



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение высшего образования  
«Дальневосточный федеральный университет»  
(ДВФУ)

---

**ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ  
ТЕХНОЛОГИЙ**

**Кафедра информатики, математического и компьютерного  
моделирования**

**Курсовой проект  
«Методы Гаусса решения систем линейных алгебраических  
уравнений»**

по дисциплине «Вычислительная математика»

Направление подготовки  
01.03.02 «Прикладная математика и информатика»

Выполнил студент  
гр. Б9120-01.03.02миопд  
Агличиев А.О. \_\_\_\_\_  
(ФИО) (подпись)

Проверил доцент, к.ф.-м.н.  
Колобов А.Г. \_\_\_\_\_  
(ФИО) (подпись)

« 4 » января 2023 г.

г. Владивосток  
2024

# Содержание

<b>1</b>	<b>Введение</b>	<b>3</b>
<b>2</b>	<b>Основная часть</b>	<b>4</b>
2.1	Постановка задачи . . . . .	4
2.2	Описание алгоритма решения задачи . . . . .	4
2.3	Описание тестов, использованных для отладки . . . . .	6
2.4	Вычислительные эксперименты . . . . .	8
2.5	Оценка количества арифметических операций . . . . .	9
<b>3</b>	<b>Заключение</b>	<b>9</b>
<b>4</b>	<b>Список использованных источников</b>	<b>10</b>
<b>5</b>	<b>Приложения (тексты программ)</b>	<b>10</b>
<b>6</b>	<b>Решение теоретических задач</b>	<b>15</b>
6.1	Задача 1 . . . . .	15
6.1.1	Постановка задачи . . . . .	15
6.1.2	Решение . . . . .	16
6.2	Задача 2 . . . . .	16
6.2.1	Постановка задачи . . . . .	16
6.2.2	Решение . . . . .	17

# 1 Введение

Объектом исследования являются численные методы решения задач линейной алгебры, а также программное обеспечение, реализующее эти методы.

Цель работы – ознакомиться с численными методами решения систем линейных алгебраических уравнений, нахождения обратных матриц, решения проблемы собственных значений, решить предложенные типовые задачи, сформулировать выводы по полученным решениям, отметить достоинства и недостатки методов, сравнить удобство использования и эффективность работы каждой использованной программы, приобрести практические навыки и компетенции, а также опыт самостоятельной профессиональной деятельности, а именно:

- создать алгоритм решения поставленной задачи и реализовать его, протестировать программы;
- освоить теорию вычислительного эксперимента; современных компьютерных технологий;
- приобрести навыки представления итогов проделанной работы в виде отчета, оформленного в соответствии с имеющимися требованиями, с привлечением современных средств редактирования и печати.

Работа над курсовым проектом предполагает выполнение следующих задач:

- дальнейшее углубление теоретических знаний обучающихся и их систематизацию;
- получение и развитие прикладных умений и практических навыков по направлению подготовки;
- овладение методикой решения конкретных задач;
- развитие навыков самостоятельной работы;
- развитие навыков обработки полученных результатов, анализа и осмысления их с учетом имеющихся литературных данных;

- приобретение навыков оформления описаний программного продукта;
- повышение общей и профессиональной эрудиции.

Изученный студентом в ходе работы материал должен способствовать повышению его качества знаний, закреплению полученных навыков и уверенности в выборе путей будущего развития своих профессиональных способностей.

## 2 Основная часть

### 2.1 Постановка задачи

Система линейных алгебраических уравнений — это система линейных алгебраических уравнений, которая записывается в виде:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = f_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = f_2 \\ \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = f_n. \end{cases} \quad (1)$$

СЛАУ можно представить в матричной форме:

$$Ax = F, \quad (2)$$

где  $A$  — матрица, образованная коэффициентами при неизвестных,  $x$  — вектор-столбец переменных,  $F$  — столбец свободных членов.

Из линейной алгебры известно, что решение (2) существует и единственно, если определитель матрицы  $A$  отличен от нуля. В данной курсовой работе это решение будем искать при помощи метода Гаусса и его модификаций (выбор ведущего элемента по строке, столбцу, всей матрице).

### 2.2 Описание алгоритма решения задачи

Алгоритм разделяется на два этапа:

## 1. Прямой ход

Путем элементарных преобразований приводят к ступенчатому виду.

