

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**Б.П. ДОВГИЙ
Є.С. ВАКАЛ
А.В. ЛОВЕЙКІН**

МЕТОДИ ОБЧИСЛЕНЬ
Конспект лекцій та практичні заняття

**Навчальний посібник
для студентів механіко-математичного-факультету**

Київ – 2024

УДК 51-37:519.6 (0.75.08)

Д 58

Рецензенти:

д-р фіз.-мат. наук, член-кореспондент НАН України Ю.В.Крак
канд. фіз.-мат наук, ст. наук. співр. О.Б. Стеля

Рекомендовано до друку
вченою радою механіко-математичного факультету
(протокол № 14 від 2 травня 2024 року)

Довгий Б.П.

Д 58 Методи обчислень. Конспект лекцій та практичні заняття :
навчальний посібник для студентів механіко-математичного
факультету / Б.П.Довгий, Є.С.Вакал, А.В. Ловейкін. – К.: 2024. – 246 с.

Навчальне видання містить лекційний матеріал, в якому розглядаються теоретичні засади використання наближених методів розв'язання інтегральних рівнянь, задач Коші та крайових задач для звичайних диференціальних рівнянь, а також методів розв'язання крайових задач для рівнянь з частинними похідним.

Матеріал практичних занять містить значну кількість розв'язаних задач, а також тексти програм із реалізованими методами. Наведено завдання для самостійної роботи студентів.

За змістом видання (як лекції, так і практичні заняття) поділено на теми згідно з навчальною програмою. Головна увага приділяється питанням практичного застосування чисельних методів. Як базовий інструмент реалізації методів обрано пакет Octave/MATLAB, в якому реалізовано низку стандартних і спеціальних чисельних методів.

Для студентів механіко-математичного факультету, які навчаються за ОПП "Математика"

УДК 51-37:519.6 (0.75.08)

©Довгий Б.П., Вакал, Є.С., Ловейкін А.В.

ВСТУП

На сучасному етапі розвитку науки, техніки, економіки та інших сфер людського життя все частіше для досягнення бажаного результату доводиться мати справу із досить складними математичними задачами. Сучасне описання економічних процесів, фізичних явищ, процесів у біології, соціальних подій, розробка нових машин та агрегатів тощо неможливе без математичного моделювання, коли реальні процеси, явища описуються мовою математики. Причому, чим більше факторів впливає на об'єкт дослідження, тим складнішою буде математична модель, задача, з якою доведеться мати справу. Таким чином, для отримання необхідного результату в різних сферах людського життя все частіше і частіше доводиться розв'язувати складні математичні задачі. Наприклад, при дослідженні фізичних явищ у нелінійній оптиці доводиться мати справу із системами нелінійних диференціальних рівнянь з частинними похідними різного типу та різного порядку, при описанні руху механічних систем порталного крану потрібно розв'язувати системи рівнянь, одні з яких є звичайними диференціальними рівняннями, а інші – рівняннями з частинними похідними, при проектуванні систем охолодження ядерного реактора потрібно вміти розв'язувати нелінійне рівняння теплопровідності, або їх системи, при прогнозуванні явищ в економіці доводиться стикатись із системами диференціальних рівнянь, в яких зовнішні фактори мають імпульсну дію (миттєвий вихід на товарний ринок великої кількості продукції) або пролонговану дію (вплив фактора відчувається із запізненням), тобто маємо рівняння із імпульсною дією чи запізненням аргумента. Можна навести ще багато прикладів складних математичних задач, які доводиться розв'язувати, щоб описати реальні процеси. Більшість із таких задач неможна розв'язати абсолютно точно, оскільки в сучасній математичній науці немає методів розв'язання цих задач. В таких випадках на допомогу приходять наближені або чисельні (числові) методи.

На механіко-математичному факультеті такі методи вивчаються в курсі "Методи обчислень" для студентів, що навчаються за спеціальністю "Математика". Основною метою курсу є ознайомлення студентів із сучасними методами наближеного розв'язання різноманітних математичних задач та розвинення практичних навичок застосування цих методів до розв'язання конкретних задач. Згідно із навчальним планом курс "Методи обчислень" вивчається в другому семестрі IV курсу. Під час навчання передбачено лекційний матеріал та проведення практичних занять.

Методична розробка акумулює матеріал курсу "Методи обчислень" у формі лекцій, практичних (аудиторних) занять та варіантів індивідуальних завдань для самостійної роботи студентів. Весь матеріал розділено на окремі теми, кожна з яких присвячена розгляду окремого методу або кількох методів розв'язання конкретного типу задач. При проведенні практичних занять основна увага студентів зосереджується на особливостях застосування наближеного методу для розв'язання заданого типу задач, доведення його до алгоритмічного виду, створення ПЗ на MATLAB/Octave [4, 5, 9, 11–13, 24, 25, 28], і розв'язання тестових задач. Така методика сприяє виконанню завдань самостійної роботи, які покликані закріпити практичні навички студентів в застосуванні числових методів, а також навчас технології програмної реалізації методів. Крім того це полегшує студентам виконання індивідуальних завдань на протязі семестру, враховуючи значний обсяг теоретичного матеріалу і короткого терміну їх виконання.

ЛЕКЦІЯ № 01

ТЕМА № 1. Числове розв'язування інтегральних рівнянь.

Інтегральним називається рівняння, в якому невідома функція $u(x)$ знаходиться під знаком інтеграла. Є два основних типи одновимірних інтегральних рівнянь (ІР):

	<i>вольтеррового типу</i>	<i>фредгольмового типу</i>
<i>Урисона</i>	$\int_a^x K(x, \theta, u(\theta)) d\theta = f(x, u(x))$	$\int_a^b K(x, \theta, u(\theta)) d\theta = f(x, u(x))$
<i>Гаммерштейна</i>	$\int_a^x K(x, \theta) F(u(\theta)) d\theta = f(x, u(x))$	$\int_a^b K(x, \theta) F(u(\theta)) d\theta = f(x, u(x))$
<i>II роду</i>	$u(x) + \int_a^x K(x, \theta) u(\theta) d\theta = f(x)$	$u(x) + \int_a^b K(x, \theta) u(\theta) d\theta = f(x)$
<i>I роду</i>	$\int_a^x K(x, \theta) u(\theta) d\theta = f(x)$	$\int_a^b K(x, \theta) u(\theta) d\theta = f(x)$

де K , F , f – відомі функції. Теорія ІР достатньо розвинена, відомі властивості розв'язку $u(x)$, його існування і стійкість залежно від типу ІР, властивостей ядра K і функції правої частини f .

Задачі для ІР фредгольмового|вольтеррового типу першого роду є типовими представниками некоректно поставленої задачі (при малих збуреннях вхідних даних збурення шуканого розв'язку стають як завгодно великими) і для них існують відповідні методи регуляризації [18, с. 134-144; 19, с. 259-261].

Враховуючи природний перехід від одновимірних ІР до багатовимірних, ми будемо в подальшому розглядати наближені методи розв'язування для одновимірних коректно поставлених задач.

В залежності від базового підґрунтя методу побудови наближеного розв'язку, можна виділити такі основні методи:

1. **метод квадратурних формул (різницьовий метод);**
2. **метод заміни ядра виродженням**, з реалізацією у варіанті:
 - а) *чисельно-аналітичного*;
 - б) *чисельного*;
3. **метод заміни шуканої функції**, який має декілька варіантів:
 - а) *кусково-стала апроксимація (метод колокації)*;
 - б) *кусково-лінійна апроксимація (метод сплайн-функції 1-го степеня)*;
 - с) *кусково-кубічна апроксимація (метод кубічних сплайнів)*;
4. **метод послідовних наближень**, з реалізацією у варіанті:
 - а) *аналітичного*;
 - б) *чисельного*.

п.1. Різницьовий метод

Це достатньо простий чисельний метод, який дозволяє знайти розв'язок одновимірних ІР з гарною точністю, а для двовимірних – з задовільною. Розглянемо його застосування для нелінійного ІР Фредгольма-Урисона

$$\int_a^b K(x, \theta, u(\theta)) d\theta = f(x, u(x)) \quad (1.1)$$

і для ІР Вольтерри II роду ($\lambda \neq 0$ – дійсне число)

$$u(x) = f(x) + \lambda \int_a^x K(x, \theta) u(\theta) d\theta, \quad x \in [a, b]. \quad (1.2)$$

Введемо сітку, наприклад,

$$\bar{X} = \{X_k = a + (k-1)h, \quad k = \overline{1, n}, \quad b-a = (n-1)h\} \quad (1.3)$$

і в кожній точці сітки запишемо відповідне ІР, при цьому замінимо шукану функцію $u(X_i)$ на сіткову функцію y_i , а інтеграл $\int_a^b \psi(\theta) d\theta$ замінимо деякою квадратурною формулою (КФ) $\sum_{j=1}^m C_j(m) \psi(\zeta_j)$. Тут $C_j(m)$ – коефіцієнти відповідної КФ, а вузли мають співпадати з сіткою (1.3), тому $\zeta_j = X_j$. В результаті отримуємо систему рівнянь відносно $\bar{Y} = [y_1, y_2, \dots, y_n]$. Так, для (1.1) маємо

$$\begin{cases} \sum_{j=1}^n C_j(n) K(X_i, X_j, y_j) = f(X_i, y_i), \\ i = 1, 2, \dots, n, \end{cases}$$

або в стандартному вигляді

$$\begin{cases} F_i(\bar{Y}) \equiv -f(X_i, y_i) + \sum_{j=1}^n C_j(n) K(X_i, X_j, y_j) = 0, \\ i = 1, 2, \dots, n, \end{cases} \quad (1.4)$$

Систему (1.4) розв'язуємо методом Ньютона:

$$\begin{cases} \left(\frac{\partial F_i(\bar{Y}^{(S)})}{\partial y_j} \right)_{i,j=1}^n * \begin{pmatrix} \Delta_1^{(S)} \\ \vdots \\ \Delta_n^{(S)} \end{pmatrix} = - \begin{pmatrix} F_1(\bar{Y}^{(S)}) \\ \vdots \\ F_n(\bar{Y}^{(S)}) \end{pmatrix}, \\ \bar{Y}^{(S+1)} = \bar{Y}^{(S)} + \bar{\Delta}^{(S)}, \quad S = 0, 1, \dots, \|\bar{\Delta}^{(S)}\| \leq \varepsilon. \end{cases} \quad (1.5)$$

Тут позначено S – номер ітерації, $\bar{Y}^{(0)}$ – деяке початкове наближення, а

$$\frac{\partial F_i(\bar{Y}^{(S)})}{\partial y_j} = \begin{cases} C_j(n) K'_u(X_i, X_j, y_j^{(S)}), & i \neq j; \\ -f'_u(X_i, y_i^{(S)}) + C_i(n) K'_u(X_i, X_i, y_i^{(S)}), & i = j. \end{cases}$$

Для (1.2) маємо наступну систему лінійних рівнянь відносно $\bar{Y}$:

$$\begin{cases} y_i = \lambda \sum_{j=1}^i C_j(i) K(X_i, X_j) y_j + f(X_i), \\ i = 1, 2, \dots, n. \end{cases} \quad (1.6)$$

Враховуючи, що (1.6) це СЛАР з нижньою трикутною матрицею, то її розв'язок знаходиться по явним формулам:

$$\begin{cases} y_i = \frac{f(X_i) + \lambda \sum_{j=1}^{i-1} C_j(i) K(X_i, X_j) y_j}{1 - \lambda C_i(i) K(X_i, X_j)}, \\ i = 1, 2, \dots, n. \end{cases} \quad (1.7)$$

Різницький метод застосовують і для однорідного рівняння Фредгольма II роду (задача на власні значення):

$$u(x) = \lambda \int_a^b K(x, \theta) u(\theta) d\theta, \quad x \in [a, b] \quad (1.8)$$

при цьому отримуємо задачу на власні значення $\tilde{\mathbf{K}} \cdot \vec{\mathbf{Y}} = \frac{1}{\lambda} \vec{\mathbf{Y}}$, де матриця $\tilde{\mathbf{K}} = (\tilde{K}_{i,j})_{i,j=1}^n$, з елементами $\tilde{K}_{i,j} = C_j(n)K(X_i, X_j)$.

При невеликих значеннях n використовують квадратурні формули Гаусса або Гаусса-Крістоффеля, але функції K, f мають бути достатньо високої гладкості (програмну реалізацію див. [14, с. 3-24]). Можна використовувати КФ трапецій на послідовності сіток, що згущаються і уточнювати розв'язок методом Рунге-Ромберга. Такий підхід можна використовувати і при недостатній гладкості K, f і навіть у випадку наявності розривів в цих функціях (важливо, щоб точки розривів співпадали з деякими вузлами сітки).

п.2. Метод заміни ядра виродженням

п.2.1. Метод заміни ядра виродженням (чисельно-аналітичний варіант)

Будемо розглядати розв'язування ІР Фредгольма II роду, записане у вигляді:

$$u(x) - \lambda \int_a^b K(x, \theta) u(\theta) d\theta = f(x), \quad (1.9)$$

де $f(x) \in C([a, b])$, $K(x, \theta) \in C([a, b] \times [a, b])$, $K(x, \theta) \neq 0$ – відомі функції, $\lambda \neq 0$ – дійсне число, яке не збігається з характеристичними числами (1.9).

Ядро $K(x, \theta)$ ІР фредгольмового типу називається *виродженням*, якщо

$$K(x, \theta) \equiv K^0(x, \theta) = \sum_{i=1}^m A_i(x) B_i(\theta), \quad (1.10)$$

де $A_i(x)$ – лінійно-незалежні функції.

Розв'язок ІР з виродженням ядром можна знайти в явній формі. Підставимо (1.10) в (1.9), поміняємо місцями операції інтегрування і суми, отримаємо

$$u(x) = \lambda \sum_{i=1}^m A_i(x) \underbrace{\int_a^b B_i(\theta) u(\theta) d\theta}_{c_i} + f(x) = \lambda \sum_{i=1}^m c_i A_i(x) + f(x), \quad (1.11)$$

тобто, права частина рівності (1.11) описує вид розв'язку (1.9). Залишається знайти вектор $\vec{c} = [c_1, c_2, \dots, c_m]$, де $c_i \in \text{Const}$. Для цього підставимо (1.11) – знайдену форму розв'язку $u(x)$ в (1.9) і отримаємо рівність

$$\lambda \sum_{i=1}^m c_i A_i(x) + f(x) - \lambda \int_a^b \left[\sum_{i=1}^m A_i(x) B_i(\theta) \left(\lambda \sum_{j=1}^m c_j A_j(\theta) + f(\theta) \right) \right] d\theta = f(x),$$

яку, зібравши коефіцієнти біля $A_i(x)$, перепишемо у вигляді

$$\sum_{i=1}^m \left[c_i - \lambda \sum_{j=1}^m c_j \int_a^b B_i(\theta) A_j(\theta) d\theta - \int_a^b B_i(\theta) f(\theta) d\theta \right] A_i(x) = 0.$$

З цього співвідношення, з огляду лінійно-незалежності функцій $A_i(x)$, отримуємо СЛАР для $\vec{c}$:

$$A\vec{c} = \vec{\varphi}, \quad (1.12)$$

де

$$A = (A_{ij})_{i,j=1}^m, \quad A_{ij} = \delta_{ij} - \lambda \sum_{j=1}^m a_{ij}, \quad a_{ij} = \int_a^b B_i(\theta) A_j(\theta) d\theta; \quad \vec{\varphi} = (\varphi_i)_{i=1}^m, \quad \varphi_i = \int_a^b B_i(\theta) f(\theta) d\theta. \quad (1.13)$$

Після розв'язання СЛАР (1.12), отримуємо чисельно-аналітичний розв'язок ІР (1.9) у вигляді (1.11).

Якщо ІР (1.9) однорідне ($f(x) \equiv 0$), то $\varphi_i \equiv 0$ і (1.12) стає задачею на власні значення, а отже для виродженого ядра маємо в точності m власних чисел.

Для застосування цього методу при довільному ядрі, необхідно його апроксимувати виродженим $K(x, \theta) \cong K^0(x, \theta)$. Наприклад, виконати розвинення $K(x, \theta)$ у певний ряд (степеневий, Фур'є і т.д.) по деякій повній ортонормованій системі функцій $A_i(x)$ і обмежитись m -членами розвитку. Оцінки точності таких наближень $u(x) \cong y_{(m)}(x)$ досить громіздкі і незручні в практичних обчисленнях. На практиці можна перевіряти виконання нерівності $\|y_{(m+1)}(x) - y_{(m)}(x)\| \leq \varepsilon$, де $0 < \varepsilon < 1$ – обрана точність.

п.2.1. Метод заміни ядра виродженим (чисельний варіант)

Обчислювальна схема методу наступна:

1. При необхідності ядро IP (1.9) апроксимуємо виродженим $K(x, \theta) \cong K^0(x, \theta)$.
2. Вводимо сітку, наприклад, рівномірну по формулам (1.3).
3. Обчислюємо значення функцій $f(x)$, $A_i(x)$, $B_i(x)$ на сітці $\bar{X}$.
4. Використовуємо формули (1.13) і КФ з вузлами $\bar{X}$ – будуємо СЛАР (1.12).
5. Розв'язуємо СЛАР і отримуємо коефіцієнти $\bar{c}$.
6. По формулі $\bar{Y}_{(m)} = \lambda \sum_{i=1}^m c_i A_i(\bar{X}) + f(\bar{X})$ отримуємо наближення до $u(\bar{X})$.

п.3. Метод заміни шуканої функції

Цей метод базується на заміні шуканої функції на деяку простішу функцію, яка не тільки апроксимує $u(x)$, а й задовольняє інтегральне рівняння. При заміні функції $u(x)$ на інтерполяційний сплайн $S_{p,q}(x; u)$ найчастіше використовують такі значення степеня $p=0,1,3$ [8, 10, с. 106-115; 19, с. 253-259; 34, 35].

п.3.1. Кусково-стала апроксимація (метод колокації)

Шукана функція $u(x)$ замінюється кусково-сталою функцією $S_{0,1}(x; u)$, $x \in [a, b]$ зі значеннями y_i в певних точках $\bar{x}_i$ (точках колокації), де

$$\bar{x}_i \in \Delta_i = [X_i, X_{i+1}], \quad h_i = X_{i+1} - X_i, \quad [a, b] = \bigcup_{i=1}^N \Delta_i, \quad N = n-1, \quad X_i \in \bar{X},$$

а отже (1.9) приводиться до системи співвідношень :

$$\begin{cases} y_i - \lambda \sum_{j=1}^N y_j \int_{x_j}^{x_{j+1}} K(\bar{x}_i, \theta) d\theta = f(\bar{x}_i), \\ i=1, 2, \dots, N. \end{cases} \quad (1.14)$$

Для обчислення інтегралів в (1.14) можна використати КФ лівих|правих прямокутників з точністю $O(h_j)$, а якщо точки колокації обрати у вигляді $\bar{x}_i = 0.5(X_i + X_{i+1})$, то інтеграли в (1.14) легко обчислюються по КФ центральних прямокутників з точністю $O(h_j^2)$. В результаті отримуємо СЛАР відносно $\bar{y} = \{y_1, y_2, \dots, y_N\}$:

$$\begin{cases} \sum_{j=1}^N (\delta_{ij} - \lambda \cdot h_j K(\bar{x}_i, \bar{x}_j)) y_j = f(\bar{x}_i), \\ i=1, 2, \dots, N. \end{cases} \quad (1.15)$$

Після розв'язування СЛАР (1.15) – маємо сплайнове наближення до функції $u(x)$ – розв'язку IP (1.9).

п.3.2. Кусково-лінійна апроксимація (метод сплайн-функції 1-го степеня)

На кожному з відрізків сітки $\bar{X}$ (1.3) замінимо функцію $u(x)$ лінійним сплайном у формі "лагранжіана"

$$S_{1,1}(x;u) = y_i \psi_1(t) + y_{i+1} \psi_2(t), \quad x \in \Delta_i;$$

$$y_i = u(X_i) = S_{1,1}(X_i;u);$$

$$\psi_1(t) = 1-t; \quad \psi_2(t) = t; \quad t \in [0,1]; \quad x = X_i + h_i t.$$

Тепер запишемо рівняння (1.9) у кожній точці X_i , $i = 1, \dots, n$ сітки $\vec{X}$ з урахуванням адитивності інтеграла:

$$y_i - \lambda \sum_{j=1}^{n-1} \int_{X_j}^{X_{j+1}} K(X_i, \theta) S_{1,1}(\theta; u) d\theta =$$

$$= y_i - \lambda \sum_{j=1}^{n-1} \left(y_j \int_0^1 K_{i,j}(t) \psi_1(t) h_j dt + y_{j+1} \int_0^1 K_{i,j}(t) \psi_2(t) h_j dt \right) = f(X_i),$$

де введено позначення

$$K_{i,j}(t) = K(X_i, X_j + h_j t). \quad (1.16)$$

Об'єднуючи рівняння для всіх точок сітки $\vec{X}$, отримаємо СЛАР відносно $\vec{y} = \{y_i\}_{i=1}^n$, яку запишемо в матричній формі

$$A \vec{y} = \vec{f},$$

$$A = \{a_{i,j}\}_{i,j=1}^n, \quad \vec{f} = \{f(X_i)\}_{i=1}^n,$$

де елементи матриці можна обчислити таким чином:

$$a_{i,j} = \delta_{ij} + \lambda \tilde{a}_{i,j},$$

$$\tilde{a}_{i,1} = h_1 \int_0^1 K_{i,1}(t) \psi_1(t) dt;$$

$$\tilde{a}_{i,j} = h_{j-1} \int_0^1 K_{i,j-1}(t) \psi_2(t) dt + h_j \int_0^1 K_{i,j}(t) \psi_1(t) dt, \quad j = 2, \dots, n-1;$$

$$\tilde{a}_{i,n} = h_{n-1} \int_0^1 K_{i,n-1}(t) \psi_2(t) dt.$$

Для знаходження елементів матриці A можна:

- обчислити інтеграли, застосовуючи квадратурні формули;
- винести значення $K_{i,j}(t)$ з під знака інтеграла в деякій точці $t \in [0,1]$ і обчислити інтеграли від функцій $\psi_{1,2}(t)$ точно.

п.3.3. Кусково-кубічна апроксимація (метод кубічних сплайнів)

На кожному з відрізків сітки $\vec{X}$ (1.9) замінимо шукану функцію кубічним сплайном у формі "лагранжіана":

$$S_{3,1}(x;u) = M_i h_i^2 \varphi_1(t) + M_{i+1} h_i^2 \varphi_2(t) + y_i \psi_1(t) + y_{i+1} \psi_2(t), \quad x \in \Delta_i;$$

$$M_i = S_{3,1}''(X_i;u), \quad y_i = u(X_i) = S_{3,1}(X_i;u);$$

$$\varphi_1(t) = -\frac{1}{6} t(1-t)(2-t), \quad \varphi_2(t) = -\frac{1}{6} t(1-t)(1+t),$$

$$\psi_1(t) = 1-t, \quad \psi_2(t) = t, \quad t \in [0,1]; \quad x = X_i + h_i t.$$

Запишемо рівняння (1.9) у кожній точці X_i , $i = 1, \dots, n$, сітки $\vec{X}$ з урахуванням адитивності інтеграла і позначень (1.16):

$$y_i - \lambda \sum_{j=1}^{n-1} \left(M_j h_j^3 \int_0^1 K_{i,j}(t) \varphi_1(t) dt + M_{j+1} h_j^3 \int_0^1 K_{i,j}(t) \varphi_2(t) dt + \right. \\ \left. + y_j h_j \int_0^1 K_{i,j}(t) \psi_1(t) dt + y_{j+1} h_j \int_0^1 K_{i,j}(t) \psi_2(t) dt \right) = f(X_i), i = \overline{1, n}. \quad (1.17)$$

До рівнянь (1.17) необхідно додати рівняння визначення параметрів $\{M_i\}_{i=1}^n$ кубічного сплайна із крайовими умовами IV типу:

$$M_1 - (1 + \eta)M_2 + \eta M_3 = 0, \\ (2 + \eta)M_2 + (1 - \eta)M_3 + \frac{6}{h_1 + h_2}(y_{1;2} - y_{2;3}) = 0, \\ \tilde{h}_i M_{i-1} + 2M_i + \tilde{h}_i M_{i+1} + \frac{6}{h_{i-1} + h_i}(y_{i-1;i} - y_{i;i+1}) = 0, \\ i = 3, \dots, n-2, \\ (1 - \mu)M_{n-2} + (2 + \mu)M_{n-1} + \frac{6}{h_{n-2} + h_{n-1}}(y_{n-2;n-1} - y_{n-1;n}) = 0, \\ \mu M_{n-2} - (1 + \mu)M_{n-1} + M_n = 0. \quad (1.18)$$

Тут введено позначення $y_{i;j}$ - поділені різниці першого порядку, а решта :

$$\tilde{h}_i = h_i / (h_{i-1} + h_i), \quad \tilde{h}_i = 1 - \tilde{h}_i, \quad \eta = h_1 h_2^{-1}, \quad \mu = h_{n-1} h_{n-2}^{-1}.$$

Таким чином, для невідомих $M_1, \dots, M_n, y_1, \dots, y_n$ розв'язуємо СЛАР (1.17) – (1.18).

п.4. Метод послідовних наближень

п.4.1. Метод послідовних наближень (аналітичний варіант)

Для рівняння (1.9) розв'язок $u(x)$ шукається у вигляді ряду по степеням параметра λ :

$$y_m(x) = \varphi_0(x) + \lambda \varphi_1(x) + \lambda^2 \varphi_2(x) + \dots + \lambda^m \varphi_m(x), \quad (1.19)$$

який збігається при $|\lambda(b-a)|M < 1$, де $|K(x, \theta)| \leq M$, $M \in \text{Const}$, а функції $\varphi_i(x)$ обчислюються за рекурентними формулами

$$\begin{cases} \varphi_0(x) = f(x), \\ \varphi_i(x) = \int_a^b K(x, \theta) \varphi_{i-1}(\theta) d\theta, \quad i = 1, 2, \dots, m. \end{cases} \quad (1.20)$$

Розрахунки по формулам (1.19) закінчуються, якщо для заданого $0 < \varepsilon < 1$ виконується умова $\|\varphi_m(x)\| \leq \varepsilon$.

п.4.2. Метод послідовних наближень (чисельний варіант)

Має наступний алгоритм :

1. Вводимо сітку $\bar{X}$, наприклад, (1.3).
2. Обчислюємо вектори $\bar{\varphi}_0 = [f(X_k)]_{k=1}^n$, $\bar{y}_0 = \bar{\varphi}_0$ і матрицю $\mathbf{B} = \lambda [C_j K(X_k, X_j)]_{k,j=1}^n$, де C_j – коефіцієнти певної КФ з вузлами X_j .

3. Наближений сітковий розв'язок обчислюється за рекурентними формулами

$$\begin{cases} \bar{\varphi}_i = \mathbf{B} * \bar{\varphi}_{i-1}, \quad \bar{y}_i = \bar{y}_{i-1} + \bar{\varphi}_i, \\ i = 1, 2, \dots, m. \end{cases} \quad (1.21)$$

4. Розрахунки по формулам (1.21) закінчуються, якщо для заданого $0 < \varepsilon < 1$ виконується умова $\|\bar{\varphi}_m\|_h \leq \varepsilon$.

ЛЕКЦІЯ № 02

ТЕМА № 2. Методи розв'язування задачі Коші для звичайних диференціальних рівнянь.

Будемо розглядати задачу Коші для системи N звичайних диференціальних рівнянь (ЗДР) першого порядку

$$\begin{cases} \frac{d\vec{u}(x)}{dx} = \vec{f}(x, \vec{u}(x)), & x \in (x_0, x_{end}], \\ \vec{u}(x_0) = \vec{u}_0, \end{cases} \quad (2.1)$$

для якої виконуються відповідні умови існування та єдиності розв'язку (задача поставлена коректно), а також вхідні дані і розв'язок мають необхідну гладкість.

Ми будемо цікавитись тільки *чисельними методами* – алгоритмами обчислення наближеного розв'язку для деяких значень аргументу x (на сітці). Для успішного застосування цих методів, крім коректності, потрібна *хороша обумовленість* (малі зміни початкових умов приводять до достатньо малих змін в інтегральних кривих). Наведемо приклад коректної, але погано обумовленої задачі Коші для ЗДР

$$\begin{cases} u'(x) = u - x, & x \in (0, 100], \\ u(0) = u_0. \end{cases}$$

Загальний розв'язок якої записується у вигляді $u(x; C) = 1 + x + C \exp(x)$, а отже маємо

Значення u_0	Значення C	Значення $u(100)$
1	0	101
1.000001	10^{-6}	$\approx 2.7 \times 10^{37}$

Розглянемо спочатку скалярний варіант (2.1), а алгоритми для системи N ЗДР отримуються із алгоритмів для скалярної задачі формальною заміною $u(x)$, $f(x, u)$ на $\vec{u}(x)$, $\vec{f}(x, \vec{u})$.

В залежності від стратегії побудови чисельного методу, існують дві групи методів:

- а) *різницеві однокрокові* і *багатокрокові* (серед них методи Адамса, Гіра);
- б) *типу Рунге-Кутта*.

Як перші, так і другі бувають *явні* і *неявні*.

п.1. Різницеві однокрокові

Загальна схема побудови методів: вводимо по змінній x сітку з певним кроком (рівномірним або нерівномірним). Виконуємо апроксимацію лівої і правої частини рівняння (2.1) в певній точці *шаблону* $S\{x_n, x_{n+1}\}$. Шаблон – це сукупність вузлів сітки, які використовуються для запису різницевої схеми.

Алгоритм явного методу Ейлера. Для апроксимації (2.1) беремо точку x_n і $u'(x_n)$ замінимо на *праву різницеву похідну* $y_{x,n} = (y_{n+1} - y_n)h_n^{-1}$, $y_n = u(x_n)$, $h_n = x_{n+1} - x_n$, отримаємо:

$$y_{n+1} = y_n + h_n f(x_n, y_n), \quad y_0 = u_0, \quad n = 0, 1, \dots \quad (2.2)$$

Алгоритм неявного методу Ейлера. Для апроксимації (2.1) беремо точку x_{n+1} , а $u'(x_{n+1})$ замінимо на *ліву різницеву похідну* $y_{\bar{x},n+1} = (y_{n+1} - y_n)h_n^{-1}$, отримаємо:

$$y_{n+1} = y_n + h_n f(x_{n+1}, y_{n+1}), \quad y_0 = u_0, \quad n = 0, 1, \dots \quad (2.3)$$

Для знаходження y_{n+1} необхідно розв'язати нелінійне рівняння

$$F(y_{n+1}) = 0, \quad (2.4)$$

де $F(\xi) = \xi - h_n f(x_{n+1}, \xi) - y_n$. Можна використати ІМ Ньютона з початковим наближенням $y_{n+1}^{(0)} = y_n$.

Алгоритм неявного методу "симетрична схема". Для апроксимації (2.1) оберемо додаткову точку $x_{n+0.5} = 0.5(x_n + x_{n+1})$ і $u'(x_{n+0.5})$ замінімо на *центрально різницеву похідну* $y_{\tilde{x},n} = (y_{n+1} - y_n)h_n^{-1}$ (побудовану по точках $x_n, x_{n+0.5}, x_{n+1}$ з кроком $0.5h_n$), а $f(x_{n+0.5}, u_{n+0.5})$ – замінімо на середнє значення в точках шаблону S , отримаємо:

$$y_{n+1} = y_n + 0.5h_n(f(x_n, y_n) + f(x_{n+1}, y_{n+1})), \quad y_0 = u_0, \quad n = 0, 1, \dots \quad (2.5)$$

Тут теж для y_{n+1} маємо рівняння виду (2.4), де $F(\xi) = \xi - 0.5h_n(f(x_n, y_n) + f(x_{n+1}, \xi)) - y_n$.

п.2. Поняття збіжності, нев'язки, порядку апроксимації

Фіксуємо точку x і будуємо послідовність сіток ω_h таких, що при $h \rightarrow 0$ і $x_n = x_0 + nh = x$, $n \rightarrow \infty$. Тоді маємо [5, с. 215]:

Озн.2.1. Чисельний метод збігається в точці x , якщо $|y_n - u(x_n)| \rightarrow 0$ при $h \rightarrow 0$, $x_n = x$.

Озн.2.2. Метод збігається на відрізку $(x_0, X]$, якщо він збігається для кожної точки $x \in (x_0, X]$.

Озн.2.3. Метод має p -й порядок точності, якщо існує таке число p , що $|y_n - u(x_n)| = O(h^p)$ при $h \rightarrow 0$.

Сіткова функція $z_n = y_n - u(x_n)$ – це *похибка методу* і для отримання рівняння, яке вона задовольняє, підставимо $y_n = z_n + u(x_n)$, наприклад, в (2.2). Тоді маємо:

$$\frac{z_{n+1} - z_n}{h_n} = f(x_n, z_n + u(x_n)) - \frac{u(x_{n+1}) - u(x_n)}{h_n}. \quad (2.6)$$

Якщо в праву частину (2.6) додати вираз $\pm f(x_n, u(x_n))$, то рівняння можна подати так

$$\frac{z_{n+1} - z_n}{h_n} = \psi_n^{(1)} + \psi_n^{(2)},$$

де введені функції визначаються наступним чином:

$$\psi_n^{(1)} = -\frac{u(x_{n+1}) - u(x_n)}{h_n} + f(x_n, u(x_n)) = [-u'(x_n) + f(x_n, u(x_n))] - 0.5h_n u''(x_n) + O(h_n^2);$$

$$\psi_n^{(2)} = f(x_n, z_n + u(x_n)) - f(x_n, u(x_n)) = \begin{cases} 0, & f(x, u) \equiv f(x); \\ \frac{\partial f(x_n, u(x_n) + \eta z_n)}{\partial u} z_n, & |\eta| \leq 1, \quad f(x, u) \neq f(x), \end{cases}$$

і $\psi_n^{(1)}$ – називається *нев'язкою* або *похибкою апроксимації різницевого рівняння на розв'язку диференціального*; а $\psi_n^{(2)}$ – в загальному випадку пропорційна похибці z_n . Отримані співвідношення можна використати для асимптотичної оцінки похибки z_n [33], з аналізу якої слідує, що при неперервності $f(x, u)$ і обмеженості $|f| \leq M_1$, $|f'_x| \leq M_2$, $|f'_u| \leq M_3 \Rightarrow |u''| \leq M_4 = M_2 + M_1 M_3$ основну роль тут грає саме оцінка функції $\psi_n^{(1)}$ при $h = \max_n(h_n) \rightarrow 0$.

Отже, для явного і неявного методу Ейлера, враховуючи що, $\psi_n^{(1)} = O(h)$, маємо *перший порядок апроксимації*, а *наближений розв'язок* при $h \rightarrow 0$ *збігається до точного рівномірно* (на обмеженому відрізку $|x - x_0| \leq X$) *з першим порядком точності*.

Знайдемо оцінку $\psi_n^{(1)}$ для неявного методу "симетрична схема". Маємо:

$$\begin{aligned}
\psi_n^{(1)} &= -\frac{u(x_{n+1}) - u(x_n)}{h_n} + 0.5[f(x_n, u(x_n)) + f(x_{n+1}, u(x_{n+1}))] = \\
&= -(u'(x_n) + 0.5h_n u''(x_n) + O(h_n^2)) + 0.5[u'(x_n) + u'(x_{n+1})] = \\
&= -u'(x_n) - 0.5h_n u''(x_n) + 0.5[u'(x_n) + u'(x_n) + h_n u''(x_n)] + O(h_n^2) = O(h_n^2),
\end{aligned}$$

отже (2.5) метод *другого порядку апроксимації*, а *наближений розв'язок* при $h \rightarrow 0$ збігається до *точного рівномірно* (на обмеженому відрізку $|x - x_0| \leq X$) з *другим порядком точності*.

п.3. Методи типу Рунге-Кутта

Ці методи дають можливість будувати схеми різного порядку точності, які зручно використовувати як при реалізації на ЕОМ, так і при «ручних» обчисленнях. Характерна особливість методів Рунге-Кутта (РК), це обчислення правої частини (2.1) не тільки в точках сітки, але і в деяких проміжних точках відрізка $[x_n, x_{n+1}]$.

Спочатку розглянемо приклад побудови методу типу РК, використовуючи схему «*предиктор-коректор*». Припустимо, що ми маємо значення x_n, y_n для деякого n . Перехід до точки x_{n+1} здійснимо в два етапи:

1. $\frac{y_{n+0.5} - y_n}{0.5h_n} = f(x_n, y_n)$ – явною схемою Ейлера знайдемо проміжне значення $y_{n+0.5}$ з точністю $O(h_n)$ (предиктор);
2. $\frac{y_{n+1} - y_n}{h_n} = f(x_n + 0.5h_n, y_{n+0.5})$ – апроксимуємо (2.1) з точністю $O(h_n^2)$ в точці $x_{n+0.5}$ і знаходимо y_{n+1} (коректор).

Для оцінки $\psi_n^{(1)}$ виконаємо її розклад в ряд по h_n в точці (x_n, y_n) :

$$\begin{aligned}
\psi_n^{(1)} &= -\frac{u_{n+1} - u_n}{h_n} + f(x_n + 0.5h_n, u_n + 0.5h_n f(x_n, u_n)) = \\
&= -(u'_n + 0.5h_n u''_n + O(h_n^2)) + \left[f_n + 0.5h_n \left(\frac{\partial f_n}{\partial x} + f_n \frac{\partial f_n}{\partial u} \right) + O(h_n^2) \right] = \\
&= -u'_n + f_n - 0.5h_n u''_n + 0.5h_n u''_n + O(h_n^2) = O(h_n^2),
\end{aligned}$$

отже маємо другий порядок апроксимації і схема явна, на відміну від схеми (2.5). Тут, для спрощення запису, ми позначили $u_n = u(x_n)$, $u'_n = u'(x_n)$, $u''_n = u''(x_n)$, $f_n = f(x_n, u(x_n))$ і скористались $(u')' = (f)' \equiv f'_x + f'_u * u' = f'_x + f * f'_u$.

Загальна схема отримання розрахункових формул методів типу Рунге-Кутта. Інтегруємо рівняння (2.1) на відрізку $[x_n, x_{n+1}]$ і для наближеного обчислення інтегралу від правої частини використовуємо спеціальні квадратурні формули

$$y_{n+1} - y_n = \int_{x_n}^{x_{n+1}} f(\xi, u(\xi)) d\xi = h_n \sum_{i=1}^m A_i K_i, \quad y_0 = u_0, \quad n = 0, 1, \dots, \quad (2.7)$$

що і дає *m-стадійну (m-етапну)* розрахункову формулу. В (2.7) завжди має виконуватись $\sum_{i=1}^m A_i = 1$ – умова апроксимації $f(x, u)$. Якщо будуємо явний варіант методу, то коефіцієнти K_i обчислюються рекурентно за наступними формулами:

$$\begin{aligned}
K_1 &= f(x_n + \alpha_1 h_n, y_n), \quad \alpha_1 = 0, \\
K_2 &= f(x_n + \alpha_2 h_n, y_n + \beta_{21} h_n K_1), \\
K_3 &= f(x_n + \alpha_3 h_n, y_n + \beta_{31} h_n K_1 + \beta_{32} h_n K_2), \\
&\dots \\
K_m &= f(x_n + \alpha_m h_n, y_n + \beta_{m1} h_n K_1 + \beta_{m2} h_n K_2 + \dots + \beta_{mm-1} h_n K_{m-1}).
\end{aligned}$$

Числові параметри $A_i, \alpha_i, \beta_{ij}, 0 < i, j \leq m$ методів РК часто записують в вигляді таблиці Бутчера:

$$\begin{array}{c|cccc}
\alpha_1 & \beta_{11} & \beta_{12} & \cdots & \beta_{1m} \\
\alpha_2 & \beta_{21} & \beta_{22} & \cdots & \beta_{2m} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\alpha_m & \beta_{m1} & \beta_{m2} & \cdots & \beta_{mm} \\
\hline
& A_1 & A_2 & \cdots & A_m
\end{array},$$

де для явного методу РК всі $\beta_{ij} = 0, i \leq j, i = \overline{1, m}$. Ці параметри знаходять із умови, що $\psi_n^{(1)} = O(h_n^{(p)})$ - вимагаємо певний порядок точності p , причому бажано як можна вищого порядку. На практиці методи Рунге-Кутта з $m > 5$ не використовуються, це зумовлено громіздкістю формул і вимогами наявності високої гладкості у $f(x, u)$.

При певному представленні головної частини похибки можна будувати не тільки *односторонні* методи РК, а і так звані *двосторонні* методи РК

Розглянемо більш детально окремі односторонні методи типу Рунге-Кутта.

При $m=1$ формули методу Рунге-Кутта співпадають з явним методом Ейлера і мають точність $O(h)$.

При $m=2$, згідно схеми побудови, маємо:

$$\begin{cases} K_1 = f(x_n, y_n), & K_2 = f(x_n + \alpha_2 h_n, y_n + \beta_{21} h_n K_1), \\ y_{n+1} = y_n + h_n (A_1 K_1 + A_2 K_2). \end{cases} \quad (2.8)$$

Запишемо $\psi_n^{(1)}$ для (2.8), виконаємо розклади в ряд по h_n і необхідні перетворення:

$$\begin{aligned}
\psi_n^{(1)} &= -\frac{u_{n+1} - u_n}{h_n} + [A_1 f_n + A_2 f(x_n + \alpha_2 h_n, u_n + \beta_{21} h_n f_n)] = \\
&= -(u'_n + 0.5 h_n u''_n + O(h_n^2)) + \left[A_1 f_n + A_2 \left(f_n + \alpha_2 h_n \frac{\partial f_n}{\partial x} + \beta_{21} h_n f_n \frac{\partial f_n}{\partial u} \right) + O(h_n^2) \right] = \\
&= -\left[u'_n + 0.5 h_n \left(\frac{\partial f_n}{\partial x} + \frac{\partial f_n}{\partial u} u'_n \right) \right] + \left[(A_1 + A_2) f_n + A_2 \left(\alpha_2 h_n \frac{\partial f_n}{\partial x} + \beta_{21} h_n f_n \frac{\partial f_n}{\partial u} \right) \right] + O(h_n^2) = \\
&= -u'_n + (A_1 + A_2) f_n + h_n \left[(A_2 \alpha_2 - 0.5) \frac{\partial f_n}{\partial x} + (A_2 \beta_{21} - 0.5) f_n \frac{\partial f_n}{\partial u} \right] + O(h_n^2).
\end{aligned}$$

При виконанні тільки умови апроксимації $A_1 + A_2 = 1$ маємо точність $O(h_n)$, а отже для отримання другого порядку апроксимації необхідно вимагати для параметрів (2.8) виконання таких рівностей (система трьох рівнянь з чотирма невідомими):

$$\begin{cases} A_1 + A_2 = 1, \\ A_2 \alpha_2 = 0.5, \\ A_2 \beta_{21} = 0.5. \end{cases}$$

Якщо позначити $A_2 = \gamma, \gamma \in (0, 1]$, то решта параметрів однозначно визначаються $A_1 = 1 - \gamma, \alpha_2 = 0.5/\gamma, \beta_{21} = 0.5/\gamma$, а отже маємо однопараметричну сім'ю двоетапних методів РК другого порядку точності:

$$\begin{aligned} \gamma &\in (0, 1]; \quad A_1 = 1 - \gamma, \quad A_2 = \gamma; \\ K_1 &= f(x_n, y_n), \quad K_2 = f\left(x_n + \frac{h_n}{2\gamma}, y_n + \frac{h_n}{2\gamma} K_1\right); \\ y_{n+1} &= y_n + h_n (A_1 K_1 + A_2 K_2), \quad y_0 = u_0, \quad n = 0, 1, \dots \end{aligned} \quad (2.9)$$

Найбільш часто використовуються такі значення: $\gamma = 0.5$, $\gamma = 1$.

При $m=4$ для отримання четвертого порядку апроксимації необхідно знайти розв'язок системи з одинадцяти рівнянь, яка містить тринадцять невідомих. Тому тут існує декілька сімейств розрахункових формул. Найбільш популярний з чотирьохстадійних методів Рунге-Кутта четвертого порядку точності є:

$$\begin{aligned} A_1 &= A_4 = \frac{1}{6}, \quad A_2 = A_3 = \frac{1}{3}; \quad \bar{h} = 0.5h_n; \\ K_1 &= f(x_n, y_n), \quad K_2 = f(x_n + \bar{h}, y_n + \bar{h}K_1), \\ K_3 &= f(x_n + \bar{h}, y_n + \bar{h}K_2), \quad K_4 = f(x_{n+1}, y_n + h_n K_3); \\ y_{n+1} &= y_n + h_n (A_1 K_1 + A_2 K_2 + A_3 K_3 + A_4 K_4), \quad y_0 = u_0, \quad n = 0, 1, \dots \end{aligned} \quad (2.10)$$

Збіжність і порядок точності методів типу РК гарантує наступна

Теорема 2.1 Нехай права частина рівняння (2.1) $f(x, u)$ задовольняє умові Ліпшиця по змінній u з константою L , а нев'язка методу Рунге-Кутта визначається по формулі

$$\psi_n^{(1)} = -\frac{u_{n+1} - u_n}{h} + \sum_{i=1}^m A_i K_i$$

Тоді для похибки методу при $nh \leq T$ має місце оцінка

$$|y_n - u(x_n)| \leq T \exp(\alpha T) \max_{1 \leq j \leq n-1} |\psi_j^{(1)}|,$$

де $\alpha = A \cdot L \cdot m \cdot (1 + L \cdot \beta \cdot h)^{m-1}$, $A = \max_{1 \leq i \leq m} |A_i|$, $\beta = \max_{\substack{2 \leq i \leq m \\ 1 \leq j \leq i-1}} |\beta_{ij}|$.

