

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
Фізико-технічний інститут

КУРСОВА
з кредитного модуля «Методи обчислень»
на тему:
«ЧИСЛОВЕ РОЗВ'ЯЗАННЯ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ У
ЧАСТКОВИХ ПОХІДНИХ»
Варіант № 9

Виконала:
студентка 3 курсу ФТІ
Миронець Аліна Володимирівна
Перевірила:
Стьопочкіна Ірина Валеріївна
Оцінка:

Зміст

1.ОГЛЯД І АНАЛІЗ МЕТОДІВ РОЗВ’ЯЗАННЯ ДРЧП	3
1.1 Постановка задачі.....	3
1.2 Огляд та аналіз існуючих методів числового розв’язання ДРЧП.....	4
1.2.1 Метод скінченних елементів(МСЕ)	4
1.2.2Метод скінченних різниць(МСП).....	6
1.2.3. Аналіз задачі та вибір методу.....	8
РОЗДІЛ 2. ОПИС ЗАСТОСУВАННЯ ОБРАНИХ МЕТОДІВ ДО ПІЛСТАВЛЕНОЇ ЗАДАЧІ	10
2.1. Дослідження умов застосування обраного методу.....	10
2.2 Опис програмної реалізації	15
РОЗДІЛ 3. СПОСОБИ ПОКРАЩЕННЯ ОБРАНИХ МЕТОДІВ	19
3.1.Огляд методів підвищення точності та ефективності	19
3.1.1.Підвищення точності методів розв’язання ДРЧП	19
3.1.2. Підвищення ефективності методів розв’язання ДРЧП.....	20
3.2. Застосування методу підвищення точності та ефективності розв’язку до конкретного прикладу.....	21
Висновки	22
Список використаних джерел.....	24
Додаток А.....	25
Додаток Б	30

1.ОГЛЯД І АНАЛІЗ МЕТОДІВ РОЗВ'ЯЗАННЯ ДРЧП

1.1 Постановка задачі

Варіант 9

Процес дифузії речовини, яка розповсюджується із точкового джерела інтенсивності $Q=10\text{г}\backslash\text{с}$, розташованого в центральній точці області. Розміри області $100\times 100\text{м}$.

Рівняння еволюції речовини:

$$\frac{\partial u}{\partial t} = k \sum_{i=1}^2 \frac{\partial^2 u}{\partial z_i^2} + c_1 \frac{\partial u}{\partial z_1} + Q \quad (1.1)$$

$$\frac{\partial u}{\partial z_i}(\Gamma) = 0 \quad (1.2)$$

$$u(t = 0) = 0 \quad (1.3)$$

$$k = 50; c_1 = 1\text{м}\backslash\text{с}$$

Зробити короткостроковий прогноз (5хв.) поведінки.

$u(x, z_1, z_2)$ – концентрація речовини в точці (z_1, z_2) в момент часу t

Q – інтенсивність точкового джерела, з якого розповсюджується дифузія.

k – коефіцієнт дифузії, тобто коефіцієнт швидкості розповсюдження(переносу) речовини

c_1 —коефіцієнт переносу

Проаналізувавши рівняння (1.1), я його класифікувала за типами. Оскільки рівняння має залежність від часу, то очевидно що рівняння еволюційне, у протилежному випадку, тобто у стаціонарній задачі, така залежність відсутня. Можна зробити висновок, що рівняння не еліптичного типу. Крім того, помітимо, що похідна по часу другого порядку відсутня, а наявна

похідна другого порядку за просторовими координатами, тому очевидно що рівняння не може бути гіперболічного типу. Отже, може зробити, що дане рівняння відноситься до параболічного типу та водночас є еволюційною задачею.

1.2 Огляд та аналіз існуючих методів числового розв'язання ДРЧП.

Загальний метод розв'язання диференціальних рівнянь в часткових похідних полягає у застосування наближених моделей. Найбільш широке використання мають моделі, які побудовані на основі методу сіток. Суть методу сіток полягає у наступному: апроксимуємо шукану неперервну функцію сукупністю наближених значень, розрахованих в скінченному наборі точок області – вузлах сітки. Використання цього методу дозволяє звести диференціальну граничну задачу до системи лінійних або нелінійних рівнянь.

Відомо, що для розв'язання двомірних еволюційних рівнянь параболічного типу зручно використовуватися такі основні методи числового розв'язання, як: метод скінченних різниць та метод скінченних елементів, які у свою чергу являються методами, побудованими на основі сіток.

Різниця між цими методами, полягає в тому, що останній з них апроксимує саме значення функції, а відповідно метод скінченних різниць здійснює апроксимацію похідних.

Зазначу, що крім цих методів, є ще багато інших, зокрема до аналітичних методів розв'язання ДРЧП належать такі методи, як: метод послідовних наближень, метод розділення змінних, метод інтегральних перетворень, метод перетворення координат, метод перетворення залежної змінної та інші. В цій курсовій роботі вони не розглядаються, оскільки не являються методами числового розв'язання.

1.2.1 Метод скінченних елементів (МСЕ)

МСЕ використовує розбиття з урахуванням геометричних особливостей області. На елементи розбивають внутрішні області, при цьому алгоритм

розбиття будується так, щоб елементи задовольняли деяким обмеженням, наприклад сторони трикутників не дуже відрізнялись по довжині і т.д. Тому областю використання МСЕ є коло задач з областю визначення з довільною геометрією, наприклад - розрахунок на міцність деталей та вузлів будівельних конструкцій, авіаційних і космічних апаратів, тепловий розрахунок двигунів і т.д. Але МСЕ набуває досить складної реалізації у випадку еволюційної задачі. Цей метод краще пристосований до застосування у випадку стаціонарних задач.

Початкова задача приводиться до задачі мінімізації функціоналу визначеного вигляду, який ми отримали за вихідними рівняннями. Тоді, згідно з методом, функція повністю визначається сукупністю значень у вершинах елементів. В такому разі функціонал мінімізується по відношенню до цих вершин. І його мінімум і дасть розв'язок задачі. Як результат роботи методу ми отримаємо співвідношення, за допомогою яких і розв'яжемо вихідну задачу.

Варто зазначити низку переваг та недолік роботи цього алгоритму.

Зокрема, серед переваг визначимо:

- Забезпечення хорошої обумовленості отриманих СЛАР
- Можна легко розглядати змішані граничні умови
- За допомогою трикутників можна апроксимувати області складної форми
- Розміри елементів можуть бути змінними. Це дозволяє збільшуватися чи зменшувати елементи сітки

Серед недоліків, зверну увагу на наступні:

- Пряма залежність результатів обрахунків від розбиттів
- Складність реалізації алгоритмів
- Відсутність способу оцінки точності отриманих результатів
- Потреба великого обсягу пам'яті комп'ютерів

- Потреба високої швидкості розрахунку

Зазначу, що останні 2 десятиліття, ІТ-техніка, практично усунула.

1.2.2 Метод скінченних різниць (МСР)

МСР, на відміну від МСЕ, не намагається напряму отримати значення елементів, а працює з кінцевими різницями. Тобто, МСР замінює похідну задачу у вигляді диференціального рівняння на її аналог, який виражений у кінцевих різницях, побудованих для вузлів прямокутної сітки (оскільки метод застосовується переважно в областях із прямолінійними границями), які покривають область задачі. В зв'язку з цим МСР частіше використовується для аналізу задач з прямолінійними границями областей визначення функцій. Традиційно розв'язувані за допомогою МСР практичні задачі – це задачі дифузії та тепломасообміну рідин та газів.

МСР дозволяє крім кроку сітки (по аналогії, як у МСЕ) обрати ще шаблон та відповідну різницеву схему, що, очевидно, і впливає на хід рішення та кінцевий результат. Взагалі, вважається, що окремі шаблони та відповідні їм схеми – це окремі методи. Розглянемо, деякі з них, а саме: явну, неявну та явно-неявну (або ще «Кранка-Ніколсона») схеми.

Основна ідея явної схеми полягає у наступному, необхідно отримати формулу, яка явно виражає наступне значення точки за значенням поточної точки та її сусідніх.

Слід зауважити недоліки такої схеми:

- оскільки точка в наступному моменті часу обчислюється лише з використанням точок поточного часу, а зв'язку між ними нема, то і очевидно що система буде мати достатньо погану стійкість

- при необхідності у збільшенні точності апроксимації похідних, то при зміні розмірів сітки зросте не лише об'єм обчислень, але і похибка округлень.

Основна ідея неявної схеми: значення в даній точці в певний момент часу не залежить явно від значень на попередньому кроці, тобто як результат ми отримаємо систему рівнянь. Можна зрозуміти, що ця система буде набагато стійкішою, від попередньої, хоча і потребує значну кількість обчислень та складну матрицю з взаємозв'язками точок наступного моменту часу.

Перейдемо до останньої схеми, яку буде розглянуто в курсовій роботі, а саме – явно-неявна схема. Вона враховує недоліки попередніх схем. В даній схемі, вводиться також ще параметр λ , який дозволяє задавати степінь неявної схеми. Можемо помітити, що дві попередні схеми, є по суті частковими випадками, цієї схеми. Тобто, при $\lambda=0$ маємо явну схему, та відповідно при $\lambda=1$ маємо неявну.

Наглядно зобразимо різницеву схему сітки в явної-неявної схемі для одновимірної задачі.

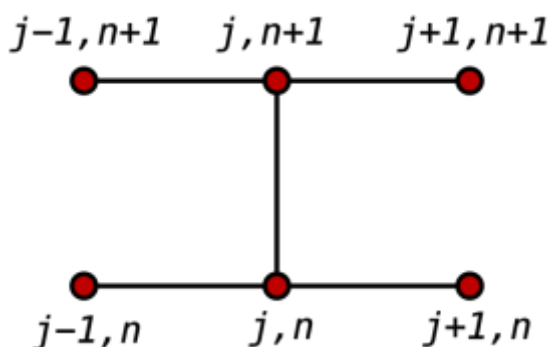


Рис1. різницева схема сітки в явної-неявної схемі для одновимірної задачі

В якості λ допускається будь яке число від 0 до 1, при $\lambda=0.5$ вводиться таке поняття, як «чиста» схема за методом Кранка-Ніколсона.

На мою думку, використання цієї схеми є найдоцільнішим, оскільки вона використовує всі дані сусідів поточного часу, і водночас забезпечує зв'язок

між елементами наступного моменту часу при розв'язанні системи лінійних алгебраїчних рівнянь.

Для загальної картини між 2 методами, а саме МСР та МСЕ, наведу також декілька недоліків і переваг і для методу скінченних різниць.

