

Санкт-Петербургский Политехнический Университет
Петра Великого.
Физико-Механический институт
Высшая школа прикладной математики и
вычислительной физики.

**Отчет по лабораторной работе №1 по дисциплине
"Интервальный анализ"**

Выполнил студент:

Романчук Е.В

группа:

5030102/10201

Проверил:

доцент

Баженов Александр Николаевич

Дата: сентябрь 2024

г. Санкт-Петербург

Содержание

1	Постановка задачи	2
2	Теория.	3
2.1	Применение интервальной арифметики для линейной регрессии в контексте поставленной задачи.	3
3	Реализация	4
3.1	Описание	4
3.2	Ссылка на репозиторий	4
3.3	Описание алгоритма	4
3.4	Алгоритм оптимизации на псевдокоде.	5
3.5	Пояснение применимости метода половинного деления	5
4	Результаты.	6
4.1	Для матрицы радиусов: выр. 3	6
4.2	Для матрицы радиусов: выр. 4	7
4.3	Пояснение к результатам.	8
4.4	Графическая интерпретация	9
4.4.1	Представление для задачи с радиусной матрицей выр. 3 .	9
4.4.2	Представление для задачи с радиусной матрицей выр. 4 .	11
5	Обсуждение	13
5.1	Сложность алгоритма	13
5.2	Актуальность задачи	14
5.3	Анализ поставленной задачи в контексте линейной регрессии . .	14
5.4	МРТ.	14

1 Постановка задачи

Пусть дана вещественная матрица выпр. 1

$$\text{mid}A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \quad (1)$$

и неотрицательное число

$$\delta \in [0, \max\{a_{ij}, i, j = \overline{1, 2}\}] \quad (2)$$

Рассмотрим матрицы радиусов:

$$\text{rad}A_1 = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \quad (3)$$

$$\text{rad}A_2 = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \quad (4)$$

Построим интервальную матрицу следующего вида

$$\mathbf{A} = \begin{pmatrix} [a_{11} - \delta \cdot A_i^{(1,1)}, a_{11} + \delta \cdot A_i^{(1,1)}] & [a_{12} - \delta \cdot A_i^{(1,2)}, a_{12} + \delta \cdot A_i^{(1,2)}] \\ [a_{21} - \delta \cdot A_i^{(2,1)}, a_{21} + \delta \cdot A_i^{(2,1)}] & [a_{22} - \delta \cdot A_i^{(2,2)}, a_{22} + \delta \cdot A_i^{(2,2)}] \end{pmatrix} \quad (5)$$

$i = \overline{1, 2}$ Необходимо найти $\min\{\delta \geq 0 | 0 \in \det A\}$. В целях конкретизации и возможности проверки решения будем использовать следующую матрицу

$$\text{mid}A = \begin{pmatrix} 1.05 & 0.95 \\ 1 & 1 \end{pmatrix} \quad (6)$$

2 Теория.

Укажем основные арифметические операции для интервалов:

$$[a, b] + [c, d] = [a + c, b + d] \quad (7)$$

$$[a, b] - [c, d] = [a - d, b - c] \quad (8)$$

$$[a, b] \cdot [c, d] = [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)] \quad (9)$$

$$\frac{[a, b]}{[c, d]} = \left[\min\left(\frac{a}{c}, \frac{a}{d}, \frac{b}{c}, \frac{b}{d}\right), \max\left(\frac{a}{c}, \frac{a}{d}, \frac{b}{c}, \frac{b}{d}\right) \right] \quad (10)$$

$$\text{mid}[a, b] = \frac{1}{2}(a + b) \quad (11)$$

$$\text{wid}[a, b] = (b - a) \quad (12)$$

$$\text{rad}[a, b] = \frac{1}{2}(b - a) \quad (13)$$

Пусть $\text{mid}A = \{a_{ij}\}_{i,j \in N}$ — точечная вещественная матрица середин, $\text{rad}A = \{r_{ij}\}_{i,j \in N}$ — точечная вещественная матрица радиусов. Операцией midrad назовем следующую функцию:

$$\text{midrad}(\text{mid}A, \text{rad}A) = \{[\text{mid}A_{ij} - \text{rad}A_{ij}], [\text{mid}A_{ij} + \text{rad}A_{ij}]\}_{i,j \in N} \quad (14)$$

Результатом операции является интервальная матрица.

