Санкт-Петербургский политехнический университет Петра Великого
Институт прикладной математики и механики
Кафедра «Прикладная математика»

ОТЧЁТ ПО ЛАБОРАТОРНЫМ РАБОТАМ ПО ДИСЦИПЛИНЕ «ВЫЧИСЛИТЕЛЬНЫЕ КОМПЛЕКСЫ»

Выполнил
студент группы 3630102/70201

Кузин А.В.

Проверил
к. ф.-м. н., доцент

Баженов Александр Николаевич

Санкт-Петербург
2020

Содержание

Список иллюстраций

1 Задачи

- Решить пример из лекции с треугольной матрицей и неправильными интервалами в правой части.
- Решить более масштабную задачу в 2 вариантах, относящуюся к компьютерной малоракурсной томографии.

2 Решение

2.1 Задача 1

Для решения задачи возьмём следующую треугольную точечную матрицу A и неправильными интервалами в правой части $\mathbf{b}$:

$$A = \begin{pmatrix} 1 & 1 \\ 0 & 0.1 \end{pmatrix}; \mathbf{b} = \begin{pmatrix} [2.1, 1.9] \\ [0.5, 0.8] \end{pmatrix} \tag{1}$$

Для начала разделяем $\mathbf{b}$ на компоненты левой и правой границ:

$$b_{inf} = \begin{pmatrix} 2.1 \\ 0.5 \end{pmatrix}; b_{sup} = \begin{pmatrix} 1.9 \\ 0.8 \end{pmatrix} \tag{2}$$

Теперь приступаем к решению.
Генерируем погружение sti для найденных границ $\mathbf{b}$:

$$\mathrm{sti}(\mathbf{b}) = \begin{pmatrix} -2.1 \\ -0.5 \\ 1.9 \\ 0.8 \end{pmatrix} \tag{3}$$

Далее конструируем знаково-блочную матрицу:

$$A^{\sim} = \left(\begin{array}{c|c} \mathrm{A}^{+} & \mathrm{A}^{-} \\ \hline \mathrm{A}^{-} & \mathrm{A}^{+} \end{array}\right) = \left(\begin{array}{cc|cc} 1 & 1 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0.1 \end{array}\right) \tag{4}$$

Умножаем обратную матрицу к знаково-блочной (4) на (3):

$$(A^{\sim})^{-1} \begin{pmatrix} -2.1 \\ -0.5 \\ 1.9 \\ 0.8 \end{pmatrix} = \left(\begin{array}{cc|cc} 1 & -10 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ \hline 0 & 0 & 1 & -10 \\ 0 & 0 & 0 & 10 \end{array}\right) \begin{pmatrix} -2.1 \\ -0.5 \\ 1.9 \\ 0.8 \end{pmatrix} = \begin{pmatrix} 2.9 \\ -5 \\ -6.1 \\ 8 \end{pmatrix} \tag{5}$$

Тогда формальное решение для системы ИСЛАУ равно:

$$\mathrm{sti}^{-1} \begin{pmatrix} 2.9 \\ -5 \\ -6.1 \\ 8 \end{pmatrix} = \begin{pmatrix} [-2.9, -6.1] \\ [5, 8] \end{pmatrix} \tag{6}$$

Проверка:

$$\begin{pmatrix} 1 & 1 \\ 0 & 0.1 \end{pmatrix} \begin{pmatrix} [-2.9, -6.1] \\ [5, 8] \end{pmatrix} = \begin{pmatrix} [-2.9, -6.1] + [5, 8] \\ [0.5, 0.8] \end{pmatrix} = \begin{pmatrix} [2.1, 1.9] \\ [0.5, 0.8] \end{pmatrix} \tag{7}$$