Серед переваг я виокремлю:

- доступна та проста реалізація алгоритмів
- можливість оцінити точність отриманих результатів
- ефективна СЛАР, яка складається з 5 діагоналей
- простота побудови сітки з фіксованим кроком, та можливість будувати сітку з перемінним кроком

Щодо недоліків:

- оскільки процес відбувається ітераційно, то неможливо одразу визначити розв'язок в будь який момент часу, за відсутності проміжних ітерацій
- сітка завжди прямокутна, що може викликати труднощі в певних випадках
- не забезпечується абсолютна стійкість

1.2.3. Аналіз задачі та вибір методу

Згідно з моїм варіантом, переді мною була поставлена двовимірна задача дифузії над квадратною областю 100×100 . Згідно з моїм аналізом, в задачі нема якихось особливих точок, з якими необхідно було б окремо працювати, також граничні умови не конфліктують між собою, тому особливої користі у використанні методу скінченних елементів для розв'язання цієї задачі.

Тому для розв'язання задачі я оберу метод скінченних різниць, а за основу методу – схему Кранка Ніколсона, коефіцієнт неявної схеми буде $\lambda = 0.25$

Відповідно на кожному шарі я обрахую значення за допомогою значень з попереднього шару, значення кожного вузла першого шару буде отримано з початкових та граничних умов. Після дискретизації лінійності спостерігатися не буде, а отже, зроблю СЛАР для кожного відповідно шару. Вибір методу вважається оптимальним.

РОЗДІЛ 2. ОПИС ЗАСТОСУВАННЯ ОБРАНИХ МЕТОДІВ ДО ПЛСТАВЛЕНОЇ ЗАДАЧІ

2.1. Дослідження умов застосування обраного методу.

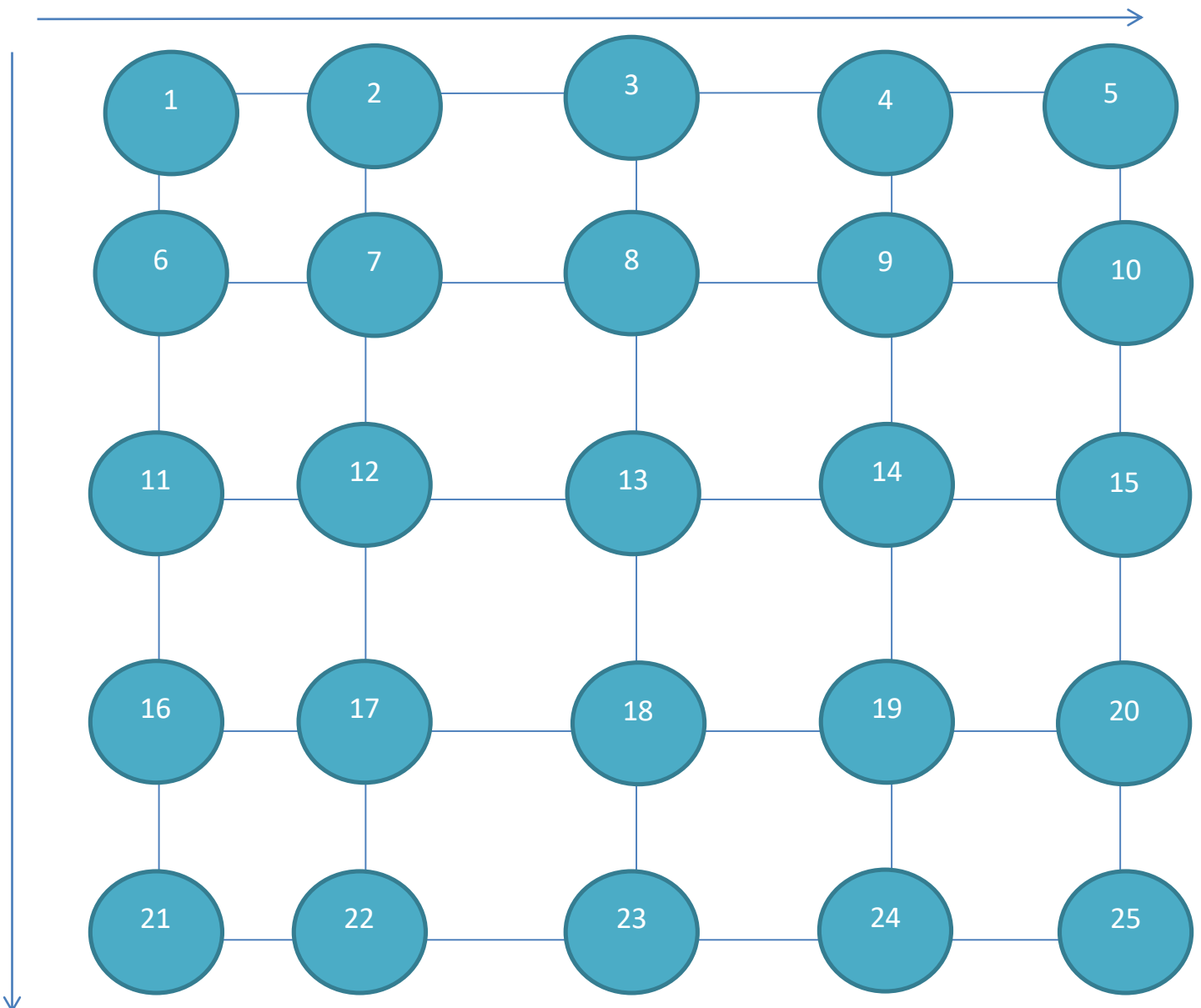
Як було зазначено, що за основну схему, для розв'язання задачі, я обрала схему Кранка-Ніколсона, відповідно з коефіцієнтом неявності схеми $\lambda = 0.25$

Дискретизація має наступний вигляд:

$$\frac{u_{ij}^{k+1} - u_{ij}^k}{\Delta t} = \lambda \left(k \cdot \frac{u_{i+1,j}^k - 2u_{ij}^k + u_{i-1,j}^k}{\Delta z_1^2} + k \cdot \frac{u_{i,j+1}^k - 2u_{ij}^k + u_{i,j-1}^k}{\Delta z_2^2} + c_1 \cdot \frac{u_{i+1,j}^k - u_{i,j}^k}{\Delta z_1} \right) + (1 - \lambda) \left(k \cdot \frac{u_{i+1,j}^{k+1} - 2u_{ij}^{k+1} + u_{i-1,j}^{k+1}}{\Delta z_1^2} + k \cdot \frac{u_{i,j+1}^{k+1} - 2u_{ij}^{k+1} + u_{i,j-1}^{k+1}}{\Delta z_2^2} + c_1 \cdot \frac{u_{i+1,j}^{k+1} - u_{ij}^{k+1}}{\Delta z_1} \right) + Q$$

(2.1)

Тоді зобразимо сітку 5×5



Підставимо формулу для дискретизації та отримаємо:

$$\frac{u_7^{k+1}-u_7^k}{\Delta t} = \lambda \left(k \frac{u_8^k-2u_7^k+u_6^k}{\Delta z_1^2} + k \frac{u_{12}^k-2u_7^k+u_2^k}{\Delta z_2^2} + c_1 \frac{u_8^k-u_7^k}{\Delta z_1} \right) + (1 - \lambda) \left(k \frac{u_8^{k+1}-2u_7^{k+1}+u_6^{k+1}}{\Delta z_1^2} + k \frac{u_{12}^{k+1}-2u_7^{k+1}+u_2^{k+1}}{\Delta z_2^2} + c_1 \frac{u_8^{k+1}-u_7^{k+1}}{\Delta z_1} \right) + Q$$

$$\frac{u_8^{k+1}-u_8^k}{\Delta t} = \lambda \left(k \frac{u_9^k-2u_8^k+u_7^k}{\Delta z_1^2} + k \frac{u_{13}^k-2u_8^k+u_3^k}{\Delta z_2^2} + c_1 \frac{u_9^k-u_8^k}{\Delta z_1} \right) + (1 - \lambda) \left(k \frac{u_9^{k+1}-2u_8^{k+1}+u_7^{k+1}}{\Delta z_1^2} + k \frac{u_{13}^{k+1}-2u_8^{k+1}+u_3^{k+1}}{\Delta z_2^2} + c_1 \frac{u_9^{k+1}-u_8^{k+1}}{\Delta z_1} \right) + Q$$

$$\frac{u_9^{k+1}-u_9^k}{\Delta t} = \lambda \left(k \frac{u_{10}^k-2u_9^k+u_8^k}{\Delta z_1^2} + k \frac{u_{14}^k-2u_9^k+u_4^k}{\Delta z_2^2} + c_1 \frac{u_{10}^k-u_9^k}{\Delta z_1} \right) + (1 - \lambda) \left(k \frac{u_{10}^{k+1}-2u_9^{k+1}+u_8^{k+1}}{\Delta z_1^2} + k \frac{u_{14}^{k+1}-2u_9^{k+1}+u_4^{k+1}}{\Delta z_2^2} + c_1 \frac{u_{10}^{k+1}-u_9^{k+1}}{\Delta z_1} \right) + Q$$

$$\frac{u_{12}^{k+1}-u_{12}^k}{\Delta t} = \lambda \left(k \frac{u_{13}^k-2u_{12}^k+u_{11}^k}{\Delta z_1^2} + k \frac{u_{17}^k-2u_{12}^k+u_7^k}{\Delta z_2^2} + c_1 \frac{u_{13}^k-u_{12}^k}{\Delta z_1} \right) + (1 - \lambda) \left(k \frac{u_{13}^{k+1}-2u_{12}^{k+1}+u_{11}^{k+1}}{\Delta z_1^2} + k \frac{u_{17}^{k+1}-2u_{12}^{k+1}+u_7^{k+1}}{\Delta z_2^2} + c_1 \frac{u_{13}^{k+1}-u_{12}^{k+1}}{\Delta z_1} \right) + Q$$

$$\frac{u_{13}^{k+1}-u_{13}^k}{\Delta t} = \lambda \left(k \frac{u_{14}^k-2u_{13}^k+u_{12}^k}{\Delta z_1^2} + k \frac{u_{18}^k-2u_{13}^k+u_8^k}{\Delta z_2^2} + c_1 \frac{u_{14}^k-u_{13}^k}{\Delta z_1} \right) + (1 - \lambda) \left(k \frac{u_{14}^{k+1}-2u_{13}^{k+1}+u_{12}^{k+1}}{\Delta z_1^2} + k \frac{u_{18}^{k+1}-2u_{13}^{k+1}+u_8^{k+1}}{\Delta z_2^2} + c_1 \frac{u_{14}^{k+1}-u_{13}^{k+1}}{\Delta z_1} \right) + Q$$

$$\frac{u_{14}^{k+1}-u_{14}^k}{\Delta t} = \lambda \left(k \frac{u_{15}^k-2u_{14}^k+u_{13}^k}{\Delta z_1^2} + k \frac{u_{19}^k-2u_{14}^k+u_9^k}{\Delta z_2^2} + c_1 \frac{u_{15}^k-u_{14}^k}{\Delta z_1} \right) + (1 - \lambda) \left(k \frac{u_{15}^{k+1}-2u_{14}^{k+1}+u_{13}^{k+1}}{\Delta z_1^2} + k \frac{u_{19}^{k+1}-2u_{14}^{k+1}+u_9^{k+1}}{\Delta z_2^2} + c_1 \frac{u_{15}^{k+1}-u_{14}^{k+1}}{\Delta z_1} \right) + Q$$

