Київський національний університет

імені Тараса Шевченка

Факультет комп’ютерних наук та кібернетики

Кафедра обчислювальної математики

Кваліфікаційна робота

на здобуття ступеня бакалавра

за спеціальністю 113 Прикладна математика

на тему:

Веб система тестування ефективності алгоритмів для варіаційних нерівностей

Виконав студент 4-го курсу

Ляшкевич Андрій Володимирович ______________

Науковий керівник:

асистент

Денисов Сергій Вікторович ______________

Засвідчую, що в цій роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент ______________

Роботу розглянуто й допущено до захисту на засіданні кафедри обчислювальної математики

«___»________________________202_ р.

Протокол №___

Завідувач кафедри

С.І. Ляшко ______________

Київ – 2022

РЕФЕРАТ

Обсяг роботи 21 сторінка, 12 ілюстрацій, 10 джерел посилань.

ВАРІАЦІЙНА НЕРІВНІСТЬ, ВЕБ СИСТЕМА, ІТЕРАЦІЙНИЙ МЕТОД, МЕТОД ПРОЕКТУВАННЯ, МОНОТОННИЙ ОПЕРАТОР.

Об’єктом роботи є процес розв’язування варіаційних нерівностей за допомогою розробленого програмного засобу та розробка засобу його тестування.

Мета роботи – розробка веб системи для тестування класичних методів розв’язання варіаційних нерівностей.

Інструменти розроблення: середовище програмування Microsoft Visual Studio Code, мова програмування Python, фреймворк для розробки веб систем Django.

Результати роботи: було досліджено простий проекційний метод та методи Корпелевич, Tseng’a, Попова та проектування з відбиттям, а також реалізовано веб систему для тестування їх ефективності на заданій користувачем задачі.

ЗМІСТ

1. Вступ

Багато прикладних математичних задач зводяться до знаходження розв'язку варіаційної нерівності. Наприклад, мінімізація диференційовної функції, пошук сідлової точки, пошук рівноваги за Нешем, пошук рівноваги у транспортних мережах. Багато математиків, зокрема Корпелевич Г. М., Попов Л. Д., П. Цзен, В.В.Семенов та Ю. В. Маліцький - доклали зусилля для розробки методів розв'язання варіаційних нерівностей. Ці методи продовжують розвиватися і на даний момент.

1.1. Варіаційні нерівності

Розглянемо деякий оператор A, такий що $A: C \to H$, де H – гільбертів простір, а $C \in H$.

Визначення. [1] Варіаційна нерівність це нерівність вигляду

$$\langle A(x), y - x \rangle \geq 0, \qquad \forall y \in C \tag{1}$$

Визначення. На множині C оператор A є монотонним, якщо виконується:

$$\langle A(x) - A(y), x - y \rangle \geq 0, \qquad \forall x, y \in C$$

Визначення. На множині C оператор A є сильно монотонним, якщо $\exists \alpha > 0$ таке, що виконується:

$$\langle A(x) - A(y), x - y \rangle \geq \alpha \|x - y\|^2, \qquad \forall x, y \in C$$

Визначення. На множині C оператор A є Ліпшицевим з константою L, якщо виконується:

$$\|A(x) - A(y)\| \leq L\|x - y\| \;\; \forall x, y \in C$$

Теорема. [2] Якщо A – неперервний оператор, а C – непорожня, опукла множина, то варіаційна нерівність (1) має розв'язок.

Доведення. Відображення $x \mapsto P_C(x - \lambda Ax)$ – є неперервний, і діє з C в C, де C – опуклий компакт. Тоді, за теоремою Брауера [4], воно має нерухому точку. З твердження, що множина розв'язків варіаційної нерівності співпадає з множиною нерухомих точок оператора, випливає що ця точка буде розв'язком варіаційної нерівності (1).

1. Застосування варіаційних нерівностей

Розглянемо деякі найпростіші приклади задач, що зводяться до варіаційних нерівностей.

1.1. Задача оптимізації

Нехай потрібно знайти мінімум функції f на C

$$f(x) \to \min_{C}$$

при опуклих f та C, x буде розв'язком коли виконуватиметься нерівність

$$\langle f'(x), y - x \rangle \geq 0, \qquad \forall y \in C$$

Доведення. Лінійна апроксимація f має вигляд:

$$f(y) = f(x) + \langle f'(x), y - x \rangle + o(\|y - x\|),$$

припускаючи, що другий доданок менше нуля за певного $y = y + z$, тоді

$$f(x + z) - f(x) = \langle f'(x), z \rangle + o(\|z\|).$$

Тепер поставимо $y = y + \varepsilon z$, звідки отримаємо

$$f(x + \varepsilon z) - f(x) = \langle f'(x), \varepsilon z \rangle + o(\|\varepsilon z\|).$$

При $\varepsilon \to +0$ перший доданок визначатиметься знаком правої частини, що зрозуміло з визначення $o(\cdot)$.

Це означає, що для достатньо малого ε права частина буде менше нуля. Тоді і ліва частина теж буде від'ємною, $f(x + \varepsilon z) - f(x) < 0$. Отже, $f(x + \varepsilon z) < f(x)$. Тоді x не буде мінімумом f на C. Отримане протиріччя завершує доведення.

Зауваження. У випадку коли f або C не будуть опуклими, то наш критерій буде необхідною умовою.

1.2. Задача пошуку сідлової точки.

Розглянемо задачу

$$f(x) \xrightarrow[g_i(x)\le 0]{} \min, \qquad i = 1..n,$$

яка є оптимізацією з обмеженнями. Для даної задачі можна побудувати функцію Лагранжа

$$L(x,y) = f + \sum_{i=1}^{n} y_i g_i(x),$$

де y_i – множники Лагранжа.

В такому випадку постає задача пошуку сідлової точки для функції L.

Визначення. Сідловою точкою функції L називають точку $(\bar{x},\bar{y})$, якщо

$$L(\bar{x},y) \le L(\bar{x},\bar{y}) \le L(x,\bar{y}), \qquad \forall x \forall y.$$

Іншими словами можна сказати, що по x досягається мінімум в $\bar{x}$, а по y – в $\bar{y}$.

