -2,1E-12 | 279 | 9,47E-14 | 1,000000000002030 | -2,0E-12 | 253 | 9,73E-14 |
| 1,999999999996830 | 3,2E-12 | 1,99999999999723 | 2,8E-12 |
| 3,000000000001440 | -1,4E-12 | 3,00000000000123 | -1,2E-12 |
| 3,999999999999910 | 9,0E-14 | 4,00000000000006 | -6,0E-14 |
| 4,999999999993830 | 6,2E-12 | 4,99999999999393 | 6,1E-12 |
| 6,000000000008860 | -8,9E-12 | 6,00000000000817 | -8,2E-12 |
| 6,999999999992380 | 7,6E-12 | 6,99999999999334 | 6,7E-12 |
| 8,000000000004740 | -4,7E-12 | 8,00000000000383 | -3,8E-12 |
| 9,000000000001130 | -1,1E-12 | 9,00000000000101 | -1,0E-12 |
| 10,000000000000700 | -7,0E-13 | 10,00000000000050 | -5,0E-13 |
| 11,000000000002100 | -2,1E-12 | 11,0000000000020 | -2,0E-12 |
| 11,999999999998400 | 1,6E-12 | 11,9999999999985 | 1,5E-12 |
|  |  |  |  |  |  |  |  |  |  |
| 5 | 0,41 | 1,000000000002040 | -2,0E-12 | 208 | 9,36E-14 | 1,000000000001700 | -1,7E-12 | 182 | 8,72E-14 |
| 1,999999999996960 | 3,0E-12 | 1,99999999999775 | 2,2E-12 |
| 3,000000000001380 | -1,4E-12 | 3,00000000000099 | -9,9E-13 |
| 3,999999999999910 | 9,0E-14 | 4,00000000000010 | -1,0E-13 |
| 4,999999999994090 | 5,9E-12 | 4,99999999999485 | 5,1E-12 |
| 6,000000000008480 | -8,5E-12 | 6,00000000000676 | -6,8E-12 |
| 6,999999999992710 | 7,3E-12 | 6,99999999999461 | 5,4E-12 |
| 8,000000000004530 | -4,5E-12 | 8,00000000000301 | -3,0E-12 |
| 9,000000000001090 | -1,1E-12 | 9,00000000000083 | -8,3E-13 |
| 10,000000000000700 | -7,0E-13 | 10,00000000000030 | -3,0E-13 |
| 11,000000000002000 | -2,0E-12 | 11,0000000000017 | -1,7E-12 |
| 11,999999999998500 | 1,5E-12 | 11,9999999999988 | 1,2E-12 |
| 6 | 0,51 | 1,000000000001890 | -1,9E-12 | 165 | 8,97E-14 | 1,000000000001530 | -1,5E-12 | 138 | 8,43E-14 |
| 1,999999999997180 | 2,8E-12 | 1,99999999999805 | 1,9E-12 |
| 3,000000000001280 | -1,3E-12 | 3,00000000000085 | -8,5E-13 |
| 3,999999999999920 | 8,0E-14 | 4,00000000000013 | -1,3E-13 |
| 4,999999999994520 | 5,5E-12 | 4,99999999999532 | 4,7E-12 |
| 6,000000000007860 | -7,9E-12 | 6,00000000000598 | -6,0E-12 |
| 6,999999999993240 | 6,8E-12 | 6,99999999999534 | 4,7E-12 |
| 8,000000000004200 | -4,2E-12 | 8,00000000000251 | -2,5E-12 |
| 9,000000000001010 | -1,0E-12 | 9,00000000000072 | -7,2E-13 |
| 10,000000000000700 | -7,0E-13 | 10,00000000000020 | -2,0E-13 |
| 11,000000000001900 | -1,9E-12 | 11,0000000000015 | -1,5E-12 |
| 11,999999999998600 | 1,4E-12 | 11,9999999999989 | 1,1E-12 |
|  |  |  |  |  |  |  |  |  |  |
| 7 | 0,61 | 1,000000000001770 | -1,8E-12 | 136 | 8,71E-14 | 1,000000000001360 | -1,4E-12 | 108 | 8,25E-14 |
| 1,999999999997360 | 2,6E-12 | 1,99999999999833 | 1,7E-12 |
| 3,000000000001200 | -1,2E-12 | 3,00000000000072 | -7,2E-13 |
| 3,999999999999930 | 7,0E-14 | 4,00000000000016 | -1,6E-13 |
| 4,999999999994860 | 5,1E-12 | 4,99999999999573 | 4,3E-12 |
