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

Крупкина Дарья

Проверил
к. ф.-м. н., доцент

Баженов Александр Николаевич

Санкт-Петербург
2020

Содержание

Список иллюстраций

1 Постановка задачи

1.1 Задача 1

Решить задачу с треугольной матрицей из лекции:

$$\mathbf{C} = \begin{pmatrix} [2,4] & [-2,1] \\ [-2,1] & [2,4] \end{pmatrix} \tag{1}$$

с неправильными(отличными от исходных) интервалами в правой части.

1.2 Задача 2

Решить задачу, относящуюся к малоракурсной томографии. Для данной матрицы размерностью 256x36 необходимо убрать избыточное число уравнений, сгенерировать пробное начальное значение. Затем найти правую часть, объинтервалить ее и затем решить в полной арифметике.

2 Теория

2.1 Субдифференциальный метод Ньютона

Выберем некоторое начальное приближение $x^{(0)} \in R^{2n}$.
Если (k-1)-е приближение $x^{(k-1)} \in R^{2n}, k = 1, 2..$ уже найдено, то вычисляем субградиент $D^{(k-1)}$ отображение Γ в точке $x^{(k-1)}$ и полагаем:

$$x^{(k)} \longleftarrow x^{(k-1)} - \tau (D^{(k-1)})^{-1} \Gamma(x^{(k-1)}), \tag{2}$$

где $\tau \in]0, 1]$ - некоторая константа.
При этом:

$$\Gamma(y) = sti(\mathbf{C} sti^{-1}(y)) - sti(d) \tag{3}$$

Функция погружения дейсвует по правилу:

$$sti(x) : (x_1, ..., x_n) \longrightarrow (-\underline{x_1}, .., -\underline{x_n}, \bar{x_1}, .., \bar{x_n}) \tag{4}$$

3 Реализация

Лабораторная выполнена с помощью средств языка Python, использованы библиотеки numpy, nunmpy.linalg для алгебраических вычислений, а также matplotlib для визуализации.

4 Результаты

4.1 Задача 1

Для начала решим задачу с правой частью, идентичной лекционной, чтобы убедиться в правильности работы методов:

$$\mathbf{d} = \begin{pmatrix} [-2, 2] \\ [-2, 2] \end{pmatrix} \tag{5}$$

Получены следующие результаты для различных точностей:

```
Accuracy: 0.001
[-0.3333333333333337, 0.3333333333333333]
[-0.3333333333333337, 0.3333333333333337]
Quantity of iterations: 3

Accuracy: 1e-05
[-0.3333333333333337, 0.3333333333333333]
[-0.3333333333333337, 0.3333333333333337]
Quantity of iterations: 3

Accuracy: 1e-10
[-0.3333333333333337, 0.3333333333333333]
[-0.3333333333333337, 0.3333333333333337]
Quantity of iterations: 3
```

Рис. 1: Решение исходной задачи

Результат получился достоверный, при этом число итераций осталось малым и не зависит от точности.
Теперь рассмотрим задачу с неправильной правой частью:

$$\mathbf{d} = \begin{pmatrix} [-1.5, 3] \\ [-2, 2] \end{pmatrix} \tag{6}$$

Результаты для такой системы:

```
Accuracy: 0.001
[-0.2083333333333333, 0.6666666666666666]
[-0.16666666666666674, 0.3333333333333334]
Quantity of iterations: 5

Accuracy: 1e-05
[-0.2083333333333333, 0.6666666666666666]
[-0.16666666666666674, 0.3333333333333334]
Quantity of iterations: 5

Accuracy: 1e-10
[-0.2083333333333333, 0.6666666666666666]
[-0.16666666666666674, 0.3333333333333334]
Quantity of iterations: 5
```

Рис. 2: Решение задачи с неправильной правой частью

Проверим влияние точности на число итераций:

