

САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ ПЕТРА ВЕЛИКОГО

ФИЗИКО-МЕХАНИЧЕСКИЙ ИНСТИТУТ

ВЫСШАЯ ШКОЛА ПРИКЛАДНОЙ МАТЕМАТИКИ И ФИЗИКИ

Отчет по курсовой работе по дисциплине  
"Интервальный анализ"

Выполнил:

Студент: Зинякова Екатерина

Группа: 5030102/00201

Принял:

к. ф.-м. н., доцент

Баженов Александр Николаевич

Санкт-Петербург

2023 г.

# Содержание

<b>1</b>	<b>Постановка задачи</b>	<b>2</b>
1.1	.....	2
1.2	.....	2
<b>2</b>	<b>Теория</b>	<b>2</b>
2.1	Введение .....	2
2.2	Особенные матрицы .....	3
2.3	Субдифференциальный метод Ньютона .....	4
2.4	Алгоритм метода .....	4
<b>3</b>	<b>Реализация</b>	<b>4</b>
<b>4</b>	<b>Результаты</b>	<b>4</b>
4.1	.....	4
4.2	Измененная ИСЛАУ .....	5
4.3	.....	6
4.4	Внесение изменений .....	7
<b>5</b>	<b>Выводы</b>	<b>8</b>

# 1 Постановка задачи

В данной курсовой работе рассмотрим субдифференциальный метод Ньютона для решения ИСЛАУ и исследуем поведения метода при внесении изменений в исходную систему.

## 1.1

Исследовать поведение субдифференциального метода Ньютона при решении ИСЛАУ  $\mathbf{x} = \mathbf{C}\mathbf{x} + \mathbf{d}$ , где:

$$\mathbf{C} = \begin{pmatrix} [9, 12] & [1, 1.5] \\ [1.5, 2.25] & [19.5, 22.5] \end{pmatrix} \quad (1)$$

$$\mathbf{d} = \begin{pmatrix} 1.5 \\ 3 \end{pmatrix} \quad (2)$$

## 1.2

Задана ИСЛАУ  $Ax = b$ , где:

$$\mathbf{A} = \begin{pmatrix} [3, 4] & [-5, -2] & [-2, 2] \\ [-3, -1] & [6, 7] & [5, 6] \\ [-1, 0] & [-1, 1] & [-4, 1] \end{pmatrix} \quad (3)$$

$$\mathbf{b} = \begin{pmatrix} [-28, 43] \\ [-60, 69] \\ [-11, 39] \end{pmatrix} \quad (4)$$

Проанализировать результаты и исследовать сходимость метода.

# 2 Теория

## 2.1 Введение

Мы имеем уравнение, заданное в рекуррентном виде,

$$x = \mathbf{C}x + \mathbf{d}, \quad \mathbf{C} = (I \ominus \mathbf{A}^c), \quad \mathbf{d} = \mathbf{b}^c,$$

и мы можем записать равносильное ему в  $\mathbb{R}\mathbb{R}$  :

$$\mathbf{C}x \ominus x + \mathbf{d} = 0.$$

Для расширения наших возможностей при нахождении формального решения мы перейдем из нелинейного пространства  $\mathbb{K}\mathbb{R}^n$  в некое линейное пространство  $U$ , построив биективное отображение, называемое **вложением**:

$$\iota : \mathbb{K}\mathbb{R}^n \rightarrow U,$$

причем для каждого отображения  $\phi : \mathbb{K}^n \rightarrow \mathbb{K}\mathbb{R}^n$  существует лишь единственное индуцированное отображение

$$\iota \circ \phi \circ \iota^{-1} : U \rightarrow U$$

где « $\circ$ » служит для обозначения композиции отображений.

Тесная связь свойств отображений  $\phi$  и  $\iota \circ \phi \circ \iota^{-1}$  позволит нам заменить исходную задачу решения уравнения в  $\mathbb{K}\mathbb{R}$

$$f(x) = 0, \quad f : x \mapsto Cx \ominus x + d$$

на задачу решения индуцированного уравнения в  $U$  :

$$\begin{aligned} \mathcal{F}(y) &= \iota(0), \\ \mathcal{F} &= \iota \circ f \circ \iota^{-1} : U \rightarrow U, \\ \mathcal{F}(y) &= \iota(C\iota^{-1}(y) \ominus \iota^{-1}(y) + d), \end{aligned}$$

и однозначно восстановить формальное интервальное решение  $x^*$  по  $y^*$  из соотношения  $x^* = \iota^{-1}(y^*)$ .