$$\frac{u_{17}^{k+1}-u_{17}^k}{\Delta t} = \lambda \left(k \frac{u_{18}^k-2u_{17}^k+u_{16}^k}{\Delta z_1^2} + k \frac{u_{22}^k-2u_{17}^k+u_{12}^k}{\Delta z_2^2} + c_1 \frac{u_{18}^k-u_{17}^k}{\Delta z_1} \right) + (1 - \lambda) \left(k \frac{u_{18}^{k+1}-2u_{17}^{k+1}+u_{16}^{k+1}}{\Delta z_1^2} + k \frac{u_{22}^{k+1}-2u_{17}^{k+1}+u_{12}^{k+1}}{\Delta z_2^2} + c_1 \frac{u_{18}^{k+1}-u_{17}^{k+1}}{\Delta z_1} \right) + Q$$

$$\frac{u_{18}^{k+1}-u_{18}^k}{\Delta t} = \lambda \left(k \frac{u_{19}^k-2u_{18}^k+u_{17}^k}{\Delta z_1^2} + k \frac{u_{23}^k-2u_{18}^k+u_{13}^k}{\Delta z_2^2} + c_1 \frac{u_{19}^k-u_{18}^k}{\Delta z_1} \right) + (1 - \lambda) \left(k \frac{u_{19}^{k+1}-2u_{18}^{k+1}+u_{17}^{k+1}}{\Delta z_1^2} + k \frac{u_{23}^{k+1}-2u_{18}^{k+1}+u_{13}^{k+1}}{\Delta z_2^2} + c_1 \frac{u_{19}^{k+1}-u_{18}^{k+1}}{\Delta z_1} \right) + Q$$

$$\frac{u_{19}^{k+1}-u_{19}^k}{\Delta t} = \lambda \left(k \frac{u_{20}^k-2u_{19}^k+u_{18}^k}{\Delta z_1^2} + k \frac{u_{24}^k-2u_{19}^k+u_{14}^k}{\Delta z_2^2} + c_1 \frac{u_{20}^k-u_{19}^k}{\Delta z_1} \right) + (1 - \lambda) \left(k \frac{u_{20}^{k+1}-2u_{19}^{k+1}+u_{18}^{k+1}}{\Delta z_1^2} + k \frac{u_{24}^{k+1}-2u_{19}^{k+1}+u_{14}^{k+1}}{\Delta z_2^2} + c_1 \frac{u_{20}^{k+1}-u_{19}^{k+1}}{\Delta z_1} \right) + Q$$

Застосовуючи дискретизацію до граничних точок, отримаю відповідно наступні рівняння:

Нижня границя:

$$u_{ij} - u_{i-1j} = 0$$

Верхня границя:

$$u_{ij} - u_{i+1j} = 0$$

Ліва границя:

$$u_{ij} - u_{ij+1} = 0$$

Права границя:

$$u_{ij} - u_{ij-1} = 0$$

Якщо я перенесу всі u^{k+1} вліво, а решту, тобто те що мені відоме залишу справа, то очевидно я отримаю рівняння, яке буде лінійним, з 5 невідомим відповідно для п'яти вузлів на $k+1$ кроці зліва, те що мені невідоме, і справа – значення u на k -ому кроці, це буде константа, так як вважається що це нам відомо. Як результат отримаю систему $Au=b$, де A -головна матриця системи, u -вектор невідомих значень, b -вектор вільних членів. Розв'язуючи таку систему на кожному кроці, будемо мати значення для кожного вузла шару, том зможемо вирішити таку систему на наступному кроці.

Отже, будемо мати наступну задачу: сукупність систем рівнянь відносно невідомих $u_{i+1j}, u_{i-1j}, u_{ij}, u_{ij+1}, u_{ij-1}$

Відповідно далі, ми виразимо, в загальній формулі коефіцієнти для U на $k+1$ кроці, їх я виведу з формули (2.1). Отриманий результат запишу:

$$u_{ij}^{k+1} = 1 - \Delta t \cdot (1 - \lambda) \cdot \left(-2 \cdot \frac{k}{\Delta x^2} - 2 \cdot \frac{k}{\Delta y^2} - \frac{c}{\Delta x}\right)$$

$$u_{i+1j}^{k+1} = -\Delta t \cdot (1 - \lambda) \cdot \left(\frac{k}{\Delta x^2}\right)$$

$$u_{i-1j}^{k+1} = -\Delta t \cdot (1 - \lambda) \cdot \left(\frac{k}{\Delta x^2} + \frac{c}{\Delta x}\right)$$

$$u_{ij-1}^{k+1} = -\Delta t \cdot (1 - \lambda) \cdot \left(\frac{k}{\Delta y^2}\right)$$

$$u_{ij+1}^{k+1} = -\Delta t \cdot (1 - \lambda) \cdot \left(\frac{k}{\Delta y^2}\right)$$

Отже, отримали необхідні нам коефіцієнти. Далі, заповнимо матрицю, вже відповідними значення.

Очевидно, що структура матриці на кожному кроці буде наступною:

На головній діагоналі будуть коефіцієнти біля u_{ij} , відповідно на координаті $ij \pm 1$ будемо мати коефіцієнти u_{ij+1}^{k+1} , u_{ij-1}^{k+1} , на координаті $ij \pm n$ будемо мати відповідно u_{i+1j}^{k+1} , u_{i-1j}^{k+1}

Тоді зобразимо схематично СЛАР для нашого випадку. Нагадаю, що працюю з схемою розміру 5×5 .

$$A \cdot \begin{bmatrix} u_1^{k+1} \\ u_2^{k+1} \\ u_3^{k+1} \\ u_4^{k+1} \\ u_5^{k+1} \\ u_6^{k+1} \\ u_7^{k+1} \\ u_8^{k+1} \\ u_9^{k+1} \\ u_{10}^{k+1} \\ u_{11}^{k+1} \\ u_{12}^{k+1} \\ u_{13}^{k+1} \\ u_{14}^{k+1} \\ u_{15}^{k+1} \\ u_{16}^{k+1} \\ u_{17}^{k+1} \\ u_{18}^{k+1} \\ u_{19}^{k+1} \\ u_{20}^{k+1} \\ u_{21}^{k+1} \\ u_{22}^{k+1} \\ u_{23}^{k+1} \\ u_{24}^{k+1} \\ u_{25}^{k+1} \end{bmatrix} = b \quad b = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ b_7 \\ b_8 \\ b_9 \\ 0 \\ 0 \\ b_{12} \\ b_{13} \\ b_{14} \\ 0 \\ 0 \\ b_{17} \\ b_{18} \\ b_{19} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Відповідно, що вектор b визначається виключно з даних у поточний момент часу, тому я отримала наступні формули, для визначення координат цього вектора.

Наприклад,

$$b_{13} = \lambda \cdot \Delta t \left(k \frac{u_{14}^k - 2u_{13}^k + u_{12}^k}{\Delta z_1^2} + k \frac{u_{18}^k - 2u_{13}^k + u_8^k}{\Delta z_2^2} + c_1 \frac{u_{14}^k - u_{13}^k}{\Delta z_1} \right) + Q + u_{13}^k$$

Загальна формула ж, наступна:

$$b_i = \lambda \cdot \Delta t \left(k \frac{u_{i+1}^k - 2u_i^k + u_{i-1}^k}{\Delta z_1^2} + k \frac{u_{i+n}^k - 2u_i^k + u_{i-n}^k}{\Delta z_2^2} + c_1 \frac{u_{i+1}^k - u_i^k}{\Delta z_1} \right) + Q + u_i^k$$

Таким чином, отримаємо СЛАР великої розмірності. Можна легко помітити, що матриця коефіцієнтів є розрідженою, та має стрічкову структуру, саме тому для її розв’язання доречно було б скористатися прямими методами розв’язку СЛАР, наприклад метод квадратного кореня або метод Гауса, оскільки вони мають основну перевагу — визначений час роботи. Проте, слід зауважити, що обернення матриця і розв’язок СЛАР можна здійснити алгоритмами асимптотично більш швидкими за порядком, ніж метод Гауса. Зокрема, один з таких методів для розв’язання системи лінійних алгебраїчних рівнянь є у математичному пакеті python, а за основу він має LU розкладання матриці коефіцієнтів A .

2.2 Опис програмної реалізації

Отже, для програмної реалізації я обрала мову програмування Python з інтерпретатором версії 3.6, з використанням таких бібліотек, як numpy, scipy, matplotlib. У програмі реалізовано:

- вибір кроку по просторових координатах
- задання розмірів сітки
- створення та заповнення матриці A та стовпчика вільних членів b
- можливість змінювати початкові умови, параметри
- вирішення СЛАР вбудованими методами python scipy
- візуалізація

Спершу я задала всі константи, а також задала розмір сітки, з якою буду працювати, обрала кроки. Далі було необхідно обрахувати значення Δt , відповідно з заданими кроками ($\Delta z_1, \Delta z_2$).

Крім цього, обрала значення параметра $\lambda = 0.25$, який являється коефіцієнтом неясності схеми.

Далі, мною було створено матрицю U на нульовому кроці, тобто так званий масив вузлів. Заповнювався він згідно з початкових умов, в моєму випадку це був масив нулів. Далі для роботи були реалізовані деякі функції.

Основні функції, використані у програмі:

1. `get_vector_b()` – значення у вузлі на попередньому кроці
 2. `get_matrix_A()` – для побудови матриці A та її заповнення (матриця коефіцієнтів для наступного часового шару). Зазначу, що цю матрицю достатньо задати лише один раз, так як вона протягом ітерацій не зазнає жодних змін.
 3. `get_vector_B1()` – повертає значення для вектора вільних членів
 4. `get_matrix_Uk()` – повертає значення матриці U на $k+1$ кроці
- Розв’язуємо отриману СЛАР за допомогою вбудованої функції. Далі заповнюємо відповідно нашу матрицю U необхідними значеннями, які вже отримали.
5. `plot_scatter()` – графік у вигляді точок
 6. `plot_surface()` – графік у вигляді ліній рівня
 7. `plot_contour()` – графік у просторі

Для розв’язання СЛАР я використала вбудовану функцію `linalg.solve()` з бібліотеки `numpy`. Відомо, що за основу ця функція має LU факторизацію матриці A , використавши пряму та обернену підстановку отримує значення кореня рівняння виду $Ax=b$.

