

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЇВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра інформатики факультету інформатики

ЧИСЕЛЬНЕ РОЗВ'ЯЗАННЯ РІВНЯННЯ ПЕРЕНОСУ МЕТОДОМ ХАРАКТЕРИСТИК. ЛІНІЙНА ІНТЕРПОЛЯЦІЯ.

**Текстова частина до курсової роботи
за спеціальністю „Комп'ютерні науки”**

Керівник курсової роботи
к.ф.-м.н, доц. Тригуб О.С.
(прізвище та ініціали)

(підпис)

“ ____ ” _____ 2021 р.

Виконала студентка Кураш Я.С.
(прізвище та ініціали)

“ ____ ” _____ 2021 р.

Київ 2021

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ
Зав.кафедри інформатики,
доцент, к.ф.-м.н.
_____ С. С. Гороховський
(підпис)
„____” _____ 2020 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на курсову роботу

студентці Кураш Ярославі факультету інформатики 3 курсу

ТЕМА: Чисельне розв'язання рівняння переносу методом характеристик.
Лінійна інтерполяція

Зміст ТЧ до курсової роботи:

Вступ

1 Теоретичні відомості

2 Опис реалізації методу характеристик з лінійною інтерполяцією

3 Висновки

4 Список літератури

5 Додатки (за необхідністю)

Дата видачі „____” _____ 2021 р. Керівник _____
(підпис)

Завдання отримала _____
(підпис)

Тема: Чисельне розв’язання рівняння переносу методом характеристик.
Лінійна інтерполяція.

Календарний план виконання роботи:

№ п/п	Назва етапу дипломного проекту (роботи)	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу.	25.10.2020	
2.	Огляд літератури за темою роботи.	28.12.2020	
3.	Виконання аналізу актуальності теми	18.01.2021	
4.	Розробка алгоритму розв’язання рівняння	01.02.2021	
4.	Програмування розробленого алгоритму	05.04.2021	
6.	Написання пояснювальної роботи.	03.05.2021	
7.	Попередня демонстрація проекту науковому керівнику	03.05.2021	
8.	Створення презентації	15.05.2021	
9.	Захист курсової роботи	3 24.05.2021	

Студент _____ Кураш Я.С. _____

Керівник _____ Тригуб О.С. _____

“ _____ ”

Зміст

Вступ	5
РОЗДІЛ 1: Теоретичні відомості	6
1.1 Чисельні й точні методи	6
1.2 Крайові задачі для диференціальних рівнянь	7
1.3 Методи сіток	10
1.4 Метод скінченних різниць	10
1.5 Різницева схема методу характеристик з лінійною інтерполяцією	12
1.6 Обчислювальні труднощі властиві класичним різницеvim схемам	14
РОЗДІЛ 2: Опис реалізації методу характеристик з лінійною інтерполяцією	16
2.1 Зведення рівняння переносу до системи алгебраїчних рівнянь	16
2.2 Розв'язання системи алгебраїчних рівнянь	17
2.3 Створення програмного додатку	18
2.4 Результат виконання програми	23
2.5 Порівняння результатів розрахунків із точним розв'язком та з іншими схемами	24
Висновки	26
Список використаної літератури	27
Додатки	28

Вступ

Однією з сучасних проблем галузі охорони навколишнього середовища залишається інтенсивне забруднення атмосфери, водних об'єктів та ґрунту різними антропогенними джерелами викидів. Прогнозування рівня забрудненості навколишнього середовища і оцінка можливого впливу на нього викидів від промислових підприємств, автомобільного транспорту та інших видів людської діяльності ґрунтується, як правило, на математичному моделюванні процесів переносу забруднень в повітряному і водному середовищах. Нині існує широкий спектр математичних моделей, що описують процеси переносу речовин в атмосфері, водоймах і підземних водах. Створені численні комп'ютерні програми, що реалізують ці моделі.

Для ряду простих випадків математичні задачі переносу допускають аналітичні рішення, які дають чітке уявлення про внесок різних механізмів переносу забруднень в просторовий розподіл концентрації забруднень.

Темою курсової роботи є дослідження чисельного методу розв'язання модельного рівняння переносу, що описує процес переносу ґрунтовими водами забруднювача, розчиненого в рідині, яка рухається зі швидкістю V при наявності дифузії з коефіцієнтом D .

Мета роботи — розглянути різницьову схему методу характеристик з лінійною інтерполяцією та її програмну реалізацію, а також порівняти результати розрахунків із точним розв'язком рівняння.

РОЗДІЛ 1: Теоретичні відомості

1.1 Чисельні й точні методи

Існує три групи методів розв'язання математичних задач:

1. Точні методи, які надають розв'язок у формі аналітичних виразів. Серед переваг точних методів є: можливість подати розв'язок у загальному вигляді, максимальна точність отриманого розв'язку, низькі вимоги до обсягу пам'яті, необхідної для зберігання розв'язку. Головним недоліком точних методів є неуніверсальність, оскільки лише невелику частину задач можна розв'язати точно.
2. Графічні методи, які знаходять розв'язок задачі візуально. Основною їх перевагою є наочність. Серед недоліків графічних методів є: трудомісткість обчислень, низька точність розв'язків, залежно від детальності побудови графіків, недостатня універсальність, оскільки графіки можна побудувати лише для задач невеликої розмірності.
3. Чисельні методи, що представляють розв'язок у вигляді набору чисел, які можна інтерпретувати різними і подавати за допомогою таблиць, діаграм, графіків. За допомогою чисельних методів розв'язування задачі зводиться до виконання скінченної кількості арифметичних і логічних дій над конкретними вхідними даними. Основний недолік - це трудомісткість обчислень. Перевагами чисельних методів є : універсальність, пристосованість для реалізації на комп'ютері.

1.2 Крайові задачі для диференціальних рівнянь

Щоб розібратися із методами розв'язку рівняння переносу необхідно розуміти основні поняття теорії диференціальних рівнянь: диференціальне рівняння, розв'язок диференціального рівняння, задача Коші, крайова задача

Диференціальні рівняння - це рівняння, що містять незалежні змінні, невідомі функції та їх диференціали (або похідні).

Звичайне диференціальне рівняння порядку n має загальний вигляд:

$$F(x, y, y', \dots, y^{(n-1)}, y^{(n)}) = 0 \quad (1)$$

Порядком диференціального рівняння називається найвищий порядок похідної, що входить до рівняння.

Якщо невідома функція, що входить у рівняння, залежить від однієї змінної, то рівняння називається звичайним диференціальним. Якщо ж невідома функція залежить від двох чи більше змінних, то рівняння називається диференціальним рівнянням із частинними похідними.

Розв'язком (інтегралом) диференціального рівняння називається функція $y(x)$, яка при підстановці в диференціальне рівняння перетворює його в тотожність.

Частковим розв'язком рівняння $F(x, y, y', \dots, y^{(n-1)}, y^{(n)}) = 0$

на інтервалі $[a, b]$ називають будь-яку n раз диференційовну функцію $y = \varphi(x)$, що задовольняє цьому рівнянню, тобто таку, що перетворює рівняння на цьому інтервалі в тотожність.

Загальним розв'язком (загальним інтегралом) рівняння

$F(x, y, y', \dots, y^{(n-1)}, y^{(n)}) = 0$ називають таке співвідношення

$\Phi(x, y, C_1, \dots, C_n) = 0$ з якого може бути отриманий частковий розв'язок

$y = \varphi(x, y, C_1, \dots, C_n)$, де $C_1 = \text{const}, \dots, C_n = \text{const}$.

Існують два роди задач, пов'язаних з визначенням розв'язків

рівняння $F(x, y, y', \dots, y^{(n-1)}, y^{(n)}) = 0$:

1. Задача Коші, або задача з початковими даними.
2. Крайова задача або задача з граничними умовами.

Задача Коші: Знайти той розв'язок $y(x)$, диференціального рівняння, який при $x = x_0$ задовольняє умовам:

$$y(x_0) = y_0, y'(x_0) = y'_0, \dots, y^{(n-1)}(x_0) = y_0^{(n-1)}.$$

Характерним для задачі Коші є те, що умови задають в одній точці $x = x_0$.

Відмінність крайової задачі від задачі Коші (задачі з початковими умовами) полягає в тому, що розв'язок диференціального рівняння повинен задовольняти граничним умовам, що пов'язують значення шуканої функції

$$y'' = f(x, y, y')$$

більш ніж в одній точці із граничними умовами, заданими на кінцях відрізка $[a, b]$: $y(a) = y_0, y(b) = y_1$ – граничні умови першого роду.

Слід знайти такий розв'язок $y(x)$ на цьому відрізку, який набуває на кінцях відрізка значення y_0, y_1 . Якщо функція $f(x, y, y')$ лінійна за аргументами y та y' , то дана задача – лінійна крайова задача, а якщо ні, то – нелінійна.

Розглянемо диференціальні рівняння із частинними похідними другого порядку з двома незалежними змінними.

У загальному випадку таке рівняння має вигляд:

$F(x, y, u, u_x, u_y, u_{xx}, u_{xy}, u_{yy}) = 0$, де x, y – невідомі змінні, u – шукана функція, $u_x, u_y, u_{xx}, u_{xy}, u_{yy}$ – її перші й другі частинні похідні по аргументах x і y (для зручності запису похідних «штрихи» не пишуть).

Розв'язком рівняння $F(x, y, u, u_x, u_y, u_{xx}, u_{xy}, u_{yy}) = 0$ називають функцію $u = u(x, y)$ що перетворює це рівняння на тотожність.

Рівняння називають лінійним, якщо воно:

1. першого степеня відносно шуканої функції й усіх її похідних,

2. не містить добутоків похідних, тобто якщо це рівняння може бути

записане у вигляді : $A \frac{\partial^2 u}{\partial x^2} + 2B \frac{\partial^2 u}{\partial x \partial y} + C \frac{\partial^2 u}{\partial y^2} + a \frac{\partial u}{\partial x} + b \frac{\partial u}{\partial y} + cu = F(x, y)$.

(2)

Рівняння зі змінними коефіцієнтами :

коефіцієнти $A(x, y), B(x, y), C(x, y), a(x, y), b(x, y), c(x, y)$ залежать тільки від x і y .

Рівняння з постійними коефіцієнтами :

коефіцієнти A, B, C, a, b, c не залежать від x, y .

Нехай $D = AC - B^2$ – дискримінант рівняння (2).

Залежно від значення функції D лінійне диференціальне рівняння відносять у даній області до одного з наступних типів:

$D > 0$ – еліптичний тип,

$D = 0$ – параболічний тип,

$D < 0$ – гіперболічний тип,

D – не зберігає постійного знака – змішаний тип.

У своїй роботі ми розглядаємо параболічне рівняння переносу (конвекції-дифузії) з постійними коефіцієнтами, яке має вигляд:

$$\frac{\partial C}{\partial t} + V \frac{\partial C}{\partial x} = D \frac{\partial^2 C}{\partial x^2} \quad (3)$$

і описує зміну концентрації $C(x, t)$ за часом t мігранта, розчиненого в рідині, яка фільтрується зі швидкістю V за наявності дифузії, заданої коефіцієнтом D .