Ці умови можна записати так:

$$\begin{cases} \langle \nabla_1 L(\bar{x},\bar{y}), x-\bar{x}\rangle \ge 0, & \forall x \in c_1 \subseteq H_1, \\ \langle -\nabla_2 L(\bar{x},\bar{y}), y-\bar{y}\rangle \ge 0, & \forall y \in C_2 \subseteq H_2, \end{cases}$$

Зауваження. Дані нерівності можна звести до одної, розглядаючи пряму суму просторів та відповідний оператор.

Надалі будемо розв'язувати таку задачу:

знайти $x \epsilon C$ такий, що задовольняє нерівність:

$$\langle Ax, y-x\rangle \ge 0, \qquad \forall y \in C.$$

Також вважатимемо, що виконуються наступні умови:

- множина $C \subseteq H$ – опукла і замкнена;
- оператор $A: H \to H$ – монотонний і ліпшицевий
- множина розв'язків нерівності є непорожньою.

2. Алгоритми

2.1. Найпростіший проекційний метод

Даний метод є найпростішим для розуміння і водночас легкий в реалізації, тому він є хорошим для того, щоб почати з нього. Отже, розглянемо ітераційний процес [3]:

1) *задамо* $x_0 \in C, \lambda > 0$.
2) *для* x_n *обчислимо*

$$x_{n+1} = P_C(x_n - \lambda A x_n)$$

3) *алгоритм зупиняється, якщо* $x_n = x_{n+1}$, *інакше замінюємо n на n + 1 і повторюємо обчислення* x_{n+1}.

Після зупинки алгоритму x_n буде розв'язком варіаційної нерівності, а рівність

$$x_{n+1} = P_C(x_n - \lambda A x_n)$$

є рівносильною варіаційній нерівності

$$(x_{n+1} - x_n + \lambda A x_n, x - x_{n+1}) \geq 0, \qquad \forall x \in C.$$

А враховуючи умову $x_n = x_{n+1}$ бачимо, що x_n належить множині розв'язків варіаційної нерівності.

2.2. Метод Корпелевич

Даний екстраградієнтний метод є одним із найпопулярніших методів розв'язування варіаційних нерівностей. Розглянемо його ітераційний процес [5]:

1) *задамо* $x_0 \in C, \lambda \in \left(0, \frac{1}{L}\right)$.
2) *для* x_n *обчислимо* $y_n = P_C(x_n - \lambda A x_n)$.
3) *алгоритм зупиняється, якщо* $x_n = y_n$, *інакше обчислимо*

$$x_{n+1} = P_C(x_n - \lambda A y_n),$$

і замінюємо n на n + 1 та повторюємо обчислення y_n.

Твердження. Якщо $x_n = y_n$, то x_n належить множині розв'язків варіаційної нерівності.

Доведення. Рівність $y_n = P_C(x_n - \lambda A x_n)$ є рівносильною варіаційній нерівності $(y_n - x_n + \lambda A x_n, x - y_n) \geq 0, \forall y \in C$. А враховуючи умову $x_n = y_n$ отримуємо, що x_n належить множині розв'язків варіаційної нерівності.

2.3. Метод Tseng'a

Варто згадати, що ми припускали те, що оператор $A: H \to H$ є монотонним і ліпшицевим на всьому H. Розглянемо ітераційний алгоритм [2]:

1) *задамо $x_0 \in H, \lambda \epsilon \left(0, \frac{1}{L}\right)$.*
2) *для x_n обчислимо $y_n = P_C(x_n - \lambda A x_n)$.*
3) *алгоритм зупиняється, якщо $x_n = y_n$, інакше обчислимо*

$$x_{n+1} = y_n - \lambda(A y_n - A x_n)$$

і замінюємо n на n + 1 та повторюємо обчислення y_n.

Даний метод є схожим на метод Корпелевич, тут так само обчислюється y_n, а для обчислення x_n не виконується додаткове проектування на множину C. Зрозуміло, що $A x_n$ не обчислюється додатково, а просто використовується двічі.

2.4. Метод Попова

Також, альтернативним методу Корпелевич можна назвати метод Попова. Розглянемо ітераційний алгоритм [7]:

1) *задамо $x_0 = y_0 \in C, \lambda \epsilon \left(0, \frac{1}{3L}\right)$.*
2) *для x_n та y_n обчислимо*

$$x_{n+1} = P_C(x_n - \lambda A y_n)$$

$$y_{n+1} = P_C(x_{n+1} - \lambda A y_n)$$

3) *алгоритм зупиняється, якщо* $x_n = y_n = x_{n+1}$*, інакше замінюємо n на n + 1 та повторюємо обчислення* x_{n+1} *та* y_{n+1}.

Як бачимо в даному методі для одної ітерації ми використовуємо тільки Ay_n, тобто вона вимагає лише одного обчислення оператора, що може покращити швидкість знаходження розв'язку задачі.

2.5. Метод проектування з відбиттям

В даному методі проводячи «дзеркальне відбиття» точки x_n від гіперплощини $T = \{y \in H: (x_{n+1} - x_n, y - x_{n+1}) = 0\}$, ми отримуємо точку y_{n+1}. Розглянемо ітераційний алгоритм [8]:

1) *задамо* $x_0 = y_0 \in C, \lambda \epsilon \left(0, \frac{1}{3L}\right)$.
2) *для* x_n *та* y_n *обчислимо*

$$x_{n+1} = P_C(x_n - \lambda A y_n)$$

3) *алгоритм зупиняється, якщо* $x_n = y_n = x_{n+1}$*, інакше обчислимо*

$$y_{n+1} = 2x_{n+1} - x_n$$

та замінюємо n на n + 1 та повторюємо обчислення x_{n+1}.

3. Практична частина. Реалізація веб системи.

У цьому розділі детально розглянемо реалізацію алгоритмів які розв'язують варіаційні нерівності, використовуючи мову програмування Python [9]. А також розробку веб системи для тестування ефективності алгоритмів для варіаційних нерівностей за допомогою фреймворку для розробки веб систем – Django [10].