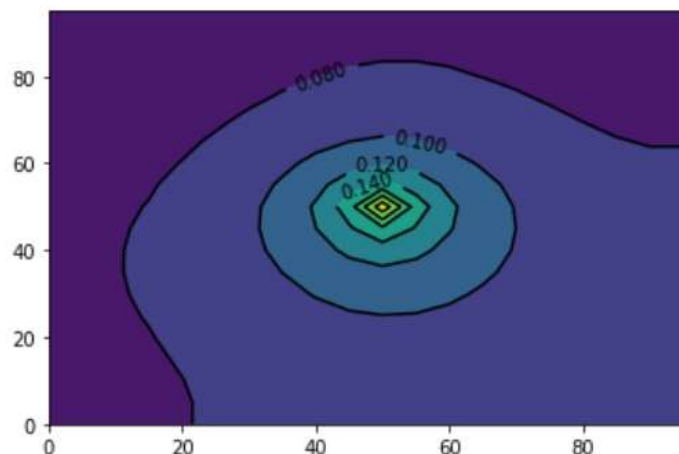
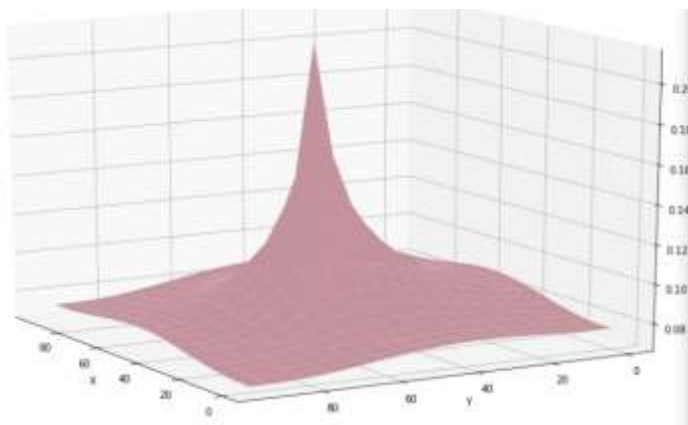
Зазначу, що LU факторизація матриці—це представлення її у вигляді добутку двох матриць. В нашому випадку маємо що LU факторизація матриці A має такий вигляд: $A=LU$, L - нижня трикутна матриця, U - відповідно верхня трикутна матриця. Тоді маємо таку систему: $Ax=b$, $A=LU$, прирівнявши ці рівняння, очевидно можна отримати наступну рівність: $LUx=b$.

Нижче наведені графіки візуалізації значень системи після 1, 2, 3,4, 5 хвилин протікання процесу

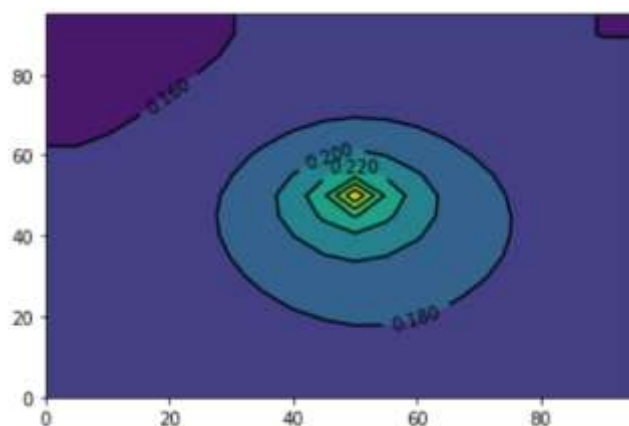
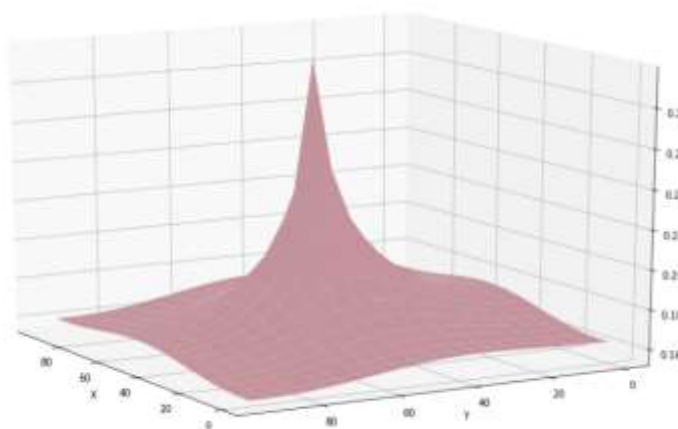
Нехай візьмемо такі дані:

Сітка 20×20 , відповідно $\Delta z_1 = 5 = \Delta z_2$, $\Delta t = 20c$

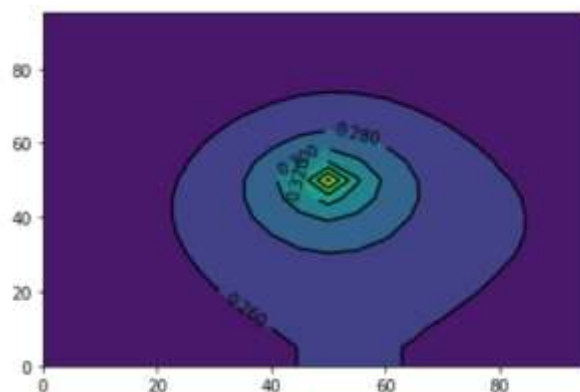
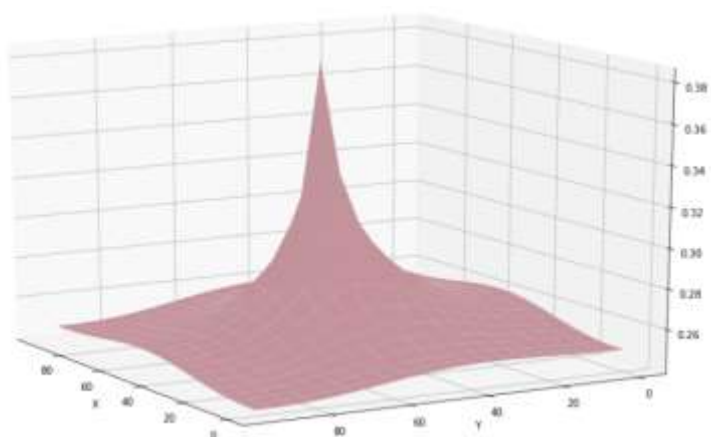
Після 1 хвилини протікання процесу



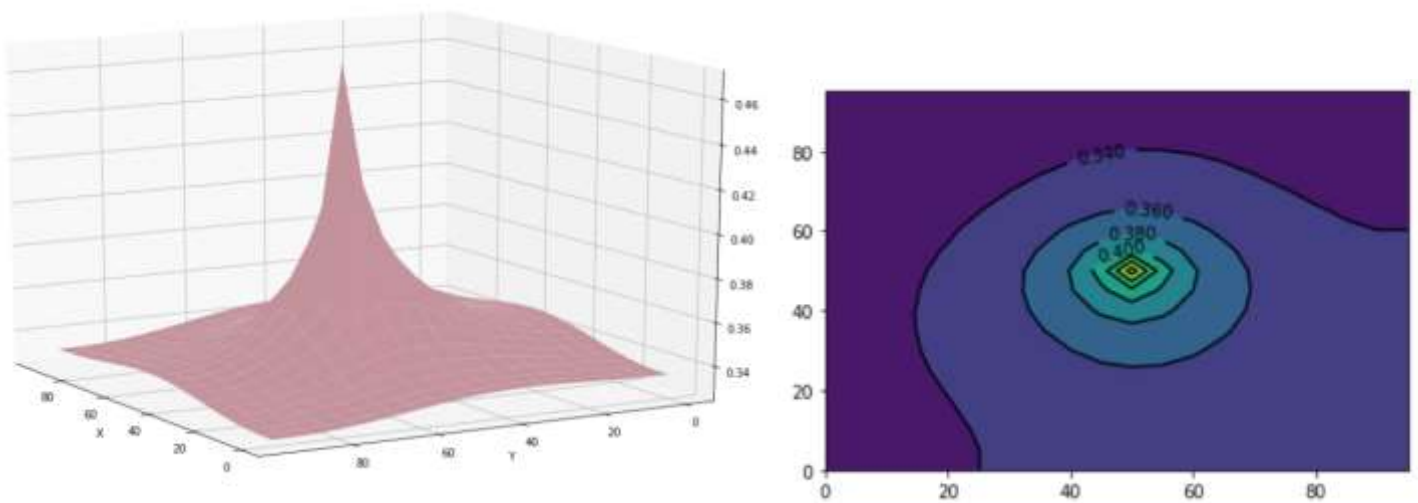
Після 2 хвилин протікання процесу



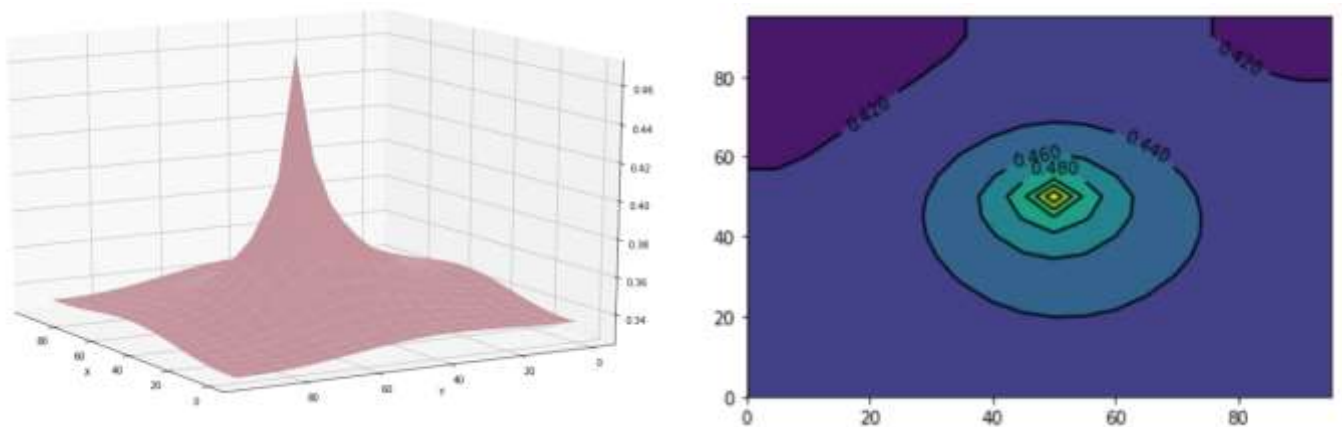
Після 3 хвилин протікання процесу



Після 4 хвилин протікання процесу



Після 5 хвилин протікання процесу



З графіків (а саме на контурах, тобто виглядах зверху) видно, що при кожній наступній ітерації концентрація речовини в джерелі та в сусідніх точках збільшується. Крім того, це видно також і на значеннях, які явно збільшуються.

РОЗДІЛ 3. СПОСОБИ ПОКРАЩЕННЯ ОБРАНИХ МЕТОДІВ

3.1.Огляд методів підвищення точності та ефективності

3.1.1.Підвищення точності методів розв’язання ДРЧП

В основі числових методів рішення ДРЧП лежить принцип заміни неперервної області визначення невідомої функції дискретною множиною точок—тобто сіткою. Для мінімізації похибки числового методу в місцях значних змін функції сітку згущають. Тому при розв’язанні багатовимірних нестационарних задач для підвищення точності найчастіше використовують варіаційний принцип побудови адаптивних сіток, а також його модифікації.