Для даного рівняння будемо розглядати крайову задачу:

$$0 < x < L, 0 < t \leq T,$$

$$C(0, t) = 1, C(L, t) = 0, C(x, 0) = 0, C(0, 0) = 1 \quad (4)$$

1.3 Методи сіток

Для вирішення крайових задач розроблено багато чисельних методів, які стали ефективнішими з появою комп'ютерів. Найбільш розповсюдженими з них є методи сіток.

Суть цих методів полягає в апроксимації початкової неперервної функції за допомогою множини наближених значень, обчислених у певних точках області визначення $[a, b]$, які називаються вузлами

$$a = x_0 < x_1 < \dots < x_{n-1} < x_n = b.$$

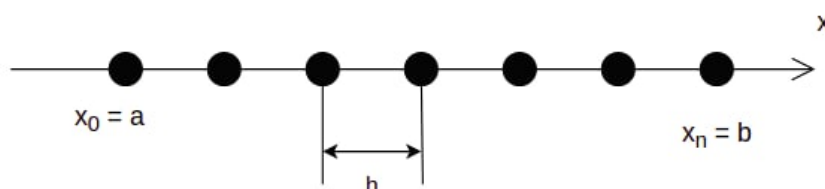


Рисунок 1 – Сітка для x , що належить проміжку $[a, b]$

Набір вузлів, з'єднаних певним чином, утворює сітку, яка є дискретною моделлю області визначення шуканої функції, а сукупність значень функції у вузлах сітки називають сітковою функцією.

У такий спосіб, за допомогою методів сіток диференціальну крайову задачу можна звести до системи лінійних або нелінійних алгебраїчних рівнянь відносно невідомих значень вузлової функції. Найбільш поширеними серед сучасних систем проектування є два методи сіток:

- метод скінченних різниць (МСР),
- метод скінченних елементів (МСЕ).

1.4 Метод скінченних різниць

Суть методу скінченних різниць полягає у зведенні диференціального рівняння до системи різницевих рівнянь відносно точок області визначення за допомогою заміни похідних диференціального рівняння їх різницевиими аналогами.

Алгоритм розв'язання диференціальної крайової задачі складається з наступних кроків:

- 1) розбити області визначення диференціальної задачі на вузлові точки й утворити сітку;
- 2) замінити похідні у рівнянні, яке описує задачу, їх різницевиими аналогами;
- 3) на основі отриманого різницевого рівняння побудувати систему різницевих рівнянь відносно всіх точок сітки;
- 4) розв'язати систему різницевих рівнянь, враховуючи граничні умови задачі;
- 5) представити результати розрахунків у зручному для аналізу вигляді.

Розглянемо сітку з рівномірним кроком h , $x_i - x_{i-1} = h$. Замість неперервної функції $y = f(x)$ розглядається сіткова функція $y_i = f(x_i)$.

Аналогічно проводиться дискретизація для функції двох змінних: $u = f(x, y)$. Розглядається сіткова функція $u_i = f(x_i, y_j)$, $i = 0, \dots, n$; $j = 0, \dots, m$, де крок по $x - h$, а по $y - k$.

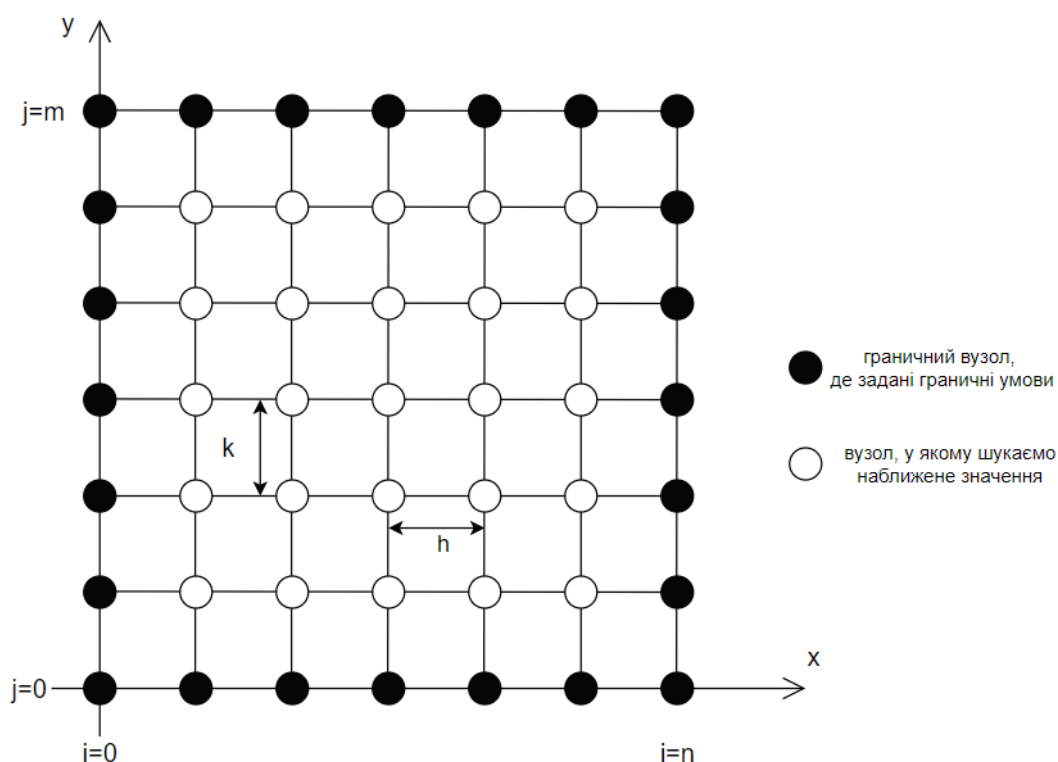


Рисунок 2 – Сітка для функції двох змінних $f(x, y)$

Користуючись цими позначеннями проілюструємо різницеві аналоги першого та другого порядків для рівняння в частинних похідних.

$$\left(\frac{\partial u}{\partial x}\right)_{ij} = \frac{u_{i+1,j} - u_{i-1,j}}{2h}$$

$$\left(\frac{\partial u}{\partial y}\right)_{ij} = \frac{u_{i,j+1} - u_{i,j-1}}{2k}$$

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_{ij} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2}$$

$$\left(\frac{\partial^2 u}{\partial y^2}\right)_{ij} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{k^2}$$

1.5 Різницева схема методу характеристик з лінійною інтерполяцією

Суть методу характеристик полягає у введенні в рівняння субстанційної похідної (характеристичного напрямку), яка описує швидкість зміни концентрації мігранта в рухомій точці простору. Після цього проводиться розщеплення під час обчислень розрахунку конвективного та дифузійного членів рівняння і кожен тип переносу розглядається окремо. Для цього розрахунок конвекції виконується на рухомих сітках(за допомогою методу Лагранжа), методом “слідкування за частинкою”, а дифузії – на нерухомих сітках (стандартним Ейлеровим методом). Щоб отримати неявні різницеві схеми сітки необхідно узгодити між собою таким чином, щоб їхні вузли в конкретний момент часу збігалися.

Розглянемо дану схему для параболічного рівняння (3), описаного раніше:

$$\frac{\partial C}{\partial t} + V \frac{\partial C}{\partial x} = D \frac{\partial^2 C}{\partial x^2}$$

Введемо субстанційну похідну, яка описує швидкість зміни концентрації мігранта в рухомій точці середовища:

$$\frac{\partial}{\partial \tau} = \frac{\partial}{\partial t} + V \frac{\partial}{\partial x} \quad (5)$$

Підставивши її в початкове рівняння отримаємо рівняння в Лагранжевій формі:

$$\frac{\partial C}{\partial \tau} = D \frac{\partial^2 C}{\partial x^2} \quad (6)$$

Побудуємо різницеву схему, що наближує отримане рівняння в момент часу $t^n = (n - 1) \tau$, $n = 1, 2, \dots, T$ на просторовій сітці $x_i = (i - 1)h$, $i = 1, 2, \dots, N$, де τ – крок часової сітки, а h – просторової. Скориставшись різницевиими аналогами похідних із попереднього пункту отримаємо різницеву схему

$$\frac{u_{i,n} - u_{n-1}(P)}{\tau} = D \frac{u_{i+1,n} - 2u_{i,n} + u_{i-1,n}}{h^2} \quad (7)$$

в якій $u_{n-1}(P)$ – значення шуканої функції в момент часу t^{n-1} в такій точці простору P , що в момент часу t^n потрапляє у вузол сітки x_i .

У такий спосіб, у вузол сітки x_i заноситься частинка, яка відстежується «назад» уздовж відповідної характеристики (звідси і назва методу) до попереднього часового шару t^{n-1} .

Для визначення концентрації у «відстежуваний» точці використовується інтерполяція за сусідніми вузлами сітки.

Використавши лінійну інтерполяцію для обчислення значення $u_{n-1}(P)$ у виразі (7), маємо співвідношення:

$$u_{n-1}(P) = \alpha u_{i-m-1,n-1} + (1 - \alpha)u_{i-m,n-1} \quad (8)$$

$$m = [Cu], Cu = \frac{V\tau}{h}, \quad \alpha = Cu - m, \quad 0 \leq \alpha \leq 1$$

де Cu – різницеве число Куранта, а квадратні дужки позначають цілу частину записаного в них дійсного числа.

1.6 Обчислювальні труднощі властиві класичним різницеvim схемам

Розглянемо використання схеми центральної різниці. Замінімо похідні в рівнянні (3) на їх різницеві аналоги таким чином:

$$\frac{\partial C}{\partial t} = \frac{u_{i,j+1} - u_{i,j}}{\tau}$$

$$\frac{\partial C}{\partial x} = \frac{u_{i+1,j+1} - u_{i-1,j+1}}{2h}$$

$$\frac{\partial^2 C}{\partial x^2} = \frac{u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}}{h^2}$$

Тоді матимемо рівняння в такому вигляді

$$\frac{u_{i,j+1} - u_{i,j}}{\tau} + V \frac{u_{i+1,j+1} - u_{i-1,j+1}}{2h} = D \frac{u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}}{h^2} \quad (9)$$

Зведемо подібні доданки й отримаємо таку різницеву схему:

$$-\left(\frac{D}{h^2} + \frac{V}{2h}\right) u_{i-1,j+1} + \left(\frac{2D}{h^2} + \frac{1}{\tau}\right) u_{i,j+1} - \left(\frac{D}{h^2} - \frac{V}{2h}\right) u_{i+1,j+1} = \frac{1}{\tau} u_{i,j} \quad (10)$$

Різницеві схеми поділяють на явні та неявні. Якщо на кожному кроці за часом різницевий рів'язок залежить лише від розв'язків із попередніх кроків, то схема називається явною. Якщо ж розв'язок зводиться до системи рівнянь відносно розв'язку на поточному кроці, то схема є неявною. Таким чином схема (10), отримана вище, є явною.