Доведення див. [33].

Наслідок. Якщо метод Рунге-Кутта апроксимує вихідне рівняння, то він збігається при $h \rightarrow 0$, причому порядок точності співпадає з порядком апроксимації.

Якими розрахунковими формулами із існуючої множини методів РК потрібно користуватися при розв'язанні практичних задач? Одним з головних критеріїв – це питання існування і обмеженість похідних від функції $f(x, u)$. Якщо ця функція має такі похідні до четвертого порядку включно, то орієнтуємося на чотирьохстадійні схеми РК, які мають досить малий коефіцієнт в залишковому члені мажорантної оцінки і швидке зростання точності при зменшенні кроку. Якщо $f(x, u)$ не має вказаних похідних, то краще використовувати двохстадійні схеми РК.

п.4. Деякі загальні аспекти використання чисельних методів

Як отримати наближений розв'язок з заданою точністю?

При реалізації розглянутих методів ми прагнемо отримати розв'язок з певною величиною точності $0 < \varepsilon < 1$. Апріорні оцінки точності методів для цього малоефективні – вони громіздкі, вимагають знання оцінки похідних від розв'язку і як правило є мажорантними. Тому основним практичним засобом досягти поставленої цілі є апостеріорна оцінка точності. Вона досягається завдяки процедурі «автоматичного вибору кроку», при цьому суттєво використовують *формули Рунге-Ромберга* на зв'язаних сітках з урахуванням порядку точності p чисельного методу.

Використовувати явний чи неявний метод?

Явний метод простий в реалізації, але є певні обмеження на його застосування з точки зору *асимптотичної стійкості*. Неявні методи таких обмежень не мають, але

виникають непрості питання при знаходженні коренів нелінійних рівнянь виду (2.4) або розв'язків відповідних систем N нелінійних рівнянь. Так, можливі випадки неіснування дійсного кореня рівняння (2.4) при певній величині кроку h_n . Якщо корінь є, то як його знайти? Використання ІМ Ньютона вимагає диференціювати $f(x, u)$. Метод дихотомії не узагальнюється на системи N нелінійних рівнянь. Застосування методу послідовних наближень (МПН) вимагає досить малого кроку для збіжності цього методу. Крім цього значну роль відіграє співвідношення кількості ітерацій ІП і точність p відповідної неявної схеми. Розглянемо такий приклад. Нехай маємо рівняння (2.1) у вигляді $u' = f(u)$, а чисельний метод – неявна "симетрична схема" (2.5) з точністю $O(h^2)$, тоді МПН має вигляд

$$\begin{cases} y_{n+1}^{(0)} = y_n, \\ y_{n+1}^{(s)} = y_n + 0.5h_n(f(y_n) + f(y_{n+1}^{(s-1)})), \\ s = 1, 2, \dots, (s > \max s) \vee |y_{n+1}^{(s)} - y_{n+1}^{(s-1)}| \leq \varepsilon_1 \end{cases} \quad (2.11)$$

Запишемо явний вигляд першої і другої ітерації ІП (2.11), маємо:

$$\begin{aligned} y_{n+1}^{(1)} &= y_n + 0.5h_n(f(y_n) + f(y_n)) \equiv y_n + h_n f(y_n), \\ y_{n+1}^{(2)} &= y_n + 0.5h_n(f(y_n) + f(y_{n+1}^{(1)})) \equiv y_n + 0.5h_n(f(y_n) + f(y_n + h_n f(y_n))). \end{aligned}$$

Таким чином, в першій ітерації реалізується явний метод Ейлера з першим порядком точності, а в другій ітерації реалізується метод Рунге-Кутта «предиктор-коректор» з другим порядком точності. Наступні ітерації не збільшують порядок точності, так як він не може бути більший, ніж закладений в схемі (2.5). Вони впливають тільки на коефіцієнт в залишковому члені і збільшують час розрахунку! Таким чином, замість схеми (2.5) і ітераційного процесу (2.11), краще використати метод РК з точністю $O(h^2)$.

Що використовувати при «екзотиці»?

Наприклад, при наступних випадках:

- ✚ функції $f(x, u)$ мають достатню гладкість, але швидкозмінні;
- ✚ є особливі точки, в яких $f(x, u)$ або якісь її похідні $= \infty$.

Для подібних прикладних задач необхідно використовувати *спеціальні методи*, які розробляються для таких вузьких класів задач. Побудові таких методів передують вивчення і використання властивостей загальних розв'язків досліджуваних класів рівнянь..

ЛЕКЦІЯ № 03

ТЕМА № 2. Методи розв'язування задачі Коші для звичайних диференціальних рівнянь.

Продовжимо розглядати різницеві методи розв'язання задачі Коші для скалярного варіанту ЗДР (2.1), але тепер багатокрокові.

п.5. Різницеві багатокрокові

Багатокрокові різницеві методи отримують при апроксимації в рівнянні (2.1) лівої $u'(x)$ або правої $f(x, u)$ або обох частин одночасно $u'(x) \& f(x, u)$ в системі точок шаблону $S\{x_{n-m}, \dots, x_{n-1}, x_n\}$. В загальному на сукупності вузлів сітки маємо:

$$\frac{a_0 y_n + a_1 y_{n-1} + \dots + a_m y_{n-m}}{h} = b_0 f_n + b_1 f_{n-1} + \dots + b_m f_{n-m}, \quad (2.12)$$

де $a_0 \neq 0$; $\sum_{i=0}^m b_i = 1$ – умова апроксимації $f(x, u)$; $f_i = f(x_i, y_i)$; $a_i, b_i \in \text{Const}$.

При $b_0 = 0$ – явна, $b_0 \neq 0$ – неявна різницева схема. У випадку, якщо $a_0 = -a_1 = 1$, а решта $a_i = 0$, $i = 2, m$, то маємо методи Адамса.

Для використання m -крокових різницевих методів, мають бути відомі значення сіткової функції в точках «розгону» $(x_{n-m}, x_{n-m+1}, \dots, x_{n-1})$

$$y_{n-m} = \mathcal{Y}_1, y_{n-m+1} = \mathcal{Y}_2, \dots, y_{n-1} = \mathcal{Y}_m. \quad (2.13)$$

Серед множини m -крокових різницевих методів найбільш поширеними є методи Адамса, Гіра.

Алгоритми явного/неявного методу Адамса

Нехай нам відомі значення (2.13) в точках «розгону». Будемо розглядати функцію $f(x, u)$ не на всій площині її аргументів, а на певній інтегральній кривій фіксованій початковим значенням, тоді маємо $f(x, u(x)) \equiv F(x)$ – функцію одного аргументу. Проінтегруємо (2.1) на інтервалі $[x_{n-1}, x_n]$

$$y_n - y_{n-1} = \int_{x_{n-1}}^{x_n} f(\xi, u(\xi)) d\xi \equiv \int_{x_{n-1}}^{x_n} F(\xi) d\xi \quad (2.14)$$

Для знаходження інтегралу в (2.14) можна використати наступні способи:

- 1) заміна функції $F(\xi)$ на інтерполяційний поліном (ІП), побудований по точкам шаблону S з подальшим точним обчисленням інтегралу;
- 2) використання квадратурних формул, а для визначення коефіцієнтів КФ – теорему про інтерполяційні КФ;
- 3) мінімізувати похибку методу, порівнюючи розклад в ряд Тейлора точного приросту функції $u(x)$ і наближення інтеграла в (2.14) по формальній КФ.

Використаємо перший спосіб, з рештою можна ознайомитися [19, с. 160-162]. Якщо нам відомі значення $\{y_i\}$ в вузлах сітки $\{x_{n-m}, \dots, x_{n-1}, x_n\}$, то відомі і значення функції $F(\xi)$ в цих вузлах, а отже можна побудувати ІП (в формі Лагранжа чи Ньютона) по значенням $\{x_{n-j}; F(x_{n-j})\}_{j=0}^m$. При $j=1$ отримаємо явний (екстраполяційний) метод Адамса, а при $j=0$ – неявний (інтерполяційний) метод Адамса.

Враховуючи ці обставини, запишемо загальний вигляд ІП у формі Ньютона для отримання:

$$\begin{aligned} \text{а) явного методу ітераційн} \\ F(\xi) = F(x_{n-1}) + (\xi - x_{n-1})F(x_{n-1}; x_{n-2}) + (\xi - x_{n-1})(\xi - x_{n-2})F(x_{n-1}; x_{n-2}; x_{n-3}) + \dots \\ + (\xi - x_{n-1}) \dots (\xi - x_{n-m+1})F(x_{n-1}; x_{n-2}; \dots; x_{n-m}), \end{aligned} \quad (2.15a)$$

б) неявного методу

$$F(\xi) = F(x_n) + (\xi - x_n)F(x_n; x_{n-1}) + (\xi - x_n)(\xi - x_{n-1})F(x_n; x_{n-1}; x_{n-2}) + \dots + (\xi - x_n) \dots (\xi - x_{n-m+1})F(x_n; x_{n-1}; \dots; x_{n-m}), \quad (2.15b)$$

де $F(x_i; x_j; \dots; x_k) \equiv F_{i,j,\dots,k}$ – поділені різниці.

Використовуючи ці формули, можна отримати розрахункові формули явного/неявного методу Адамса певного порядку точності p як на рівномірній сітці з кроком h , так і на нерівномірній сітці.

Легко зрозуміти, що для першого порядку точності, методи Адамса співпадають з відповідними методами Ейлера.

Для явного/неявного методу Адамса другого порядку точності на рівномірній сітці з кроком h маємо:

Таблиця 2.1

<i>Назва</i>	<i>Явний (екстраполяційний)</i>	<i>Неявний (інтерполяційний)</i>
<i>Шаблон</i>	$S[x_{n-2}, x_{n-1}, x_n]$	$S[x_{n-1}, x_n]$
<i>ІП для F</i>	$F(\xi) = F_{n-1} + (\xi - x_{n-1})F_{n-1;n-2}$	$F(\xi) = F_n + (\xi - x_n)F_{n;n-1}$
<i>Розрахункова формула</i>	$y_n = y_{n-1} + 0.5h(-f_{n-2} + 3f_{n-1})$	$y_n = y_{n-1} + 0.5h(f_{n-1} + f_n)$
<i>Похибка</i>	$O(h^2)$	$O(h^2)$
<i>Порядок p через значення m, враховуючи $\sum_{k=0}^m b_k f_{n-k}$</i>	$2 = m$	$2 = m+1$

Розглянемо «плюси» і «мінуси» явних методів Адамса.

«Мінуси»:

1. для визначення $\{g_i\}_{i=1}^m$ в точках «розгону» необхідні додаткові методи того-ж порядку точності, що і метод Адамса, який ми використовуємо. Це означає, що при розв'язанні задачі (2.1) на ЕОМ, необхідно одночасно використовувати два варіанти програм («Адамс» і «не Адамс»).
2. формули розрахунку зі змінним кроком громіздкі, а використання простих розрахункових формул з постійним кроком вимагає нестандартних дій при зміні кроку (для процедури «автоматичного вибору кроку»). А саме, перехід до формул з нерівномірним кроком, виконання по ним p -кроків (для методу p -го порядку точності) і повернення до формул з рівномірним кроком. Це теж ускладнює програмний код реалізації методів Адамса.

«Плюси»:

1. за один крок методу Адамса необхідно обчислювати тільки **один** раз функцію $f(x, u)$, що є суттєво при значній складності її обрахунку. Але, якщо порівняти залишкові члени в оцінці точності, наприклад, для методу Адамса порядку $O(h^4)$ і методу Рунге-Кутта того-ж порядку точності (**чотири** рази звертання до $f(x, u)$!), то маємо відношення **960:1**. Значить, при однаковій точності, схема Рунге-Кутта дозволяє обирати крок в $\sqrt[4]{960} \approx 5.57$ раз більший, тобто фактично звертань до $f(x, u)$ потрібно менше, ніж в методі Адамса !

Тому зараз явні методи Адамса і аналогічні їм (Мілна, Коуелла) не мають практичного застосування, особливо в вигляді стандартного ПЗ на ЕОМ.

А як з неявними методами Адамса ? Відповідь отримаємо згодом, після розгляду деяких важливих питань.

п.6. Стійкість і збіжність різницевого m -крокового методів

Розглянемо для (2.12) однорідне рівняння

$$a_0 v_n + a_1 v_{n-1} + \dots + a_m v_{n-m} = 0. \quad (2.16)$$

Його розв'язок будемо шукати у вигляді $v_k = q^k$ (якщо деякі корені є кратні, то розв'язок буде $v_k = k^j q^k$, $j = 0, 1, \dots, r_k - 1$, де r_k – кратність кореня) і для q отримуємо *характеристичне рівняння*:

$$a_0 q^m + a_1 q^{m-1} + \dots + a_{m-1} q + a_m = 0. \quad (2.17)$$

Озн.2.4. Метод (2.12) задовольняє умові коренів, якщо $|q_i| \leq 1$, $i = 1, 2, \dots, m$, причому $|q_{i_0}| = 1$ не є кратним.

Озн.2.5. Метод (2.12), при виконанні умови коренів є *стійким*.

Озн.2.6. Рівняння (2.17) *стійке по початковим даним*, якщо існує постійна M_1 , яка не залежить від n і така, що при довільних початкових даних $\{g_i\}_{i=1}^m$ для його розв'язку виконується оцінка

$$|v_n| \leq M_1 \max_{0 \leq j \leq m-1} |v_j|, \quad n = m, m+1, \dots$$

Теорема 2.2 Умова коренів необхідна і достатня для стійкості рівняння (2.17) по початковим даним.

Доведення див. [33].

Теорема 2.3 (*стійкість по правій частині*) Якщо рівняння (2.17) стійке по початковим даним, то для неоднорідного рівняння

$$a_0 y_n + a_1 y_{n-1} + \dots + a_m y_{n-m} = h g_{n-m}$$

справедлива оцінка

$$|y_n| \leq M_1 \max_{0 \leq j \leq m-1} |y_j| + M_2 h \sum_{k=0}^{n-m} |g_k|, \quad n = m, m+1, \dots$$

де постійні M_1, M_2 не залежить від n .

Доведення див. [5, с. 240-244].

Як слідує із Теор.2.2-2.3, умова коренів є основним чинником відповідних означень *стійкості по початковим даним* і *по правій частині*.

Теорема 2.4 Якщо для рівняння (2.12) виконується умова коренів і для правої частини умова Ліпшиця $|f'_u(x, u)| \leq L$ для $x \in (x_0, X]$, то при обмеженні на крок

$$h \leq h_0 = \frac{|a_0|}{2|b_0|L}$$

для $|z_n| = |y_n - u(x_n)|$ – похибки розв'язку рівняння (2.12) справедлива оцінка

$$|z_n| \leq M_1 \exp(c_1 X) \max_{0 \leq j \leq m-1} |z_j| + M_2 h \sum_{k=0}^{n-m} |\psi_k|, \quad n = m, m+1, \dots$$

де $M_1 = \|Q\|_c \|Q^{-1}\|_c$, $M_2 = \frac{2M_1}{|a_0|}$, $c_1 = M_1 w L$, $w = \frac{2}{a_0^2} \sum_{k=1}^m (|a_0| |b_k| + |a_k| |b_0|)$, $\hat{S} = Q S Q^{-1}$,

Q – матриця перетворення подібності, S – Фробеніусова матриця для характеристичного рівняння (2.17), $\hat{S}$ – модифікована Жорданова матриця.

Доведення див. [33].

Як наслідок Теор.2.4 маємо, що $|z_j| = |y_j - u(x_j)| \rightarrow 0$ при $h \rightarrow 0$ (сходиться до розв'язку) і *порядок точності* методу (2.12) співпадає з порядком апроксимації на розв'язку вихідного рівняння.

Таким чином, дослідження збіжності методу (2.12) зводиться до аналізу похибки апроксимації і перевірки умови коренів (УК). Розглянемо приклади:

- 1) методи Ейлера дають характеристичне рівняння $q-1=0$, $q_1=1$ – не кратний і УК виконується, отже збіжність з першим порядком;
- 2) методи Адамса мають характеристичне рівняння $q-1=0$, $q_1=1$ – не кратний і УК виконується, отже збіжність з порядком апроксимації;
- 3) схема порядку $O(h^2)$ $\frac{y_n - y_{n-2}}{2h} = f_{n-1}$ має характеристичне рівняння $q^2 - 1 = 0$, $q_{1,2} = \pm 1 \Rightarrow$ не виконується УК – метод нестійкий.

п.7. Стійкість умовна і абсолютна. Жорсткість систем ЗДР.

Деякі методи розв'язування задачі Коші для ЖС ЗДР

Якщо розглядати модельну задачу Коші для скалярного варіанту (2.1) з функцією $f(x, u(x)) = \lambda \cdot u(x)$, $x > 0$, $\lambda < 0$, то стійкість означає виконання нерівності

$$|u(x+h)| \leq |u(x)|, \quad h > 0.$$

Використовуючи різницевий метод, потрібно вимагати, щоб для його розв'язку виконувалась нерівність, аналогічна диференціальному розв'язку, тобто була *асимптотична стійкість*.

Озн.2.7. Різницевий метод є *абсолютно стійким* при $\forall h > 0$, і *умовно стійким*, якщо він стійкий при певних обмеженнях на крок h .

Застосуємо Озн.2.7 для розглянутих нами різницевих методів з урахуванням модельної задачі Коші.

а) Для явного методу Ейлера (2.2): $|y_{n+1}| \leq |1 + \lambda h| |y_n|$ і маємо:

$$0 < h|\lambda| \leq 2 \quad (2.18)$$

б) Аналогічне обмеження на крок можна отримати для методу Рунге-Кутта 2-го порядку точності ($\gamma=1$), а для 4-го порядку точності – $|\lambda|h \leq 2.78$.

в) Для неявного м.Ейлера: $|y_{n+1}| \leq \left| \frac{1}{1-\lambda h} \right| |y_n|$ і маємо завжди $\frac{1}{1-\lambda h} < 1$ при $0 < h$ & $\lambda < 0$.

Із наведених прикладів бачимо, що явні методи умовно стійкі, а неявні – безумовно стійкі, але вимагають розв'язання нелінійних рівнянь.

Розглянуті нами явні методи без зусиль переносяться на системи ЗДР, але виникає проблема зі стійкістю, пов'язана з різномасштабністю процесів, які описуються системою ЗДР. Простий приклад пояснює характер цих проблем при чисельних розрахунках:

$$\begin{cases} u_1'(x) + a_1 u_1(x) = 0, \\ u_2'(x) + a_2 u_2(x) = 0. \end{cases}$$

$$x > 0; \quad a_{1,2} > 0, \quad a_2 \gg a_1.$$

Або у векторній формі запису :

$$\frac{d\vec{u}(x)}{dx} = \mathbf{A} \cdot \vec{u}(x), \quad \mathbf{A} = \begin{pmatrix} -a_1 & 0 \\ 0 & -a_2 \end{pmatrix}, \quad \vec{u}(x) = \begin{pmatrix} u_1(x) \\ u_2(x) \end{pmatrix}. \quad (2.19)$$

Враховуючи, що $a_{1,2} > 0$, $a_2 \gg a_1$, з деякого x ми повинні мати (аналітично)

$$\vec{u}(x) = \begin{pmatrix} u_1(x) \\ u_2(x) \end{pmatrix} \approx \begin{pmatrix} u_1(x) \\ 0 \end{pmatrix}, \quad \text{тобто поведінка } \vec{u}(x) \text{ повністю визначається компонентою}$$

$u_1(x)$. Проте чисельно, якщо використовувати явний метод Ейлера, ми маємо обмеження на крок у вигляді нерівності (2.18), що приводить до $h a_2 \leq 2$, тобто

неочікуваний вплив компоненти $u_2(x)$! Одним із варіантів вирішення цієї парадоксальної ситуації є використання безумовно стійких неявних методів.

Озн.2.7. Нехай матриця A – постійна, квадратна, розмірності N . Система (2.19) називається *жорсткою*, якщо:

1) $\operatorname{Re}(\lambda_k) < 0, k = 1, 2, \dots, N$ - система асимптотично стійка по Ляпунову;

2) $S = \frac{\max_k |\operatorname{Re}(\lambda_k)|}{\min_k |\operatorname{Re}(\lambda_k)|}$ - число жорсткості системи велике.

Означення жорсткості можна поширити на лінійні системи ЗДР, в яких матриця A залежить від x , а також на нелінійні системи (2.1). Крім асимптотичної стійкості можуть визначатись спеціальні умови стійкості: A -*стійкість* і $A(\alpha)$ -*стійкість* [33].

Для чисельного інтегрування жорстких систем ЗДР досить часто використовують *метод Гіра* – чисто неявний m -кроковий різницьевий метод (частинний випадок (2.12) при $b_0 = 1, b_i = 0, i = \overline{1, m}$)

$$\frac{a_0 y_n + a_1 y_{n-1} + \dots + a_m y_{n-m}}{h} = f(x_n, y_n), \quad (2.20)$$

який має порядок апроксимації $p = m$, при виконанні умов [33]:

$$a_0 = -\sum_{k=1}^m a_k, \quad \begin{cases} a_1 + 2a_2 + \dots + ma_m = -1, \\ a_1 + 2^2 a_2 + \dots + m^2 a_m = 0, \\ \dots \\ a_1 + 2^m a_2 + \dots + m^m a_m = 0 \end{cases}$$

Система лінійних рівнянь відносно коефіцієнтів $a_i, i = \overline{1, m}$ має єдиний розв'язок, враховуючи відмінність від нуля її визначника.

При $m = 1$ маємо неявний метод Ейлера (2.3).

Для $m = 2, 3, 4$ отримаємо такі розрахункові формули:

$$\frac{3y_n - 4y_{n-1} + y_{n-2}}{2h} = f(x_n, y_n), \quad (2.21)$$

$$\frac{11y_n - 18y_{n-1} + 9y_{n-2} - 2y_{n-3}}{6h} = f(x_n, y_n), \quad (2.22)$$

$$\frac{25y_n - 48y_{n-1} + 36y_{n-2} - 16y_{n-3} + 3y_{n-4}}{12h} = f(x_n, y_n), \quad (2.23)$$

які мають, відповідно, другий, третій і четвертий порядок точності. На практиці використовують методи Гіра до десятого порядку точності.

Для розв'язання нелінійного рівняння (2.20) використовують ітераційні процеси, наприклад, метод Ньютона [1–3, 16, 22, 29, 30].

Важливо відмітити, що методи Гіра мають гарні умови стійкості. Для прикладу розглянемо метод другого порядку апроксимації (2.21) з модельною функцією $f(x, u(x)) \equiv \lambda \cdot u(x), x > 0, \lambda < 0$:

$$1.5y_n - 2y_{n-1} + 0.5y_{n-2} = \mu \cdot y_n, \quad (2.24)$$

де $\mu = h\lambda$. Характеристичне рівняння, яке відповідає (2.24), буде

$$(1.5 - \mu)q^2 - 2q + 0.5 = 0. \quad (2.25)$$

Отже, потрібно знайти множину точок G комплексної площини $\mu = \mu_R + i\mu_I$, для яких виконується умова коренів $|q_{1,2}(\mu)| < 1$ – це зовнішня частина області площини,

яка визначається кривою M . Для визначення цієї кривої, виразимо із рівняння (2.25) параметр μ через змінну q :

$$\mu = 1.5 - 2q^{-1} + 0.5q^{-2}. \quad (2.26)$$

Звідси крива M буде визначатись при $|q|=1$ (як границя області G), тобто, при $q = \exp(-i\varphi)$ маємо рівняння цієї кривої

$$\mu(\varphi) = 1.5 - 2\exp(i\varphi) + 0.5\exp(i2\varphi), \quad (2.27)$$

де $\varphi \in [0, 2\pi]$. Позначивши $v = \cos(\varphi)$, рівняння (2.27) можна переписати у вигляді

$$\mu = (1-v)^2 \pm i(2-v)\sqrt{1-v^2}, \quad v \in [-1, 1].$$

Звідки слідує, що крива M розташована в правій півплощині (див. Рис.2.1) і область стійкості методу (2.21) повністю містить ліву півплощину $\operatorname{Re}(\mu) < 0$, а отже метод A – стійкий.

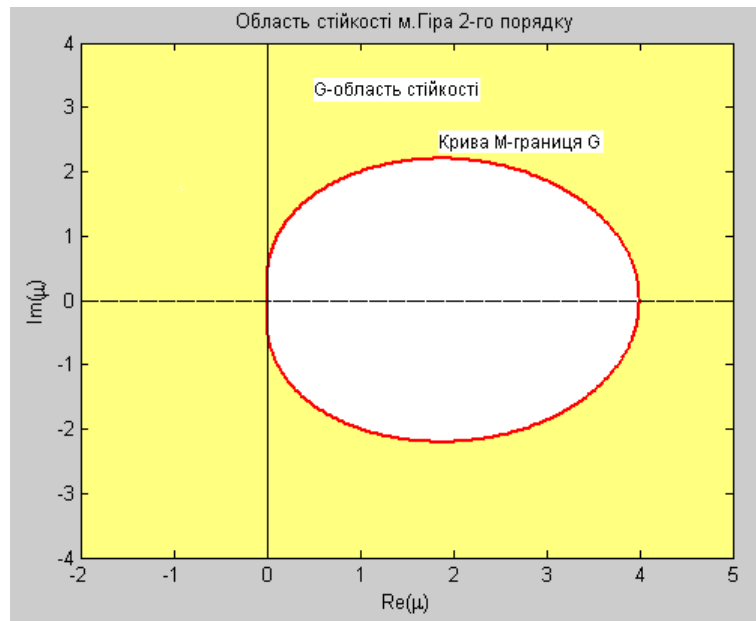


Рис. 2.1 G -область стійкості методу (2.21) – частина комплексної площини $\operatorname{Re}(\mu) + i\operatorname{Im}(\mu)$, розміщена зовні кривої M .

Якщо виконати дослідження стійкості методу (2.23) [33], то отримаємо, що він $A(\alpha)$ – стійкий, де $\alpha = \arctg(\sqrt{6}) \approx 67.79^\circ$.

Для ефективного чисельного розв’язання задачі Коші (2.1) з жорсткістю також використовуються:

- ✚ методи Розенброка;
- ✚ діагонально неявні методи Рунге-Кутта.

В математичному пакеті MATLAB для задач з жорсткістю існують солвери, які реалізують методи Гіра і Розенброка. Щодо нежорстких задач (2.1), то в MATLAB представлені солвери, які реалізують методи типу Рунге-Кутта.

ЛЕКЦІЯ № 04

ТЕМА № 3. Різницеві методи розв'язування крайових задач для ЗДР.

Для чисельного розв'язання диференціальних рівнянь звичайних і в частинних похідних, як лінійних так і нелінійних, застосовується потужний метод *сіток* або *різницевий метод*. Розглянемо основні ідеї методу різницевих схем (МРС) на прикладі лінійної крайової задачі, записаної в операторному вигляді:

$$Lu(x) = f(x), \quad x \in D; \quad (3.1)$$

$$lu(x) = g(x), \quad x \in \partial D. \quad (3.2)$$

Тут $Lu(x)$, $lu(x)$ – деякі лінійні диференціальні оператори; $f(x)$, $g(x)$ – функції відповідної гладкості, які описують праву частину ЗДР і крайові умови; $\bar{D} = D \cup \partial D \equiv [a, b]$ – область (відрізок), де шукається розв'язок. Для (3.1)-(3.2) виконуються відповідні умови існування та єдиності (задача поставлена коректно).

п.1. Основні поняття методу скінченних різниць

Основні принципи МРС полягають в систематичній дискретизації всього, що фігурує в постановці задачі (3.1)-(3.2).

1. Область $\bar{D}$ заміняємо сітковою областю $D_h \cup \partial D_h$. Вводиться *сітка* $\bar{D}_h$ – довільна скінченна множина (неспівпадаючих) точок (*вузлів сітки*):

$$\text{нерівномірна } \bar{D}_h = \{a = x_1 < x_2 < \dots < x_N = b, \quad h_k = x_{k+1} - x_k\}; \quad (3.3)$$

$$\text{або рівномірна } \bar{D}_h = \{x_k = a + (k-1)h, \quad k = 1, 2, \dots, N, \quad b - a = (N-1)h\}. \quad (3.4)$$

2. Функцій, як відомі, так і невідомі, заміняємо сітковими функціями, при цьому будемо позначати $f(x_k) = f_k$, $g(x_k) = g_k$; $u(x_k) = u_k \equiv y_k$, $x_k \in \bar{D}_h$.

3. Диференціальні оператори заміняємо на різницеві $Lu(x) \rightarrow L_h u$, $lu(x) \rightarrow l_h u$ – зважені суми $\sum_{x_j \in S\{x_k\}} c_j u_j$ шуканої сіткової функції на певній сукупності вузлів

(шаблоні) $S\{x_k\} \in \bar{D}_h$. Найчастіше для цього використовують:

- ✓ метод безпосередньої заміни диференціальних операторів на формули числового диференціювання (метод різницевої апроксимації);
- ✓ інтегро-інтерполяційний метод (метод балансу);
- ✓ варіаційно-різницевий метод (метод скінченних елементів).

Можуть використовуватись і інші методи: невизначених коефіцієнтів [19, с.201-202; 20], суматорних тотожностей і апроксимації квадратичного функціонала [33].

4. Для отримання певних теоретичних оцінок про побудовану *різницеву схему*:

$$L_h u = \varphi(x_k), \quad x_k \in D_h; \quad (3.5)$$

$$l_h u = \mu(x_k), \quad x_k \in \partial D_h. \quad (3.6)$$

– сім'ю систем лінійних алгебраїчних рівнянь (СЛАР) з параметром h , вводимо сіткові норми $\|\cdot\|_h$, відповідний аналог неперервної норми $\|\cdot\|$, такий, щоб при $h \rightarrow 0 \Rightarrow \|\cdot\|_h \rightarrow \|\cdot\|$.

Для отримання наближеного дискретного розв'язку задачі (3.1)-(3.2) необхідно розв'язати СЛАР (3.5)-(3.6), а при цьому виникають наступні питання:

- 1) чи існує її розв'язок?
- 2) як фактично знаходити розв'язок: який алгоритм реалізувати, чи буде він ефективним при великих значеннях N ?

І звичайно для різницевої схеми (РС) суттєвим є ще одне питання: чи буде сіткова функція збігатися до точного розв'язку при $h \rightarrow 0$. Все це вимагає певних теоретичних досліджень РС.

Побудова і аналіз РС у МСР базується на таких поняттях, як *апроксимація*, *стійкість* та *збіжність*.

Озн.3.1. Похибкою апроксимації оператора L оператором L_h називають різницю

$$\psi(x_k) = L_h u - (Lu)_h, \quad (3.7)$$

де $(Lu)_h$ – проектор неперервного оператора на сітку.

Озн.3.2. Оператор L_h має m -й порядок апроксимації в точці $x_k \in D_h$ (локальний), якщо

$$\psi(x_k) = O(h^m) \text{ або } |\psi(x_k)| = Mh^m, \quad m > 0, \quad (3.8)$$

де $M, m \in \text{Const}$ і не залежать від кроку сітки.

Озн.3.3. Оператор L_h має m -й порядок апроксимації на сітці, тобто в деякій сітковій нормі, якщо

$$\|\psi(x_k)\|_h = O(h^m), \quad m > 0. \quad (3.9)$$

Озн.3.4. Сіткова функція $z(x_k) \equiv z_k = y_k - (u)_h$, називається *похибкою різницевої схеми в точці* $x_k \in D_h$.

Якщо підставимо $y_k = z_k + (u)_h$, в рівняння (3.5) $L_h u \equiv L_h z_k + L_h (u)_h = \varphi(x_k)$, то отримаємо, що похибка задовольняє рівняння

$$L_h z_k = \psi_h(x_k). \quad (3.10)$$

Озн.3.5. Сіткова функція $\psi_h(x_k) = \varphi(x_k) - L_h(u)_h$ називається *похибкою апроксимації різницевого рівняння (3.5) на розв'язку диференціального рівняння (3.1) в точці* $x_k \in D_h$.

Виконаємо над функцією $\psi_h(x_k)$ ланцюжок очевидних перетворень

$$\psi_h(x_k) = 0 + \varphi(x_k) - L_h(u)_h = (Lu - f)_h + \varphi(x_k) - L_h(u)_h = [(Lu)_h - L_h(u)_h] + [\varphi(x_k) - (f)_h]$$

і в результаті представимо у наступному вигляді:

$$\psi_h(x_k) = \psi_{h,1}(x_k) + \psi_{h,2}(x_k),$$

де

$$\psi_{h,1}(x_k) = (Lu)_h - L_h(u)_h, \quad \psi_{h,2}(x_k) = \varphi(x_k) - (f)_h.$$

Озн.3.6. Сіткові функції $\psi_{h,1}(x_k)$ і $\psi_{h,2}(x_k)$ називається відповідно *похибкою апроксимації диференціального оператора* (див.(3.7)) і *похибкою апроксимації правої частини*.

Озн.3.7. Різницеве рівняння (3.5) апроксимує диференціальне рівняння (3.1), якщо

$$\|\psi_h(x_k)\|_h \rightarrow 0 \text{ при } |h| \rightarrow 0 \quad (3.11)$$

і має p -й порядок апроксимації, якщо

$$\|\psi_h(x_k)\|_h \leq M_1 |h|^p, \quad p > 0, \quad (3.12)$$

де $M_1, p \in \text{Const}$ і не залежать від кроку сітки.

Аналогічно визначаються похибки апроксимації для крайових умов

$$\nu_h(x_k) = [(lu)_h - l_h(u)_h] + [\mu(x_k) - (g)_h]$$

і порядок її апроксимації

$$\|\nu_h(x_k)\|_h \leq M_2 |h|^r, \quad r > 0, \quad M_2, r \in \text{Const}. \quad (3.13)$$

Для дослідження сіткових функцій $\psi_h(x_k), \nu_h(x_k)$ використовують розвинення по h в ряд Тейлора в околі деякого сіткового вузла.

Озн.3.8. Різницева схема (3.5)- (3.6) називається *коректною*, якщо

1) її розв'язок існує і єдиний при довільних даних $\varphi(x_k), \mu(x_k)$;

2) є *стійкість*, тобто, існують $M_3, M_4 \in \text{Const}$, не залежні від кроку сітки, такі, що виконується оцінка (стійкість по початковим даним і по правій частині)

$$\|y_k\|_h \leq M_3 \|\varphi(x_k)\|_h + M_4 \|\mu(x_k)\|_h \quad (3.14)$$

Озн.3.9. Розв'язок різницевої задачі (3.5)-(3.6) *збігається* до розв'язку диференціальної задачі (3.1)-(3.2), якщо при $|h| \rightarrow 0$

$$\|z(x_n)\|_h \rightarrow 0 \quad (3.15)$$

і має q -й *порядок точності* $O(|h|^q)$, якщо

$$\|z(x_k)\|_h \leq \tilde{M}|h|^q, \quad q > 0, \quad (3.16)$$

де $\tilde{M}, q \in \text{Const}$ і не залежні від кроку сітки.

Теорема 3.1 (Лакса) Нехай лінійна диференціальна задача (3.1)-(3.2) поставлена коректно. Різницева задача (3.5)-(3.6) є коректною і апроксимує вихідну задачу на розв'язку $u(x)$. Тоді розв'язок різницевої задачі y_k збігається до розв'язку диференціальної задачі $u(x)$, причому порядок точності співпадає з порядком апроксимації.

Доведення. Враховуючи, що задача для похибки z_k записується у вигляді

$$L_h z_k = \psi_h(x_k);$$

$$l_h z_k = v_h(x_k),$$

і має ту ж структуру, що і різницева задача (3.5)-(3.6), а отже на основі її коректності маємо оцінку

$$\|z_k\|_h \leq M_3 \|\psi_h(x_k)\|_h + M_4 \|v_h(x_k)\|_h.$$

Підставимо сюди відповідні оцінки апроксимації (3.12), (3.13), отримаємо

$$\|z_k\|_h \leq M_3 M_1 |h|^p + M_4 M_2 |h|^r \leq \tilde{M} |h|^q, \quad q = \min(p, r).$$

Значення теореми Лакса: дослідження збіжності РС розпадається на два окремих етапи дослідження

а) доведення апроксимації РС на розв'язку вихідної задачі;

б) отримання оцінок типу (3.14) – *апріорних оцінок* для доведення стійкості.

п.2. Різницеві схеми для лінійних задач

Розглянемо застосування МРС на прикладі лінійної крайової задачі:

$$Lu(x) \equiv u''(x) - p(x)u(x) = -f(x), \quad x \in D \equiv (a, b); \quad (3.17)$$

$$\bar{D} = D \cup \partial D, \quad \partial D = a \cup b;$$

$$u(a) = d_1, \quad u(b) = d_2. \quad (3.18)$$

Будемо вважати, що задані функції $p(x) \geq 0$, $f(x)$ двічі неперервно диференційовані, а отже $u(x)$ має четверту неперервну похідну.

Введемо рівномірну сітку (3.4) і використаємо для побудови різницевої задачі (РЗ) метод безпосередньої заміни диференціальних операторів на формули числового диференціювання. Якщо на триточковому шаблоні $S\{x_{k-1}, x_k, x_{k+1}\}$ замінимо $u''(x_k)$ на $y_{\bar{x},k} - \text{другу різницеву похідну}$, то для (3.17) отримаємо РЗ

$$\frac{y_{k-1} - 2y_k + y_{k+1}}{h^2} - p_k y_k = -f_k, \quad k = \overline{2, N-1}; \quad (3.19)$$

$$y_1 = d_1, \quad y_N = d_2. \quad (3.20)$$

Отриману РЗ представимо у вигляді $A\bar{y} = -\bar{f}$ – СЛАР для сіткової функції $\bar{y} = (y_1, \dots, y_N)^T$:

$$\mathbf{A} = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 1 & -(2+h^2 p_2) & 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & -(2+h^2 p_3) & 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & \ddots & \dots & 0 & 0 \\ 0 & \dots & 0 & 1 & -(2+h^2 p_{N-2}) & 1 & 0 \\ 0 & \dots & 0 & 0 & 1 & -(2+h^2 p_{N-1}) & 1 \\ 0 & \dots & 0 & 0 & 0 & 0 & -1 \end{pmatrix}, \quad \vec{\mathbf{f}} = \begin{pmatrix} d_1 \\ h^2 f_2 \\ h^2 f_3 \\ \vdots \\ h^2 f_{N-2} \\ h^2 f_{N-1} \\ d_2 \end{pmatrix}.$$

Матриця $\mathbf{A}$ цієї СЛАР – тридіагональна і враховуючи $p(x) \geq 0$, маємо діагональну перевагу її елементів, а отже розв’язок (3.19)-(3.20) існує і єдиний. Для його знаходження використовують стійкий *метод монотонної лівої/правої прогонки*.

Дослідимо апроксимацію РЗ. Використовуючи розвинення сіткових функцій $\psi(x_k)$ (3.7) і $\psi_h(x_k)$ (3.10) в ряд Тейлора

$$\begin{aligned} \psi(x_k) &= L_h y - (Lu)_h = \frac{y_{k-1} - 2y_k + y_{k+1}}{h^2} - u''(x_k) = \frac{h^2}{12} u^{(4)}(x_k) + O(h^4) = O(h^2), \\ \psi_h(x_k) &= [(u''(x_k) - p(x_k)u(x_k)) - (y_{\bar{x},k} - p_k y_k)] + [-f_k + f(x_k)] = \\ &= \psi_{h,1}(x_k) + \psi_{h,2}(x_k) = -\frac{h^2}{12} u^{(4)}(x_k) = O(h^2) \end{aligned}$$

і точну апроксимацію крайових умов $v_h(x_k) = 0$, $x_k = a, b$ отримаємо другий порядок апроксимації.

Для отримання *априорних оцінок* (3.14) скористаємся *принципом максимуму*. З урахуванням того, що маємо наступні співвідношення

$$\|y_k\|_c = \max_{1 \leq k \leq N} |y_k| = \max\{|d_1|, |d_2|, \max_{1 \leq k \leq N} |y_k|\} = \max\{|d_1|, |d_2|, |y_i|\}, \quad (1 < i < N);$$

виконаємо перетворення в рівнянні (3.19), записаному в вузлі з індексом i

$$(2 + h^2 p_i) y_i = y_{i-1} + y_{i+1} + h^2 f_i;$$

Проведемо оцінки

$$(2 + h^2 p_i) |y_i| \leq |y_{i-1}| + |y_{i+1}| + h^2 |f_i| \leq 2|y_i| + h^2 |f_i|; \quad |y_i| \leq \left| \frac{f_i}{p_i} \right|,$$

а отже маємо стійкість РС у вигляді $\|y_k\|_c \leq \left\| \frac{f_k}{p_k} \right\|_c + \max(|d_1|, |d_2|)$

З отриманої оцінки і враховуючи вид $\psi_h(x_k)$, для похибки РС $z(x_k)$, $x_k \in \bar{D}_h$ маємо

нерівність $\|z_k\|_c \leq \frac{h^2}{12} \left\| \frac{u^{(4)}(x_k)}{p_k} \right\|_c$, а отже точність РС (3.19)-(3.20) є другого порядку.

Питання: чи можна на розглянутому триточковому шаблоні $S\{x_{k-1}, x_k, x_{k+1}\}$ отримати для задачі (3.17),(3.18) різницеву схему, точність якої більш високого порядку, ніж другий? Для спрощення викладок, обмежимося випадком $p(x) \equiv P \in \text{Const}$. Враховуючи, що крайові умови (3.18) не впливають на порядок РС, апроксимацію (3.18) запишемо наступним чином

$$y_{\bar{x},k} - \xi_k y_k = -\varphi_k,$$

де сіткові функції ξ_k , φ_k необхідно вибрати таким чином, щоб в (3.12) показник степеня був $p > 2$. Запишемо (3.12) і виконаємо ланцюжок нескладних перетворень з урахуванням розв’язку диференціальної задачі

$$\begin{aligned}
\psi_h(x_k) &= [(u''(x_k) - Pu(x_k)) - (y_{\bar{x}x,k} - \xi_k y_k)] + [-\varphi_k + f(x_k)] = \\
&= (-\varphi_k + f_k) - (P - \xi_k)u(x_k) - \frac{h^2}{12}u^{(4)}(x) + O(h^4) = \\
&= \left[\begin{aligned} u'' &= Pu - f, \\ u''' &= Pu' - f', \\ u^{(4)} &= Pu'' - f'' = P(Pu - f) - f'' = P^2u - Pf - f'' \end{aligned} \right] = \\
&= \left[-\varphi_k + f_k + \frac{h^2}{12}(Pf_k + f_k'') \right] + \left[\xi_k - P \left(1 + P \frac{h^2}{12} \right) \right] u(x_k) + O(h^4)
\end{aligned}$$

З останнього співвідношення, при

$$\xi_k = P \left(1 + \frac{h^2}{12} P \right), \quad \varphi_k = \left(1 + \frac{h^2}{12} P \right) f_k + \frac{h^2}{12} f_k''$$

маємо $\psi_h(x_k) = O(h^4)$. Такий прийом підвищення апроксимації на розв'язку диференціальної задачі можна застосовувати і для крайових умов II, III-го роду.

В розглянутій задачі (3.17), (3.18) ми мали крайові умови першого роду, тому вони апроксимувались точно. У випадку наявності $u'(x)$ в крайових умовах, використовують певну методику для отримання апроксимації того-ж порядку, що і для диференціального рівняння, тобто в (3.13) має бути $r = p$. Розглянемо найбільш уживані на прикладі крайової задачі для рівняння (3.17), де в лівій граничній точці задано умову третього роду

$$u'(a) + g_1 u(a) = -d_1, \quad (3.21)$$

а в правій граничній точці задано умову другого роду

$$u'(b) = d_2. \quad (3.22)$$

Метод підвищення порядку апроксимації на розв'язку. Застосуємо до умови (3.21). Спочатку замінимо $u'(a) \approx y_{x,1}$ – правою різницевою похідною і отримаємо апроксимацію цієї умови з першим порядком

$$\frac{y_2 - y_1}{h} + g_1 y_1 = -d_1.$$

Для отримання порядку $O(h^2)$ виконаємо розклад y_2 в ряд Тейлора

$$\frac{(y_1 + hu'(a) + 0.5h^2u''(a) + O(h^3)) - y_1}{h} + g_1 y_1 = -d_1,$$

а для обчислення $u''(a)$ скористаємось неперервним продовженням диференціального рівняння (3.17) на межу $x = a$. В результаті отримаємо

$$u'(a) + 0.5h[p(a)u(a) - f(a)] + O(h^2) + g_1 u(a) = -d_1.$$

Порівнюючи цей вираз з (3.21), бачимо, що для їх рівності з точністю до величин $O(h^2)$, нам необхідно "прибрати" вираз $0.5h[p(a)u(a) - f(a)]$ порядку $O(h)$. Отже, різницева умова, яка апроксимує крайову умову (3.21) з другим порядком, буде наступна:

$$\frac{y_2 - y_1}{h} + g_1 y_1 = -d_1 + 0.5h(p_1 y_1 - f_1),$$

яку перетворимо до форми

$$-\eta_1 y_1 + y_2 = -\mu_1, \quad (3.23)$$

де $\eta_1 = 1 - hg_1 + 0.5h^2 p_1$, $\mu_1 = hd_1 + 0.5h^2 f_1$.