**Определение.** Биективные вложения  $\iota : \mathbb{K}^n \rightarrow U$ , которые

- сохраняют аддитивную алгебраическую структуру  $\mathbb{K}^n$ , т. е. такие, что  $\iota(u + v) = \iota(u) + \iota(v)$  для любых  $u, v \in \mathbb{K}^n$ ,
- сохраняют топологическую структуру  $\mathbb{K}^n$ , т. е. такие, что само отображение  $\iota : \mathbb{K}^n \rightarrow U$  и его обратное  $\iota^{-1} : U \rightarrow \mathbb{K}^n$  непрерывны,

будем называть **погружениями интервального пространства**  $\mathbb{K}^n$  в линейное пространство  $U$ .

Определение свойств погружения позволяет сделать два важных вывода. Во-первых, оно однозначно задает линейное пространство:  $U$  должно быть евклидовым пространством  $\mathbb{R}^{2n}$ . Во-вторых, из определения погружения следует, что любые два погружения  $\mathbb{K}^n$  в  $\mathbb{R}^{2n}$  одинаковы с точностью до неособенного линейного преобразования  $\mathbb{R}^{2n}$ .

**Определение.** Погружение  $sti : \mathbb{K}^n \rightarrow \mathbb{R}^{2n}$ , которое действует по правилу

$$(x_1, x_2, \dots, x_n) \mapsto (-\underline{x}_1, -\underline{x}_2, \dots, -\underline{x}_n, \bar{x}_1, \bar{x}_2, \dots, \bar{x}_n),$$

т.е. такое, при котором взятые с противоположным знаком левые концы интервалов  $x_1, x_2, \dots, x_n$  становятся первой, второй, ...,  $n$ -ой компонентами точечного  $2n$ -вектора, а правые концы  $x_1, x_2, \dots, x_n$  становятся  $(n+1)$ -ой, ...,  $2n$ -ой компонентами точечного  $2n$ -вектора соответственно, будем называть **стандартным погружением интервального пространства**  $\mathbb{K}^n$  в  $\mathbb{R}^{2n}$ .

## 2.2 Особенные матрицы

Интервальная матрица  $A \in \mathbb{R}^{n \times n}$  называется неособенной (невырожденной), если неособенными являются все точечные  $n \times n$ -матрицы  $A \in \mathbf{A}$ . Интервальная матрица  $A \in \mathbb{R}^{n \times n}$  называется особенной (вырожденной), если она содержит особенную точечную матрицу.

Теорема

Пусть интервальная матрица  $A \in \mathbb{R}^{n \times n}$  такова, что  $\text{mid } A$  неособенная и

$$\max_{1 \leq j \leq n} (\text{rad } A * |(\text{mid } A)^{-1}|)_{jj} \geq 1 \quad (5)$$

Тогда  $A$  особенная.

## 2.3 Субдифференциальный метод Ньютона

Итерационный метод строится по следующей формуле:

$$x^{(k)} = x^{(k-1)} - \tau(D^{(k-1)})^{(-1)} * F(x^{(k-1)}) \quad (6)$$

где:

$(D^{(k-1)})^{(-1)} * F(x^{(k-1)})$  - субградиент в  $x^{(k-1)}$

$\tau \in [0, 1]$  - релаксационный параметр, с помощью которого можно расширить область сходимости. На практике рекомендуется брать  $\tau = 1$ , тогда метод даст наиболее точное решение. В этой работе  $\tau = 1$ .

При этом:

$$F(y) = sti(Asti^{-1}(y)b) \quad (7)$$

$$sti(x) : (x_1, \dots, x_n) \rightarrow (\underline{x_1}, \dots, \underline{x_n}, \overline{x_1}, \dots, \overline{x_n}) \quad (8)$$

## 2.4 Алгоритм метода

- Выбираем начальное приближение  $x^0 \in \mathbb{R}^{2n}$
- Применяя итерационный алгоритм находим следующее приближение  $x^k$ . Для этого считаем субградиент  $D^{k-1}$  отображения F в точке  $x^{k-1}$ :  
 $x^{(k)} \leftarrow x^{(k-1)} - \tau(D^{(k-1)})^{(-1)} * F(x^{(k-1)})$

## 3 Реализация

Лабораторная работа выполнена на языках программирования Python и C в среде разработки PyCharm и Visual Studio.

## 4 Результаты

### 4.1

Будем использовать параметр релаксации  $\tau = 1$  в методе Ньютона.

С помощью теоремы получили, что субдифференциальный метод на данной системе сходится за конечное число итераций.