Використання в розрахунках адаптивних сіток пояснюється бажанням забезпечити необхідну точність, мінімізуючи число вузлів розрахункової сітки. Суть методу полягає в тому, що ці сітки підлаштовуються під поточний розв’язок та похідні за рахунок зменшення кроку, лише там де це необхідно, відповідно з метою отримання більшої точності. Хоча для еволюційних задач цей метод є не дуже ефективним.

Крім адаптивних сіток, можна ще застосовувати нерівномірні сітки. Суть побудови такої сітки полягає в тому, що ми розташовуємо її більш щільно навколо якоїсь певної області, яка несе за собою більший інтерес – тобто, наприклад, там знаходиться джерело, або ж відбувається процес.

Також, для рівнянь теплообміну доречно застосовувати метод скінченних об’ємів. В свою чергу це такий ж чисельний метод інтегрування систем диференціальних рівнянь з частинними похідними. Аналогічно до МСР та МСЕ використовується сітка. Скінченний об’єм – невеликий об’єм навколо кожної вузлової точки сітки. У цьому методі, інтеграли об’єму, які містять вирази з дивергенцією, перетворюються у поверхневі інтеграли за допомогою формули Остроградського. Потім ж ці вирази оцінюються як поверхневі потоки кожного скінченного об’єму. Так, як потік, який входить у

даний об'єм, є ідентичними до потоку, який виходить із суміжного об'єму, то ці методи є консервативними, крім того МСО легко формулюються для неструктурованої сітки, що у свою чергу є також вагомою перевагою.

3.1.2. Підвищення ефективності методів розв'язання ДРЧП

Крім того, варто згадати, що так, як при розв'язанні ДРЧП застосовувався скінченно-різницеий підхід, то отримуємо СЛАР достатньо великої розмірності. Одержана матриця коефіцієнтів, в нашому випадку матриця A , є розрідженою матрицею, більше того вона має стрічкову структуру. В такому разі, підвищити ефективність розв'язку СЛАР можна за допомогою перепорядкування елементів матриці коефіцієнтів лінійних рівнянь, до такої де елементи розташовані більш щільно. Схематично це можна зобразити наступним чином:

$$\begin{bmatrix} B_{11} & B_{12} & 0 & \cdots & \cdots & 0 \\ B_{21} & B_{22} & B_{23} & \ddots & \ddots & \vdots \\ 0 & B_{32} & B_{33} & B_{34} & \ddots & \vdots \\ \vdots & \ddots & B_{43} & B_{44} & B_{45} & 0 \\ \vdots & \ddots & \ddots & B_{54} & B_{55} & B_{56} \\ 0 & \cdots & \cdots & 0 & B_{65} & B_{66} \end{bmatrix} \longrightarrow \begin{bmatrix} 0 & B_{11} & B_{12} \\ B_{21} & B_{22} & B_{23} \\ B_{32} & B_{33} & B_{34} \\ B_{43} & B_{44} & B_{45} \\ B_{54} & B_{55} & B_{56} \\ B_{65} & B_{66} & 0 \end{bmatrix}$$

Рис 2. Схематичне пере впорядкування елементів матриці

В такому разі, необхідно скористатися стрічковим методом пере впорядкування. Він допоможе зекономити машинні ресурси для матриць великої розмірності.

Якщо ж, ми вирішили розв'язувати систему лінійних алгебраїчних рівнянь методом LU-декомпозиції, то в такому разі, задля підвищення ефективності, буде доречно скористатися алгоритмом COlumn Approximate Minimum Degree (COLAMD)---переставляє рядки та стовпчики матриці з метою підвищення розрідженості її LU-розкладання.

3.2. Застосування методу підвищення точності та ефективності розв'язку до конкретного прикладу

Отже, в мене є матриця A —розріджена матриця, для підвищення ефективності розв'язку СЛАР її необхідно раціонально зберегти в пам'яті, для економії машинних ресурсів. Існує декілька методів перевпорядкування елементів, зокрема серед них: метод квадратного кореня, алгоритм Катхілла-Маккі, та COLAMD. Важлива відмінність: перші два методи не працюють з несиметричними матрицями.

Але скористаюся, достатньо зрозумілим та відносно легким методом, а саме методом квадратного кореня. Спершу, необхідно, за допомогою лінійних перетворень привести матрицю до її симетричного вигляду. Далі слід перевірити матрицю чи є вона додатньовизначеною.

Тоді, якщо A - симетрична, додатньовизначена матриця, з півшириною p , то півширина множника методу квадратного кореня не більше p . В такому разі, елементи матриці L будуть мати такі значення:

$$l_{ij} = \sqrt{a_{ij} - \sum_{k=\max[1, j-p]}^{j-1} l_{jk}^2}, j = 1, \dots, n$$

$$l_{ij} = \frac{1}{l_{ij}} \left[a_{ij} - \sum_{k=\max[1, i-p]}^{j-1} l_{ik} l_{jk} \right], i=j+1, \dots, \min(n, p+j), j=1, \dots, n$$

В іншому раз -- $l_{ij} = 0$

Тоді для розв'язку системи необхідно буде застосувати певні перетворення.

Якщо, ж матрицю не вдасться звести до симетричного вигляду, то доведеться скористатися методом COLAMD, він використовується такою функцією з python, як: `scipy.sparse.linalg.splu`, яка будує LU-декомпозицію, я одночасно здійснює перестановик для максимально можливого збереження розрідженості матриці.

Висновки

Під час виконання курсової роботи було проведено дослідження процесу дифузії речовини, яка розповсюджується із точкового джерела, що розташоване в центральній точці області, та має певну інтенсивність.

Розмір області, в якій відбувається фізичний процес 100×100 м.

Відомо, що диференціальні рівняння в часткових похідних широко застосовується у моделювання різних природних явищ.

Зазначу, що рівняння цієї задачі описують та моделюють багато базових процесів, зокрема це процес є дифузії речовини (відповідно з цим процесом я працювала) або ж теплообмін у речовині, та ще деякі інші. Саме тому, такі задачі є дійсно актуальні.

Для вирішення ДРЧП я обрала метод скінченних різниць, та використала схему Кранка-Ніколсона.

Відповідно, розв'язок СЛАР я отримала за допомогою вбудованої функції з python-бібліотеки `scipy.linalg.solve()`. В основі цієї функції лежить LU розкладання матриці A . Крім цього, з допомогою засобів візуалізації python, а саме бібліотеки `matplotlib`, мені вдалося змодельовати цей процес на графіках у вигляді ліній рівня та у просторі.

Протягом роботи, виникали деякі труднощі, основні з них:

- раціонально підійти до вибору розв'язання СЛАР для матриць великої розмірності
- підібрати вхідні значення, так щоб вони приблизно відповідали реальному процесу

Після проробленої роботи, я проаналізувала отримані результати, та з них отримала висновки, зокрема деякі з них:

- при використанні методу скінченних різниць важливо знати та дотримуватися умов стійкості, так як, при їх порушенні, навіть незначному, в системі можуть виникнути деякі збурення
- розв'язок СЛАР, є дуже громіздкою задачею, оскільки матриця достатньо велика, тому при можливості не завадить скористатися оптимізацією, деякі методи були зазначені у 3 розділі
- зменшення кроку Δt спричиняє корекцію результатів практично на всіх ітераціях

З фізичної точки зору, отримані результати це є концентрація речовини в конкретній точці та її околі. Відповідно на графіках можна побачити значення функції u в певний момент часу в кожному вузлі, яка показує концентрацію речовини в цій точці, а також зміну стану системи.

Список використаних джерел

1. Жданова І. В. Числові методи. Методичні вказівки до виконання курсової роботи. 68с.
2. [Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы. — М.:](#)
3. [Волков Е.А. Численные методы. — СПб.: Лань, 2004. — 256 с.](#)
4. [Вержбицкий В.М. Основы численных методов.-М.:Высшая школа, 2008. — 848 с](#)
5. [Самарский А.А., Гулин А.В. Численные методы. — М.: Наука, 1989. — 432 с.](#)
6. [Самарский А.А., Николаев Е. С. Методы решения сеточных уравнений.- М.: Наука, 1978. — 430 с.](#)
7. [Ортега Дж. Пул У. Введение в численные методы решения дифференциальных уравнений. — М.:Наука, 1986. — 288 с.](#)

Додаток А

Програмна реалізація

```
import numpy as np

import scipy

from scipy import linalg

from mpl_toolkits.mplot3d import Axes3D

import matplotlib.pyplot as plt


dimension_n = 20

const_k = 50

const_c = 1

dz1 = 100 / dimension_n

dz2 = 100 / dimension_n

dt = 40

lamda = 0.25


matrix_U = np.zeros((21, dimension_n, dimension_n))

matrix_Q = np.zeros((dimension_n, dimension_n))

matrix_Q[int(dimension_n / 2)][int(dimension_n / 2)] = 10


def get_vector_b(U1, k, i, j):

    vector_b = dt * lamda * (const_k * ((U1[k][i + 1][j] - 2 * U1[k][i][j] + U1[k][i - 1][j]) / (dz1
2) + (

        U1[k][i][j + 1] - 2 * U1[k][i][j] + U1[k][i][j - 1]) / (dz2 ** 2)) + const_c * (

        U1[k][i + 1][j] - U1[k][i][j]) / dz1) + Q[i][j] + U1[k][i][j])

    return vector_b
```

```

def get_matrix_A():

    matr_A = np.zeros((dimension_n**2, dimension_n**2))

    len_A = len(matr_A)

    for i in range(len_A):

        for j in range(len_A):

            if i <= dimension_n-1:

                if i == j:

                    matr_A[i][j] = 1

                    matr_A[i][j+dimension_n] = -1

            elif i >= len_A - dimension_n:

                if i == j:

                    matr_A[i][j] = 1

                    matr_A[i][j-dimension_n] = -1

            elif j % dimension_n == 0:

                if i == j:

                    matr_A[i][j] = 1

                    matr_A[i][j+1] = -1

            elif (j+1) % dimension_n == 0:

                if i == j:

                    matr_A[i][j] = 1

                    matr_A[i][j-1] = -1

            else:

                if i == j:

                    matr_A[i][j] = 1 - dt*(1-lamda)*(-2*const_k/(dz1**2) - 2*const_k/(dz2**2) -
const_c/dz1)

                    matr_A[i][j-dimension_n] = -dt*(1-lamda)*const_k/(dz1**2)

                    matr_A[i][j+dimension_n] = -dt*(1-lamda)*(const_k/(dz1**2) + const_c/dz1)

                    matr_A[i][j-1] = -dt*(1-lamda)*const_k/(dz2**2)

                    matr_A[i][j+1] = -dt*(1-lamda)*const_k/(dz2**2)