Користувач, який бажає протестувати ефективність того чи іншого алгоритмів для конкретної задачі повинен ввести всі вхідні дані. До таких даних відносяться: оператор A, межі підпростору C та λ. Для того щоб вводити ці дані було зручно створюється таблиця із полями в які можна вводити елементи оператора A, а також верхні і нижні межі C. Але перед тим як її створювати потрібно знати розмірність оператора. Для цього створено окрему сторінку із всього одним полем:

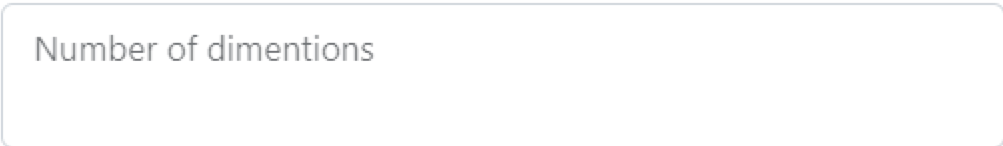

Рис. 1

Нижче поля знаходиться кнопка, яка дозволяє перейти до введення безпосередньо умов задачі:

Рис. 2

Після натискання кнопки відбувається перевірка чи введені в поле дані є натуральним числом. Якщо це не так, то перехід на сторінку для введення умов не відбувається. Загальний вигляд даної сторінки зображено на Рис.3.

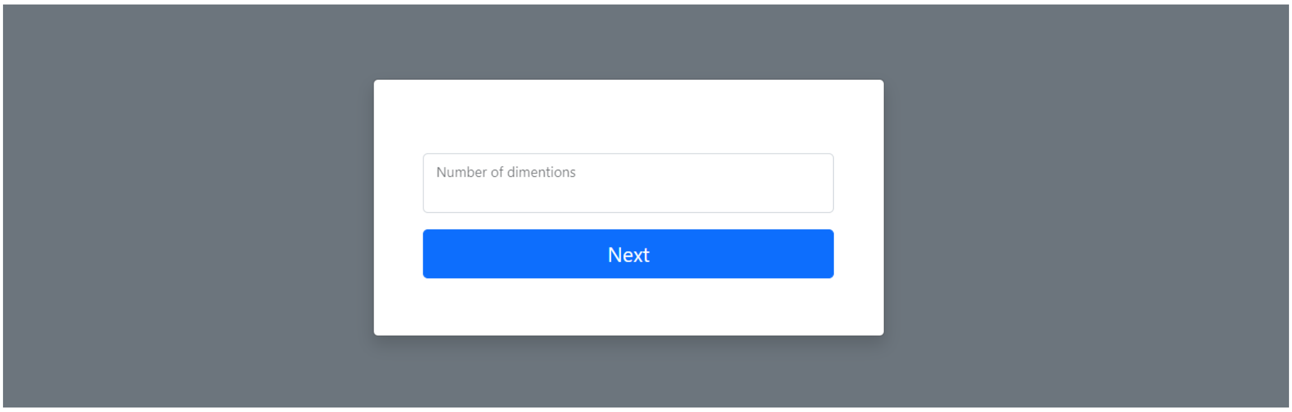

Рис. 3

Якщо ж розмірність була введена правильно, то відбувається перехід на сторінку, на якій створюється таблиця із $n \times (n+2)+2$ полями. Приклад сторінки для розмірності 3 зображений на Рис. 4.

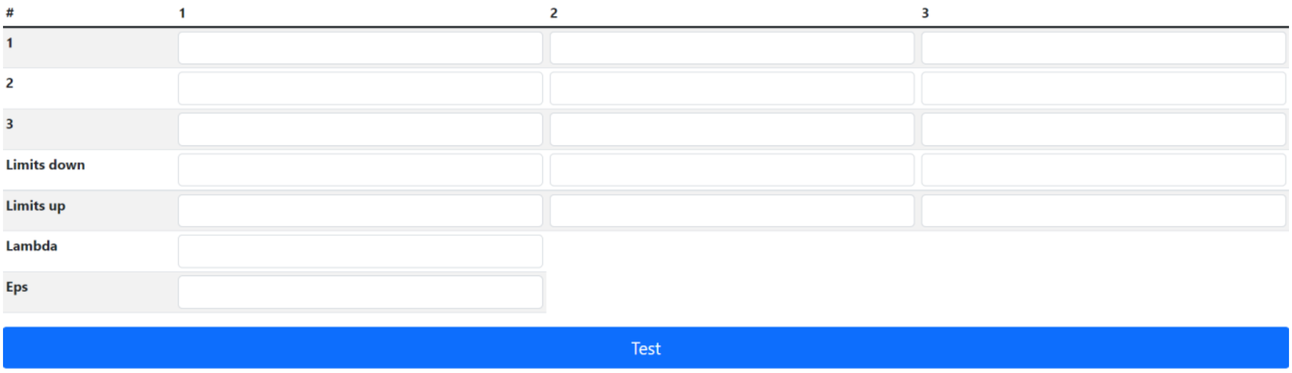

Рис. 4

Перші $n \times n$ полів використовуються для введення оператора A. Наступні $n \times 2$ – для того, щоб ввести границі множини C. І одне поле для введення λ.

Також в останньому полі слід ввести бажану точність розв'язку (ε). Дана точність використовується для того, щоб значно зменшити кількість ітерацій алгоритмів, так як найбільше приближення до розв’язку відбувається на перших ітераціях, а з кожною наступною ітерацією відстань до розв’язку скорочується все менше і менше.

Відповідно у всіх методах, де було зазначено, що алгоритм зупинятиметься при $x_n = x_{n+1}$ у нашому випадку алгоритми зупинятимуться за виконання умови:

$$|x_n, x_{n+1}| \leq \varepsilon,$$

де $|a, b|$ – відстань між точками a і b в n-вимірному просторі.