Метод фіктивної (законтурної) точки. Застосуємо до умови (3.22). За допомогою центральної різниці в точці $b = x_N$ замінимо $u'(b) = \frac{y_{N+1} - y_{N-1}}{2h} + O(h^2)$, а для виключення y_{N+1} скористаємось рівнянням $y_{\bar{x}x,N} - p_N y_N = -f_N$. Отже, різницева умова, яка апроксимує крайову умову (3.22) з другим порядком, буде наступна:

$$\frac{y_{N-1} - 2y_N + (y_{N-1} + 2hd_2)}{h^2} - p_N y_N = -f_N,$$

яку перетворимо до форми

$$y_{N-1} - \eta_2 y_N = -\mu_2, \quad (3.24)$$

де $\eta_2 = 1 + 0.5h^2 p_N$, $\mu_2 = hd_2 + 0.5h^2 f_N$. Отже розв'язок РС (3.19), (3.23), (3.24) з другим порядком точності сходиться до розв'язку крайової задачі (3.17), (3.21), (3.22).

п.3. Різницеві схеми для нелінійних задач

Розглянемо застосування МРС до розв'язання наступної квазілінійної крайової задачі

$$\begin{cases} u''(x) = -f(x, u), & x \in (a, b); \\ u(a) = d_1, & u(b) = d_2. \end{cases} \quad (3.25)$$

Якщо припустити, що $f(x, u)$ обмежена і неперервна разом зі своїми другими похідними, то існує обмежена і неперервна $u^{(4)}(x)$. Позначимо $M_1 = \max|f'_u|$, $M_4 = \max|u^{(4)}|$. Аналогічно побудові РС для лінійної задачі, на рівномірній сітці (3.4) з триточковим шаблоном, отримаємо наступну нелінійну різницеву задачу

$$\begin{cases} y_{k-1} - 2y_k + y_{k+1} = -h^2 f(x_k, y_k), & k = \overline{2, N-1}; \\ y_1 = d_1, & y_N = d_2. \end{cases} \quad (3.26)$$

Для цієї задачі можна отримати: порядок апроксимації $O(h^2)$, коректність і збіжність з другим порядком ($\|z_k\|_c \leq \frac{h^2 M_4}{12m_1}$), при умові, що виконуються ще обмеження $|f'_u| \geq m_1 > 0$.

Виникає питання, як знайти розв'язок отриманої нелінійної системи?. Враховуючи, що метод послідовних наближень (МПН) можна записати у формі:

А. $2y_k^{(s+1)} = y_{k-1}^{(s)} + y_{k+1}^{(s)} + h^2 f(x_k, y_k^{(s)})$, але при цьому збіжність МПН відсутня!

В. $y_{k-1}^{(s)} - 2y_k^{(s)} + y_{k+1}^{(s)} = -h^2 f(x_k, y_k^{(s-1)})$, умова збіжності МПН з першим порядком $(b-a)^2 M_1 < 8$, а при $(b-a)^2 M_1 > \pi^2$ він може розбігатись.

Використаємо для розв'язання нелінійної системи (3.26) метод *лінеаризації* (метод *Ньютона*), умова збіжності якого є $\left\| \frac{f''_{uu}}{2f'_u} \right\|_c < 1$ і збіжність квадратична.

Алгоритм

1. Покладемо $y_k^{(0)} = d_1 + \frac{d_2 - d_1}{b-a}(x_k - a)$, $k = \overline{1, N}$;
2. Представимо $y_k^{(s+1)} = y_k^{(s)} + \Delta_k$, $k = \overline{1, N}$, тут s – номер ітерації;
3. Підставимо $y_k^{(s+1)}$ в (3.26), виконаємо лінеаризацію $f(x_k, y_k^{(s)} + \Delta_k) = f(x_k, y_k^{(s)}) + \Delta_k f'_u(x_k, y_k^{(s)})$ і для знаходження $\bar{\Delta}$ отримаємо СЛАР

$$\begin{cases} \Delta_{\bar{x}x,k} + f'_u(x_k, y_k^{(s)}) \Delta_k = -(f(x_k, y_k^{(s)}) + y_{\bar{x}x,k}^{(s)}), & k = \overline{2, N-1}; \\ \Delta_1 = 0, \quad \Delta_N = 0. \end{cases} \quad (3.27)$$

яка ефективно розв'язується *методом немонотонної прогонки*.

4. Пункти 2, 3 повторюємо до виконання умови $\|\Delta_k\|_c \leq \varepsilon \vee s > \max s$ – закінчення ІІІ. Тут $0 < \varepsilon < 1$ – задана точність, а $\max s > 0$ – максимальна кількість кроків ітерації (страховка від зациклювання).

Для розв'язку нелінійних задач досить ефективна *комплексна побудова розрахунку*, яка дозволяє при невеликому об'ємі розрахунків досягти високої точності.

Схематично опишемо цей алгоритм.

- Оберемо першу сітку з невеликим числом вузлів $N_1 \approx 2 \div 8$.
- Знайдемо розв'язок (3.26) описаним методом Ньютона (МН).
- Сітку згущуємо (число вузлів подвоюємо $N_{i+1} = 2N_i$), початкове наближення в МН (п.1) обчислюємо лінійною інтерполяцією з попередньої сітки.
- Знайдемо розв'язок (3.26) методом Ньютона.
- Повторюємо пункти с, d до виконання умови $\|\bar{y}(N_{i+1}) - \bar{y}(N_i)\|_c \leq \varepsilon_1$.
- Отриманий розв'язок $\bar{y}(N_{i+1})$ уточнюємо по другій формулі Рунге-Ромберга.

Якщо виконуються серії розрахунків з варіюванням параметрів вихідної задачі, то має сенс брати результати попереднього варіанта в якості початкового наближення на першій сітці вказаного алгоритму.

Розглянемо стратегію розв'язання, коли нелінійність присутня і в крайових умовах. Нехай це буде умова в точці $x = a$.

Випадки крайових умов виду (умовно назвемо «першого», «другого» роду)

$$\begin{aligned} u(a) &= \chi_1(u(a)); \quad u'(a) = \chi_2(u'(a)); \\ \chi_1(u(a)) &= 0; \quad \chi_2(u'(a)) = 0; \end{aligned}$$

не становлять суттєвої проблеми. Ці нелінійні рівняння попередньо «легко» розв'язуємо деяким ітераційним методом і приходимо до звичайних умов першого, другого роду

$$u(a) = d_1; \quad u'(a) = d_1.$$

Для випадку крайової умови виду («третього» роду)

$$u'(a) = \chi_1(u(a))$$

методом підвищення порядку апроксимації на розв'язку:

$$\frac{y_2 - y_1}{h} = \chi_1(y_1); \quad y_2 = y_1 + hu'(a) + 0.5h^2 u''(a) + O(h^3) = y_1 + h\chi_1(y_1) - 0.5h^2 f(a, y_1) + O(h^3)$$

можна отримати нелінійне різницеве рівняння другого порядку

$$y_1 - 0.5h^2 f(a, y_1) + h\chi_1(y_1) - y_2 = 0,$$

яке додамо до загальної нелінійної системи різницевого рівнянь.

Аналогічно приходиться поступати і в випадку умов виду

$$\chi_1(u(a), u'(a)) = 0.$$

ЛЕКЦІЯ № 05

ТЕМА № 3. Різницеві методи розв'язування крайових задач для ЗДР.

Розглянутий нами метод різницевої апроксимації дозволяє легко будувати РС відповідного порядку апроксимації на прямокутній сітці для рівнянь з неперервними і достатньо гладкими коефіцієнтами. Але цей метод складно і даже неможливо застосувати у випадку рівнянь з розривними коефіцієнтами, на не прямокутних сітках і для рівнянь високого порядку на нерівномірних сітках. Одним з надійних методів, який може бути використаний у цих випадках, є інтегро-інтерполяційний метод (ІІМ) побудови РС.

Схема застосування ІІМ:

- ♦ введення сітки і вибір шаблону;
- ♦ область інтегрування розбивається на комірки, які певним чином пов'язані з шаблоном РС;
- ♦ диференціальне рівняння інтегрують по комірці, по формулам векторного аналізу приводять до інтегральної форми, яка відповідає певному фізичному закону збереження (балансу);
- ♦ отримані інтеграли обчислюють по квадратурним формулам і будують різницеву задачу;
- ♦ виконують дослідження: порядок апроксимації, коректність і збіжність РС.

п.4. Інтегро-інтерполяційний метод побудови різницевих схем для лінійних задач з неперервними або кусково-неперервними коефіцієнтами

Розглянемо ІІМ побудови РС відповідної якості на прикладі лінійної крайової задачі [33], яка описує стаціонарний розподіл тепла в стержні довжини $b-a$ на правому кінці якого підтримується постійна температура, а на лівому – теплообмін з зовнішнім середовищем по закону Ньютона

$$\frac{d}{dx}(K(x)u'(x)) - p(x)u(x) = -f(x), \quad (3.28)$$

$$K(x) \geq K_0 > 0, \quad p(x) \geq 0, \quad x \in D \equiv (a, b);$$

$$\bar{D} = D \cup \partial D, \quad \partial D = a \cup b;$$

$$-K(a)u'(a) + g_1 u(a) = d_1, \quad g_1 \geq 0, \quad (3.29)$$

$$u(b) = d_2. \quad (3.30)$$

Будемо враховувати, що задані функції $K(x)$ – коефіцієнт теплопровідності і $p(x)$, $f(x)$ – щільність теплових джерел, можуть бути як достатньо гладкі, так і мати розриви (інтегровані особливості).

Введемо рівномірну сітку (3.4):

$$\bar{D}_h = \{x_k = a + (k-1)h, \quad k = 1, 2, \dots, N, \quad b-a = (N-1)h\},$$

оберемо шаблон $S\{x_{k-1}, x_k, x_{k+1}\}$ і означимо комірку (відрізок) з граничними точками $x_{k \pm 0.5} = x_k \pm 0.5h$.

Проінтегруємо рівняння (3.28) на відрізку $[x_k - 0.5h, x_k + 0.5h]$ і отримаємо баланс тепла

$$W_{k+0.5} - W_{k-0.5} - \int_{x_k-0.5h}^{x_k+0.5h} p(x)u(x)dx = - \int_{x_k-0.5h}^{x_k+0.5h} f(x)dx.$$

Тут позначено: $-W(x) = -K(x)u'(x)$ – тепловий потік, $W_{k \pm 0.5} = W(x_k \pm 0.5h)$. Якщо вважати $u(x) \in \text{Const}$, $x \in [x_{k-0.5}, x_{k+0.5}] \Rightarrow u(x) = u(x_k)$ на цьому відрізку і апроксимувати інтеграли наступним чином

$$\int_{x_k-0.5h}^{x_k+0.5h} p(x)u(x)dx = u(x_k) \int_{x_k-0.5h}^{x_k+0.5h} p(x)dx = hP_k y_k, \quad P_k = h^{-1} \int_{x_k-0.5h}^{x_k+0.5h} p(x)dx;$$

$$\int_{x_k-0.5h}^{x_k+0.5h} f(x)dx = hF_k, \quad F_k = h^{-1} \int_{x_k-0.5h}^{x_k+0.5h} f(x)dx,$$

то отримаємо таке різницеве рівняння

$$\frac{W_{k+0.5} - W_{k-0.5}}{h} - P_k y_k = -F_k. \quad (3.31)$$

Виразимо $W_{k\pm 0.5}$ через значення функції $u(x)$ в точках сітки D_h . Для цього проінтегруємо вираз $u'(x) = W(x)/K(x)$ на відрізку $[x_{k-1}, x_k]$

$$\int_{x_{k-1}}^{x_k} u'(x)dx \equiv u(x_k) - u(x_{k-1}) = \int_{x_{k-1}}^{x_k} W(x)/K(x)dx \approx W_{k-0.5} \int_{x_{k-1}}^{x_k} K^{-1}(x)dx$$

і отримаємо

$$W_{k-0.5} = A_k \frac{y_k - y_{k-1}}{h} \equiv A_k y_{\bar{x},k}; \quad A_k = \left(\frac{1}{h} \int_{x_{k-1}}^{x_k} \frac{dx}{K(x)} \right)^{-1}.$$

Аналогічно отримаємо вираз для $W_{k+0.5} = A_{k+1} y_{\bar{x},k+1} \equiv A_{k+1} y_{x,k}$. Таким чином, різницеве рівняння (3.31) прийме вигляд

$$\frac{1}{h} (A_{k+1} y_{\bar{x},k+1} - A_k y_{\bar{x},k}) - P_k y_k = -F_k.$$

З урахуванням усіх внутрішніх точок сітки, маємо систему

$$\begin{cases} (A_k y_{\bar{x}})_{x,k} - P_k y_k = -F_k, \\ k = 2, N-1. \end{cases} \quad (3.32)$$

Для отримання апроксимації крайової умови в сітковій точці $x_1 = a$ інтегруємо рівняння (3.28) на відрізку $[x_1, x_{1.5}] = [a, a + 0.5h]$:

$$W_{1+0.5} - W(a) - \int_a^{a+0.5h} p(x)u(x)dx = - \int_a^{a+0.5h} f(x)dx.$$

Після підстановки в цю рівність значення сіткової функції $W_{1+0.5}$, значення $-W(a)$ з умови (3.29) і введення відповідних позначень для апроксимації інтегралів

$$P_1 = \frac{1}{0.5h} \int_a^{a+0.5h} p(x)dx, \quad F_1 = \frac{1}{0.5h} \int_a^{a+0.5h} f(x)dx,$$

отримуємо сіткове рівняння

$$A_2 y_{\bar{x},2} + d_1 - g_1 y_1 - 0.5h P_1 y_1 = -0.5h F_1,$$

або при позначеннях $\tilde{g}_1 = g_1 + 0.5h P_1$, $\tilde{d}_1 = d_1 + 0.5h F_1$:

$$-A_2 y_{x,1} + \tilde{g}_1 y_1 = \tilde{d}_1,$$

яке також можна перетворити до форми

$$-\eta_1 y_1 + \theta_1 y_2 = -\mu_1, \quad (3.33)$$

де $\eta_1 = A_2 + h(g_1 + 0.5h P_1)$, $\theta_1 = A_2$, $\mu_1 = h(d_1 + 0.5h F_1)$.

Крайова умова в сітковій точці $x_N = b$ апроксимується точно

$$y_N = d_2. \quad (3.34)$$

Отже сукупність рівнянь (3.32)-(3.34) складе різницеву задачу, яка ставиться у відповідність крайовій задачі (3.28)-(3.30).

Знайдемо умови, при виконанні яких, для нашої РЗ отримаємо апроксимацію $O(h^2)$. Розглянемо спочатку апроксимацію диференціального виразу, який входить в рівняння (3.28)

$$\psi(x_k) = (A_k y_{\bar{x}})_{x,k} - \frac{d}{dx} (K(x)u'(x)) \Big|_{x=x_k} = (A_k y_{\bar{x}})_{x,k} - (K'(x)u'(x) + K(x)u''(x)) \Big|_{x=x_k}$$

Враховуючи розклад $u(x_k \pm h)$ в ряд Тейлора для сіткових операторів

$$y_{\bar{x},k+1} = u'(x_k) + \frac{h}{2}u''(x_k) + \frac{h^2}{6}u'''(x_k) + O(h^3),$$

$$y_{\bar{x},k} = u'(x_k) - \frac{h}{2}u''(x_k) + \frac{h^2}{6}u'''(x_k) + O(h^3),$$

які входять в $(A_k y_{\bar{x}})_{x,k}$, отримуємо

$$\psi(x_k) = \left[\frac{A_{k+1} - A_k}{h} - K'(x_k) \right] u'(x_k) + \left[\frac{A_{k+1} + A_k}{2} - K(x_k) \right] u''(x_k) + \frac{h}{6}(A_{k+1} - A_k)u'''(x_k) + O(h^2).$$

Якщо прирівняємо нулю вирази в квадратних дужках, то маємо для похибки $\psi(x_k)$ достатні умови апроксимації другого порядку:

$$\frac{A_{k+1} - A_k}{h} = K'(x_k) + O(h^2), \quad \frac{A_{k+1} + A_k}{2} = K(x_k) + O(h^2). \quad (3.35)$$

Обчислюючи коефіцієнти

$$P_k = p(x_k) + O(h^2), \quad F_k = f(x_k) + O(h^2), \quad k = \overline{2, N-1},$$

з аналогічною точністю, ми отримаємо апроксимацію рівняння (3.28) різницеvim рівнянням (3.32) з другим порядком по h , тобто $\psi_h(x_k) = O(h^2)$.

Тепер розглянемо апроксимацію крайової умови в точці $x = a$ на розв'язку рівняння (3.28). Виконуємо ланцюжок перетворень для похибки $v_h(a) = v_h(x_1)$:

$$\begin{aligned} v_h(x_1) &= (A_2 y_{\bar{x},2} + d_1 - g_1 y_1 - 0.5hP_1 y_1 + 0.5hF_1) - (K(a)u'(a) - g_1 u(a) + d_1) = \\ &= A_2 y_{\bar{x},2} - 0.5hP_1 y_1 + 0.5hF_1 - K(a)u'(a) = \\ &= \left[\begin{aligned} A_2 y_{\bar{x},2} &\equiv W_{1+0.5} = W_1 + 0.5hW'_1 + O(h^2) = \\ &= K(a)u'(a) + 0.5h(K(a)u'(a))' + O(h^2); \\ P_1 &= p(a) + O(h); F_1 = f(a) + O(h); \end{aligned} \right] = \\ &= K(a)u'(a) + 0.5h \left[\underbrace{(K(a)u'(a))' - p(a)u(a) + f(a)}_{=0} + O(h) \right] + O(h^2) - K(a)u'(a) = \\ &= 0.5h[O(h)] + O(h^2) = O(h^2). \end{aligned}$$

Отже, при достатній гладкості коефіцієнтів $K(x)$, $p(x)$, $f(x)$ і розв'язку $u(x)$ РС (3.32)-(3.34) апроксимує задачу (3.28)-(3.30) з другим порядком по h .

При практичному використанні РС (3.32)-(3.34) слідуємо такій стратегії для знаходження коефіцієнтів A_k, P_k, F_k :

у випадку неперервних коефіцієнтів $K(x)$, $p(x)$, $f(x)$ можна

- 1) Безпосередньо (точно) обчислювати інтеграли;
- 2) Використовувати наближене обчислення інтегралів з порядком не нижче $O(h^2)$, наприклад, квадратурні формули *центральных прямоугольников*:

$$A_k = K(x_k - 0.5h), P_k = p(x_k), F_k = f(x_k);$$

або *трапецій*

$$A_k = \frac{2K(x_{k-1})K(x_k)}{K(x_{k-1}) + K(x_k)}, P_k = \frac{p(x_{k-0.5}) + p(x_{k+0.5})}{2}, F_k = \frac{f(x_{k-0.5}) + f(x_{k+0.5})}{2};$$

у випадку розривних коефіцієнтів $K(x)$, $p(x)$, $f(x)$ потрібно

- 1) Будувати сітку $\bar{D}_h$ таким чином, щоб розрив(и) коефіцієнта $K(x)$ був в півцілих точках сітки, а розрив(и) $p(x)$, $f(x)$ – в цілих точках сітки;
- 2) Застосовувати наступні формули обчислення [33]

$$A_k = 0.5(K(x_{k-0.5} - 0) + K(x_{k-0.5} + 0)), \\ P_k = 0.5(p(x_k - 0) + p(x_k + 0)), \quad F_k = 0.5(f(x_k - 0) + f(x_k + 0)).$$

Для отримання апіорних оцінок типу (3.14) і доведення збіжності у випадку $K(x) \in C^1([a, b])$, $p(x), f(x) \in C([a, b])$, $u(x) \in C^4([a, b])$ застосуємо метод енергетичних нерівностей [33]. В цьому методі використовують деякі різницеві тотожності і нерівності, тому спочатку розглянемо їх.

Означимо наступні скалярні добутки і норми для сіткових функцій, заданих на сітці $\bar{D}_h$ (3.13):

$$(w, v) = \sum_{i=2}^{N-1} w_i v_i h, \quad \|v\| = (v, v); \quad (w, v] = \sum_{i=2}^N w_i v_i h, \quad \|v] = (v, v].$$

Формула сумування по частинам. Має місце наступна різницева тотожність:

$$(w, v_x) = w_N v_N - w_1 v_2 - (v, w_{\bar{x}}] \quad (3.36)$$

Для доведення виконаємо ланцюжок перетворень

$$(w, v_x) = \sum_{i=2}^{N-1} w_i (v_{i+1} - v_i) = \sum_{i=3}^N w_{i-1} v_i - \sum_{i=2}^{N-1} w_i v_i = \sum_{i=2}^N w_{i-1} v_i - w_1 v_2 - \sum_{i=2}^N w_i v_i + w_N v_N = \\ = w_N v_N - w_1 v_2 - \sum_{i=2}^N v_i (w_i - w_{i-1}).$$

Перша різницева формула Гріна. Виконаємо підстановку в (3.36) $z \mapsto w, az_{\bar{x}} \mapsto v$, отримаємо наступну тотожність:

$$(z, (az_{\bar{x}})_x) = a_N z_{\bar{x}, N} z_N - a_2 z_{x, 1} z_1 - (a, (z_{\bar{x}})^2]. \quad (3.37)$$

Якщо сіткова функція $z_N = 0$, то (3.37) буде

$$(z, (az_{\bar{x}})_x) = - (a, (z_{\bar{x}})^2] - a_2 z_{x, 1} z_1. \quad (3.38)$$

Для довільної сіткової функції, такої, що $z_N = 0$, виконується нерівність

$$\|z\|_{Ch}^2 \leq (b - a) \|z_{\bar{x}}\|^2. \quad (3.39)$$

Для доведення використаємо тотожність

$$z_i = - \sum_{j=i+1}^N h z_{\bar{x}, j} = - \sum_{j=i+1}^N \sqrt{h} (\sqrt{h} z_{\bar{x}, j}), \quad i = \overline{1, N-1},$$

до якого застосуємо нерівність Коші-Буняковського

$$\left| \sum_{j=i+1}^N a_j b_j \right|^2 \leq \sum_{j=i+1}^N a_j^2 \sum_{j=i+1}^N b_j^2.$$

Тоді отримаємо

$$|z_i|^2 \leq \sum_{j=i+1}^N h \sum_{j=i+1}^N h (z_{\bar{x}, j})^2 = (b - a - x_i) \sum_{j=i+1}^N h (z_{\bar{x}, j})^2 \leq (b - a) \sum_{j=1}^N h (z_{\bar{x}, j})^2, \quad i = \overline{1, N-1},$$

звідки і маємо (3.39).

Тепер розглянемо збіжність побудованої РС до розв'язку диференціальної задачі. Стандартним чином отримаємо задачу для похибки $z_k = y_k - (u)_h$

$$\begin{cases} (A_k z_{\bar{x}})_{x,k} - P_k z_k = -\psi_h(x_k), & k = \overline{2, N-1}; \\ -A_2 z_{x,1} + \tilde{g}_1 z_1 = \nu_h(x_1), & z_N = 0, \end{cases} \quad (3.40)$$

де стандартно позначено $\psi_h(x_k) = -(A_k u_{\bar{x}})_{x,k} + P_k u_k + F_k$ – похибка апроксимації диференціального рівняння (3.28) різницеvim (3.32) на розв’язку диференціального і $\nu_h(x_1) = A_2 u_{x,1} - \tilde{g}_1 u_1 + \tilde{d}_1$ – похибка апроксимації крайової умови в точці $x = a$ на розв’язку рівняння (3.28). Помножимо перше рівняння (3.40) на hz_k і просумуємо по k від 2 до $N-1$:

$$((A_k z_{\bar{x}})_{x,k}, z) - (P_k, (z_k)^2) = -(\psi_h, z).$$

До отриманої рівності спочатку застосуємо різницеву формулу Гріна (3.38)

$$(A, (z_{\bar{x}})^2) + A_2 z_{x,1} z_1 + (P, z^2) = (\psi_h, z),$$

потім підставимо $A_2 z_{x,1} = \tilde{g}_1 z_1 - \nu_h(x_1)$ з другого рівняння (3.40) і отримаємо таку тотожність

$$(A, (z_{\bar{x}})^2) + \tilde{g}_1 (z_1)^2 + (P, z^2) = (\psi_h, z) + \nu_h(x_1) z_1. \quad (3.41)$$

Із умов на вхідні дані $K(x) \geq K_0 > 0$, $p(x) \geq 0$, $g_1 \geq 0$, слідує, що

$$A_k \geq K_0 > 0, \quad \tilde{g}_1 \geq 0, \quad P_k \geq 0. \quad (3.42)$$

Скориставшись (3.42), оцінимо доданки, які входять в ліву частину тотожності (3.41), наступним чином:

$$\begin{aligned} (A, (z_{\bar{x}})^2) &= \sum_{i=2}^N A_i (z_{\bar{x},i})^2 h \geq K_0 \sum_{i=2}^N (z_{\bar{x},i})^2 h = K_0 \|z_{\bar{x}}\|^2, \\ \tilde{g}_1 (z_1)^2 &\geq 0, \quad (P, z^2) \geq 0. \end{aligned}$$

Тоді, з (3.41) маємо

$$K_0 \|z_{\bar{x}}\|^2 \leq |(\psi_h, z)| + |\nu_h(x_1) z_1|. \quad (3.43)$$

Праву частину останньої нерівності можна оцінити наступним чином

$$|(\psi_h, z)| + |\nu_h(x_1) z_1| \leq \sum_{k=2}^{N-1} |\psi_h(x_k)| |z_k| h + |\nu_h(x_1)| |z_1| \leq \|z\|_{Ch} \left(\sum_{k=2}^{N-1} |\psi_h(x_k)| h + |\nu_h(x_1)| \right).$$

Підставимо цю оцінку в (3.43) і врахуємо нерівність (3.39), отримаємо

$$\frac{K_0}{b-a} (\|z\|_{Ch})^2 \leq \|z\|_{Ch} \left(\sum_{k=2}^{N-1} |\psi_h(x_k)| h + |\nu_h(x_1)| \right) \leq \|z\|_{Ch} \left((b-a) \sum_{k=2}^{N-1} |\psi_h(x_k)| h + |\nu_h(x_1)| \right).$$

Звідки маємо оцінку

$$\|z\|_{Ch} \leq \frac{b-a}{K_0} ((b-a) \|\psi_h\|_{Ch} + |\nu_h(x_1)|). \quad (3.44)$$

Враховуючи, що при дослідженні апроксимації побудованої РС ми отримали $|\psi_h(x_k)| = O(h^2)$, $k = \overline{2, N-1}$ і $|\nu_h(x_1)| = O(h^2)$ при $h \rightarrow 0$ на розв’язку диференціальної задачі (3.28), то з (3.44) слідує $\|z\|_{Ch} = O(h^2)$ при $h \rightarrow 0$. Отже ми довели, що розв’язок різницевої задачі (3.32)-(3.34) збігається при $h \rightarrow 0$ до розв’язку вихідної диференціальної задачі (3.28)-(3.30) з другим порядком по h на розв’язку диференціальної задачі (3.28).

Випадки розривних коефіцієнтів $K(x)$, $p(x)$, $f(x)$ і нерівномірних сіток (3.3) розглядається в [33].

ЛЕКЦІЯ № 06

ТЕМА № 3. Різницеві методи розв'язування крайових задач для ЗДР.

Для чисельно-аналітичного розв'язання крайових задач (КЗ)

$$Lu(x) = f(x), \quad x \in D; \quad lu(x) = g(x), \quad x \in \partial D.$$

використовують проекційні і варіаційні методи, в яких наближений розв'язок будується у вигляді збіжної послідовності функцій $u_N(x) = \sum_{i=1}^N c_i \varphi_i(x)$ таким чином, що він точно задовольняє крайові умови і наближено саме рівняння. Знаходження $u_N(x)$ в проекційних методах (ПМ) базується на рівності нулю моментів $(Lu_N(x) - f(x), \psi_j(x))$, а в варіаційних (ВМ) – на мінімізації деякого функціоналу $\Phi(u)$, мінімум $\tilde{u}(x)$ якого і є розв'язком крайової задачі. Як перші, так і другі знаходять коефіцієнти $\{c_i\}_{i=1}^N$ з розв'язку СЛАР, матриця якої є повністю заповнена. Різні ПМ відрізняються вибором проекційної системи $\psi_j(x)$, а ВМ – вибором функціоналу Φ , який залежить від властивостей оператора L .

Ідеї ВМ Рітца, при певному виборі функцій $\varphi_i(x), \psi_j(x)$ і $L = L^* > 0$ використовують для побудови РС *варіаційно-різницеvim методом*.

п.5. Проекційно-різницеvim метод побудови різницеvim схем

Розглянемо проекційно-різницеvim метод побудови РС для варіанта *несамоспряженого оператора* $Lu(x)$ і однорідних крайових умов [31–33]

$$\frac{d}{dx} (K(x)u'(x)) + r(x)u'(x) - p(x)u(x) = -f(x), \quad (3.45)$$

$$K(x) > 0, \quad p(x) \geq 0, \quad x \in (0,1);$$
$$u(0) = 0, \quad u(1) = 0. \quad (3.46)$$

Введемо рівномірну сітку

$$\bar{D}_h = \{x_i = a + (i-1)h, \quad i = 1, 2, \dots, N, \quad b-a = (N-1)h\}.$$

Наближений розв'язок (3.45) будемо шукати у вигляді

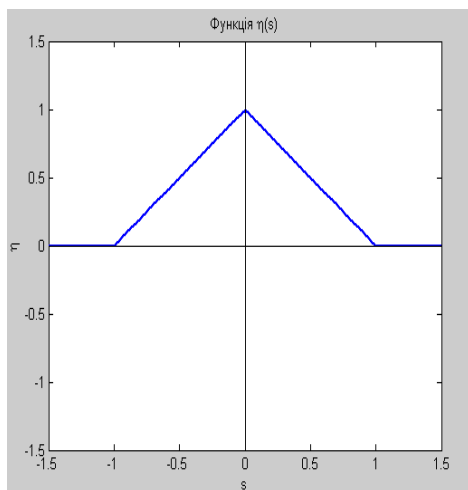
$$u_N(x) = \sum_{i=2}^{N-1} y_i \varphi_i(x) \quad (3.47)$$

де

$y_i = u(x_i), \quad i = \overline{1, N}$ – невідомі коефіцієнти, причому $y_1 = y_N = 0$, враховуючи крайові умови (3.46);

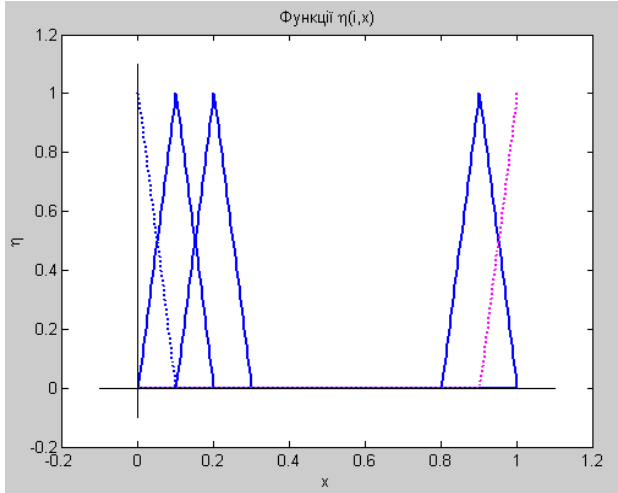
$\varphi_i(x), \quad i = \overline{2, N-1}$ – базис скінченномірного простору V_{N-2} .

Введемо функцію:



$$\eta(s) = \begin{cases} 0, & (s \leq -1) \vee (s \geq 1), \\ 1+s, & -1 < s \leq 0, \\ 1-s, & 0 < s < 1. \end{cases} \quad (3.48)$$

Виберемо базисні функції $\varphi_i(x)$ у вигляді $\varphi_i(x) = \eta\left(\frac{x-x_i}{h}\right) \equiv \eta_i(x)$, $i = \overline{2, N-1}$:



$$\eta_i(x) = \begin{cases} 0, & (x \leq x_{i-1}) \vee (x \geq x_{i+1}), \\ \frac{x-x_{i-1}}{h}, & x_{i-1} < x \leq x_i, \\ \frac{x_{i+1}-x}{h}, & x_i < x < x_{i+1}. \end{cases} \quad (3.49)$$

На рисунку позначені пунктирними лініями і функції $\eta_1(x)$, $\eta_N(x)$

$$\eta_1(x) = \begin{cases} 0, & (x \leq 0) \vee x \geq x_2, \\ \frac{x_2-x}{h}, & x_1 < x < x_2 \end{cases}, \quad \eta_N(x) = \begin{cases} 0, & (x \leq x_{N-1}) \vee (x \geq 1), \\ \frac{x-x_{N-1}}{h}, & x_{N-1} < x < x_N \end{cases},$$

які використовуються при крайових умовах другого і третього роду.

В подальшому нам буде потрібно і похідні від функцій $\eta_i(x)$, $i = \overline{2, N-1}$, а отже маємо

$$\eta'_i(x) = \begin{cases} 0, & (x \leq x_{i-1}) \vee (x \geq x_{i+1}), \\ h^{-1}, & x_{i-1} < x \leq x_i, \\ -h^{-1}, & x_i < x < x_{i+1}. \end{cases} \quad (3.50)$$

Тепер, згідно з методом Гальоркіна ($\psi_j(x) = \varphi_j(x)$), запишемо умову $(Lu_N(x) - f(x), \eta_i(x)) = 0$ для визначення невідомих коефіцієнтів y_i , $i = \overline{2, N-1}$ з урахуванням (3.45), (3.47), (3.48):

$$\int_0^1 \frac{d}{dx} \left(K(x) \sum_{j=2}^{N-1} y_j \eta'_j(x) \right) \eta_i(x) dx + \int_0^1 r(x) \sum_{j=2}^{N-1} y_j \eta'_j(x) \eta_i(x) dx - \int_0^1 p(x) \sum_{j=2}^{N-1} y_j \eta_j(x) \eta_i(x) dx = - \int_0^1 f(x) \eta_i(x) dx, \\ i = 2, 3, \dots, N-1.$$

Перший інтеграл в цьому рівнянні перетворимо за формулою інтегрування по частинам:

$$\langle 1 \rangle = \left(K(x) \sum_{j=2}^{N-1} y_j \eta'_j(x) \eta_i(x) \right) \Big|_0^1 - \int_0^1 K(x) \sum_{j=2}^{N-1} y_j \eta'_j(x) \eta'_i(x) dx = 0 - \sum_{j=2}^{N-1} y_j \tilde{A}_{i,j}, \quad \tilde{A}_{i,j} = \int_0^1 K(x) \eta'_j(x) \eta'_i(x) dx,$$

другий і третій інтеграли представимо у формі:

$$\langle 2 \rangle = \sum_{j=2}^{N-1} y_j \int_0^1 r(x) \eta'_j(x) \eta_i(x) dx = \sum_{j=2}^{N-1} y_j \tilde{R}_{i,j}, \quad \tilde{R}_{i,j} = \int_0^1 r(x) \eta'_j(x) \eta_i(x) dx;$$

$$\langle 3 \rangle = \sum_{j=2}^{N-1} y_j \int_0^1 p(x) \eta_j(x) \eta_i(x) dx = \sum_{j=2}^{N-1} y_j \tilde{P}_{i,j}, \quad \tilde{P}_{i,j} = \int_0^1 p(x) \eta_j(x) \eta_i(x) dx,$$

а четвертий інтеграл позначимо $b_i = \int_0^1 f(x) \eta_i(x) dx$. В результаті маємо наступну СЛАР

для визначення y_i , $i = \overline{2, N-1}$:

$$\begin{cases} \sum_{j=2}^{N-1} A_{i,j} y_j = b_i, \\ i = 2, 3, \dots, N-1. \end{cases} \quad (3.51)$$

де елементи матриці $\mathbf{A}$ визначаються за формулою $A_{i,j} = \tilde{A}_{i,j} - \tilde{R}_{i,j} + \tilde{P}_{i,j}$.

Використаємо властивості функцій $\eta_i(x)$ для більш детального запису коефіцієнтів $\tilde{A}_{i,j}$, $\tilde{R}_{i,j}$, $\tilde{P}_{i,j}$, b_i .

Коефіцієнти b_i :

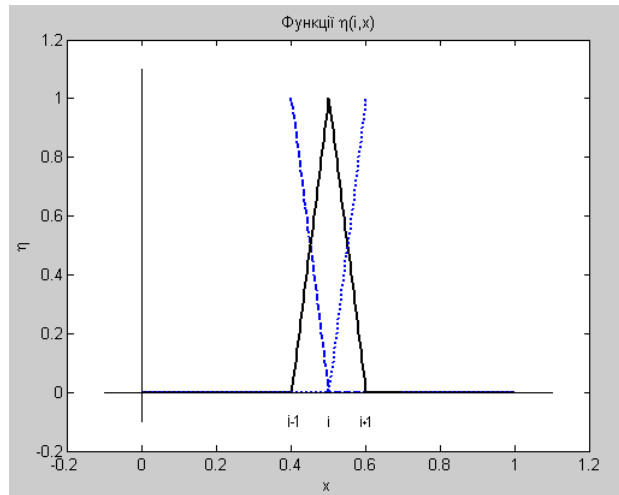
$$b_i = \int_0^1 f(x) \eta_i(x) dx = \frac{1}{h} \int_{x_{i-1}}^{x_i} f(x)(x - x_{i-1}) dx + \frac{1}{h} \int_{x_i}^{x_{i+1}} f(x)(x_{i+1} - x) dx = h \varphi_i,$$

$$\varphi_i = \frac{1}{h^2} \left[\int_{x_{i-1}}^{x_i} f(x)(x - x_{i-1}) dx + \int_{x_i}^{x_{i+1}} f(x)(x_{i+1} - x) dx \right].$$

Для коефіцієнтів $\tilde{P}_{i,j}$ спочатку використаємо функцію $\eta_i(x)$:

$$\tilde{P}_{i,j} = \int_0^1 p(x) \eta_i(x) \eta_j(x) dx = \frac{1}{h} \int_{x_{i-1}}^{x_i} p(x)(x - x_{i-1}) \eta_j(x) dx + \frac{1}{h} \int_{x_i}^{x_{i+1}} p(x)(x_{i+1} - x) \eta_j(x) dx,$$

а потім функцію $\eta_j(x)$ з урахуванням інтервалу обчислення $[x_{i-1}, x_{i+1}]$, що приводить до використання $\eta_j(x)$ з індексами $j = i-1, i, i+1$, як показано на рисунку



, а отже при y_{i-1}, y_i, y_{i+1} будуть різні коефіцієнти. Обчислимо їх:

$$\text{для } j = i-1: \frac{1}{h^2} \int_{x_{i-1}}^{x_i} p(x)(x - x_{i-1})(x_i - x) dx + 0,$$

$$\text{для } j = i: \frac{1}{h^2} \int_{x_{i-1}}^{x_i} p(x)(x - x_{i-1})^2 dx + \frac{1}{h^2} \int_{x_i}^{x_{i+1}} p(x)(x_{i+1} - x)^2 dx,$$

$$\text{для } j = i+1: 0 + \frac{1}{h^2} \int_{x_i}^{x_{i+1}} p(x)(x_{i+1} - x)(x - x_i) dx.$$

Для коефіцієнтів $\tilde{R}_{i,j}$ також спочатку використаємо функцію $\eta_i(x)$:

$$\tilde{R}_{i,j} = \int_0^1 r(x) \eta_i(x) \eta'_j(x) dx = \frac{1}{h} \int_{x_{i-1}}^{x_i} r(x)(x - x_{i-1}) \eta'_j(x) dx + \frac{1}{h} \int_{x_i}^{x_{i+1}} r(x)(x_{i+1} - x) \eta'_j(x) dx,$$

а потім функцію $\eta'_j(x)$ з урахуванням (3.50) і інтервалу обчислення $[x_{i-1}, x_{i+1}]$, що приводить до використання $\eta'_j(x)$ з індексами $j = i-1, i, i+1$, а отже коефіцієнти при y_{i-1}, y_i, y_{i+1} будуть різні:

$$\text{для } j = i-1: -\frac{1}{h^2} \int_{x_{i-1}}^{x_i} r(x)(x - x_{i-1})dx + 0,$$

$$\text{для } j = i: \frac{1}{h^2} \int_{x_{i-1}}^{x_i} r(x)(x - x_{i-1})dx - \frac{1}{h^2} \int_{x_i}^{x_{i+1}} r(x)(x_{i+1} - x)dx,$$

$$\text{для } j = i+1: 0 + \frac{1}{h^2} \int_{x_i}^{x_{i+1}} r(x)(x_{i+1} - x)dx.$$

Для коефіцієнтів $\tilde{A}_{i,j}$ спочатку використаємо функцію $\eta'_i(x)$:

$$\tilde{A}_{i,j} = \int_0^1 K(x) \eta'_i(x) \eta'_j(x) dx = \frac{1}{h} \int_{x_{i-1}}^{x_i} K(x) \eta'_j(x) dx - \frac{1}{h} \int_{x_i}^{x_{i+1}} K(x) \eta'_j(x) dx,$$

а потім функцію $\eta'_j(x)$ з урахуванням інтервалу обчислення $[x_{i-1}, x_{i+1}]$, що приводить до використання $\eta'_j(x)$ з індексами $j = i-1, i, i+1$, а отже коефіцієнти при y_{i-1}, y_i, y_{i+1} будуть різні:

$$\text{для } j = i-1: -\frac{1}{h^2} \int_{x_{i-1}}^{x_i} K(x) dx + 0,$$

$$\text{для } j = i: \frac{1}{h^2} \int_{x_{i-1}}^{x_i} K(x) dx + \frac{1}{h^2} \int_{x_i}^{x_{i+1}} K(x) dx,$$

$$\text{для } j = i+1: 0 + \frac{1}{h^2} \int_{x_i}^{x_{i+1}} K(x) dx.$$

Після підстановки обчислених значень $\tilde{A}_{i,j}, \tilde{R}_{i,j}, \tilde{P}_{i,j}, b_i$ в (3.51), групування коефіцієнтів при y_{i-1}, y_i, y_{i+1} і введення наступних позначень:

$$\begin{aligned} a_i &= \frac{1}{h} \left[\int_{x_{i-1}}^{x_i} K(x) dx - \int_{x_{i-1}}^{x_i} p(x)(x - x_{i-1})(x_i - x) dx \right], \\ d_i &= \frac{1}{h^2} \left[\int_{x_{i-1}}^{x_i} p(x)(x - x_{i-1})^2 dx + \int_{x_i}^{x_{i+1}} p(x)(x_{i+1} - x)^2 dx \right], \\ b_i^- &= \frac{1}{h^2} \int_{x_{i-1}}^{x_i} r(x)(x - x_{i-1}) dx, \quad b_i^+ = \frac{1}{h^2} \int_{x_i}^{x_{i+1}} r(x)(x_{i+1} - x) dx, \\ \varphi_i &= \frac{1}{h^2} \left[\int_{x_{i-1}}^{x_i} f(x)(x - x_{i-1}) dx + \int_{x_i}^{x_{i+1}} f(x)(x_{i+1} - x) dx \right] \end{aligned} \quad (3.52)$$

система рівнянь (3.51) прийме звичну для нас форму різницевої схеми

$$\begin{cases} y_1 = 0, \\ \frac{1}{h^2} [a_{i+1}(y_{i+1} - y_i) - a_i(y_i - y_{i-1})] + b_i^+ \frac{y_{i+1} - y_i}{h} + b_i^- \frac{y_i - y_{i-1}}{h} - d_i y_i = -\varphi_i, \quad i = \overline{2, N-1}, \\ y_N = 0, \end{cases} \quad (3.53)$$

або з використанням різницевих операторів

$$\begin{cases} y_1 = 0, \\ (a \cdot y_{\bar{x}})_{x,i} + b_i^+ y_{x,i} + b_i^- y_{\bar{x},i} - d_i y_i = -\varphi_i, \quad i = \overline{2, N-1}, \\ y_N = 0. \end{cases}$$

При $r(x) \equiv 0$ отримана схема буде співпадати з РС, яку можна побудувати варіаційно-різницеvim методом на базі методу Рітца.

У випадку постійних коефіцієнтів $K(x)$, $r(x)$, $p(x)$, обчислення коефіцієнтів РС (3.53) спрощується і ми маємо:

$$a_i = K - \frac{h^2}{6} p, \quad d_i = p, \quad b_i^- = b_i^+ = \frac{r}{2}, \quad (b_i^- y_{\bar{x},i} + b_i^+ y_{x,i} = r y_{\bar{x},i}).$$

Потрібно відзначити, що при такому представленні базисних і проєкційних функцій $\varphi_i(x)$, $\psi_i(x)$ який ми обрали – $\varphi_i(x) = \psi_i(x) = \eta \left(\frac{x - x_i}{h} \right)$, розглянутий нами проєкційно-різницеvim метод співпадає з *методом скінченних елементів* для одновимірного випадку.

Для отримання чисельного результату, систему (3.53) потрібно представити у такій формі запису, щоб можна було застосувати метод (монотонної|немонотон-ної) прогонки, а саме

$$\begin{cases} y_1 = 0, \\ A_i y_{i-1} - C_i y_i + B_i y_{i+1} = -F_i, \quad i = \overline{2, N-1}, \\ y_N = 0, \end{cases}$$

де уведені наступні позначення

$$A_i = a_i - \frac{\tilde{b}_i^-}{h}, \quad B_i = a_{i+1} + \frac{\tilde{b}_i^+}{h}, \quad C_i = A_i + B_i + \tilde{d}_i, \quad F_i = \tilde{\varphi}_i; \\ \tilde{b}_i^- = h^2 b_i^-, \quad \tilde{b}_i^+ = h^2 b_i^+, \quad \tilde{d}_i = h^2 d_i, \quad \tilde{\varphi}_i = h^2 \varphi_i.$$

Відносно обчислення інтегралів в коефіцієнтах отриманої РС, тобто в формулах (3.52), то можна притримуватись тієї ж стратегії, що і при використанні інтегро-інтерполяційного методу.