2.1 Применение интервальной арифметики для линейной регрессии в контексте поставленной задачи.

Линейная регрессия — это метод, который часто используется для нахождения зависимости между независимыми переменными (факторами) и зависимой переменной (предсказываемым признаком). В стандартной форме она задаётся уравнением:

$$y = A \cdot x + b \quad (15)$$

где A — матрица коэффициентов модели, x — вектор независимых переменных, b — вектор смещений (или свободный член), а y — предсказываемая величина.

Обычно известны m значений $a_i \in A$ и y_i , где $i \in \overline{0, m}$, которые формируют матрицу A и вектор y . Далее стандартным образом решается задача оптимизации, например методом наименьших квадратов (МНК), что позволяет определить вектор x и свободный член b . Несмотря на наличие оценок

для найденных коэффициентов, важно учитывать возможные погрешности в измерениях, которые могут привести к неразрешимости системы. Собственно, выполнение условия выпр. 19 позволяет выявить такие ситуации.

Очевидно, что задачу выпр. 15 можно переформулировать в более удобной форме, дополнив матрицу A единичным вектором и соответствующим образом изменив вектор x , добавив к нему свободный член b :

$$y = A_{ex} \cdot x_{ex} \quad (16)$$

Кроме того, эта задача эквивалентна следующей:

$$y^T = x_{ex}^T \cdot A_{ex}^T \quad (17)$$

3 Реализация

3.1 Описание

Данная лабораторная работа была выполнена на языке Python 3.10 в среде разработки PyCharm. Дополнительно был реализован класс Interval, описывающий интервальную арифметику для удобства написания кода. Кроме того, использовалась библиотека NumPy

Отчёт подготовлен с помощью языка LaTeX.

3.2 Ссылка на репозиторий

https://github.com/UUyy-Geniy/Interval_analysis - GitHub репозиторий

3.3 Описание алгоритма

1. **Инициализация алгоритма:** положим интервал неопределённости $\delta \in [0, \delta_{max}]$, где

$$\delta_{max} = \max_{a_{i,j} \in midA} |a_{i,j}| \quad (18)$$

Заметим, что при этом на левой границе при $\delta = 0$, предполагается, что $0 \notin \det \mathbf{A}(\delta)$. В случае же если это не так, алгоритм оптимизации сразу же выходит по очевидным причинам. Правая граница выбрана таковой в силу того, что $\mathbf{A}(\delta_{max})$ будет содержать 0-ую точечную матрицу, которая по очевидным причинам удовлетворяет условию:

$$0 \in \det \mathbf{A}(\delta_{max}) \quad (19)$$

Таким образом при инициализации алгоритма условие выпр. 19 на левой границе не выполнено, а на правой же оно выполнено. Это правило является основным при работе алгоритма.

2. Найдя допустимый интервал значений δ , с помощью метода половинного деления будем сужать его до тех пор, пока не достигнем заданной точности $\varepsilon = 10^{-10}$ - это условия конца алгоритма оптимизации.

3.4 Алгоритм оптимизации на псевдокоде.