Після натискання на кнопку «Test» проводяться обчислення для кожного з алгоритмів і виводяться порівняльні графіки збіжності кожного всіх методів. На Рис. 5 зображено приклад макету сторінки із результуючими графіками:

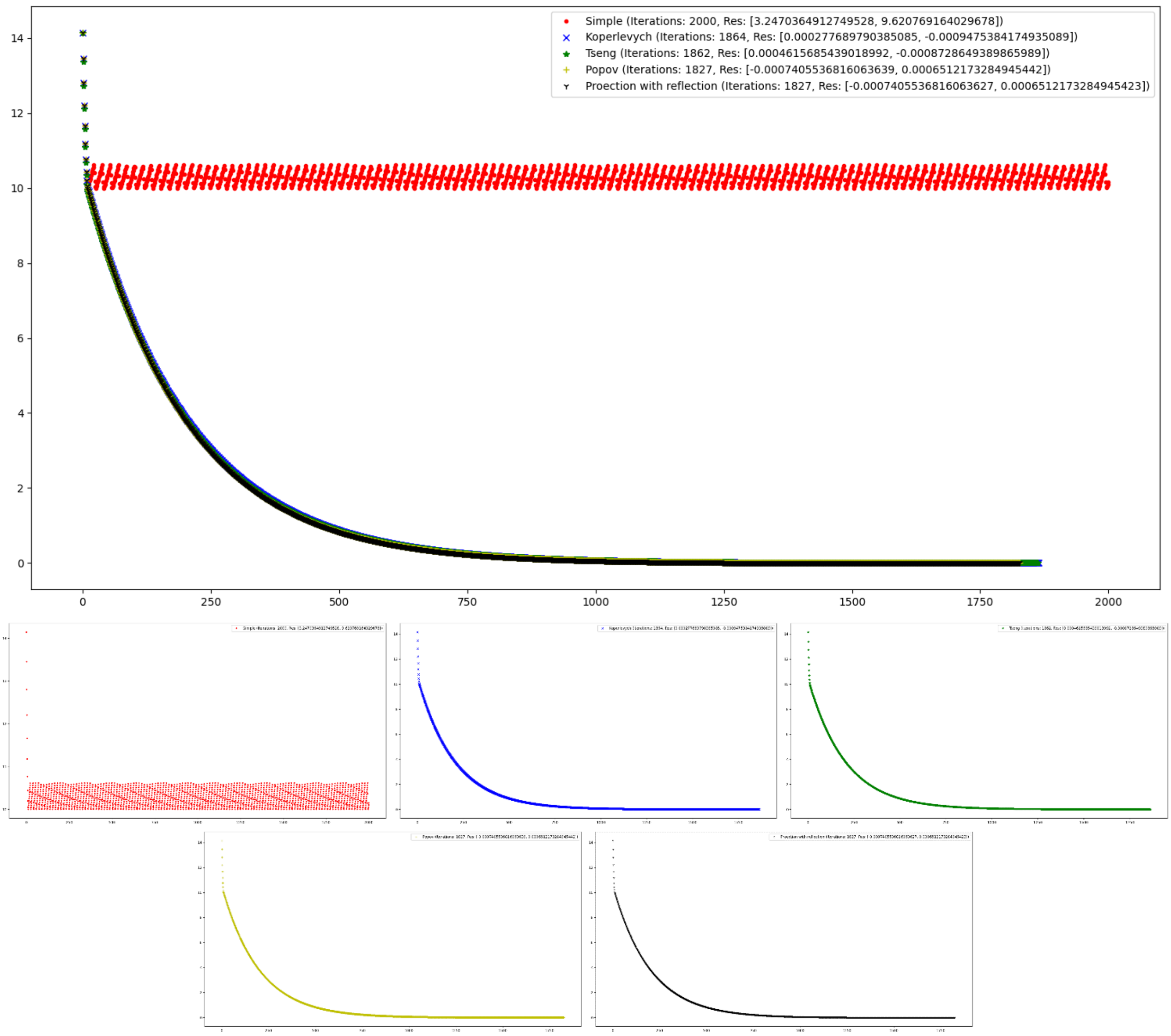

Рис. 5

4. Результати тестування певних задач.

В цьому розділі буде показано результати тестування алгоритмів для певних варіаційних нерівностей.

Розглянемо для початку задачу оптимізації:

Дана задача є по суті задачею мінімізації і представляється одновимірним оператором для варіаційної нерівності. На Рис. 6 зображено результат тестування алгоритмів на прикладі коли розв'язок знаходиться всередині проміжку C ($A = [4]$; $C = [-10; 10]$):