Оскільки кожен наступний часовий шар отримуємо шляхом апроксимації за значеннями з попереднього, то похибка зроблена під час обчислення деякого $u_{i,j}$ почне наростати з переходом до наступних часових шарів. Такий процес збільшення похибки називається нестійкістю різницевої схеми. А стійкою різницева схема називається тоді, коли зменшення h і τ не приводить до збільшення початкової похибки.

Різницеву схему називають абсолютно стійкою, якщо вона залишається стійкою за будь-якого закону зменшення h і τ .

Явні різницеві схеми є стійкими за певних умов, тому їх називають умовно стійкими. Вони вимагають використання маленьких кроків за часом.

Таким чином проведення точних розрахунків за однією із класичних різницевих схем можливо тільки при дотриманні жорстких обмежень на кроки сітки, коли величина $\frac{vh}{D} \ll 2$ чи принаймні $\frac{vh}{D} \leq 2$. Величину $\frac{vh}{D}$ називають числом Пекле і позначають Pe .

При порушенні даної умови у розв'язку спостерігається так звана 'схемна' дифузія та нефізичні сплески.

Оскільки задачі переносу зазвичай вимагають обчислень на великих розмірах області фільтрації та прогнозування на значні проміжки часу, використання класичних різницевих схем є мало ефективним.

РОЗДІЛ 2: Опис реалізації методу характеристик з лінійною інтерполяцією

2.1 Зведення рівняння переносу до системи алгебраїчних рівнянь

У різницькій схемі (7), отриманій у пункті 1.5

$$\frac{u_{i,n} - u_{n-1}(P)}{\tau} = D \frac{u_{i+1,n} - 2u_{i,n} + u_{i-1,n}}{h^2}$$

зведемо подібні доданки й перенесемо в ліву частину рівняння значення функції в момент часу t^n , а в праву – в момент часу t^{n-1} :

$$-\left(\frac{D}{h^2}\right) u_{i-1,n} + \left(\frac{2D}{h^2} + \frac{1}{\tau}\right) u_{i,n} - \left(\frac{D}{h^2}\right) u_{i+1,n} = \frac{1}{\tau} u_{n-1}(P) \quad (11)$$

Введемо позначення для коефіцієнтів при змінних $u_{i,j}$:

$$A(k) = \frac{D}{h^2} = B(k), \quad C(k) = A(k) + B(k) + \frac{1}{\tau}, \quad F(k) = \frac{1}{\tau} u_{n-1}(P) \quad (12)$$

Тоді для кожного часового шару на момент часу $t = t^n$ отримаємо систему алгебраїчних рівнянь у такому вигляді:

$$-A(k)u_{i-1,n} + C(k)u_{i,n} - B(k)u_{i+1,n} = F(k), \quad (13)$$

де $A(k)$, $C(k)$, $B(k)$ і $F(k)$ – піддіагональні, діагональні, наддіагональні елементи матриці та вектор правих частин відповідно, а k – номер вузла просторової сітки. Причому за умовою задачі для першого часового шару, де $t = 0$, концентрація мігранта в початковій точці $x = 0$ рівна одиниці, а в решті точок – нулю. Тобто $F(1) = 1$, $F(k) = 0$, $k = \overline{2, n1}$, де $n1$ – кількість вузлів сітки за простором. Також за умовою $C(0, t) = 1$, $C(L, t) = 0$, тоді рівняння (13) для точок $x = 0$ та $x = n1$ для кожного з часових шарів матиме вигляд $u_{0,n} = 1$ і $u_{n1,n} = 0$ відповідно. Таким чином $A(1) = 0$, $C(1) = 1$, $B(1) = 0$, $F(1) = 1$ і $A(n1) = 0$, $C(n1) = 1$, $B(n1) = 0$, $F(n1) = 0$.

2.2 Розв'язання системи алгебраїчних рівнянь

Для розв'язання системи алгебраїчних рівнянь скористаємось методом прогонки. За його допомогою можна знаходити розв'язки специфічних систем рівнянь, що мають не більше трьох невідомих у кожному рівнянні. Тоді, якщо представити систему у векторному вигляді $Ax = b$, то матриця A є трьохдіагональною.

$$A = \begin{pmatrix} C_1 & -B_1 & 0 & \dots & 0 & 0 \\ -A_2 & C_2 & -B_2 & \dots & 0 & 0 \\ & \vdots & & \ddots & \vdots & \\ & 0 & 0 & \dots & -A_{n-1} & C_{n-1} & -B_{n-1} \\ & 0 & 0 & & 0 & -A_n & C_n \end{pmatrix} \quad b = \begin{pmatrix} b_1 \\ b_1 \\ \vdots \\ b_{n-1} \\ b_n \end{pmatrix}$$

Залежність сусідніх розв'язків можна описати таким чином:

$x_{i-1} = \alpha_{i-1}x_i + \beta_{i-1}$, де α_k і β_k - якісь невідомі числа, які називають прогонковими коефіцієнтами. Якщо ми знайдемо їх значення та значення однієї змінної, то зможемо отримати й значення решти змінних.

Алгоритм складається із прямого й оберненого ходу.

Значення прогонкових коефіцієнтів знаходять під час прямого ходу за наступними формулами

$$\alpha_1 = \frac{B_1}{C_1}, \quad \alpha_i = \frac{B_i}{C_i - A_i \alpha_{i-1}}, \quad i = \overline{2, n}, \quad (14)$$

$$\beta_1 = \frac{b_1}{C_1}, \quad \beta_i = \frac{b_i + A_i \beta_{i-1}}{C_i - A_i \alpha_{i-1}}, \quad i = \overline{2, n} \quad (15)$$

Під час оберненого ходу знаходимо розв'язок системи рівнянь за допомогою наступних формул:

$$x_n = \beta_n, \quad x_i = \alpha_i x_{i+1} + \beta_i, \quad i = \overline{n-1, 1} \quad (16)$$

Для розв'язку рівняння, отриманого в попередньому пункті необхідно застосувати метод прогонки для кожного часового шару сітки.

Згадаємо постановку крайової задачі:

$$\frac{\partial C}{\partial t} + V \frac{\partial C}{\partial x} = D \frac{\partial^2 C}{\partial x^2}, \quad 0 < x < L, 0 < t \leq T,$$

$$C(0, t) = 1, \quad C(L, t) = 0, \quad C(x, 0) = 0, \quad C(0, 0) = 1$$

Виходячи з початкових умов, маємо значення сіткової функції для першого часового шару сітки, де $t = 0$, а також для кожного з наступних маємо значення сіткової функції для точки $x = 0$ та $x = L$.

Таким чином ітеративно, знаходимо значення сіткової функції для кожного часового шару, доки не дійдемо до останнього шару, для якого $t = T$, він і буде шуканим розв'язком задачі.

2.3 Створення програмного додатку

Я обрала мову програмування Java для програмної реалізації знаходження чисельного розв'язку рівняння переносу.

Для того, щоб наочно продемонструвати результати роботи програми я побудувала графіки на основі точного й чисельного розв'язків рівняння, використавши бібліотеку JFreeChart.

Структурно програма складається із двох пакетів: *graphic* і *thomasalgorithm*.

У пакеті *thomasalgorithm* знаходиться клас *ThomasAlgorithm*, що містить методи, які реалізують логіку розв'язання рівняння. Він містить у собі константи, які було обрано для розв'язку рівняння: $V = 1$ - швидкість фільтрації, $D = 0.1$ - коефіцієнт дифузії, $L = 10$ - довжина відрізка за простором, $T = 5$ - довжина відрізка за часом.

```
private static final double V = 1.0;
private static final double D = 0.1;
private static final double L = 10.0;
private static final double T = 5.0;
```

Метод *calculateExact(double pe)* приймає в якості аргументу число Пекле, та знаходить точний розв'язок рівняння, який має вигляд:

$$C(x, t) = 0.5 \left\{ \operatorname{erfc} \left[\frac{x - Vt}{2\sqrt{Dt}} \right] + \exp \left(\frac{Vx}{D} \right) \operatorname{erfc} \left[\frac{x + Vt}{2\sqrt{Dt}} \right] \right\}$$

Де *erfc* – доповнююча функція помилок, що обчислюється через функцію помилок таким чином:

$$\operatorname{erfc} x = 1 - \operatorname{erf} x = 1 - \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt.$$

Для обчислення значення доповнюючої функції помилок, експоненційної функції та квадратного кореня, я скористалась класами *Erf*, *Exp* та *Sqrt* із бібліотеки *Apache Commons Math*.

В методі проводиться обчислення величини кроку за просторовою сіткою за формулою $h = \frac{Pe D}{V}$ за загальної кількості кроків просторової сітки за формулою $n1 = \frac{L}{h} + 1$, а потім ітеративно для кожного із значень x проводиться обчислення значення функції в кожній із точок просторової сітки в момент часу $t = T$ та записується в результуючий масив u і в текстовий файл "output_exact.txt".

```
public static double[] calculateExact(double pe) {
    double h = pe * D / V;           // step on x
    int n1 = (int) (L / h) + 1;      // amount of steps on x
    double[] u = new double[n1 + 1]; // result array

    PrintWriter pw = null;
    try {
        pw = new PrintWriter("output_exact.txt");
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    }
}
```

```

double x; // current x
for (int i = 1; i <= n1; i++) {
    x = (i-1) * h;
    Sqrt sqrt = new Sqrt();
    Exp exp = new Exp();
    u[i] = 0.5 * (Erf.erfc((x - V * T) / (2 * sqrt.value(D * T))) +
        exp.value((V * x) / D) *
        Erf.erfc((x + V * T) / (2 * sqrt.value(D * T))));

    pw.printf(Locale.US, "x = %f  c = %f\n", x, u[i]);
}
pw.flush();
pw.close();
return u;
}

```

Метод *getXSet(double pe)* обчислює значення аргументу x у вузлах сітки із кроком h , обчисленим на основі числа Пекле.

```

public static double[] getXSet(double pe) {
    double h = pe * D / V; // step on x
    int n1 = (int) (L / h) + 1; // amount of steps on x

    double[] x = new double[n1+1];

    for (int i = 1; i <= n1; i++) {
        x[i] = (i-1) * h;
    }
    return x;
}

```

Метод *characteristicsScheme(double cu, double pe)* знаходить розв'язок задачі для заданих чисел Пекле й Куранта. На їх основі обчислюються кроки за часом і простором та кількість кроків у сітці.

Масиви A , B , C і F зберігають масиви коефіцієнтів рівняння для поточного часового шару. Їх значення встановлюються за умовами (12), отриманими в пункті 2.1:

```

// equation coefficients
double[] A = new double[n1 + 1];
double[] B = new double[n1 + 1];
double[] C = new double[n1 + 1];
double[] F = new double[n1 + 1];

// the result vector
double[] u = new double[n1 + 1];

double ab = D / (h * h);

```

```
double c = 2 * ab + 1 / tau;

// filling up matrix and vector of right sides
A[1] = 0;
B[1] = 0;
C[1] = 1;
F[1] = 1;
for (int i = 2; i <= n1 - 1; i++) {
    A[i] = ab;
    B[i] = ab;
    C[i] = c;
    F[i] = 0.;
}
A[n1] = 0;
B[n1] = 0;
C[n1] = 1;
F[n1] = 0;
```

Масиви *alpha* та *beta* зберігають значення прогонкових коефіцієнтів.