Рассмотрим пример:

$$\begin{cases} a_{11} \cdot x_1 + a_{12} \cdot x_2 + \dots + a_{1n} \cdot x_n = b_1 \\ a_{21} \cdot x_1 + a_{22} \cdot x_2 + \dots + a_{2n} \cdot x_n = b_2 \\ \dots \\ a_{n1} \cdot x_1 + a_{n2} \cdot x_2 + \dots + a_{nn} \cdot x_n = b_n \end{cases}$$

Делим первую строку на  $a_{11}$  и прибавляем первую строку к остальным с такими коэффициентами, чтобы их коэффициенты в первом столбце обнулились – очевидно, что при прибавлении к первой строке необходимо умножить на  $-a_{i1}$ .

$$\begin{cases} a_{11} \cdot x_1 + a_{12} \cdot x_2 + \dots + a_{1n} \cdot x_n = b_1 \\ a'_{22} \cdot x_2 + \dots + a'_{2n} \cdot x_n = b'_2 \\ \dots \\ a'_{n2} \cdot x_2 + \dots + a'_{nn} \cdot x_n = b'_n \end{cases}$$

Продельываем те же операции и с другими строками и получим:

$$\begin{cases} a_{11} \cdot x_1 + a_{12} \cdot x_2 + \dots + a_{1n} \cdot x_n = b_1 \\ a'_{22} \cdot x_2 + \dots + a'_{2n} \cdot x_n = b'_2 \\ \dots \\ a^{(n-1)}_{nn} \cdot x_n = b^{(n-1)}_n \end{cases}$$

## 2. Обратных ход

Начиная с последнего уравнения, выражаем переменную на главной диагонали и подставляем вычисленные решения, и так далее, «поднимаясь вверх».

$$x_i = \frac{1}{a_{ii}} \left( b_i - \sum_{j=i+1}^n a_{ij} \cdot x_j \right)$$

Описанный выше алгоритм работает, только если  $a_{ii} \neq 0$ . Чтобы алгоритм работал в таком случае необходимо выбрать ведущий элемент и переставить строки/столбцы так, чтобы  $a_{ii}$  стал ненулевым элементом. В качестве ведущего элемента стоит выбирать наибольший по модулю, чтобы вычислительная погрешность медленнее накапливалась.

## 2.3 Описание тестов, использованных для отладки

Для тестов и отладки использовались следующие СЛАУ:

$$1. \begin{cases} 2x_1 + x_2 - x_3 = 8 \\ -3x_1 - x_2 + 2x_3 = 11 \\ -2x_1 + x_2 - x_3 = -3 \end{cases}$$

Метод	$x_1$	$x_2$	$x_3$
Гаусс	2	3	-1
Гаусс с выбором ведущего элемента в строке	2	3	-1
Гаусс с выбором ведущего элемента в столбце	2	3	-1
Гаусс с выбором ведущего элемента в матрице	2	3	-1

$$2. \begin{cases} 0.183x_1 + 0.081x_2 + 0.521x_3 + 0.498x_4 = 0.263 \\ 0.887x_1 + 5.526x_2 + 0.305x_3 + 0.037x_4 = 0.744 \\ 0.678x_1 + 0.658x_2 + 2.453x_3 + 0.192x_4 = 0.245 \\ 4.957x_1 + 0.398x_2 + 0.232x_3 + 0.567x_4 = 0.343 \end{cases}$$

Метод	$x_1$	$x_2$	$x_3$	$x_4$
Гаусс	0,051614	0,128859	0,047621	0,043766
Гаусс с выбором ведущего элемента в строке	0,051614	0,128859	0,047621	0,043766
Гаусс с выбором ведущего элемента в столбце	0,051614	0,128859	0,047621	0,043766
Гаусс с выбором ведущего элемента в матрице	0,051614	0,128859	0,047621	0,043766

$$3. \begin{cases} 0.183x_1 + 0.081x_2 + 0.521x_3 + 0.498x_4 = 0.263 \\ 0.887x_1 + 5.526x_2 + 0.305x_3 + 0.037x_4 = 0.744 \\ 0.678x_1 + 0.658x_2 + 2.453x_3 + 0.192x_4 = 0.245 \\ 0.678x_1 + 0.658x_2 + 2.453x_3 + 0.192x_4 = 0.245 \end{cases}$$

В третьем тесте не получили ответ, т.к. слау содержит одинаковые строки и имеет бесконечное число решений.

