

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ  
СІКОРСЬКОГО»

КУРСОВА

з кредитного модуля «Методи обчислень»

на тему:

«ЧИСЛОВЕ РОЗВ’ЯЗАННЯ ДИФЕРЕНЦІЙНИХ  
РІВНЯНЬ У ЧАСТКОВИХ ПОХІДНИХ»

Варіант № 1

Виконала

студентка 3 курсу ФТІ

Групи ФІ-81

Бражник Марія Андріївна

номер залікової книжки 8101

Перевірив:

Оцінка:

Київ 2021

<b>1. ОГЛЯД І АНАЛІЗ МЕТОДІВ РОЗВ'ЯЗАННЯ ДРЧП.</b>	<b>3</b>
1.1. Постановка задачі.	3
1.2. Особливості задачі.	3
1.3. Огляд та аналіз існуючих методів розв'язання ДРЧП.	4
1.4. Вибір методу.	6
<b>2. РОЗВ'ЯЗАННЯ ДРЧП.</b>	<b>7</b>
2.1 Дослідження умов застосування обраного методу.	7
<b>3. ПРОГРАМНА РЕАЛІЗАЦІЯ.</b>	<b>9</b>
3.1. Передпрограмовий етап.	9
3.2. Програмна реалізація.	9
3.3. Приклади роботи програми.	10
<b>4. ОГЛЯД МЕТОДІВ ПІДВИЩЕННЯ ТОЧНОСТІ ТА ЕФЕКТИВНОСТІ</b>	<b>15</b>
4.1 Підвищення точності методів розв'язання ДРЧП	15
4.2 Підвищення ефективності методів розв'язання ДРЧП	15
<b>5. ВИСНОВКИ</b>	<b>17</b>
<b>6. СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</b>	<b>18</b>
<b>7. ДОДАТКИ.</b>	<b>19</b>
Додаток 1. Лістинг коду програми мовою Python.	19
Додаток 2. Масиви числових значень розв'язку.	21

# 1. ОГЛЯД І АНАЛІЗ МЕТОДІВ РОЗВ'ЯЗАННЯ ДРЧП.

## 1.1. Постановка задачі.

Варіант 1.

Процес розповсюдження забруднюючої домішки описується диференціальним рівнянням у часткових похідних:

$$\frac{\partial u}{\partial t} = k \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) - d \cdot u + f,$$

де  $u$  - концентрація речовини,

$k$  - коефіцієнт дифузії,  $k = 20$ .

$d$  - коефіцієнт поглинання домішки,  $d = 0.00006$ .

$f$  - функція джерела забруднення,  $f = 0.8$  мкг\с.

Розміри області - 1х1 км, висота джерела забруднення 300 м.

Вирази для граничних умов:

$$\frac{\partial u}{\partial n} = 0,$$

На початку розвитку процесу у всіх внутрішніх точках області встановлюється однакова концентрація. В початковий момент часу концентрація речовини дорівнює  $100 \text{ мкг}\backslash\text{м}^2$ .

Змодельовати поведінку забруднюючої домішки протягом 8 хвилин.

## 1.2. Особливості задачі.

Це рівняння є рівнянням параболічного типу (перша похідна по часу та друга по координатам). Воно є еволюційним, так як є залежність  $u$  від часу.

Рівняння параболічного типу найбільш часто зустрічаються при вивченні процесів теплопровідності та дифузії.

Найпростішим представником рівнянь параболічного типу є рівняння теплопровідності.

### 1.3. Огляд та аналіз існуючих методів розв'язання ДРЧП.

Диференціальні рівняння у часткових похідних широко використовуються у якості моделей багатьох практично значущих явищ оточуючого середовища і промисловості. За допомогою диференціальних рівнянь у часткових похідних моделюється ряд важливих процесів, такий як процеси теплопровідності, дифузії, коливань у пружних середовищах та ін.

Іноді складно отримати аналітичний розв'язок загального виду для диференціальних рівнянь у часткових похідних, а у ряді процесів взагалі неможливо. Тож застосування апарату числових методів є одним зі способів розв'язання задач моделювання, де використовуються такі рівняння на етапі постановки задачі.

Загальний метод розв'язання диференційних рівнянь в часткових похідних (ДРЧП) полягає у застосуванні наближених моделей. Широкого застосування набули моделі, побудовані на методі сіток, який полягає в апроксимації шуканої неперервної функції сукупністю наближених значень, розрахованих в скінченному наборі точок області – вузлах. Сукупність вузлів утворює сітку. Використання цього методу дозволяє звести диференціальну граничну задачу до системи нелінійних рівнянь, або, у частковому випадку, ітераційної формули обчислень відносно невідомих вузлових значень функції.

Часто використовуваними є метод скінченних елементів (МСР) та метод скінченних різниць (МСР).

Основною відмінністю цих методів є те, що в МСР апроксимуються похідні шуканих функцій, а в МСР – сам розв'язок - залежність від просторових координат і часу. Також методи відрізняються в конструкціях використовуваних сіток.

Розглянемо дані методи ближче.

У МСР використовуються регулярні сітки. Цей метод використовується переважно в областях із прямолінійними границями, тож відбувається

апроксимація прямокутникам. Традиційно розв'язувані за допомогою МСР практичні задачі – це задачі дифузії та тепломасообміну рідин та газів.

Областю використання МСЕ є коло задач з довільною геометрією (розрахунок на міцність деталей та вузлів будівельних конструкцій, авіаційних і космічних апаратів, тепловий розрахунок двигунів тощо), тож для цього методу використовується розбиття з урахуванням геометричних особливостей області. Також цей метод не дуже підходить для еволюційних задач, тому що в такому випадку вона набуває досить складної реалізації. Краще застосовувати його для стаціонарних задач.

Загальною проблемою обох методів є висока розмірність результуючої системи алгебраїчних рівнянь. Тому прикладні реалізації МСР і МСЕ потребують розробки спеціальних способів зберігання матриці коефіцієнтів системи і методів розв'язання СЛАР.

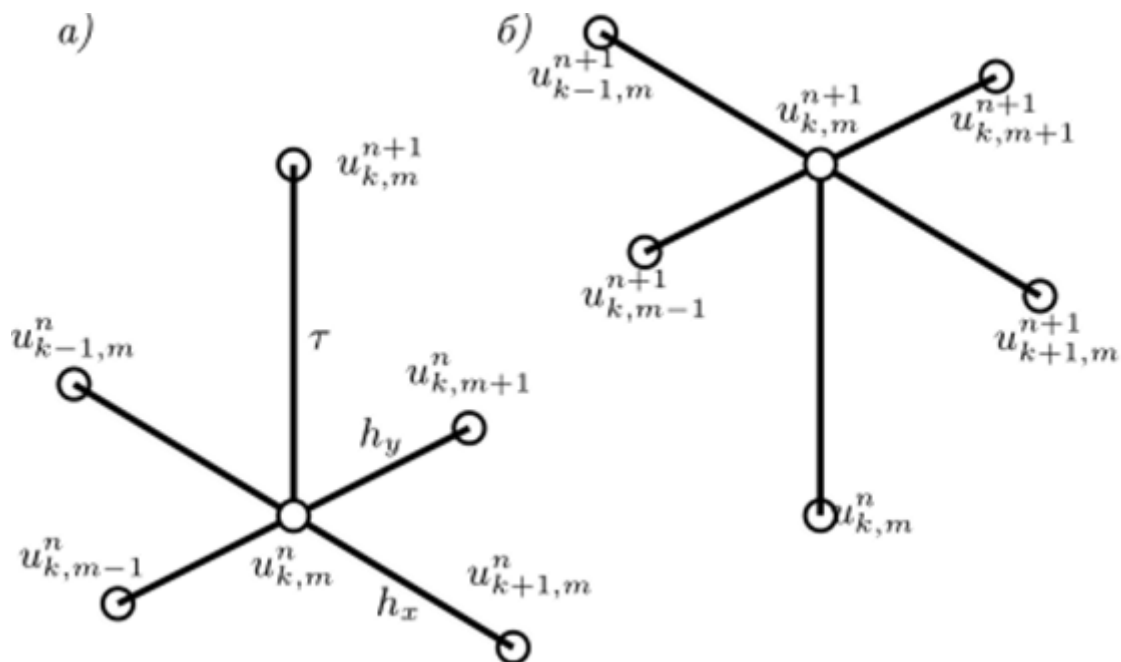
Розглянемо явну і неявну схеми.