Частина коду, де реалізовано алгоритм методу прогонки послідовно для кожного з часових шарів:

```
// helping arrays of alphas and betas
double[] alpha = new double[n1];
double[] beta = new double[n1];

for (int j = 1; j < n2+1; j++) {
    // right stroke
    alpha[1] = B[1] / C[1];
    beta[1] = F[1] / C[1];

    for (int i = 2; i <= n1 - 1; i++) {
        double denominator = C[i] - A[i] * alpha[i - 1];
        alpha[i] = B[i] / denominator;
        beta[i] = (F[i] + A[i] * beta[i - 1]) / denominator;
    }
    // back stroke
    u[n1] = (F[n1] + A[n1] * beta[n1 - 1]) /
            (C[n1] - A[n1] * alpha[n1 - 1]);
    for (int i = n1 - 1; i >= 1; i--) {
        u[i] = alpha[i] * u[i + 1] + beta[i];
    }
    for (int i = n1 - 1; i > 1; i--) {
        F[i] = interpolate(u, cu, i)/tau;
    }
    F[1] = 1;
    F[n1] = 0;
}
```

Вектор правих частин, що зберігається в масиві F , для кожного наступного шару обчислюється на основі розв'язку з попереднього методом лінійної інтерполяції, який реалізовано в методі *interpolate(double[] u, double cu, int i)*.

```
private static double interpolate(double[] u, double cu, int i){
    int m = (int) Math.floor(cu);
    double alpha = cu - Math.floor(cu);

    return alpha * u[i-m-1] + (1-alpha)*u[i-m];
}
```

Результат обчислень записується в результуючий масив u і в текстовий файл "output_characteristics.txt".

Також я реалізувала метод *classicScheme(double cu, double pe)*, що здійснює пошук розв'язку задачі із застосуванням класичної різницевої схеми (10), описаної в пункті 1.6. Він відрізняється від розв'язку за схемою методу характеристик відсутністю інтерполяції та формулами для обчислення коефіцієнтів A , B і C :

```
double a = D / (h * h) + V / (2 * h);
double b = D / (h * h) - V / (2 * h);
double c = a + b + 1 / tau;

// filling up matrix and vector of right sides
A[1] = 0;
B[1] = 0;
C[1] = 1;
F[1] = 1;
for (int i = 2; i <= n1 - 1; i++) {
    A[i] = a;
    B[i] = b;
    C[i] = c;
    F[i] = 0.;
}
A[n1] = 0;
B[n1] = 0;
C[n1] = 1;
F[n1] = 0;
```

Вектор правих частин обчислюється як вектор розв'язку, поділений на крок за часом τ :

```
F[i] = u[i]/tau;
```

Результат обчислень записується в масив u і в текстовий файл "output_classic.txt".

2.4 Результат виконання програми

Рисунок 3 демонструє інтерфейс розробленої програми. Після запуску застосунку користувач бачить систему координат, на якій буде відображено графік, побудований на основі розв’язку задачі. Два поля, які визначають числа Куранта і Пекле повинні бути заповнені числовими значеннями. Також користувач має змогу додати чи прибрати графік, побудований за допомогою класичної різницевої схеми за допомогою прапорця “With classic scheme” та запустити побудову графіка, натиснувши кнопку “Draw the graphic”.

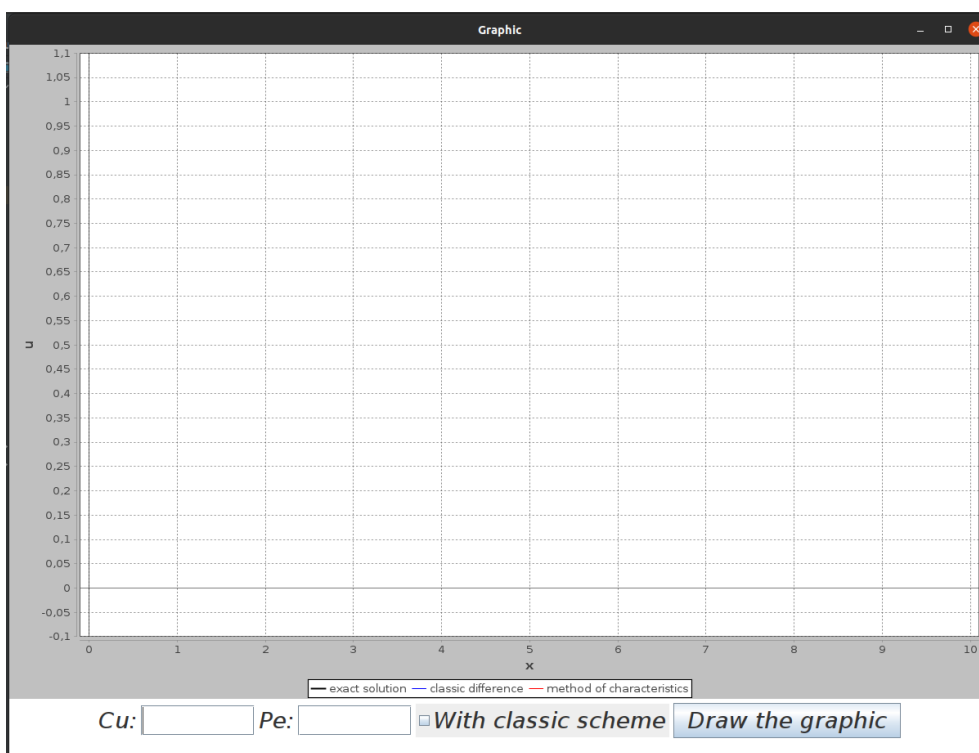


Рисунок 3 – Інтерфейс користувача програми

Після цього на системі координат з’явиться графік чорного кольору, побудований із використанням точного розв’язком, червоний - із використанням методу характеристик та синій - із використанням класичної різницевої схеми.

У разі введення символів, відмінних від цифр, чи відсутності значень полів користувач отримає повідомлення (Рисунок 4) з проханням ввести числа.



Рисунок 4 – Вікно з попередженням про вимоги до вхідних параметрів

2.5 Порівняння результатів розрахунків із точним розв’язком та з іншими схемами

Розрахунки було проведено для таких наборів чисел Пекле й Куранта

Pe	Cu
0.3	0.3
0.3	0.7
0.3	1
0.3	3
0.7	1
0.7	3
1.2	0.3
1.2	0.7
1.2	1
1.2	5
1.2	10
3	0.3
3	0.7
3	1

Проаналізувавши побудовані графіки, бачимо що вони узгоджуються із властивостями обох схем, описаними у пунктах 1.5 та 1.6, а саме:

- схема характеристик з лінійною інтерполяцією дозволяє збільшити крок часової сітки, не зменшуючи точності розв'язку;
- у різницевій схемі вплив “схемної” дифузії починає помітно роявлятися при $Pe \geq 2$;
- у методі характеристик схемна дифузія зменшується зі збільшенням числа Куранта, а при цілих значення - відсутня, тобто схема набуває третього порядку точності. Це пояснюється тим, що $P = x - Vt$ і потреба в інтерполяції відпадає;
- при малих числах Куранта схема характеристик має приблизно таку ж точність як і різницева чи навіть поступається їй, однак при $Cu > 1$ схема характеристик значно випереджає різницеву;
- у схемі характеристик точність розв'язку підвищується для таких пар значень чисел Куранта й Пекле, що $Cu \approx Pe$.

Висновки

Існує багато схем, що застосовуються для розв'язування рівняння переносу. Кожен із них має певні недоліки, що виражаються у появі похибок різної природи. Для ефективного моделювання процесів переносу необхідно знизити їх вплив на точність розв'язку, при цьому збільшивши крок за часом, для підвищення ефективності обчислень.

Було розглянуто дві схеми розв'язування: класичну різницеву схему та схему характеристик із лінійною інтерполяцією. Класичні різницеві схеми значно обмежують величину кроку за часом, оскільки лише для детальних просторової та часової сітки можливе отримання розв'язку з мінімальним впливом похибок, а саме “схемної” дифузії та нефізичних сплесків.

Метод характеристик з лінійною інтерполяцією дійсно має перевагу над різницевими методами й забезпечує високу точність розв'язків при доволі великих кроках за часом. Крім того, похибка мінімізується при цілих числах Куранта. Найкращих результатів обчислення можна досягти при таких числах Куранта й Пекле, що сиприбл= ре.

В результаті виконання роботи було розроблено програму, що реалізує класичну різницеву схему, схему характеристик з лінійною інтерполяцією для розв'язання рівняння переносу та обчислення точного розв'язку. Застосунок подає результати розрахунків у вигляді графіків, що дає змогу легко порівняти точність розрахунків.

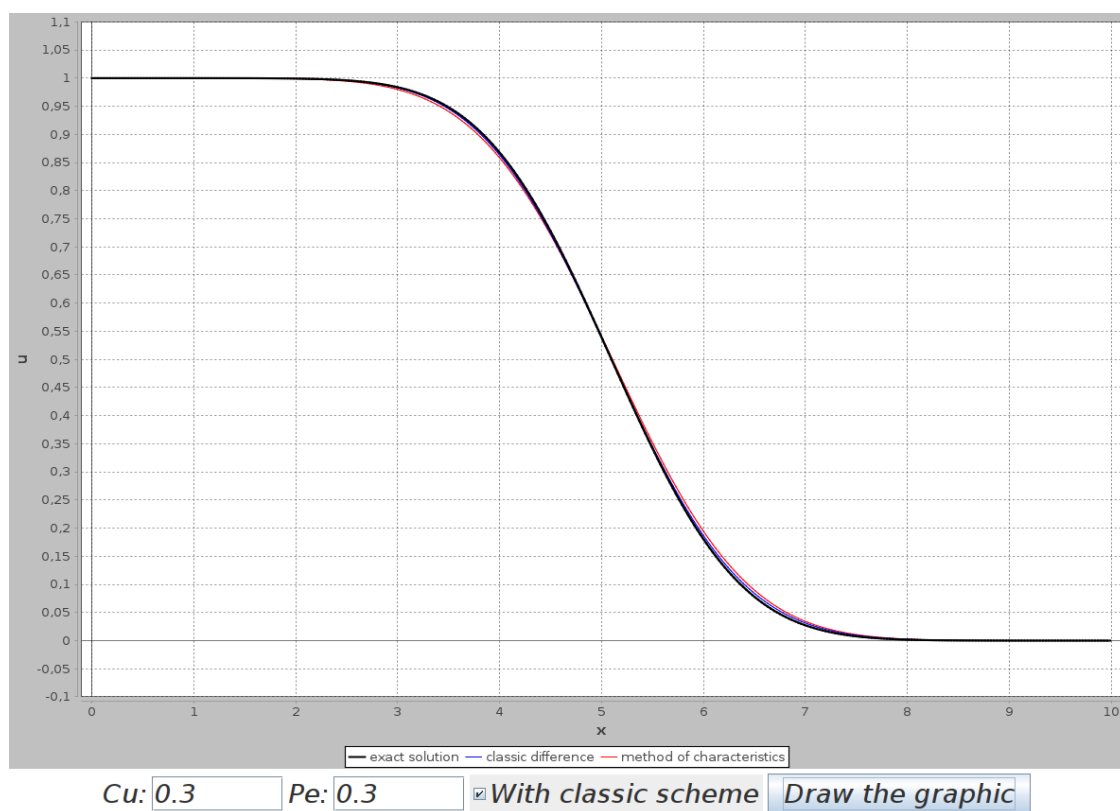
Таким чином на основі результатів виконання програми для різних наборів чисел Куранта й Пекле можна зробити висновок про правдивість теоретичних даних, викладених у першому розділі.