```

```
return matr_A
```

```
def get_vector_B1(k):
```

```
    vect_B = np.zeros(dimension_n**2)
```

```
    U1 = np.zeros((dimension_n, dimension_n))
```

```
    for i in range(1, dimension_n-1):
```

```
        for j in range(1, dimension_n - 1):
```

```
            U1[i][j] = get_vector_b(U, k, i, j)
```

```
    U1 = np.reshape(U1, (dimension_n**2, 1))
```

```
    for i in range(len(vect_B)):
```

```
        if i <= dimension_n-1 or i >= len(vect_B) - dimension_n or i % dimension_n == 0 or (i+1)
        % dimension_n == 0:
```

```
            vect_B[i] = 0
```

```
        else:
```

```
            vect_B[i] = U1[i]
```

```
    return vect_B
```

```
def get_matrix_Uk(k):
```

```
    U1 = np.zeros((dimension_n, dimension_n))
```

```
    res = linalg.solve(get_matr_A(), get_vect_B1(k))
```

```
    res = np.reshape(res,(dimension_n, dimension_n))
```

```
    for i in range(dimension_n):
```

```
        for j in range(dimension_n):
```

```
            U1[i][j] = res[i][j]
```

```
    return U1
```

```
for k in range(1, 21):
```

```
    matrix_U[k] = get_matrix_Uk(k-1)
```

```

def plot_scatter():

    fig = plt.figure(figsize=[15, 10])

    ax = fig.gca(projection='3d')

    ax.view_init(15, 150)

    X = np.arange(0, dimension_n * dz1, dz1)

    Y = np.arange(0, dimension_n * dz2, dz2)

    X, Y = np.meshgrid(X, Y)

    ax.scatter(X, Y, U[2], color='black')

    ax.set_xlabel('X')

    ax.set_ylabel('Y')

    plt.show()

plot_scatter()

```

```

def plot_surface():

    fig = plt.figure(figsize=[15, 10])

    ax = fig.gca(projection='3d')

    ax.view_init(15, 150)

    X = np.arange(0, dimension_n*dz1, dz1)

    Y = np.arange(0, dimension_n*dz2, dz2)

    X, Y = np.meshgrid(X, Y)

    ax.plot_surface(X, Y, U[20], color = 'pink')

    ax.set_xlabel('X')

    ax.set_ylabel('Y')

    plt.show()

plot_surface()

```

```
def plot_contour():  
    X = np.arange(0, dimension_n * dz1, dz1)  
    Y = np.arange(0, dimension_n * dz2, dz2)  
    X, Y = np.meshgrid(X, Y)  
    C = plt.contour(X, Y, U[5], 8, colors='black')  
    plt.contourf(X, Y, U[5], 8)  
    plt.clabel(C, inline=1, fontsize=10)  
    plt.show()  
plot_contour()
```

Додаток Б

Коефіцієнти отриманої матриці А при dimension_n=5

[[1. , 0. , 0. , 0. , 0. , -1. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.],

[0. , 1. , 0. , 0. , 0. , 0. , -1. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.],

[0. , 0. , 1. , 0. , 0. , 0. , 0. , -1. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.],

[0. , 0. , 0. , 1. , 0. , 0. , 0. , 0. , -1. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.],

[0. , 0. , 0. , 0. , 1. , 0. , 0. , 0. , 0. , -1. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.],

[0. , 0. , 0. , 0. , 0. , 1. , -1. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.],

[0. , -1.40625, 0. , 0. , 0. , -1.40625, 7.1875, -1.40625, 0. , 0. , 0. , -1.96875, 0.
, 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.],

[0. , 0. , -1.40625, 0. , 0. , 0. , -1.40625, 7.1875, -1.40625, 0. , 0. , 0. , -
1.96875, 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.],

[0. , 0. , 0. , -1.40625, 0. , 0. , 0. , -1.40625, 7.1875, -1.40625, 0. , 0. , 0. , -
1.96875, 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.],

[0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , -1. , 1. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.],

[0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 1. , -1. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.],

[0. , 0. , 0. , 0. , 0. , 0. , -1.40625, 0. , 0. , 0. , -1.40625, 7.1875, -
1.40625, 0. , 0. , 0. , -1.96875, 0. , 0. , 0. , 0. , 0. , 0. , 0.],

[0. , 0. , 0. , 0. , 0. , 0. , 0. , -1.40625, 0. , 0. , 0. , -1.40625, 7.1875, -
1.40625, 0. , 0. , 0. , -1.96875, 0. , 0. , 0. , 0. , 0. , 0. , 0.],

[0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , -1.40625, 0. , 0. , 0. , -1.40625,
7.1875, -1.40625, 0. , 0. , 0. , -1.96875, 0. , 0. , 0. , 0. , 0. , 0.],

[0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.],

Значення u у вузлах сітки на 1 хвилині для сітки 20 на 20 з кроком $\Delta t=20$ с:

[[0.07664444, 0.07664444, 0.07714917, 0.07812241, 0.0794911 ,
0.08114493, 0.08293839, 0.08469789, 0.0862376 , 0.08738549,
0.08801699, 0.08808602, 0.08763811, 0.08679658, 0.08573068,
0.08462111, 0.08363284, 0.08289812, 0.08250779, 0.08250779],
[0.07664444, 0.07664444, 0.07714917, 0.07812241, 0.0794911 ,
0.08114493, 0.08293839, 0.08469789, 0.0862376 , 0.08738549,

0.08801699, 0.08808602, 0.08763811, 0.08679658, 0.08573068,
0.08462111, 0.08363284, 0.08289812, 0.08250779, 0.08250779],
[0.0768444 , 0.0768444 , 0.07738154, 0.07842029, 0.07988806,
0.081673 , 0.08362323, 0.08555105, 0.08724703, 0.08850857,
0.0891828 , 0.08921165, 0.0886532 , 0.08765966, 0.08643157,
0.08517414, 0.08406755, 0.08325168, 0.08282037, 0.08282037],
[0.07719574, 0.07719574, 0.07779414, 0.07895754, 0.08061597,
0.08265709, 0.08491972, 0.08719068, 0.08921314, 0.09071765,
0.0914845 , 0.09142509, 0.09062903, 0.08931646, 0.08775613,
0.0862023 , 0.08486271, 0.08388912, 0.08337878, 0.08337878],
[0.07762727, 0.07762727, 0.07831125, 0.07965078, 0.08158374,
0.08400362, 0.08674385, 0.0895608 , 0.09212623, 0.09405283,
0.0949857 , 0.09476534, 0.09355355, 0.09170716, 0.08961453,
0.08760359, 0.08591556, 0.08471131, 0.08408695, 0.08408695],
[0.07805166, 0.07805166, 0.07883888, 0.08039437, 0.08267322,
0.08558887, 0.08898582, 0.09259979, 0.09601415, 0.09865043,
0.09988126, 0.09936745, 0.09745194, 0.0947656 , 0.09189005,
0.08924428, 0.08709376, 0.0855932 , 0.0848252 , 0.0848252],
[0.0783732 , 0.0783732 , 0.07927122, 0.08106353, 0.08373569,
0.08724471, 0.09148215, 0.09620495, 0.10092436, 0.10478337,
0.10659729, 0.10550306, 0.10236885, 0.09838447, 0.09441209,

0.09094563, 0.08823324, 0.08638825, 0.08545765, 0.08545765],
[0.07849801, 0.07849801, 0.07950003, 0.08152094, 0.08459089,
0.08874124, 0.0939705 , 0.10015692, 0.10685722, 0.11292069,
0.1160577 , 0.11364009, 0.10830225, 0.10234028, 0.0969118 ,
0.09246779, 0.08913886, 0.08693702, 0.08584361, 0.08584361],
[0.0783471 , 0.0783471 , 0.07942845, 0.08163053, 0.0850361 ,
0.08977688, 0.09603049, 0.10397106, 0.11354849, 0.12373216,
0.13048227, 0.12444753, 0.11498618, 0.10614537, 0.09896386,
0.09350133, 0.08959546, 0.08708389, 0.08585538, 0.08585538],
[0.07787146, 0.07787146, 0.07898947, 0.08128239, 0.08487723,
0.09000243, 0.09704754, 0.10666235, 0.11984776, 0.13752503,
0.15612163, 0.1382323 , 0.12126946, 0.10881326, 0.09995109,
0.09369253, 0.08940115, 0.08670544, 0.08540241, 0.08540241],
[0.07706564, 0.07706564, 0.07816578, 0.08042763, 0.0839919 ,
0.08912414, 0.09632283, 0.10659874, 0.1223184 , 0.150214 ,
0.21378705, 0.15090928, 0.12371576, 0.10871226, 0.09917494,
0.09274742, 0.08843215, 0.08574879, 0.08445785, 0.08445785],
[0.07597468, 0.07597468, 0.07700316, 0.07911081, 0.08241115,
0.08710911, 0.09355557, 0.1023378 , 0.1143624 , 0.13046411,
0.14739324, 0.13114418, 0.11572849, 0.10440219, 0.09633776,
0.09063739, 0.08672518, 0.08426574, 0.08307631, 0.08307631],

[0.07468879, 0.07468879, 0.0756065 , 0.07747244, 0.08035111,
0.08434594, 0.08959721, 0.09624192, 0.104232 , 0.11271013,
0.11833377, 0.11337287, 0.10556217, 0.09824915, 0.09229666,
0.0877593 , 0.08450832, 0.08241433, 0.08138893, 0.08138893],
[0.07332587, 0.07332587, 0.07411542, 0.07570412, 0.07810869,
0.08134434, 0.08539995, 0.09017329, 0.09532098, 0.09996972,
0.10239323, 0.10061435, 0.09661344, 0.09212009, 0.08801106,
0.08463361, 0.08209442, 0.08041005, 0.07957204, 0.07957204],
[0.07200923, 0.07200923, 0.07267364, 0.07399569, 0.07595728,
0.07851746, 0.0815881 , 0.08498795, 0.08836837, 0.09113141,
0.09245919, 0.09175858, 0.08962439, 0.08687609, 0.08411304,
0.081685 , 0.07977392, 0.07846808, 0.07780752, 0.07780752],
[0.07084852, 0.07084852, 0.07140525, 0.07250141, 0.07409831,
0.07612682, 0.07847148, 0.08094743, 0.08327549, 0.08507886,
0.08595575, 0.08569057, 0.0844992 , 0.08278344, 0.08091965,
0.07918565, 0.07776378, 0.07676516, 0.07625199, 0.07625199],
[0.0699277 , 0.0699277 , 0.07040208, 0.07132781, 0.0726561 ,
0.07430702, 0.07616178, 0.07805511, 0.07977379, 0.08107474,
0.08174046, 0.0816741 , 0.08097165, 0.0798494 , 0.07854843,
0.0772787 , 0.07620047, 0.07542489, 0.07502077, 0.07502077],
[0.06930015, 0.06930015, 0.06972033, 0.07053504, 0.07169145,