Algorithm 1 Модифицированный алгоритм оптимизации методом половинного деления

```

1: Вход: Начальные границы интервала  $[\delta_{\text{left init}}, \delta_{\text{right init}}] = [0, \delta_{max}]$ , точность  $\varepsilon = 10^{-10}$ 
2: Выход: Оптимальное значение  $\delta_{\text{opt}}$ 
3:
4: function OPTIMIZE( $\delta_{\text{left init}}, \delta_{\text{right init}}, \varepsilon$ )
5:    $\delta_{\text{left}}, \delta_{\text{right}} \leftarrow \delta_{\text{left init}}, \delta_{\text{right init}}$ 
6:   while  $|\delta_{\text{right}} - \delta_{\text{left}}| > \varepsilon$  do
7:      $\delta_{\text{mid}} \leftarrow \frac{\delta_{\text{left}} + \delta_{\text{right}}}{2}$  ▷ Находим середину интервала
8:     Вычислить  $\mathbf{A}(\delta_{\text{mid}})$  ▷ Генерация матрицы для текущего  $\delta$ 
9:     if  $0 \in \det \mathbf{A}(\delta_{\text{mid}})$  then
10:       $\delta_{\text{right}} \leftarrow \delta_{\text{mid}}$  ▷ Перемещаем правую границу интервала
11:     else
12:       $\delta_{\text{left}} \leftarrow \delta_{\text{mid}}$  ▷ Перемещаем левую границу интервала
13:     end if
14:   end while
15:   return  $\delta_{\text{mid}}$ 
16: end function

```

3.5 Пояснение применимости метода половинного деления

В данной задаче метод половинного деления используется в контексте оптимизации в модифицированной форме, так как целевая функция явно не вычисляется, и алгоритму по-сути не требуется её непрерывности. Вместо этого

основная идея метода заключается в последовательном сужении интервала значений δ , на котором проверяется условие выпр. 19.

4 Результаты.

4.1 Для матрицы радиусов: выпр. 3

При $\delta = 0.525$

$$A_1 = \begin{pmatrix} [0.525, 1.575] & [0.425, 1.475] \\ [0.475, 1.525] & [0.475, 1.525] \end{pmatrix} \quad (20)$$

Итоговый интервал из определителя $[-2.0, 2.2]$

Текущие границы $\delta \in [0, 0.525]$

При $\delta = 0.2625$

$$A_2 = \begin{pmatrix} [0.788, 1.313] & [0.688, 1.21] \\ [0.738, 1.26] & [0.738, 1.26] \end{pmatrix} \quad (21)$$

Итоговый интервал из определителя $[-0.949, 1.15]$

Текущие границы $\delta \in [0, 0.263]$

При $\delta = 0.0164$

$$A_6 = \begin{pmatrix} [1.034, 1.066] & [0.934, 0.966] \\ [0.984, 1.016] & [0.984, 1.016] \end{pmatrix} \quad (22)$$

Итоговый интервал из определителя $[0.0344, 0.166]$ Текущие границы $\delta \in [0.0164, 0.0328]$

$$A_{34} = \begin{pmatrix} [1.025, 1.075] & [0.925, 0.975] \\ [0.975, 1.025] & [0.975, 1.025] \end{pmatrix} \quad (23)$$

Итоговый интервал из определителя $[-1.51 * 10^{-10}, 0.2]$

Текущие границы $\delta = 0.025 \pm 4 * 10^{-10}$

4.2 Для матрицы радиусов: выр. 4

$$\delta = 0.525$$

$$A_1 = \begin{pmatrix} [0.525, 1.575] & [0.425, 1.475] \\ [1, 1] & [1, 1] \end{pmatrix} \quad (24)$$

Итоговый интервал из определителя $[-0.95, 1.15]$

$$\delta \in [0, 1.05]$$

$$\delta = 0.2625$$

$$A_2 = \begin{pmatrix} [0.7875, 1.3125] & [0.6875, 1.2125] \\ [1, 1] & [1, 1] \end{pmatrix} \quad (25)$$

Итоговый интервал из определителя $[-0.425, 0.625]$

$$\delta \in [0, 0.525]$$

$$\delta = 0.0328$$

$$A_5 = \begin{pmatrix} [1.0172, 1.0828] & [0.9172, 0.9828] \\ [1, 1] & [1, 1] \end{pmatrix} \quad (26)$$

Итоговый интервал из определителя $[0.0344, 0.167]$

$$\delta \in [0, 0.0656]$$

$$\delta = 0.0492$$

$$A_6 = \begin{pmatrix} [1.001, 1.099] & [0.901, 0.999] \\ [1, 1] & [1, 1] \end{pmatrix} \quad (27)$$

Итоговый интервал из определителя $[0.0016, 0.1984]$

$$\delta \in [0.0328, 0.0656]$$

$$\delta = 0.0500$$

$$A_{34} = \begin{pmatrix} [1.00, 1.10] & [0.899, 1.00] \\ [1, 1] & [1, 1] \end{pmatrix} \quad (28)$$

Итоговый интервал из определителя $[-2.91 \times 10^{-11}, 0.200]$

Текущие границы $\delta = 0.0500 \pm 3 * 10^{-10}$

4.3 Пояснение к результатам.