В майбутньому дослідження можна розширити, розглянувши інші схеми розв'язування рівняння переносу та використавши такі інструменти оцінки їх якості, як обчислення квадратичної похибки та метод диференціальних наближень.

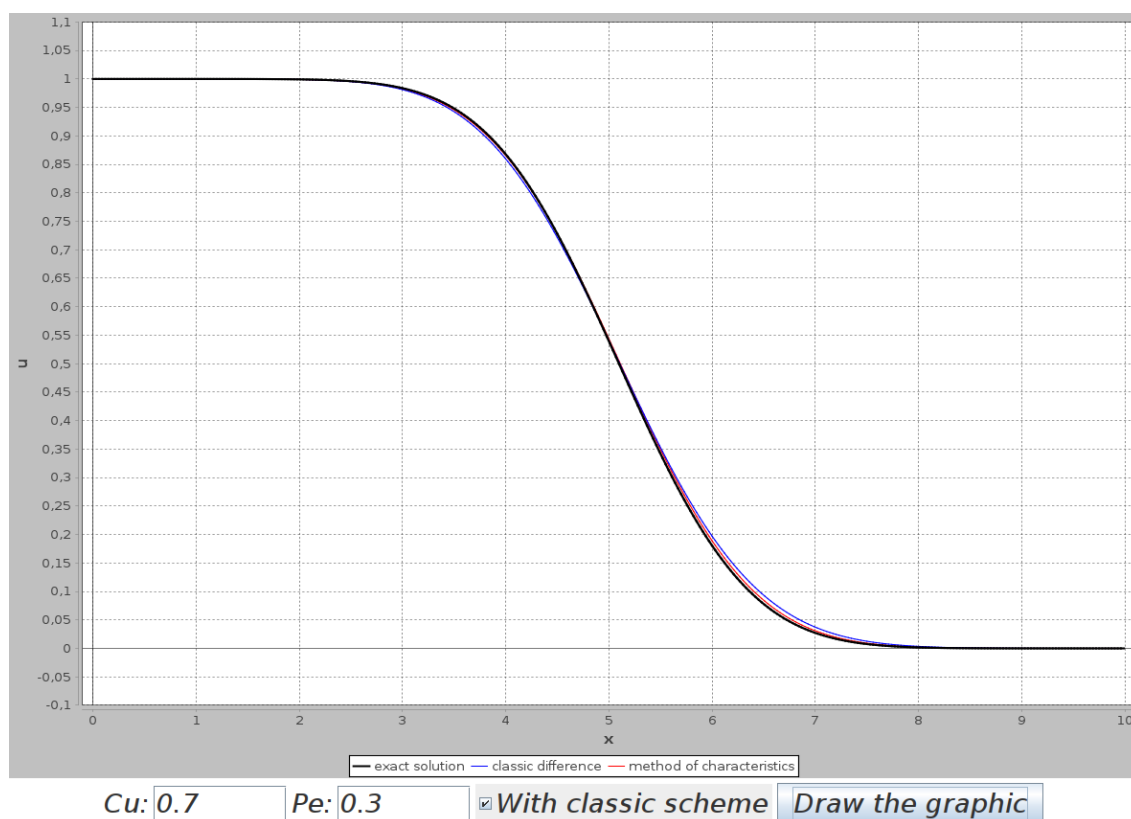
Список використаної літератури

1. Супруненко О.О. Чисельні методи в інформатиці. Курс лекцій: для студентів, які навчаються за напрямом підготовки 050101 „Комп’ютерні науки”, 050103 „Програмна інженерія”. – Черкаси: ЧНУ, 2009. – 130 с.
2. Козуб Г. О., Смагіна О. О. Чисельні методи : метод. рек. до виконання лабораторних робіт для студ. спец. 6.050103 „Програмна інженерія”. – Луганськ : ДЗ „ЛНУ імені Тараса Шевченка”, 2012. – 111 с.
3. Тригуб О.С. Аналіз застосування методу сіток до чисельного розв’язання параболічних рівнянь // Наукові записки. - 2003. - Том 21. Комп’ютерні науки. - С. 22-33.
4. Тригуб О.С. Порівняльний аналіз деяких різницевих схем для рівняння конвективної дифузії // Наукові записки. - 1999. - Том 16. Комп’ютерні науки. - С. 61-71.
5. Стеля О. Б., Тригуб О. С., Ляшко В. І. Аналіз застосування інтерполяційних процедур при реалізації методу характеристик // Наукові записки НАУКМА. - 2012. Том 138. С. 85-90.
6. Douglas J.Jr., Russel T.F. Numerical Methods for Convection Dominated Diffusion Problems Based on Combining the Method of Characteristics with Finite Element or Finite Difference Procedures. SIAM J. Number. Anal. - 1982. - №5. - P.871-885.

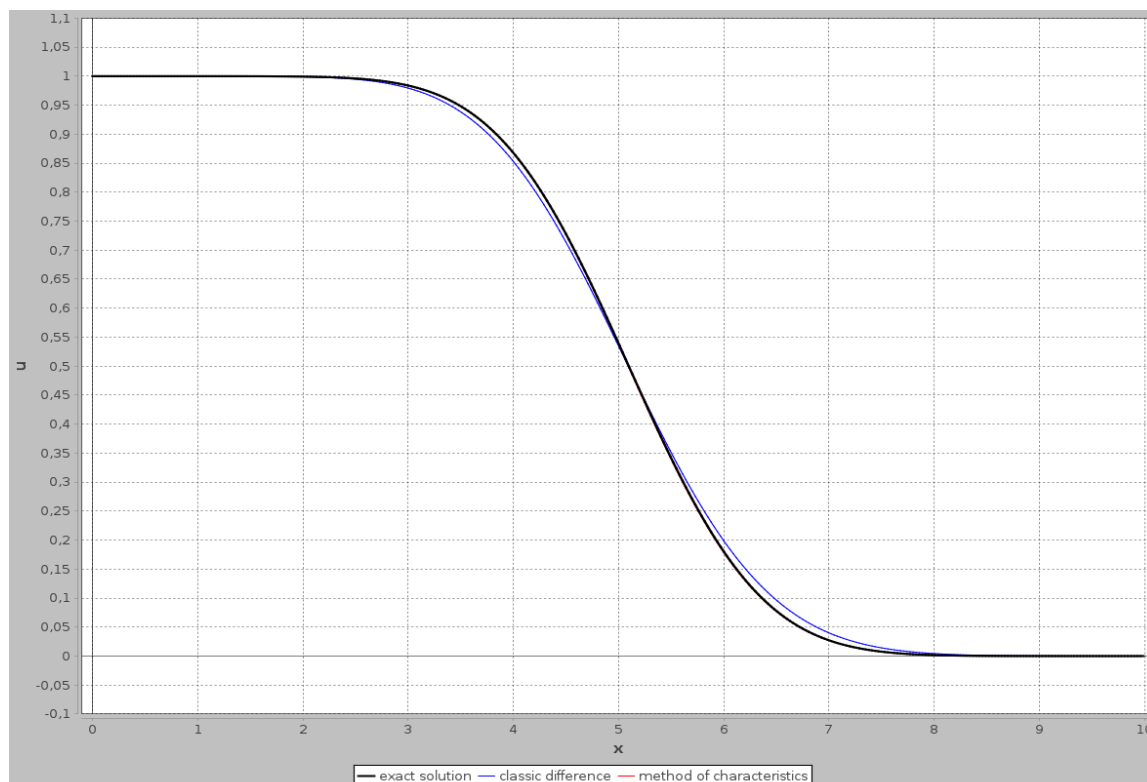
Додатки



Додаток 1



Додаток 2

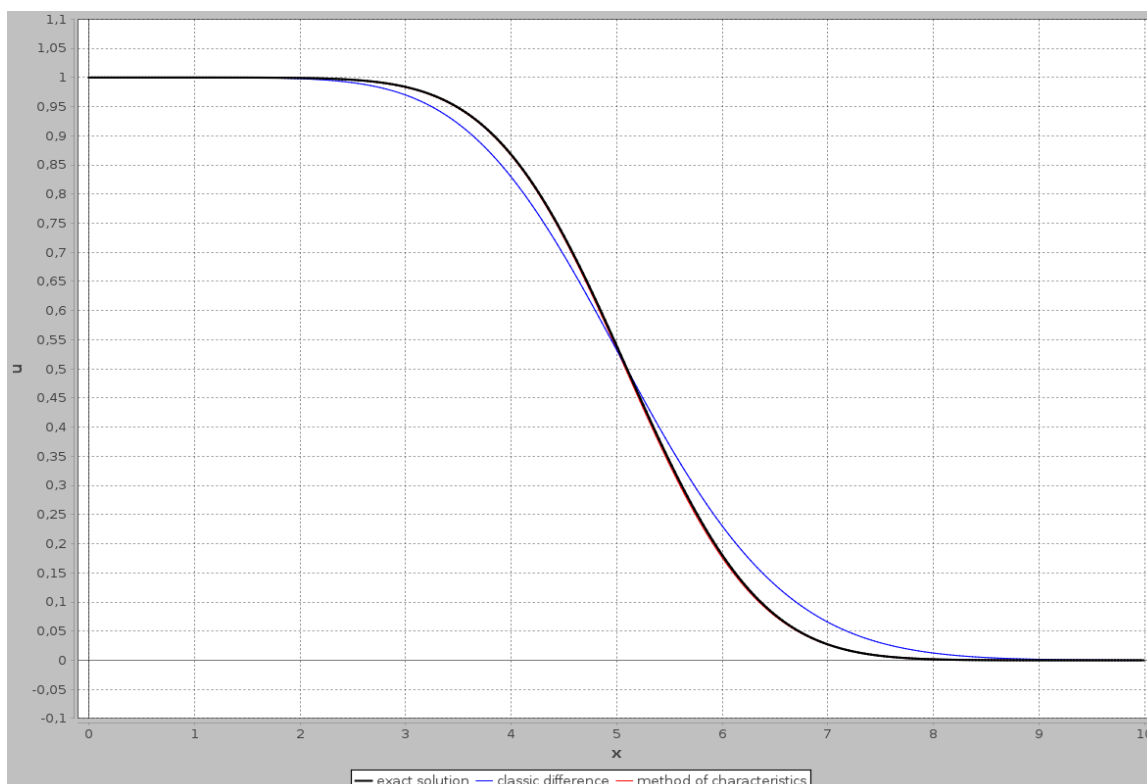


Cu: 1

Pe: 0.3

☒ With classic scheme[Draw the graphic](#)