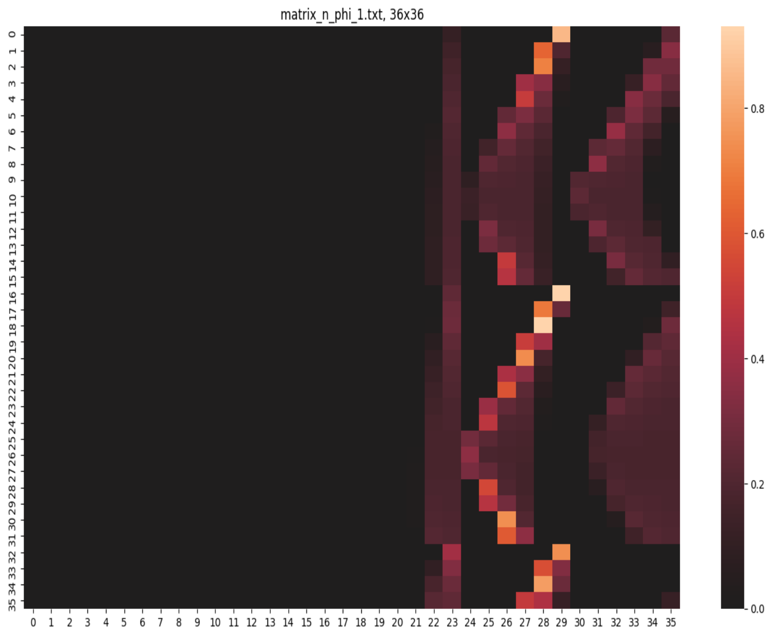
Получен верный результат.

2.2 Задача 2

2.2.1 1 вариант

Из файла *matrix_n_phi_1.txt* загружаем матрицу и приводим её к квадратной, так как в файле прямоугольная.

Квадратная матрица имеет следующую карту величин:



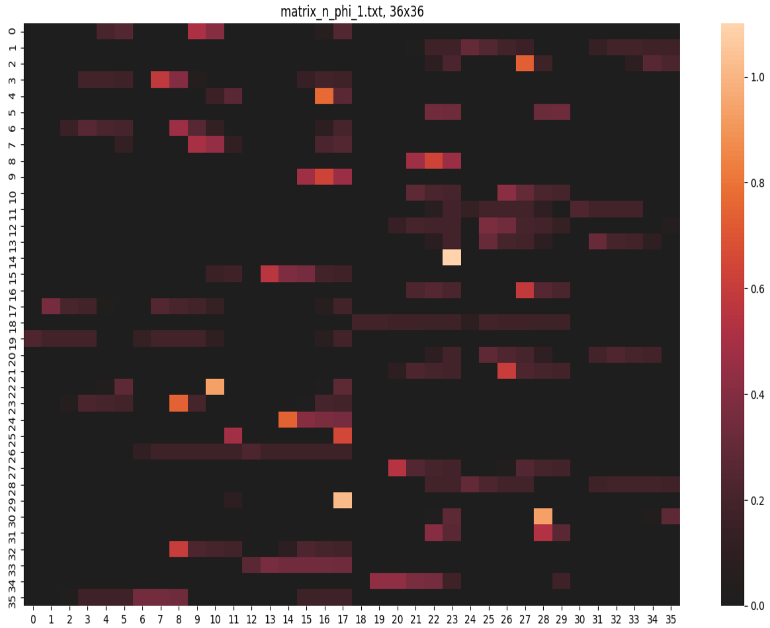
У текущей матрицы есть проблема - определитель равный нулю. Для нахождения решения близкого к истинному лучше брать матрицу с основательно отличительным от нуля определителем. Так как в текущей задаче размерность квадратной матрицы dim(A) = [36, 36] и выбор строк из 256 вариантов, то двоичный перебор с реализацией более-менее "умного"двоичного перебора (без рассмотрения вариантов, которые точно не подходят в виду взятия элементов меньше или больше необходимого, то есть 36) сложен и времязатратен. Наиболее простая идея - случайный перебор вариантов с заранее определённым количеством итераций.

В рамках такого подхода после выполнения алгоритма порядка 10 раз с различным количеством итераций (от $1 * 10^5$ до $5 * 10^5$) следующий полученный результат оказался лучшим:

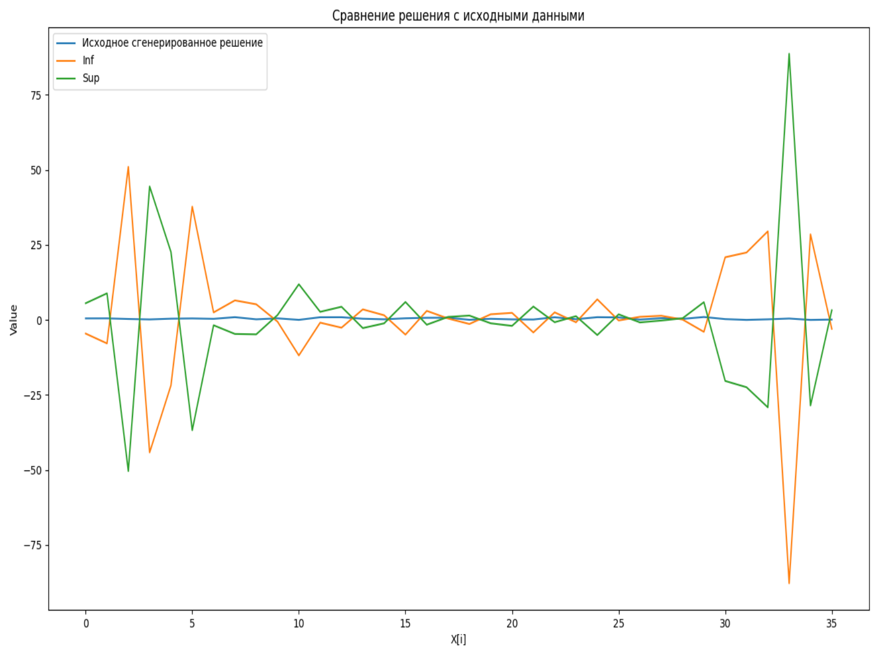
```
Task  36 x 36
Brute force progress:  0.0 %
Current max det:  0
Brute force progress:  10.0 %
Current max det:  4.8381615653028694e-24
Brute force progress:  20.0 %
Current max det:  7.7034297188408605e-22
Brute force progress:  30.0 %
Current max det:  3.6215169820029746e-19
Brute force progress:  40.0 %
Current max det:  3.6215169820029746e-19
Brute force progress:  50.0 %
Current max det:  3.6215169820029746e-19
Brute force progress:  60.0 %
Current max det:  3.6215169820029746e-19
Brute force progress:  70.0 %
Current max det:  3.6215169820029746e-19
Brute force progress:  80.0 %
Current max det:  3.6215169820029746e-19
Brute force progress:  90.0 %
Current max det:  3.6215169820029746e-19

Max det in find matrix : 3.6215169820029746e-19
```

Рис. 1: Случайная выборка 36 строк из 256 за $1 * 10^5$ итераций

То есть определитель конечно выбранной матрицы $\det(\mathrm{A}) \approx 3.62 * 10^{-19}$.
Тогда карта величин станет следующей:



Сгенерируем x и rad$\mathbf{b}$ с помощью равномерного распределения и получим $\mathbf{b}$.
Далее всё аналогично примеру из 1 задачи.
График сравнения решения с исходными данными:

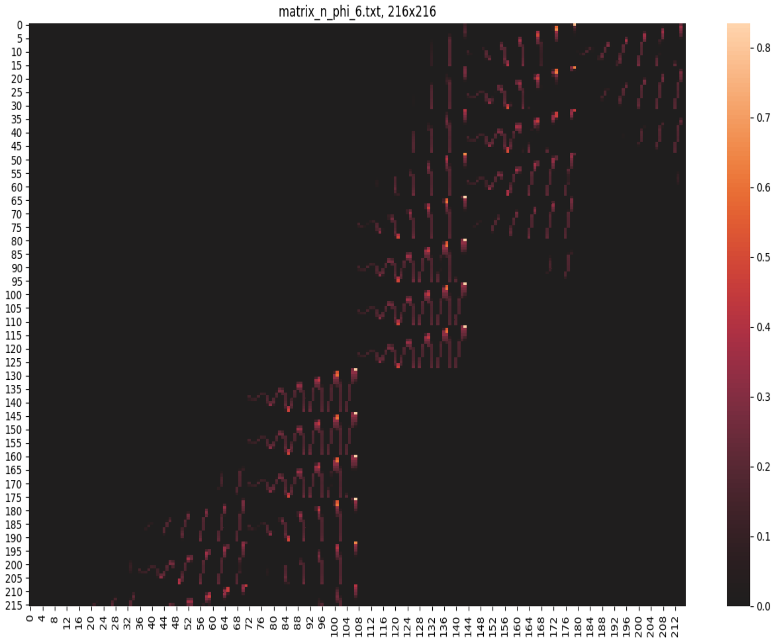


Как видим, **x** содержит все реальные решения внутри себя, а также интервалы для некоторых компонент достигают значений большего порядка, чем через способ диагонального преобладания.