Розглянемо приклад розв'язання задачі (3.45)–(3.46) для модельного варіанту постановки. Створимо відповідне програмне забезпечення :

```
% М-файл      МКЕ_MAIN.M
%
% Розв'язання модельної задачі
% (k(x) * U'(x))' + r(x) * U'(x) - p(x) * U(x) = -f(x),
%          x ∈ (1,2);
%          U(1) = 0,   U(2) = 0.
% Для k(x) = x, r(x) = 0, p(x) = 1/x-x, f(x) = x^2
% розв'язок буде
% U(x) = 3.6072*besselj(1,x) + 0.75195*bessely(1,x).
%

clear all, clc

% Вхідні дані :
a = 1; b = 2;

% Наближений розв'язок :
X = linspace(a, b, 21);
Y = m_mke(@mke_k, @mke_r, @mke_p, @mke_f, X);
Y = Y + X;
```

```

% Точний розв'язок :
U = mke_u(X);

plot(X, Y, X, U)
grid on
title('Метод скінченних елементів');
xlabel('x'); ylabel('u(x)');
legend('y', 'u(x)', 'Location', 'Best');
pause
disp(strcat('Похибка в нормі C=', num2str(norm(Y - U))));
plot(X, abs(Y - U))
grid on

function f = mke_u(x)
% Задання функції u(x)
f = 3.6072 .* besselj(1, x) + 0.75195 .* bessely(1, x);
return

function f = mke_k(x)
% Задання функції k(x)
f = x;
return

function f = mke_r(x)
% Задання функції r(x)
f = 0;
return

function f = mke_p(x)
% Задання функції p(x)
f = 1 ./ x - x;
return

function f = mke_f(x)
% Задання функції f(x)
f = x .* x;
return

function Y = m_mke(kfun, rfun, pfun, ffun, X)
% Наближений розв'язок несамопряженої крайової задачі
%  $(k(x) * u'(x))' + r(x) * u'(x) - p(x) * u(x) = -f(x)$ ,
%  $k(x) \geq k_0 > 0$ ,  $p(x) \geq 0$ ,  $x \in (a, b)$ ,
%  $u(a) = 0$ ,  $u(b) = 0$ .
% Різницева схема побудована
% проекційно-різничевим методом
% (методом скінченних елементів).
%
% Вхідні дані :
% kfun - функція користувача, з описом k(x);
% rfun - функція користувача, з описом r(x);
% pfun - функція користувача, з описом p(x);
% ffun - функція користувача, з описом f(x);
% X - масив зі значеннями різницевої сітки.
%
% Вихідні дані:
% Y - масив зі значеннями наближеного розв'язку.

if nargin < 5
error('M_MKE:NotEnoughInputs',...
'Not enough input arguments. See M_MKE.');
```

```

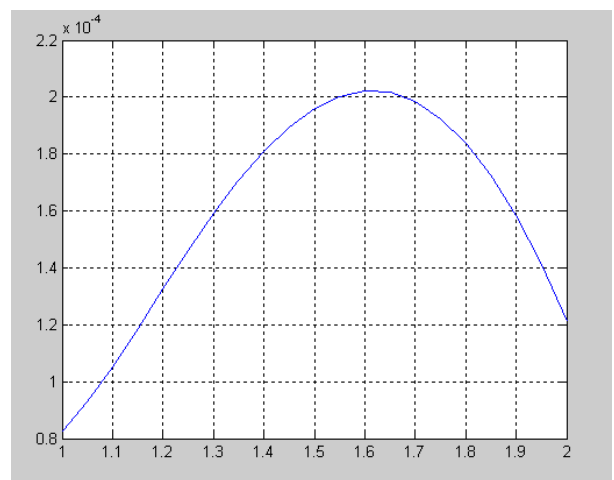
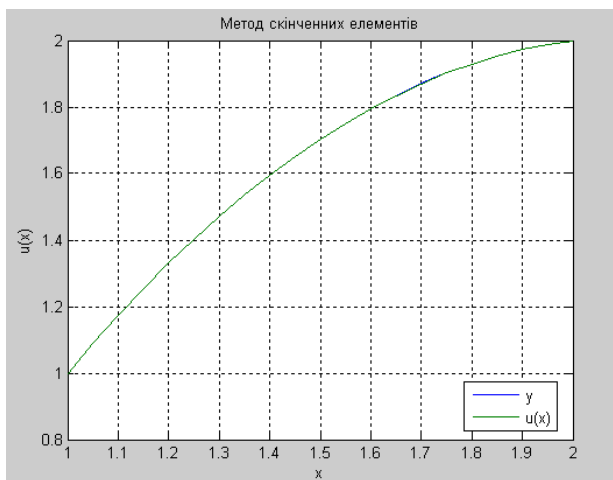
% допоміжні змінні
n = length(X); n1 = n - 1;
h = X(2) - X(1); h_2 = 0.5 .* h; hh = h .* h;
hh_2 = h_2 .* h; hh_6 = hh ./ 6;

% формування коефіцієнтів
% A(k),C(k),B(k),F(k), k=1,...,n для СЛАР
A = zeros(1, n); C = ones(1,n); B = A; F = A;
x = X(1) + h_2; rxx = h_2 .* rfun(x);
pxx = pfun(x); fxx = hh_2 .* ffun(x);
akp = kfun(x); pkp= hh_6 .* pxx;
for k = 2 : n1
    ak = akp; pk = pkp; rx = rxx; px = pxx; fx = fxx;
    A(k) = ak + pk - rx;
    xx = X(k) + h_2; rxx = h_2 .* rfun(xx);
    pxx = pfun(xx); akp = kfun(xx); pkp = hh_6 .* pxx;
    B(k) = akp + pkp + rxx;
    C(k) = A(k) + B(k) + hh_2 .* (px + pxx);
    fxx = hh_2 .* ffun(xx);
    F(k) = fx + fxx; x = xx;
end

% знаходження сіткової функції при розв'язанні СЛАР
% методом немонотонної прогонки
[Y AL] = m_progn(A, C, B, F);
return

```

Результат виконання М-файлу *МКЕ_MAIN.M*:



Похибка в нормі $C=0.00075011$

ЛЕКЦІЯ № 07

ТЕМА № 4. Різницький метод розв'язання крайових задач для одновимірних лінійних рівнянь гіперболічного типу першого порядку. Консервативні схеми для квазілінійних задач.

Розглянемо застосування МРС до розв'язання нестационарних крайових задач математичної фізики. Почнемо з одновимірних лінійних рівнянь гіперболічного типу першого порядку – *рівнянь переносу*.

Нехай в області $\bar{D} = D \cup \partial D_1 \cup \partial D_2 = [a, b] \times [0, T]$ маємо *мішану задачу Коші*:

$$\frac{\partial u(x, t)}{\partial t} + c \frac{\partial u(x, t)}{\partial x} = f(x, t), \quad (4.1)$$

$$c = \text{const} > 0, \quad (x, t) \in D \equiv (a, b] \times (0, T],$$

з крайовою умовою

$$u(a, t) = \mu_1(t), \quad t \in \partial D_2 \equiv [0, T] \quad (4.2)$$

і початковою умовою

$$u(x, 0) = u_0(x), \quad x \in \partial D_1 \equiv [a, b] \quad (4.3)$$

Будемо вважати, що:

- початкові і крайові дані мають достатню гладкість і для них виконуються необхідні умови узгодження в точці $(a; 0)$;
- функція f неперервно диференційована;
- шукана функція u також достатньо гладка.

п.1 Побудова сімейства різницьких схем

Уведемо рівномірні сітки

$$\bar{\omega}_h = \{x_k = a + (k-1)h, \quad k = \overline{1, N}, \quad b - a = (N-1)h\};$$

$$\bar{\omega}_\tau = \{t_j = j\tau, \quad j = \overline{0, M}, \quad T = M\tau\},$$

і замінимо область $\bar{D}$ сітковою областю $\bar{\omega}_{h\tau} = \bar{\omega}_h \times \bar{\omega}_\tau$, а також будемо використовувати наступні позначення для сіткових функцій

$$y_k^j \equiv y_k \equiv y = u(x_k, t_j), \quad y_k^{j+1} \equiv \hat{y}_k \equiv \hat{y} = u(x_k, t_{j+1}).$$

Виберемо чотири сіткові шаблони :

- $S\{(x_{k-1}, t_j), (x_k, t_j), (x_k, t_{j+1})\};$
- $S\{(x_k, t_j), (x_{k+1}, t_j), (x_k, t_{j+1})\};$
- $S\{(x_k, t_j), (x_k, t_{j+1}), (x_{k-1}, t_{j+1})\};$
- $S\{(x_{k-1}, t_j), (x_k, t_j), (x_k, t_{j+1}), (x_{k-1}, t_{j+1})\},$

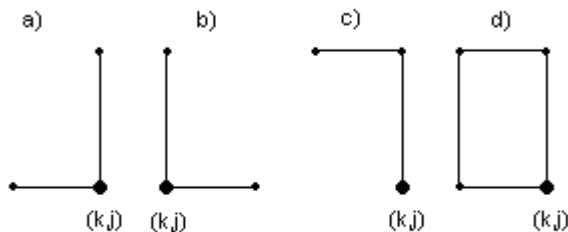


Рис. 4.1 Сіткові шаблони.

на яких, методом безпосередньої заміни диференціальних операторів на різниці, запишемо для (4.1) наступні різницькі рівняння :

$$\frac{\hat{y}_k - y_k}{\tau} + c \frac{y_k - y_{k-1}}{h} = \bar{\varphi}_{k,j} \equiv f_{k-0.5, j+0.5}; \quad (4.4)$$

$$\frac{\hat{y}_k - y_k}{\tau} + c \frac{y_{k+1} - y_k}{h} = f_k^j; \quad (4.5)$$

$$\frac{\hat{y}_k - y_k}{\tau} + c \frac{\hat{y}_k - \hat{y}_{k-1}}{h} = \bar{\varphi}_{k,j}; \quad (4.6)$$

$$\frac{1}{2} \left(\frac{\hat{y}_{k-1} - y_{k-1}}{\tau} + \frac{\hat{y}_k - y_k}{\tau} \right) + \frac{c}{2} \left(\frac{y_k - y_{k-1}}{h} + \frac{\hat{y}_k - \hat{y}_{k-1}}{h} \right) = \bar{\varphi}_{k,j}. \quad (4.7)$$

для сіткових точок (x_k, t_j) з $k = \overline{2, N}$; $j = \overline{0, M-1}$.

Апроксимація крайової умови

$$y_1^j = \mu_1(t_j), \quad j = \overline{0, M}, \quad (4.8)$$

і початкової умови

$$y_k^0 = u_0(x_k), \quad k = \overline{1, N}, \quad (4.9)$$

виконується точно.

Озн.4.1. Сукупність точок сітки, які належать лінії (гіперплощині) $t = t_j$, називають шаром.

Озн.4.2. Якщо для нестационарної задачі різницева схема записана на двох шарах сітки, то таку РС називають двошаровою, на трьох – тришаровою.

Озн.4.3. Якщо в записі дво(три)шарової РС на верхньому шарі присутнє тільки одне невідоме значення сіткової функції, то таку схему називають явною, якщо більше одного – неявною.

Отже за цим означенням схеми (4.4), (4.5) явні, а (4.6), (4.7) – неявні.

п.2 Похибка апроксимації

Розглянемо схему (4.4). Для точки (x_k, t_j) маємо розклади функцій в ряд Тейлора :

$$\begin{aligned} \hat{y}_k &= y_k + \tau \cdot u'_t(x_k, t_j) + 0.5\tau^2 u''_{tt}(x_k, t_j) + O(\tau^3), \\ y_{k-1} &= y_k - h \cdot u'_x(x_k, t_j) + 0.5h^2 u''_{xx}(x_k, t_j) + O(h^3) \\ \bar{\varphi}_{k,j} &= f_{k,j} + 0.5\tau \cdot f'_t(x_k, t_j) - 0.5h \cdot f'_x(x_k, t_j) + O(h^2 + \tau^2) \end{aligned}$$

які при підстановці в вираз для похибки

$$\begin{aligned} \psi_{k,j} &= (y_{t,j} + c y_{\bar{x},k} - \bar{\varphi}_{k,j}) - (u'_t(x_k, t_j) + c u'_x(x_k, t_j) - f(x_k, t_j)) = \\ &= 0.5\tau [u''_{tt}(x_k, t_j) - f''_t(x_k, t_j)] + 0.5h [f'_x(x_k, t_j) - c u''_{xx}(x_k, t_j)] + O(h^2 + \tau^2) \end{aligned}$$

дають $O(h + \tau)$ при $f \in C^{(1,1)}(D)$, $u \in C^{(2,2)}(D)$. Аналогічний порядок маємо в сітковій $\|\cdot\|_C$.

Завдання 1. Для схеми (4.5) в околі точки (x_k, t_j) і для схеми (4.6) в околі точки (x_k, t_{j+1}) отримати похибку $O(h + \tau)$. Для схеми (4.7) в околі точки $(x_{k-0.5}, t_{j+0.5})$ отримати похибку $O(h^2 + \tau^2)$ і необхідні умови гладкості розв'язку $u(x, t)$.

Озн.4.4. Двошарова РС записана в формі з розділеним групуванням сіткової функції на шарах сітки, якщо вона представлена у вигляді

$$\sum_k \alpha_k \hat{y}_{n+k} = \sum_r \beta_r y_{n+r} + \varphi_{n,j}, \quad (4.10)$$

де сумування на кожному шарі виконується по точкам шаблону S біля n -го вузла, для якого в нумерації коефіцієнтів α_k виконується умова $|\alpha_0| = \max_k |\alpha_k|$.

Озн.4.5. Двошарова РС записана в канонічній формі, якщо вона представлена у вигляді

$$B(t_j) \frac{\hat{y}_k - y_k}{\tau} + A(t_j) y_k = \varphi_{k,j}, \quad (4.11)$$

де $B(t_j)$, $A(t_j)$ – деякі сіткові оператори на точках шаблону S .

Озн.4.6. Лінійна двошарова РС називається *рівномірно стійкою* за початковими даними, якщо

$$\|y^I(t) - y^II(t)\| \leq K \|y^I(t_*) - y^II(t_*)\|, \quad t_0 \leq t_* < t < T, \quad (4.12)$$

де $K \in \text{const}$ і не залежить від t_*, h , а y^I, y^II – розв’язки РС $A_h y = \varphi$ з різними початковими даними і незмінними правими частинами (можна з $\varphi = 0$).

п.3 Стійкість і збіжність різницевої схеми

Ознака рівномірної стійкості. Якщо $A_h y^I = A_h y^II$, то для рівномірної стійкості за початковими даними достатньо, щоб при $\forall j$ виконувалась нерівність

$$\|\hat{y}^I - \hat{y}^II\| \leq (1 + C\tau) \|y^I - y^II\|, \quad C \in \text{const}, \quad C \geq 0. \quad (4.13)$$

Озн.4.7. Нехай лінійна двошарова РС рівномірно стійка за початковими даними і така, що на деякому сітковому шарі $y^I = y^II$, а на наступному шарі виконується нерівність

$$\|\hat{y}^I - \hat{y}^II\| \leq \vartheta \tau \|\varphi^I - \varphi^II\|, \quad (4.14)$$

де $\vartheta \in \text{const}$, $\vartheta > 0$ і не залежить від τ, h , тоді РС *стійка за правою частиною*.

Дослідження стійкості. В залежності від вибраної норми, найчастіше використовують наступні способи дослідження стійкості:

- А) принцип максимуму** (у нормі $\|\cdot\|_C$);
- В) метод відокремлення змінних (метод Фур’є)** (у нормі $\|\cdot\|_{L_2}$);
- С) метод енергетичних нерівностей** (у нормі $\|\cdot\|_A, \|\cdot\|_{A^{-1}}$);
- Д) метод операторних нерівностей** (у нормі $\|\cdot\|_A$).

Розглянемо перші два методи в застосуванні до наших РС.

Принцип максимуму для двошарової РС (4.10) формулюється так:

1) РС рівномірно стійка за початковими даними за умови

$$(1 + C\tau) |\alpha_0| \geq \sum_{k \neq 0} |\alpha_k| + \sum_r |\beta_r|, \quad C \in \text{const}, \quad (4.15)$$

2) РС стійка за правою частиною, якщо виконується (4.15) і умова

$$|\alpha_0| - \sum_{k \neq 0} |\alpha_k| \geq \frac{\theta}{\tau}, \quad \theta \in \text{const}, \quad (4.16)$$

Найчастіше в оцінках (4.15), (4.16) обирають $C = 0$ і $\theta = 1$.

Застосуємо це для однорідної схеми (4.4), яку попередньо представимо у формі (4.10)

$$\alpha_0 \cdot \hat{y}_k = \beta_{-1} \cdot y_{k-1} + \beta_0 \cdot y_k,$$

а отже $\alpha_0 = \frac{1}{\tau}$, $\beta_{-1} = \frac{c}{h}$, $\beta_0 = \frac{1}{\tau} - \frac{c}{h}$. Підставимо ці значення в (4.15) при $C = 0$,

отримаємо нерівність $\frac{1}{\tau} - \frac{c}{h} \geq \left| \frac{1}{\tau} - \frac{c}{h} \right|$, розв’язок якої дає умову Куранта

$$c\tau \leq h. \quad (4.17)$$

Таким чином, за виконання умови Куранта, РС (4.4) є *умовно-стійка* за початковими даними в нормі $\|\cdot\|_C$. При виконанні (4.17), виконується і умова (4.16), а отже маємо умовну стійкість схеми (4.4) за правою частиною.

Завдання 2. Для однорідної схеми (4.6) виконати аналогічні дослідження її стійкості у нормі $\|\cdot\|_C$.

Метод відокремлення змінних. Якщо двошарову однорідну РС у формі (4.11) перепишемо у форму (4.10) $B(t_j) \hat{y}_k = (B(t_j) - \tau \cdot A(t_j)) y_k$, то легко отримаємо рівняння для

похибки $B(t_j) \cdot \delta \hat{y}_k = (B(t_j) - \tau \cdot A(t_j)) \cdot \delta y_k$. Будемо шукати частинний розв'язок цього рівняння методом розділення змінних:

$$\delta y_k^j = \rho_q^j \exp(iqx_k), \quad q = 0, \pm 1, \pm 2, \dots; \quad i = \sqrt{-1}. \quad (4.18)$$

При цьому $\delta \hat{y}_k = \rho_q^{j+1} \exp(iqx_k) = \rho_q \cdot \delta y_k^j$, де ρ_q – множник зростання q -ї гармоніки при переході з шару j на шар $j+1$. Враховуючи ознаку рівномірної стійкості (4.13), для схем з постійними операторами $B(t_j) = B$, $A(t_j) = A$, після підстановки (4.18) отримаємо ознаку стійкості за початковими даними у формі умови на *множник зростання* ρ_q при довільному індексі гармоніки q

$$|\rho_q| \leq 1 + C\tau, \quad C \in \text{const}, \quad (4.19)$$

(частіше обирають $C = 0$).

Підставимо (4.18) в однорідну схему (4.4) і отримаємо вираз для значення множника зростання ρ_q

$$\rho_q = 1 - \frac{c\tau}{h} (1 - \exp(-iqh)).$$

Якщо умова Куранта (4.17) не виконується, то для тих гармонік, у яких $\cos(qh) = -1$, модуль множника зростання $|\rho_q| = \left| 1 - \frac{2c\tau}{h} \right| = \frac{2c\tau}{h} - 1 > 1$, а отже амплітуди цих гармонік необмежено зростають і РС (4.4) не стійка. При виконанні умови (4.17), різницева задача (4.4), (4.8), (4.9) збігається до розв'язку диференціальної задачі зі швидкістю $O(h + \tau)$.

Для однорідної схеми (4.5):

$$\rho_q = 1 + \gamma(1 - \exp(iqh)) = 1 + \gamma(1 - \cos(qh)) - i\gamma \sin(qh), \quad \gamma = \frac{c\tau}{h};$$

$$|\rho_q|^2 = 1 + 4\gamma(1 + \gamma)\sin^2(0.5qh).$$

З останньої рівності видно, що $|\rho_q| > 1$ при $\forall \gamma$, а отже ця схема є *абсолютно нестійка* і при $c > 0$ не використовується для розрахунків.

Для однорідної схеми (4.6):

$$\rho_q(1 + \gamma(1 - \exp(-iqh))) = 1; \Rightarrow |\rho_q| \leq 1, \quad \forall q,$$

а отже ця схема є *безумовно стійка*. Різницева задача (4.6), (4.8), (4.9) збігається до розв'язку диференціальної задачі зі швидкістю $O(h + \tau)$.

Для однорідної схеми (4.7):

$$\rho_q = \exp(-iqh) \frac{(1 + \gamma) + (1 - \gamma)\exp(iqh)}{(1 + \gamma) + (1 - \gamma)\exp(-iqh)}; \Rightarrow |\rho_q| = 1, \quad \forall q,$$

а отже ця схема є *безумовно стійка*. Критерій стійкості по правій частині (4.16) виконується при $\theta = 2$. Різницева задача (4.7) – (4.9) безумовно збігається до розв'язку $u(x, t) \in C^{(3,3)}(D)$ диференціальної задачі (4.1)-(4.3) у нормі $\|\cdot\|_{L_2}$ зі швидкістю $O(h^2 + \tau^2)$.

п.4 Знаходження розв'язку

Для РС з шаблонами **a), c), d)** маємо наступний алгоритм розрахунку:

1. Обираємо сітку і визначаємо незалежні від k, j константи A, B, C ;
2. Визначаємо сіткову функцію $y_k^0 = u_0(x_k)$, $k = \overline{1, N}$;
3. Послідовно, для $j = 0, 1, \dots, M - 1$ визначаємо

$$3.1) \quad \text{значення } y_1^{j+1} = \mu_1(t_{j+1})$$

3.2) сіткову функцію y_k^{j+1}

шаблон а) $y_k^{j+1} = A * y_{k-1} + B * y_k + \tau * \bar{\varphi}_{k,j}, \quad k = \overline{2, N};$
 $A = \gamma, \quad B = 1 - \gamma.$

шаблон б) $y_k^{j+1} = A * y_{k-1}^{j+1} + B * y_k + C * \bar{\varphi}_{k,j}, \quad k = \overline{2, N};$
 $B = (1 + \gamma)^{-1}, \quad A = \gamma * B, \quad C = \tau * B.$

шаблон в) $y_k^{j+1} = A * (y_k - y_{k-1}^{j+1}) + y_{k-1} + C * \bar{\varphi}_{k,j}, \quad k = \overline{2, N};$
 $B = (1 + \gamma)^{-1}, \quad A = (1 - \gamma) * B, \quad C = 2 * \tau * B.$

Тепер розглянемо варіант $c = \text{const} < 0$. У цьому випадку для рівняння (4.1) ставиться крайова умова у правій точці границі ∂D_1

$$u(b, t) = \mu_2(t), \quad t \in \partial D_2. \quad (4.20)$$

З розглянутих РС використовують явну схему (4.5), яка є умовно-стійка при умові $|c|\tau \leq h$ Неявну безумовно стійку схему записують у вигляді

$$\frac{\hat{y}_k - y_k}{\tau} + c \frac{\hat{y}_{k+1} - \hat{y}_k}{h} = \bar{\varphi}_{k,j}, \quad (4.21)$$

$$k = N - 1, N - 2, \dots, 1.$$

У випадку знакозмінного коефіцієнта $c(x, t)$ в рівнянні (4.1) ставляться обидві крайові умови – (4.2) і (4.20). Вихідне рівняння апроксимують неявною безумовно стійкою схемою, яку записують у вигляді

$$\frac{\hat{y}_k - y_k}{\tau} + \hat{c}_k \frac{\hat{y}_{k+1} - \hat{y}_{k-1}}{2h} = \bar{\varphi}_{k,j}, \quad (4.22)$$

$$k = \overline{2, N - 1}.$$

Для розрахунків використовують *монотонну прогонку*, стійкість якої буде тільки при виконанні умови $|c|\tau \leq h$.

Розглянуті нами РС можна узагальнити на *рівняння переносу з поглинанням*

$$\frac{\partial u(x, t)}{\partial t} + c \frac{\partial u(x, t)}{\partial x} + p * u(x, t) = f(x, t),$$

$$c = \text{const}, \quad p = \text{const} > 0;$$

а також на *багатовимірні рівняння переносу*.

п.5 Консервативні схеми для квазілінійних рівнянь переносу.

Розглянемо побудову РС для *нев'язкого рівняння Бюргерса*, яке може бути представлене в двох, еквівалентних між собою, формах – *недивергентній* (4.23) і *дивергентній* (4.24)

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = 0, \quad (4.23)$$

$$\frac{\partial u}{\partial t} + \frac{1}{2} \frac{\partial (u^2)}{\partial x} = 0, \quad (4.24)$$

Покажемо, що для рівняння (4.24) має місце певний фізичний закон збереження. Проінтегруємо це рівняння по окремій комірці сітки $\bar{\omega}_{h\tau}$, і отримаємо точне інтегральне співвідношення

$$\int_{x_{n-1}}^{x_n} (u(x, t_{m+1}) - u(x, t_m)) dx + \frac{1}{2} \int_{t_m}^{t_{m+1}} (u^2(x_n, t) - u^2(x_{n-1}, t)) dt = 0. \quad (4.25)$$

Аналогічно можна отримати співвідношення для всієї області $\bar{D}$

$$\int_a^b (u(x, T) - u(x, 0)) dx + \frac{1}{2} \int_0^T (u^2(b, t) - u^2(a, t)) dt = 0. \quad (4.26)$$

Якщо співвідношення (4.25) просумувати по всіх комірках сітки $n = \overline{2, N}$, $m = \overline{0, M-1}$, і скористатися адитивністю інтеграла, то отримаємо (4.26). Отже маємо висновок: закон збереження у всій області є точне наслідування закону збереження в окремій комірці.

Тепер перейдемо до побудови різницевих рівнянь, які будуть апроксимувати (4.23), (4.24). Будемо вважати, що крайові і початкові умови для цих рівняння такі, що коректно і однозначно визначають функцію $u(x, t) > 0$ і не мають ні *слабких розривів розв'язку* (похідних $u(x, t)$), а ні *сильних розривів* (самої функції $u(x, t)$).

Використаємо шаблон с) і побудуємо аналоги схеми (4.6) для (4.23), (4.24)

$$\frac{\hat{y}_k - y_k}{\tau} + \hat{y}_k \frac{\hat{y}_k - \hat{y}_{k-1}}{h} = 0, \quad (4.27)$$

$$\frac{\hat{y}_k - y_k}{\tau} + \frac{1}{2} \frac{(\hat{y}_k)^2 - (\hat{y}_{k-1})^2}{h} = 0. \quad (4.28)$$

Попробуємо отримати різницевий аналог закону (4.26) для наших схем і почнемо з (4.28)

$$\sum_{k=2}^N \sum_{j=0}^{M-1} \left(\frac{\hat{y}_k - y_k}{\tau} + \frac{1}{2} \frac{(\hat{y}_k)^2 - (\hat{y}_{k-1})^2}{h} \right) * h \tau = 0;$$

$$h \sum_{k=2}^N (y_k^M - y_k^0) + \frac{1}{2} \tau \sum_{j=0}^{M-1} ((\hat{y}_N)^2 - (\hat{y}_1)^2) = 0.$$

Останнє співвідношення є аналогом (4.26) для сіткової функції, де перший інтеграл обчислено по КФ правих прямокутників, а другий – по КФ лівих прямокутників.

Озн.4.8. Різницева схема називається *консервативною*, якщо для неї виконується сітковий аналог закону збереження диференціальної задачі. РС, для якої це не виконується, називається *неконсервативною*.

Виконаємо аналогічні перетворення для схеми (4.27):

$$\sum_{k=2}^N \sum_{j=0}^{M-1} \left(\frac{\hat{y}_k - y_k}{\tau} + \hat{y}_k \frac{\hat{y}_k - \hat{y}_{k-1}}{h} \right) * h \tau = 0;$$

$$h \sum_{k=2}^N (y_k^M - y_k^0) + \tau \sum_{k=2}^N \sum_{j=0}^{M-1} \left((\hat{y}_k)^2 - \hat{y}_k \hat{y}_{k-1} \pm \frac{1}{2} (\hat{y}_{k-1})^2 \right) = 0;$$

$$h \sum_{k=2}^N (y_k^M - y_k^0) + \frac{1}{2} \tau \sum_{j=0}^{M-1} ((\hat{y}_N)^2 - (\hat{y}_1)^2) + \Delta = 0,$$

де

$$\Delta = \frac{1}{2} \tau \sum_{k=2}^N \sum_{j=0}^{M-1} (\hat{y}_k - \hat{y}_{k-1})^2 > 0$$

це *дисбаланс*, на який порушуються сітковий аналог закону (4.26), а отже схема (4.27) розв'язує ІНШУ фізичну задачу !

Завдання 3. Використати шаблон **а)** для побудови РС, яка апроксимує рівняння (4.24) і дослідити її на консервативність.

Розглянемо приклади розв'язання задачі (4.1)-(4.3) з постійним коефіцієнтом c . М-файл *gulprjmain.m* реалізує явну схему (4.4), *gulprnmain.m* – неявну схему (4.6). Результати виконання цих програм подано на Рис. 4.2 - Рис. 4.7.

```
%% gulprjmain.m
% Розв'язання гіперболічного рівняння першого порядку
% з постійним коефіцієнтом (явна схема)
clear all, close all, clc
```

```

c = 1;
f = @(x,t) (3+x.*(1-x)+t.*(3-2.*x));
u0 = @(x) (x.*(3-x));
g1 = @(t) (0);
a = 0; b = 3;

N = 301; x = linspace(a,b,N); h = x(2)-x(1); h2 = 0.5*h;
tau = 0.001; M = 5000; tau2 = 0.5*tau;

A = c.*tau./h; B = 1-A;

t = 0; y = u0(x); U0 = y;
for j = 1:M
    t = t+tau;
    yr = y(1); y(1) = g1(t);
    for k = 2:N
        yl = yr; yr = y(k);
        y(k) = A.*yl + B.*yr + tau.*f(x(k)-h2,t-tau2);
    end
    if j.*2 == M
        U1 = y;
    end
end
save fileYJ y

% побудова графіків розв'язку при t=0, M*tau/2, M*tau :
t=M.*tau; tau2=t./2; T1=num2str(tau2); T=num2str(t);
plot(x,U0,'b.-',x,U1,'k.--',x,y,'r*-');
xlabel('x'); ylabel('u');
title('Наближений розв'язок (явна РС)');
legend('u(x,0)', ['u(x,',T1,')'], ['u(x,',T,')'], 'Location','Best');
grid on

%% gupnmain.m
% Розв'язання гіперболічного рівняння першого порядку
% з постійним коефіцієнтом (неявна схема)
clear all, close all, clc

c = 1;
f = @(x,t) (3+x.*(1-x)+t.*(3-2.*x));
u0 = @(x) (x.*(3-x));
g1 = @(t) (0);
a = 0; b = 3;

N = 301; x = linspace(a,b,N); h = x(2)-x(1); h2 = 0.5*h;
tau = 0.01; M = 500; tau2 = 0.5*tau;

C = c.*tau./h; B = 1./(1+C); A = C.*B; C = tau.*B;

t = 0; y = u0(x); U0 = y;
for j = 1:M
    t = t+tau; y(1) = g1(t);
    for k = 2:N
        y(k) = A.*y(k-1) + B.*y(k) + C.*f(x(k)-h2,t-tau2);
    end
    if j.*2 == M
        U1 = y;
    end
end

% побудова графіків розв'язку при t=0, M*tau/2, M*tau :
t=M.*tau; tau2=t./2; T1=num2str(tau2); T=num2str(t);

```

```

plot(x,U0,'b.-',x,U1,'k.--',x,y,'r*-');
xlabel('x'); ylabel('u');
title('Наближений розв'язок (явна РС)');
legend('u(x,0)', ['u(x,',T1,')'], ['u(x,',T,')'], 'Location','Best');
grid on; pause(); close all

u = @(x,t)((t+1).*x.*(3-x));
U = u(x,t); U0 = y;
load fileYJ y
plot(x, y, 'b.-', x, U, 'k.--');
xlabel('x'); ylabel('u');
title(['Набл.роз-к(явна РС) і точний розв'язок для t=',T]);
legend('явна','модель','Location','Best');
grid on; pause(); close all

plot(x, abs(y-U));
xlabel('x'); ylabel('u');
title(['Похибка |явна РС-модель| при t=',T]);
legend('|явна-модель|','Location','Best');
grid on; norm(y-U,inf); pause(); close all

plot(x, U0, 'b.-', x, U, 'k.--');
xlabel('x'); ylabel('u');
title(['Набл.роз-к(неявна РС) і точний розв'язок для t=',T]);
legend('неявна','модель','Location','Best');
grid on; pause(); close all

plot(x, abs(U0-U));
xlabel('x'); ylabel('u');
title(['Похибка |неявна РС-модель| при t=',T]);
legend('|неявна-модель|','Location','Best');
grid on; norm(U0-U,inf)

```

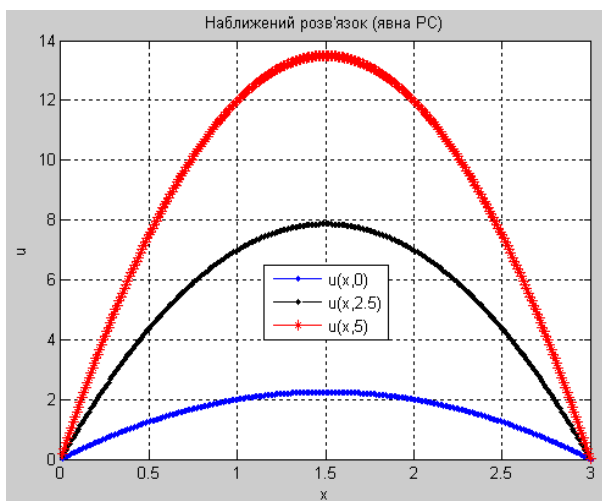


Рис. 4.2 Наближений розв'язок (явна РС)

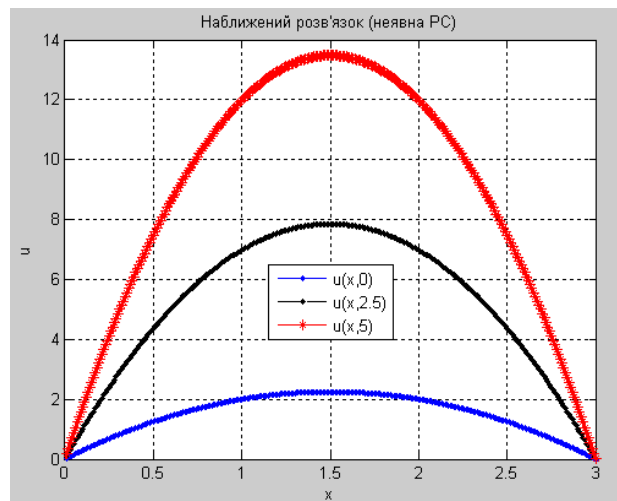


Рис. 4.3 Наближений розв'язок (неявна РС)

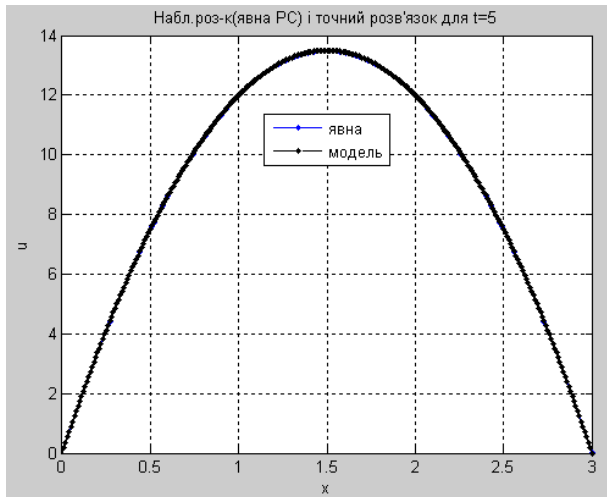


Рис. 4.4 Точний і розв'язок по явній РС

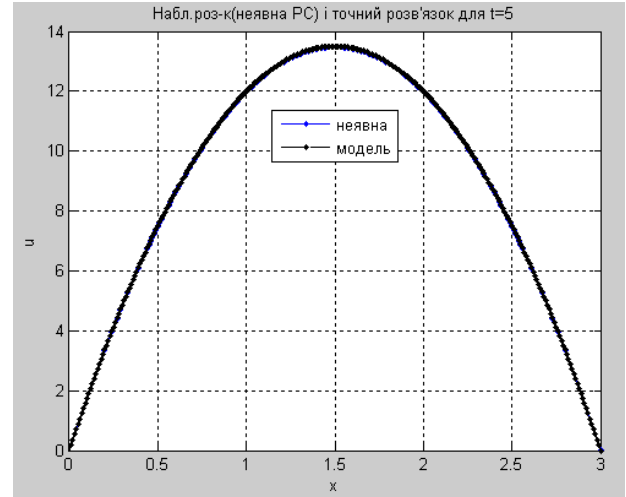


Рис. 4.5 Точний і розв'язок по неявній РС

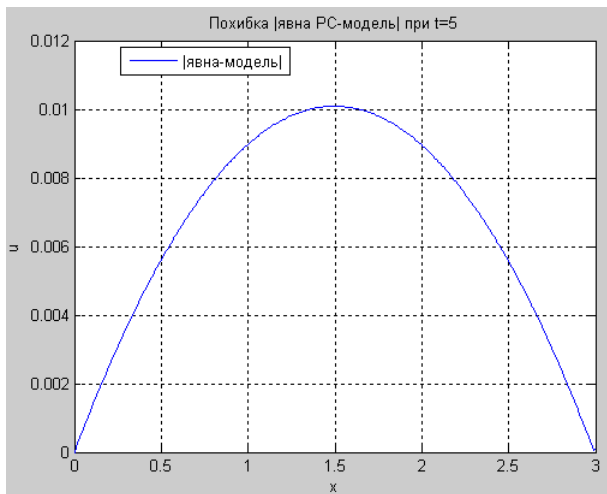


Рис. 4.6 Абсолютна похибка (явна РС)

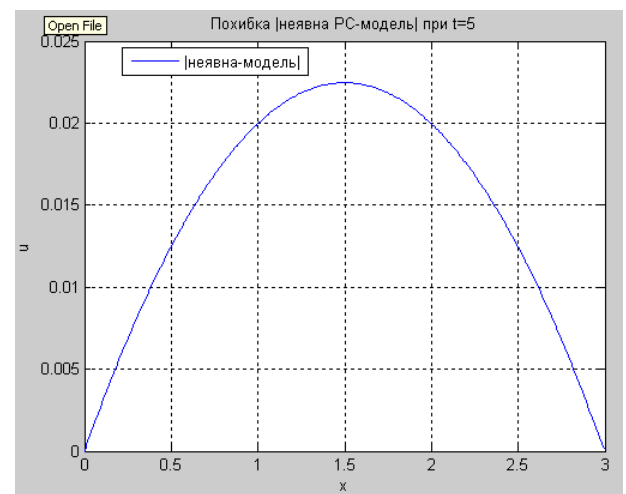


Рис. 4.7 Абсолютна похибка (неявна РС)

Розглянемо приклад розв'язання задачі (4.1)-(4.3) зі змінним коефіцієнтом $c(x,t)$. М-файл *gulpstain.m* реалізує схему (4.22). Результати виконання програми подано на Рис. 4.8 - Рис. 4.9.

```
%% gulpstain.m
% Розв'язання гіперболічного рівняння першого порядку
% зі змінним коефіцієнтом  $c(x,t)$ 
clear all, close all, clc

c = @(x,t) ((t+0.5).*(0.6-x).^3);
f = @(x,t) (sin(x-t));
u0 = @(x) (x.*(1-x));
g1 = @(t) (t);
g2 = @(t) (2.*t);
a = 0; b = 1;

N = 101; x = linspace(a,b,N); h = x(2)-x(1); N1 = N-1;
tau = 0.01; M = 200;

th = 0.5.*tau./h;

A = zeros(N,1); B = A; C = -ones(N,1); F = A;
t = 0; y = u0(x); U0 = y; U = zeros(N,M+1); U(:,1) = y;
for j = 1:M
    t = t+tau;
```

```

% формуємо коефіцієнти СЛАР
F(1) = -g1(t);
for k = 2:N1
    z = c(x(k),t).*th; A(k) = -z; B(k) = z;
    F(k) = -(y(k)+tau.*f(x(k),t));
end
F(N) = -g2(t);

[y, alfa] = m_progn(A,C,B,F);
U(:,j+1) = y;

if j.*2 == M
    U1 = y;
end
end
clear alfa A C B F

% побудова графіків розв'язку при t=0, M*tau/2, M*tau :
t = M.*tau; z = t./2; T1 = num2str(z); T = num2str(t);
plot(x,U0,'b.-',x,U1,'k.--',x,y,'r*');
xlabel('x'); ylabel('u');
title('Наближений розв'язок');
legend('u(x,0)', ['u(x,',T1,')'], ['u(x,',T,')'], 'Location', 'Best');
grid on, pause(); close all, clear y U0 U1

tt = linspace(0,t,M+1);
[T,X] = meshgrid(tt,x);
surfc(X,T,U);
xlabel('x'); ylabel('t'); zlabel('u');
title('Наближений розв'язок u(X,T)');
shading flat

```

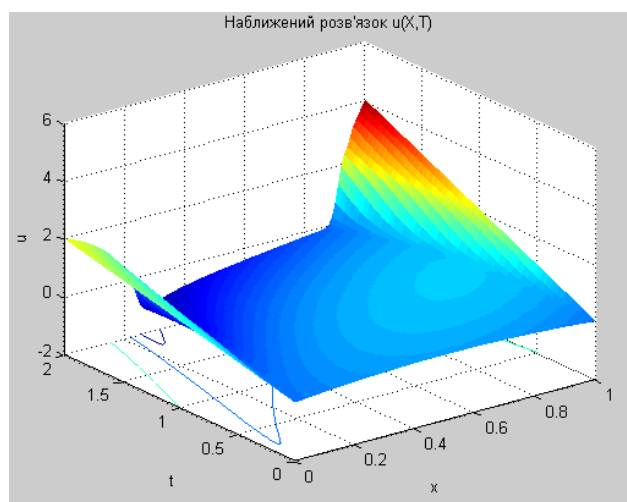
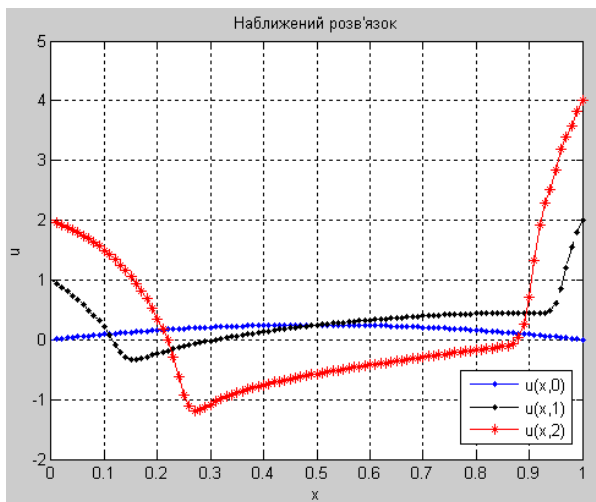


Рис. 4.8 Наближений розв'язок при $t=0,1,2$ Рис. 4.9 Наближений розв'язок $u(X,T)$

ЛЕКЦІЯ № 08

ТЕМА № 5. Різницький метод розв'язання крайових задач для рівнянь параболічного типу.

5.1 Різницькі схеми для лінійних одновимірних задач

Потрібно знайти в області $\bar{D} = D \cup \partial D = [a, b] \times [0, T]$ розв'язок мішаної задачі Коші для рівняння параболічного типу :

$$\frac{\partial u(x, t)}{\partial t} = \frac{\partial^2 u(x, t)}{\partial x^2} + f(x, t), \quad (x, t) \in D \equiv (a, b) \times (0, T], \quad (5.1)$$

з крайовими умовами першого роду

$$u(a, t) = \mu_1(t), \quad u(b, t) = \mu_2(t), \quad t \in [0, T] \quad (5.2)$$

і початковою умовою

$$u(x, 0) = u_0(x), \quad x \in [a, b] \quad (5.3)$$

Будемо вважати, що:

- початкові і крайові дані мають достатню гладкість і для них виконуються необхідні умови узгодження в точках $(a; 0)$, $(b; 0)$;
- функція f неперервно диференційована;
- шукана функція u також достатньо гладка.

