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 Постановка задачи

Требуется решить ИСЛАУ с применением аппарата линейного программирования для проведения регуляризации рассматриваемой системы.

2 Конкретизация задачи и теория

При решении данной задачи рассмотрим ИСЛАУ $Ax = b$ с точечной марицей A и интервальной правой частью $\mathbf{b}$ при которых система не имеет решений до проведения регуляризации. В данной работе выбрана несовместная ИСЛАУ:

$$\begin{pmatrix} 8 & -11 & -15 \\ 0.5 & 0 & 0 \\ 10 & 11 & 15 \end{pmatrix} \cdot x = \begin{pmatrix} [5; 7] \\ [-10; -8] \\ [7; 9] \end{pmatrix} \tag{1}$$

В первую очередь с помощью распознающего функционала $\mathrm{Tol}(x)$ проверяется отсутствие решений у данной системы. С помощью программы tolsolvty были найдены максимум функционала распознающего функционала $\max \mathrm{Tol}$ и значение аргумента, в которой он достигался $\arg\max \mathrm{Tol}$:

$$\max \mathrm{Tol} = -7.89473884; \arg\max \mathrm{Tol} = \begin{pmatrix} -0.21052232 \\ 0.03848595 \\ 0.05248085 \end{pmatrix} \tag{2}$$

Поскольку $\max \mathrm{Tol} < 0$, допусковое множество ИСЛАУ пусто и система несовместна. Далее для получения решения проводится l_1-регуляризация, заключающуюся в изменении радиусов компонент вектора $\mathbf{b}$ их поэлементным домножением на вектор масштабирующих множителей ω:

$$\mathbf{b} = \begin{pmatrix} [\mathrm{mid}b_1 - \mathrm{rad}b_1; \mathrm{mid}b_1 + \mathrm{rad}b_1] \\ [\mathrm{mid}b_2 - \mathrm{rad}b_2; \mathrm{mid}b_2 + \mathrm{rad}b_2] \\ [\mathrm{mid}b_3 - \mathrm{rad}b_3; \mathrm{mid}b_3 + \mathrm{rad}b_3] \end{pmatrix} \to \bar{\mathbf{b}} = \begin{pmatrix} [\mathrm{mid}b_1 - \omega_1\mathrm{rad}b_1; \mathrm{mid}b_1 + \omega_1\mathrm{rad}b_1] \\ [\mathrm{mid}b_2 - \omega_2\mathrm{rad}b_2; \mathrm{mid}b_2 + \omega_2\mathrm{rad}b_2] \\ [\mathrm{mid}b_3 - \omega_3\mathrm{rad}b_3; \mathrm{mid}b_3 + \omega_3\mathrm{rad}b_3] \end{pmatrix} \tag{3}$$

При этом масштабирующие множители подбираются так, чтобы регуляризованная ИСЛАУ $A \cdot x = \bar{\mathbf{b}}$ стала разрешима, но сумма этих множителей $\sum_i \omega_i$ была минимально возможной.

Накладывая на масштабирующие множители естественное требование их неотрицательности, и введя вектор $u = \begin{pmatrix} x \\ \omega \end{pmatrix}$, можно записать полученную задачу в виде:

$$\begin{cases} u_{4,5,6} \geq 0 \\ c \cdot u = (0,0,0,1,1,1) \cdot u = (0,0,0,1,1,1) \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} = \sum_i \omega_i = \underset{u}{min} \\ C \cdot u \leq r, \text{где } C = \begin{pmatrix} -A & -\mathrm{diag}(\mathrm{rad}(\mathbf{b})) \\ A & -\mathrm{diag}(\mathrm{rad}(\mathbf{b})) \end{pmatrix}, r = \begin{pmatrix} -\mathrm{mid}(\mathbf{b}) \\ \mathrm{mid}(\mathbf{b}) \end{pmatrix} \end{cases} \tag{4}$$

Полученная задача и решается линейным программированием с применением стандартной функции linprog пакета scipy.optimize. В результате решения определяются одновременно и необходимые масштабирующие множители, и соответствующее им появившееся в результате регуляризации решения ИСЛАУ.