0.07310704, 0.07466729, 0.07622614, 0.07761383, 0.0786579 ,
 0.07922123, 0.07924878, 0.07879394, 0.07799177, 0.07701167,
 0.07601888, 0.07515276, 0.07451816, 0.07418392, 0.07418392],
 [0.06898874, 0.06898874, 0.06938268, 0.07014396, 0.07121854,
 0.07252388, 0.07394904, 0.0753585 , 0.07660286, 0.07753918,
 0.0780605 , 0.07812583, 0.07777413, 0.07710984, 0.07627234,
 0.07540588, 0.07463849, 0.07407044, 0.07376946, 0.07376946],
 [0.06898874, 0.06898874, 0.06938268, 0.07014396, 0.07121854,
 0.07252388, 0.07394904, 0.0753585 , 0.07660286, 0.07753918,
 0.0780605 , 0.07812583, 0.07777413, 0.07710984, 0.07627234,
 0.07540588, 0.07463849, 0.07407044, 0.07376946, 0.07376946]]

Значення u у вузлах сітки на 2 хвилині для сітки 20 на 20 з кроком $\Delta t=20$ с:

[[0.16299458, 0.16299458, 0.16350549, 0.16449112, 0.16587836,
 0.1675566 , 0.16937958, 0.17117245, 0.17274765, 0.17393116,
 0.17459656, 0.17469639, 0.17427545, 0.17345686, 0.17241004,
 0.17131587, 0.17033938, 0.1696127 , 0.16922646, 0.16922646],
 [0.16299458, 0.16299458, 0.16350549, 0.16449112, 0.16587836,
 0.1675566 , 0.16937958, 0.17117245, 0.17274765, 0.17393116,
 0.17459656, 0.17469639, 0.17427545, 0.17345686, 0.17241004,
 0.17131587, 0.17033938, 0.1696127 , 0.16922646, 0.16922646],
 [0.16318307, 0.16318307, 0.16372689, 0.16477904, 0.16626686,
 0.16807814, 0.17006014, 0.17202362, 0.17375708, 0.17505559,
 0.1757642 , 0.17582334, 0.17529048, 0.17431787, 0.1731066 ,

0.17186239, 0.17076578, 0.16995667, 0.1695288 , 0.1695288],
[0.163513 , 0.163513 , 0.16411903, 0.16529773, 0.16697905,
0.16905022, 0.17134888, 0.17365986, 0.17572358, 0.17726758,
0.17806965, 0.17803963, 0.17726657, 0.17597116, 0.17442334,
0.17287859, 0.17154553, 0.17057628, 0.17006814, 0.17006814],
[0.16391495, 0.16391495, 0.16460787, 0.16596535, 0.1679252 ,
0.17038031, 0.17316257, 0.17602556, 0.17863728, 0.18060654,
0.1815756 , 0.18138358, 0.18019158, 0.17835729, 0.17627119,
0.17426353, 0.17257718, 0.17137388, 0.17074998, 0.17074998],
[0.16430336, 0.16430336, 0.16510108, 0.16667769, 0.16898822,
0.17194532, 0.17539135, 0.17905812, 0.18252374, 0.18520488,
0.18647205, 0.18598635, 0.18408836, 0.18140915, 0.17853346,
0.17588411, 0.17372946, 0.17222576, 0.17145614, 0.17145614],
[0.16458417, 0.16458417, 0.16549433, 0.16731111, 0.17001998,
0.17357689, 0.17787031, 0.18265138, 0.18742363, 0.19132415,
0.19317042, 0.19210821, 0.18899482, 0.18501599, 0.18103806,
0.17756134, 0.17483882, 0.17298637, 0.17205194, 0.17205194],
[0.16466493, 0.16466493, 0.16568059, 0.167729 , 0.17084018,
0.17504392, 0.18033402, 0.18657945, 0.19332382, 0.19940753,
0.20255663, 0.20019124, 0.19489545, 0.18894779, 0.18351311,
0.17905418, 0.17571 , 0.17349684, 0.17239758, 0.17239758],
[0.16446788, 0.16446788, 0.16556398, 0.16779579, 0.17124579,
0.17604329, 0.18235861, 0.19034991, 0.19994053, 0.21007506,
0.21675303, 0.2108547 , 0.20150475, 0.19270913, 0.18552976,
0.1800516 , 0.17612742, 0.17360181, 0.17236597, 0.17236597],
[0.16394513, 0.16394513, 0.16507842, 0.16740208, 0.17104266,
0.17622522, 0.18332859, 0.19297597, 0.20611169, 0.22356224,
0.24173569, 0.22433375, 0.20765988, 0.19531176, 0.18646994,
0.18019926, 0.17588909, 0.17317825, 0.17186723, 0.17186723],
[0.16309244, 0.16309244, 0.16420769, 0.16649994, 0.17010931,

0.17529754, 0.18255039, 0.19284337, 0.2084467 , 0.23580526,
0.29741488, 0.23656478, 0.20997056, 0.19514178, 0.18564029,
0.17920475, 0.17487203, 0.17217413, 0.17087545, 0.17087545],
[0.16195614, 0.16195614, 0.16299892, 0.16513536, 0.16847853,
0.17323027, 0.17973169, 0.18854585, 0.20052786, 0.21642625,
0.23297362, 0.21717059, 0.20202048, 0.19079515, 0.18275167,
0.17704239, 0.17311475, 0.17064271, 0.1694466 , 0.1694466],
[0.16062784, 0.16062784, 0.16155855, 0.16345067, 0.16636845,
0.17041329, 0.17571939, 0.18241047, 0.19041641, 0.19885927,
0.20442885, 0.19958631, 0.19187317, 0.18460267, 0.17865662,
0.17411032, 0.17084736, 0.16874394, 0.1677136 , 0.1677136],
[0.15922683, 0.15922683, 0.1600278 , 0.16163949, 0.16407848,
0.16735877, 0.17146562, 0.17628972, 0.18147765, 0.18614927,
0.18858985, 0.18685824, 0.18289677, 0.17842154, 0.1743145 ,
0.17093151, 0.16838533, 0.16669555, 0.16585471, 0.16585471],
[0.15787759, 0.15787759, 0.15855181, 0.15989356, 0.16188469,
0.16448344, 0.16759945, 0.17104737, 0.17447306, 0.17727429,
0.17863435, 0.17796585, 0.17585582, 0.17312062, 0.17036216,
0.16793424, 0.16602186, 0.16471479, 0.16405359, 0.16405359],
[0.15669058, 0.15669058, 0.15725568, 0.15836863, 0.15999067,
0.16205208, 0.16443594, 0.16695473, 0.16932576, 0.17116885,
0.17208054, 0.171845 , 0.17067628, 0.16897592, 0.16712187,
0.16539399, 0.16397612, 0.16298007, 0.16246821, 0.16246821],
[0.15575022, 0.15575022, 0.15623184, 0.15717206, 0.15852203,
0.16020135, 0.16209017, 0.16402138, 0.16577927, 0.16711813,
0.16781839, 0.16778196, 0.16710401, 0.16600091, 0.16471459,
0.16345598, 0.16238596, 0.16161593, 0.16121462, 0.16121462],
[0.15510992, 0.15510992, 0.15553659, 0.15636428, 0.15754005,
0.15898104, 0.16057186, 0.16216507, 0.16358901, 0.16466914,
0.16526642, 0.16532452, 0.16489606, 0.16411599, 0.16315402,

0.16217573, 0.16132064, 0.16069356, 0.16036314, 0.16036314],
[0.15479235, 0.15479235, 0.15519241, 0.15596592, 0.15705876,
0.15838802, 0.15984205, 0.16128414, 0.16256324, 0.16353461,
0.16408955, 0.16418578, 0.16386147, 0.16322079, 0.16240314,
0.16155283, 0.16079784, 0.16023827, 0.15994161, 0.15994161],
[0.15479235, 0.15479235, 0.15519241, 0.15596592, 0.15705876,
0.15838802, 0.15984205, 0.16128414, 0.16256324, 0.16353461,
0.16408955, 0.16418578, 0.16386147, 0.16322079, 0.16240314,
0.16155283, 0.16079784, 0.16023827, 0.15994161, 0.15994161]]

Значення u у вузлах сітки на 3 хвилині для сітки 20 на 20 з кроком $\Delta t=20$ с:

[[0.24918262, 0.24918262, 0.24969388, 0.25068019, 0.25206839,
0.25374784, 0.25557222, 0.25736662, 0.25894342, 0.26012855,
0.26079556, 0.26089692, 0.2604774 , 0.25966007, 0.25861432,
0.25752101, 0.25654515, 0.25581888, 0.25543284, 0.25543284],
[0.24918262, 0.24918262, 0.24969388, 0.25068019, 0.25206839,
0.25374784, 0.25557222, 0.25736662, 0.25894342, 0.26012855,
0.26079556, 0.26089692, 0.2604774 , 0.25966007, 0.25861432,
0.25752101, 0.25654515, 0.25581888, 0.25543284, 0.25543284],
[0.24937076, 0.24937076, 0.24991493, 0.25096775, 0.25245653,
0.25426901, 0.25625238, 0.25821737, 0.25995239, 0.2612525 ,
0.2619627 , 0.26202339, 0.26149197, 0.26052067, 0.25931049,
0.25806716, 0.25697119, 0.25616251, 0.25573484, 0.25573484],
[0.2497 , 0.2497 , 0.25030637, 0.25148575, 0.25316802,
0.25524036, 0.25754034, 0.25985275, 0.26191795, 0.26346348,
0.26426711, 0.26423867, 0.26346713, 0.2621731 , 0.26062646,
0.25908264, 0.25775026, 0.25678146, 0.25627354, 0.25627354],
[0.25010098, 0.25010098, 0.25079424, 0.25215238, 0.25411315,
0.25656937, 0.25935284, 0.26221711, 0.26483015, 0.26680082,
0.2677714 , 0.267581 , 0.26639065, 0.2645579 , 0.26247314,
0.26046653, 0.25878093, 0.25757811, 0.25695444, 0.25695444],

[0.25048817, 0.25048817, 0.25128623, 0.25286347, 0.25517487,
0.25813296, 0.26158002, 0.26524782, 0.2687145 , 0.27139684,
0.27266546, 0.27218145, 0.27028532, 0.26760792, 0.26473381,
0.2620857 , 0.25993193, 0.25842877, 0.25765941, 0.25765941],
[0.25076757, 0.25076757, 0.25167805, 0.25349542, 0.25620506,
0.25976279, 0.26405698, 0.26883873, 0.27361167, 0.27751313,
0.27936079, 0.27830033, 0.27518906, 0.27121241, 0.26723643,
0.26376122, 0.26103975, 0.25918794, 0.2582538 , 0.2582538],
[0.25084676, 0.25084676, 0.25186272, 0.25391167, 0.25702351,
0.26122786, 0.2665184 , 0.2727641 , 0.27950878, 0.28559338,
0.28874412, 0.28638023, 0.28108663, 0.27514153, 0.2697092 ,
0.26525211, 0.26190919, 0.25969679, 0.25859788, 0.25859788],
[0.25064805, 0.25064805, 0.25174443, 0.25397672, 0.25742726,
0.26222518, 0.26854063, 0.27653187, 0.28612279, 0.29625914,
0.3029407 , 0.29704193, 0.28769323, 0.27890019, 0.27172351,
0.26624749, 0.26232477, 0.25980003, 0.25856459, 0.25856459],
[0.2501236 , 0.2501236 , 0.25125715, 0.25358126, 0.2572223 ,
0.26240514, 0.26950848, 0.27915575, 0.29229264, 0.30974909,
0.32793656, 0.31052376, 0.29384704, 0.28150064, 0.27266154,
0.26639319, 0.26208461, 0.25937472, 0.25806413, 0.25806413],
[0.24926923, 0.24926923, 0.25038474, 0.25267743, 0.25628724,
0.26147573, 0.26872855, 0.27902166, 0.29462777, 0.32200154,
0.38367798, 0.32276422, 0.29615785, 0.28132918, 0.27183017,
0.26539694, 0.26106584, 0.2583689 , 0.25707065, 0.25707065],
[0.24813133, 0.24813133, 0.24917437, 0.25131127, 0.25465493,
0.25940702, 0.26590852, 0.27472271, 0.28670592, 0.30260982,
0.31917007, 0.30335731, 0.28820476, 0.27698111, 0.26894022,
0.26323314, 0.25930702, 0.25683588, 0.25564019, 0.25564019],
[0.24680154, 0.24680154, 0.24773253, 0.24962514, 0.25254351,
0.25658887, 0.26189531, 0.26858656, 0.27659302, 0.28503765,