Основна ідея явної схеми полягає в отриманні формули, що явно виражає значення рішення для конкретної точки в момент часу на основі значень в попередній момент часу. Недоліком даної схеми є те, що якщо є необхідність збільшення точності апроксимації похідних, то при зміні розмірів сітки зростає не тільки об'єм обчислень, але й похибка округлення. Якщо зменшувати кроки по часу та простору, то похибка апроксимації буде зменшуватись також, але кількість обчислень буде зростати.

Неявна схема використовує значення точок в поточний момент часу і вже явно не залежить від значень на попередньому кроці.

З цих двох схем явна є більш стійкою, але потребує значну кількість обчислень та складну матрицю зі взаємозв'язками точок наступного моменту часу. Хоча цей метод не використовує явно дані з сусідніх точок поточного моменту часу.

Схема Кранка-Ніколсон є поєднанням попередніх двох методів, використовуючи як значення точок на попередньому кроці, так і на поточному.



а) Шаблон для явної схеми, б) Шаблон для неявної схеми.

#### 1.4. Вибір методу.

Для двовимірного випадку рівняння параболічного типу було обрано метод скінченних різниць із схемою Кранка-Ніколсон (явно-неявна схема), тому що я вважаю його найраціональнішим через використання даних сусідніх вузлів на попередньому кроці та на поточному.

На кожному шарі буде обчислено значення за допомогою значень з попереднього шару і сусідніх вузлів, а значення кожного вузла першого шару ми отримаємо з граничних і початкових умов. Після дискретизації у мене немає жодної нелінійності, тому можна робити СЛАР для кожного шару (по п'ять невідомих на рядок). Отже, мій вибір є оптимальним.

## 2. РОЗВ'ЯЗАННЯ ДРЧП.

### 2.1 Дослідження умов застосування обраного методу.

В якості різницевої схеми використовується схема Кранка-Ніколсон.

Розглянемо рівняння:

$$\frac{\partial u}{\partial t} = k \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) - d \cdot x + f.$$

Запишемо його дискретне представлення:

$$\begin{aligned} \frac{u_{ij}^{k+1} - u_{ij}^k}{\Delta t} = & \lambda \left( k \left( \frac{u_{i+1j}^k - 2u_{ij}^k + u_{i-1j}^k}{\Delta x^2} + \frac{u_{ij+1}^k - 2u_{ij}^k + u_{ij-1}^k}{\Delta y^2} \right) - d \cdot u_{ij}^k + f_{ij}^k \right) \\ & \times \left( k \left( \frac{u_{i+1j}^{k+1} - 2u_{ij}^{k+1} + u_{i-1j}^{k+1}}{\Delta x^2} + \frac{u_{ij+1}^{k+1} - 2u_{ij}^{k+1} + u_{ij-1}^{k+1}}{\Delta y^2} \right) - d \cdot u_{ij}^{k+1} + f_{ij}^{k+1} \right) \\ & . \end{aligned}$$

Застосовуючи дискретизацію до граничних точок, отримаємо такі рівняння на нижній границі:

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

На верхній границі:

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

На лівій границі:

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

На правій границі:

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

$$i, j = 0, 1, \dots, n.$$

$$k = 0, 1, \dots, T.$$

Якщо перенести всі  $u$  з індексами  $k+1$  вліво, а все інше - вправо, то побачимо, що в нас є лінійне рівняння з 5 невідомими для 5 вузлів на  $k+1$  кроці зліва і дійсне число справа (значення  $u$  на  $k$  кроці вважається відомим).

Перебираючи індекс  $n$  (де  $n$  кількість вузлів на стороні квадратної сітки), маємо систему лінійних алгебраїчних рівнянь з  $n^2$  невідомими.

Загальний вигляд такої системи:  $Au = b$ , де  $A$  - матриця системи  $n^2 \times n^2$ ,  $u$  - стовпчик невідомих,  $b$  - стовпчик вільних членів  $n^2 \times 1$ .

Вирішуючи цю систему на кожному кроці, отримаємо значення для кожного вузла шару і змогу вирішувати таку систему на наступному кроці.

Розіб'ємо квадрат  $1 \times 1$  км на  $n \times n$  дискретних вузлів з кроком  $\Delta x$ ,  $\Delta y = 1000/n$ , оберемо крок по часу  $\Delta t = 480/T$ .

Індекс координати простору  $x$   $i = 0, 1, 2, \dots, n$  відповідатиме точкам простору  $i \cdot \Delta x$ , індекс координати простору  $y$   $j = 0, 1, 2, \dots, n$  відповідатиме точкам простору  $j \cdot \Delta y$ , а індекс часу  $k = 0, 1, 2, \dots, T$  відповідатиме точкам часу  $k \cdot \Delta t$ .

Тепер маємо задачу - сукупність систем рівнянь відносно невідомих  $u_{i+1j}, u_{i-1j}, u_{ij}, u_{ij+1}, u_{ij-1}$ , вирішуючи її на  $k+1$  кроці за допомогою значень з  $k$  кроці, будемо отримувати значення вузлів на  $k+1$  кроці. Матриця  $A$  на кожному кроці буде завжди мати вигляд, де на головній діагоналі будуть коефіцієнти біля  $u_{ij}$ , на координаті  $ij \pm 1$  коефіцієнти біля  $u_{ij+1}, u_{ij-1}$  відповідно, на координаті  $ij \pm n$  коефіцієнти для  $u_{i+1j}$  та  $u_{i-1j}$  відповідно.

Після дискретизації отримаємо доволі велику СЛАР.

Для її розв'язку доцільно використовувати такі прямі методи розв'язання СЛАР як метод квадратного кореня або метод Гауса.

Я використовую вбудований в Python метод LU декомпозиції розкладання матриці коефіцієнтів  $A$ , тому що в нашому випадку матриця  $A$  не змінюється, тож її можна розкласти один раз і далі підстановкою розв'язувати кожну з конкретних систем.



### 3. ПРОГРАМНА РЕАЛІЗАЦІЯ.

#### 3.1. Передпрограмовий етап.

Після дискретизації рівняння, зведення подібних та перенесення всіх змінних на  $k+1$  кроці вліво, а всі інші доданки - вправо, було отримано рівняння:

$$\begin{aligned} & (1 - \Delta t(1 - \lambda))(k(\frac{-2}{\Delta x^2} + \frac{-2}{\Delta y^2}) - d)u_{ij}^{k+1} - \Delta t(1 - \lambda) \times \\ & \times k(\frac{u_{i+1j}^{k+1} + u_{i-1j}^{k+1}}{\Delta x^2} + \frac{u_{ij+1}^{k+1} + u_{ij-1}^{k+1}}{\Delta y^2}) = \\ & = \lambda \Delta t(k(\frac{u_{i+1j}^k - 2u_{ij}^k + u_{i-1j}^k}{\Delta x^2} + \frac{u_{ij+1}^k - 2u_{ij}^k + u_{ij-1}^k}{\Delta y^2}) - d \cdot u_{ij}^k) + f \end{aligned}$$

Було обрано  $\lambda = 0.75$  і  $\Delta t = 20c, 15c$  для стійкості системи.

#### 3.2. Програмна реалізація.

Для програмної реалізації я обрала мову Python з використанням бібліотек `numpy`, `matplotlib`.

У програмі реалізовано:

- Вибір довжини і ширини сітки ( $n$ ).
- Вибір кроку у просторових координатах.
- Можливість змінювати параметри, початкові умови та функцію.
- Створення та заповнення матриці  $A$  та стовпчика вільних членів  $b$ .
- Вирішення СЛАР засобами Python.
- Візуалізація.