п.1 Побудова сімейства різницьких схем

Уведемо рівномірні сітки

$$\bar{\omega}_h = \{x_k = a + (k-1)h, \quad k = \overline{1, N}, \quad b - a = (N-1)h\};$$
$$\bar{\omega}_\tau = \{t_j = j\tau, \quad j = \overline{0, M}, \quad T = M\tau\},$$

і замінимо область $\bar{D}$ сітковою областю $\bar{\omega}_{h\tau} = \bar{\omega}_h \times \bar{\omega}_\tau$, а також будемо використовувати наступні позначення

$$\dot{u} = \frac{\partial u}{\partial t}, \quad Lu = \frac{\partial^2 u}{\partial x^2}; \quad \dot{u} = Lu + f, \quad L\dot{u} = L^2u + Lf, \quad L^2u = L\dot{u} - Lf;$$

$$y_{t,j} = \frac{\hat{y}_k - y_k}{\tau}, \quad \Lambda y = y_{\bar{x}x,k};$$

$$0 \leq \sigma \leq 1; \quad y^{(\sigma)} \equiv Y = \sigma \hat{y} + (1 - \sigma)y = 0.5(\hat{y} + y) + \tau(\sigma - 0.5)y_{t,j}.$$

На шеститочковому шаблоні $S\{(x_{k-1}, t_j), (x_k, t_j), (x_{k+1}, t_j), (x_{k-1}, t_{j+1}), (x_k, t_{j+1}), (x_{k+1}, t_{j+1})\}$ рівнянню (5.1) поставимо у відповідність однопараметричну сім'ю двошарових схем – схему з вагою

$$y_{t,j} = \Lambda Y + \varphi, \quad (5.4)$$

$$k = \overline{2, N-1}; \quad j = \overline{0, M-1}.$$

Крайові і початкові умови апроксимуються точно

$$y_1^j = \mu_1(t_j), \quad y_N^j = \mu_2(t_j), \quad j = \overline{1, M}; \quad (5.5)$$

$$y_k^0 = u_0(x_k), \quad k = \overline{1, N}. \quad (5.6)$$

Від вибору параметра σ залежить точність і стійкість схеми (5.4). Розглянемо найбільш уживані частинні випадки параметра σ .

Явна схема. При $\sigma = 0$ на шаблоні $S\{(x_{k+1}, t_j), (x_k, t_j), (x_k, t_{j+1})\}$ схему (5.4) запишемо у формі (4.10)

$$\frac{1}{\tau} y_k^{j+1} = \frac{1}{h^2} y_{k-1}^j + \left(\frac{1}{\tau} - \frac{2}{h^2} \right) y_k^j + \frac{1}{h^2} y_{k+1}^j + \varphi, \quad (5.7)$$

$$k = 2, \dots, N-1,$$

де введено позначення

$$\varphi \equiv \bar{\varphi}_{k,j} = f(x_k, \bar{t}_j), \quad \bar{t}_j \equiv t_{j+0.5} = t_j + 0.5\tau.$$

Використовуючи формулу (5.7) послідовно для шарів $j = \overline{1, M-1}$ і додаючи умови (5.5), (5.6) можемо знайти сіткову функцію для всієї області $\bar{\omega}_{h\tau}$.

Неявна схема. При $0 < \sigma < 1$ схема (5.4), записана у формі (4.10), має вигляд

$$-\frac{\sigma}{h^2} y_{k-1}^{j+1} + \left(\frac{1}{\tau} + \frac{2\sigma}{h^2} \right) y_k^{j+1} - \frac{\sigma}{h^2} y_{k+1}^{j+1} = \frac{1-\sigma}{h^2} y_{k-1}^j + \left(\frac{1}{\tau} - \frac{2(1-\sigma)}{h^2} \right) y_k^j + \frac{1-\sigma}{h^2} y_{k+1}^j + \varphi, \quad (5.8)$$

$$k = 2, \dots, N-1.$$

Чисто неявна схема. При $\sigma = 1$ на шаблоні $S\{(x_k, t_j), (x_{k+1}, t_{j+1}), (x_k, t_{j+1})\}$ схему (5.4) запишемо у формі (4.10)

$$-\frac{1}{h^2} y_{k-1}^{j+1} + \left(\frac{1}{\tau} + \frac{2}{h^2} \right) y_k^{j+1} - \frac{1}{h^2} y_{k+1}^{j+1} = \frac{1}{\tau} y_k^j + \varphi, \quad (5.9)$$

$$k = 2, \dots, N-1.$$

Симетрична схема (схема Кранка-Ніколсона). При $\sigma = 0.5$ схема (5.4), записана у формі (4.10), має вигляд

$$-\frac{0.5}{h^2} y_{k-1}^{j+1} + \left(\frac{1}{\tau} + \frac{1}{h^2} \right) y_k^{j+1} - \frac{0.5}{h^2} y_{k+1}^{j+1} = \frac{0.5}{h^2} y_{k-1}^j + \left(\frac{1}{\tau} - \frac{1}{h^2} \right) y_k^j + \frac{0.5}{h^2} y_{k+1}^j + \varphi, \quad (5.10)$$

$$k = 2, \dots, N-1.$$

Для неявних схем зі значенням $0 < \sigma \leq 1$, визначення y_k^{j+1} на новому шарі передбачає розв'язання СЛАР, яка утворюється відповідно рівняннями (5.8) | (5.9) | (5.10) і умовами (5.5). В програмній реалізації зазвичай рівняння (5.8) | (5.9) | (5.10) множимо на τ . Ці СЛАР ефективно розв'язуємо *монотонною прогонкою*, яка є стійка, враховуючи умови на вхідні дані і діагональну перевагу коефіцієнтів СЛАР.

п.2 Похибка апроксимації

Позначимо $z = y - u(x_k, t_j)$ – *похибка розв'язку* і підставимо $y = z + u(x_k, t_j)$ в РЗ (5.4)-(5.6), отримаємо для z різницеву задачу

$$\begin{cases} z_{t,j} = \Lambda Z + \psi, & k = \overline{2, N-1}, j = \overline{0, M-1}; \\ z_1^j = 0, \quad z_N^j = 0, & j = \overline{1, M}; \\ z_k^0 = 0, & k = \overline{1, N}, \end{cases} \quad (5.11)$$

де ψ – *похибка на розв'язку задачі* (5.1)-(5.3) має вигляд

$$\psi = \Lambda u^{(\sigma)} + \varphi - u_{t,j}. \quad (5.12)$$

Знайдемо її величину в околі точки $(x_k, \bar{t}_j)$. Враховуючи, що :

$$\begin{aligned} \hat{u} &= \bar{u} + \frac{\tau}{2} \bar{u} + \frac{\tau^2}{8} \bar{u} + \frac{\tau^3}{48} \bar{u} + O(\tau^4), \quad u = \bar{u} - \frac{\tau}{2} \bar{u} + \frac{\tau^2}{8} \bar{u} - \frac{\tau^3}{48} \bar{u} + O(\tau^4), \\ u^{(\sigma)} &= 0.5(\hat{u} + u) + \tau(\sigma - 0.5)u_{t,j}, \quad 0.5(\hat{u} + u) = \bar{u} + O(\tau^2), \quad u_{t,j} = \bar{u} + O(\tau^2), \\ \Lambda u &= Lu + \frac{h^2}{12} L^2 u + O(h^4), \end{aligned}$$

отримаємо наступний ланцюжок перетворень для ψ :

$$\begin{aligned} \psi &= L\bar{u} + \frac{h^2}{12} L^2 \bar{u} + \tau(\sigma - 0.5)L\bar{u} + \varphi - \bar{u} + O(h^4 + \tau^2) = \\ &= (L\bar{u} - \bar{u} + \bar{f}) + (\varphi - \bar{f}) + \frac{h^2}{12} (L\bar{u} - L\bar{f}) + \tau(\sigma - 0.5)L\bar{u} + O(h^4 + \tau^2) = \\ &= (\varphi - \bar{f}) + \frac{h^2}{12} (L\bar{u} - L\bar{f}) + \tau(\sigma - 0.5)L\bar{u} + O(h^4 + \tau^2) = \end{aligned}$$

$$= \left(\frac{h^2}{12} + \tau(\sigma - 0.5) \right) L\bar{u} + \left(\varphi - \bar{f} - \frac{h^2}{12} L\bar{f} \right) + O(h^4 + \tau^2)$$

і аналізуючи отримане співвідношення, маємо такі оцінки для порядку похибки

$$\psi = \begin{cases} O(h^2 + \tau), & \varphi = \bar{f} \text{ \& } \sigma \neq 0.5 \text{ \& } 0 \leq \sigma \leq 1; \\ O(h^2 + \tau^2), & \varphi = \bar{f} \text{ \& } \sigma = 0.5; \\ O(h^4 + \tau^2), & \varphi = \bar{f} + \frac{h^2}{12} L\bar{f} \text{ \& } \sigma = \sigma_* = 0.5 - \frac{h^2}{12\tau}. \end{cases} \quad (5.13)$$

Схема з $\sigma = \sigma_*$ називається *схемою підвищеного порядку точності*.

За наявності у крайових умовах (5.2) похідних, для отримання необхідного порядку апроксимації можна використовувати як *метод законтурних точок* так і *метод підвищення порядку апроксимації на розв'язку*.

п.3 Стійкість і збіжність різницевих схем

Дослідження стійкості виконаємо для задачі (5.4)-(5.6) двома способами: принципом максимуму (у нормі $\|\cdot\|_C$) і методом відокремлення змінних (у нормі $\|\cdot\|_{L_2}$).

Застосуємо *принцип максимуму*. Порівнюючи формули однорідної схеми (5.8)

$$\text{і (6.14), отримаємо } \alpha_0 = \frac{1}{\tau} + \frac{2\sigma}{h^2}, \alpha_{-1} = \alpha_1 = -\frac{2\sigma}{h^2}, \beta_0 = \frac{1}{\tau} - \frac{2(1-\sigma)}{h^2}, \beta_{-1} = \beta_1 = \frac{1-\sigma}{h^2}.$$

Підставимо ці значення в (4.15) при $C=0$, отримаємо нерівність $\frac{1}{\tau} - \frac{2(1-\sigma)}{h^2} \geq \left| \frac{1}{\tau} - \frac{2(1-\sigma)}{h^2} \right|$, яка виконується завжди (безумовно) при $\sigma=1$. Для схем з

$\sigma \in [0,1)$ маємо умову $\tau \leq \frac{h^2}{2(1-\sigma)}$, а отже умовну стійкість по початковим даним.

Умова (4.16) при $\theta=1$ виконується для будь-яких співвідношень кроків h, τ і значень σ .

Тепер розглянемо *метод відокремлення змінних*. Якщо підставимо (4.18) в однорідну схему (5.8), то отримаємо вираз для множника зростання q -ї гармоніки при переході з шару на шар:

$$\rho_q = \frac{1 - (1-\sigma)4h^{-2}\tau \sin^2(0.5qh)}{1 + \sigma 4h^{-2}\tau \sin^2(0.5qh)}.$$

З останньої рівності видно, що $\rho_q < 1$ при $\sigma \geq 0$. Залишається перевірити виконання нерівності $-1 \leq \rho_q$. З отриманої формули для ρ_q , маємо ланцюжок нерівностей

$$\sigma \geq \frac{1}{2} - \frac{h^2}{4\tau \sin^2(0.5qh)} \geq \frac{1}{2} - \frac{h^2}{4\tau}. \quad (5.14)$$

Таким чином, для явної схеми ($\sigma=0$) маємо умовну стійкість, при виконанні нерівності $\tau \leq 0.5h^2$. Виконання нерівності (5.14) для значень $\sigma = \{1; 0.5; \sigma_*\}$ легко перевіряється, а отже маємо *безумовну* стійкість. Критерій стійкості по правій частині (4.16) при $\theta=1$ також виконується.

Різницева задача (5.4) – (5.5), за умов виконання стійкості, відповідної гладкості вхідних даних і шуканої функції, збігається до розв'язку диференціальної задачі (5.1)-(5.3) у нормах $\|\cdot\|_C, \|\cdot\|_{L_2}$ із порядком, що співпадає із порядком апроксимації (5.13).

5.2 Різницеві схеми для циліндрично- і сферично-симметричних задач теплопровідності

При вивченні процесів теплопровідності або дифузії в тілах, які мають форму циліндра, природно користуватися циліндричною системою координат (r, φ, z) . Якщо температура не залежить від φ, z (циліндрична симетрія), то в області $\bar{D} = \{0 < r < R, 0 < t \leq T\}$ рівняння теплопровідності буде записано у вигляді

$$\frac{\partial u(r, t)}{\partial t} = Lu(r, t) + f(r, t), \quad (5.15)$$

де

$$Lu(r, t) = \frac{1}{r} \frac{\partial}{\partial r} \left(K(r, t) r \frac{\partial u(r, t)}{\partial r} \right). \quad (5.16)$$

У випадку сферичної симетрії (координати (r, θ, φ) , незалежність від θ, φ) рівняння теплопровідності буде записано у вигляді (5.15) з оператором

$$Lu(r, t) = \frac{1}{r^2} \frac{\partial}{\partial r} \left(K(r, t) r^2 \frac{\partial u(r, t)}{\partial r} \right). \quad (5.17)$$

Крайові умови в точці $r = R$ можуть бути першого або третього роду, наприклад,

$$u(R, t) = \mu_2(t), \quad 0 \leq t \leq T, \quad (5.18)$$

а при $r = 0$ ставлять умову обмеженості розв'язку $|u(0, t)| < \infty$:

$$\text{для (5.16)} \quad \lim_{r \rightarrow 0} \left(K(r, t) r \frac{\partial u(r, t)}{\partial r} \right) = 0, \quad 0 \leq t \leq T, \quad (5.19)$$

$$\text{для (5.17)} \quad \lim_{r \rightarrow 0} \left(K(r, t) r^2 \frac{\partial u(r, t)}{\partial r} \right) = 0, \quad 0 \leq t \leq T. \quad (5.20)$$

Початкові умови при $t = 0$ записуються звичним чином

$$u(r, 0) = u_0(r), \quad r \in [0, R]. \quad (5.21)$$

Спочатку побудуємо різницеву задачу для варіанта циліндричної симетрії (5.15), (5.16), (5.18), (5.19), (5.21).

Уведемо рівномірні сітки

$$\begin{aligned} \bar{\omega}_h &= \{r_i = (i-1)h, \quad i = \overline{1, N}, \quad R = (N-1)h\}; \\ \bar{\omega}_\tau &= \{t_j = j\tau, \quad j = \overline{0, M}, \quad T = M\tau\}, \end{aligned}$$

і замінимо область $\bar{D}$ сітковою областю $\bar{\omega}_{h\tau} = \bar{\omega}_h \times \bar{\omega}_\tau$. Використовуючи інтегро-інтерполяційний метод, отримаємо наступну різницеву задачу

$$\begin{aligned} \frac{\hat{y}_i - y_i}{\tau} &= \frac{1}{r_k} \left(K(r_{i-0.5}, \bar{t}_j) r_{i-0.5} Y_{\bar{r}, i} \right)_{r, i} + f(r_i, \bar{t}_j), \quad i = \overline{2, N-1}, \\ \frac{\hat{y}_1 - y_1}{\tau} &= \frac{4}{h} K(0.5h, \bar{t}_j) Y_{r, 1} + f(0, \bar{t}_j), \quad \hat{y}_N = \mu_2(t_{j+1}), \quad j = \overline{0, M-1}; \\ y_i^0 &= u_0(r_i), \quad i = \overline{1, N}. \end{aligned} \quad (5.22)$$

Якщо $\sigma \neq 0$, то для визначення $\hat{y}_i$ на шарі $j+1$ використовуємо метод прогонки.

Позначимо

$$\Lambda(\bar{t})y = \begin{cases} \frac{4}{h} K(0.5h, \bar{t}) y_r, & r = 0, \\ \frac{1}{r} \left(\bar{r} K(\bar{r}, \bar{t}) y_{\bar{r}} \right)_r, & r \in \omega_h; \end{cases} \quad \bar{r} = r_i - 0.5h,$$

тоді для похибки $z = y - (u)_h$ отримаємо задачу

$$z_t = \Lambda(\bar{t})z^{(\sigma)} + \psi, \quad r \in [0, R), \quad t_j \geq 0, \quad z_N = 0, \quad z^0 = 0,$$

де

$$\psi = \Lambda(\bar{t})u^{(\sigma)} + \varphi - u_t = \frac{1}{r}(\bar{r}\eta)_r + \psi^* + \psi^{**}, \quad \eta = K(\bar{r}, \bar{t})u_{\bar{r}}^{(\sigma)} - (Ku')_{r=\bar{r}, t=\bar{t}},$$

$$\eta = O(h^2 + \tau^{m_\sigma}), \quad \psi^* = O\left(\frac{h^2}{r}\right), \quad \psi^{**} = O(h^2 + \tau^2), \quad m_\sigma = \begin{cases} 2, \sigma = 0.5, \\ 1, \sigma \in [0, 1] \setminus 0.5. \end{cases}$$

В подальшому для оцінки з застосовується *метод стаціонарних неоднорідностей* [33] і відповідні апріорні оцінки, що дає збіжність РЗ (5.22) до крайової задачі (5.15), (5.16), (5.18), (5.19), (5.21) з порядком $O(h^2 + \tau^{m_\sigma})$. Для чисто неявної схеми можна використовувати принцип максимуму і довести, що схема збігається зі швидкістю $O(h^2 + \tau)$.

Аналогічним чином отримуємо РС для крайової задачі у випадку сферичної симетрії і запишемо її в операторній безіндексній формі:

$$\begin{aligned} (y_i)_t &= \Lambda(\bar{t})Y + f(r, \bar{t}), \quad r \in \bar{\omega}_h \setminus r_N, \\ y_N &= \mu_2(t), \quad t \in \bar{\omega}_\tau \setminus t_M; \\ y^0 &= u_0(r), \quad r \in \bar{\omega}_h. \end{aligned} \tag{5.23}$$

де різницевий оператор $\Lambda(\bar{t})y$ визначається наступним чином

$$\Lambda(\bar{t})y = \begin{cases} \frac{6}{h} K(0.5h, \bar{t})y_r, & r = 0, \\ \frac{1}{r^2} (\bar{r}^2 K(\bar{r}, \bar{t})y_{\bar{r}})_r, & r \in \omega_h. \end{cases}$$

Для знаходження сіткової функції $\hat{y}_i$ РЗ (5.23) на шарі $j+1$ при $\sigma \neq 0$ маємо СЛАР, яку розв'язуємо методом прогонки.

Дослідження нев'язки і оцінка точності РС (5.23) проводиться аналогічно випадку циліндричної симетрії.

Для задачі (5.15) з операторами (5.16), (5.17) часто використовують РС побудовані на «потоківій сітці». В цьому випадку сітка $\bar{\omega}_h$ визначається наступним чином:

$$\bar{\omega}_h = \{r_1 = 0, \bar{r}_2 = 0.5h, \dots, \bar{r}_{N-1} = (N-1.5)h, \bar{r}_N = (N-0.5)h = R\},$$

а сама побудова РС здійснюється ПМ [33].

ЛЕКЦІЯ № 09

ТЕМА № 5. Різницький метод розв'язання крайових задач для рівнянь параболічного типу.

5.3. Економічні різницькі схеми. Економічні РС для двовимірних рівнянь.

п.1. Передумови для написання економічних РС

Розглянемо в області $\bar{D} = D \cup \partial D = \bar{G} \times [0, T] = \{x_\alpha \in [0, 1], \alpha = \overline{1, p}\} \times [0, T]$ мішану задачу Коші для p -вимірного ($p \geq 2$) лінійного рівняння параболічного типу :

$$\frac{\partial u(x, t)}{\partial t} = Lu(x, t), \quad (x, t) \in G \times (0, T], \quad x \equiv (x_1, x_2, \dots, x_p), \quad (5.24)$$
$$L = \sum_{\alpha}^p L_{\alpha}; \quad L_{\alpha} u = \frac{\partial^2 u}{\partial x_{\alpha}^2},$$

$$u(x, t) = \mu(x, t), \quad x \in \partial G, \quad (5.25)$$

$$u(x, 0) = u_0(x), \quad x \in \bar{G}. \quad (5.26)$$

Будемо вважати, що задача поставлена коректно і розв'язок існує, єдиний, має відповідну достатню гладкість.

Для побудови різницевої задачі, яку поставимо у відповідність задачі (5.24)-(5.26), введемо рівномірну сітку

$$\bar{\omega}_{h\tau} = \bar{\omega}_h \times \bar{\omega}_\tau, \quad \bar{\omega}_h = \{i_1 h_1, i_2 h_2, \dots, i_p h_p\}, \quad i_\alpha = \overline{0, N_\alpha}, \quad h_\alpha = 1/N_\alpha\}$$
$$\bar{\omega}_\tau = \{j\tau, \quad j = \overline{0, M}, \quad \tau = T/M\}$$

і замінимо область $\bar{D}$ сітковою областю $\bar{\omega}_{h\tau}$, а також будемо використовувати наступні позначення

$$0 \leq \sigma \leq 1; \quad y^{(\sigma)} \equiv Y = \sigma \hat{y} + (1 - \sigma)y.$$

Тоді, аналогічно одновимірному рівнянню, використаємо двошарову схему – *схему з вагою*

$$y_{t,j} = \Lambda Y,$$
$$\Lambda = \sum_{\alpha=1}^p \Lambda_{\alpha}, \quad \Lambda_{\alpha} y = y_{x_{\alpha} x_{\alpha}} = \frac{y_{k_{\alpha}-1} - 2y_{k_{\alpha}} + y_{k_{\alpha}+1}}{h_{\alpha}}, \quad (5.27)$$
$$k_{\alpha} = \overline{1, N_{\alpha} - 1}; \quad \alpha = \overline{1, p}; \quad j = \overline{0, M - 1}.$$

Крайові і початкові умови апроксимуються точно

$$y|_{\partial \bar{\omega}_h} = \mu(x, t)|_{\partial \bar{\omega}_h}, \quad t \in \bar{\omega}_\tau; \quad (5.28)$$

$$y^0 = u_0(x), \quad x \in \bar{\omega}_h. \quad (5.29)$$

Від вибору параметра σ , як ми знаємо, залежить точність і стійкість схеми (5.27)-(5.29). Аналогічно одновимірним задачам, методом відокремлення змінних можна отримати наступну оцінку

$$\tau \leq \frac{1}{2(1-\sigma)} \cdot \left(\sum_{\alpha=1}^p \frac{1}{h_{\alpha}^2} \right)^{-1}. \quad (5.30)$$

Якщо $h_{\alpha} = h$, $\alpha = \overline{1, p}$ і використовувати значення параметра $\sigma = 0$, то обмеження на кроки буде наступне

$$\tau \leq \frac{h^2}{2p}.$$

З останньої нерівності видно, що зі збільшенням p потрібно сильно зменшувати τ для "алгебраїчного розрахунку" при переході з j -го шару на $j+1$ шар. З оцінки (5.30) також слідує, що *чисто неявна схема* ($\sigma=1$) *безумовно стійка*, але вимагає дуже великої кількості операцій на один вузол сітки при розв'язанні відповідної СЛАР.

Наприклад, використання матричної прогонки, буде потребувати операцій $\approx O(N^2)$, де N - кількість вузлів сітки.

Виникає питання: чи можна для багатовимірних задач побудувати схему, яка поєднує найкращі якості явної і неявної схем?

Озн.5.1. Різницеву схему для p -вимірною ($p \geq 2$) рівняння (5.24) будемо називати *економічною*, якщо:

- 1) вона безумовно стійка;
- 2) число арифметичних операцій на один вузол сітки $\approx O(N)$ (пропорційне кількості вузлів сітки).

Розглянемо найбільш ефективні економічні РС:

 *поздовжно-поперечна схема* (ППС);

 *локально-одновимірні схеми* (ЛОС).

п.2. Поздовжно-поперечна схема

ППС ще називають *схемою змінних напрямків*, або *схемою Писмена-Рекфорда* (по імені авторів, які вперше запропонували її).

Розглянемо неоднорідне рівняння (5.24) при розмірності $p=2$. Основна алгоритмічна ідея ППС полягає в тому, що перехід з j -го шару на $j+1$ шар виконується в два етапи, використовуючи додатковий проміжний $j+\frac{1}{2}$ шар, який відповідає значенню $\bar{t}_j = t_j + 0.5\tau$ і при цьому тільки певна частина просторового оператора записується на цих шарах:

$$\frac{y^{j+0.5} - y^j}{0.5\tau} = \Lambda_1 y^{j+0.5} + \Lambda_2 y^j + \varphi, \quad (5.31)$$

$$\frac{y^{j+1} - y^{j+0.5}}{0.5\tau} = \Lambda_1 y^{j+0.5} + \Lambda_2 y^{j+1} + \varphi. \quad (5.32)$$

Ці рівняння записуються у всіх внутрішніх вузлах просторової сітки ($\varphi = f(x, \bar{t})$).

Звичайно, до рівнянь (5.31), (5.32) необхідно додати крайові умови в граничних вузлах

$$y^{j+0.5} = \tilde{\mu}^j \equiv \frac{1}{2}(\mu^j + \mu^{j+1}) + \frac{\tau}{4}\Lambda_2(\mu^j - \mu^{j+1}), \quad (5.33)$$

$$y^{j+1} = \mu^{j+1}, \quad (5.34)$$

і початкові умови

$$y^0 = u_0(x), \quad x \in \bar{\omega}_h. \quad (5.35)$$

Щоб була зрозуміла умова (5.33), знайдемо вираз $y^{j+0.5}$ через значення сіткових функцій y^j , y^{j+1} . Для цього віднімемо (5.32) від (5.31) і виконаємо ланцюжок перетворень

$$\begin{aligned} -\frac{y^{j+1} + y^j}{0.5\tau} + 2\frac{y^{j+0.5}}{0.5\tau} &= -\Lambda_2(y^{j+1} - y^j), \\ 2y^{j+0.5} &= (y^{j+1} + y^j) - 0.5\tau\Lambda_2(y^{j+1} - y^j), \\ y^{j+0.5} &= \frac{1}{2}(y^j + y^{j+1}) + \frac{\tau}{4}\Lambda_2(y^j - y^{j+1}). \end{aligned} \quad (5.36)$$

Враховуючи, що вираз (5.36) повинен виконуватись і на границі сітки, маємо умову (5.33).

Знаходження різницевого розв'язку. Схема (5.31) неявна по напрямку x_1 і явна по x_2 , а схема (5.32) – навпаки, явна по напрямку x_1 і неявна по x_2 . Запишемо більш детально алгоритм ППС (в вигляді придатному для програмування).

1-й етап ППС – перехід на додатковий проміжний $j+\frac{1}{2}$ шар:

$$\begin{aligned}
k_2 = 0: & \quad \bar{y}_{k_1, k_2} = \tilde{\mu}_{k_1, k_2}^j, k_1 = \overline{0, N_1}; \\
k_2 = \overline{1, N_2 - 1}: & \quad \begin{cases} \bar{y}_{0, k_2} = \tilde{\mu}_{0, k_2}^j, \\ \gamma_1 \bar{y}_{k_1-1, k_2} - (1 + 2\gamma_1) \bar{y}_{k_1, k_2} + \gamma_1 \bar{y}_{k_1+1, k_2} = -\bar{F}_{k_1, k_2}, k_1 = \overline{1, N_1 - 1}, \\ \bar{y}_{N_1, k_2} = \tilde{\mu}_{N_1, k_2}^j; \end{cases} \\
k_2 = N_2: & \quad \bar{y}_{k_1, k_2} = \tilde{\mu}_{k_1, k_2}^j, k_1 = \overline{0, N_1}.
\end{aligned} \tag{5.37}$$

Тут введено наступні позначення

$$\begin{aligned}
y_{k_1, k_2} &= y^j, \quad \bar{y}_{k_1, k_2} = y^{j+0.5}, \quad \hat{y}_{k_1, k_2} = y^{j+1}; \quad \gamma_1 = 0.5\tau h_1^{-2}, \quad \gamma_2 = 0.5\tau h_2^{-2}; \\
\bar{F}_{k_1, k_2} &= \gamma_2 y_{k_1, k_2-1} + (1 - 2\gamma_2) y_{k_1, k_2} + \gamma_2 y_{k_1, k_2+1} + 0.5\tau\varphi; \quad \varphi = f_{k_1, k_2}^{j+0.5}.
\end{aligned}$$

2-й етап ППС – перехід на $j+1$ шар:

$$\begin{aligned}
k_1 = 0: & \quad \hat{y}_{k_1, k_2} = \mu_{k_1, k_2}^{j+1}, k_2 = \overline{0, N_2}; \\
k_1 = \overline{1, N_1 - 1}: & \quad \begin{cases} \hat{y}_{k_1, 0} = \mu_{k_1, 0}^{j+1}, \\ \gamma_2 \hat{y}_{k_1, k_2-1} - (1 + 2\gamma_2) \hat{y}_{k_1, k_2} + \gamma_2 \hat{y}_{k_1, k_2+1} = -\hat{F}_{k_1, k_2}, k_2 = \overline{1, N_2 - 1}, \\ \hat{y}_{k_1, N_2} = \mu_{k_1, N_2}^{j+1}; \end{cases} \\
k_1 = N_1: & \quad \hat{y}_{k_1, k_2} = \mu_{k_1, k_2}^{j+1}, k_2 = \overline{0, N_2},
\end{aligned} \tag{5.38}$$

де $\hat{F}_{k_1, k_2} = \gamma_1 y_{k_1-1, k_2} + (1 - 2\gamma_1) y_{k_1, k_2} + \gamma_1 y_{k_1+1, k_2} + 0.5\tau\varphi$.

Як бачимо, на кожному етапі ППС, є СЛАР: в (5.37) при фіксованому індексу $k_2 = \overline{1, N_2 - 1}$, а в (5.38) – при фіксованому індексу $k_1 = \overline{1, N_1 - 1}$. Матриця кожної з цих СЛАР – тридіагональна з діагональною перевагою, а отже *монотонна ліва|права прогонка* стійка, а різницевий розв’язок існує і єдиний.

Так як реалізація прогонки вимагає на один вузол число арифметичних дій, яке не залежить від просторових кроків сітки, то п.2) Озн.5.1 економічності схеми має місце.

Стійкість ППС. Використаємо для дослідження стійкості *метод відокремлення змінних*, але при цьому враховуємо, що множники росту гармонік для першого і другого етапів ППС можуть бути різними, а отже

$$\begin{aligned}
y_{k_1, k_2}^j &= \rho_{q_1, q_2} \exp(iq_1 x_{k_1} + iq_2 x_{k_2}), \quad q_\alpha = 0, \pm 1, \pm 2, \dots; \quad \alpha = 1, 2; \quad i = \sqrt{-1}; \\
\bar{y}_{k_1, k_2} &= \bar{\rho}_{q_1, q_2} y_{k_1, k_2}^j, \quad \hat{y}_{k_1, k_2} = \hat{\rho}_{q_1, q_2} y_{k_1, k_2}^j.
\end{aligned} \tag{5.39}$$

Підставимо співвідношення (5.39) в однорідні рівняння (5.31)-(5.32) і отримаємо такі значення множників росту

$$\bar{\rho}_{q_1, q_2} = \frac{1 - 2\tau h_2^{-2} \sin^2(0.5q_2 h_2)}{1 + 2\tau h_1^{-2} \sin^2(0.5q_1 h_1)}, \quad \hat{\rho}_{q_1, q_2} = \frac{1 - 2\tau h_1^{-2} \sin^2(0.5q_1 h_1)}{1 + 2\tau h_2^{-2} \sin^2(0.5q_2 h_2)}.$$

Як бачимо, що для всіх гармонік при довільних кроках сітки, виконується нерівність

$$|\bar{\rho}_{q_1, q_2} \hat{\rho}_{q_1, q_2}| \leq 1,$$

а отже схема (5.31)-(5.32) *рівномірно і безумовно стійка по початковим даним*.

Нескладно перевірити, що додаткова умова *стійкості за правими частинами* виконується на кожному півкроці по часу, а отже ППС стійка і п.1) Озн.5.1 економічності схеми також має місце.

Апроксимація. Для дослідження порядку похибки апроксимації приведемо схему (5.31)-(5.32) до еквівалентного, але записаного тільки на "цілих шарах" виду. Спочатку додамо рівняння (5.31), (5.32)

$$\frac{\hat{y} - y}{0.5\tau} = 2\Lambda_1 \bar{y} + \Lambda_2 (\hat{y} + y) + 2\varphi,$$

для виключення з нього $\bar{y}$ використаємо (5.36)

$$\frac{\hat{y} - y}{\tau} = \Lambda_1 \left(\frac{1}{2} (\hat{y} + y) - \frac{\tau}{4} \Lambda_2 (\hat{y} - y) \right) + 0.5\Lambda_2 (\hat{y} + y) + \varphi$$

і зробимо нескладні перетворення:

$$\begin{aligned} \frac{\hat{y} - y}{\tau} &= 0.5\Lambda_1 (\hat{y} + y) - \frac{\tau^2}{4} \Lambda_1 \Lambda_2 \frac{\hat{y} - y}{\tau} + 0.5\Lambda_2 (\hat{y} + y) + \varphi = \\ &= 0.5\Lambda (\hat{y} + y) + \varphi - \underbrace{\frac{\tau^2}{4} \Lambda_1 \Lambda_2 \frac{\hat{y} - y}{\tau}}_{O(\tau^2)}. \end{aligned} \quad (5.40)$$

Якщо співвідношення (5.40) порівняти з симетричною ($\sigma = \frac{1}{2}$) РС для рівняння (5.24)

$$\frac{\hat{y} - y}{\tau} = 0.5\Lambda (\hat{y} + y) + \varphi, \quad (5.41)$$

порядок якої $O(\tau^2 + h_1^2 + h_2^2)$, то бачимо, що (5.40) різниться з (5.41) елементом порядку $O(\tau^2)$, а отже порядок апроксимації ППС $O(\tau^2 + h_1^2 + h_2^2)$.

Використовуючи співвідношення (5.40) покажемо, що алгоритм ППС (5.37), (5.38) знаходження $\hat{y}$ еквівалентний розв'язанню СЛАР для $\hat{y}$ з деякою факторизованою матрицею. Враховуючи рівність $\hat{y} = y + \tau * y_{t,j}$, співвідношення (5.40) можна подати так

$$\begin{aligned} y_{t,j} &= 0.5\Lambda (2y + \tau * y_{t,j}) + \varphi - 0.25\tau^2 \Lambda_1 \Lambda_2 y_{t,j}, \\ y_{t,j} - 0.5\tau (\Lambda_1 + \Lambda_2) y_{t,j} + 0.25\tau^2 \Lambda_1 \Lambda_2 y_{t,j} &= \Lambda y + \varphi, \\ (E - 0.5\tau \Lambda_1)(E - 0.5\tau \Lambda_2) y_{t,j} &= \Lambda y + \varphi \end{aligned}$$

і остаточно:

$$(E - 0.5\tau \Lambda_1)(E - 0.5\tau \Lambda_2) \hat{y} = (E - 0.5\tau \Lambda_1)(E - 0.5\tau \Lambda_2) y + \tau * (\Lambda y + \varphi).$$

Збіжність. Враховуючи дослідження апроксимації, стійкості ППС і лінійність задачі (5.24)-(5.26), доходимо до висновку, що розв'язок РС (5.31)-(5.35) збігається у нормі $\|\cdot\|_{L_2}$ до розв'язку диференціальної задачі, з тим же порядком, що і порядок апроксимації, а отже з порядком $O(\tau^2 + h_1^2 + h_2^2)$.

Нажаль алгоритм ППС не переноситься на розмірності $p > 2$ з повним успадкуванням властивостей економічності, а саме ППС при $p > 2$ стає умовно стійкою, крім цього, порядок апроксимації зменшується до $O(\tau + h_1^2 + h_2^2 + \dots + h_p^2)$.

ЛЕКЦІЯ № 10

ТЕМА № 5. Різницевий метод розв'язання крайових задач для рівнянь параболічного типу.

5.4. Лінійні крайові задачі для багатовимірних рівнянь.

Продовжимо знайомитись з економічними різницевиими схемами, які почали розглядати в попередній лекції на прикладі ППС для крайової задачі ($p = 2$):

$$\frac{\partial u(x, t)}{\partial t} = Lu(x, t), \quad (x, t) \in G \times (0, T], \quad x \equiv (x_1, x_2, \dots, x_p), \quad (5.42)$$

$$L = \sum_{\alpha} L_{\alpha}; \quad L_{\alpha} u = \frac{\partial^2 u}{\partial x_{\alpha}^2},$$

$$u(x, t) = \mu(x, t), \quad x \in \partial G, \quad (5.43)$$

$$u(x, 0) = u_0(x), \quad x \in \bar{G}. \quad (5.44)$$

Розглянемо загальний підхід отримання економічних схем, який можна успішно використовувати для нестационарних рівнянь зі змінними коефіцієнтами

$$L_{\alpha} u = \frac{\partial}{\partial x_{\alpha}} \left(K(x, t) \frac{\partial u}{\partial x_{\alpha}} \right); \quad \text{з розривними коефіцієнтами } K(x, t); \quad \text{квазілінійних рівнянь, де}$$

$K(x, t, u)$; що означені в довільних p -вимірних ($p \geq 2$) областях.

п.1. Поняття сумарної апроксимації, адитивності РС

При дослідженні апроксимації ППС ми виключили із системи рівнянь ППС сіткову функцію на проміжному $j + \frac{1}{2}$ -му шарі і отримали еквівалентну схему. Але така процедура можлива тільки для прямокутної просторової області і при цьому граничні умови на $j + \frac{1}{2}$ шарі визначаються спеціальним чином (див. задання в ППС сіткової функції $\tilde{u}^j$ на границі).

Дослідимо апроксимацію ППС "напряму" – згідно означень. Представимо розв'язок ППС в виді

$$y_{k_1, k_2}^j = u(x_1^{(k_1)}, x_2^{(k_2)}, t_j) + z_{k_1, k_2}^j, \quad y_{k_1, k_2}^{j+0.5} = u(x_1^{(k_1)}, x_2^{(k_2)}, t_j + 0.5\tau) + z_{k_1, k_2}^{j+0.5},$$

де проекцію розв'язку вихідної диференціальної задачі на сітку позначимо

$$(u)^j \equiv (u)_{k_1, k_2}^j = u(x_1^{(k_1)}, x_2^{(k_2)}, t_j), \quad (u)^{j+0.5} \equiv (u)_{k_1, k_2}^{j+0.5} = u(x_1^{(k_1)}, x_2^{(k_2)}, t_j + 0.5\tau),$$

а функції $z_{k_1, k_2}^j \equiv z^j$, $z_{k_1, k_2}^{j+0.5} \equiv z^{j+0.5}$ – похибка різницевого методу на розв'язку вихідної задачі. Підставимо ці вирази в рівняння ППС (5.31), (5.32), отримаємо рівняння, яким задовольняє похибка методу

$$\frac{z^{j+0.5} - z^j}{0.5\tau} = \Lambda_1 z^{j+0.5} + \Lambda_2 z^j + \psi_1^j, \quad (x_1^{(k_1)}, x_2^{(k_2)}) \in \omega_h,$$

$$\frac{z^{j+1} - z^{j+0.5}}{0.5\tau} = \Lambda_1 z^{j+0.5} + \Lambda_2 z^{j+1} + \psi_2^j, \quad (x_1^{(k_1)}, x_2^{(k_2)}) \in \omega_h,$$

де сіткові функції ψ_1^j, ψ_2^j – похибки апроксимації рівнянь ППС на розв'язку вихідної задачі, визначаються :

$$\psi_1^j = -\frac{(u)^{j+0.5} - (u)^j}{0.5\tau} + \Lambda_1 (u)^{j+0.5} + \Lambda_2 (u)^j,$$

$$\psi_2^j = -\frac{(u)^{j+1} - (u)^{j+0.5}}{0.5\tau} + \Lambda_1 (u)^{j+0.5} + \Lambda_2 (u)^{j+1}.$$

Згідно з методикою РС, розкладемо функції (по τ до $O(\tau^3)$), які входять в вирази ψ_1, ψ_2 , по формулі Тейлора в точці $(x_1^{(k_1)}, x_2^{(k_2)}, t_j)$ і використовуючи (5.42), отримаємо

$$\begin{aligned}\psi 1^j &= -\frac{\tau}{4} \left(\frac{\partial^2 u}{\partial t^2} \right)_{k_1, k_2}^j + \frac{\tau}{2} L_1 \left(\frac{\partial u}{\partial t} \right)_{k_1, k_2}^j + O(\tau^2 + h_1^2 + h_2^2), \\ \psi 2^j &= -\frac{3\tau}{4} \left(\frac{\partial^2 u}{\partial t^2} \right)_{k_1, k_2}^j + \frac{\tau}{2} L_1 \left(\frac{\partial u}{\partial t} \right)_{k_1, k_2}^j + \tau L_2 \left(\frac{\partial u}{\partial t} \right)_{k_1, k_2}^j + O(\tau^2 + h_1^2 + h_2^2).\end{aligned}$$

Таким чином, кожне з рівнянь ППС апроксимує вихідне диференціальне рівняння з першим порядком по τ і другим порядком по h_1, h_2 .

А тепер обчислимо сумарну похибку апроксимації $\psi^j = \psi 1^j + \psi 2^j$:

$$\begin{aligned}\psi^j &= -\tau \left(\frac{\partial^2 u}{\partial t^2} \right)_{k_1, k_2}^j + \tau (L_1 + L_1) \left(\frac{\partial u}{\partial t} \right)_{k_1, k_2}^j + O(\tau^2 + h_1^2 + h_2^2) = \\ &= \tau \left(\frac{\partial}{\partial t} \left[\underbrace{(L_1 + L_1)u - \frac{\partial u}{\partial t}}_{=0} \right] \right)_{k_1, k_2}^j + O(\tau^2 + h_1^2 + h_2^2) = \\ &= O(\tau^2 + h_1^2 + h_2^2)\end{aligned}$$

Бачимо, що вона співпала з отриманим раніше порядком апроксимації ППС, при цьому порядок по τ підвищився до значення два!

Отже, з огляду на класичне означення апроксимації РС на розв'язку диференціальної задачі (одне з фундаментальних в теорії РС!), в ППС маємо його ослаблення і заміну на сумарну апроксимацію адитивної (складеної) схеми.

Ця група схем для p -вимірних ($p \geq 2$) нестационарних рівнянь має дві основні характеристики:

- 1) перехід з j -го шару на $j+1$ шар виконується за допомогою певного ланцюжка p -проміжних звичайних (двошарових, тришарових і т.д.) одновимірних схем;
- 2) похибка апроксимації адитивної схеми визначається як сума нев'язок проміжних схем, при цьому проміжна схема ланцюга може не апроксимувати вихідне дифрівняння на шарі $j + \frac{m}{p}$, $m = \overline{1, p-1}$ (тобто, має порядок $O(1)$).

Приведемо приклад такої економічної схеми, побудованої локально-одновимірним методом. Для крайової задачі (5.42)-(5.44) запишемо РС

$$\frac{1}{2} \frac{y^{j+0.5} - y^j}{0.5\tau} = \Lambda_1 y^{j+0.5}, \quad (x_1^{(k_1)}, x_2^{(k_2)}) \in \omega_h, \quad (5.45)$$

$$\frac{1}{2} \frac{y^{j+1} - y^{j+0.5}}{0.5\tau} = \Lambda_2 y^{j+1}, \quad (x_1^{(k_1)}, x_2^{(k_2)}) \in \omega_h. \quad (5.46)$$

Аналогічно ППС, знайдемо сіткові функції $\psi 1^j, \psi 2^j$ – похибки апроксимації рівнянь (5.45), (5.46) на розв'язку (5.42):

$$\begin{aligned}\psi 1^j &= -\frac{(u)^{j+0.5} - (u)^j}{\tau} + \Lambda_1 (u)^{j+0.5} = -0.5 \left(\frac{\partial u}{\partial t} \right)_{k_1, k_2}^j + L_1 (u)_{k_1, k_2}^j + O(\tau + h_1^2 + h_2^2) = O(1), \\ \psi 2^j &= -\frac{(u)^{j+1} - (u)^{j+0.5}}{\tau} + \Lambda_2 (u)^{j+1} = -0.5 \left(\frac{\partial u}{\partial t} \right)_{k_1, k_2}^j + L_2 (u)_{k_1, k_2}^j + O(\tau + h_1^2 + h_2^2) = O(1),\end{aligned}$$

Для отримання цих співвідношень, розклад в ряд по τ зробили тільки до $O(\tau^2)$. Обчислення сумарної похибки апроксимації $\psi^j = \psi 1^j + \psi 2^j$ дає $\psi^j = O(\tau + h_1^2 + h_2^2)$!

Перейдемо тепер до загальної теорії побудови економічних РС для багатовимірних нестационарних рівнянь, використовуючи адитивні (складені) схеми з сумарною апроксимацією.

п.2. Побудова для КЗ параболічного типу адитивних РС з сумарною апроксимацією на основі локально-одновимірного методу

Нехай маємо p -вимірне ($p \geq 2$) лінійне рівняння

$$P u \equiv \frac{\partial u(x, t)}{\partial t} - L u - f(x, t) = 0, \quad (x, t) \in G \times (0, T], \quad x \equiv (x_1, x_2, \dots, x_p), \quad (5.47)$$

де $\bar{G} = G \cup \partial G$ – деяка p -вимірна область, на границі якої задані крайові умови першого роду (але це не обов'язково!) і при $t = 0$ початкові умови $u(x, 0) = u_0(x)$.

Побудова економічних адитивних схем з сумарною апроксимацією *локально-одновимірним методом* можлива при виконанні двох умов:

- 1) оператор L повинен мати представлення в вигляді суми операторів простої структури $L = L_1 + L_2 + \dots + L_p$;
- 2) функція f має бути представлена у вигляді $f = f_1 + f_2 + \dots + f_p$, де f_α , $\alpha = \overline{1, p}$ – довільні функції тієї-ж гладкості, що і $f(x, t)$ (умова нормування).