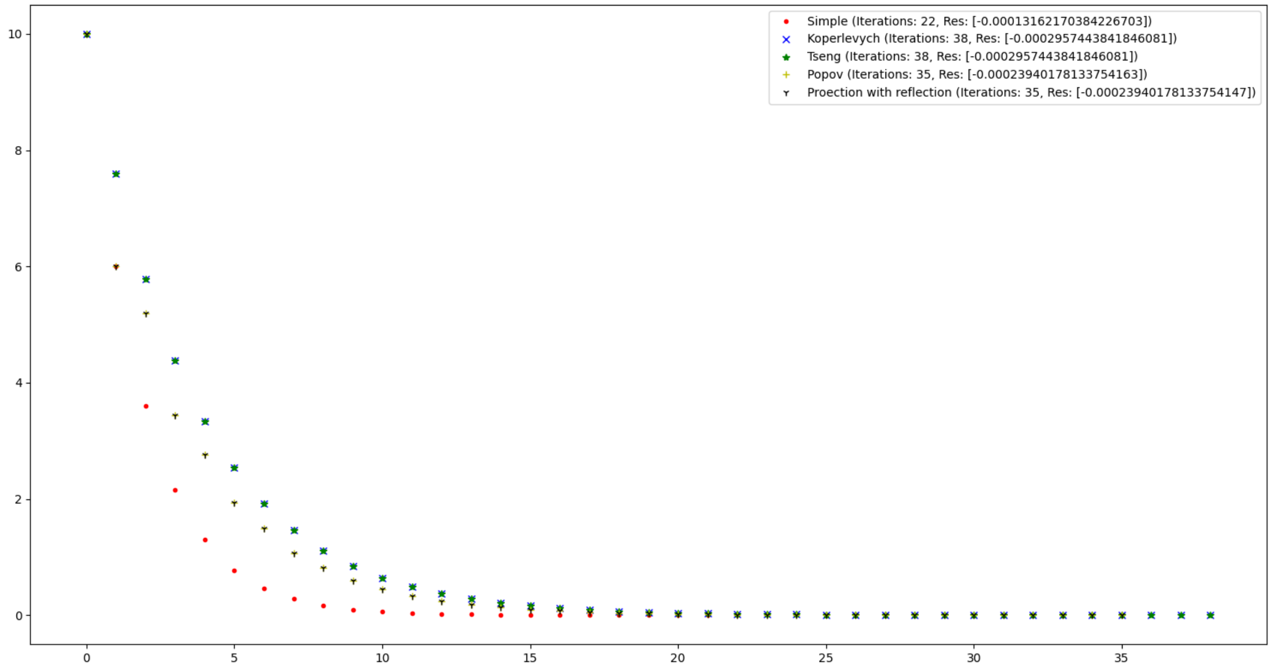

Рис. 6

Дані результати показують, що на тривіальній задачі найпростіший алгоритм збігається за найменшу кількість ітерацій, а якщо згадати що його ітераційний процес є найпростішим, то можна зробити висновок, що він значно швидкодійніший для таких задач. З іншого боку на Рис. 7 зображено результат тестування для задачі оптимізації, коли розв'язок знаходиться на краю C ($A = [-4]$; $C = [1; 400]$). В даному випадку найпростіший ітераційний метод вже не є настільки ж ефективним. Варто звернути увагу на метод Tseng'a. Якщо в якості результата брати x_n , то даний метод дає розв'язок, який виходить за межі C. Все тому що в ітераційному процесі даного алгоритму на множину C проектується тільки y_n, а от x_n – ні.

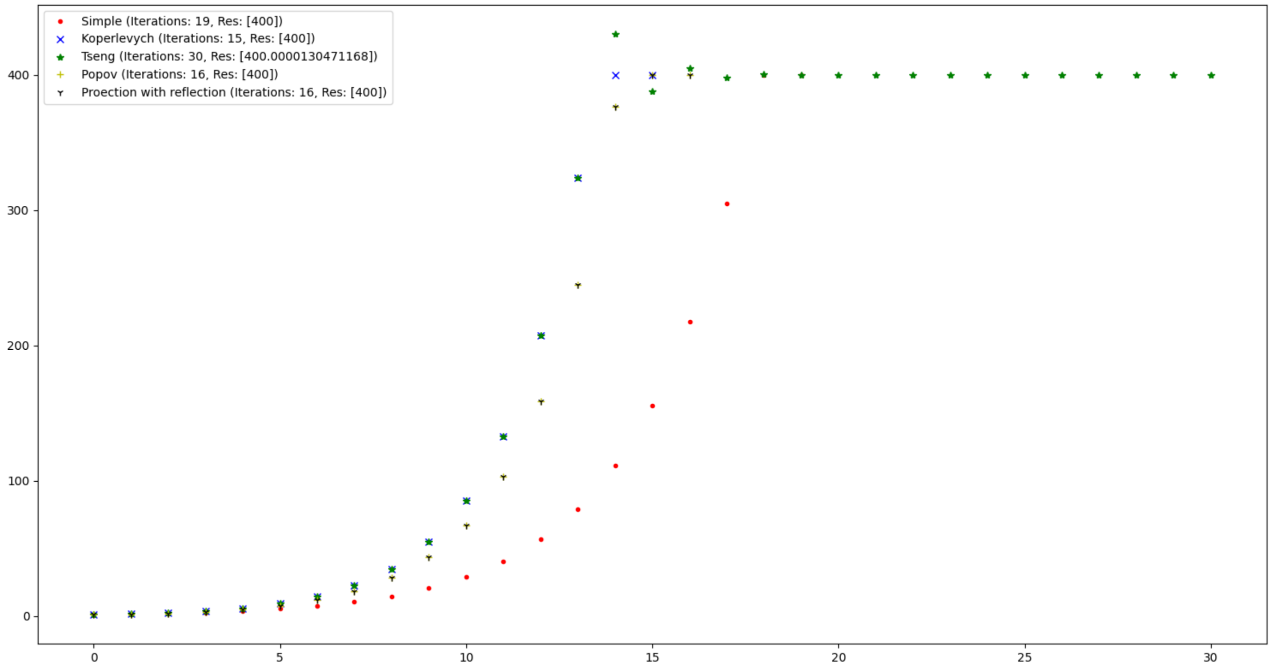

Рис. 7

Далі розглянемо задачу приклад тестування якої зображено в попередньому розділі (Рис. 5):

Оператор A задається наступним чином:

$$a_{i,j} = \begin{cases} -1, & j = m - 1 - i > i, \\ 1, & j = m - 1 - i < i, \\ 0, & \text{інакше.} \end{cases}$$

а C довільно задамо $[-10; 10]$ для всіх змінних x_i.

Відповідно за розмірності 2 оператор матиме вигляд:

$$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

Як бачимо на Рис. 8 перший метод не збігається до розв’язку нерівності (нульового вектора). Варто зазначити, що для того щоб у випадку незбіжності методу для певних нерівностей система не продовжувала обчислення нескінченну кількість часу накладено строге обмеження на максимальну кількість ітерацій в 2000.