Пример 20 и 24: В обоих случаях при $\delta = 0.525$ матрицы A_1 показывают широкий интервал для определителя $[-2.0, 2.2]$ и $[-0.95, 1.15]$ соответственно. Это косвенно говорит о необходимости уменьшения интервала неопределённости для δ . Так как ноль попадает в интервал. Начинаем сужение границ интервала справа.

Пример 21 и 25: С уменьшением δ до 0.2625, интервал определителя сужается до $[-0.949, 1.15]$ и $[-0.425, 0.625]$ соответственно. Хотя интервал стал уже, ноль всё ещё попадает в интервал, указывая на необходимость уменьшения δ . Соответственно изменяем правую границу.

Пример 22 и 26: При дальнейшем уменьшении δ до 0.0164 и 0.0328, интервалы определителей сужаются до $[0.0344, 0.166]$ и $[0.0344, 0.167]$ соответственно. Целевое значение нуля не попало в интервал. Необходимо продолжить сужение интервала неопределённости изменяя левую границу.

Пример 23 и 28: В финальных примерах, при $\delta = 0.025 \pm 4 \times 10^{-10}$ и $\delta = 0.0500 \pm 3 \times 10^{-10}$, итоговые интервалы определителей составляют $[-1.51 \times 10^{-10}, 0.2]$ и $[-2.91 \times 10^{-11}, 0.2]$ соответственно. Эти интервалы демонстрируют, что решение задачи достигнуто, и ноль очень близко находится к границе интервала. Это подтверждает, что решение совпадает с теоретическим результатом, найденным вручную.

Ремарка. Хотя в описанных шагах рассматриваются интервалы полученные из определителя, так или иначе, алгоритм смотрит лишь на ширину интервала неопределённости по δ .

4.4 Графическая интерпретация

Для графической интерпретации будем считать, что столбцы матрицы $midA$ — это точки в двумерном пространстве, представленные в виде $(x, y)^T$. Можно отметить, что суть алгоритма в графическом смысле заключается в сужении секторов радиус-векторов, проведённых к границам окрестностей, окружающих точки. Именно поэтому для иллюстрации будут показаны секторы, относящиеся к точкам, а также их окрестности.

4.4.1 Представление для задачи с радиусной матрицей выр. 3

В данной задаче δ -окрестность представлена в виде квадрата, что, по сути, соответствует окружности в метрике l_1 .

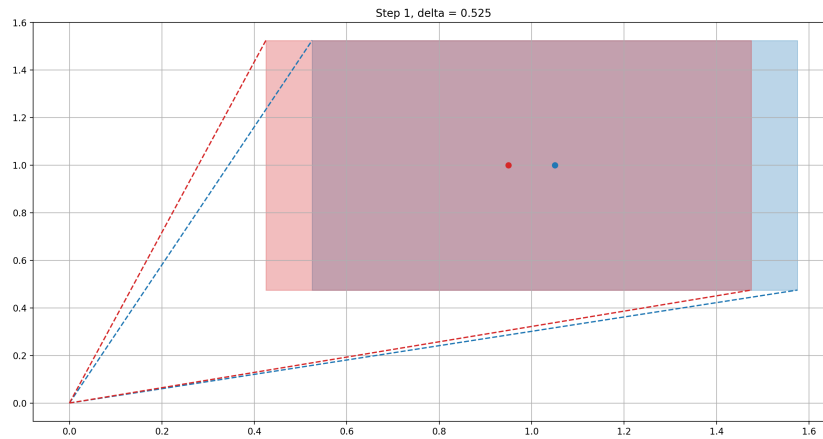


Рис. 1: Графическое представление точек из выр. 20

На первой итерации видно, что выполняется условие выр. 19. Это графически означает наличие пересечения секторов.