Після введення всіх констант і переходимо до основної частини програми.

Створюємо масив вузлів  $U$ , заповнюємо його на нульовому кроці згідно до початкових умов, функцію  $b$ , що повертає значення у вузлі на попередньому кроці.

Створюємо матрицю  $A$  і заповнюємо згідно до отриманого рівняння. Матриця  $A$  не змінюється від кроку до кроку, тому достатньо заповнити її один раз.

Створюємо функцію `get_vect_B`, яка повертає значення для стовпчика вільних членів.

Згідно до граничних умов, значення у вузлі на границі буде дорівнювати сусідньому внутрішньому вузлу. Тому відповідний рядок у стовпці буде дорівнювати 0, всі інші заповнюємо відповідно до формул, які відповідають значенню функції на попередньому кроці.

Створюємо функцію, що повертає значення  $U$  на  $k+1$  кроці.

СЛАР вирішую за допомогою вбудованого пакету Python.

В циклі заповнюємо  $U$  з  $k = 1$  по  $k=T$ .

Для значення  $U$  на кожному кроці виводиться графік.

### 3.3. Приклади роботи програми.

Візьмемо довжину матриці  $U$   $n = 20$ ,  $\Delta x$  та  $\Delta y = 50$  м,  $\Delta t = 20, 15$  с.

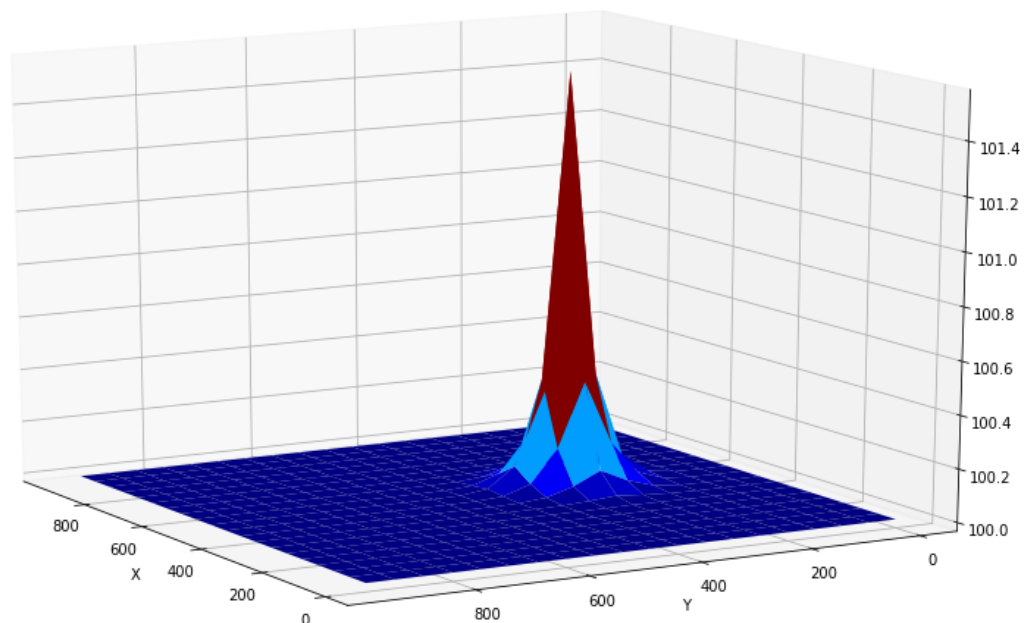
Вийде  $T = 24, 32$  кроки по часу.

Отже, потрібно подивитись розподіл концентрації на 2, 5 та 8 хвилинах.

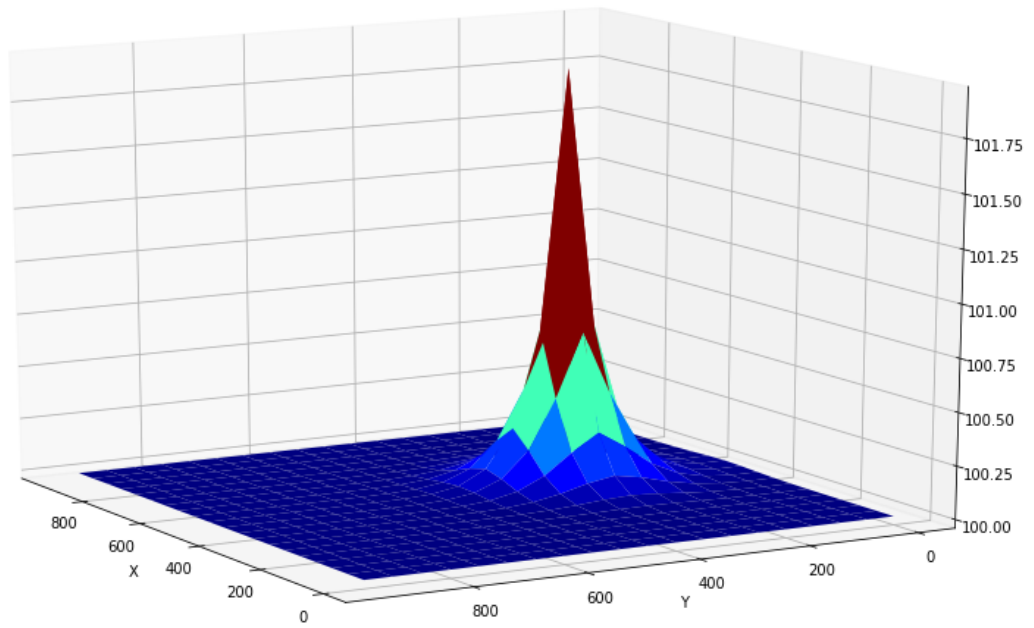
Візьмемо  $\Delta t = 20$  с.

На самих графіках можна чітко побачити розташування джерела забруднення та поширення забруднення на сусідні вузли.

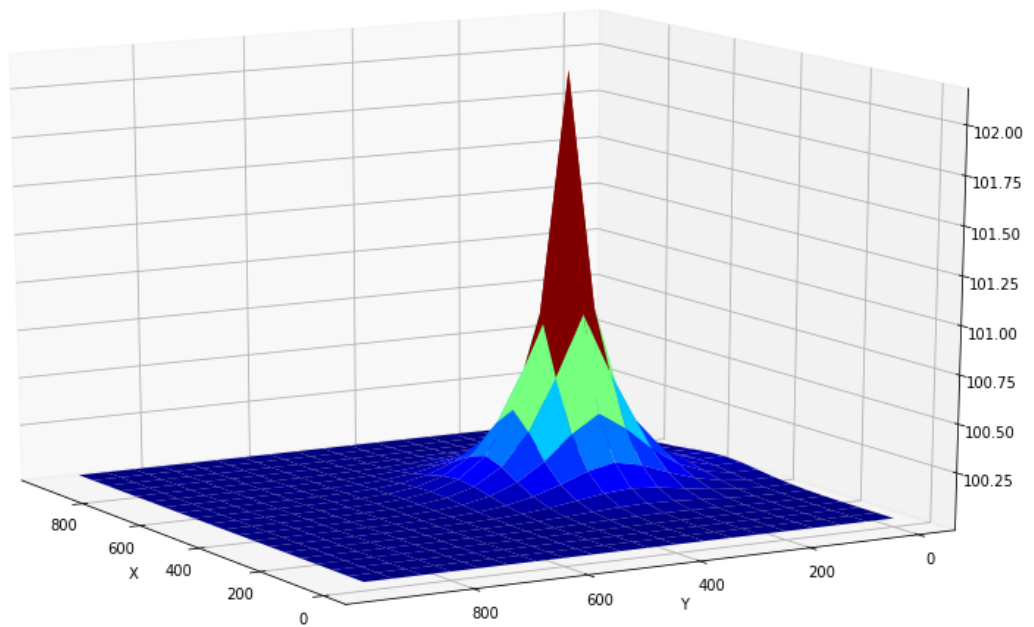
Графік концентрації  $u$  на 2 хвилині:



Графік концентрації  $u$  на 5 хвилині:

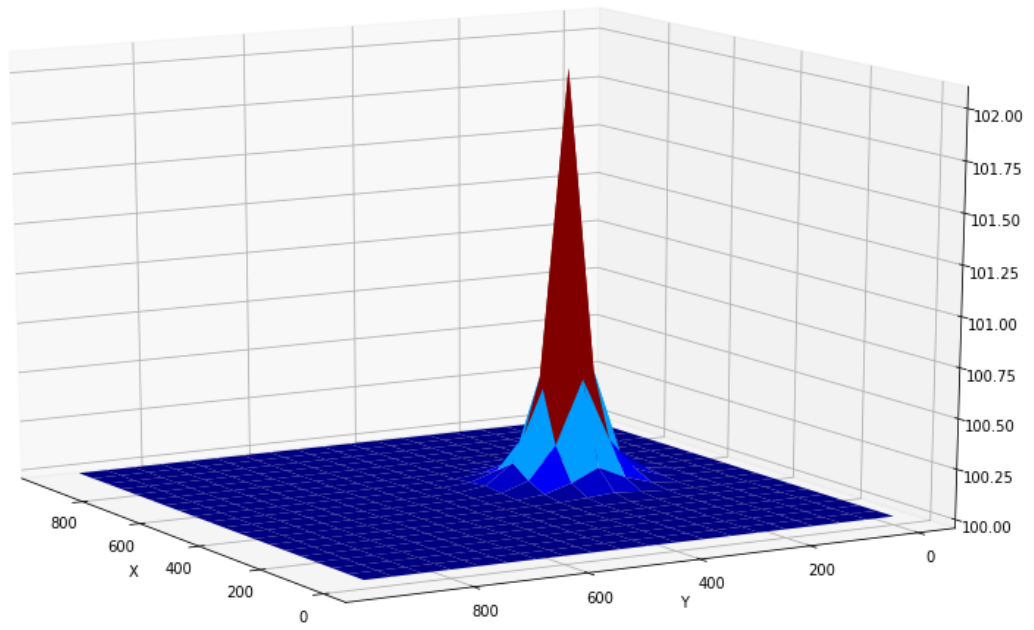


Графік концентрації  $u$  на 8 хвилині:

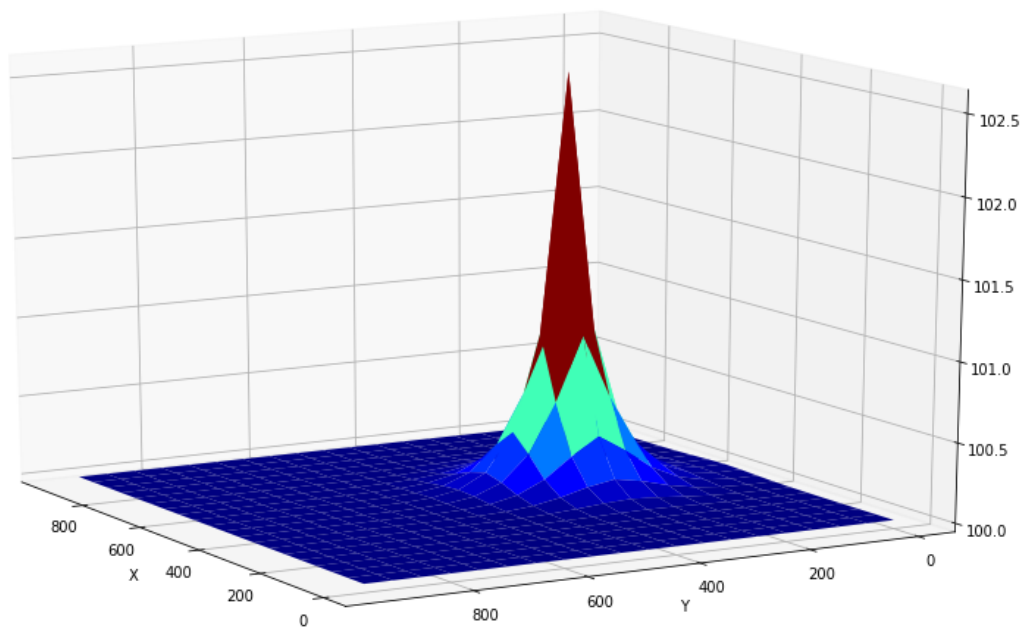


При зміні кроку на  $\Delta t = 15$  с розглянемо графік сітки 20 на 20.

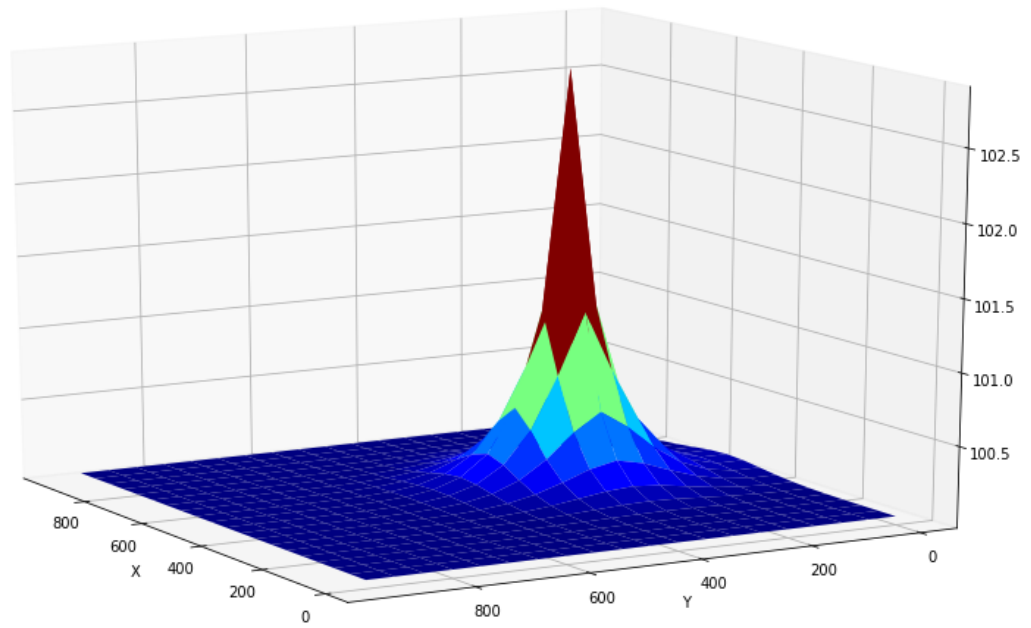
На 2 хвилині:



На 5 хвилині:

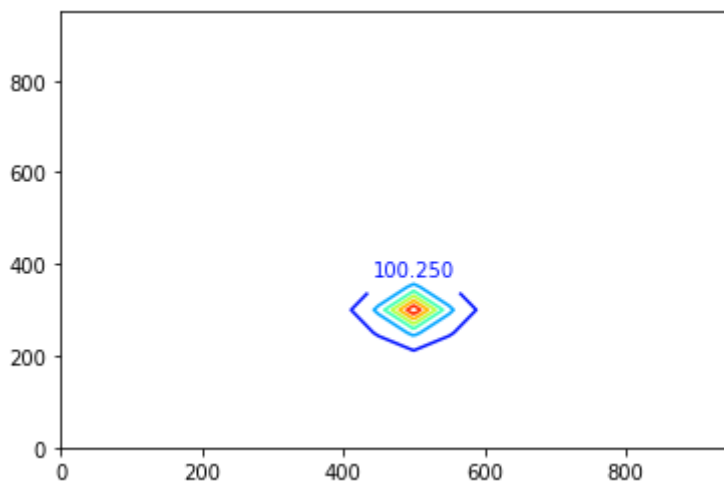


На 8 хвилині:

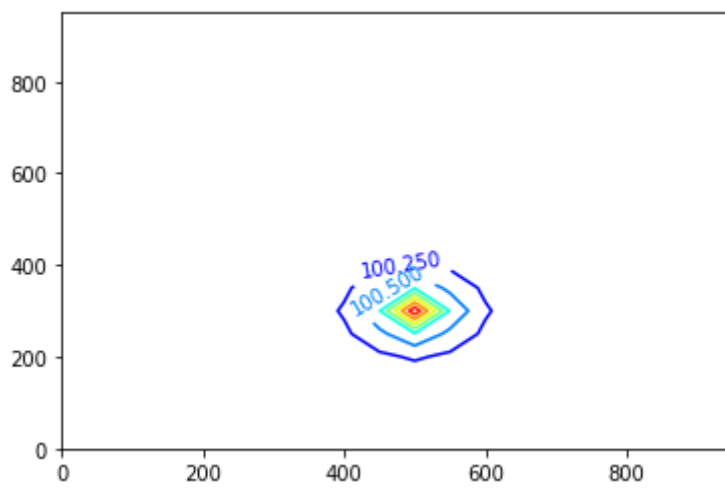


Також можна розглянути контур (вид зверху) процесу на 2, 5 та 8 хвилинах для  $\Delta t = 20c$ .

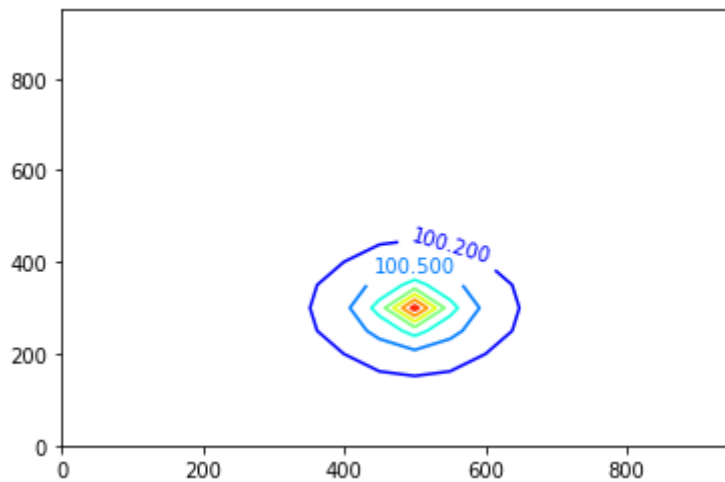
На 2 хвилині:



На 5 хвилині:



На 8 хвилині:



З цих графіків видно, що з кожною ітерацією концентрація домішки в джерелі збільшується, як і у сусідніх точках. На значеннях це видно лише у одиницях та знаках після коми, тому що у концентрація вимірюється у мкг/с. При зменшенні кроку по часу також збільшується точність значення функції.

## 4. ОГЛЯД МЕТОДІВ ПІДВИЩЕННЯ ТОЧНОСТІ ТА ЕФЕКТИВНОСТІ

### 4.1 Підвищення точності методів розв'язання ДРЧП

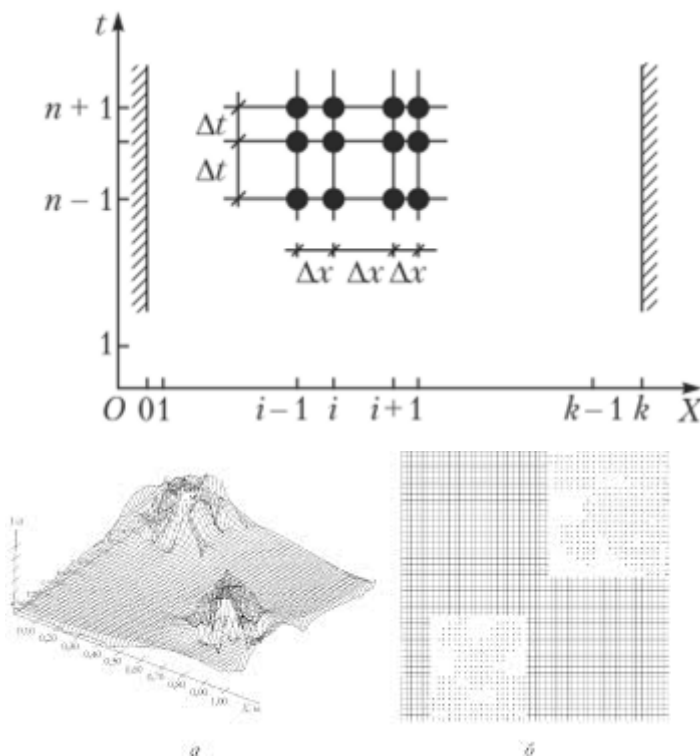
Зазвичай, для підвищення точності використовують адаптивні сітки, але через непередбачуваність моєї задачі, яка є еволюційною, їх застосування може бути неефективним.

Застосовуючи нерівномірні сітки, які будуються більш щільно у області, де є активний розвиток системи, можна збільшити точність розв'язку. В моїй задачі такою областю є джерело з околom.

Нерівномірні сітки це такі сітки, в яких крок залежить від номеру вузла.

У різницевих схемах із нерівномірною сіткою використовується підхід, заснований на розтягуванні координат. При рішенні різницеви́ми методами різних диференціальних рівнянь нерівномірні сітки використовуються дуже широко. Але питання про збіжність різницеви́х схем на цих сітках вивчений погано.

Найпростіші приклади показують, що критерій, який часто використовується для судження про якість різницеви́х схем - рівномірна оцінка або оцінка в середньому похибки апроксимації схеми - у випадку нерівномірних сіток є незмістовною і може дати неправильне представлення про порядок точності схеми.



### 4.2 Підвищення ефективності методів розв'язання ДРЧП

При розв'язанні ДРЧП за допомогою скінченно-різницевого підходу ми переходимо від неперервного розв'язку до дискретного розв'язку. Внаслідок

дискретизації рівняння при застосуванні неявної схеми виникає система алгебраїчних рівнянь великої розмірності. Розглянемо випадок, коли ця система є лінійною.

При застосуванні ітераційних методів у випадку, якщо діагональна перевага вихідної матриці була слабкою ми не досягали взагалі діагональної переваги, лише привели систему  $Ax = b$  до вигляду, коли матриця системи є симетричною та додатньо визначеною, ми можемо зіткнутися із дуже повільною збіжністю.

Після дискретизації отримуємо систему дуже великої розмірності, тож привести матрицю до діагональної переваги буде громіздким процесом, а іноді - нездійсненним. Така особливість робить застосування ітераційних методів для розв'язання великих СЛАР не найбільш вдалим рішенням.

Недолік застосування прямих методів полягає у їхній вимогливості до ресурсів ЕОМ.

Одержана в результаті дискретизації матриця  $A$  є сильно розрідженою, і має зазвичай стрічкову структуру. Перевпорядкування матриці до такої, де елементи розташовані більш щільно, і утворюють, наприклад, стрічку навколо головної діагоналі підвищує ефективність розв'язання таких систем.

Моя матриця має п'ятидіагональну структуру, але дві з цих діагоналей знаходяться на значній відстані від головної діагоналі. Хоча це не заважає раціонально зберігати цю матрицю в пам'яті.

Часто використовуваним методом є метод стрічковий метод квадратного кореня.

Моя матриця не є симетричною, тож для використання методу квадратного кореня треба буде спочатку привести її до симетричного вигляду.

Нехай  $A$  - симетрична додатньовизначена матриця, з півшириною -  $p$ , тоді півширина множника методу квадратного кореня не більше  $p$ .

Формули методу:

$$L_{jj} = \sqrt{a_{jj} - \sum_{k=\max[1, j-p]}^{j-1} L_{ik}^2}, \quad i \in [1, n],$$

$$L_{ij} = (a_{ij} - \sum_{k=\max[1, j-p]}^{i-1} L_{ik} L_{jk}) / L_{jj}, \quad j \in [1, n], \quad j \in [j+1, \min[n, p+j]].$$

Інакше  $L_{ij} = 0$ .