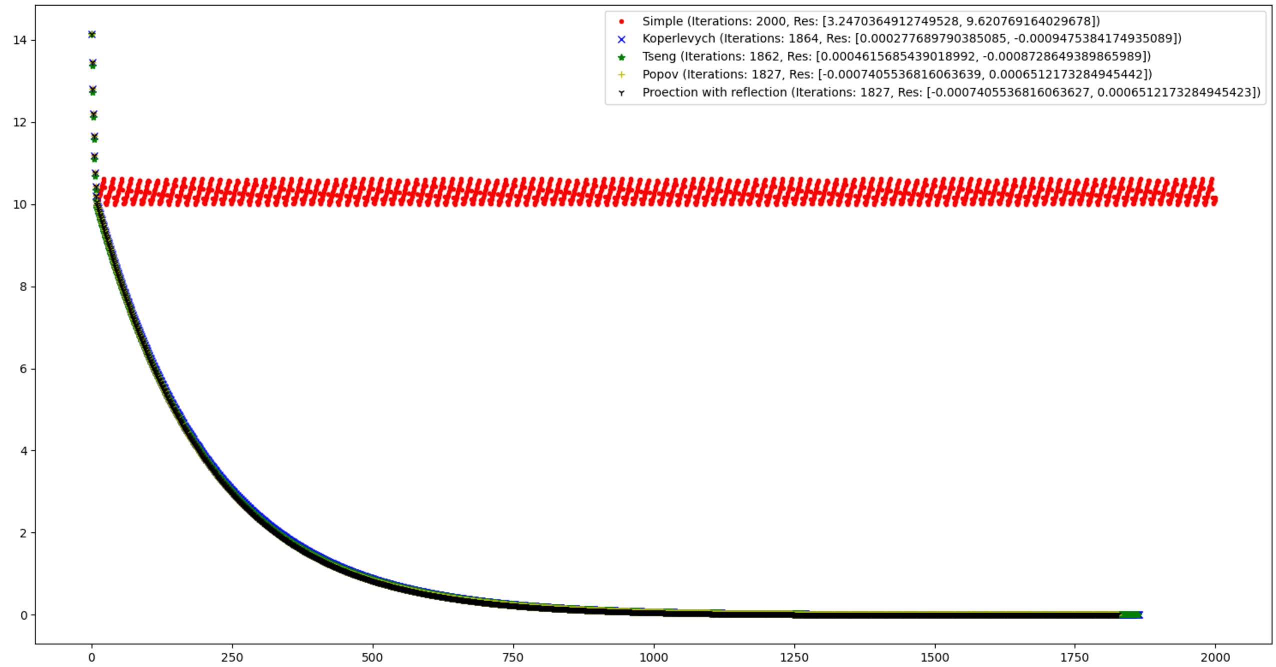

Рис. 8

Для такої ж задачі із розмірністю 4 теж було проведено тестування, результати якого зображені на Рис. 8, а оператор A має наступний вигляд:

$$\begin{pmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

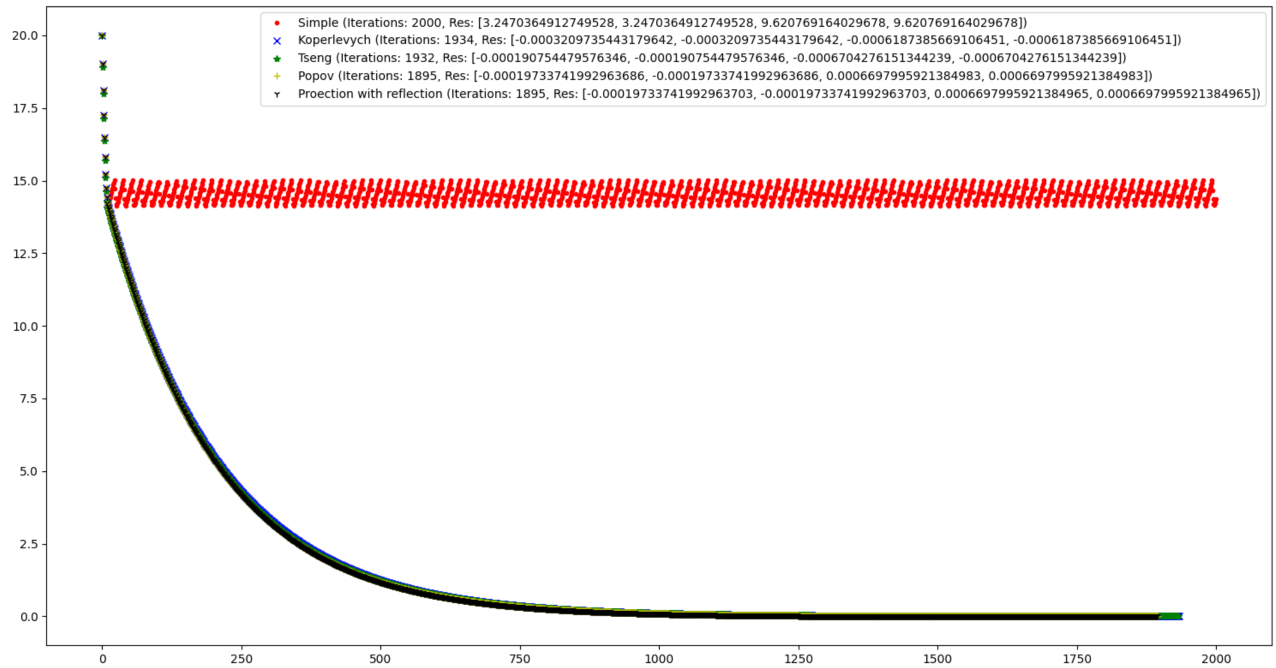

Рис. 9

Очевидно, що і для цієї задачі ітераційний метод не збігається, а інші збіжні за приблизно однакову кількість ітерацій, хоча методи Попова та Проекції з відбиттям знаходять розв'язок на $\approx 5\%$ швидше за методи Корпелевич та Tseng'a.

Для тестування системи було розглянуто іще одну задачу:

C так само довільно задамо $[-10; 10]$, а оператор A буде із наступними елементами:

$$a_{i,j} = \begin{cases} 1, & j = i-1 \\ 4, & j = i \\ -2, & j = i+1' \\ 0, & \text{інакше} \end{cases}$$

тобто для розмірності 3 оператор матиме наступний вигляд:

$$\begin{pmatrix} 4 & -2 & 0 \\ 1 & 4 & -2 \\ 0 & 1 & 4 \end{pmatrix},$$

а результати тестування зображені на Рис.10:

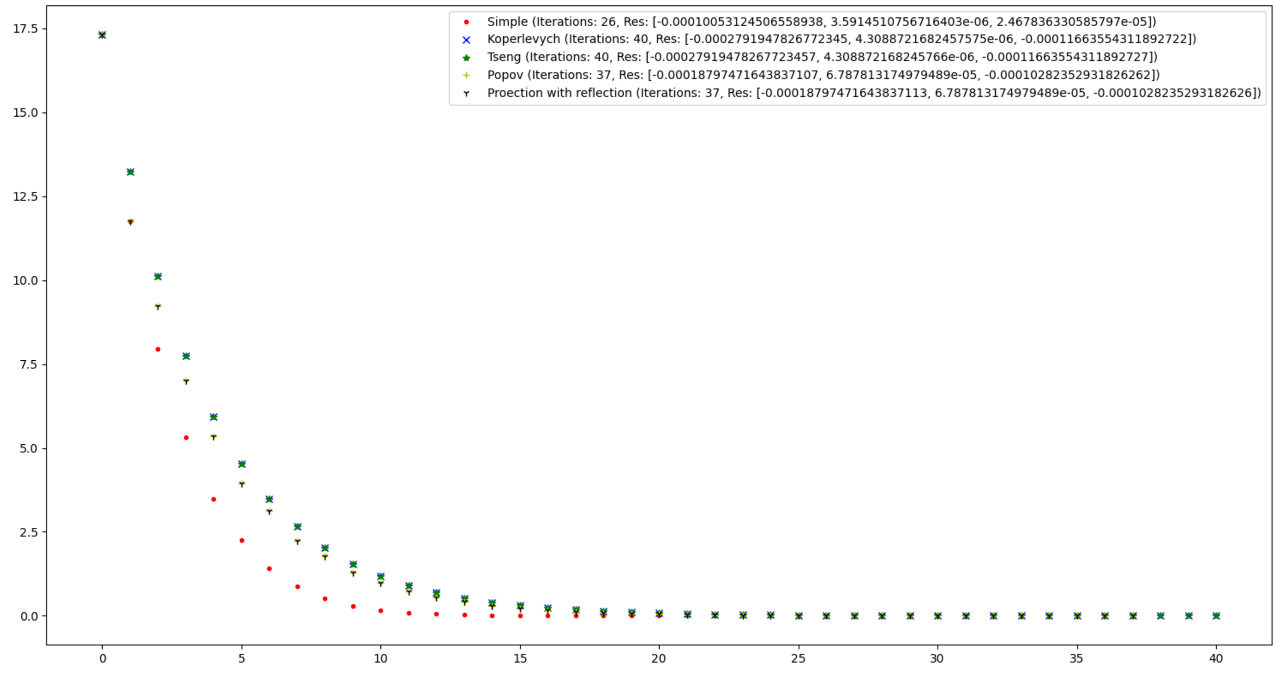

Рис. 10

Тут можемо бачити, що простий ітераційним метод знову виявився швидшим за інші методи приблизно у півтори рази, а також методи Попова та Проекції з відбиттям є дещо швидшими за Корпелевич та Tseng'a.

Для такої ж задачі розмірностей 4 та 5 оператор матиме вигляд:

$$\begin{pmatrix} 4 & -2 & 0 & 0 \\ 1 & 4 & -2 & 0 \\ 0 & 1 & 4 & -2 \\ 0 & 0 & 1 & 4 \end{pmatrix},$$

$$\begin{pmatrix} 4 & -2 & 0 & 0 & 0 \\ 1 & 4 & -2 & 0 & 0 \\ 0 & 1 & 4 & -2 & 0 \\ 0 & 0 & 1 & 4 & -2 \\ 0 & 0 & 0 & 1 & 4 \end{pmatrix},$$

а результати тестування зображені на Рис. 11 та Рис. 12, де результати не відрізняються суттєво від прикладу із розмірністю 3 і ми теж бачимо, що найпростіший метод є найшвидшим.