п.2.1. Варіант ЛОС (перший ланцюг) З урахуванням цих умов, поставимо рівняння $P u = 0$ у відповідність ланцюг «одновимірних» рівнянь (так званий *перший ланцюг*) [33], а саме:

$$\sum_{\alpha=1}^p P_\alpha u = 0, \quad P_\alpha u = \frac{1}{p} \frac{\partial u(x, t)}{\partial t} - L_\alpha u - f_\alpha(x, t),$$

На відрізку $0 \leq t \leq T$ введемо рівномірну сітку $\bar{\omega}_\tau = \{t_j = j\tau, j = 0, 1, \dots, M\}$ і кожен інтервал $(t_j, t_{j+1}]$ розіб'ємо на p частин точками $t_{j+\alpha/p} = t_j + \alpha \frac{\tau}{p}$, $\alpha = \overline{1, p}$. Введемо позначення $\Delta_\alpha = (t_{j+(\alpha-1)/p}, t_{j+\alpha/p}]$, тоді $(t_j, t_{j+1}] = \bigcup_{\alpha=1}^p \Delta_\alpha$, а також функції (неперервні, не сіткові!) $u(x, t) = v_{(\alpha)}(x, t)$, $t \in \Delta_\alpha$ див. рис. 5.1

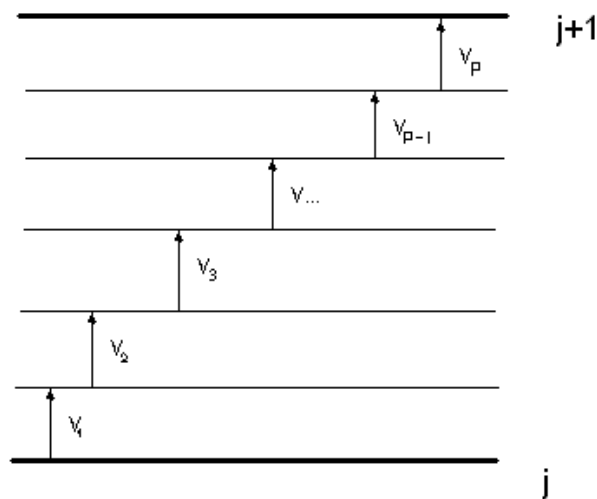


Рис.5.1. Схематичне зображення функцій $v_{(\alpha)}(x, t)$, $\alpha = \overline{1, p}$, $t \in (t_j, t_{j+1}]$ для першого ланцюга.

Будемо послідовно, починаючи з $\alpha = 1, 2, \dots$, розв'язувати рівняння

$$P_\alpha v_{(\alpha)} = 0, \quad x \in G, \quad t \in \Delta_\alpha, \quad \alpha = 1, 2, \dots, p, \quad (5.48)$$

додаючи при цьому початкові і граничні умови

$$\begin{aligned} v_{(1)}(x,0) &= u_0(x), \quad v_{(\alpha)}(x, t_{j+(\alpha-1)/p}) = v_{(\alpha-1)}(x, t_{j+(\alpha-1)/p}), \\ v_{(\alpha)}(x, t_{j+\alpha/p}) &= \mu(x, t_{j+\alpha/p}), \quad x \in \partial G, \quad \alpha = 1, 2, \dots, p. \end{aligned} \quad (5.49)$$

Розв'язком цієї задачі назвемо $u(x, t_j) = v_{(p)}(x, t_j)$, $j = 1, \dots, M$.

Апроксимуючи кожне з рівнянь (5.48) двошаровою схемою з вагою σ_α або чисто неявною схемою, отримуємо ланцюг p одновимірних схем

$$\Pi_\alpha y_{(\alpha)} = 0, \quad \alpha = 1, 2, \dots, p, \quad (5.50)$$

які і назвемо *локально-одновимірними схемами* (ЛОС). Деталізуємо (5.50) для схеми з вагою σ_α :

$$\left\{ \begin{aligned} \frac{y^{j+\alpha/p} - y^{j+(\alpha-1)/p}}{p^*(\tau/p)} &= \Lambda_\alpha (\sigma_\alpha y^{j+\alpha/p} + (1-\sigma_\alpha) y^{j+(\alpha-1)/p}) + f_\alpha(x, t_j + 0.5\tau), \\ y^{j+\alpha/p} &= \mu(x, t_j + \alpha/p), \quad x \in \partial G_{h,\alpha}, \quad \alpha = 1, 2, \dots, p; \\ j &= 0, 1, \dots, M-1; \\ y^0 &= u_0(x), \quad x \in \bar{G}_{h,\alpha}. \end{aligned} \right. \quad (5.50^*)$$

Тут триточковий різницевий оператор $\Lambda_\alpha \sim L_\alpha$, $\Lambda_\alpha y \equiv y_{\bar{x}_\alpha x_\alpha k_\alpha} = \frac{1}{h_\alpha^2} (y_{k_\alpha-1} - 2y_{k_\alpha} + y_{k_\alpha+1})$ і

залишені тільки ті індекси, які змінюються.

Схема (5.50) апроксимує кожне рівняння (5.48) в звичайному смислі

$$\Psi_\alpha = \Pi_\alpha v_{(\alpha)} - (P_\alpha v_{(\alpha)})_h = O(\tau + h_\alpha^2) \rightarrow 0, \quad \tau \rightarrow 0 \text{ \& } h_\alpha \rightarrow 0, \quad \alpha = 1, 2, \dots, p.$$

Нехай ψ_α – похибка на розв'язку рівняння $Pu = 0$ для одної схеми номера α :

$$\psi_\alpha = \Pi_\alpha v_{(\alpha)} - (P_\alpha v_{(\alpha)})_h = \Psi_\alpha,$$

і враховуючи, що $(P_\alpha v_{(\alpha)})_h = (P_\alpha u)_h^{j+0.5} + O(\tau)$, представимо ψ_α у вигляді

$$\psi_\alpha = (P_\alpha u)_h^{j+0.5} + O(\tau) + \Psi_\alpha = \psi_\alpha^0 + \psi_\alpha^*,$$

де введено позначення

$$\begin{aligned} \psi_\alpha^0 &= (P_\alpha u)_h^{j+0.5} = O(1); \\ \psi_\alpha^* &= O(\tau + h_\alpha^2), \quad \|\psi_\alpha^*\| \rightarrow 0, \quad \tau \rightarrow 0 \text{ \& } h_\alpha \rightarrow 0, \quad \alpha = 1, 2, \dots, p. \end{aligned}$$

Тепер знайдемо сумарну апроксимацію

$$\psi = \sum_{\alpha=1}^p \psi_\alpha = \sum_{\alpha=1}^p (P_\alpha u)_h^{j+0.5} + \sum_{\alpha=1}^p \psi_\alpha^* = \left(\sum_{\alpha=1}^p P_\alpha u \right)_h^{j+0.5} + \sum_{\alpha=1}^p \psi_\alpha^* = \sum_{\alpha=1}^p \psi_\alpha^*.$$

Таким чином, в регулярних вузлах сітки ЛОС має сумарну апроксимацію $O(\tau + |h|^2)$, де $|h|^2 = h_1^2 + h_1^2 + \dots + h_p^2$, якщо кожна з схем (5.50) апроксимує в звичайному смислі відповідне рівняння (5.48) [33].

п.2.2. Варіант ЛОС (другий ланцюг) Можна отримати і інший варіант локально-одновимірних схем. Для цього на відрізку $0 \leq t \leq T$ введемо рівномірну сітку $\bar{\omega}_\tau = \{t_j = j\tau, j = 0, 1, \dots, M\}$ і поставимо рівнянню $Pu = 0$ у відповідність ланцюг «одновимірних» рівнянь (так званий *другий ланцюг*) [33], а саме:

$$Pu = \sum_{\alpha=1}^p P_{\alpha} v_{(\alpha)} = 0, \quad (5.51)$$

$$P_{\alpha} v_{(\alpha)} = \frac{\partial v_{(\alpha)}(x, t)}{\partial t} - L_{\alpha} v_{(\alpha)} - f_{\alpha}(x, t), \quad x \in G, \quad t \in (t_j, t_{j+1}]$$

який доповнимо крайовими

$$v_{(\alpha)}(x, t) = \mu(x, t), \quad x \in \partial G, \quad \alpha = 1, 2, \dots, p; \quad (5.52)$$

і початковими умовами

$$v_{(1)}(x, t_j) = \begin{cases} v_{(p)}(x, t_{j+1}), & j = 1, 2, \dots, \\ u_0(x), & j = 0; \end{cases} \quad (5.53)$$

$$v_{(\alpha)}(x, t_j) = v_{(\alpha-1)}(x, t_{j+1}), \quad \alpha = 2, 3, \dots, p.$$

Розв'язком цієї задачі назвемо функцію $u(x, t_j) = v_{(p)}(x, t_j)$, $j = 1, \dots, M$. На відміну від варіанта першого ланцюга, кожне із рівнянь номера α розв'язується на всьому інтервалі $t \in (t_j, t_{j+1}]$, див. рис. 5.2

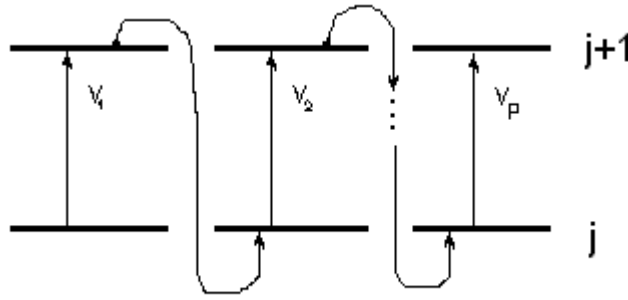


Рис.5.2. Схематичне зображення функцій $v_{(\alpha)}(x, t)$, $\alpha = \overline{1, p}$, $t \in (t_j, t_{j+1}]$ для другого ланцюга.

Зауваження. У випадку однорідних крайових умов I роду, $f \equiv 0$ & $(f_{\alpha} \equiv 0, \alpha = \overline{1, p})$ і переставних операторів L_p , розв'язок задачі (5.48)-(5.49) і задачі (5.51)-(5.52) – співпадають.

Апроксимуючи кожне з рівнянь (5.51) двошаровою схемою з вагою $0 < \sigma_{\alpha} \leq 1$ або чисто неявною схемою, отримуємо ланцюг з p одновимірних схем

$$\Pi_{\alpha} y_{(\alpha)} = 0, \quad \alpha = 1, 2, \dots, p, \quad (5.54)$$

які, як у попередньому випадку, називаємо ЛОС. Деталізуємо (5.54), наприклад, для чисто неявної схеми :

$$\left\{ \begin{array}{l} \frac{y_{(\alpha)}^{j+1} - y_{(\alpha)}^j}{\tau} = \Lambda_{\alpha} y_{(\alpha)}^{j+1} + f_{\alpha}(x, t_j + 0.5\tau), \\ y_{(\alpha)}^{j+1} = \mu(x, t_{j+1}), \quad x \in \partial G_{h, \alpha}, \quad \alpha = 1, 2, \dots, p; \\ y_{(\alpha)}^j = \begin{cases} u_0(x), & j = 0 \\ y_{(p)}^{j+1}, & j > 0, \alpha = 1, \\ y_{(\alpha-1)}^{j+1}, & \forall j, \alpha = 2, \dots, p, \end{cases} \quad x \in \partial \bar{G}_{h, \alpha}; \\ j = 0, 1, \dots, M-1. \end{array} \right. \quad (5.54^*)$$

Для (5.54) також розглянемо $\psi_{\alpha} = \Pi_{\alpha} u \equiv (P_{\alpha} v_{(\alpha)})_h + \Psi_{\alpha}$ – похибку на розв'язку рівняння $Pu = 0$ для одної схеми номера α . Так, як $(P_{\alpha} v_{(\alpha)})_h \equiv (P_{\alpha} u)_h = (P_{\alpha} u)_h^{j+0.5} + O(\tau)$, то

отримаємо $\psi_\alpha = \psi_\alpha^0 + \psi_\alpha^*$, де $\psi_\alpha^0 = (P_\alpha u)_h^{j+0.5} = O(1)$, $\psi_\alpha^* = O(\tau + h_\alpha^2)$ і, як наслідок, знайдемо сумарну апроксимацію

$$\psi = \sum_{\alpha=1}^P \psi_\alpha = \underbrace{\sum_{\alpha=1}^P \psi_\alpha^0}_{=0} + \sum_{\alpha=1}^P \psi_\alpha^* = O(\tau + |h|^2).$$

Знаходження різницевого розв'язку ЛОС виду (5.50*) або (5.54*) для кожного номера $\alpha \in$ розв'язання сукупності СЛАР (як і в ППС). Кожна окрема СЛАР має тридіагональну матрицю з діагональною перевагою, а отже *монотонна ліва|права прогонка* є стійка, а різницевий розв'язок існує і єдиний.

За допомогою *принципу максимуму* можна отримати апріорні оцінки для стійкості ЛОС у метриці C і рівномірну збіжність зі швидкістю $O(\tau + h^2)$, $h = \max_{1 \leq \alpha \leq p} (h_\alpha)$ [33].

ЛОС без особливих зусиль поширюється на крайові задачі :

1) задані в p -вимірних областях, для яких:

- i) перетин області з довільною прямою, паралельною осі координат Ox_α , може складатись з скінченної кількості інтервалів;
- ii) можливість побудови в області $\bar{G}$ зв'язної сітки $\bar{\omega}_h$.

2) з крайовими умовами другого, третього роду.

3) для лінійних рівнянь параболічного типу зі змінними коефіцієнтами, де

$$L_\alpha u = \frac{\partial}{\partial x_\alpha} \left(K_\alpha(x, t) \frac{\partial u}{\partial x_\alpha} \right).$$

4) для квазілінійних рівнянь параболічного типу, де $L_\alpha u = \frac{\partial}{\partial x_\alpha} \left(K_\alpha(x, t, u) \frac{\partial u}{\partial x_\alpha} \right)$ і $f(x, t, u)$.

Додатково зауважимо, що програмна реалізація як першого ланцюга ЛОС, так і другого, проводиться досить економічно по відношенню до оперативної пам'яті (ОЗП) комп'ютера і до кількості арифметико-логічних операцій. Зокрема, все, що стосується зберігання $y^{j+\alpha/p} | y_{(\alpha)}^j$, $\alpha = \overline{1, p}$, $j = \overline{0, M}$ в ОЗП, то зберігання можна реалізувати тільки в одному масиві $Y(k_1, k_2, \dots, k_p)$, $k_\alpha = \overline{1, N_\alpha}$, $\alpha = \overline{1, p}$, а метод прогонки розв'язання СЛАР – економічний по усім характеристикам. На один вузол сітки використовується $O(1)$ операцій. Якщо є необхідність використання отриманого числового розв'язку на багатьох шарах t_j , наприклад, анімаційна візуалізація, то тут можна скористатись записом потрібної інформації в файл(и) на зовнішні носії.

ЛЕКЦІЯ № 11

ТЕМА № 6. Різницеві схеми для рівняння Пуассона з крайовими умовами Діріхле.

Розглянемо задачу Діріхле для рівняння Пуассона [33]: знайти неперервну в прямокутній області $\bar{D} = D \cup \partial D = \{(x_1, x_2) \in [0, l_1] \times [0, l_2]\}$ функцію $u(x_1, x_2)$, яка справджує рівняння

$$\Delta u(x_1, x_2) = -f(x_1, x_2), \quad x = (x_1, x_2) \in D, \quad (6.1)$$

і крайові умови

$$u(x) = \mu(x), \quad x \in \partial D. \quad (6.2)$$

Тут введено позначення

$$\Delta u = L_1 u + L_2 u, \quad L_\alpha u = \frac{\partial^2 u}{\partial x_\alpha^2}, \quad \alpha = 1, 2.$$

Вважаємо при цьому, що при заданих функціях $f(x)$, $\mu(x)$, розв'язок поставленої задачі існує, єдиний і достатньо гладкий.

На відміну від крайових задач для рівнянь гіперболічного і параболічного типу (нестационарних), крайова задача (6.1), (6.2) – стаціонарна, а рівняння (6.1), з огляду класифікації ДРЧП, є еліптичного типу.

п.1 Побудова різницевої схеми

Крайові задачі, які ми розглядали попередніх темах, записувались в однозв'язних просторових областях: відрізок, прямокутних, p -вимірний паралелепіпед. Відповідно і сіткові області $\bar{\omega}_h$ в різницевих схемах мали поділ на внутрішні сіткові вузли ω_h і граничні $\partial\omega_h$, при цьому жодних проблем з введенням сітки не виникало.

Якщо ми будемо розглядати постановку цих задач в однозв'язних плоских областях типу трикутник, трапеція, то уже виникає певна «проблема граничних точок», а саме – помістити вузли сіткової границі $\partial\omega_h$ на границю ∂D неперервної області так, щоб $\partial\omega_h \in \partial D$. Як варіанти вирішення цієї проблеми є:

- а.** введення рівномірних сіток, кроки яких зв'язані певною пропорцією $h_1/h_2 = \frac{m}{n}$;
- б.** введення нерівномірних сіток.

Набагато складніші завдання постають при побудові сітки для розв'язання задач в довільних областях (спрощений варіант – область складена з прямокутників, трикутників, трапецій), як однозв'язних, так і неоднотв'язних. Тут до «проблеми граничних точок» додається ще і обов'язкова вимога для сітки і сіткового шаблону $S(x)$, на якому записується РС – умова зв'язності сітки. Розглянемо ці поняття, які є загальними при побудові p -вимірних просторових сіток, як для стаціонарних так і для нестационарних крайових задач [33].

Нехай в p -вимірному евклідовому просторі маємо скінченну множину точок – сітка Ω . Кожній точці $x \in \Omega$ співставимо один і тільки один шаблон $S(x)$ – деяку підмножину точок Ω , якій належить точка x .

Озн.6.1. Околом точки x назовемо множину $S'(x) = S(x) \setminus \{x\}$, причому ця множина може бути і пустою.

Озн.6.2. Сітку Ω будемо називати зв'язною сіткою, якщо для довільних двох вузлів x_0, x'_0 таких, що хоча-б один з них має непустий окіл, існує множина вузлів $x_i \in \Omega$, $i = 1, 2, \dots, m$, таких що $x_1 \in S'(x_0)$, $x_2 \in S'(x_1)$, ..., $x_m \in S'(x_{m-1})$, $x'_0 \in S'(x_m)$.

Тобто, з довільного вузла x_0 сітки Ω , шаблон РС $S(x)$ можна послідовно перемістити («протягти», використовуючи точки сітки Ω) в вузол x'_0 . Наприклад, якщо маємо зображений на Рис.6.2 шаблон $S(x)$ для сітки Ω_1 Рис.6.1, то з жодного вузла нижнього/верхнього прямокутника ми його не «протягнемо» через область, яка їх поєднує. Отже така сітка – незв'язна, на відміну від сітки Ω_2 Рис.6.1.

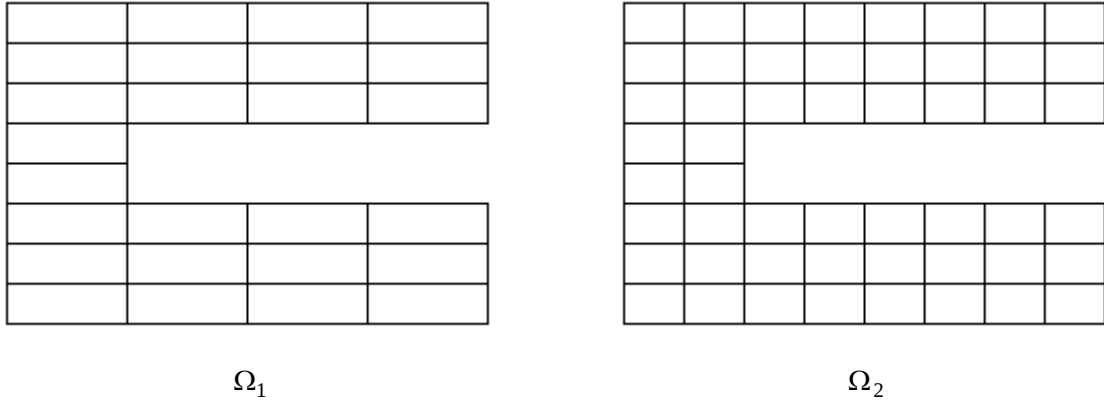


Рис.6.1 Прямокутні сітки в однозв'язній області: Ω_1 – незв'язна сітка; Ω_2 – зв'язна.

Ці загальні вимоги до побудови просторових сіток ми будемо використовувати при розв'язанні практичних задач по даній темі.

Тепер перейдемо до побудови РС для задачі (6.1)-(6.2). Аналогічно двовимірним рівнянням параболічного типу, уведемо в просторовій області $\bar{D}$ прямокутну рівномірну сітку $\bar{\omega}_h$:

$\bar{\omega}_h = \omega_h \cup \partial\omega_h = \bar{\omega}_{h_1} \times \bar{\omega}_{h_2}$, $\bar{\omega}_{h_\alpha} = \{x_\alpha^{(k_\alpha)} = (k_\alpha - 1)h_\alpha, k_\alpha = \overline{1, N_\alpha}, l_\alpha = (N_\alpha - 1)h_\alpha\}$, $\alpha = 1, 2$, і для апроксимації диференціальних операторів (6.1) будемо використовувати мінімальну кількість точок $\bar{\omega}_h$ – п'ятиточковий шаблон $S(x) = \{x\} \cup S'(x)$, $x = (x_1^{(k_1)}, x_2^{(k_2)})$, $S'(x) = \{x_1^{(k_1 \mp 1)}, x_2^{(k_2 \mp 1)}\}$, зображений на Рис.6.2.

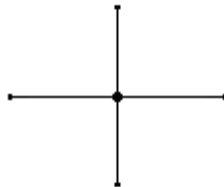


Рис.6.2 П'ятиточковий шаблон $S(x) = \{x\} \cup S'(x)$ з центральною точкою $x = (x_1^{(k_1)}, x_2^{(k_2)})$.

Задачі (6.1), (6.2) поставимо у відповідність наступну різницеву задачу:

$$\Delta y(x) = -\varphi(x), \quad x \in \omega_h; \quad (6.3)$$

$$y(x) = \mu(x), \quad x \in \partial\omega_h, \quad (6.4)$$

де позначено

$$\Delta y = \Delta_1 y + \Delta_2 y, \quad \Delta_\alpha y = y_{(\bar{x}_\alpha, x_\alpha), k_\alpha}, \quad \alpha = 1, 2;$$

$$\varphi(x) - f(x) = O(|h|^2), \quad |h|^2 = h_1^2 + h_2^2.$$

п.2 Похибка апроксимації

Якщо $f \in C^{(2)}(D)$, то $\varphi(x) = f(x)$, $x \in \omega_h$ і для похибки апроксимації (6.3) отримаємо, що він співпадає з похибкою апроксимації диференціального оператора різницеvim:

$$\psi = (\Lambda y(x) - \varphi(x)) - (Lu(x) - f(x)) = \Lambda(x) - Lu(x), \quad x \in \omega_h^-.$$

Враховуючи, що $\Lambda_\alpha u = L_\alpha u + \frac{h_\alpha^2}{12} L_\alpha^2 u + O(h_\alpha^4)$, маємо

$$\psi = \frac{h_1^2}{12} L_1^2 u + \frac{h_2^2}{12} L_2^2 u + O(h_1^4 + h_2^4), \quad u \in C^{(4)}(D),$$

а позначивши $M_4 = \max_{D, \alpha} \left| \frac{\partial^4 u}{\partial x_\alpha^4} \right|$, отримаємо $|\psi| \leq M_4 \frac{|h|^2}{12} = O(h_1^2 + h_2^2)$.

п.3 Стійкість і збіжність РС

Для доведення стійкості по крайовим умовам і по правій частині, а також доведення збіжності, використовують *принцип максимуму* для РС, побудований на аналогу принципу максимуму для еліптичних задач в диференціальному виді. При проведенні цього дослідження, сімейство РС для стаціонарних задач записують в *канонічному вигляді*

$$A(\eta)y(\eta) = \sum_{\xi} B(\eta, \xi)y(\xi) + F(\eta), \quad \eta \in S(x), \quad \xi \in S'(x) \quad (6.5)$$

де коефіцієнти цього рівняння пов'язані певними співвідношеннями – *умовами додатності коефіцієнтів*

$$A(\eta) > 0, \quad B(\eta, \xi) > 0, \quad A(\eta) - \sum_{\xi} B(\eta, \xi) \geq 0.$$

Форма (6.5) при $x \in \omega_h^-$ представляє різницеву схему (6.3). Для межових точок $x \in \partial\omega_h^-$ шаблон $S'(x)$ є пуста множина і (6.5) приймає вигляд $A(\eta)y(\eta) = F(\eta) \equiv A(\eta)\mu(\eta)$, а отже представляє крайові умови (6.4).

З огляду значного об'єму викладок по застосування принципу максимуму для еліптичних задач ми його розглядати не будемо, а при бажанні можна ознайомитись у [33]).

Використовуючи відповідні теореми принципу максимуму, для різницевої задачі (6.3), (6.4) маємо їх стійкість по граничним умовам і по правій частині, а також збіжність з порядком, який відповідає порядку апроксимації.

Завдання 1. Знайти коефіцієнти форми (6.5) для схеми (6.3) і перевірити умови додатності коефіцієнтів.

Завдання 2. Знайти похибку апроксимації рівняння (6.1) на шаблоні $S(x)$ (Рис.6.2) наступним різницеvim рівнянням

$$\Lambda y(x) = -\frac{1}{6} \sum_{\xi \in S} \beta(\xi) f(\xi), \quad x \in \omega_h^-; \quad \beta(\xi) = \begin{cases} 2, & \xi \in S(x); \\ 1, & \xi \in S'(x). \end{cases}$$

п.4 Знаходження розв'язку

Для стаціонарних різницеvих задач типу (6.3), (6.4) досить складним є питання фактичного обчислення сіткової функції. З ще більшими труднощами ми стикаємося при узагальненні цієї задачі для розмірностей $p > 2$. Це зумовлено тим, що матриця A таких СЛАР характеризується :

- високим порядком – число невідомих $\approx N^p$ для p -вимірної схеми з N інтервалами по кожній координаті;
- стрічковою будовою, при цьому стрічка слабо заповнена і має півширину N^{p-1} ;
- великим числом обумовленості $\text{cond}(A) \approx O(h^{-2})$;
- реалізація *точних* методів (типу Гаусса) вимагає кількість операцій $\approx N^{3p-2}$.

В зв'язку з цим, для деяких частинних випадків еліптичних задач, розроблені спеціальні точні методи (див. [31–33]):

1. швидке перетворення Фур'є;
2. метод повної редукції;
3. метод матричної прогонки.

У випадку складних областей $\bar{D}$ (саме такі ми будемо використовувати в учбових задачах), або загальних виразів диференціальних операторів для еліптичних задач, отримані РЗ розв'язують за допомогою ітераційних методів (ІМ) [26–30]. Почнемо з простих по реалізації методів – Якобі і Зейделя. З огляду представлення ІМ у вигляді однокрокового ІП в канонічному виді :

$$\mathbf{B} \frac{y^{(s+1)} - y^{(s)}}{\tau_{(s+1)}} + \mathbf{A} y^{(s)} = f,$$

то м.Якобі – явний, а м.Зейделя – неявний, але при цьому матриця $\mathbf{B}$ – трикутна, що дає можливість легко знайти обернену. Ці методи мають досить прості алгоритми, які можна легко реалізувати на MATLAB, Octave та інших СКМ або СП.

Алгоритм ІМ Якобі для системи (6.3), (6.4) записуємо у вигляді :

1. $H = 2(h_1^2 + h_2^2)$, $A = h_2^2/H$, $B = h_1^2/H$, $C = h_1^2 h_2^2/H$; розрахунок const, незалежних від s ;
2. $y(x) = \mu(x)$, $x \in \partial\omega_{\bar{h}}$; визначення розв'язку на границі області $\partial\omega_{\bar{h}}$;
3. $s = 0$: $y^{(0)}(x)$ – лінійна інтерполяція по вертикалі|горизонталі крайових умов, $x \in \omega_{\bar{h}}$;
4. Цикл по $s = 0, 1, 2, \dots$:

$$y_{k_1, k_2}^{(s+1)} = A * (y_{k_1-1, k_2}^{(s)} + y_{k_1+1, k_2}^{(s)}) + B * (y_{k_1, k_2-1}^{(s)} + y_{k_1, k_2+1}^{(s)}) + C * f_{k_1, k_2},$$

$$k_1 = 2, N_1 - 1, \quad k_2 = 2, N_2 - 1;$$

до виконання умови $\|y^{(s+1)} - y^{(s)}\|_C \leq \varepsilon$, $0 < \varepsilon < 1$;

Алгоритм ІМ Зейделя для системи (6.3), (6.4) записуємо аналогічно, змінюємо тільки розрахункову формулу (6.6) на наступну :

$$y_{k_1, k_2}^{(s+1)} = A * (y_{k_1-1, k_2}^{(s+1)} + y_{k_1+1, k_2}^{(s)}) + B * (y_{k_1, k_2-1}^{(s+1)} + y_{k_1, k_2+1}^{(s)}) + C * f_{k_1, k_2},$$

$$k_1 = 2, N_1 - 1, \quad k_2 = 2, N_2 - 1;$$

Швидкість збіжності цих ІМ невисока, причому м.Зейделя збігається трохи швидше за м.Якобі, але теоретична оцінка кількості ітерацій обох методів становить $s \approx O(|h|^{-2})$ і для отримання точності $\varepsilon \approx [10^{-6}, 10^{-8}]$ необхідно виконати значне число ітерацій $\approx 10^5$. Детальніше цю інформацію розглянемо в п.5.

Якщо сіткову функцію y_{k_1, k_2} представити у вигляді прямокутної матриці $U(i, j)$, де $k_1 = i$ і $k_2 = j$ – індекси i -рядка і j -стовпця, то програмна реалізація цих алгоритмів дійсно проста, а при певній побудові програми – не вимагає зберігання окремих матриць для ітерацій s і $s+1$. З деталями також розберемося в п.5.

Розглянемо ще одного представника неявних ІМ – *метод релаксації* з $\tau_{(s+1)} = \omega$, $\omega = \text{const} \in (0, 2)$. При значенні $\omega = 1$ він співпадає з методом Зейделя, а при $\omega \in (1, 2)$ – має назву *метод верхньої релаксації*. Для його реалізації змінимо в алгоритмі м.Зейделя обчислення констант

1. $H = 2(h_1^{-2} + h_2^{-2})$, $C = \omega/H$, $A = h_1^{-2}C$, $B = h_2^{-2}C$, $D = 1 - \omega$.

і розрахункову формулу (6.7) на наступну

$$y_{k_1, k_2}^{(s+1)} = A * (y_{k_1-1, k_2}^{(s+1)} + y_{k_1+1, k_2}^{(s)}) + B * (y_{k_1, k_2-1}^{(s+1)} + y_{k_1, k_2+1}^{(s)}) + C * f_{k_1, k_2} + D * y_{k_1, k_2}^{(s)},$$

$$k_1 = 2, N_1 - 1, \quad k_2 = 2, N_2 - 1;$$

Для задачі Діріхле (6.1)–(6.2) при умові $(h_1 = h_2 = h) \& (l_1 = l_2 = l)$, відоме оптимальне значення ω , яке визначається за формулою $\omega_{opt} = (0.5 + \sin(0.5\pi * h))^{-1}$ і теоретичні оцінки числа ітерацій м.релаксації з $\omega = \omega_{opt}$ становлять $\approx O(h^{-1})$.

В подальшому ми розглянемо і більш потужні (по зменшенню числа ітерацій) ІМ розв’язання різницевої задачі для еліптичних крайових задач.

п.5 Порівняння ІМ Якобі, Зейделя, релаксації та їх програмна реалізація

Розглянемо порівняння ІМ Якобі, Зейделя і релаксації з $\omega = \omega_{opt}$, а також технологію їх програмної реалізації, яка буде використовуватися і в подальшому для більш складних задач.

Почнемо з аналізу розрахункових формул (6.6), (6.7), (6.8) цих методів. По кількості арифметичних дій для знаходження $y_{k_1, k_2}^{(s+1)}$ м.Якобі і м.Зейделя співпадають, а м.релаксації – на одну дію більше, тобто, можна вважати, що відрізняються зовсім мало. Отже будемо їх порівнювати по числу ітерацій. Число ітерацій залежить від ε – точності ІМ і h – кроку різницевої схеми, який безпосередньо пов’язаний з $N = (N_1 - 2) \times (N_2 - 2)$ – вимірністю A матриці СЛАР (6.6) | (6.7) | (6.8). Найменше (δ) і найбільше (Δ) власне число різницевого оператора Лапласа (Δ) обчислюються за формулами

$$\delta = 8h^{-2} \sin^2\left(\frac{\pi}{2}h\right), \quad \Delta = 8h^{-2} \cos^2\left(\frac{\pi}{2}h\right),$$

а їх відношення $\xi = \delta/\Delta = \tan^2\left(\frac{\pi}{2}h\right) \underset{h \rightarrow 0}{=} \frac{1}{4}(\pi * h)^2 + O(h^4)$. Наслідком малості числа ξ є погана обумовленість матриці СЛАР, а також повільна збіжність ІМ Якобі і Зейделя. Якщо позначити $n_0(\varepsilon)$ – число ітерацій, необхідних для досягнення точності ε , то при $h \rightarrow 0$, маємо такі теоретичні оцінки [33]:

Таблиця 6.1 Теоретичні оцінки числа ітерацій для ІМ.

	<i>м. Якобі</i>	<i>м. Зейделя</i>	<i>м. верхньої релаксації ($1 < \omega < 2$)</i>
$n_0(\varepsilon) \approx$	$2 \frac{\lg(\varepsilon^{-1})}{\pi^2 h^2}$	$\frac{\lg(\varepsilon^{-1})}{\pi^2 h^2}$	$2 \frac{\lg(\varepsilon^{-1})}{\pi * h}$

Зрозуміло, що реальне число ітерацій залежить також і від початкового наближення $y^{(0)}(x)$, але теоретичні значення в Таблиці 6.1 отримані для довільного початкового наближення.

Виконаємо деяке співставлення (воно, зрозуміло, не може являтися загальним показником, але ...) цих теоретичних оцінок з практичними результатами для деякої модельної задачі, а саме

$$l_{1,2} = 1; \quad \mu(x) = 1, \quad x \in \partial D; \quad f(x_1, x_2) = 30(x_1(1 - x_1) + x_2(1 - x_2)). \quad (6.9)$$

При таких параметрах, точним розв’язком крайової задачі (6.1), 6.2) буде функція

$$u(x_1, x_2) = 1 + 15x_1(1 - x_1)x_2(1 - x_2).$$

Тепер розглянемо деякі ключові моменти програмної реалізації числового розв’язання модельної задачі (6.1), 6.2).

По-перше, запис СЛАР різницевої задачі у вигляді $A * \bar{Y} = \bar{F}$ передбачає, що невідомі і рівняння певним чином упорядковані. Для ІМ, під упорядкуванням розуміють порядок, в якому відбувається уточнення компонент наближеного розв’язку в алгоритмі ІМ. Це може бути *блочне, точкове* (1-натуральне упорядкування по горизонтальним|вертикальним лініям; 2-червоно-чорне розбиття по лініям). Технічно – це спосіб перенумерування двовірного масиву індексів різнице-

вої задачі $I_J = (k_1, k_2)_{k_1, k_2=2}^{N_1-2, N_2-2}$ в одновимірний масив індексів K , а отже цей процес впливає на структуру матриці A СЛАР.

Потрібно зауважити, що упорядкування невідомих всередині лінії не впливає на поведінку ітераційного процесу, але може суттєво вплинути на число арифметико-логічних дій на одній ітерації. Також зауважимо, що деякі ІМ «працюють» тільки з певним упорядкуванням.

Відмітимо, що при реалізації ІМ Якобі, Зейделя і релаксації немає необхідності переходити від матриці індексів I_J до вектора індексів K . Тобто, ми можемо ефективно реалізувати відповідний ІМ, використовуючи для зберігання невідомих деяку матрицю $U(I_J)$.

По-друге, виникає питання: чи потрібно використовувати різні матриці $U(I_J)$ для різних (s) і $(s+1)$ номерів ітерацій? Розглянемо пильніше схему розрахунку по формулам (6.6), (6.7), (6.8). Дійсно, невідомі, пов'язані шаблоном $S(x)$ у точці x з індексами $(k_1 = i, k_2 = j)$ належать до різних (s) і $(s+1)$ номерів ітерацій. Але, якщо розрахункову схему ІМ підпорядкувати певній процедурі «переміщення» шаблону $S(x)$ по вузлам сітки ω_h , наприклад, переміщення в такому порядку:

$$\begin{aligned} (i=2, j=2) \rightarrow (i=3, j=2) \rightarrow \dots \rightarrow (i=N_1-1, j=2) \rightarrow \\ \rightarrow (i=2, j=3) \rightarrow \dots \rightarrow (i=N_1-1, j=3) \rightarrow \\ \rightarrow \dots \rightarrow \\ \rightarrow (i=2, j=N_2-1) \rightarrow \dots \rightarrow (i=N_1-1, j=N_2-1), \end{aligned}$$

то можна скористатись тільки однією матрицею $U(I_J)$ і деякими додатковими змінними. Дійсно, знаходження $y_{k_1=i, k_2=j}^{(s+1)}$ в (6.6), (6.7), (6.8), як елемента матриці $U(i, j)$, реалізуємо за схемою (позначення див. Рис.6.3) :

$$old = U(i, j);$$

$$new = A * (oldm + U(i+1, j)) + B * (old0 + U(i, j+1)) + C * f_{k_1, k_2} + \left\{ \begin{matrix} 0 \\ D \end{matrix} \right\} * old; \quad (6.10)$$

...

$$U(i, j) = new;$$

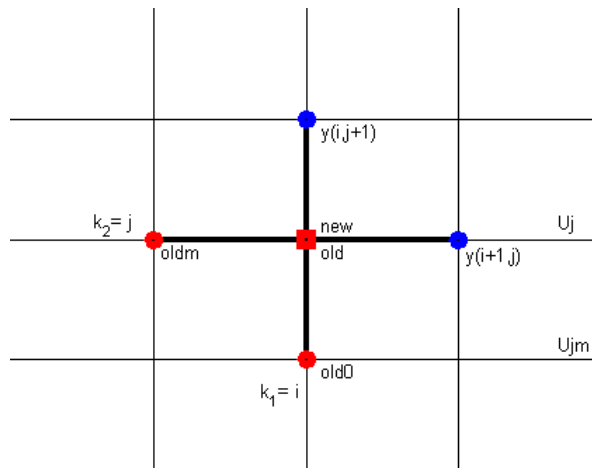


Рис.6.3 Фрагмент сітки ω_h з шаблоном $S(x)$.

В схемі (6.10), якщо реалізуємо тільки ІМ Зейделя або тільки ІМ релаксації, то додаткові змінні не потрібні. При реалізації ІМ Якобі з використанням схеми (6.10) (тобто, тільки одна матриця $U(I_J)$), маємо скористатися двома додатковими змінними – це одновимірні вектори $\vec{Ujm}, \vec{Uj}$ (див. Рис.6.3), які будуть послідовно

зберігати значення $y_{k_1, k_2}^{(s)}$ для $k_2 = j-1$ і $k_2 = j$. Тобто, підтримувати наступну рекурентну реалізацію :

```

 $\vec{U_j} = U(1:N_1, 1);$ 
for  $j = 2:N_2 - 1$ 
 $\vec{U_{jm}} = \vec{U_j}; \vec{U_j} = U(1:N_1, j);$ 

```

...

end

Вся інша частина програмної реалізації ІМ Якобі, Зейделя, релаксації повністю відповідає реалізації алгоритмів, описаних в попередньому п.4 Лекції.

Зазначимо, що в програмі *Lp11.m*, створеній для співставлення теоретичних оцінок числа ітерацій $\rho(\epsilon)$ з результатами числового експерименту на модельній задачі (6.1), (6.2), (6.9), всі ці методи реалізовані (згідно (6.10)) в одній функції *im_JZr*. В цій функції, враховуючи, що за умову закінчення ІМ обрано поелементне співпадіння значень $y_{k_1=i, k_2=j}^{(s)}$ і $y_{k_1=i, k_2=j}^{(s+1)}$, використовуємо прийом «прапорця» : інструкція обчислення умови $Fla = abs(new-old) < ep$ відбувається тільки при значеннях *true* змінної *Fla* (це зменшує кількість арифметичних операцій).

Текст М-функції *Lp11* подано нижче. Тут використовуються функції: *u, f* – анонімні (опис точного розв'язку і правої частини рівняння Пуассона), *muh, muh* – внутрішні (опис крайових умов); *im_JZr, viewU* – зовнішні (реалізація ІМ і візуалізація результатів наближеного, точного розв'язку, похибки і числових характеристик ІМ) :

```

function Lp11
clear all, close all, clc
% Задача Діріхле для р-я Пуассона в квадратній області.
% Порівняння ітераційних процесів - 1.

a = 1; b = 1;
u = @(x,y) (1+15.*(x.*(a-x).*y.*(b-y)));
f = @(x,y) (30.*(x.*(a-x)+y.*(b-y)));

ep = 1e-8; itmax = 30000; Lep = log10(1./ep);
fm = '%12s %5u %5u%9.3f %13.5e';

N = [5; 11; 21; 41; 81];
nN = length(N); Kit = zeros(nN,3);
for n = 1 : nN
Nx = N(n); Ny = Nx; % кількість вузлів сітки
X = linspace(0, a, Nx); Y = linspace(0, b, Ny);
Hx = X(2)-X(1); Hy = Y(2)-Y(1); n0e = 2.*Lep./(pi.*Hx);

UU = zeros(Ny,Nx);
for j = 1 : Ny
for i = 1 : Nx
UU(j,i) = u(X(i),Y(j));
end
end

mc = 'М.Якобі';
t1 = clock();
[U, it] = im_JZr(0, f, @muh, @muv, X, Y, 0, ep, itmax);
t = clock(); t = etime(t,t1); Kit(n,1) = it;
fprintf('Nx=Ny=%4u Hx=Hy=%7.5f\n', Nx, Hx);
fprintf('Ітерац.метод Ітер. ІтерТ Час(сек) max(max|U-u|) ro\n');
t1 = max(max(abs(U-UU))); n0 = fix(n0e./(pi.*Hx));

```

```

fprintf([fm, '\n'], mc, it, n0, t, t1);
viewU(2, X, Y, U, 15, UU);

mc = 'м.Зейделя';
t1 = clock();
[U, it] = im_JZr(1, f, @muh, @muv, X, Y, 1, ep, itmax);
t = clock(); t = etime(t, t1); Kit(n, 2) = it;
t1 = max(max(abs(U-UU))); n0 = fix(n0e./(2.*pi.*Hx));
fprintf([fm, '\n'], mc, it, n0, t, t1);
viewU(2, X, Y, U, 15, UU);

mc = 'м.релаксації';
ro = 1./(0.5+sin(0.5.*pi.*Hx)); % опт.параметр
t1 = clock();
[U, it] = im_JZr(2, f, @muh, @muv, X, Y, ro, ep, itmax);
t = clock(); t = etime(t, t1); Kit(n, 3) = it;
t1 = max(max(abs(U-UU))); n0 = fix(n0e);
fprintf([fm, ' %7.5f\n'], mc, it, n0, t, t1, ro);
viewU(2, X, Y, U, 15, UU);
end

pause, close all
clear U UU
N = N - 1; M = linspace(N(1), N(end), 101);
r = 2.*Lep./pi.*M; J = r./pi.*M; Z = 0.5.*J;
plot(N, Kit(:, 1), 'ks ', 'LineWidth', 2); hold on;
plot(N, Kit(:, 2), 'ko ', 'LineWidth', 2); hold on;
plot(N, Kit(:, 3), 'b* ', 'LineWidth', 2); hold on;
plot(M, J, 'r -.', 'LineWidth', 2); hold on;
plot(M, Z, 'r -.', 'LineWidth', 2); hold on;
plot(M, r, 'r --', 'LineWidth', 2); hold off;
legend('Якобі', 'Зейделя', 'релаксації', ...
       'Як n0', 'Зе n0', 'pe n0', 'Location', 'Best');
grid on; xlabel('h^{-1}'); ylabel('Kit');
title('Число ітерацій від h');

end

function [g1, g2] = muh(x)
    g1 = 1.0; g2 = 1.0;
end

function [g1, g2] = muv(y)
    g1 = 1.0; g2 = 1.0;
end

function [U, it] = im_JZr(m, f, muh, muv, X, Y, ro, ep, itmax)
% m=0 м.Якобі, m=1 м.Зейделя, m=2 м.релаксації
% параметри різницевої схеми :
Nx = length(X); Nx1 = Nx - 1; Hx = X(2) - X(1);
Ny = length(Y); Ny1 = Ny - 1; Hy = Y(2) - Y(1);
% обчислення констант розрахункових формул :
if m == 2
    Hx2 = 1./(Hx.*Hx); Hy2 = 1./(Hy.*Hy); H = 2.*(Hx2+Hy2);
    C = ro./H; A = Hx2.*C; B = Hy2.*C; D = 1-ro;
else
    Hx2 = Hx.*Hx; Hy2 = Hy.*Hy; H = 2.*(Hx2+Hy2);
    A = Hy2./H; B = Hx2./H; C = Hx2.*Hy2./H; D = 0;
end
% формування граничних умов і початкового наближення :
U = zeros(Ny, Nx);
for i = 1 : Nx % КУ на "горизонталях" y=0, y=b
    [c1, c2] = muh(X(i)); U(1,i) = c1; U(Ny,i) = c2;

```

```

end
for j = 2 : Ny1          % КУ на "вертикалях" x=0, x=a
    [c1, c2] = muv(Y(j));
    % + лінійна інтерполяція по горизонталі
    U(j,:) = linspace(c1, c2, Nx);
end
clear c1 c2

it = 0; Fla = 0;
while ~Fla & it < itmax    % ітер.процес
    Fla = 1; it = it+1;
    if m == 0
        Uj = U(1,:);
    end
    for j = 2 : Ny1        % рух "знизу вгору"
        jm = j-1; y = Y(j);
        if m == 0
            Ujm = Uj; Uj = U(j,:);
        end
        for i = 2 : Nx1    % рух "зліва направо"
            if m == 0
                old0 = Ujm(i); oldm = Uj(i-1);
            else
                old0 = U(jm,i); oldm = U(j,i-1);
            end
            old = U(j,i);
            new = A.*(oldm + U(j,i+1)) + ...
                B.*(old0 + U(j+1,i)) + C.*f(X(i),y);
            if m == 2
                new = new + D.*old;
            end
            if Fla
                Fla = abs(new-old) < ep;
            end
            U(j,i) = new;
        end
    end
end
end
end
end

```

```

function viewU(s, X, Y, U, kc, UU)
    [Mx, My] = meshgrid(X, Y); close all;
    surf(Mx, My, U);
    e2 = 'Різнцевий розв'язок';
    title(e2); xlabel('x'); ylabel('y'); zlabel('U');
    shading interp; colormap(pink); %view(40,20);
    if s, pause(s);
    else, pause;
    end
    close all;
    if kc
        contour(Mx, My, U, kc);
        xlabel('x'); ylabel('y'); title(e2); colormap hsv;
        if s, pause(s);
        else, pause;
        end
        close all;
    end

    if nargin == 5
        return
    end

    surf(Mx, My, UU);

```

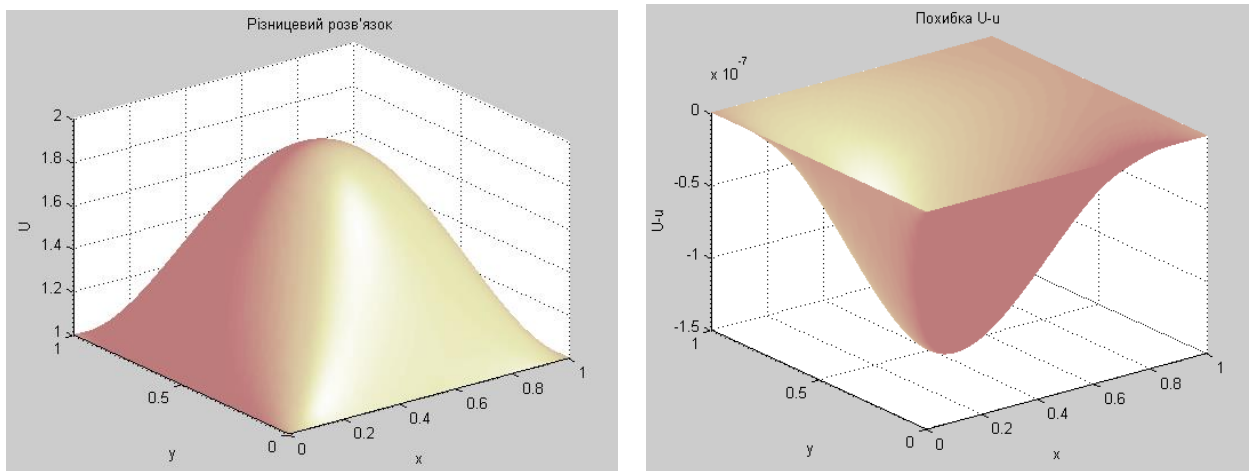
```

e2 = 'Точний розв'язок';
title(e2); xlabel('x'); ylabel('y'); zlabel('u');
shading interp; colormap(pink); %view(40,20);
if s, pause(s);
else, pause;
end
close all;
if kc
    contour(Mx, My, UU, kc);
    xlabel('x'); ylabel('y'); title(e2); colormap hsv;
    if s, pause(s);
    else, pause;
    end
    close all;
end

Uu = U - UU;
surfl(Mx, My, Uu);
e2 = 'Похибка U-u';
title(e2); xlabel('x'); ylabel('y'); zlabel('U-u');
shading interp; colormap(pink); %view(40,20);
if s, pause(s);
else, pause;
end
end
end

```

Отримано наступний результат її виконання (показано графік наближеного розв'язку методом релаксації при $h = 0.0125$, а точний розв'язок тут не приводимо з огляду на величину похибки) :



а)

б)

Рис.6.4 а) – графік числового розв'язку модельної задачі; б) – графік похибки.