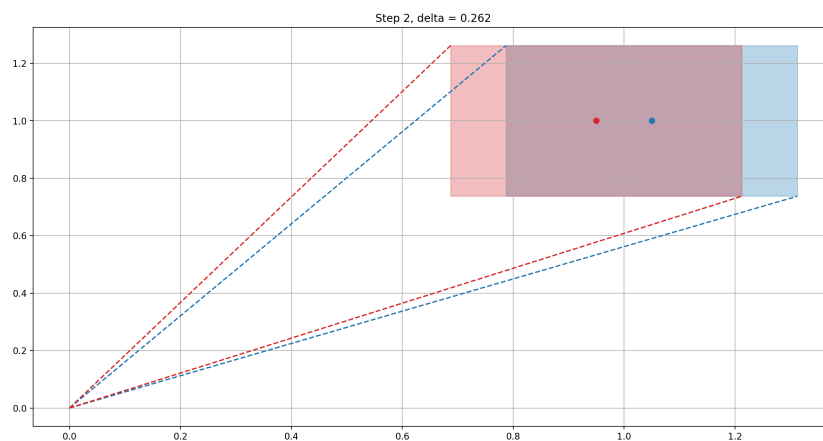


Рис. 2: Графическое представление точек из выр. 21

Здесь видно, что, как и следовало ожидать, происходит сближение границ секторов за счёт уменьшения δ -окрестности.

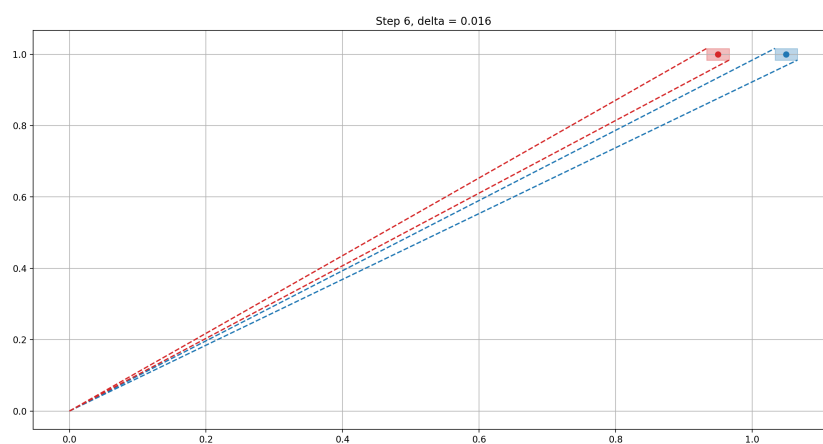


Рис. 3: Графическое представление точек из выр. 22

Как описано в пояснении к выр. 22, условие выр. 19 не выполняется. Это графически означает, что множества секторов не пересекаются, что указывает на отсутствие коллинеарных векторов.

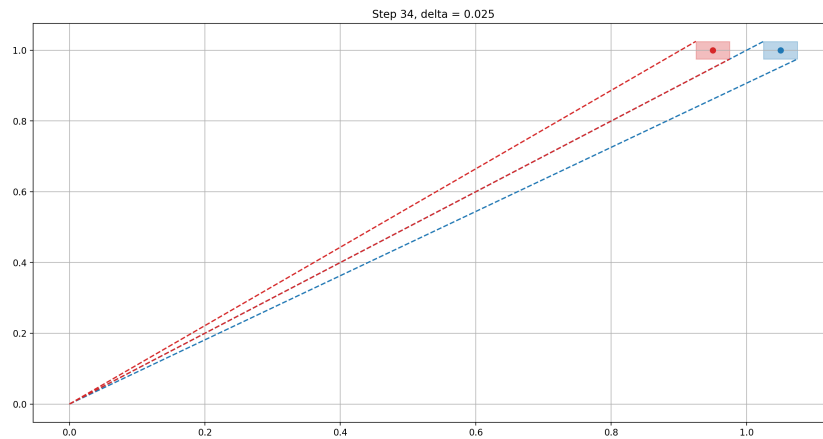


Рис. 4: Графическое представление точек из выр. 23

Здесь показана ситуация для оптимального значения δ . Графически это означает наложение одной из границ секторов.