| 6,000000000007380 | -7,4E-12 | 6,00000000000528 | -5,3E-12 |
| 6,999999999993660 | 6,3E-12 | 6,99999999999601 | 4,0E-12 |
| 8,000000000003940 | -3,9E-12 | 8,00000000000206 | -2,1E-12 |
| 9,000000000000940 | -9,4E-13 | 9,00000000000063 | -6,3E-13 |
| 10,000000000000600 | -6,0E-13 | 10,00000000000010 | -9,9E-14 |
| 11,000000000001700 | -1,7E-12 | 11,0000000000013 | -1,3E-12 |
| 11,999999999998700 | 1,3E-12 | 11,9999999999991 | 9,0E-13 |
| 8 | 0,71 | 1,000000000001720 | -1,7E-12 | 115 | 8,73E-14 | 1,000000000001220 | -1,2E-12 | 86 | 8,19E-14 |
| 1,999999999997440 | 2,6E-12 | 1,99999999999857 | 1,4E-12 |
| 3,000000000001160 | -1,2E-12 | 3,00000000000062 | -6,2E-13 |
| 3,999999999999930 | 7,0E-14 | 4,00000000000019 | -1,9E-13 |
| 4,999999999995030 | 5,0E-12 | 4,99999999999610 | 3,9E-12 |
| 6,000000000007130 | -7,1E-12 | 6,00000000000465 | -4,6E-12 |
| 6,999999999993870 | 6,1E-12 | 6,99999999999660 | 3,4E-12 |
| 8,000000000003810 | -3,8E-12 | 8,00000000000167 | -1,7E-12 |
| 9,000000000000910 | -9,1E-13 | 9,00000000000056 | -5,6E-13 |
| 10,000000000000600 | -6,0E-13 | 10,00000000000000 | 0,0E+00 |
| 11,000000000001700 | -1,7E-12 | 11,0000000000012 | -1,2E-12 |
| 11,999999999998700 | 1,3E-12 | 11,9999999999992 | 8,0E-13 |
|  |  |  |  |  |  |  |  |  |  |
| 9 | 0,81 | 1,000000000001730 | -1,7E-12 | 99 | 9,14E-14 | 1,000000000001060 | -1,1E-12 | 69 | 8,09E-14 |
| 1,999999999997420 | 2,6E-12 | 1,99999999999882 | 1,2E-12 |
| 3,000000000001170 | -1,2E-12 | 3,00000000000051 | -5,1E-13 |
| 3,999999999999930 | 7,0E-14 | 4,00000000000021 | -2,1E-13 |
| 4,999999999994980 | 5,0E-12 | 4,99999999999651 | 3,5E-12 |
| 6,000000000007200 | -7,2E-12 | 6,00000000000398 | -4,0E-12 |
| 6,999999999993810 | 6,2E-12 | 6,99999999999721 | 2,8E-12 |
| 8,000000000003850 | -3,8E-12 | 8,00000000000129 | -1,3E-12 |
| 9,000000000000920 | -9,2E-13 | 9,00000000000048 | -4,8E-13 |
| 10,000000000000600 | -6,0E-13 | 10,00000000000000 | 0,0E+00 |
| 11,000000000001700 | -1,7E-12 | 11,0000000000010 | -1,0E-12 |
| 11,999999999998700 | 1,3E-12 | 11,9999999999993 | 7,0E-13 |
| 10 | 0,85 | 1,000000000001570 | -1,6E-12 | 94 | 8,45E-14 | 1,000000000001070 | -1,1E-12 | 63 | 8,67E-14 |
| 1,999999999997650 | 2,3E-12 | 1,99999999999884 | 1,2E-12 |
| 3,000000000001060 | -1,1E-12 | 3,00000000000050 | -5,0E-13 |
| 3,999999999999930 | 7,0E-14 | 4,00000000000024 | -2,4E-13 |
| 4,999999999995430 | 4,6E-12 | 4,99999999999643 | 3,6E-12 |
| 6,000000000006540 | -6,5E-12 | 6,00000000000398 | -4,0E-12 |
| 6,999999999994370 | 5,6E-12 | 6,99999999999726 | 2,7E-12 |
| 8,000000000003490 | -3,5E-12 | 8,00000000000123 | -1,2E-12 |
| 9,000000000000830 | -8,3E-13 | 9,00000000000048 | -4,8E-13 |
| 10,000000000000500 | -5,0E-13 | 9,99999999999997 | 3,0E-14 |
| 11,000000000001500 | -1,5E-12 | 11,0000000000010 | -1,0E-12 |