Nx=Ny= 5 Nx=Ny=0.25000	
Ітерац.метод	Ітер. ІтерТ Час(сек) max(max U-u) ro
м.Якобі	51 25 0.016 2.09548e-008
м.Зейделя	27 12 0.000 9.27685e-009
м.релаксації	18 20 0.000 4.00514e-009 1.13291
Nx=Ny= 11 Nx=Ny=0.10000	
Ітерац.метод	Ітер. ІтерТ Час(сек) max(max U-u) ro
м.Якобі	308 162 0.703 1.93591e-007
м.Зейделя	162 81 0.485 8.98061e-008
м.релаксації	38 50 0.047 9.68729e-009 1.52338
Nx=Ny= 21 Nx=Ny=0.05000	
Ітерац.метод	Ітер. ІтерТ Час(сек) max(max U-u) ro
м.Якобі	1133 648 4.703 8.01211e-007

м.Зейделя	596	324	7.688	3.89121e-007	
м.релаксації	70	101	0.906	2.83825e-008	1.72873
Nx=Ny= 41 Nx=Ny=0.02500					
Ітерац.метод	Ітер.	ІтерТ	Час(сек)	max(max U-u)	ro
м.Якобі	4095	2593	129.204	3.22537e-006	
м.Зейделя	2160	1296	90.047	1.61372e-006	
м.релаксації	133	203	2.906	6.51855e-008	1.85439
Nx=Ny= 81 Nx=Ny=0.01250					
Ітерац.метод	Ітер.	ІтерТ	Час(сек)	max(max U-u)	ro
м.Якобі	14590	10375	1865.859	1.29535e-005	
м.Зейделя	7745	5187	1346.938	6.47385e-006	
м.релаксації	255	407	42.610	1.30574e-007	1.92443

Порівнюючи значення в колонці **Ітер.** (числовий експеримент) і в колонці **ІтерТ** ($\rho_0(\varepsilon)$ – теорія), бачимо, що ці значення суттєво (≈ 2 рази) відрізняються при «великих» значеннях h . При зменшенні значення h ця відмінність повільно, але зменшується і, мабуть, при $h \rightarrow 0$ буде мало різнитися. Розрахунки виконувалися зі значенням $\varepsilon = 10^{-8}$.

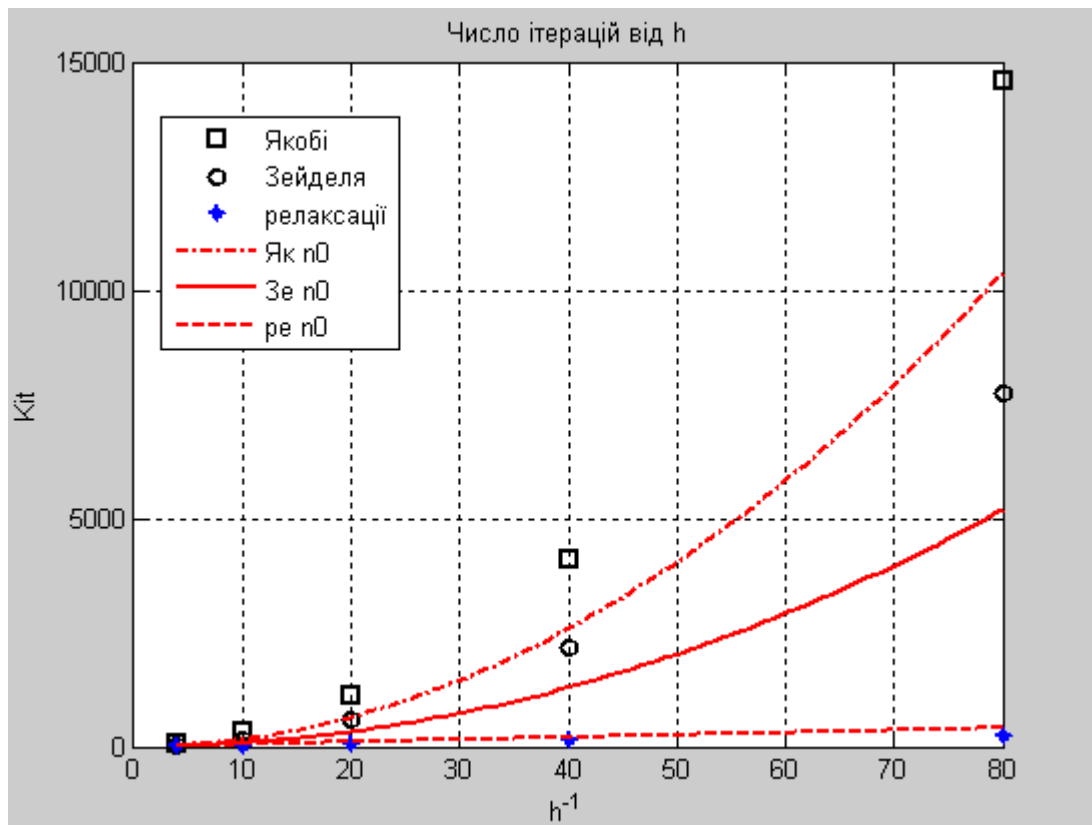


Рис.6.5 Залежність числа ітерацій від значення h . Меркерами відмічено результати чисельного експерименту.

ЛЕКЦІЯ № 12

ТЕМА № 6. Різницьові схеми для рівняння Пуассона з крайовими умовами Діріхле.

Продовжимо розглядати ітераційні методи розв'язання СЛАР, які отримані при побудові різницьових задач для стаціонарних крайових задач. Крім простих алгоритмів з повільною збіжністю існують і більш ефективні ІМ. Одним з напрямків отримання таких алгоритмів є *обчислення на усталювання*, коли за розв'язок крайової задачі для еліптичного рівняння беруть розв'язок $w(x_1, x_2, t)$ при $t \rightarrow \infty$ мішаної задачі для параболічного рівняння з тим же просторовим оператором, співпадаючими крайовими умовами і довільною початковою умовою:

$$\begin{cases} \frac{\partial w(x_1, x_2, t)}{\partial t} = \Delta w(x_1, x_2, t) + f(x_1, x_2), \\ x \equiv (x_1, x_2) \in D, \quad t > 0; \\ w(x, t) = \mu(x), \quad x \in \partial D; \\ w(x, 0) = w_0(x), \quad x \in D, \quad \forall w_0(x). \end{cases} \quad (6.11)$$

Усталювання стаціонарного розв'язку відбувається досить швидко завдяки експоненційному характеру затухання початкових даних. Отже, побудувавши деяку РС для еволюційної задачі, будемо її інтерпретувати як певний ітераційний алгоритм для стаціонарної задачі. Використаємо для розв'язання СЛАР (6.3), (6.4) наступні ітераційні методи:

- ✓ метод змінних напрямків (МЗН) з вибором оптимальних ітераційних параметрів (по Жордану);
- ✓ позмінно-трикутний метод (ПТМ) з чебишовським набором параметрів (ЧНП).

Розглянемо опис алгоритмів вказаних ІМ в тому об'ємі, який необхідний для написання програмного забезпечення (детальні теоретичні відомості про побудову ІМ, вибір ітераційних параметрів, дослідження збіжності і затрати операцій на одну сіткову точку див. [33]).

п.6 Алгоритм методу змінних напрямків з вибором оптимальних ітераційних параметрів (по Жордану)

Запишемо *поздовжно-поперечну РС* для параболічного рівняння (6.11), пов'язаного з (6.1)

$$y^{(0)}(x) = \begin{cases} \mu(x), & x \in \partial \omega_h^-; \\ \forall \text{ value}, & x \in \omega_h^-. \end{cases} \quad (6.12)$$

$$\begin{cases} \frac{\bar{y} - y^{(j)}}{\tau_{j+1}^{(1)}} = \Lambda_1 \bar{y} + \Lambda_2 y^{(j)} + f(x), & x \in \omega_h^-, \\ \bar{y}(x) = \mu(x), & x \in \partial \omega_h^-; \\ \frac{\hat{y} - \bar{y}}{\tau_{j+1}^{(2)}} = \Lambda_1 \bar{y} + \Lambda_2 \hat{y} + f(x), & x \in \omega_h^-, \\ \hat{y}(x) = \mu(x), & x \in \partial \omega_h^-; \end{cases}$$

Тут введено наступні позначення:

$$\bar{y} \equiv y^{(j+0.5)} \equiv y^{(j+0.5)}(x) = y(x, t_{j+0.5}), \quad \hat{y} \equiv y^{(j+1)} \equiv y^{(j+1)}(x) = y(x, t_{j+1}), \quad x \in \bar{\omega}_h,$$

де t_j – «сіткове значення часу», а $\tau_j^{(1)}, \tau_j^{(2)}$ – «сіткові кроки по часу» це параметри ІМ. У випадку прямокутної області і оператора $\Lambda = \Lambda_1 + \Lambda_2$, де Λ_α – додатно означені, самоспряжені, перестановні оператори, можна вибрати $\tau_j^{(1)}, \tau_j^{(2)}$ оптимальним чином

(по Жордану). В цьому випадку для обчислення однієї ітерації по МЗН потрібно $O((h_1 h_2)^{-1})$ арифметичних операцій, тобто $O(1)$ операцій на вузол сітки.

Тепер запишемо послідовність кроків цього ПП.

1. Виконуємо низку обчислень параметрів, пов'язаних з областю $\bar{D}$ і спектральними характеристиками різницьових операторів Λ_α :

$$\begin{aligned} o_\alpha &= 0.5\pi * h_\alpha / l_\alpha, \quad \delta_\alpha = 4h_\alpha^{-2} \sin^2(o_\alpha), \quad \Delta_\alpha = 4h_\alpha^{-2} \cos^2(o_\alpha), \quad \alpha = 1, 2; \\ t &= \sqrt{\frac{(\Delta_1 - \delta_1)(\Delta_2 - \delta_2)}{(\Delta_1 + \delta_2)(\Delta_2 + \delta_1)}}, \quad \eta = \frac{1-t}{1+t}, \quad \zeta = \frac{(\Delta_1 - \delta_1)\Delta_2}{(\Delta_2 + \delta_1)\Delta_1}; \\ p &= \frac{\zeta - t}{\zeta + t}, \quad r = \frac{\Delta_1 - \Delta_2 + (\Delta_1 + \Delta_2)p}{2\Delta_1\Delta_2}, \quad q = r + \frac{1-p}{\Delta_1}, \quad \theta = 0.0625(1 + 0.5\eta^2)\eta^2. \end{aligned}$$

2. По заданій точності $0 < \varepsilon < 1$ обчислюємо число ітерацій n :

$$n(\varepsilon) = \pi^{-2} \ln(4/\varepsilon) \ln(4/\eta), \quad n = [n(\varepsilon)].$$

3. Обчислюємо сіткову функцію на границі області $\bar{D}$, використовуючи (6.12)

$$y(x) = \mu(x), \quad x \in \partial\omega_{\bar{h}}.$$

4. Для ітерації $j = 0$ («початкові умови») обчислюємо сіткову функцію в області D $y^{(0)}(x)$, $x \in \omega_{\bar{h}}$. Тут маємо варіанти:

а) задати довільні значення, наприклад, $y^{(0)}(x) = 0$, $x \in \omega_{\bar{h}}$;

б) виконати лінійну інтерполяцію по горизонталі|вертикалі крайових умов.

5. Для кожної ітерації $j = 1, 2, \dots, n$

а) обчислюємо значення оптимальних ітераційних параметрів $\tau_j^{(1)}, \tau_j^{(2)}$:

$$k = \frac{2j-1}{2n}, \quad \omega = \frac{(0.5 + \theta)(1 + \theta^k)}{(1 + \theta^{1-k} + \theta^{1+k})\theta^{k/2}}; \quad \tau_j^{(1)} = \frac{q\omega + r}{1 + p\omega}, \quad \tau_j^{(2)} = \frac{q\omega - r}{1 - p\omega}.$$

б) послідовно для стовпців $k_2 = \overline{2, N_2 - 1}$, $(x_2 \in \omega_{h_2})$ розв'язуємо методом прогонки системи лінійних рівнянь:

$$\begin{cases} \bar{y}(1, x_2) = \mu_{y0}(x_2), \\ \bar{y} - \tau_j^{(1)} \Lambda_1 \bar{y} = y^{(j-1)} + \tau_j^{(1)} (\Lambda_2 y^{(j-1)} + f), \quad k_1 = \overline{2, N_1 - 1}, \\ \bar{y}(N_1, x_2) = \mu_{y0}(x_2). \end{cases}$$

в) послідовно для рядків $k_1 = \overline{2, N_1 - 1}$, $(x_1 \in \omega_{h_1})$ розв'язуємо методом прогонки системи лінійних рівнянь:

$$\begin{cases} y^{(j)}(x_1, 1) = \mu_{x0}(x_1), \\ y^{(j)} - \tau_j^{(2)} \Lambda_2 y^{(j)} = \bar{y} + \tau_j^{(2)} (\Lambda_1 \bar{y} + f), \quad k_2 = \overline{2, N_2 - 1}, \\ y^{(j)}(x_1, N_2) = \mu_{x0}(x_1). \end{cases}$$

п.7 Алгоритм ПТМ з чебишовським набором параметрів

Записуємо *чисто неявну РС* порядку $O(\tau + |h|^2)$ для параболічного рівняння (6.11), пов'язаного з (6.1)

$$\begin{aligned} y^{(0)}(x) &= \begin{cases} \mu(x), & x \in \partial\omega_{\bar{h}}; \\ \forall \text{ value}, & x \in \omega_{\bar{h}}. \end{cases} \\ \begin{cases} \frac{\hat{y} - y}{\tau} = (\Lambda_1 + \Lambda_2) \hat{y} + f(x), & x \in \omega_{\bar{h}}, \\ \hat{y}(x) = \mu(x), & x \in \partial\omega_{h_2}; \end{cases} \end{aligned}$$

Виконаємо деякі перетворення в різницьовому рівнянні

$$\begin{aligned} \frac{\hat{y}-y}{\tau} - (\Lambda_1 + \Lambda_2) \hat{y} \pm (\Lambda_1 + \Lambda_2) y &= f(x); \\ (E - \tau * \Lambda_1 - \tau * \Lambda_2) \frac{\hat{y}-y}{\tau} &= \Lambda y + f(x). \end{aligned} \quad (6.13)$$

Представимо матрицю $\Lambda_1 + \Lambda_2$ (складну для обчислення оберненої!) через трикутні матриці $\Lambda_1 + \Lambda_2 = -(R_1 + R_2)$, де

$$R_1 y = \frac{1}{h_1^2} (y_{k_1, k_2} - y_{k_1-1, k_2}) + \frac{1}{h_2^2} (y_{k_1, k_2} - y_{k_1, k_2-1}), \quad R_2 y = \frac{1}{h_1^2} (y_{k_1, k_2} - y_{k_1+1, k_2}) + \frac{1}{h_2^2} (y_{k_1, k_2} - y_{k_1, k_2+1}).$$

Враховуючи порядок РС, змінимо оператор біля похідної в рівнянні (6.13) на величину $\tau^2 R_1 R_2$ і виконаємо ланцюг перетворень:

$$\begin{aligned} (E + \tau R_1 + \tau R_2 + \tau^2 R_1 R_2) \frac{\hat{y}-y}{\tau} &= \Lambda y + f(x), \\ \mathbf{B} \frac{\hat{y}-y}{\tau} &= \Lambda y + f(x), \quad \mathbf{B} = (E + \tau R_1)(E + \tau R_2). \end{aligned} \quad (6.14)$$

В результаті отримали РС з факторизованою матрицею $\mathbf{B}$, залишається тільки перетворити (6.14) в ітераційний процес. Якщо τ буде пробігати чебишовський набір параметрів, то число ітерацій ПТМ буде порядку $O(|h|^{-1})$.

Тепер запишемо послідовність кроків розглянутого ПП.

1. Виконуємо низку обчислень параметрів, пов'язаних з областю $\bar{D}$ і спектральними характеристиками різницевого операторів Λ_α :

$$\begin{aligned} o_\alpha &= 0.5\pi * h_\alpha / l_\alpha, \quad \alpha = 1, 2, \quad \delta = 4h_1^{-2} \sin^2(o_1) + 4h_2^{-2} \sin^2(o_2), \quad \Delta = 4h_1^{-2} + 4h_2^{-2}; \\ \eta &= \frac{\delta}{\Delta}, \quad \omega = \frac{2}{\sqrt{\Delta\delta}}, \quad \gamma_1 = \frac{0.5\delta}{1 + \sqrt{\eta}}, \quad \gamma_2 = \frac{0.25\delta}{\sqrt{\eta}}, \quad t = \frac{\gamma_1}{\gamma_2}; \\ \rho_0 &= \frac{1-t}{1+t}, \quad \tau_0 = \frac{2}{\gamma_1 + \gamma_2}. \end{aligned}$$

2. По заданій точності $0 < \varepsilon < 1$ обчислюємо число ітерацій n :

$$n(\varepsilon) = \frac{\ln(2/\varepsilon)}{2\sqrt{2}\sqrt{\eta}}, \quad n = [n(\varepsilon)].$$

3. Будуємо числову множину M_n^* , т.з. *стійку множину оптимального набору чебишовських параметрів* [33].

4. Обчислюємо сіткову функцію на границі області $\bar{D}$, використовуючи (6.12)

$$y(x) = \mu(x), \quad x \in \partial\omega_h^-.$$

5. Для ітерації $j=0$ («початкові умови») обчислюємо сіткову функцію в області D $y^{(0)}(x)$, $x \in \omega_h^-$. Тут маємо варіанти:

а) задати довільні значення, наприклад, $y^{(0)}(x) = 0$, $x \in \omega_h^-$;

б) виконати лінійну інтерполяцію по горизонталі|вертикалі крайових умов.

6. Для кожної ітерації $j = 1, 2, \dots, n$

а) обчислюємо значення ітераційного параметру $\tau_j^* = \frac{\tau_0}{1 + \rho_0 \sigma_j^*}$, $\sigma_j^* \in M_n^*$.

б) послідовно для стовпців $k_2 = \overline{2, N_2 - 1}$, з лівого нижнього кута, біжучим розрахунком по рядках, знаходимо розв'язок системи лінійних рівнянь з трикутною (нижньою) матрицею:

$$(E + \omega R_1) \bar{W} = \Phi, \quad \bar{W}(x) = 0, \quad x \in \partial\omega_h^-;$$

де

$$R_1 y = (h_1^{-2} + h_2^{-2})y - (h_1^{-2} y_{k_1-1} + h_2^{-2} y_{k_2-1}),$$

$$\Phi = \Lambda V^{(j-1)} + f, \quad V^{(j-1)}(x) = \begin{cases} y^{(j-1)}(x), & x \in \omega_{\bar{h}}; \\ \mu(x), & x \in \partial\omega_{\bar{h}}. \end{cases}$$

в) послідовно для рядків $k_1 = \overline{2, N_1 - 1}$, з правого верхнього кута, біжучим розрахунком по стовпцям, знаходимо розв'язок системи лінійних рівнянь з трикутною (верхньою) матрицею:

$$(E + \omega R_2)W = \bar{W}, \quad W(x) = 0, \quad x \in \partial\omega_{\bar{h}_q};$$

де

$$R_2 y = (h_1^{-2} + h_2^{-2})y - (h_1^{-2} y_{k_1+1} + h_2^{-2} y_{k_2+1}), \quad \Lambda = \Lambda_1 + \Lambda_2 = -R_1 - R_2.$$

г) для внутрішніх точок сіткової області $\omega_{\bar{h}}$ знаходимо нове наближення :

$$y^{(j)}(x) = y^{(j-1)}(x) + \tau_j^* * W(x), \quad x \in \omega_{\bar{h}}.$$

У випадку лінійних рівнянь еліптичного типу :

- ◆ які розв'язуються для довільної (не прямокутної) області;
 - ◆ які містять в $L_\alpha u$ змінні коефіцієнти ($\Delta u(x_1, x_2) = \sum_{\alpha=1}^p L_\alpha u$, $L_\alpha u = \frac{\partial}{\partial x_\alpha} \left(K_\alpha(x) \frac{\partial u}{\partial x_\alpha} \right)$) і при цьому коефіцієнти $K_\alpha(x)$ в деякій локальній області можуть змінюватись в досить широкому діапазоні
- застосовують *модифікований ПТМ з ЧНП*. Оцінка кількості числа ітерацій цього методу становить $O(|h|^{-1/2})$!

п.8 Порівняння МЗН і ПТМ та їх програмна реалізація

Розглянемо порівняння ітераційних методів МЗН і ПТМ. Використаємо для цього модельну задачу (6.9) (див. Лекцію №11). Метод МЗН з оптимальними параметрами по Жордану програмно реалізуємо М-функцією *ADM_optJ*, а метод ПТП з ЧНП – М-функцією *ATM_Ch_ps*. Для самого порівняння методів створимо М-функцію *Lp12*, в якій використовуються функції *u, f* – анонімні (опис точного розв'язку і правої частини рівняння Пуассона), *muh, muh* – внутрішні (опис крайових умов); *AMD_optJ, ATM_Ch_ps, viewU* – зовнішні (реалізації ІМ і візуалізація результатів наближеного, точного розв'язку, похибки і числових характеристик ІМ) :

```
function Lp12
clear all, close all, clc
% Задача Діріхле для р-я Пуассона в квадратній області.
% Порівняння ітераційних процесів - 2.

a = 1; b = 1;
u = @(x,y) (1+15.*(x.*(a-x)).*y.*(b-y));
f = @(x,y) (30.*(x.*(a-x)+y.*(b-y)));

ep = 1e-6;
Nx = 101; Ny = Nx; % кількість вузлів сітки
Nx1 = Nx-1; Ny1 = Ny-1;
X = linspace(0, a, Nx); Y = linspace(0, b, Ny);
Hx = X(2)-X(1); Hy = Y(2)-Y(1);

UU = zeros(Ny,Nx);
for j = 1 : Ny
    for i = 1 : Nx
        UU(j,i) = u(X(i),Y(j));
```

```

    end
end

t1 = clock();
[U, it] = ADM_optJ(f, @muh, @muv, X, Y, ep);
t = clock(); t = etime(t, t1);
fprintf('М.Змін.Напр. Nx=%4u Ny=%4u Hx=%7.5f Hy=%7.5f\n', Nx, Ny, Hx, Hy);
fprintf('Ітер. Час(сек) max(max|U-u|)\n');
t1 = max(max(abs(U-UU)));
fprintf('%5u%9.3f %13.5e\n', it, t, t1);
viewU(2, X, Y, U, 15, UU);

t1 = clock();
[U, it] = ATM_Ch_ps(f, @muh, @muv, X, Y, ep);
t = clock(); t = etime(t, t1);
fprintf('ПТМ з ЧНП Nx=%4u Ny=%4u Hx=%7.5f Hy=%7.5f\n', Nx, Ny, Hx, Hy);
fprintf('Ітер. Час(сек) max(max|U-u|)\n');
t1 = max(max(abs(U-UU)));
fprintf('%5u%9.3f %13.5e\n', it, t, t1);
viewU(2, X, Y, U, 15, UU);

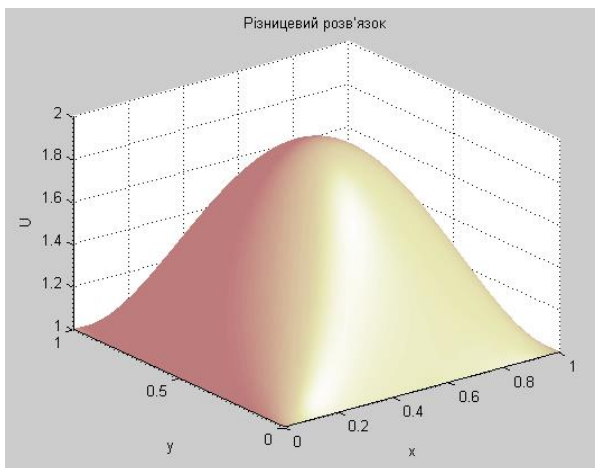
end

function [g1, g2] = muh(x)
    g1 = 1.0; g2 = 1.0;
end

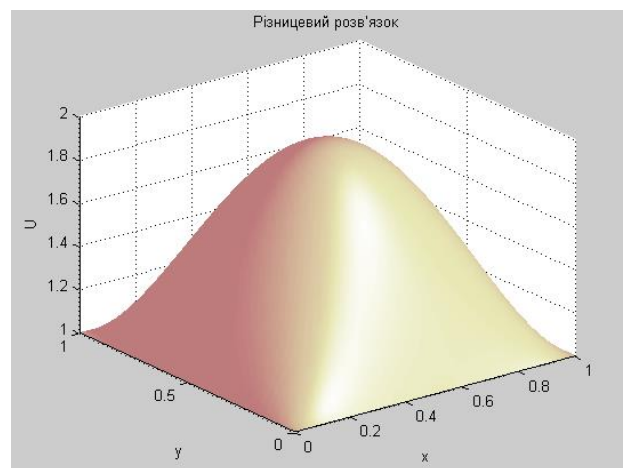
function [g1, g2] = muv(y)
    g1 = 1.0; g2 = 1.0;
end

```

Отримано наступний результат її виконання (показано графіки наближеного розв'язку, а точний розв'язок тут не приводимо з огляду на величину похибки) :

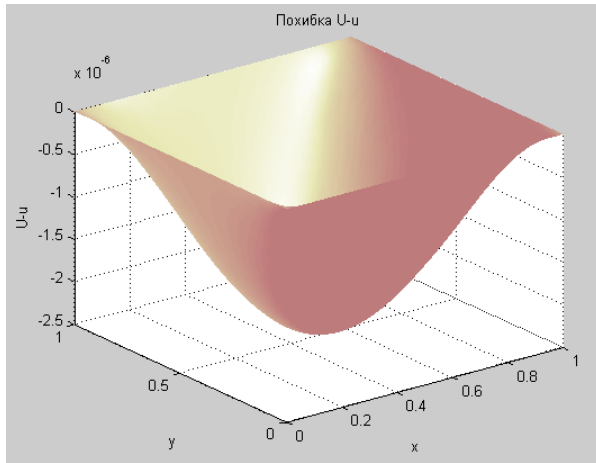


а)

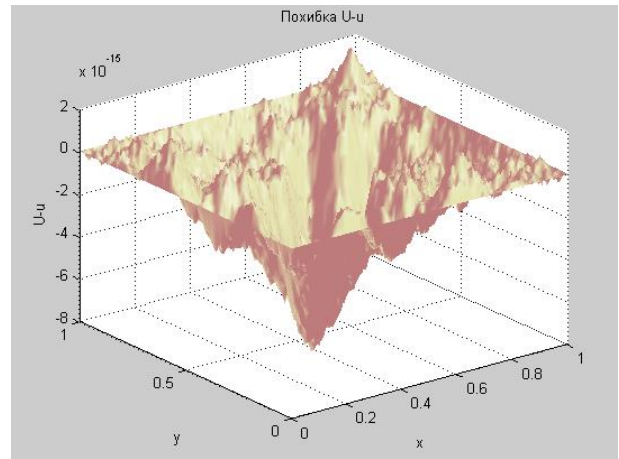


б)

Рис.6.6 Наближені розв'язки модельної задачі: **а)** – МЗП; **б)** – ПТМ.



а)



б)

Рис.6.7 Похибка наближеного розв'язку модельної задачі: **а)** – МЗП; **б)** – ПТМ.

А також значення числових характеристик ІМ :

```
м.Змін.Напр. Nx= 101 Ny= 101 Hx=0.01000 Hy=0.01000
Ітер. Час(сек) max(max|U-u|)
  14    4.187  2.43019e-006
ПТМ з ЧНП Nx= 101 Ny= 101 Hx=0.01000 Hy=0.01000
Ітер. Час(сек) max(max|U-u|)
  103   10.672  7.54952e-015
```

Тепер зробимо розрахунок (не змінюючи $\varepsilon = 10^{-6}$) для сітки згущеної вдвічі :

```
м.Змін.Напр. Nx= 201 Ny= 201 Hx=0.00500 Hy=0.00500
Ітер. Час(сек) max(max|U-u|)
  17   49.157  1.06606e-006
ПТМ з ЧНП Nx= 201 Ny= 201 Hx=0.00500 Hy=0.00500
Ітер. Час(сек) max(max|U-u|)
  146   53.500  7.54952e-015
```

ЛЕКЦІЯ № 13

ТЕМА № 7. Різницевий метод розв'язання крайових задач для лінійних рівнянь гіперболічного типу другого порядку.

п.1 РС для одновимірних рівнянь гіперболічного типу другого порядку

Розглянемо в області $\bar{D} = D \cup \partial D = [a, b] \times [0, T]$ крайову задачу для рівняння гіперболічного типу (математична модель фізичної задачі про коливання однорідної струни):

$$\frac{\partial^2 u(x, t)}{\partial t^2} = \frac{\partial^2 u(x, t)}{\partial x^2} + f(x, t), \quad (x, t) \in D \equiv (a, b) \times (0, T], \quad (7.1)$$

з крайовими умовами першого роду

$$u(a, t) = \mu_1(t), \quad u(b, t) = \mu_2(t), \quad t \in [0, T] \quad (7.2)$$

і початковими умовами

$$u(x, 0) = u_0(x), \quad (7.3)$$

$$\frac{\partial u(x, 0)}{\partial t} = u_1(x), \quad x \in [a, b] \quad (7.4)$$

Будемо вважати, що :

- початкові і крайові дані мають достатню гладкість і для них виконуються необхідні умови узгодження в точках $(a; 0)$, $(b; 0)$;
- функція f неперервно диференційована;
- шукана функція u також достатньо гладка,

а отже розв'язок (7.1)-(7.4) існує, єдиний і неперервно залежить від вхідних даних.

п.1.1 Побудова сімейства різницевих схем

Уведемо рівномірні сітки

$$\bar{\omega}_{h\tau} = \bar{\omega}_h \times \bar{\omega}_\tau; \quad \bar{\omega}_h = \{x_k = a + (k-1)h, \quad k = \overline{1, N}, \quad b - a = (N-1)h\};$$
$$\bar{\omega}_\tau = \{t_j = j\tau, \quad j = \overline{0, M}, \quad T = M\tau\},$$

і замінімо область $\bar{D}$ сітковою областю $\bar{\omega}_{h\tau}$, а також будемо використовувати наступні позначення

$$\dot{u} = \frac{\partial u}{\partial t}, \quad \ddot{u} = \frac{\partial^2 u}{\partial t^2}, \quad Lu = \frac{\partial^2 u}{\partial x^2}; \quad \ddot{u} = Lu + f, \quad L\ddot{u} = L^2u + Lf, \quad L^2u = L\dot{u} - Lf;$$

$$y_{t,j} = \frac{\hat{y}_k - y_k}{\tau}, \quad y_{\bar{t},j} = \frac{y_{t,j} - y_{\bar{t},j}}{\tau} = \frac{\hat{y}_k - 2y_k + \check{y}_k}{\tau^2}; \quad \Lambda y = y_{\bar{x},k};$$

$$0 \leq \sigma \leq \frac{1}{2}; \quad y^{(\sigma)} \equiv Y = \sigma \hat{y} + (1 - 2\sigma)y + \sigma \check{y} = y + \sigma \tau^2 y_{\bar{t},j}.$$

На дев'ятиточковому шаблоні $S = \{(x_{k+1}, t_{j-1}), (x_k, t_{j-1}), (x_{k+1}, t_j), (x_k, t_j), (x_{k+1}, t_{j+1}), (x_k, t_{j+1})\}$

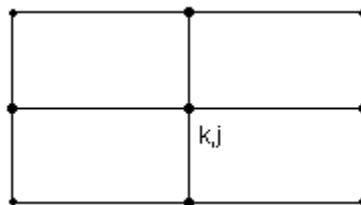


Рис.7.1 Дев'ятиточковий шаблон S з центральною точкою (x_k, τ_j) .

рівнянню (7.1) поставимо у відповідність однопараметричну сім'ю тришарових схем

$$y_{\bar{t},j} = \Lambda Y + \varphi, \quad (7.5)$$

$$k = \overline{2, N-1}; \quad j = \overline{1, M-1}.$$

Крайові умови (7.2) і початкова умова (7.3) апроксимуються точно

$$y_1^j = \mu_1(t_j), \quad y_N^j = \mu_2(t_j), \quad j = \overline{2, M}; \quad (7.6)$$

$$y_k^0 = u_0(x_k), \quad k = \overline{1, N}. \quad (7.7)$$

Для отримання апроксимації умови (7.4) з другим порядком по τ (враховуючи, що $y_{it,j} = \ddot{u}(x_k, \tau_j) + O(\tau^2)$), використаємо метод законтурних точок:

$$\begin{cases} \frac{y_k^1 - y_k^{-1}}{2\tau} = u_1(x_k), \quad O(\tau^2), \\ \frac{y_k^1 - 2y_k^0 + y_k^{-1}}{\tau^2} = y_{xx,k}^0 + f(x_k, 0), \quad O(\tau^2 + h^2). \end{cases}$$

Виключимо з цієї системи y_k^{-1} і отримаємо розрахункову формулу для y_k^1 , яка з порядком $O(\tau^2 + h^2)$ апроксимує умову (7.4)

$$y_k^1 = y_k^0 + \tau * u_1(x_k) + 0.5\tau^2(y_{xx,k}^0 + f(x_k, 0)), \quad k = \overline{1, N}. \quad (7.8)$$

Для отримання різницевого рівняння, яке апроксимує умову (7.4) з другим порядком, можна також використати метод підвищення порядку на розв'язку задачі. Запишемо ланцюжок дій, для отримання апроксимації порядку $O(\tau^2)$:

$$\begin{aligned} \frac{y_k^1 - y_k^0}{\tau} &= u_1(x_k), \quad O(\tau), \\ \frac{y^1 - y^0}{\tau} &\equiv \frac{u(x, \tau) - u(x, 0)}{\tau} = \frac{\partial u(x, 0)}{\partial t} + \frac{\tau}{2} \frac{\partial^2 u(x, 0)}{\partial t^2} + O(\tau^2) = \\ &= u_1(x) + \frac{\tau}{2} \left(\frac{\partial^2 u(x, 0)}{\partial x^2} + f(x, 0) \right) + O(\tau^2) = u_1(x) + \frac{\tau}{2} (u_0''(x) + f(x, 0)) + O(\tau^2) \end{aligned}$$

Отже маємо різницеве рівняння, необхідного порядку апроксимації

$$\frac{y_k^1 - y_k^0}{\tau} = u_1(x_k) + \frac{\tau}{2} (u_0''(x_k) + f(x_k, 0)) + O(\tau^2),$$

яке запишемо у вигляді розрахункової формули для y_k^1 (тут порядок $O(\tau^2)$!):

$$y_k^1 = y_k^0 + \tau * u_1(x_k) + 0.5\tau^2(u_0''(x_k) + f(x_k, 0)), \quad k = \overline{1, N}. \quad (7.8^*)$$

Для розрахунків можна користуватися як формулою (7.8), так і формулою (7.8*).

Від вибору параметра σ залежить точність і стійкість схеми (7.5), тому розглянемо найбільш уживані частинні випадки значень цього параметра.

Явна схема. При $\sigma = 0$ на шаблоні $S = \{(x_{k+1}, t_j), (x_k, t_j), (x_k, t_{j+1})\}$ – "хрест" (на Рис.7.1 він позначений більш «жирними» крапками) схему (7.5) запишемо у вигляді наступної розрахункової формули

$$\begin{aligned} y_k^{j+1} &= 2y_k^j - y_k^{j-1} + \tau^2(y_{xx,k}^j + f), \\ k &= \overline{2, N-1}, \quad j = \overline{1, M-1}. \end{aligned} \quad (7.9)$$

де $\varphi = f \equiv f_{k,j} = f(x_k, t_j)$.

Алгоритм розрахунку:

1. Визначаємо сіткову функцію y_k^0 , $k = \overline{1, N}$ за формулою (7.7).
2. Знаходимо y_k^1 , $k = \overline{1, N}$, використовуючи формулу (7.8).
3. Використовуємо формули (7.6), (7.9) і послідовно для $j = \overline{1, M-1}$ знаходимо сіткову функцію на решті шарів сітки $\bar{\omega}_\tau$.

Неявна схема. При $0 < \sigma \leq \frac{1}{2}$ схему (7.5) запишемо у вигляді:

$$\begin{aligned} \sigma\tilde{\gamma} * y_{k-1}^{j+1} - (1 + 2\sigma\tilde{\gamma}) * y_k^{j+1} + \sigma\tilde{\gamma} * y_{k+1}^{j+1} &= -F_k, \\ k &= \overline{2, N-1}. \end{aligned} \quad (7.10)$$

де $\gamma = \tau/h$, $\tilde{\gamma} = \gamma^2$, а компоненти вектора $\vec{F}$ обчислюються наступним чином

$$F_k = \sigma \tilde{\gamma} * y_{xx,k}^{j-1} - y_k^{j-1} + (1 - 2\sigma) \tilde{\gamma} * y_{xx,k}^j + 2y_k^j + \tau^2 f.$$

Алгоритм розрахунку:

1. Використовуємо описані вище п.1.-2.
2. Для кожного значення $j = \overline{1, M-1}$ маємо СЛАР, яка утворюється формулами (7.6) і (7.10). Ці СЛАР ефективно розв'язуються *монотонною прогонкою*, яка є стійкою.

Зауваження 1. Різницеві задачі для одновимірних нестационарних рівнянь в частинних похідних, які ми розглядали в попередніх лекціях (гіперболічного типу першого порядку, параболічного типу), також приводили до певних алгоритмів знаходження розв'язку на j -шарах сіткової області. Характерною особливістю цих алгоритмів є можливість економного використання оперативної пам'яті комп'ютера. Дійсно, при належній програмній реалізації цих алгоритмів (формально це рекурентний процес 1-го порядку), можна використовували лише один масив для знаходження розв'язку на черговому j -му шарі сіткової області. З опису алгоритмів знаходження числового розв'язку задачі (7.5)-(7.8), як у варіанті явної схеми, так і для неявної схеми, ми маємо рекурентний процес 2-го порядку. А отже програмна реалізація вимагає три масиви і плюс ланцюжок операторів переприсвоєнь для знаходження розв'язку на черговому j -му шарі. Зрозуміло, що коефіцієнт три буде і при реалізації РЗ для багатовимірних КЗ гіперболічного типу другого порядку.

п.1.2 Похибка апроксимації

Позначимо $z = y - u(x_k, t_j)$ – *похибка розв'язку* і підставимо $y = z + u(x_k, t_j)$ в РЗ (7.5)-(7.8), отримаємо для z різницеву задачу

$$\begin{cases} z_{\bar{t}t,j} = \Lambda z + \psi, & k = \overline{2, N-1}, j = \overline{0, M-1}; \\ z_1^j = 0, \quad z_N^j = 0, & j = \overline{1, M}; \\ z_k^0 = 0, & k = \overline{1, N}; \\ z_{t,0}(x_k) = v_k = O(\tau^2), & k = \overline{2, N-1}. \end{cases} \quad (7.11)$$

де ψ – *похибка на розв'язку задачі* (7.1)-(7.4) має вигляд

$$\psi = \Lambda u^{(\sigma)} + \varphi - u_{\bar{t}t,j}. \quad (7.12)$$

Знайдемо її величину в околі точки (x_k, t_j) . Враховуючи, що:

$$\hat{u} = u + \tau * u_{t,j}, \quad \ddot{u} = u - \tau * u_{\bar{t},j}; \quad u^{(\sigma)} = u + \sigma \tau^2 u_{\bar{t}t,j};$$

$$u_{\bar{t}t,j} = \ddot{u} + O(\tau^2); \quad \Lambda u = Lu + \frac{h^2}{12} L^2 u + O(h^4),$$

отримаємо наступний ланцюжок перетворень для ψ :

$$\begin{aligned} \psi &= \Lambda u + \sigma \tau^2 \Lambda u_{\bar{t}t,j} + \varphi - u_{\bar{t}t,j} = Lu + \frac{h^2}{12} L^2 u + \sigma \tau^2 L \ddot{u} + \varphi - \ddot{u} + O(h^4 + \tau^2) \pm f = \\ &= (Lu - \ddot{u} + f) + (\varphi - f + \sigma \tau^2 Lf) + \left(\frac{h^2}{12} + \sigma \tau^2 \right) L^2 u + O(h^4 + \tau^2) = \\ &= (\varphi - f + \sigma \tau^2 Lf) + \left(\frac{h^2}{12} + \sigma \tau^2 \right) L^2 u + O(h^4 + \tau^2). \end{aligned}$$

Аналізуючи отримане співвідношення, маємо *оцінку похибки*

$$\psi = \begin{cases} O(h^2 + \tau^2), & \varphi = f \text{ \& } \forall \sigma \in [0, \frac{1}{2}]; \\ O(h^4 + \tau^2), & \varphi = f - \sigma \tau^2 Lf \text{ \& } \sigma = \sigma_*. \end{cases} \quad (7.13)$$

Схема з $\sigma = \sigma_* \equiv \tilde{\sigma} - \frac{h^2}{12\tau^2}$, $\tilde{\sigma} \in \text{Const}$, $\tilde{\sigma} \geq \frac{1}{4(1-\varepsilon)}$, $\varepsilon > 0$ називається *схемою підвищеного порядку точності*.