в качестве начального приближения  $x^{(0)}$  мы можем выбрать решение специальной  $2n \times 2n$ -системы точечных уравнений:

$$(I - (midC)^\sim)x = sti(d)$$

Ниже приведены результаты на каждой итерации работы метода:

k	$x^k$
0	$\begin{pmatrix} [-1.3323, -1.3323] \\ [-0.1341, -0.1341] \end{pmatrix}$
1	$\begin{pmatrix} [-0.1190, -1.1689] \\ [-0.1270, -0.1484] \end{pmatrix}$

Видим, что метод имеет высокую сходимость. Результат был получен на первой итерации работы алгоритма.

## 4.2 Измененная ИСЛАУ

Исследуем поведение работы метода путем внесения изменений в систему. Изменим радиус вектора  $d$ . Увеличим его на 4. После изменений вектор  $d$  выглядит следующим образом:

$$d = \begin{pmatrix} [-2.5, 5.5] \\ 3 \end{pmatrix} \quad (9)$$

Результаты после внесения изменений:

k	$x^k$
0	$\begin{pmatrix} [-0.5378, 0.2714] \\ [-0.0971, -0.1711] \end{pmatrix}$
1	$\begin{pmatrix} [0.2498, -0.4906] \\ [-0.1656, -0.1024] \end{pmatrix}$
2	$\begin{pmatrix} [0.3431, -0.6741] \\ [-0.1634, -0.1075] \end{pmatrix}$

Видим, что результаты были получены на 2 итерации цикла. Увеличим вторую компоненту вектора  $d$ , аналогичным образом. Вектор  $d$  примет следующий вид:

$$d = \begin{pmatrix} 1.5 \\ [-1, 7] \end{pmatrix} \quad (10)$$

Получим следующий результат:

k	$x^k$
0	$\begin{pmatrix} [-0.1085, -0.1579] \\ [-0.3315, 0.0632] \end{pmatrix}$
1	$\begin{pmatrix} [-0.1447, -0.1281] \\ [0.0616, -0.3166] \end{pmatrix}$
2	$\begin{pmatrix} [-0.1428, -0.1416] \\ [0.0714, -0.3668] \end{pmatrix}$

Таким образом, увеличивая радиус компоненты  $d_1$ , мы «превратили» компоненту вектора-решения  $x_2$  в правильный интервал. И наоборот: «расширяя» компоненту  $d_2$ , мы добились того, что компонента  $x_1$  стала правильным интервалом. Аналогичного результата можно добиться путем поочередного увеличения радиусов элементов интервальной матрицы  $C$ . Проверим, какой результат получится, если увеличить обе компоненты вектора  $d$ .

$$d = \begin{pmatrix} [-2.5, 5.5] \\ [-1, 7] \end{pmatrix} \quad (11)$$

Получим следующий результат:

Видим, что обе компоненты вектора  $x$  являются неправильными векторами. Сравним последнее полученное нами решение  $x$  ИСЛАУ с расширенным вектором  $d$  с решением  $x^{(1)}$  исходной ИСЛАУ:

$$x^{(1)} = \begin{pmatrix} [-0.1190, -1.1689] \\ [-0.1270, -0.1484] \end{pmatrix} \quad (12)$$

k	$x^k$
0	$\begin{pmatrix} [-0.5131, 0.24672] \\ [-0.2945, 0.02621] \end{pmatrix}$
1	$\begin{pmatrix} [0.2241, -0.4621] \\ [0.0230, -0.2772] \end{pmatrix}$
2	$\begin{pmatrix} [0.3088, -0.6467] \\ [0.0290, -0.3259] \end{pmatrix}$

Поскольку к неправильному интервалу в полной интервальной арифметике Каухе-ра неприменимо понятие «ширина», определим правильные проекции сравниваемых векторов:

$$pro(x) = \begin{pmatrix} [-0.6467, 0.3088] \\ [-0.3259, 0.0290] \end{pmatrix} \quad (13)$$

$$pro(x^{(1)}) = \begin{pmatrix} [-1.1689, -0.1190] \\ [-0.1484, -0.1270] \end{pmatrix} \quad (14)$$