| 11,999999999998800 | 1,2E-12 | 11,9999999999993 | 7,0E-13 |
|  |  |  |  |  |  |  |  |  |  |
| 11 | 0,86 | 1,000000000001580 | -1,6E-12 | 93 | 8,13E-14 | 1,000000000000880 | -8,8E-13 | 62 | 7,27E-14 |
| 1,999999999997950 | 2,0E-12 | 1,99999999999905 | 9,5E-13 |
| 3,000000000001120 | -1,1E-12 | 3,00000000000041 | -4,1E-13 |
| 4,000000000000070 | -7,0E-14 | 4,00000000000020 | -2,0E-13 |
| 4,999999999995900 | 4,1E-12 | 4,99999999999704 | 3,0E-12 |
| 6,000000000006210 | -6,2E-12 | 6,00000000000328 | -3,3E-12 |
| 6,999999999994910 | 5,1E-12 | 6,99999999999776 | 2,2E-12 |
| 8,000000000003370 | -3,4E-12 | 8,00000000000100 | -1,0E-12 |
| 9,000000000000910 | -9,1E-13 | 9,00000000000040 | -4,0E-13 |
| 10,000000000000600 | -6,0E-13 | 9,99999999999997 | 3,0E-14 |
| 11,000000000001600 | -1,6E-12 | 11,0000000000008 | -8,0E-13 |
| 11,999999999999000 | 1,0E-12 | 11,9999999999994 | 6,0E-13 |
| 12 | 0,87 | 0,999999999999910 | 9,0E-14 | 98 | 7,95E-14 | 1,000000000001160 | -1,2E-12 | 60 | 9,74E-14 |
| 1,999999999999290 | 7,1E-13 | 1,99999999999876 | 1,2E-12 |
| 2,999999999999800 | 2,0E-13 | 3,00000000000054 | -5,4E-13 |
| 3,999999999999620 | 3,8E-13 | 4,00000000000028 | -2,8E-13 |
| 4,999999999998930 | 1,1E-12 | 4,99999999999608 | 3,9E-12 |
| 6,000000000000640 | -6,4E-13 | 6,00000000000431 | -4,3E-12 |
| 6,999999999998780 | 1,2E-12 | 6,99999999999707 | 2,9E-12 |
| 8,000000000000180 | -1,8E-13 | 8,00000000000130 | -1,3E-12 |
| 8,999999999999760 | 2,4E-13 | 9,00000000000053 | -5,3E-13 |
| 9,999999999999720 | 2,8E-13 | 9,99999999999995 | 5,0E-14 |
| 10,999999999999800 | 2,0E-13 | 11,0000000000011 | -1,1E-12 |
| 11,999999999999400 | 6,0E-13 | 11,9999999999993 | 7,0E-13 |
|  |  |  |  |  |  |  |  |  |  |
| 13 | 0,91 | 0,999999999999607 | 3,9E-13 | 146 | 8,24E-14 | 1,000000000000970 | -9,7E-13 | 55 | 8,77E-14 |
| 1,999999999999580 | 4,2E-13 | 1,99999999999899 | 1,0E-12 |
| 2,999999999999570 | 4,3E-13 | 3,00000000000044 | -4,4E-13 |
| 3,999999999999560 | 4,4E-13 | 4,00000000000025 | -2,5E-13 |
| 4,999999999999560 | 4,4E-13 | 4,99999999999666 | 3,3E-12 |
| 5,999999999999560 | 4,4E-13 | 6,00000000000358 | -3,6E-12 |
| 6,999999999999580 | 4,2E-13 | 6,99999999999762 | 2,4E-12 |
| 7,999999999999580 | 4,2E-13 | 8,00000000000102 | -1,0E-12 |
| 8,999999999999570 | 4,3E-13 | 9,00000000000044 | -4,4E-13 |
| 9,999999999999560 | 4,4E-13 | 9,99999999999993 | 6,9E-14 |
| 10,999999999999500 | 73+J76:S111+M122 | 11,0000000000009 | -9,0E-13 |
| 11,999999999999500 | 5,0E-13 | 11,9999999999994 | 6,0E-13 |
| 14 | 1,01 | Решение не было получено, метод разошёлся | | | | 1,000000000000830 | -8,3E-13 | 43 | 9,61E-14 |
| 1,99999999999918 | 8,2E-13 |
| 3,00000000000037 | -3,7E-13 |
| 4,00000000000030 | -3,0E-13 |
| 4,99999999999692 | 3,1E-12 |
| 6,00000000000305 | -3,1E-12 |
| 6,99999999999813 | 1,9E-12 |