4.4.2 Представление для задачи с радиусной матрицей выр. 4

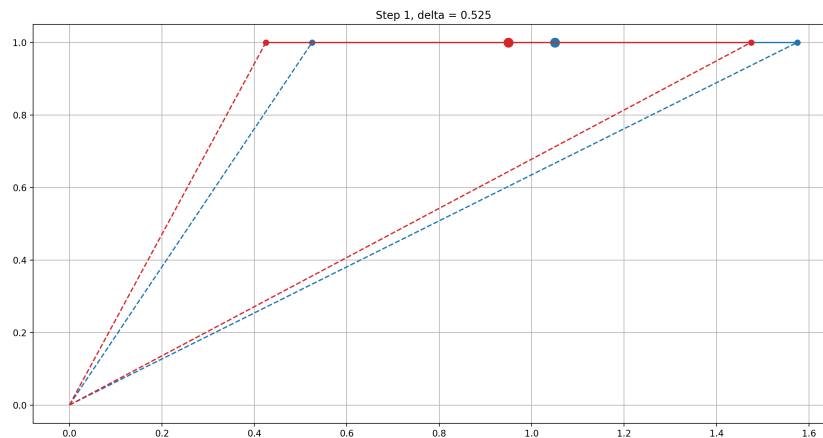


Рис. 5: Графическое представление точек из выр. 24

В случае, когда одна из координат точек фиксирована, как видно из иллюстрации, происходит пересечение секторов.

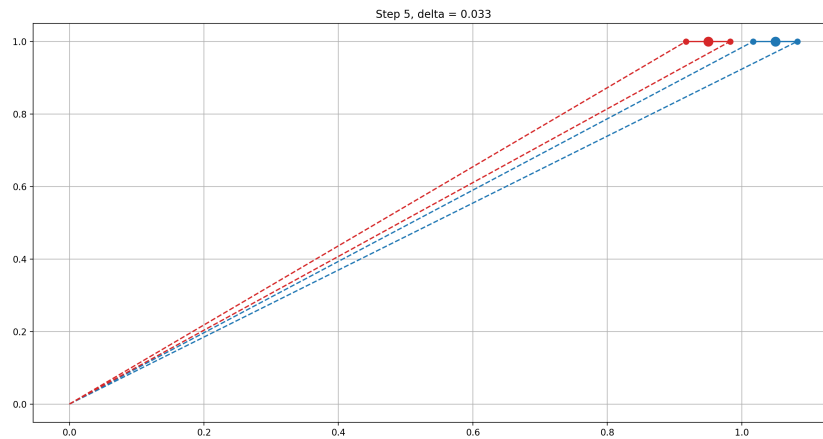


Рис. 6: Графическое представление точек из выр. 26

Здесь видно, что секторы не пересекаются, что означает, что условие выр. 19 не выполняется.

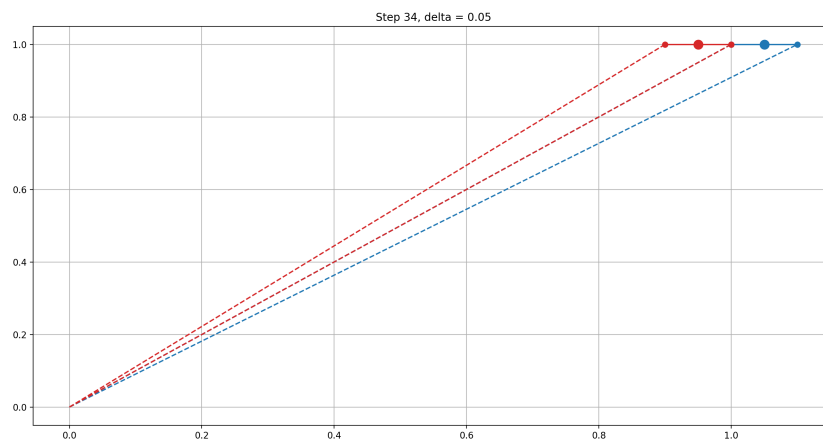


Рис. 7: Графическое представление точек из выр. 28

На этой иллюстрации видно, что секторы, связанные с множествами, накладываются друг на друга по границе, что графически соответствует оптимальному значению δ .