## 5. ВИСНОВКИ

Під час виконання даної курсової роботи з'явилися навички з реалізації та використання чисельного метод Кранка-Ніколсон для розв'язання диференціального рівняння у часткових похідних параболічного типу. Це дуже корисні та актуальні навички для спеціальності, на якій я навчаюсь, тому що це є безпосереднім моделюванням реальних систем для отримання значень функції і прогнозу поведінки системи, коли аналітично отримати розв'язок дуже важко або неможливо.

Важливо дотримуватись умов стійкості, тому що при порушеннях цих умов система перестає бути адекватною і її вже неможливо використовувати для рішення задачі.

У схемі Кранка-Ніколсон збільшення коефіцієнта  $\lambda$  та збільшення кроку по часу сприяє стійкості системи.

І на відміну від явної та неявної схем, схема Кранка-Ніколсон є набагато кращою в плані стійкості, тому що більш міцно пов'язує значення функції в попередній момент часу за наступним.

Було проведено дослідження розповсюдження забруднюючої домішки у речовині, яка знаходиться у просторі розміром 1 на 1 км. Цю задачу було вирішено за допомогою методу скінченних різниць схеми Кранка-Ніколсон. Розв'язок отриманої СЛАР було отримано за допомогою функції Python, яка використовує LU декомпозицію.

При виконанні роботи виникли деякі труднощі із

При дуже великій кількості вузлів СЛАР буде надзвичайно великою, тому дуже важливо оптимізувати її вигляд і використовувати методи, які дозволять вирішувати систему швидше з меншим використанням ресурсів ЕОМ.

З фізичної точки зору ми маємо джерело забруднення в точці, яка знаходиться на відстані 300 метрів від нижньої границі, звідки починається процес. Далі забруднення поширюється на сусідні точки за допомогою явища дифузії та поглинання домішки.

На самих графіках в кожному вузлі в певний момент часу можна побачити значення функції  $u$ , яка показує концентрацію забруднюючої домішки в цій точці та зміну стану системи від одного моменту до іншого.

## **6. СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

- 1) Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы. — М.: Лаборатория базовых знаний, 2007. — 636 с.
- 2) В. М. Марченко, О. Н. Пыжкова - УРАВНЕНИЯ МАТЕМАТИЧЕСКОЙ ФИЗИКИ, 2013, - 45 с.
- 3) Волков Е.А. Численные методы. — СПб.: Лань, 2004. — 256 с.
- 4) Шевцов Г.С., Крюкова О.Г., Мызникова Б.И. Численные методы линейной алгебры. — М., Инфра-М, 2008. — 480 с.
- 5) Фарлоу С. Уравнения с частными производными для научных работников и инженеров. — М. Мир, 1985. — 383 с.
- 6) Зенкевич О.К. Метод конечных элементов в технике. — М.: Мир, 1975. — 350 с.
- 7) А. Н. Тихонов, А. А. Самарский однородные разностные схемы на неравномерных сетках - Журнал вычислительной математики и математической физики Октябрь 1962 г 5 с.

## 7. ДОДАТКИ.

### Додаток 1. Лістинг коду програми мовою Python.

```
import numpy as np
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
n = 20
k1 = 20
k2 = 20
d = 0.00006
dx = 1000/n
dy = 1000/n
dt = 20
lam = 0.75

U = np.zeros((25, n, n))
b = np.zeros(n)
f = np.zeros((n, n))

f[int(n/2)][int(300/dy)] = 0.8

for i in range(n):
    for j in range(n):
        if i == 0 or i == n-1 or j == 0 or j == n-1:
            U[0][i][j] = 100
        else:
            U[0][i][j] = 100

def b(U1, k, i, j):
    return dt*lam*(k1*(U1[k][i+1][j] - 2*U1[k][i][j] + U1[k][i-1][j])/(dx**2) + k2*(U1[k][i][j+1] - 2*U1[k][i][j] + U1[k][i][j-1])/(dy**2)) + f[i][j] + U1[k][i][j]

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

m = len(matr_A)

for i in range(m):
    for j in range(m):
        if i <= n-1:
            if i == j:
                matr_A[i][j] = 1
                matr_A[i][j+n] = -1
            elif i >= m - n:
                if i == j:
                    matr_A[i][j] = 1
                    matr_A[i][j-n] = -1
            elif j % n == 0:
                if i == j:
                    matr_A[i][j] = 1
                    matr_A[i][j+1] = -1
```

```

elif (j+1) % 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-lam)*(-2*k1/(dx**2) - 2*k2/(dy**2))
            matr_A[i][j-n] = -dt*(1-lam)*k1/(dx**2)
            matr_A[i][j+n] = -dt*(1-lam)*k1/(dx**2)
            matr_A[i][j-1] = -dt*(1-lam)*k2/(dy**2)
            matr_A[i][j+1] = -dt*(1-lam)*k2/(dy**2)

def get_vect_B(k):
    vect_B = np.zeros(n**2)
    U1 = np.zeros((n, n))
    for i in range(1, n-1):
        for j in range(1, n-1):
            U1[i][j] = b(U, k, i, j)
    U1 = np.reshape(U1, (n**2, 1))
    for i in range(len(vect_B)):
        if i <= n-1 or i >= len(vect_B) - n or i % n == 0 or (i+1) % n == 0:
            vect_B[i] = 0
        else:
            vect_B[i] = U1[i]
    return vect_B

def U_k(k):
    U1 = np.zeros((n, n))
    res = np.linalg.solve(matr_A, get_vect_B(k))
    res = np.reshape(res, (n, n))
    for i in range(n):
        for j in range(n):
            U1[i][j] = res[i][j]
    return U1

for k in range(1, 25):
    U[k] = U_k(k-1)

fig = plt.figure(figsize=[15, 10])
ax = fig.gca(projection='3d')
ax.view_init(15, 150)
X = np.arange(0, n*dx, dx)
Y = np.arange(0, n*dy, dy)
X, Y = np.meshgrid(X, Y)
# ax.plot_surface(X, Y, U[24], cmap = 'jet')
ax.scatter(X, Y, U[24], cmap = 'jet')
ax.set_xlabel('X')
ax.set_ylabel('Y')

plt.show()

```

## Додаток 2. Масиви числових значень розв'язку.

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

```

[[100.    100.    100.    100.    100.
  100.    100.00000001 100.    100.    100.
  100.    100.    100.    100.    100.
  100.    100.    100.    100.    100. ]

[100.    100.    100.    100.    100.
  100.    100.00000001 100.    100.    100.
  100.    100.    100.    100.    100.
  100.    100.    100.    100.    100. ]

[100.    100.    100.    100.00000001 100.00000001
  100.00000003 100.00000005 100.00000003 100.00000001 100.00000001
  100.    100.    100.    100.    100.
  100.    100.    100.    100.    100. ]

[100.    100.    100.00000001 100.00000004 100.00000011
  100.00000026 100.00000039 100.00000026 100.00000011 100.00000004
  100.00000001 100.    100.    100.    100.
  100.    100.    100.    100.    100. ]

[100.00000002 100.00000002 100.00000006 100.00000024 100.00000078
  100.00000202 100.00000333 100.00000202 100.00000078 100.00000024
  100.00000006 100.00000001 100.    100.    100.
  100.    100.    100.    100.    100. ]

[100.00000008 100.00000008 100.00000035 100.00000149 100.00000545
  100.00001551 100.00002808 100.00001551 100.00000545 100.00000148
  100.00000034 100.00000007 100.00000001 100.    100.
  100.    100.    100.    100.    100. ]

[100.00000004 100.00000004 100.00000184 100.00000874 100.00003584
  100.00011519 100.00023515 100.00011519 100.00003584 100.00000874
  100.00000183 100.00000034 100.00000006 100.00000001 100.
  100.    100.    100.    100.    100. ]
[100.00000172 100.00000172 100.00000878 100.00004684 100.00021835
  100.00081274 100.00194245 100.00081274 100.00021835 100.00004683
  100.00000874 100.00000148 100.00000024 100.00000004 100.00000001
  100.    100.    100.    100.    100. ]

[100.00000623 100.00000623 100.00003595 100.00021837 100.0011847
  100.00528748 100.01568109 100.00528748 100.0011847 100.00021835
  100.00003584 100.00000545 100.00000078 100.00000011 100.00000001
  100.    100.    100.    100.    100. ]

[100.00001752 100.00001752 100.00011545 100.00081277 100.00528748
  100.02980551 100.12145551 100.02980551 100.00528748 100.00081274
  100.00011519 100.00001551 100.00000202 100.00000026 100.00000003
  100.    100.    100.    100.    100. ]

[100.00003141 100.00003141 100.00023554 100.00194249 100.01568109