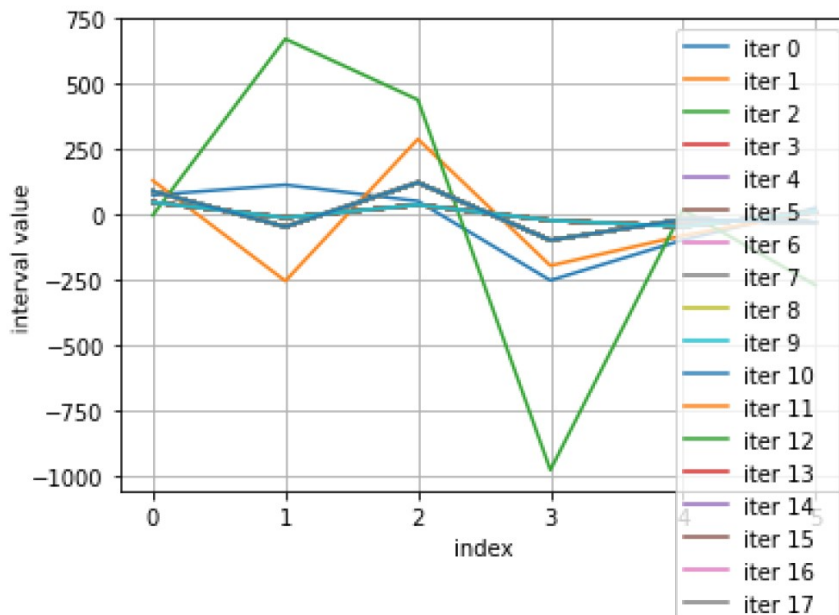
Заметим, что радиусы элементов вектора  $pro(x)$ , относящегося к ИСЛАУ с расширенным вектором  $d$ , значительно больше радиусов элементов вектора  $pro(x^{(1)})$ , который является решением исходной ИСЛАУ с точечным вектором  $d$ .

### 4.3

Для ИСЛАУ 3 метод останавливается после 100 итераций, но так и не сходится к точному значению. Получаем следующий выход метода

$$\begin{pmatrix} [-1.66666, -56.6666] \\ [-46.1111, 19.44444] \\ [41.22222, -8.77777] \end{pmatrix} \quad (15)$$

Посмотрим на график изменения значений выходного вектора в зависимости от числа итераций, по оси  $x$  отмечены индексы вектора интервалов, так например в интервале  $[a_{11}, a_{12}] a_{11} = 1 a_{12} = 2$



С помощью теоремы из теории получаем, что матрица является особенной. Субдифференциальный метод на данной системе не сходится за конечное число итераций. Для этой интервальной линейной системы алгоритм генерирует осциллирующую последовательность, которая, очевидно, не сходится ни к какому пределу. Интересно отметить, что правая часть этой системы шире, чем в предыдущем примере, а все элементы матрицы, кроме  $a_{33}$ , тоньше, что говорит о том, что данная ИСЛАУ еще лучше подчиняется теореме о сходимости. Тем не менее метод не работает.

#### 4.4 Внесение изменений

Для исследования поведения метода внесем изменения в компоненту  $a_{33}$  и проверим сходимость метода. Результаты коррекции значения  $a_{33}$  при  $\tau = 1$  :

$A_{33-inf}$	Results
0.0	Diverged
0.5	Diverged
1.0	Diverged
1.5	Converged
2.0	Diverged
2.5	Diverged
3.0	Diverged
3.5	Converged
4.0	Diverged
4.5	Diverged
5.0	Converged
5.5	Converged
6.0	Diverged
6.5	Diverged
7.0	Converged
7.5	Diverged
8.0	Converged
8.5	Converged
9.0	Converged
9.5	Converged

С каждым шагом изменялся  $\inf$  значение интервала  $a_{33}$ . Стоит заметить, что во всех случаях, когда мы замечаем сходимость интервал  $a_{33}$  будет неправильный.

Проведем еще один эксперимент и попробуем изменить параметр релаксации на  $\tau = 0.6$ :



$A_{33-inf}$	Results
0.0	Diverged
0.5	Diverged
1.0	Diverged
1.5	Converged
2.0	Converged
2.5	Converged
3.0	Converged
3.5	Converged
4.0	Converged
4.5	Converged
5.0	Converged
5.5	Converged
6.0	Converged
6.5	Converged
7.0	Converged
7.5	Converged
8.0	Converged
8.5	Converged
9.0	Converged
9.5	Converged

Стоит обратить внимание на то, что область сходимости расширилась.

## 5 Выводы

- Реализован субдифференциальный метод Ньютона. Полученные результаты свидетельствует о высокой скорости сходимости субдифференциального метода Ньютона.
- Алгоритм дает точное решение задачи за небольшое конечное число итераций. В некоторых случаях параметр релаксации  $\tau < 1$  улучшает сходимость субдифференциального метода Ньютона. Но чем меньше  $\tau$ , тем медленнее работает алгоритм.
- Варьируя ширину или, что равносильно, радиусы элементов интервальных матрицы  $\mathbf{C}$  и вектора  $\mathbf{d}$ , можно в достаточно широких пределах управлять решением ИСЛАУ вида  $\mathbf{x} = \mathbf{C}\mathbf{x} + \mathbf{d}$
- Метод показывает преимущества при решении ИСЛАУ, в которой содержатся неправильные интервалы в свободной части и нет знакопеременности интервалов: быструю сходимость и точность.