## 2.4 Вычислительные эксперименты

$$1. \begin{cases} 23x_1 + 27x_2 + 23x_3 + 19x_4 + 34x_5 + 20x_6 = 34 \\ 13x_1 + 15x_2 + 30x_3 + 26x_4 + 22x_5 + 11x_6 = 22 \\ 12x_1 + 34x_2 + 28x_3 + 11x_4 + 18x_5 + 33x_6 = 27 \\ 21x_1 + 25x_2 + 23x_3 + 33x_4 + 25x_5 + 21x_6 = 11 \\ 30x_1 + 10x_2 + 13x_3 + 29x_4 + 19x_5 + 12x_6 = 27 \\ 11x_1 + 14x_2 + 17x_3 + 26x_4 + 35x_5 + 20x_6 = 17 \end{cases}$$

—	Гаусс	Гаусс с выбором ведущего элемента в строке	Гаусс с выбором ведущего элемента в столбце	Гаусс с выбором ведущего элемента в матрице
$x_1$	1,541615	1,541615	1,541615	1,541615
$x_2$	-1,339768	-1,339768	-1,339768	-1,339768
$x_3$	1,284753	1,284753	1,284753	1,284753
$x_4$	-1,392177	-1,392177	-1,392177	-1,392177
$x_5$	0,536146	0,536146	0,536146	0,536146
$x_6$	0,719484	0,719484	0,719484	0,719484

$$2. \begin{cases} 49x_1 + 8x_2 + 31x_3 + 17x_4 + 24x_4 + 22x_6 + 4x_7 + 3x_8 + 39x_9 + 36x_{10} = 13 \\ 10x_1 + 36x_2 + 3x_3 + 5x_4 + 42x_4 + 46x_6 + 36x_7 + 48x_8 + 5x_9 + 26x_{10} = 19 \\ 3x_1 + 46x_2 + 10x_3 + 35x_4 + 50x_4 + 8x_6 + 29x_7 + 37x_8 + 26x_9 + 36x_{10} = 44 \\ 20x_1 + 24x_2 + 39x_3 + 30x_4 + 39x_4 + 27x_6 + 26x_7 + 14x_8 + 41x_9 + 9x_{10} = 48 \\ 44x_1 + 7x_2 + 15x_3 + 43x_4 + 19x_4 + 39x_6 + 16x_7 + 33x_8 + 21x_9 + 27x_{10} = 23 \\ 8x_1 + 4x_2 + 24x_3 + 44x_4 + 35x_4 + 20x_6 + 15x_7 + 5x_8 + 41x_9 + 20x_{10} = 45 \\ 27x_1 + 46x_2 + 31x_3 + 19x_4 + 38x_4 + 19x_6 + 41x_7 + 27x_8 + 24x_9 + 33x_{10} = 26 \\ 30x_1 + 24x_2 + 17x_3 + 23x_4 + 48x_4 + 33x_6 + 31x_7 + 13x_8 + 29x_9 + 4x_{10} = 3 \\ 31x_1 + 23x_2 + 41x_3 + 8x_4 + 44x_4 + 50x_6 + 42x_7 + 45x_8 + 6x_9 + 36x_{10} = 41 \\ 29x_1 + 40x_2 + 43x_3 + 36x_4 + 6x_4 + 42x_6 + 28x_7 + 7x_8 + 11x_9 + 27x_{10} = 49 \end{cases}$$



—	Гаусс	Гаусс с выбором ведущего элемента в строке	Гаусс с выбором ведущего элемента в столбце	Гаусс с выбором ведущего элемента в матрице
$x_1$	-0,902286	-0,902286	-0,902286	-0,902286
$x_2$	0,236589	0,236589	0,236589	0,236589
$x_3$	1,261807	1,261807	1,261807	1,261807
$x_4$	0,525887	0,525887	0,525887	0,525887
$x_5$	-0,012908	-0,012908	-0,012908	-0,012908
$x_6$	0,220490	0,220490	0,220490	0,220490
$x_7$	-0,840477	-0,840477	-0,840477	-0,840477
$x_8$	0,664099	0,664099	0,664099	0,664099
$x_9$	0,046808	0,046808	0,046808	0,046808
$x_{10}$	0,062953	0,062953	0,062953	0,062953

## 2.5 Оценка количества арифметических операций

Будем рассматривать квадратную матрицу размерностью  $n$ .

Посчитаем количество операций во время прямого хода. Прибавление  $i$ -ой строки к следующей требует  $n - i$  операций. Тогда количество операций умножения и сложения для преобразования матрицы к ступенчатой равно  $\sum_{i=1}^{n-1} (n - i)^2 = \frac{1}{6}n(n - 1)(2n - 1)$ , а делений:  $\sum_{i=1}^{n-1} (n - i) = \frac{1}{2}n(n - 1)$ .