Додаток 3

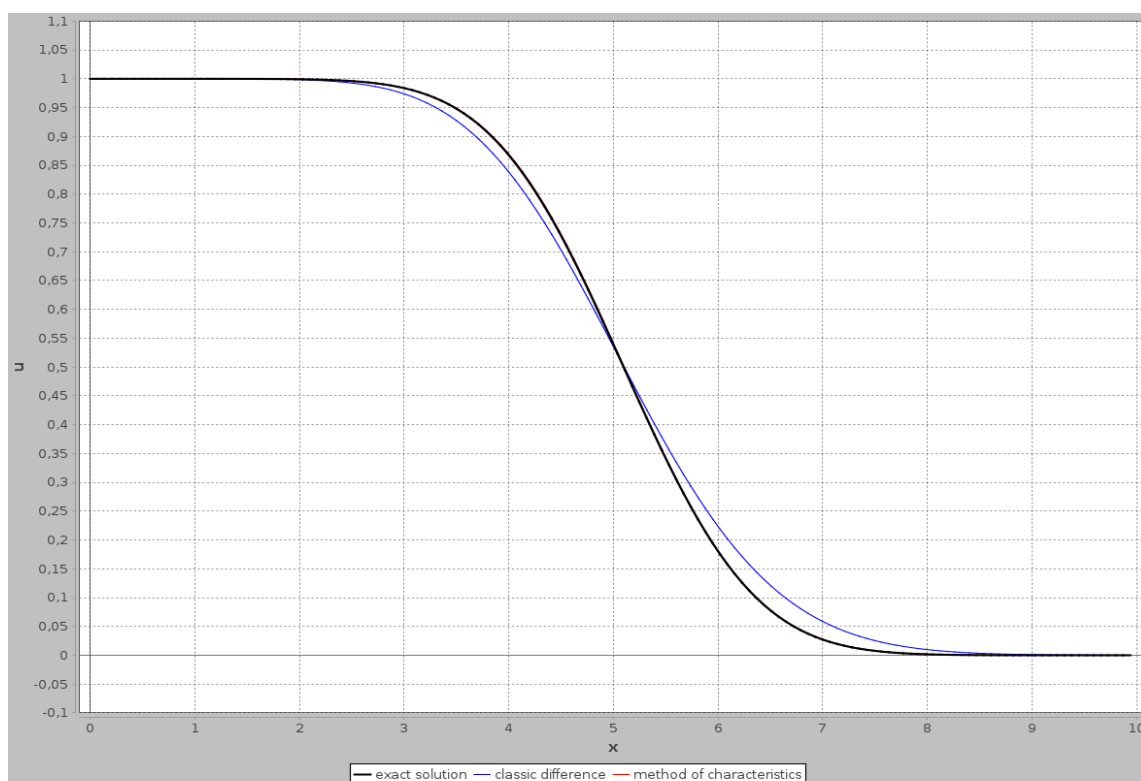


Cu: 3

Pe: 0.3

☒ With classic scheme[Draw the graphic](#)

Додаток 4

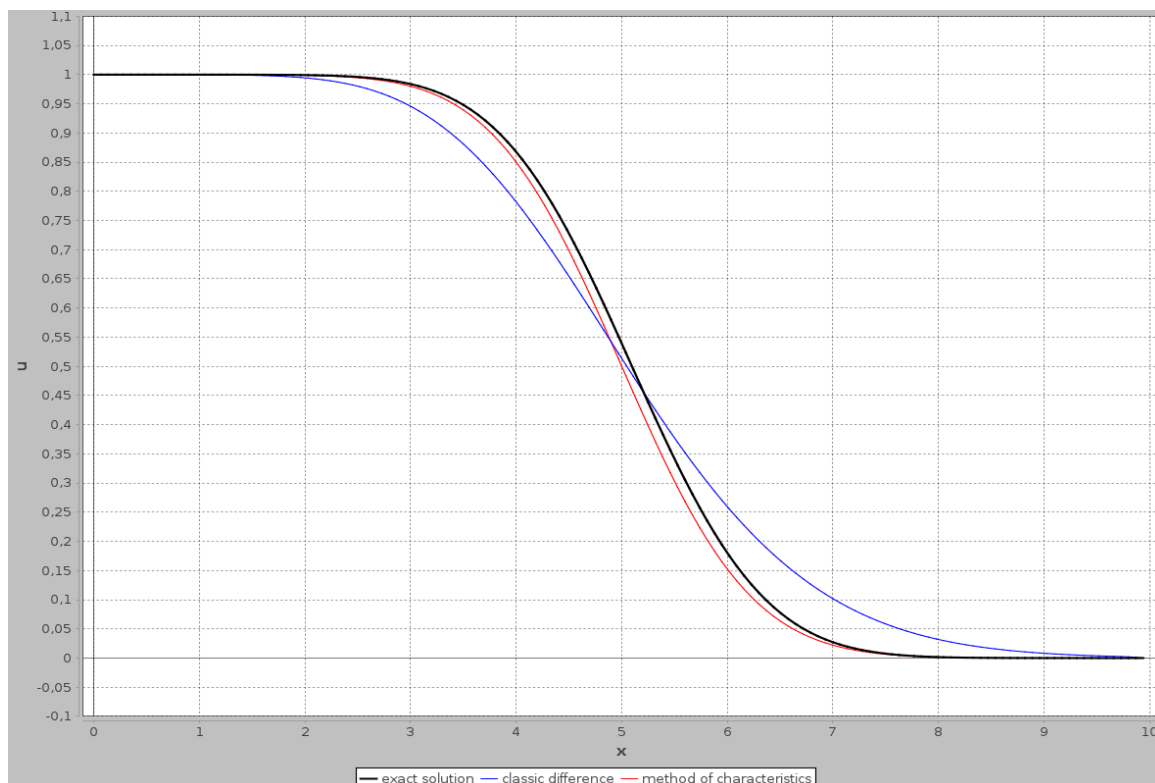


Cu: 1

Pe: 0.7

☒ With classic scheme[Draw the graphic](#)

Додаток 5

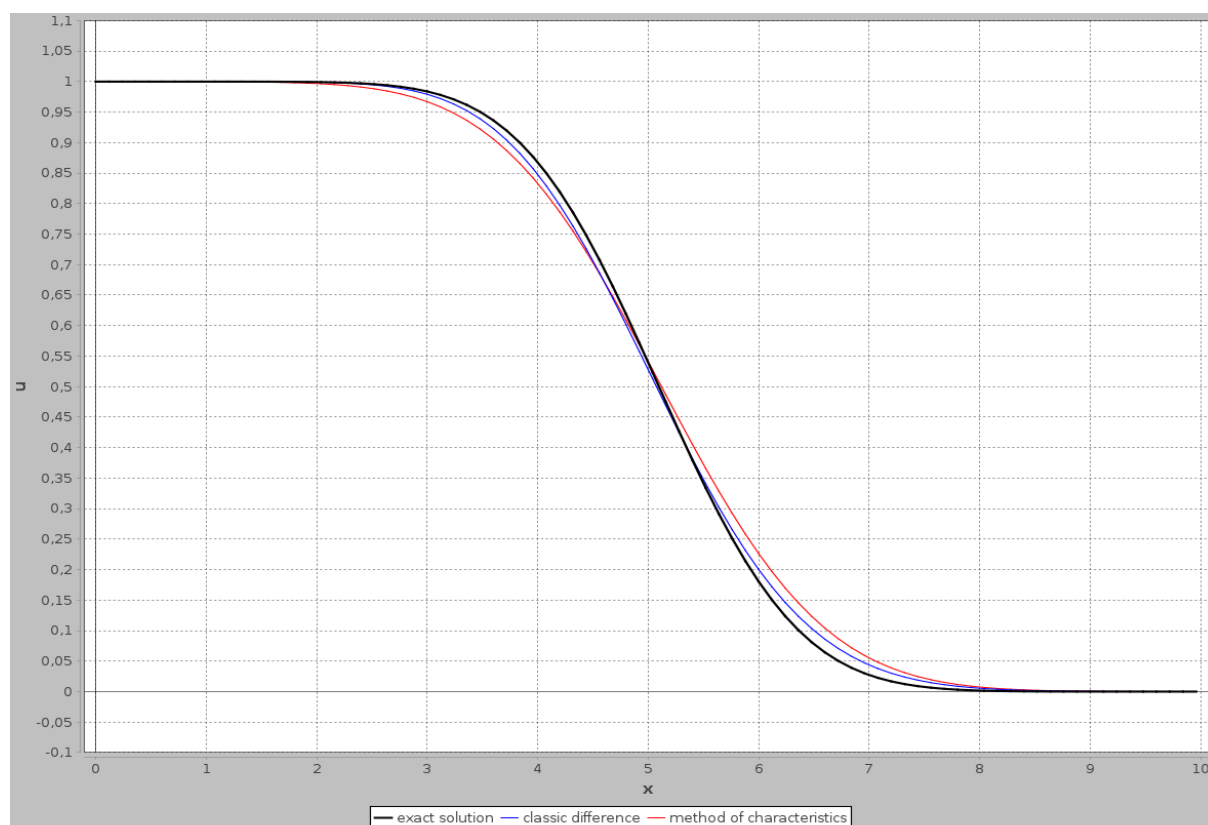


Cu: 3

Pe: 0.7

☒ With classic scheme[Draw the graphic](#)