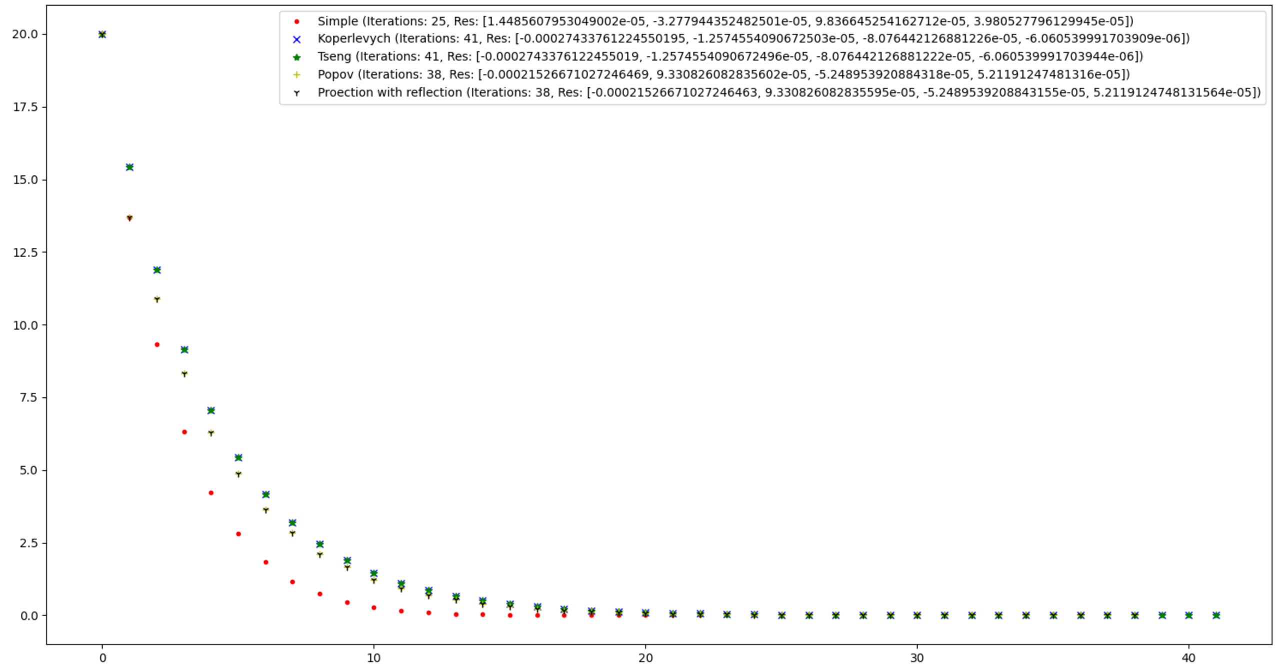

Рис. 11

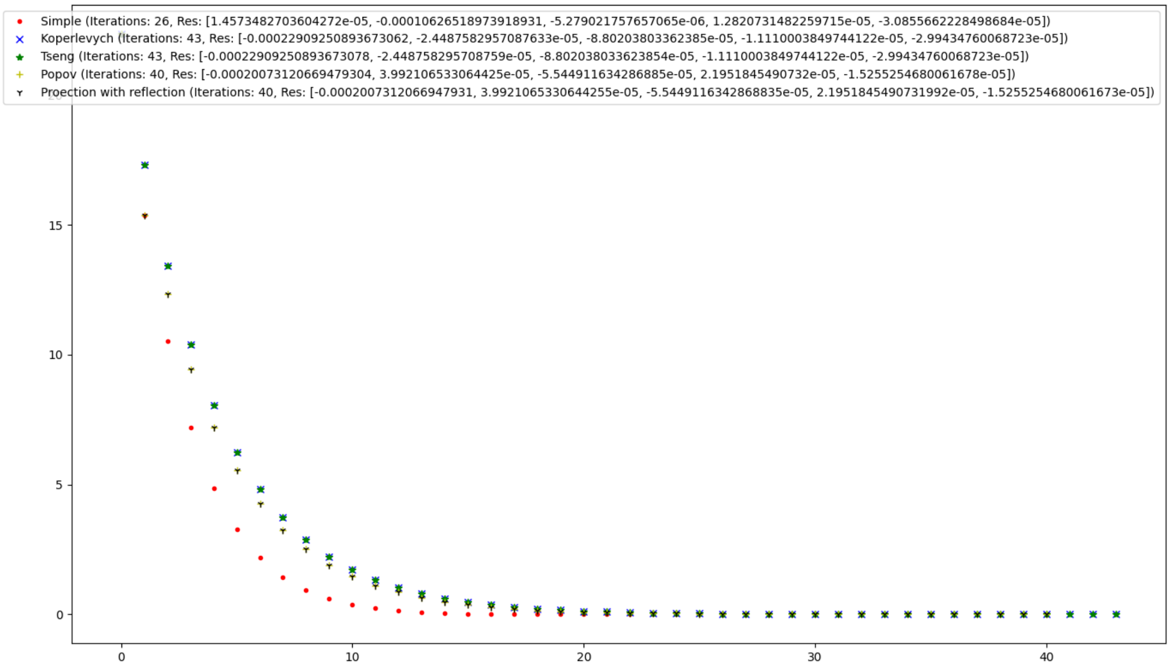
Simple (Iterations: 26, Res: [1.4573482703604272e-05, -0.00010626518973918931, -5.279021757657065e-06, 1.2820731482259715e-05, -3.0855662228498684e-05])
Koperlevych (Iterations: 43, Res: [-0.00022909250893673062, -2.4487582957087633e-05, -8.80203803362385e-05, -1.1110003849744122e-05, -2.99434760068723e-05])
Tseng (Iterations: 43, Res: [-0.00022909250893673078, -2.448758295708759e-05, -8.802038033623854e-05, -1.1110003849744122e-05, -2.99434760068723e-05])
Popov (Iterations: 40, Res: [-0.00020073120669479304, 3.992106533064425e-05, -5.544911634286885e-05, 2.1951845490732e-05, -1.5255254680061678e-05])
Proection with reflection (Iterations: 40, Res: [-0.0002007312066947931, 3.9921065330644255e-05, -5.5449116342868835e-05, 2.1951845490731992e-05, -1.5255254680061673e-05])
15
10
5
0
0
10
20
30
40

Рис. 12

5. Висновок

У вигляді варіаційних нерівностей можна сформулювати задачі опуклого програмування, задачі пошуку сідлових точок опукло-угнутих функцій та різні актуальні задачі пошуку рівноваги. Для різних задач одні алгоритми можуть працювати швидше, інші повільніше, а треті й зовсім не збігатись до розв’язку (в залежності від вибору кроку). Саме тому актуальною задачею є розробка системи для тестування алгоритмів, на заданих варіаційних нерівностях.

В рамках дипломної роботи були виконані наступні завдання:

- розглянуто певні методи розв’язання варіаційних нерівностей: простий ітераційний алгоритм, Корпелевич, Tseng’a, Попова, проекції з відбиттям;
- розроблено веб систему для тестування цим алгоритмів на конкретних задачах;
- проведено тестування на певних задач для наглядної перевірки веб системи на ефективність.

Зроблено акцент на зручності користування системою для звичайного середньостатистичного користувача.

Автор переконаний, що така веб система є корисною, ефективною та може використовуватись для подальших тестувань методів для розв’язування варіаційних нерівностей.

Бібліографія

[1] Семенов, В.В. Вступ до теорії варіаційних нерівностей / Семенов, В.В. // Варіаційні нерівності. Навчальний посібник для студентів факультету кібернетики – 2014. – P. 4-23.

[2] Tseng, P. on linear convergence of iterative methods for the variational inequality problem / P. Tseng // Journal of Computational and Applied Mathematics. – 1995. – Vol. 60. – P. 237-252.

[3] Ляшко С.І., Гончаренко Ю.В., Клюшин Д.А., Семенов В.В., Бондар О.С., Тимошенко А.А., Номіровський Д.А., Чабак Л.М., Ведель Я.І. // Математичні моделі та обчислювальні процеси. – 2019. – P. 113-151.

[4] Стампаккья Г., Киндерлерер Д. Введение в вариационные неравенства и их приложения. Москва, Мир – 1983.

[5] Корпелевич, Г.М. Экстраградиентный метод для поиска седловой точки и других задач / Корпелевич, Г.М. // Экономика и математические методы. – 1976. – Vol. 12. – P. 747-756.

[6] А. В. Гасников и др. Введение в математическое моделирование транспортных потоков: Учебное пособие. М.: МЦНМО – 2013

[7] Попов, Л.Д. Модификация метода Эрроу-Гурвица поиска седловых точек / Попов, Л.Д. // Математические заметки. – 1980. – Vol. 28. – P. 845-848.

[8] Malitsky, Y. projected reflected gradient methods for monotone variational inequalities / Yura Malitsky // SIAM Journal on Optimization. – 2015. – Vol. 25. – P. 502-520.

[9] Van Rossum, G., & Drake, F. L. Python 3 Reference Manual. Scotts Valley, CA: CreateSpace. – 2009.

[10] Django Software Foundation. Django – 2019. – https://djangoproject.com