На обратном ходу нам потребуется  $2 \sum_{i=1}^{n-1} (n - i) = n(n - 1)$  операций умножения и вычитания и  $n$  делений.

Таким образом, в сумме нам нужно  $\frac{n^3}{2} + \frac{4n^2}{3} - \frac{5n}{6}$  операций.

## 3 Заключение

В результате работы над курсовым проектом были реализованы метод Гаусса и его модификации: с выбором главного элемента по строке, столбцу, всей матрице. Методы были протестированы и отлажены на серии тестов, а затем применены для вычислительных экспериментов.

Приобрел практические навыки владения:

- современными численными методами решения задач линейной алгебры;
- основами алгоритмизации для численного решения задач линейной алгебры на языке программирования Python 3;
- инструментальными средствами, поддерживающими разработку программного обеспечения для численного решения задач линейной алгебры;

а также навыками представления итогов проделанной работы в виде отчета, оформленного в соответствии с имеющимися требованиями, с привлечением современных средств редактирования и печати.

## 4 Список использованных источников

1. Бахвалов Н.С. Численные методы / Н.С. Бахвалов, Н.П. Жидков, Г.М. Кобельков. – М.: Наука, 2002 г. – 630 с..
2. Фаддеев Л.К. Вычислительные методы линейной алгебры / Л.К. Фаддеев, В.Н. Фаддеева. – М.: Физматгиз, 1963. – 656 с.

## 5 Приложения (тексты программ)

Метод Гаусса:

```
public class GaussLinearEquationSolver {  
  
    public double[] solve(double[][] a, double[] b) {  
        if (a.length != a[0].length) {  
            throw new IllegalArgumentException();  
        }  
        double[][] f = new double[a.length][2];  
        for (int i = 0; i < a.length; ++i) {  
            f[i] = Arrays.copyOf(a[i], a[i].length);  
        }  
        double[] s = Arrays.copyOf(b, b.length);
```

```

        forwardStep(f, s);
        return backwardStep(f, s);
    }

    public void forwardStep(double[][] a, double[] b) {
        int lastIndexA = a.length - 1;
        for(int i = 0; i < lastIndexA; ++i) {
            if (a[i][i] == 0) {
                throw new ArithmeticException();
            }
            for(int j = i + 1; j < a.length; ++j) {
                double multiplier = a[j][i]/a[i][i];
                double[] subtractLineA = copyAndMultiplyArray(a[i], multiplier);
                b[j] -= b[i]*multiplier;
                subtractLine(a[j], subtractLineA);
            }
        }
        if (a[lastIndexA][lastIndexA] == 0) {
            throw new ArithmeticException();
        }
    }

    public double[] backwardStep(double[][] a, double[] b) {
        double[] ans = Arrays.copyOf(b, b.length);
        int lastIndexA = a.length - 1;
        for(int i = lastIndexA; i >= 0; --i) {
            for(int j = i + 1; j < b.length; ++j) {
                ans[i] -= ans[j]*a[i][j];
            }
            ans[i] /= a[i][i];
        }
        return ans;
    }
}

```

Метод Гаусса с выбором ведущего элемента в столбце:

```

public class GaussLinearEquationSolverColumnPivot extends
    GaussLinearEquationSolver {

    @Override
    public void forwardStep(double[][] a, double[] b) {
        int lastIndexA = a.length - 1;
        for(int i = 0; i < lastIndexA; ++i) {
            swapRows(a, b, getColumnPivot(a, i), i);
            if (a[i][i] == 0) {
                throw new ArithmeticException(String.format();
            }
        }
    }
}

```

```

        for(int j = i + 1; j < a.length; ++j) {
            double multiplier = a[j][i]/a[i][i];
            double[] subtractLineA = copyAndMultiplyArray(a[i], multiplier);
            b[j] -= b[i]*multiplier;
            subtractLine(a[j], subtractLineA);
        }
    }
    if (a[lastIndexA][lastIndexA] == 0) {
        throw new ArithmeticException();
    }
}

public int getColumnPivot(double[][] a, int colNumber) {
    int idxOfMax = colNumber;
    for (int row = colNumber; row < a.length; ++row) {
        idxOfMax = Math.abs(a[row][colNumber]) > Math.abs(a[idxOfMax][colNumber]) ? row : idxOfMax;
    }
    return idxOfMax;
}
}