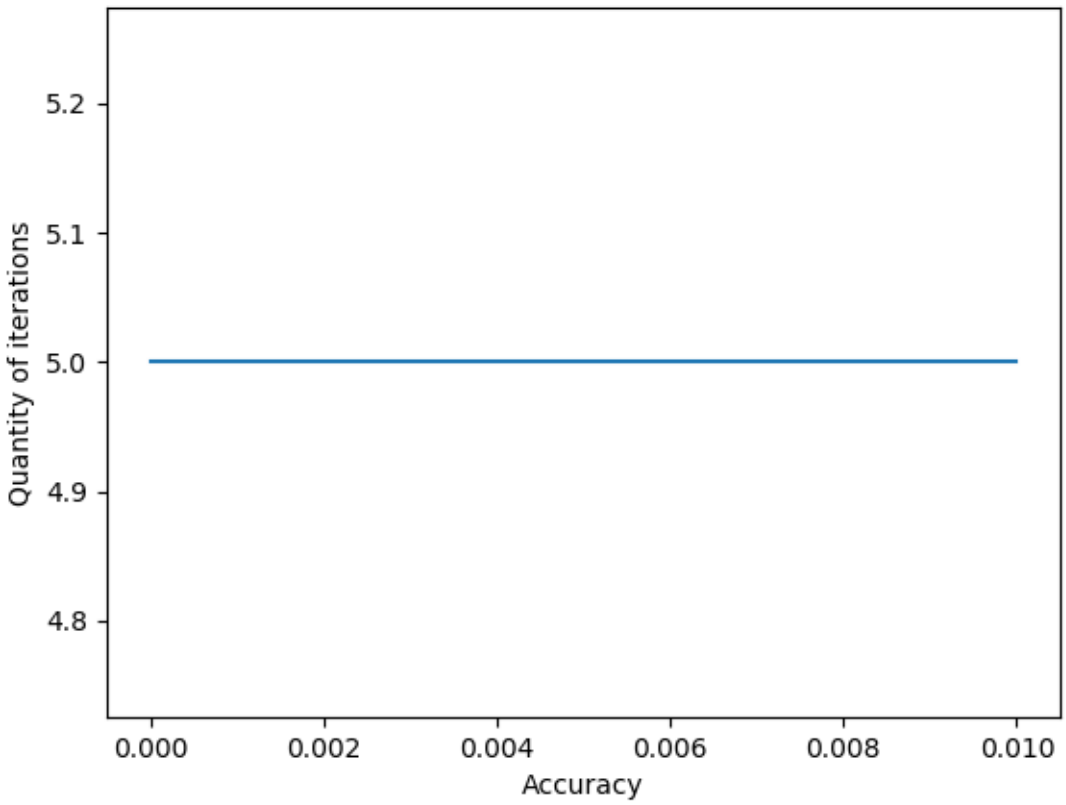

Рис. 3: Зависимость количества итераций от точности значений

Как видно из графика, точность в пределах выбранных значений не оказывает никакого влияния на количество итераций, которое остается одинаковым для точности от 0.01 до 10^{-20}.

4.2 Задача 2

Для задачи 2 имеем следующий алгоритм действий:

1. Берем матрицу 128x18, в которой нет нулевых столбцов.
2. Нам необходимо обрезать матрицу, сделав ее квадратной, поэтому среди 128 строк находим те наиболее заполненные 18, при которых в каждом стоблце будет хотя бы одно достаточно большое значение.
   При этом можно переставлять строки и столбцы местами.
3. Проверяем невырожденность такой матрицы (с компьютерной точки зрения).
4. Находим решения для данной части матрицы.

Для первой матрицы($matrix - n - phi - 1$) в результате первых 2-х шагов получаем матрицу, определитель которой равен
$3.4195006383438334e - 18$, что можно рассмотреть как отличный от 0.
Для такой матрицы можно построить обратную с помощью встроенных методов.
Затем генерируем по закону равномерного распределения вектор значений x, который далее будет являться решением. Получаем $b = A \cdot x$, далее опять с помощью равномерного распределения получаем значения радиусов, с помощью которых объинтерваливаем правую часть.
Получаем для этой матрицы:

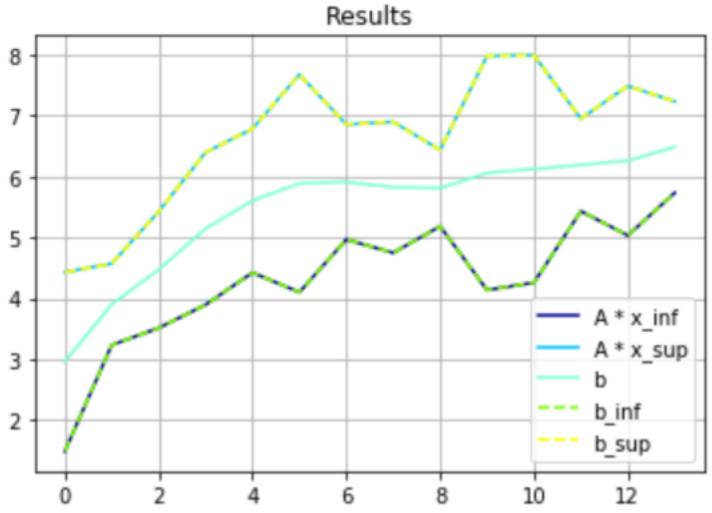

Рис. 4: Область решений для матрицы 1

Количество итераций: 68.
Аналогично поступим с матрицей $matrix - n - phi - 6$, ее определитель равен
$6.381233859251037e - 18$ для нее результат:

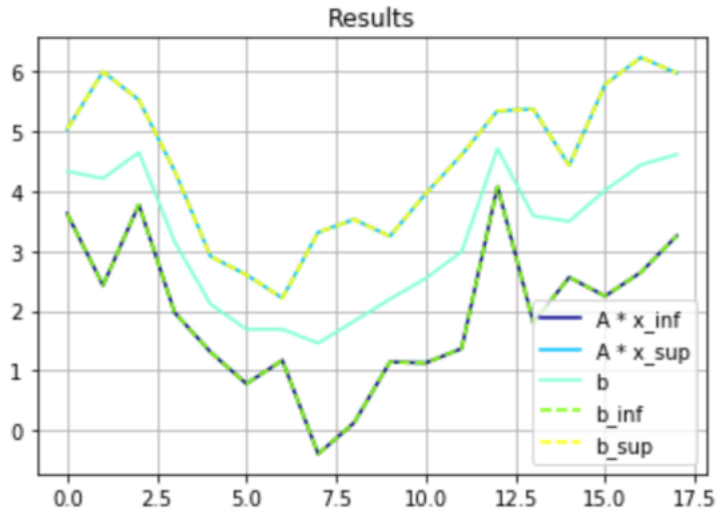

Рис. 5: Область решений для матрицы 2

Количество итераций: 24.
Заметим, что для обеих задач b содержится в между $A \cdot x_{inf}$ и $A \cdot x_{sup}$: $b \in [A \cdot x_{inf}; A \cdot x_{sup}]$, что позволяет дать внешнюю оценку решений.
Для анализа системы(второй матрицы) были взяты модельные правая часть и решение. Правая часть была объинтервалена, как и прежде, а затем получено решение с помощью субдифференциального метода Ньютона.

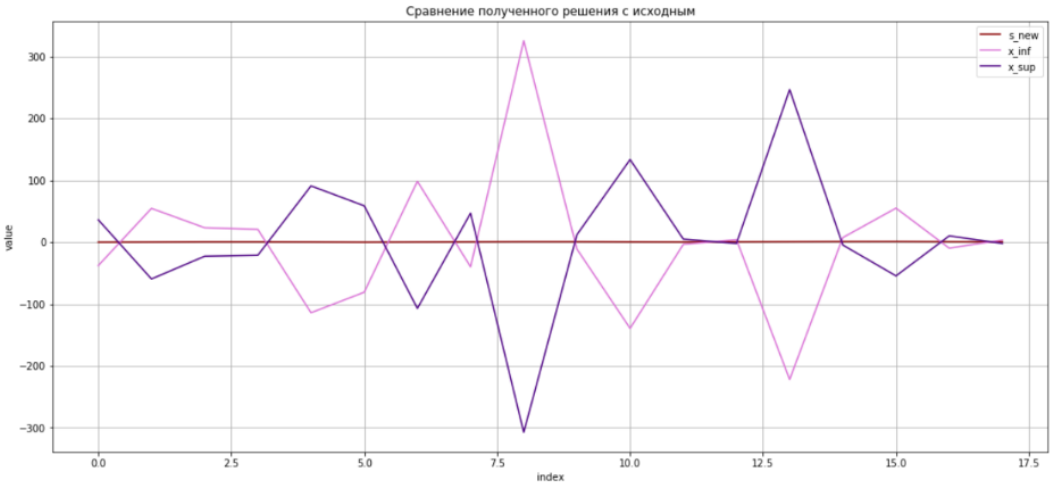

Рис. 6: Сравнение полученного и модельного решений

Как видно из (6), модельное решение находится в области между верхней и нижней границами полученного интервального решения, в некоторых местах совпадая или пересекаясь. Однако здесь достаточно сильные разбросы, которые встречаются в силу того, что в систему внесены произвольные величины.

5 Приложения

Код программы на GitHub, URL: `https://github.com/DariaKrup/Computational_complexes`