| 8,00000000000072 | -7,2E-13 |
| 9,00000000000040 | -4,0E-13 |
| 9,99999999999986 | 1,4E-13 |
| 11,0000000000008 | -8,0E-13 |
| 11,9999999999995 | 5,0E-13 |
|  |  |  | | | |  |  |  |  |
| 15 | 1,14 |  | | | | 1,000000000000270 | -2,7E-13 | 29 | 3,99E-14 |
| 1,99999999999993 | 7,0E-14 |
| 3,00000000000015 | -1,5E-13 |
| 4,00000000000008 | -8,0E-14 |
| 4,99999999999934 | 6,6E-13 |
| 6,00000000000063 | -6,3E-13 |
| 6,99999999999964 | 3,6E-13 |
| 8,00000000000006 | -6,0E-14 |
| 9,00000000000013 | -1,3E-13 |
| 9,99999999999987 | 1,3E-13 |
| 11,0000000000000 | 0,0E+00 |
| 11,9999999999999 | 9,9E-14 |
| 16 | 1,15 |  | | | | 1,000000000000130 | -1,3E-13 | 29 | 2,49E-14 |
| 2,00000000000012 | -1,2E-13 |
| 3,00000000000024 | -2,4E-13 |
| 4,00000000000002 | -2,0E-14 |
| 5,00000000000004 | -4,0E-14 |
| 6,00000000000022 | -2,2E-13 |
| 6,99999999999990 | 1,0E-13 |
| 7,99999999999998 | 2,0E-14 |
| 9,00000000000016 | -1,6E-13 |
| 9,99999999999993 | 6,9E-14 |
| 10,9999999999998 | 2,0E-13 |
| 11,9999999999999 | 9,9E-14 |
|  |  |  | | | |  |  |  |  |
| 17 | 1,16 |  | | | | 0,999999999999946 | 5,4E-14 | 29 | 6,16E-14 |
| 2,00000000000019 | -1,9E-13 |
| 3,00000000000047 | -4,7E-13 |
| 4,00000000000011 | -1,1E-13 |
| 5,00000000000016 | -1,6E-13 |
| 6,00000000000032 | -3,2E-13 |
| 7,00000000000003 | -3,0E-14 |
| 8,00000000000002 | -2,0E-14 |
| 9,00000000000033 | -3,3E-13 |
| 10,00000000000000 | 0,0E+00 |
| 10,9999999999997 | 3,0E-13 |
| 11,9999999999999 | 9,9E-14 |
| 18 | 1,21 |  | | | | 0,999999999999299 | 7,0E-13 | 32 | 7,34E-14 |
| 1,99999999999961 | 3,9E-13 |
| 2,99999999999975 | 2,5E-13 |
| 4,00000000000033 | -3,3E-13 |
| 4,99999999999941 | 5,9E-13 |
| 6,00000000000011 | -1,1E-13 |
| 7,00000000000033 | -3,3E-13 |
| 8,00000000000021 | -2,1E-13 |
| 8,99999999999998 | 2,0E-14 |
| 10,00000000000040 | -4,0E-13 |
| 11,0000000000004 | -4,0E-13 |
| 11,9999999999997 | 3,0E-13 |
|  |  |  | | | |  |  |  |  |
| 19 | 1,41 |  | | | | 0,999999999999359 | 6,4E-13 | 48 | 6,22E-14 |
| 1,99999999999954 | 4,6E-13 |
| 2,99999999999917 | 8,3E-13 |
| 4,00000000000052 | -5,2E-13 |
| 4,99999999999930 | 7,0E-13 |
| 6,00000000000001 | -9,8E-15 |
| 7,00000000000017 | -1,7E-13 |
| 8,00000000000029 | -2,9E-13 |
| 8,99999999999975 | 2,5E-13 |
| 10,00000000000030 | -3,0E-13 |
| 11,0000000000010 | -1,0E-12 |
| 11,9999999999998 | 2,0E-13 |
| 20 | 1,61 |  | | | | 1,000000000000480 | -4,8E-13 | 95 | 9,16E-14 |
| 2,00000000000004 | -4,0E-14 |
| 3,00000000000035 | -3,5E-13 |
| 3,99999999999975 | 2,5E-13 |
| 5,00000000000040 | -4,0E-13 |
| 6,00000000000000 | 0,0E+00 |
| 6,99999999999926 | 7,4E-13 |
| 7,99999999999974 | 2,6E-13 |
| 9,00000000000060 | -6,0E-13 |
| 9,99999999999944 | 5,6E-13 |
| 10,9999999999993 | 7,0E-13 |
| 12,0000000000019 | -1,9E-12 |
|  |  |  | | | |  |  |  |  |