3 Решение

На данный момент имеем $\mathrm{rad}\mathbf{b} = 1$, $\mathrm{mid}\mathbf{b} = \begin{pmatrix} 6 & -9 & 8 \end{pmatrix}$.
После этапа регуляризации получаем следующее:

$$c = (0,0,0,1,1,1); C = \begin{pmatrix} 8 & -11 & -15 & -1 & 0 & 0 \\ 0.5 & 0 & 0 & 0 & -1 & 0 \\ 10 & 11 & 15 & 0 & 0 & -1 \\ -8 & 11 & 15 & -1 & 0 & 0 \\ -0.5 & 0 & 0 & 0 & -1 & 0 \\ -10 & -11 & -15 & 0 & 0 & -1 \end{pmatrix}; r = \begin{pmatrix} 6 & -9 & 8 & -6 & 9 & -8 \end{pmatrix} \tag{5}$$

В результате применения стандартного linprog для решения задачи линейного программирования с использованием значений из (5) без дополнительных ограничений получены следующие результаты:

- Решение регуляризованной ИСЛАУ interior-point методом:

$$x \approx \begin{pmatrix} 0.7778 \\ 0.0071 \\ 0.0096 \end{pmatrix}, \omega \approx \begin{pmatrix} 0 \\ 9.3889 \\ 0 \end{pmatrix} \tag{6}$$

- Решение регуляризованной ИСЛАУ simplex методом:

$$x \approx \begin{pmatrix} 0.7778 \\ 0.0 \\ 0.0148 \end{pmatrix}, \omega \approx \begin{pmatrix} 0 \\ 9.3889 \\ 0 \end{pmatrix} \tag{7}$$

Результат из программы:

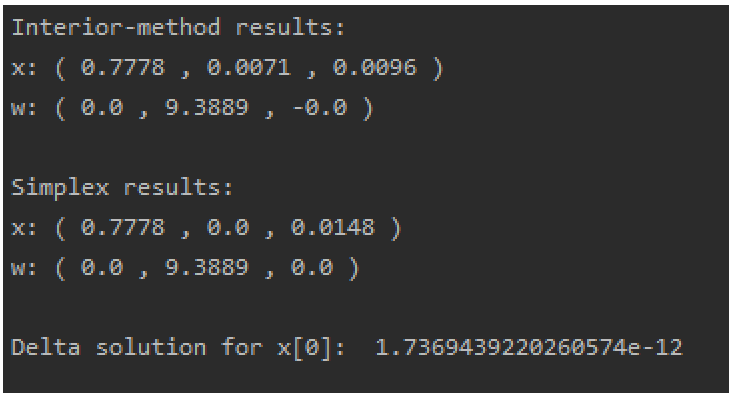

Рис. 1: Решения регуляризованной ИСЛАУ методами linprog без дополнительных ограничений

Из (3) выходит, что:

$$\bar{\mathbf{b}} = \begin{pmatrix} [6 - 0 * 1; 6 + 0 * 1] \\ [-9 - 9.3889 * 1; -9 + 9.3889 * 1] \\ [8 - 0 * 1; 8 + 0 * 1] \end{pmatrix} = \begin{pmatrix} [6; 6] \\ [-18, 3889; 0.3889] \\ [8; 8] \end{pmatrix} \tag{8}$$

Заметно, что масштабирующие коэффициенты в обеих задачах совпали - ненулевые только по 2 компоненте - и равны 9.3889. Это означает, что на факт отсутствия решения исходной ИСЛАУ повлиял интервал правой части по второй компоненте, в то время как другие компоненты без потерь возможности решения можно было сжать. Кроме того, первая компонента вектора оказалась равна в обоих случаях (с точностью до 10^{-12}).

Это может означать, что для третьей компоненты решения может существовать множество решений, удовлетворяющих задаче минимизации.

Рассмотрим изменения нижних границ для 2-й и 3-й границ, чтобы убедиться в расширении интервала для достоверных решений.