5 Обсуждение

5.1 Сложность алгоритма

Предложенный алгоритм основан на методе половинного деления, сложность которого в стандартной реализации составляет $O(\log(\frac{1}{\varepsilon}))$, где ε — требуемая точность результата. В нашем случае это $\varepsilon = 10^{-10}$, что приводит к достаточно глубокому делению интервала, однако данный шаг обеспечивает необходимую точность.

С другой стороны, на каждом шаге метода половинного деления требуется вычислять определитель матрицы, содержащей интервальные значения. Для этого использована классическая интервальная арифметика, которая значительно усложняет процесс. Если для обычных матриц сложность нахождения определителя матрицы размером $n \times n$ составляет $O(n!)$, то в случае использования интервальной арифметики вычисления становятся еще более ресурсоемкими из-за необходимости работы с диапазонами возможных значений. Это может существенно увеличивать количество операций, необходимых для получения результата.

Таким образом, итоговая сложность алгоритма, реализованного в рамках лабораторной работы, складывается из двух компонентов:

$$O(\log(\frac{1}{\varepsilon}) \cdot n! \cdot k), \quad (29)$$

где $n!$ — сложность вычисления определителя в рамках классической интервальной арифметики, а $\log(\frac{1}{\varepsilon})$ — сложность метода половинного деления.

Необходимо отметить, что хотя данный подход позволяет получить точное решение с использованием метода половинного деления, он может быть не самым оптимальным с точки зрения вычислительных затрат. Возможными путями оптимизации могут быть:

- Применение адаптивных методов деления, где шаг не фиксирован, а динамически подстраивается в зависимости от поведения функции;
- Исследование применения других численных методов, которые могли бы предложить лучшее соотношение точности и скорости, например, методов, основанных на графической интерпретации задачи.

5.2 Актуальность задачи

5.3 Анализ поставленной задачи в контексте линейной регрессии

Рассмотрим задачи в постановке регрессии выпр. 17, где система выпр. 6 используется в качестве A_{ex}^T .

Задачи, решаемые в данном контексте, имеют смысл с точки зрения оценки погрешности, при которой система становится неразрешимой. Это указывает на нахождение области устойчивости исходной системы.

Кроме того, важно отметить, что в большинстве случаев система может содержать больше столбцов, что не является препятствием для работы алгоритма. Например, если матрица имеет размер $2 \times m$, необходимо рассмотреть все возможные подматрицы размером 2×2 . Таких подматриц существует:

$$I = C_m^2, \quad (30)$$

где I — общее количество подматриц. Таким образом, можно заключить, что:

$$\delta_{min} = \max_{i \in \overline{1, I}} \delta_{min, i}, \quad (31)$$

где $\delta_{min, i}$ — минимальное значение δ , найденное для i -й подматрицы с использованием предложенного алгоритма.

Задача, связанная с радиус-матрицей выпр. 3, моделирует случай, когда учитываются погрешности как в векторе b , так и в матрице A . В то время как радиус-матрица выпр. 4 предполагает наличие погрешностей только в матрице A (обозначения из разд. 2.1).

5.4 МРТ.

Кроме того, похожая постановка задачи может встречаться в медицинской диагностике, например, при проведении магнитно-резонансной томографии (МРТ). Здесь интервальные методы позволяют учитывать погрешности при анализе изображений, полученных под разными углами, и находить область устойчивости, в которой изображение остаётся чётким и диагностически полезным. Основная цель — определить допустимые границы отклонений, минимизируя влияние погрешностей и обеспечивая высокую точность визуализации.