Додаток 6

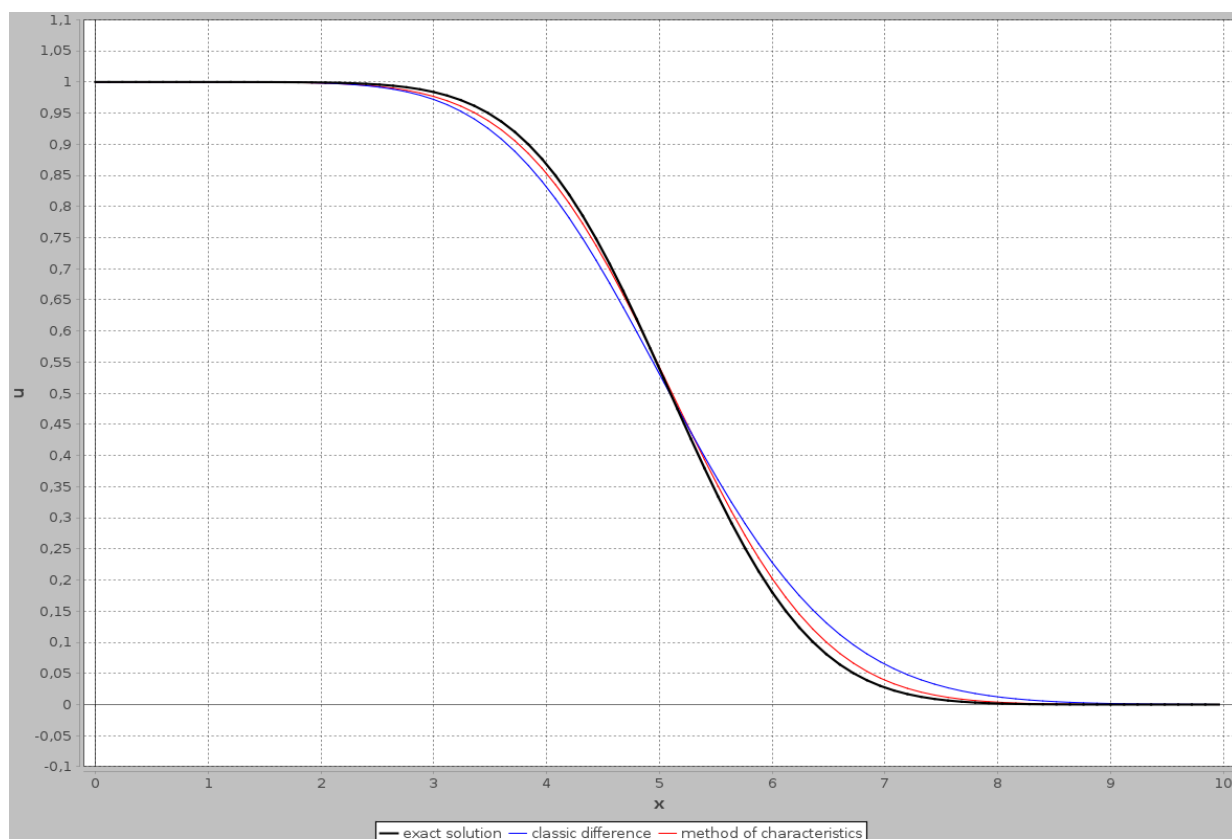


Cu: 0.3

Pe: 1.2

☒ With classic scheme[Draw the graphic](#)

Додаток 7

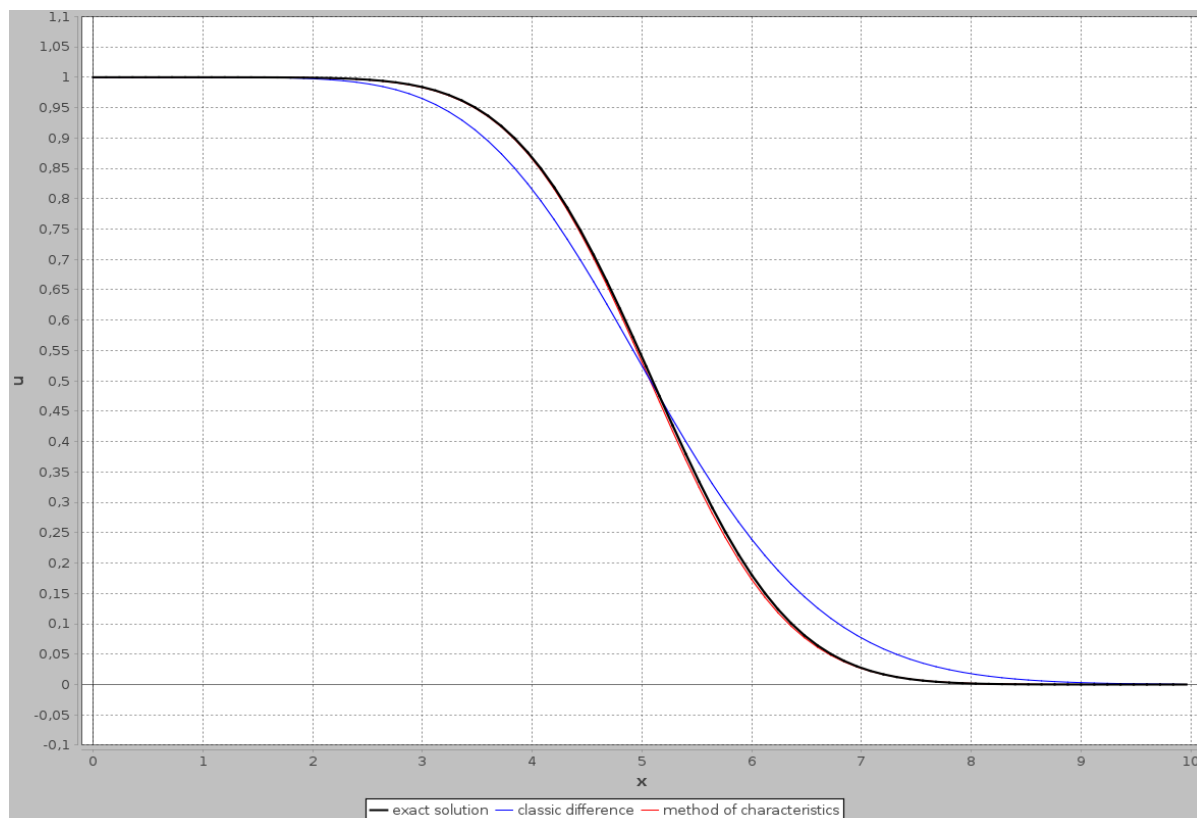


Cu: 0.7

Pe: 1.2

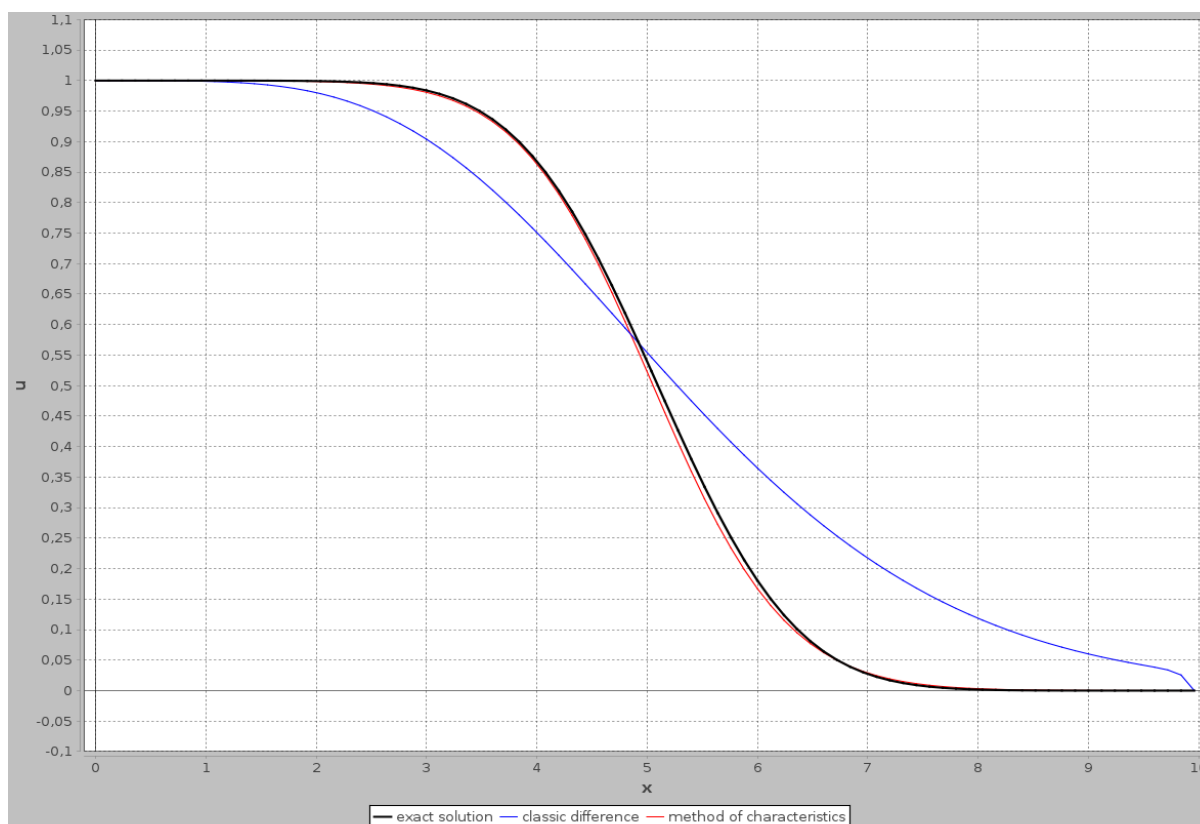
☒ With classic scheme[Draw the graphic](#)

Додаток 8



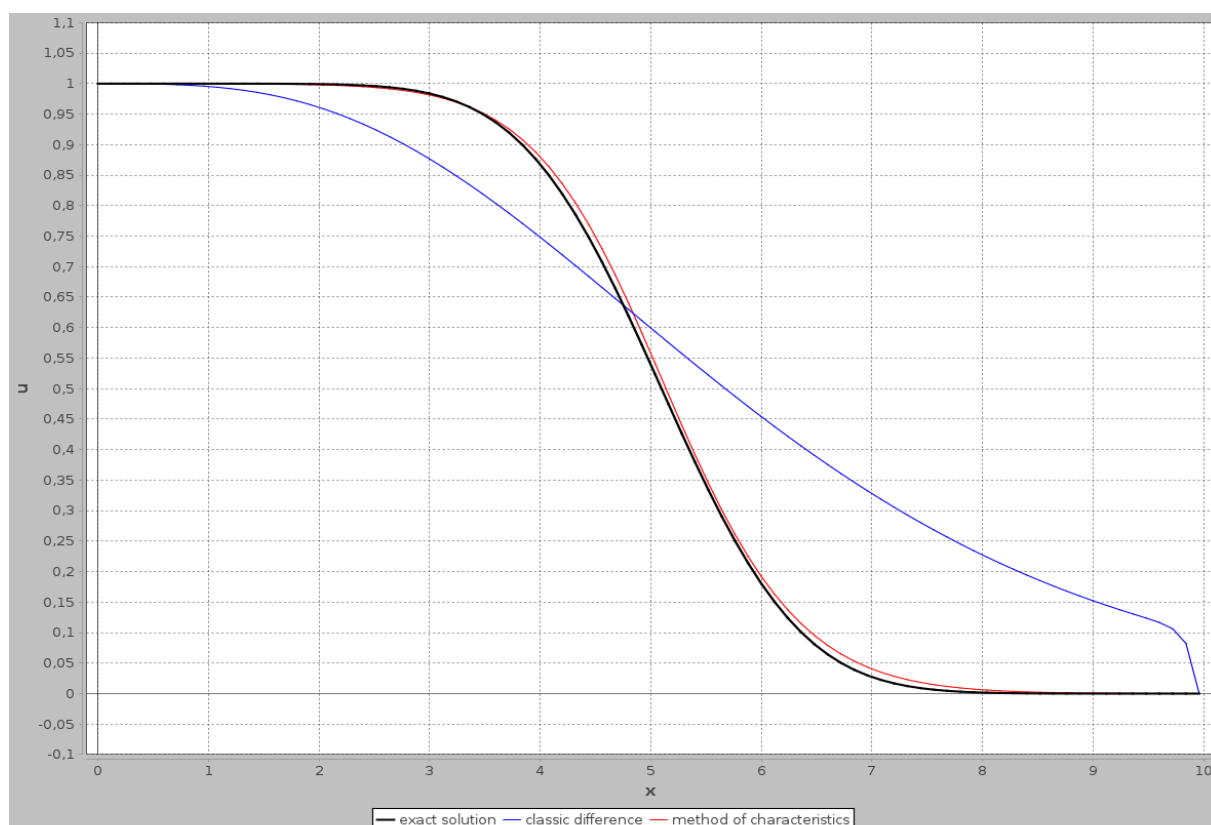
$Cu: 1$ $Pe: 1.2$ ☒ With classic scheme [Draw the graphic](#)