2.2.2 2 вариант

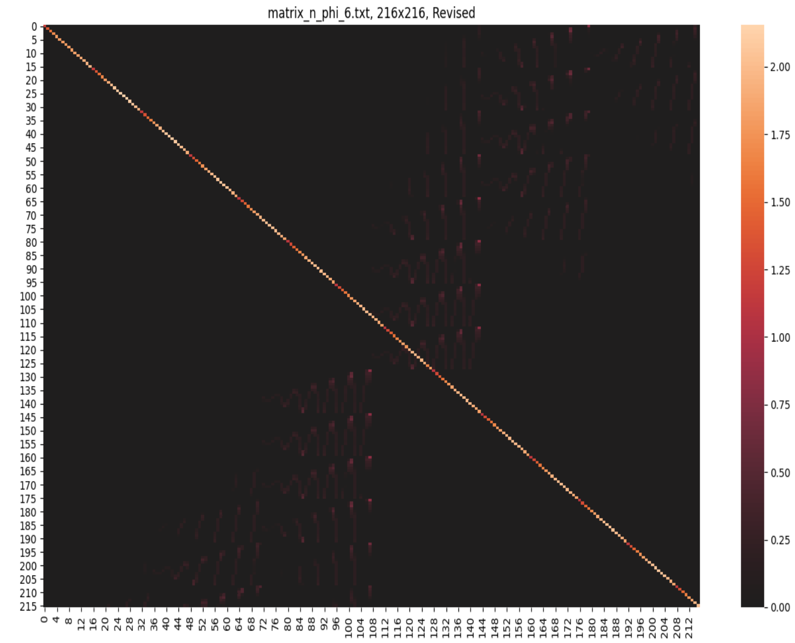
Загружаем данные из файла *matrix_n_phi_6.txt* и всё аналогично предыдущим решениям.

Первоначальная карта величин для матрицы:

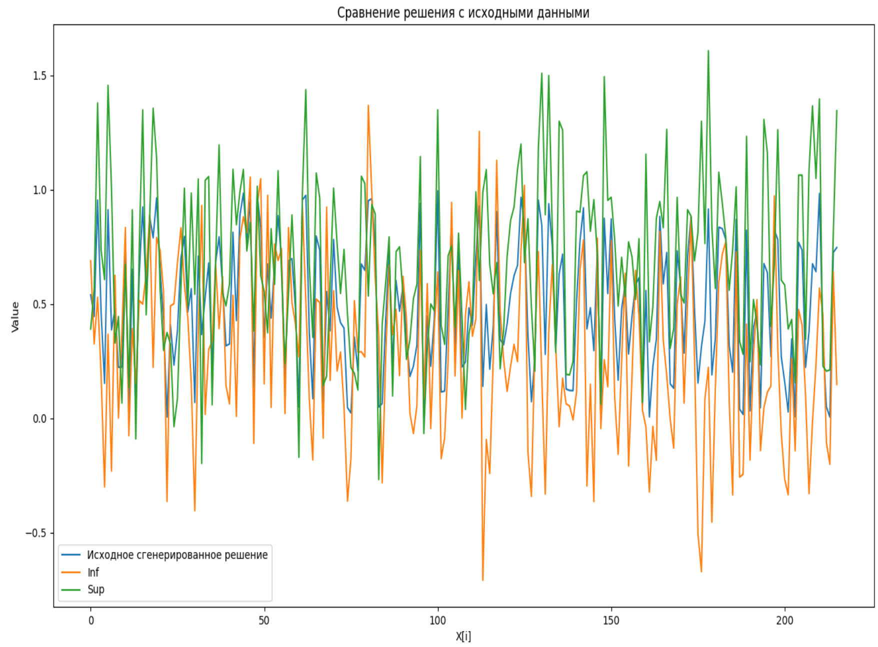


Так как определители равны нулю, добавим к матрице свойство диагонального преобладания.

Карта величин теперь следующая:



После генерации данных и решения получаем следующее сравнение решения с данными:



Из-за большой размерности задачи сложно разглядеть, но видно, что вершины ломаных всё также находятся между границами интервалов.

3 Приложения

Код программы на GitHub, URL: `https://github.com/Kexon5/Comp_complex.git`