| 21 | 1,81 |  | | | | Решение не было получено, метод разошёлся | | | |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Оценка числа обусловленности по решениям:**

Оценка числа обусловленности будет проводиться по следующей формуле:



Результат оценки числа обусловленности для оптимального решения:

* матрицы A по методу Якоби: ≈76,01
* матрицы A по методу Зейделя: ≈9,13
* матрицы B по методу Якоби: ≈ 4,95
* матрицы B по методу Зейделя: ≈ 0,76

# Выводы:

1. В ходе исследования мы заметили, что от значения параметра релаксации зависит то, за сколько итераций метод сойдётся к приблизительно верному ответу. Причём для каждой матрицы и для каждого метода решения есть своё оптимальное значение параметра релаксации, при котором решение будет достигнуто быстрее всего.
2. Для обоих рассмотренных методов, при отдалении в любую сторону от идеального значения параметра релаксации увеличивается число итераций, необходимых для достижения приблизительно верного ответа, причём зависимость эта не линейна.
3. Для обоих методов, есть такие значения параметра релаксации, при котором метод начинает расходиться и, соответственно, решение получить невозможно. Скорее всего это связано с тем, что мы работаем с нессиметричными положительно определёнными матрицами, тогда как эти границы справедливы для симметричных матриц.
4. Для метода Якоби на матрице А, оптимальное значение параметра релаксации оказалось больше единицы, а в методе Зейделя – не доходит до верхней границы.
5. При оценке числа обусловленности матрицы, для оптимального решения разных методов на одной матрице были получены сильно отличающиеся значения, причём для метода Якоби они всегда больше. Связано это с тем, что метод Зейделя получил более точные решения (и следственно меньшую относительную невязку), чем метод Якоби.
6. При оценке числа обусловленности матрицы для оптимального решения метода Зейделя получилось значение, меньшее единицы. Связано это с тем, что сама оценка числа обусловленности - это ограничение числа обусловленности снизу, поэтому она может быть меньше 1.
7. Для матрицы А с большим числом числа обусловленности потребовалось больше шагов, чтобы достичь правильного ответа, нежели для матрицы B. Также для матрицы A для эффективной сходимости методов потребовались значения параметра релаксации большие, чем для эффективной сходимости методов для матрицы B с меньшим числом обусловленности.
8. Метод Гаусса-Зейделя оказался эффективнее на обоих матрицах по сравнению с методом Якоби: даже на одинаковых значениях параметра релаксации метод Зейделя сходился быстрее, чем метод Якоби. Можно полагать, что метод Зейделя оказался эффективнее за счёт того, что при вычислениях каждого нового элемента вектора Х он использует уже вычисленные до этого новые значения вектора Х, более приближенных к верному ответу.