```

100.12145551 100.86642243 100.12145551 100.01568109 100.00194245  
 100.00023515 100.00002808 100.00000333 100.00000039 100.00000005  
 100.00000001 100. 100. 100. 100. ]

[100.00001752 100.00001752 100.00011545 100.00081277 100.00528748  
 100.02980551 100.12145551 100.02980551 100.00528748 100.00081274  
 100.00011519 100.00001551 100.00000202 100.00000026 100.00000003  
 100. 100. 100. 100. 100. ]

[100.00000623 100.00000623 100.00003595 100.00021837 100.0011847  
 100.00528748 100.01568109 100.00528748 100.0011847 100.00021835  
 100.00003584 100.00000545 100.00000078 100.00000011 100.00000001  
 100. 100. 100. 100. 100. ]

[100.00000172 100.00000172 100.00000878 100.00004684 100.00021835  
 100.00081274 100.00194245 100.00081274 100.00021835 100.00004683  
 100.00000874 100.00000148 100.00000024 100.00000004 100.00000001  
 100. 100. 100. 100. 100. ]

[100.00000004 100.00000004 100.00000184 100.00000874 100.00003584  
 100.00011519 100.00023515 100.00011519 100.00003584 100.00000874  
 100.00000183 100.00000034 100.00000006 100.00000001 100.  
 100. 100. 100. 100. 100. ]

[100.00000008 100.00000008 100.00000035 100.00000149 100.00000545  
 100.00001551 100.00002808 100.00001551 100.00000545 100.00000148  
 100.00000034 100.00000007 100.00000001 100. 100.  
 100. 100. 100. 100. 100. ]

[100.00000002 100.00000002 100.00000006 100.00000024 100.00000078  
 100.00000202 100.00000333 100.00000202 100.00000078 100.00000024  
 100.00000006 100.00000001 100. 100. 100.  
 100. 100. 100. 100. 100. ]

[100. 100. 100.00000001 100.00000004 100.00000011  
 100.00000026 100.00000039 100.00000026 100.00000011 100.00000004  
 100.00000001 100. 100. 100. 100.  
 100. 100. 100. 100. 100. ]

[100. 100. 100. 100.00000001 100.00000002  
 100.00000004 100.00000005 100.00000004 100.00000002 100.00000001  
 100. 100. 100. 100. 100.  
 100. 100. 100. 100. 100. ]

[100. 100. 100. 100.00000001 100.00000002  
 100.00000004 100.00000005 100.00000004 100.00000002 100.00000001  
 100. 100. 100. 100. 100.  
 100. 100. 100. 100. 100. ]]

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

[[100.00001256 100.00001256 100.00002055 100.00003696 100.00005999  
 100.0000823 100.00009201 100.00008226 100.00005983 100.00003647  
 100.00001915 100.00000887 100.00000369 100.0000014 100.00000049  
 100.00000016 100.00000005 100.00000001 100.00000001 100.00000001]

[100.00001256 100.00001256 100.00002055 100.00003696 100.00005999  
100.0000823 100.00009201 100.00008226 100.00005983 100.00003647  
100.00001915 100.00000887 100.00000369 100.00000014 100.00000049  
100.00000016 100.00000005 100.00000001 100.00000001 100.00000001]

[100.00003426 100.00003426 100.00005811 100.00010835 100.00018131  
100.00025445 100.00028723 100.00025434 100.00018092 100.00010712  
100.00005449 100.00002443 100.00000984 100.00000362 100.00000123  
100.00000039 100.00000012 100.00000003 100.00000001 100.00000001]

[100.00010736 100.00010736 100.00018874 100.00036489 100.00062997  
100.00090592 100.00103403 100.00090563 100.00062888 100.00036135  
100.000178 100.00007723 100.00003013 100.00001074 100.00000354  
100.00000109 100.00000032 100.00000009 100.00000003 100.00000003]

[100.00031779 100.00031779 100.00058024 100.00116598 100.00208425  
100.00308468 100.00357144 100.00308392 100.00208139 100.00115636  
100.00055016 100.0002307 100.00008709 100.00003008 100.00000962  
100.00000288 100.00000081 100.00000022 100.00000007 100.00000007]

[100.00086342 100.00086342 100.00164325 100.00344708 100.0064174  
100.00984533 100.01162914 100.00984354 100.00641047 100.00342316  
100.00156614 100.00063275 100.00023067 100.00007711 100.00002393  
100.00000696 100.00000191 100.00000051 100.00000015 100.00000015]  
[100.00211623 100.00211623 100.00421584 100.00928352 100.01814695  
100.02917863 100.03552906 100.02917484 100.0181319 100.00923007  
100.0040381 100.00156613 100.00055011 100.00017775 100.00005346  
100.00001511 100.00000405 100.00000104 100.00000031 100.00000031]

[100.00457938 100.00457938 100.00959095 100.02232004 100.04632838  
100.07943564 100.10175513 100.07942855 100.04629956 100.02221481  
100.00923006 100.00342313 100.00115626 100.0003609 100.00010525  
100.00002892 100.00000755 100.00000019 100.00000055 100.00000055]

[100.00849162 100.00849162 100.01876002 100.04647741 100.10399948  
100.1955481 100.27510245 100.19553664 100.10395196 100.04629949  
100.01813185 100.00641041 100.00208122 100.00062818 100.00017796  
100.00004768 100.00001216 100.000003 100.00000085 100.00000085]

[100.01292704 100.01292704 100.03007927 100.07967847 100.19560171  
100.42173994 100.71937672 100.42172437 100.19553615 100.07942814  
100.02917459 100.00984338 100.00308366 100.00090469 100.00025037  
100.00006577 100.0000165 100.00000401 100.00000112 100.00000112]

[100.01519959 100.01519959 100.03656096 100.10203607 100.27517314  
100.71939109 101.92250734 100.71937376 100.2750997 100.10175329  
100.03552803 100.01162859 100.003571 100.00103294 100.00028282  
100.00007367 100.00001835 100.00000443 100.00000123 100.00000123]

[100.01292704 100.01292704 100.03007927 100.07967847 100.19560171  
100.42173994 100.71937672 100.42172437 100.19553615 100.07942814  
100.02917459 100.00984338 100.00308366 100.00090469 100.00025037  
100.00006577 100.0000165 100.00000401 100.00000112 100.00000112]

[100.00849162 100.00849162 100.01876002 100.04647742 100.10399949  
100.19554811 100.27510246 100.19553664 100.10395197 100.0462995  
100.01813185 100.00641041 100.00208122 100.00062818 100.00017796  
100.00004768 100.00001216 100.000003 100.00000085 100.00000085]

[100.00457939 100.00457939 100.00959097 100.02232006 100.04632842  
100.07943569 100.10175517 100.07942859 100.0462996 100.02221484  
100.00923007 100.00342313 100.00115626 100.0003609 100.00010525  
100.00002892 100.00000755 100.0000019 100.00000055 100.00000055]

[100.00211627 100.00211627 100.0042159 100.00928362 100.01814711  
100.02917883 100.03552927 100.02917504 100.01813205 100.00923017  
100.00403815 100.00156616 100.00055012 100.00017775 100.00005347  
100.00001511 100.00000405 100.00000104 100.00000031 100.00000031]