```

Метод Гаусса с выбором ведущего элемента в строке:

```

public class GaussLinearEquationSolverStrokePivot extends
    GaussLinearEquationSolver {
    @Override
    public double[] solve(double[][] a, double[] b) {
        if (a.length != a[0].length) {
            throw new IllegalArgumentException();
        }
        double[][] f = new double[a.length][a.length];
        for (int i = 0; i < a.length; ++i) {
            f[i] = Arrays.copyOf(a[i], a[i].length);
        }
        double[] s = Arrays.copyOf(b, b.length);
        int[] ansConsequence = new int[a[0].length];
        for (int i = 0; i < ansConsequence.length; ++i) {
            ansConsequence[i] = i;
        }
        forwardStep(f, s, ansConsequence);
        return backwardStep(f, s, ansConsequence);
    }

    public void forwardStep(double[][] a, double[] b, int[] ansConsequence) {
        int lastIndexA = a.length - 1;

        for(int i = 0; i < lastIndexA; ++i) {

```

```

        swapCols(a, getStrokePivot(a, i), i, ansConsequence);
        if (a[i][i] == 0) {
            throw new ArithmeticException(String.format();
        }
        for(int j = i + 1; j < a.length; ++j) {
            double multiplier = a[j][i]/a[i][i];
            double[] subtractLineA = copyAndMultiplyArray(a[i], multiplier);
            b[j] -= b[i]*multiplier;
            subtractLine(a[j], subtractLineA);
        }
        if (a[lastIndexA][lastIndexA] == 0) {
            throw new ArithmeticException();
        }
    }
}

public double[] backwardStep(double[][] a, double[] b, int[]
ansConsequence) {
    double[] ans = super.backwardStep(a, b);
    return buildAnswer(ans, ansConsequence);
}

public int getStrokePivot(double[][] a, int rowNumber) {
    int idxOfMax = rowNumber;
    for (int col = rowNumber; col < a[0].length; ++col) {
        idxOfMax = Math.abs(a[rowNumber][col]) > Math.abs(a[rowNumber][
idxOfMax]) ? col : idxOfMax;
    }
    return idxOfMax;
}

double[] buildAnswer(double[] ans, int[] ansConsequence) {
    double[] res = new double[ans.length];
    for (int i = 0; i < ans.length; ++i) {
        res[ansConsequence[i]] = ans[i];
    }
    return res;
}
}

```

Метод Гаусса с выбором ведущего элемента во всей матрице:

```

public class GaussLinearEquationSolverPivot extends
GaussLinearEquationSolver {

    public static class Point {
        private final int x;
        private final int y;
    }
}

```

```

    public int getX() {
        return x;
    }

    public int getY() {
        return y;
    }

    public Point(int x, int y) {
        this.x = x;
        this.y = y;
    }
}

@Override
public double[] solve(double[][] a, double[] b) {
    if (a.length != a[0].length) {
        throw new IllegalArgumentException();
    }
    double[][] f = new double[a.length][a[0].length];
    for (int i = 0; i < a.length; ++i) {
        f[i] = Arrays.copyOf(a[i], a[i].length);
    }
    double[] s = Arrays.copyOf(b, b.length);
    int[] ansConsequence = new int[a[0].length];
    for (int i = 0; i < ansConsequence.length; ++i) {
        ansConsequence[i] = i;
    }
    forwardStep(f, s, ansConsequence);
    return backwardStep(f, s, ansConsequence);
}

public void forwardStep(double[][] a, double[] b, int[] ansConsequence) {
    int lastIndexA = a.length - 1;
    for (int i = 0; i < lastIndexA; ++i) {
        Point max = findPivot(a, i, i);
        swapRows(a, b, max.getX(), i);
        swapCols(a, max.getY(), i, ansConsequence);
        if (a[i][i] == 0) {
            throw new ArithmeticException();
        }
        for (int j = i + 1; j < a.length; ++j) {
            double multiplier = a[j][i] / a[i][i];
            double[] subtractLineA = copyAndMultiplyArray(a[i], multiplier);
            b[j] -= b[i] * multiplier;
            subtractLine(a[j], subtractLineA);
        }
    }
    if (a[lastIndexA][lastIndexA] == 0) {