При изменении границ для одной из координат в обоих методах компоненты вектора решения симметрично меняются:

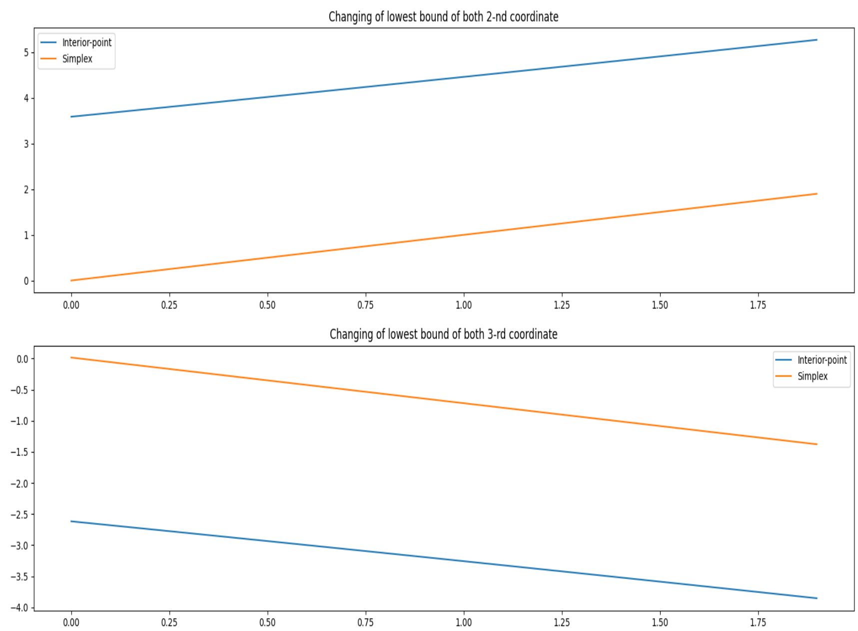

Рис. 2: Изменения нижней границы одной координаты решения в обоих методах

В случае, если вносить изменения в границы 2-х компонент решения, выйдет, что решения будут совпадать:

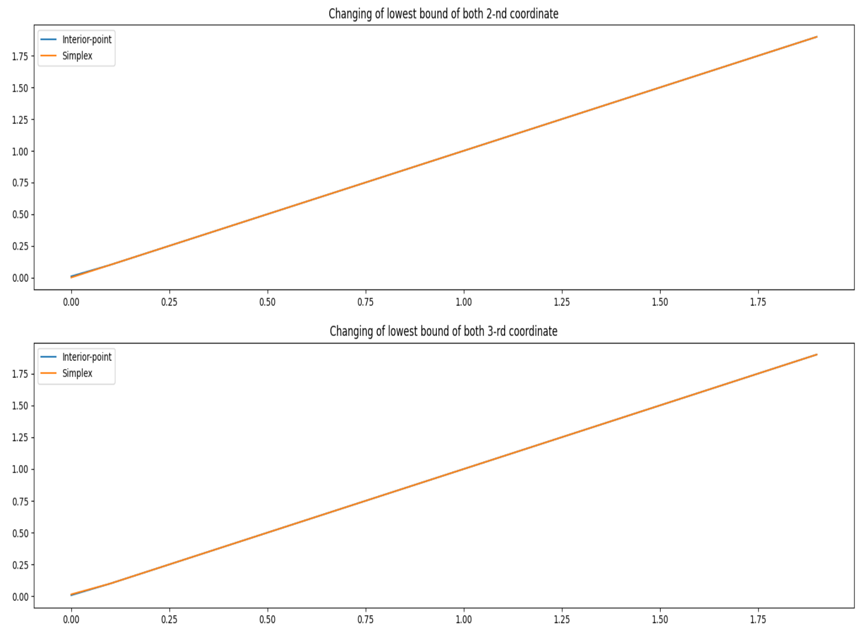

Рис. 3: Изменение нижней границы второй и третьей компонент вектора решений в обоих методах

Выходит, что существует множество решений поставленной задачи линейного программирования, соответствующее фиксированному значению $x_1 = 0.7778$ и целой полосе возможных значений по другим компонентам.

4 Приложения

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