[100.00086359 100.00086359 100.00164351 100.00344751 100.00641806  
100.0098462 100.0116301 100.00984442 100.00641113 100.00342358  
100.00156638 100.00063286 100.00023073 100.00007713 100.00002393  
100.00000696 100.00000191 100.00000051 100.00000015 100.00000015]

[100.00031847 100.00031847 100.00058129 100.00116779 100.00208707  
100.00308844 100.0035756 100.00308768 100.0020842 100.00115814  
100.00055114 100.00023117 100.0000873 100.00003016 100.00000965  
100.00000289 100.00000082 100.00000022 100.00000007 100.00000007]

[100.0001099 100.0001099 100.00019282 100.00037209 100.00064146  
100.0009215 100.00105136 100.0009212 100.00064034 100.00036845  
100.00018179 100.00007902 100.00003089 100.00001103 100.00000364  
100.00000112 100.00000033 100.00000009 100.00000003 100.00000003]

[100.00004338 100.00004338 100.00007319 100.00013574 100.00022613  
100.00031629 100.00035653 100.00031615 100.00022563 100.00013417  
100.00006857 100.00003088 100.0000125 100.00000462 100.00000158  
100.0000005 100.00000015 100.00000004 100.00000001 100.00000001]

[100.00004338 100.00004338 100.00007319 100.00013574 100.00022613  
100.00031629 100.00035653 100.00031615 100.00022563 100.00013417  
100.00006857 100.00003088 100.0000125 100.00000462 100.00000158  
100.0000005 100.00000015 100.00000004 100.00000001 100.00000001]]

Значення  $u$  у вузлах сітки на 8 хвилині для сітки 20 на 20 з кроком  $\Delta t = 20c$ :

[[100.00023996 100.00023996 100.00032414 100.00047974 100.00067118  
100.00083503 100.00090018 100.00083286 100.00066474 100.00046358  
100.00028664 100.00015935 100.00008061 100.00003751 100.00001619  
100.00000654 100.00000249 100.00000092 100.0000004 100.0000004 ]

[100.00023996 100.00023996 100.00032414 100.00047974 100.00067118  
100.00083503 100.00090018 100.00083286 100.00066474 100.00046358  
100.00028664 100.00015935 100.00008061 100.00003751 100.00001619  
100.00000654 100.00000249 100.00000092 100.0000004 100.0000004 ]

[100.00049627 100.00049627 100.00068606 100.0010424 100.00149036  
100.00188243 100.00204125 100.00187834 100.00147808 100.00101093



100.00061142 100.00033212 100.00016416 100.00007465 100.00003152  
100.00001246 100.00000465 100.00000169 100.00000072 100.00000072]

[100.00120235 100.00120235 100.00170422 100.0026638 100.00390111  
100.005014 100.00547585 100.00500505 100.00387375 100.00259219  
100.00153048 100.00081109 100.00039127 100.00017377 100.00007172  
100.00002772 100.00001013 100.00000359 100.00000151 100.00000151]

[100.00280475 100.00280475 100.00408303 100.00658063 100.00990267  
100.01299805 100.01432777 100.01297913 100.00984392 100.0064237  
100.00369379 100.00190731 100.00089746 100.0003893 100.00015714  
100.00005949 100.0000213 100.00000741 100.00000304 100.00000304]

[100.00606698 100.00606698 100.00910156 100.01519387 100.02362983  
100.03188316 100.03562686 100.03184618 100.02351313 100.01487586  
100.00829484 100.00416083 100.00190619 100.00080683 100.00031841  
100.00011806 100.00004146 100.00001415 100.0000057 100.0000057 ]

[100.01198574 100.01198574 100.01860725 100.0323684 100.05246647  
100.0735495 100.08402196 100.07348366 100.05225541 100.03178162  
100.01708444 100.00829424 100.00369156 100.00152299 100.00058745  
100.00021335 100.00007352 100.00002464 100.00000975 100.00000975]

[100.02129459 100.02129459 100.03436123 100.06273937 100.10724711  
100.15893624 100.18894934 100.15883079 100.10690406 100.06176716  
100.03178102 100.01487459 100.00642009 100.00258049 100.0009732  
100.00034656 100.00011733 100.00003868 100.00001505 100.00001505]

[100.03334898 100.03334898 100.05611014 100.10832571 100.19833092  
100.31967472 100.41167418 100.31952482 100.19783695 100.10690187  
100.05225318 100.02351058 100.00983853 100.00385734 100.00142519  
100.00049877 100.00016633 100.00005408 100.00002077 100.00002077]

[100.04481132 100.04481132 100.07846001 100.16063191 100.32013504  
100.58660367 100.90119946 100.58641705 100.31951444 100.15882067  
100.07347617 100.03184033 100.01297114 100.00498429 100.00181287  
100.00062642 100.00020668 100.00006655 100.00002533 100.00002533]

[100.04992589 100.04992589 100.08944889 100.19088145 100.4122986  
100.90135663 102.12338046 100.90115553 100.41162751 100.18891325  
100.08399799 100.03561153 100.01431452 100.00545138 100.00196994  
100.0006773 100.00022257 100.00007142 100.00002709 100.00002709]

[100.04481136 100.04481136 100.07846006 100.16063198 100.32013512  
100.58660377 100.90119957 100.58641715 100.31951452 100.15882073  
100.07347621 100.03184036 100.01297115 100.0049843 100.00181288  
100.00062642 100.00020668 100.00006655 100.00002533 100.00002533]

[100.03334914 100.03334914 100.05611034 100.10832597 100.19833126  
100.31967513 100.4116746 100.31952522 100.19783728 100.10690212  
100.05225334 100.02351068 100.00983858 100.00385737 100.0014252  
100.00049877 100.00016634 100.00005408 100.00002077 100.00002077]  
[100.02129515 100.02129515 100.03436193 100.06274032 100.10724836  
100.15893773 100.18895093 100.15883228 100.10690529 100.06176807  
100.03178162 100.01487495 100.00642029 100.00258058 100.00097325

100.00034658 100.00011734 100.00003868 100.00001506 100.00001506]

[100.01198761 100.01198761 100.01860964 100.03237172 100.05247087  
100.0735548 100.0840276 100.07348894 100.05225975 100.03178479  
100.01708651 100.00829546 100.00369221 100.00152331 100.00058759  
100.00021342 100.00007354 100.00002465 100.00000975 100.00000975]

[100.00607307 100.00607307 100.00910943 100.01520498 100.02364476  
100.03190126 100.03564619 100.03186422 100.02352787 100.0148865  
100.00830169 100.00416481 100.0019083 100.00080785 100.00031888  
100.00011825 100.00004154 100.00001418 100.00000571 100.00000571]  
[100.00282365 100.00282365 100.00410787 100.00661626 100.00995129  
100.01305752 100.01439148 100.01303841 100.00989197 100.00645797  
100.00371554 100.00191974 100.00090393 100.0003924 100.00015852  
100.00006006 100.00002153 100.0000075 100.00000308 100.00000308]

[100.00125821 100.00125821 100.00177893 100.00277311 100.00405259  
100.00520122 100.00567717 100.00519175 100.00402368 100.00269765  
100.00159632 100.00084806 100.00041016 100.00018264 100.00007558  
100.0000293 100.00001073 100.00000382 100.00000161 100.00000161]

[100.00065273 100.00065273 100.00089941 100.00136146 100.00194039  
100.00244532 100.00264927 100.00243988 100.00192407 100.00131974  
100.00080074 100.00043636 100.00021638 100.0000987 100.00004179  
100.00001656 100.00000619 100.00000225 100.00000097 100.00000097]

[100.00065273 100.00065273 100.00089941 100.00136146 100.00194039  
100.00244532 100.00264927 100.00243988 100.00192407 100.00131974  
100.00080074 100.00043636 100.00021638 100.0000987 100.00004179  
100.00001656 100.00000619 100.00000225 100.00000097 100.00000097]]