```

```

        throw new ArithmeticException();
    }
}

public double[] backwardStep(double[][] a, double[] b, int[]
ansConsequence) {
    double[] ans = super.backwardStep(a, b);
    return buildAnswer(ans, ansConsequence);
}

double[] buildAnswer(double[] ans, int[] ansConsequence) {
    double[] res = new double[ans.length];
    for (int i = 0; i < ans.length; ++i) {
        res[ansConsequence[i]] = ans[i];
    }
    return res;
}

public Point findPivot(double[][] a, int rowNumber, int colNumber) {
    int colOfMax = colNumber;
    int rowOfMax = rowNumber;
    for (int i = rowNumber; i < a.length; ++i) {
        for (int j = colNumber; j < a[0].length; ++j) {
            if (Math.abs(a[i][j]) > Math.abs(a[rowOfMax][colOfMax])) {
                rowOfMax = i;
                colOfMax = j;
            }
        }
    }
    return new Point(rowOfMax, colOfMax);
}
}

```

## 6 Решение теоретических задач

### 6.1 Задача 1

#### 6.1.1 Постановка задачи

Найдите соотношение эквивалентности, связывающее норму  $M(A) = n \times \max_{1 \leq i, j \leq n} |a_{ij}|$  с  $\|A\|_\infty$ . Проверьте экспериментально.

### 6.1.2 Решение

$$\|A\|_{\infty} = \max_i \sum_{j=1}^n |a_{ij}| \geq \max_{i,j} |a_{ij}| = \frac{1}{n} M(A)$$

$$\|A\|_{\infty} = \max_i \sum_{j=1}^n |a_{ij}| \leq n \times \max_{i,j} |a_{ij}| = M(A)$$

$$\frac{1}{n} M(A) \leq \|A\|_{\infty} \leq M(A)$$

Проверим экспериментально:

1.  $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 4 & 4 \\ 2 & 1 & 2 \end{pmatrix}$

$$M(A) = 3 \cdot 4 = 12$$

$$\|A\|_{\infty} = 4 + 4 + 4 = 12$$

$$4 \leq 12 \leq 12$$

2.  $\begin{pmatrix} 1 & 2 & 3 \\ 0 & 0 & 7 \\ 2 & 1 & 2 \end{pmatrix}$

$$M(A) = 3 \cdot 7 = 21$$

$$\|A\|_{\infty} = 7$$

$$7 \leq 7 \leq 21$$

## 6.2 Задача 2

### 6.2.1 Постановка задачи

Докажите теоретически и проверьте экспериментально, что число обусловленности  $\mu(A) = \mu(\alpha A)$ , где  $\alpha$  - число,  $\alpha \neq 0$ .



### 6.2.2 Решение

Формула для вычисления обусловленности:

$$\mu(A) = \|A\| \cdot \|A^{-1}\|$$

Найдем число обусловленности для  $\alpha A$ :

$$\mu(\alpha A) = \|\alpha A\| \cdot \|(\alpha A)^{-1}\| = \|\alpha A\| \cdot \left\| \frac{1}{\alpha} A^{-1} \right\| = \alpha \|A\| \cdot \frac{1}{\alpha} \|A^{-1}\| = \|A\| \cdot \|A^{-1}\|$$

Получили равенство  $\mu(A) = \mu(\alpha A)$ .

Проверим экспериментально:

$$A = \begin{pmatrix} 7 & 7 & 6 \\ 14 & 12 & 9 \\ 12 & 6 & 14 \end{pmatrix} \quad \alpha A = \begin{pmatrix} 14 & 14 & 12 \\ 28 & 24 & 18 \\ 24 & 12 & 28 \end{pmatrix}, \alpha = 2$$

$$A^{-1} = \begin{pmatrix} -0.64044944 & 0.34831461 & 0.0505618 \\ 0.49438202 & -0.14606742 & -0.11797753 \\ 0.33707865 & -0.23595506 & 0.078651695 \end{pmatrix}$$

$$(\alpha A)^{-1} = \begin{pmatrix} -0.32022472 & 0.1741573 & 0.0252809 \\ 0.24719101 & -0.07303371 & -0.05898876 \\ 0.16853933 & -0.11797753 & 0.03932584 \end{pmatrix}$$

$$\mu(A) = \|A\|_{\infty} \cdot \|A^{-1}\|_{\infty} = 35 \cdot 1.039326 = 36.376404$$

$$\mu(\alpha A) = \|\alpha A\|_{\infty} \cdot \|(\alpha A)^{-1}\|_{\infty} = 70 \cdot 0.519663 = 36.376404$$

$$\mu(A) = \mu(\alpha A) = 36.376404$$