Додаток 9



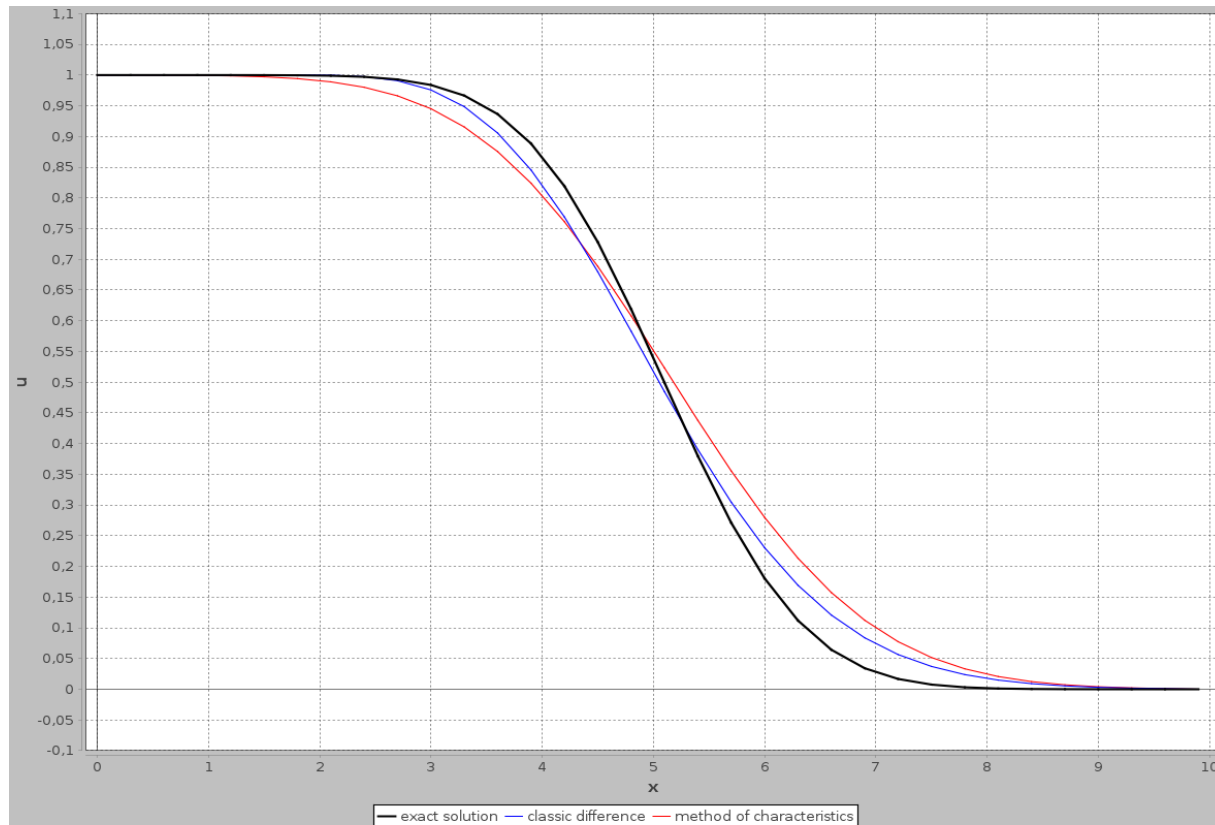
$Cu: 5$ $Pe: 1.2$ ☒ With classic scheme [Draw the graphic](#)

Додаток 10



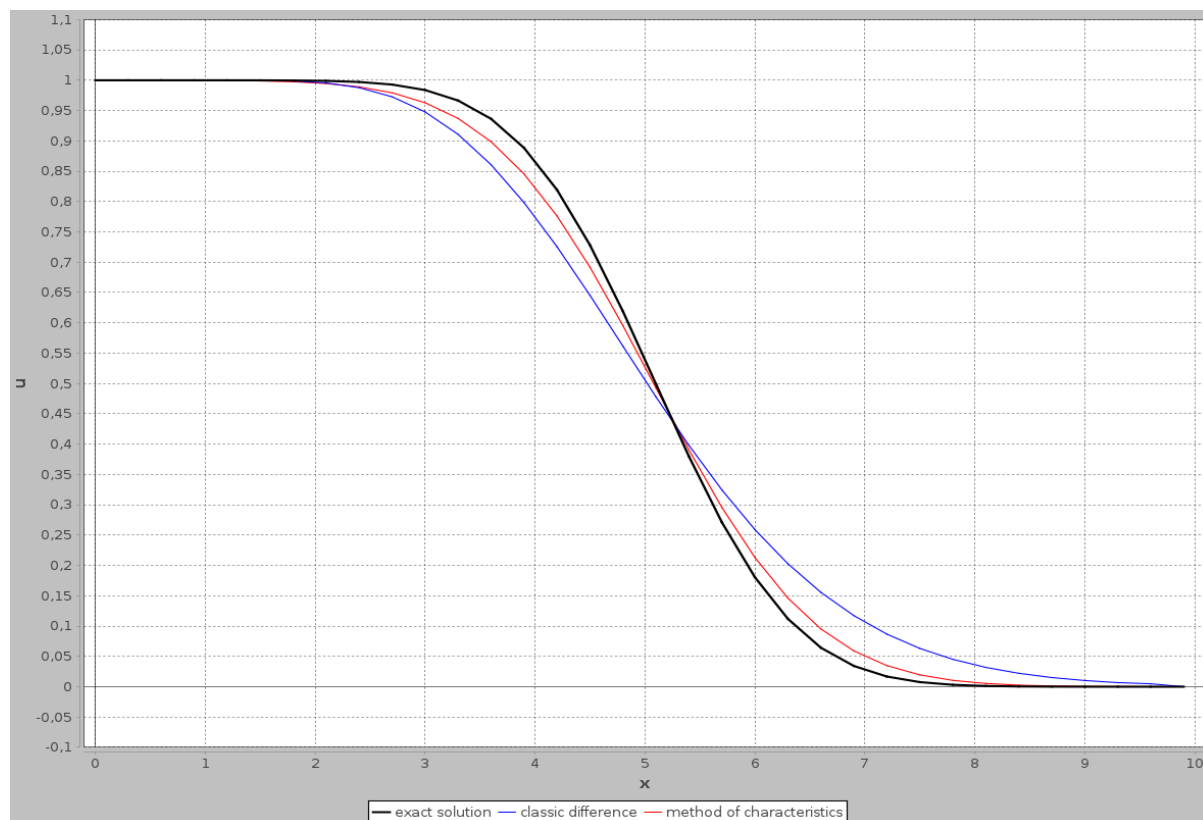
$Cu: 10$ $Pe: 1.2$ ☒ With classic scheme [Draw the graphic](#)

Додаток 11



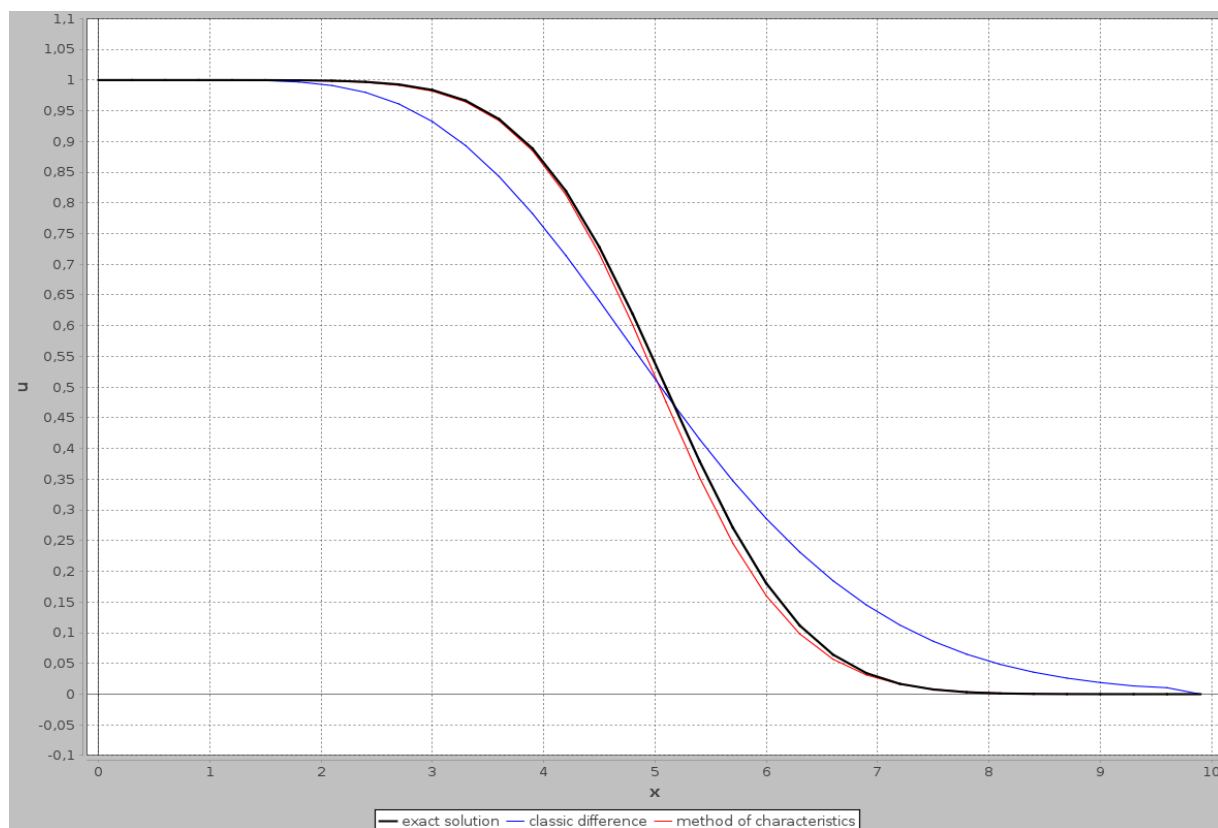
$Cu: 0.3$ $Pe: 3$ ☒ With classic scheme [Draw the graphic](#)

Додаток 12



$Cu: 0.7$ $Pe: 3$ ☒ With classic scheme [Draw the graphic](#)

Додаток 13



$Cu: 1$ $Pe: 3$ ☒ With classic scheme [Draw the graphic](#)

Додаток 14