0.29061047, 0.28576785, 0.27805599, 0.27078786, 0.26484424,
0.26029989, 0.25703828, 0.25493566, 0.2539057 , 0.2539057],
[0.24539919, 0.24539919, 0.24620046, 0.24781269, 0.25025237,
0.25353339, 0.25764088, 0.26246551, 0.26765407, 0.27232674,
0.27476895, 0.27303886, 0.2690794 , 0.26460643, 0.26050146,
0.25712009, 0.25457506, 0.25288597, 0.25204547, 0.25204547],
[0.24404879, 0.24404879, 0.24472332, 0.24606566, 0.24805757,
0.25065722, 0.25377415, 0.25722298, 0.26064962, 0.26345199,
0.26481352, 0.2641467 , 0.26203858, 0.25930531, 0.25654855,
0.25412197, 0.25221055, 0.25090409, 0.25024318, 0.25024318],
[0.24286085, 0.24286085, 0.24342627, 0.24453982, 0.24616271,
0.24822514, 0.25061011, 0.25313008, 0.25550235, 0.25734678,
0.25825997, 0.25802608, 0.25685907, 0.25516035, 0.25330772,
0.25158099, 0.25016395, 0.24916844, 0.24865683, 0.24865683],
[0.24191977, 0.24191977, 0.24240171, 0.24334256, 0.24469341,
0.24637382, 0.24826388, 0.25019643, 0.25195574, 0.25329608,
0.2539979 , 0.25396306, 0.25328668, 0.25218504, 0.25089998,
0.24964238, 0.24857311, 0.24780357, 0.24740251, 0.24740251],
[0.241279 , 0.241279 , 0.24170599, 0.24253431, 0.24371099,
0.2451531 , 0.24674522, 0.24833986, 0.24976531, 0.250847 ,
0.25144586, 0.25150553, 0.25107856, 0.25029985, 0.24933905,
0.24836171, 0.24750734, 0.24688072, 0.24655053, 0.24655053],
[0.2409612 , 0.2409612 , 0.24136158, 0.24213572, 0.24322947,
0.24455987, 0.24601523, 0.24745878, 0.24873943, 0.24971239,
0.25026892, 0.2503667 , 0.25004385, 0.2494045 , 0.24858798,
0.2477386 , 0.24698431, 0.24642519, 0.24612876, 0.24612876],
[0.2409612 , 0.2409612 , 0.24136158, 0.24213572, 0.24322947,
0.24455987, 0.24601523, 0.24745878, 0.24873943, 0.24971239,
0.25026892, 0.2503667 , 0.25004385, 0.2494045 , 0.24858798,
0.2477386 , 0.24698431, 0.24642519, 0.24612876, 0.24612876]]

Значення u у вузлах сітки на 5 хвилині для сітки 20 на 20 з кроком $\Delta t=20$ с:

[[0.56519014, 0.56519014, 0.56570141, 0.56668775, 0.56807598,
0.56975549, 0.57157992, 0.57337439, 0.57495126, 0.57613647,
0.57680356, 0.57690501, 0.57648556, 0.57566831, 0.57462262,
0.57352936, 0.57255355, 0.57182731, 0.57144129, 0.57144129],
[0.56519014, 0.56519014, 0.56570141, 0.56668775, 0.56807598,
0.56975549, 0.57157992, 0.57337439, 0.57495126, 0.57613647,
0.57680356, 0.57690501, 0.57648556, 0.57566831, 0.57462262,
0.57352936, 0.57255355, 0.57182731, 0.57144129, 0.57144129],
[0.56537826, 0.56537826, 0.56592244, 0.56697529, 0.56846411,
0.57027664, 0.57226007, 0.57422512, 0.57596022, 0.57726041,
0.57797068, 0.57803146, 0.57750012, 0.57652888, 0.57531878,
0.5740755 , 0.57297957, 0.57217092, 0.57174327, 0.57174327],
[0.56570747, 0.56570747, 0.56631386, 0.56749326, 0.56917557,
0.57124796, 0.573548 , 0.57586047, 0.57792574, 0.57947135,
0.58027507, 0.5802467 , 0.57947525, 0.57818129, 0.5766347 ,
0.57509095, 0.57375861, 0.57278984, 0.57228193, 0.57228193],
[0.56610841, 0.56610841, 0.56680168, 0.56815985, 0.57012066,
0.57257692, 0.57536045, 0.57822479, 0.58083791, 0.58280865,
0.58377932, 0.583589 , 0.58239873, 0.58056604, 0.57848134,
0.57647479, 0.57478923, 0.57358645, 0.57296279, 0.57296279],
[0.56649555, 0.56649555, 0.56729362, 0.56887089, 0.57118232,
0.57414046, 0.57758758, 0.58125545, 0.58472222, 0.58740464,
0.58867334, 0.58818942, 0.58629335, 0.58361602, 0.58074197,
0.57809391, 0.57594018, 0.57443706, 0.57366771, 0.57366771],
[0.56677489, 0.56677489, 0.56768539, 0.56950278, 0.57221246,
0.57577024, 0.58006449, 0.58484632, 0.58961935, 0.59352091,
0.59536867, 0.59430828, 0.59119707, 0.58722047, 0.58324453,
0.57976937, 0.57704795, 0.57519617, 0.57426205, 0.57426205],
[0.56685402, 0.56685402, 0.56787 , 0.56991897, 0.57303084,

0.57723525, 0.58252586, 0.58877165, 0.59551646, 0.60160116,
0.604752 , 0.60238818, 0.59709462, 0.59114955, 0.58571726,
0.5812602 , 0.57791733, 0.57570495, 0.57460606, 0.57460606],
[0.56665525, 0.56665525, 0.56775164, 0.56998396, 0.57343453,
0.57823251, 0.58454805, 0.5925394 , 0.60213046, 0.61226692,
0.61894852, 0.61304986, 0.60370121, 0.59490818, 0.58773152,
0.58225553, 0.57833285, 0.57580814, 0.57457271, 0.57457271],
[0.56613074, 0.56613074, 0.5672643 , 0.56958843, 0.57322952,
0.57841242, 0.58551585, 0.59516325, 0.60830027, 0.62575674,
0.64394396, 0.62653157, 0.60985499, 0.5975086 , 0.58866951,
0.58240118, 0.57809263, 0.57538276, 0.57407219, 0.57407219],
[0.56527631, 0.56527631, 0.56639183, 0.56868455, 0.5722944 ,
0.57748295, 0.58473587, 0.5950291 , 0.61063531, 0.63800883,
0.69968326, 0.63877167, 0.6121657 , 0.59733708, 0.58783808,
0.58140487, 0.5770738 , 0.57437688, 0.57307865, 0.57307865],
[0.56413836, 0.56413836, 0.56518141, 0.56731833, 0.57066203,
0.57541418, 0.58191577, 0.59073008, 0.60271342, 0.61861735,
0.63517738, 0.619365 , 0.60421257, 0.59298895, 0.58494807,
0.57924101, 0.57531492, 0.57284381, 0.57164814, 0.57164814],
[0.56280852, 0.56280852, 0.56373952, 0.56563216, 0.56855056,
0.57259598, 0.5779025 , 0.58459386, 0.59260044, 0.60104518,
0.60661805, 0.60177553, 0.59406372, 0.58679562, 0.58085202,
0.57630771, 0.57304613, 0.57094354, 0.56991359, 0.56991359],
[0.56140613, 0.56140613, 0.56220741, 0.56381966, 0.56625938,
0.56954045, 0.57364801, 0.57847274, 0.58366139, 0.58833417,
0.59077647, 0.58904646, 0.58508704, 0.58061412, 0.57650918,
0.57312786, 0.57058287, 0.56889381, 0.56805333, 0.56805333],
[0.56005569, 0.56005569, 0.56073024, 0.5620726 , 0.56406455,
0.56666424, 0.56978124, 0.57323014, 0.57665687, 0.57945934,
0.58082095, 0.58015421, 0.57804616, 0.57531294, 0.57255623,

0.5701297 , 0.56821833, 0.5669119 , 0.566251 , 0.566251],
[0.55886772, 0.55886772, 0.55943315, 0.56054673, 0.56216966,
0.56423213, 0.56661717, 0.56913721, 0.57150955, 0.57335408,
0.57426735, 0.57403353, 0.57286659, 0.57116794, 0.56931538,
0.5675887 , 0.5661717 , 0.56517622, 0.56466462, 0.56466462],
[0.55792662, 0.55792662, 0.55840857, 0.55934945, 0.56070034,
0.5623808 , 0.56427092, 0.56620354, 0.56796292, 0.56930334,
0.57000524, 0.56997048, 0.56929418, 0.56819261, 0.56690761,
0.56565007, 0.56458084, 0.56381133, 0.56341028, 0.56341028],
[0.55728584, 0.55728584, 0.55771284, 0.55854118, 0.5597179 ,
0.56116006, 0.56275225, 0.56434696, 0.56577248, 0.56685425,
0.56745319, 0.56751294, 0.56708605, 0.56630741, 0.56534667,
0.56436939, 0.56351505, 0.56288847, 0.5625583 , 0.5625583],
[0.55696803, 0.55696803, 0.55736842, 0.5581426 , 0.55923638,
0.56056683, 0.56202225, 0.56346586, 0.56474659, 0.56571963,
0.56627624, 0.5663741 , 0.56605133, 0.56541205, 0.5645956 ,
0.56374627, 0.56299202, 0.56243293, 0.56213652, 0.56213652],
[0.55696803, 0.55696803, 0.55736842, 0.5581426 , 0.55923638,
0.56056683, 0.56202225, 0.56346586, 0.56474659, 0.56571963,
0.56627624, 0.5663741 , 0.56605133, 0.56541205, 0.5645956 ,
0.56374627, 0.56299202, 0.56243293, 0.56213652, 0.56213652]]