За наявності у крайових умовах (7.2) похідних, для отримання необхідного порядку апроксимації можна використовувати як *метод законтурних точок* так і *метод підвищення порядку апроксимації на розв'язку*. Наприклад, крайові умови третього роду :

$$\begin{aligned} -\frac{\partial u(a,t)}{\partial x} + \beta_1 u(a,t) &= \mu_1(t), \\ \frac{\partial u(b,t)}{\partial x} + \beta_2 u(b,t) &= \mu_2(t), \quad t \in [0, T] \end{aligned}$$

апроксимуються такими різницевиими рівняннями [33]

$$\begin{aligned} \theta_1 y_{it,j} &= \Lambda^- Y + \varphi^-, \quad k=1, \\ \theta_2 y_{it,j} &= \Lambda^+ Y + \varphi^+, \quad k=N, \end{aligned}$$

де оператори $\Lambda^- Y$, $\Lambda^+ Y$, а також різницеві функції записані в безіндексній формі і визначаються наступним чином

$$\begin{aligned} \Lambda^- w &= \frac{w_x - \beta_1 w}{\tilde{h}}, \quad \varphi^- = \theta_1 \varphi + \frac{v_1}{\tilde{h}}, \quad \tilde{h} = 0.5h, \\ \Lambda^+ w &= -\frac{w_{\bar{x}} + \beta_2 w}{\tilde{h}}, \quad \varphi^+ = \theta_2 \varphi + \frac{v_2}{\tilde{h}}. \end{aligned}$$

Якщо обрати

$$\varphi = f(x,t), \quad \theta_1 = \theta_2 = 1, \quad v_1 = \mu_1(t), \quad v_2 = \mu_2(t),$$

то похибка апроксимації крайових умов буде $O(\tau^2 + h^2)$.

Якщо обрати

$$\sigma = \sigma_*, \quad \varphi = f - \sigma \tau^2 Lf, \quad \theta_1 = 1 + \frac{h\beta_1}{3}, \quad \theta_2 = 1 + \frac{h\beta_2}{3},$$

$$v_1 = \mu_1(t) + \frac{h^2}{6} (\ddot{\mu}_1(t) + f'(a,t)),$$

$$v_2 = \mu_2(t) + \frac{h^2}{6} (\ddot{\mu}_2(t) - f'(b,t)),$$

то похибка апроксимації крайових умов буде $O(\tau^2 + h^4)$.

п.1.3 Стійкість і збіжність РС

Дослідження стійкості для задачі (7.5)-(7.8) виконаємо методом Фур'є [19, с. 246-247]. Для цього підставимо $y_k^j = \rho_q^j \exp(iq x_k)$, $q = 0, \pm 1, \pm 2, \dots$; $i = \sqrt{-1}$. в одну-рідну схему (7.5) та отримаємо для множника зростання q -ї гармоніки при переході з шару на шар квадратне рівняння

$$(\rho_q)^2 - 2\beta * \rho_q + 1 = 0,$$

де

$$\beta = 1 - \frac{2\theta^2}{1 + 4\sigma\theta^2}, \quad \theta = \frac{\tau}{h} \sin\left(\frac{qh}{2}\right).$$

Стійкість буде у випадку, якщо значення коренів $|(\rho_q)_1| = |(\rho_q)_2| = 1$, а отже отримуємо нерівність $|\beta| \leq 1$, що дає умову *стійкості*

$$\frac{\tau^2}{h^2}(1-4\sigma) \leq 1 \text{ або } \sigma \geq \frac{1+\varepsilon}{2} - \frac{h^2}{4\tau^2} \text{ \& } \forall \varepsilon > 0. \quad (7.14)$$

Із цієї нерівності маємо висновки відносно *стійкості по початковим даним*:

- ✓ явна схема $\sigma = 0$ стійка, при виконанні умови Куранта $\tau \leq h$;
- ✓ схема з $\sigma \in (0, \frac{1}{4})$ також умовно стійка ($\tau \leq h\sqrt{1-4\sigma}$);
- ✓ схема з $\sigma \in [\frac{1}{4}, \frac{1}{2}]$ абсолютно стійка.

Для дослідження стійкості РЗ по правій частині використовують метод розділення змінних і принцип суперпозиції [33]. Якщо $\sigma \geq 0$, то при виконанні наступного співвідношення між кроками сітки $\frac{\tau^2}{h^2} \leq \frac{1}{1+\varepsilon}$, $\varepsilon > 0$ і для сіткової функції з однорідними граничними умовами справедлива умова

$$\|y^j\| \leq \sqrt{\frac{1+\varepsilon}{\varepsilon}} \left(\|y^0\| + \|y_{t,0}\|_{A^{-1}} + \sum_{m=0}^{j-1} \tau \|\varphi_{k,m}\|_{A^{-1}} \right).$$

Негативна норма (норма в $H_{A^{-1}}$) для $A = -\Lambda$ в просторі H – просторі сіткових функцій з $y_1 = 0$, $y_N = 0$, визначається по формулі

$$\|y\|_{A^{-1}} = \sqrt{(A^{-1}y, y)} = \sqrt{\sum_{k=2}^{N-1} \frac{(y_k)^2}{\lambda_k}}, \quad \lambda_k = \frac{4}{h^2} \sin^2 \left(\frac{\pi(k-1)h}{2} \right).$$

Різницева задача (7.5) – (7.8), за умов виконання стійкості, відповідної гладкості вхідних даних і шуканої функції, збігається до розв'язку диференціальної задачі (7.1)-(7.4) у відповідних нормах з порядком, що співпадає з порядком апроксимації (7.13).

Зауваження 2. Розглянута однопараметрична сім'я ($\sigma \in [0, \frac{1}{2}]$) тришарових схем (7.5) є частинним випадком (симетричним $\sigma_1 = \sigma_2 = \sigma$) двопараметричної сім'ї $((\sigma_1, \sigma_2 \geq 0) \& (\sigma_{1,2} \in \mathbb{R}))$ тришарових схем. Для таких схем різницеве рівняння (7.5) форму не змінює, але сіткова функція Y визначається наступним виразом

$$Y \equiv y^{(\sigma_1, \sigma_2)} = \sigma_1 \hat{y} + (1 - \sigma_1 - \sigma_2) y + \sigma_2 \check{y}.$$

Стійкість двопараметричних схем визначається умовою $\sigma_1 \geq \sigma_2 \text{ \& } \sigma_1 + \sigma_2 > \frac{1}{2} - \frac{h^2}{2\tau^2}$.

п.2 РС для багатовимірних рівнянь гіперболічного типу другого порядку

Розглянемо крайову задачу для p -вимірного ($p \geq 2$) рівняння гіперболічного типу:

$$\frac{\partial^2 u(x, t)}{\partial t^2} = Lu(x, t) + f(x, t), \quad (x, t) \in G \times (0, T], \quad x \equiv (x_1, x_2, \dots, x_p), \quad (7.15)$$

$$L = \sum_{\alpha}^p L_{\alpha}, \quad L_{\alpha} u = \frac{\partial^2 u}{\partial x_{\alpha}^2};$$

$$u(x, t) = \mu(x, t), \quad x \in \partial G; \quad (7.16)$$

$$u(x, 0) = u_0(x), \quad x \in \bar{G}; \quad (7.17)$$

$$\frac{\partial u(x, 0)}{\partial t} = u_1(x), \quad x \in \bar{G}. \quad (7.18)$$

Якщо від чисельних розрахунків задачі (7.15)-(7.18) не вимагається особливо високої надійності, то можна використати для апроксимації (7.15) *явну схему*

$$y^{j+1} = 2y^j - y^{j-1} + \tau^2 \left(\sum_{\alpha=1}^p \Lambda_{\alpha} y + f(x, t_j) \right), \quad x \in \omega_h, \quad j = \overline{1, M-1}, \quad (7.19)$$

яка має порядок апроксимації $O\left(\tau^2 + \sum_{\alpha=1}^p h_{\alpha}^2\right)$.

Методом Фур'є можна дослідити, що ця схема стійка при виконанні умови Куранта

$$\tau < \sqrt{\sum_{\alpha=1}^p \frac{1}{h_{\alpha}^2}} \sim \frac{h}{\sqrt{p}}.$$

У випадку, коли умова Куранта стає обтяжливою при чисельних розрахунках задачі (7.15)-(7.18), використовують безумовно стійкі економічні схеми. Для рівнянь гіперболічного типу *локально-одновимірні схеми* мають більш громіздкий і штучний вигляд у порівнянні з ЛОС для параболічних рівнянь [33]. Більш зручні в цьому випадку *факторизовані схеми (схеми з розщепленням)*. Розглянемо послідовність кроків побудови факторизованої схеми.

1-й етап Спочатку будують базову схему (безумовно стійку схему з вагою $\frac{1}{4} \leq \sigma \leq \frac{1}{2}$):

$$\frac{1}{\tau^2} (y^{j+1} - 2y^j + y^{j-1}) = \sum_{\alpha=1}^p \Lambda_{\alpha} (\sigma y^{j+1} + (1-2\sigma)y^j + \sigma y^{j-1}) + f(x, t_j),$$

$$x \in \omega_h, \quad j = \overline{1, M-1},$$

яка має порядок $O\left(\tau^2 + \sum_{\alpha=1}^p h_{\alpha}^2\right)$ при всіх значеннях параметра σ .

2-й етап Базову схему перепишемо в вигляді схеми (7.19) з операторними коефіцієнтами

$$A y^{j+1} = F(x, t_j, t_{j-1}), \quad x \in \omega_h, \quad j = \overline{1, M-1},$$

де

$$A = E - \tau^2 \sigma \sum_{\alpha=1}^p \Lambda_{\alpha}, \quad B = 2E + \tau^2 (1-2\sigma) \sum_{\alpha=1}^p \Lambda_{\alpha}, \quad C = -A;$$

$$F(x, t_j, t_{j-1}) = B y^j + C y^{j-1} + \tau^2 f(x, t_j)$$

3-й етап Перетворимо (замінімо) оператор A на деякий близький до нього (по порядку апроксимації) оператор $\tilde{A}$, але простіший для знаходження оберненого оператора (ми уже зустрічались з такою ситуацією, згадайте позмінно-трикутний метод). Введемо оператор $\tilde{A}$ наступним чином

$$\tilde{A} \equiv \prod_{\alpha=1}^p (E - \tau^2 \sigma \Lambda_{\alpha}) = E - \tau^2 \sigma \sum_{\alpha=1}^p \Lambda_{\alpha} + \tau^4 \sigma^2 \sum_{\alpha=1}^{p-1} \sum_{\beta=1+\alpha}^p \Lambda_{\alpha} \Lambda_{\beta} + \dots = A + O(\tau^4).$$

В результаті, отримуємо факторизовану схему:

$$\tilde{A} y^{j+1} = F(x, t_j, t_{j-1}), \quad x \in \omega_h, \quad j = \overline{1, M-1}. \quad (7.20)$$

З урахуванням порядку апроксимації базової схеми і того, що схема (7.20) відрізняється від базової на порядок $O(\tau^4)$, маємо для факторизованої схеми такий же порядок апроксимації $O\left(\tau^2 + \sum_{\alpha=1}^p h_{\alpha}^2\right)$.

Стійкість схеми (7.20) по початковим даним досліджуємо методом Фур'є [33]. Для цього в однорідне рівняння (7.20) підставимо багатовимірну гармоніку

$$y^j = \rho^j \exp\left(i \sum_{\alpha} q_{\alpha} x_{k_{\alpha}}\right), \quad q_{\alpha} = 0, \pm 1, \pm 2, \dots; \quad i = \sqrt{-1},$$

виконаємо деякі обчислення і отримаємо квадратне рівняння, відносно ρ

$$a\rho^2 - 2b\rho + c = 0,$$

де

$$a = \prod_{\alpha} (1 + 2\sigma\theta_{\alpha}), \quad b = 1 - (1 - 2\sigma) \sum_{\alpha} \theta_{\alpha}, \quad c = 1 + 2\sigma \sum_{\alpha} \theta_{\alpha}, \quad \theta_{\alpha} = 2 \left(\frac{\tau}{h_{\alpha}} \sin\left(\frac{q_{\alpha} h_{\alpha}}{2}\right) \right)^2, \\ a \geq 1, \quad c \geq 1, \quad \theta_{\alpha} \geq 0.$$

Нас цікавлять тільки ті розв'язки цього рівняння, які $|(\rho)_{1,2}| < 1$, а це можливо тоді і тільки тоді, коли виконується умова $(c \leq a) \& (2|b| \leq a + c)$. Враховуючи, що $a > c$ (безпосередньо перевіряється), отримуємо нерівність $|b| \leq a$, яка виконується при умові

$$\tau^2 (1 - 4\sigma) \sum_{\alpha=1}^p \frac{1}{h_{\alpha}^2} \leq 1. \quad (7.21)$$

Це і є *достатня умова стійкості* факторизованої схеми (7.20). З умови (7.21) видно, що якщо вагу σ схеми вибирати в діапазоні $[\frac{1}{4}, \frac{1}{2}]$, то ця нерівність виконується при довільних кроках h_{α} , τ , а отже факторизована схема є *безумовно стійка*.

Збіжність схеми (7.20) слідує із апроксимації і безумовної стійкості.

Алгоритм розрахунку по факторизованій схемі близький до схеми реалізації алгоритма одновимірних задач – обчислюємо сіткову функцію для шарів $j=0$ і $j=1$, а після цього, для кожного значення $j=1, \overline{M-1}$ маємо послідовність (СЛАР) $_{\alpha}$, $\alpha=1, \overline{p}$. Враховуючи вид оператора $\tilde{A}$ в схемі (7.20), кожна з СЛАР по напрямку x_{α} ефективно розв'язується *монотонною прогонкою*, яка є стійка. Кількість операцій пропорційна кількості вузлів просторової сітки, а отже факторизована схема є *економічною*.

Практичне заняття № 01

ТЕМА № 1. Числове інтегрування (2 год).

Теоретична інформація:

Необхідно наближено обчислити визначений інтеграл

$$I = \int_a^b f(x)dx \quad \text{або} \quad I = \int_a^b \rho(x)u(x)dx, \quad (1.1)$$

де $\rho(x) > 0$ – вагова функція неперервна на (a,b) , $f(x)|u(x)$ – підінтегральні функції достатньої гладкості на $[a,b]$.

Більшість методів чисельного інтегрування ґрунтуються на заміні (1.1) скінченною сумою, т.з. *квадратурною формулою* (КФ)

$$I_n = \sum_{k=1}^n C_k f(x_k) \quad \text{або} \quad I_n = \sum_{k=1}^n C_k u(x_k), \quad (1.2)$$

де C_k – *квадратурні коефіцієнти (ваги)*, x_k – *вузли КФ*.

Озн.1 Якщо граничні точки $x=a, b$ входять в систему вузлів КФ, то формули (1.2) називаються *закритого типу*, інакше – *відкритого типу*.

Озн.2 КФ називається *чисельно стійкою*, якщо всі C_k невід’ємні.

Озн.3 *Похибкою* КФ називається величина $|I - I_n| = R_n$. Вона залежить від вибору $\{C_k\}, \{x_k\}$, гладкості функції $f|u$.

Формули (1.2) в загальному містять $2n$ параметрів, тому є можливість будувати різні КФ, що базуються на певній стратегії:

- різні методи заміни підінтегральної функції $f|u$, як на всьому інтервалі $[a,b]$, так і на його складових $[x_{k-1}, x_k]$ (*поліноміальна апроксимація, сплайни*);
- фіксація вибраного набору параметрів $\{x_k\}$ або $\{C_k\}$ з подальшим визначенням решти параметрів, що задовольняють певним додатковим умовам (*КФ Маркова, Чебишова*);
- забезпечення найменшої оцінки залишку R_n для всіх функції $u(x)$ з деякого класу U (*оптимальні КФ на класі функцій із $W_2^1(L)$ точних на сплайн-функціях першого порядку*);
- забезпечення *точного* ($R_n = 0$) обчислення (1.1) формулою (1.2) для всіх функції $u(x)$ з деякого класу U (*КФ найвищого алгебраїчного степеня точності – Гаусса-Крістоффеля*).

Найпростіші формули чисельного інтегрування отримують, обираючи на $[a,b]$ рівномірну сітку $\omega_h = \{x_k = a + (k-1)h, k = \overline{1, n}, h = (b-a)/(n-1); \Delta_k = [x_{k-1}, x_k]\}$, користуючись адитивністю (1.1) і замінюючи підінтегральну функцію поліномом

$$I = \int_a^b f(x)dx = \sum_{k=2}^n \int_{x_{k-1}}^{x_k} f(x)dx, \quad (1.3)$$

Таблиця 1.1 Найпростіші КФ.

Назва	Обчислення $I_n(k) = \int_{x_{k-1}}^{x_k} f(x)dx$	Складена КФ $I_n = \sum_{k=2}^n I_n(k)$	Похибка на $\Delta_k, [a,b]$
1. КФ <i>прямокутників</i> а) <i>лівих</i>	hf_{k-1}	$h \sum_{k=1}^{n-1} f_k$	$\frac{h^2 M_{1,k}}{2}, O(h^1)$
б) <i>правих</i>	hf_k	$h \sum_{k=2}^n f_k$	$\frac{h^2 M_{1,k}}{2}, O(h^1)$

г) центральних	$hf_{k-0.5},$ $x_{k-0.5} = 0.5(x_{k-1} + x_k)$	$h \sum_{k=2}^n f_{k-0.5}$	$\frac{h^3 M_{2,k}}{24}, O(h^2)$
2. КФ трапецій	$0.5h(f_{k-1} + f_k)$	$h \left(0.5(f_1 + f_n) + \sum_{k=2}^{n-1} f_k \right)$	$\frac{h^3 M_{2,k}}{12}, O(h^2)$
3. КФ Сімпсона	$\frac{0.5h}{3}(f_{k-1} + 4f_{k-0.5} + f_k)$	$\frac{h}{3} \left(f_1 + f_n + 4 \sum_{k=2,4,\dots}^{n-1} f_k + 2 \sum_{k=3,5,\dots}^{n-2} f_k \right)$	$\frac{h^5 M_{4,k}}{90}, O(h^4)$

Для отримання значення (1.1) з заданою точністю $0 < \varepsilon < 1$ використовують адаптивні алгоритми (як рекурентні так і рекурсивні). Ці алгоритми будуються на апостеріорній оцінці похибки R_n за допомогою першої формули Рунге-Ромберга.

КФ Гаусса для чисельного інтегрування (1.1) з $\rho(x) \equiv 1$ і $[a, b] = [-1, 1]$ отримують, обираючи $\{C_k\}, \{\gamma_k\}$ так, щоб формула (1.2) була точна для будь-якого полінома степеня не вище $2n - 1$. У загальному випадку, коли $[a, b] \neq [-1, 1]$, то відомі вузли ξ_i і ваги γ_i формули Гаусса на $[-1, 1]$, наприклад

$$n=1: \quad \xi_1 = 0; \quad \gamma_1 = 2.$$

$$n=2: \quad -\xi_1 = \xi_2 = \sqrt{1/3}; \quad \gamma_1 = \gamma_2 = 1.$$

$$n=3: \quad -\xi_1 = \xi_3 = \sqrt{3/5}, \xi_2 = 0; \quad \gamma_1 = \gamma_3 = 5/9, \gamma_2 = 8/9.$$

попередньо перетворюють:

$$x_i = 0.5(a + b) + 0.5(b - a)\xi_i, \quad C_i = 0.5(b - a)\gamma_i, \quad i = \overline{1, n}.$$

Формула Гаусса розрахована на функції $u(x)$, які мають достатньо високі похідні, причому не дуже великі по абсолютній величині.

КФ Гаусса-Крістоффеля залежать від діапазону $[a, b]$ і вагової функції $\rho(x)$.

Таблиця 1.2.

$\rho(x)$	$[a, b]$	Ортогональний многочлен	Назва КФ
1	$[-1, 1]$	Лежандра	Гаусса
$1/\sqrt{1-x^2}$	$[-1, 1]$	Чебишова 1-го роду	Гаусса-Чебишова
$(1-x)^\alpha (1+x)^\beta, \alpha, \beta > -1$	$[-1, 1]$	Якобі	Якобі
$x^\alpha \exp(-x), \alpha > -1$	$[0, \infty)$	Чебишова-Лагерра	Лагерра
$\exp(-x^2)$	$(-\infty, \infty)$	Чебишова-Ерміта	Ерміта

і мають вид :

$$I \approx \sum_{k=1}^n (\gamma_k * u(\xi_k)),$$

де γ_k – вагові коефіцієнти, а ξ_k – вузли, які є коренями відповідних ортогональних поліномів і для найбільш уживаних вагових функцій $\rho(x)$ уже обчислені і можна скористатись відповідними довідниками.

Збіжність КФ. Узагальнені формули, наведені в Табл.1.1 і формули Гаусса-Крістоффеля при $n \rightarrow \infty$ для довільної неперервної функції є збіжними. Якщо $R_n = O(h^p)$, то ми маємо формулу збіжності з p -порядком точності. Оцінка точності (див. Табл.1.1) прийнятна, якщо функція має неперервну (кусово-неперервну) похідну, яка входить в оцінку залишкового члена. Наявність більш високих похідних не покращує цю оцінку, а от відсутність відповідної обмеженої похідної призводить до погіршення збіжності [15–17, 21, 23]:

Таблиця 1.3 Мажорантні оцінки похибки КФ для $u(x) \in C^{(p)}[L], \max_{x \in L} |u^{(p)}(x)| \leq M_p$.

Формула	p				
	1	2	3	4	Багато
Трапецій	$\frac{L}{4} h M_1$	$\frac{L}{12} h^2 M_2$	→	→	→
Центральних прямокутників	$\frac{L}{4} h M_1$	$\frac{L}{24} h^2 M_2$	→	→	→
Сімпсона	$\frac{5L}{18} h M_1$	$\frac{4L}{81} h^2 M_2$	$\frac{L}{72} h^3 M_3$	$\frac{L}{180} h^4 M_4$	→
Гаусса $n=4, L=b-a=2$	$0.276 M_1$	$0.022 M_2$	$0.0024 M_3$	$0.0003 M_4$	$10^{-8} M_8$

З цієї таблиці видно, що для функцій, які мають тільки першу або другу похідну, кращі результати має формула центральних прямокутників. Для функцій високої гладкості вигідні формули Гаусса.

Короткий огляд складніших ситуацій числового інтегрування

Наявність розривів у $f(x)$: розбиваємо $[a, b]$ на відрізки Δ_c , де $f(x), f^{(m)}(x)$ неперервні, використовуємо адитивність інтеграла, застосовуємо на кожному відрізку Δ_c КФ, наприклад, середніх (при невеликих m) або Сімпсона чи Гаусса ($m \geq 4$).

Швидко осцилюючі $f(x)$: представляємо $f(x) = u(x) \exp(i\omega * x)$, де частота ω відома, а амплітуда $u(x)$ мало змінюється за період основного коливання, далі використовуємо КФ Філона або КФ на базі сплайнів.

Змінна верхня границя $F(x) = \int_a^x f(x) dx$: для поодиноких обчислень можна використати КФ середніх, Сімпсона, а для значень $x \in \vec{X}$, де $\vec{X}$ – впорядкований за зростанням масив, використовуємо адитивність $F(x) = F(X_N) + \int_{X_N}^x f(x) dx, X_N \leq x \leq X_{N+1}$, де для обчислення інтегралу використовуємо звичайні КФ.

Невласні інтеграли: є декілька варіантів, зокрема використання КФ Лагерра і Ерміта для обчислення інтегралів на півосі $[0, \infty)$ і осі $(-\infty, \infty)$ відповідно.

m -кратні інтеграли: для кратності $m = \{2, 3\}$ використовують послідовне інтегрування, метод комірок, а для великих значень кратності ($m > 4$) метод Монте-Карло.

Аудиторне завдання:

п.1. Використовуючи КФ з указаним n і інформацію подану в Таблиці 1.3, обчислити $\int_0^1 \sin(\pi * x) dx$ і заповнити таблицю:

КФ	Наближене значення	Абсолютна похибка	Теоретична оцінка похибки
центр.прямокутників $n=7$			
трапецій $n=7$			
Сімпсона $n=7$			
Гаусса $n=4$			

Напишемо відповідну М-функцію *p01a01*:

```
function P01a01
clear all, close all, clc

n = 7; a = 0; b = 1; ng = 4;
L = b-a; h = L./(n-1);
I = int(sym('sin(pi*x)'), sym('x'), a, b); I = eval(I);
% Обчислення КФ :
Cr = i_crectn(a, b, @f, n); % центр.прямокутників
```

```

T = i_trapn(a, b, @f, n); % трапецій
S = i_simpn(a, b, @f, n); % Сімпсона
G = i_gauss(a, b, @f, ng); % Гаусса

h2 = h.*h; h4 = h2.*h2;
M2 = pi.*pi; M4 = M2.*M2; p = L.*h2.*M2./12;
% Друк таблиці результатів
fprintf('%12s%19.12f\n', 'Точне знач.', I);
fprintf('%12s %12s %12s %12s\n', ...
        'Назва КФ', 'Наближене значення', ...
        'Абсолютна похибка', 'Теор. оцінк. похибки');
fr = '%12s%19.12f%19.10e%19.10e\n';
NP = sprintf('n=%u', n); NG = sprintf('n=%u', ng);
fprintf(fr, ['центр.пр ', NP], Cr, abs(I-Cr), p./2);
fprintf(fr, ['трапецій ', NP], T, abs(I-T), p);
fprintf(fr, ['Сімпсона ', NP], S, abs(I-S), L.*h4.*M4./180);
fprintf(fr, ['Гаусса ', NG], G, abs(I-G), 1e-8.*M4.*M4);
end

function y = f(x)
    y = sin(pi.*x);
end

```

Запустимо її на виконання і отримаємо наступний результат :

```

Точне знач.=      0.636619772368
Назва КФ Наближене значення Абсолютна похибка Теор. оцінк. похибки
центр.пр n=7      0.643950550859  7.3307784918e-003  1.1423153242e-002
трапецій n=7      0.622008467928  1.4611304439e-002  2.2846306484e-002
Сімпсона n=7      0.636894534174  2.7476180674e-004  4.1756297597e-004
Гаусса  n=4       0.636614752130  5.0202378321e-006  9.4885310161e-005

```

Завдання для самостійної роботи:

п.1. Виконати табуляцію функції $f(t)$ з рівномірним кроком, а кількість інтервалів розбиття обрати 10 :

Var(k,5)	$f(t)$	Інтервал зміни t	Параметри
1	$a \int_0^t \sqrt{1 - \omega^2 \sin^2(t)} dt$	$[0, \pi/2]$	$a = k + 4, b = g, \omega = \sqrt{1 - (b/a)^2}$
2	$a \int_0^t \sqrt{1 + \omega^2 \sin^2(t)} dt$	$[0, \pi/2]$	$a = g + 2, b = k, \omega = \sqrt{1 + (b/a)^2}$
3	$(a + b) \int_0^t \sqrt{1 - \omega \sin^2(t/2)} dt$	$[0, \pi/2]$	$a = k, b = k + g, \omega = 4ab(a + b)^{-2}$
4	$a \sqrt{2} \int_0^t \cos^{-0.5}(2t) dt$	$[0, \pi/3]$	$a = (k + g)/10$
5	$a \sqrt{2} \int_0^t \sqrt{1 - \cos(t)} dt$	$[0, \pi]$	$a = (k + g)/10$

Обчислення інтеграла виконати з точністю $eps = 10^{-8}$ і використати узагальнені КФ

Var(d,3)	КФ
1	центр.прямокутників
2	трапецій
3	Сімпсона

Тут (і надалі) позначено :

- g – номер групи,
- k – номер за списком студента у групі,
- d – день народження студента.

Функція $Var(J,p)$ – описує циклічність J з періодом p , і визначається наступним чином

$$Var(J, p) = \begin{cases} i, & i \neq 0, \\ p, & i = 0; \end{cases}, \text{ де } i = \text{mod}(J, p).$$

п.2. Знайти гама-функцію $\Gamma(x) = \int_0^\infty e^{-t} t^{x-1} dt$ для значення

$$x = \frac{1}{k+g} + \begin{cases} 1.0, & Var(d,4)=1; \\ 2.7, & Var(d,4)=2; \\ 3.4, & Var(d,4)=3; \\ 5.0, & Var(d,4)=4. \end{cases}$$

Використати КФ Лагерра ($\rho(t)=e^{-t}$, $u(t)=t^{x-1}$) з кількістю вузлів $n \in \{2, 3, 4, 5\}$ і порівняти зі значенням, отриманим за допомогою вбудованої спеціальної функції $\text{gamma}(x)$ MATLAB. Результати представити у наступній формі, наприклад :

```

Г(1.000)=      1.00000
КФ Лагерра
вузлів Наближ.знач абсол.похибка
  2      1.00000  0.00000e+000
  3      1.00000  3.00000e-007
  4      1.00000  1.95000e-007
  5      1.00000  5.30000e-007

```

п.3. Обчислити значення $S = \int_a^b \left(\int_{\alpha(x)}^{\beta(x)} f(x, y) dy \right) dx$,

де

$Var(g,5)$	$[a,b]$	$\alpha(x)$	$\beta(x)$	$f(x, y)$
1	[0,1]	$1-x$	$\sqrt{(x+1)(1-x)}$	$\sin(x-y^2)$
2	[1,2]	$\sqrt{x-1}$	$x^2-0.5$	$\sqrt{x^3-y^2}$
3	[-1,0]	$x-2$	$\sin(2-2\pi * x)$	$\cos(xy^2)$
4	[-2,-1]	$1+x$	$2+x^2$	$\ln(x^2+y)$
5	[0,2]	$(1+x)/2$	$\sqrt{x^4+1}$	$\sin(x^2+\ln(y))$

Для знаходження двократного інтегралу застосувати метод послідовного інтегрування, використати КФ з $n=7$ і організувати обчислення наступним чином :

Інтеграл по x		Інтеграл по y	
КФ Гаусса	$Var(k,3)=1$	КФ Гаусса	$Var(d,3)=1$
КФ Чебишова	$Var(k,3)=2$	КФ Чебишова	$Var(d,3)=2$
КФ Маркова	$Var(k,3)=3$	КФ Маркова	$Var(d,3)=3$

Програмне забезпечення на MATLAB:

```

function    s = i_trap(a, b, f, ep, mki)
% Знаходження з точністю eps значення визначеного інтеграла
%
%
%
%

$$s = \int_a^b f(x) \, dx$$

%
% з допомогою квадратурної формули методу трапецій.
% Вхідні дані :
% a, b - проміжок інтегрування;
% f - функція з описом f(x);
% ep - точність (0 < ep < 1);
% mki - максимальна кількість згущення сітки.
% Вихідні дані :
% s - значення інтегралу.
A = a; B = b; z = 1;
if a < b
    A = b; B = a; z = -1;
end
k = 0; h = B - A;
s0 = 0.5.*(f(A) + f(B)); s1 = 0; s = h.*s0;
F = 0;
while ~F & k < mki
    k = k + 1; ho = h; h = 0.5.*h;
    x = A + h;
    while x < B
        s1 = s1 + f(x); x = x + ho;
    end
    ss = h.*(s0 + s1);
    F = abs(ss - s)./3 < ep; s = ss;
end
s = z.*s;
return

```

```

function    s = i_simp(a, b, f, ep, mki)
% Знаходження з точністю eps значення визначеного інтеграла
%
%
%
%

$$s = \int_a^b f(x) \, dx$$

%
% з допомогою квадратурної формули методу Сімпсона.
% Використовується адаптивний рекурентний алгоритм.
% Вхідні дані :
% a, b - проміжок інтегрування;
% f - функція з описом f(x);
% ep - точність (0 < ep < 1);
% mki - максимальна кількість згущення сітки.
% Вихідні дані :
% s - значення інтегралу.
A = a; B = b; z = 1;
if a < b
    A = b; B = a; z = -1;
end
k = 0; h = B - A; h = 0.5.*h;
s1 = f(A) + f(B); s2 = 0; s4 = f(0.5.*(B+A));
s = h.*(s1 + 4.*s4)./3;
F = 0;
while ~F & k < mki
    k = k+1; ho = h; h = 0.5.*h;
    s2 = s2 + s4;
    s4 = 0; x = A + h;

```

```

while x < B
    s4 = s4 + f(x); x = x + ho;
end
ss = h.*(s1 + 4.*s4 + 2.*s2)./3;
F = abs(ss - s)./15 < ep; s = ss;
end
s = z.*s;
return

```

```

function s = i_gauss(a, b, f, n)
% Знаходження значення визначеного інтеграла
%
% 
$$s = \int_a^b f(x) dx$$

%
% з допомогою квадратурної формули Гаусса.
% Вхідні дані :
% a, b - проміжок інтегрування;
% f - функція з описом f(x);
% n - кількість вузлів (0<n<8).
% Вихідні дані :
% s - значення інтегралу.
s = 0;
if a > b
    error('I_GAUSS:error a>b.', 'a>b. See I_GAUSS. ');
    return
end
if n < 1 | n > 7
    error('I_GAUSS:error n<1 or n>7.', 'See I_GAUSS. ');
    return
end
% в залежності від n визначаємо вузли і вагові коефіцієнти :
X=zeros(1,n); C=ones(1,n);
switch n
    case 1
    case 2
        X(2) = 0.5773502691896258; X(1) = - X(2); C = [0.5, 0.5];
    case 3
        X(3) = 0.7745966692414834; X(1)= - X(3); C = [5, 8, 5]./ 18;
    case 4
        X(4) = 0.8611363115940492; X(1) = - X(4);
        X(3) = 0.3399810435848646; X(2) = - X(3);
        C(4) = 0.1739274225687284; C(1) = C(4);
        C(3) = 0.3260725774312716; C(2) = C(3);
    case 5
        X(5) = 0.9061798459386640; X(1) = -X(5);
        X(4) = 0.5384693101056830; X(2) = - X(4);
        C(5) = 0.1184634425280945; C(1) = C(5);
        C(4) = 0.2393143352496832; C(2) = C(4); C(3) = 64 ./ 225;
    case 6
        X(6) = 0.9324695142031520; X(1) = - X(6);
        X(5) = 0.6612093864662644; X(2) = - X(5);
        X(4) = 0.2386191860831970; X(3) = - X(4);
        C(6) = 0.0856622461895852; C(1) = C(6);
        C(5) = 0.1803807865240693; C(2) = C(5);
        C(4) = 0.2339569672863455; C(3) = C(4);
    case 7
        X(7) = 0.9491079123427596; X(1) = - X(7);

```

```

        X(6) = 0.7415311855993944; X(2) = - X(6);
        X(5) = 0.4058451513773970; X(3) = - X(5);
        C(7) = 0.0647424830844348; C(1) = C(7);
        C(6) = 0.1398526957446384; C(2) = C(6);
        C(5) = 0.1909150252525595; C(3) = C(5); C(4) = 256 ./ 1225;
    end
    C = 2 .* C;
    % якщо [a,b]<>[-1,1], то виконуємо лінійне перетворення :
    if a ~= -1 | b ~= 1
        ab = 0.5 .* (a + b); ba = 0.5 .* (b - a);
        for i = 1 : n
            X(i) = ab + ba .* X(i); C(i) = ba .* C(i);
        end
    end
    % наближене значення інтеграла :
    for i = 1:n
        s = s + C(i) .* f(X(i));
    end
    return

```

```

function s = i_cheb(a, b, f, n)
% Знаходження значення визначеного інтеграла
%
% 
$$s = \int_a^b f(x) dx$$

%
% з допомогою квадратурної формули Чебишева.
% Вхідні дані :
% a, b - проміжок інтегрування;
% f - функція з описом f(x);
% n - кількість вузлів n є [1,7] або n=9.
% Вихідні дані :
% s - значення інтегралу.
s=0;
if a > b
    error('I_CHEB:error a>b.', 'a>b. See I_CHEB. ');
    return
end
if n < 1 | n == 8 | n > 10
    error('I_CHEB:error n<1 or n=8 or n>10. ');
    return
end
% в залежності від n визначаємо вузли і вагові коефіцієнти :
X = zeros(1, n); C = 2 ./ n;
switch n
    case 1
    case 2
        X(2) = 0.5773502691; X(1) = - X(2);
    case 3
        X(3) = 0.7071067812; X(1) = - X(3);
    case 4
        X(4) = 0.7946544723; X(1) = - X(4);
        X(3) = 0.1875924741; X(2) = - X(3);
    case 5
        X(5) = 0.8324974870; X(1) = - X(5);
        X(4) = 0.3745414096; X(2) = - X(4);
    case 6
        X(6) = 0.8662468181; X(1) = - X(6);

```

```

        X(5) = 0.4225186538; X(2) = - X(5);
        X(4) = 0.2666354015; X(3) = - X(4);
    case 7
        X(7) = 0.8838617008; X(1) = - X(7);
        X(6) = 0.5296567753; X(2) = - X(6);
        X(5) = 0.3239118105; X(3) = - X(5);
    case 9
        X(9) = 0.9115893007; X(1) = - X(9);
        X(8) = 0.6010186554; X(2) = - X(8);
        X(7) = 0.5287617831; X(3) = - X(7);
        X(6) = 0.1679061842; X(4) = - X(6);
end
% якщо [a,b]<>[-1,1], то виконуємо лінійне перетворення :
if a ~= -1 | b ~= 1
    ab = 0.5 .* (a + b); ba = 0.5 .* (b - a);
    X = ab + ba .* X; C = ba .* C;
end
% наближене значення інтеграла :
for i = 1:n
    s = s + f(X(i));
end
s = C .* s;
return

function s = i_markov(a, b, f, n)
% Знаходження значення визначеного інтеграла
%
% 
$$s = \int_a^b f(x) dx$$

%
% з допомогою квадратурної формули Маркова.
% Вхідні дані :
% a, b - проміжок інтегрування;
% f - функція з описом f(x);
% n - кількість вузлів (n є [2,10]).
% Вихідні дані :
% s - значення інтегралу.

s=0;
if a > b
    error('I_MARKOV:error a>b.','a>b. See I_MARKOV.');
```

```

    return
end
if n < 2 | n > 10
    error('I_MARKOV:error n<2 or n>10.');
```

```

    return
end
% в залежності від n визначаємо вузли і вагові коефіцієнти :
X = linspace(-1, 1, n); C = ones(1, n);
switch n
    case 2
        C = [1, 1] ./ 2;
    case 3
        C = [1, 4, 1] ./ 3;
    case 4
        X(3) = 1./5; X(2) = - X(3); C = [1, 5, 5, 1] ./ 6;
    case 5
        X(4) = 3 ./ 7; X(2) = - X(4); C = [9, 49, 64, 49, 9] ./ 90;
    case 6
        X(5) = 0.4225186538; X(2) = - X(5);
        X(4) = 0.2666354015; X(3) = - X(4);
    case 7
        X(7) = 0.8838617008; X(1) = - X(7);
        X(6) = 0.5296567753; X(2) = - X(6);
        X(5) = 0.3239118105; X(3) = - X(5);
    case 9
        X(9) = 0.9115893007; X(1) = - X(9);
        X(8) = 0.6010186554; X(2) = - X(8);
        X(7) = 0.5287617831; X(3) = - X(7);
        X(6) = 0.1679061842; X(4) = - X(6);
end
% якщо [a,b]<>[-1,1], то виконуємо лінійне перетворення :
if a ~= -1 | b ~= 1
    ab = 0.5 .* (a + b); ba = 0.5 .* (b - a);
    X = ab + ba .* X; C = ba .* C;
end
% наближене значення інтеграла :
for i = 1:n
    s = s + f(X(i));
end
s = C .* s;
return

function s = i_markov(a, b, f, n)
% Знаходження значення визначеного інтеграла
%
% 
$$s = \int_a^b f(x) dx$$

%
% з допомогою квадратурної формули Маркова.
% Вхідні дані :
% a, b - проміжок інтегрування;
% f - функція з описом f(x);
% n - кількість вузлів (n є [2,10]).
% Вихідні дані :
% s - значення інтегралу.

s=0;
if a > b
    error('I_MARKOV:error a>b.','a>b. See I_MARKOV.');
```

```

    return
end
if n < 2 | n > 10
    error('I_MARKOV:error n<2 or n>10.');
```

```

    return
end
% в залежності від n визначаємо вузли і вагові коефіцієнти :
X = linspace(-1, 1, n); C = ones(1, n);
switch n
    case 2
        C = [1, 1] ./ 2;
    case 3
        C = [1, 4, 1] ./ 3;
    case 4
        X(3) = 1./5; X(2) = - X(3); C = [1, 5, 5, 1] ./ 6;
    case 5
        X(4) = 3 ./ 7; X(2) = - X(4); C = [9, 49, 64, 49, 9] ./ 90;
    case 6
        X(5) = 0.4225186538; X(2) = - X(5);
        X(4) = 0.2666354015; X(3) = - X(4);
    case 7
        X(7) = 0.8838617008; X(1) = - X(7);
        X(6) = 0.5296567753; X(2) = - X(6);
        X(5) = 0.3239118105; X(3) = - X(5);
    case 9
        X(9) = 0.9115893007; X(1) = - X(9);
        X(8) = 0.6010186554; X(2) = - X(8);
        X(7) = 0.5287617831; X(3) = - X(7);
        X(6) = 0.1679061842; X(4) = - X(6);
end
% якщо [a,b]<>[-1,1], то виконуємо лінійне перетворення :
if a ~= -1 | b ~= 1
    ab = 0.5 .* (a + b); ba = 0.5 .* (b - a);
    X = ab + ba .* X; C = ba .* C;
end
% наближене значення інтеграла :
for i = 1:n
    s = s + f(X(i));
end
s = C .* s;
return

```

```

X(5) = 0.76505532; X(2) = - X(5);
X(4) = 0.28523152; X(3) = - X(4);
C(6) = 0.06666667; C(1) = C(6);
C(5) = 0.37847496; C(2) = C(5);
C(4) = 0.55485837; C(3) = C(4);
case 7
X(6) = 0.83022390; X(2) = - X(6);
X(5) = 0.46884879; X(3) = - X(5);
C(7) = 0.04761905; C(1) = C(7);
C(6) = 0.27682605; C(2) = C(6);
C(5) = 0.43174538; C(3) = C(5); C(4)=0.48761905;
case 8
X(7) = 0.87174015; X(2) = - X(7);
X(6) = 0.59170018; X(3) = - X(6);
X(5) = 0.20929922; X(2) = - X(5);
C(8) = 0.03571429; C(1) = C(8);
C(7) = 0.21070423; C(2) = C(7);
C(6) = 0.34112268; C(3) = C(6);
C(5) = 0.41245881; C(4) = C(5);
case 9
X(8) = 0.83022390; X(2) = - X(8);
X(7) = 0.83022390; X(3) = - X(7);
X(6) = 0.83022390; X(4) = - X(6);
C(9) = 0.02777778; C(1) = C(9);
C(8) = 0.16549536; C(2) = C(8);
C(7) = 0.27453872; C(3) = C(7);
C(6) = 0.34642851; C(4) = C(6); C(5) = 0.37151927;
case 10
X(9) = 0.91953391; X(2) = - X(9);
X(8) = 0.73877386; X(3) = - X(8);
X(7) = 0.47792495; X(4) = - X(7);
X(6) = 0.16527896; X(5) = - X(6);
C(10) = 0.02222222; C(1) = C(10);
C(9) = 0.13330599; C(2) = C(9);
C(8) = 0.22488934; C(3) = C(8);
C(7) = 0.29204268; C(4) = C(7);
C(6) = 0.32753976; C(5) = C(6);
end
% якщо [a,b]<>[-1,1], то виконуємо лінійне перетворення :
if a ~= -1 | b ~= 1
    ab = 0.5 .* (a + b); ba = 0.5 .* (b - a);
    for i = 1 : n
        X(i) = ab + ba .* X(i); C(i) = ba .* C(i);
    end
end
% наближене значення інтеграла :
for i = 1:n
    s = s + C(i) .* f(X(i));
end
return

function S = i_lager(f, n)
% Знаходження значення невластного інтеграла
%
% 
$$S = \int_0^{\infty} \exp(-x) * f(x) dx$$

%
% з допомогою квадратурної формули Лагерра.
% Вхідні дані :

```

```

% f - функція з описом f(x);
% n - кількість вузлів (0<n<6).
% Вихідні дані :
% S - значення інтегралу.
S = 0;
if n < 1 | n > 5
    error('I_LAGER:error n<1 or n>5. ');
    return
end
% в залежності від n визначаємо вузли і вагові коефіцієнти :
switch n
    case 1
        X = [1]; C = [1];
    case 2
        X = [0.585786, 3.414214]; C = [0.853553, 0.146447];
    case 3
        X = [0.415775, 2.29428, 6.289945];
        C = [0.711093, 0.278518, 0.0103893];
    case 4
        X = [0.322548, 1.745761, 4.53662, 9.395071];
        C = [0.603154, 0.357419, 0.0388879, 0.000539295];
    case 5
        X = [0.26356, 1.413403, 3.596426, 7.08581, 12.640801];
        C = [0.521756, 0.398667, 0.0759424, 0.00361176, 0.00002337];
end
S = dot(C, f(X)); % наближене значення інтеграла
return

```

Практичне заняття № 02

ТЕМА № 2. Інтегральні рівняння (2 год).

Теоретична інформація:

Необхідно наближено розв'язати інтегральне рівняння Фредгольма II роду:

$$u(x) - \lambda \int_a^b K(x, \theta) u(\theta) d\theta = f(x), \quad (2.1)$$

де $f(x) \in C([a, b])$, $K(x, \theta) \in C([a, b] \times [a, b])$, $K(x, \theta) \neq 0$ – відомі функції, $\lambda \neq 0$ – дійсне число.

Алгоритм різницевого методу (метод квадратурних формул).

п.1. Вводимо сітку, наприклад,

$$\bar{X} = \{X_k = a + (k-1)h, \quad k = \overline{1, n}, \quad b-a = (n-1)h\}. \quad (2.2)$$

п.2. Записуємо рівняння (2.1) в точках сітки, а інтеграл замінюємо деякою квадратурною формулою (КФ), вузли якої співпадають з сіткою (2.2) – в результаті отримуємо СЛАР відносно $\vec{y} = \{y_1, y_2, \dots, y_n\}$:

$$\begin{cases} \sum_{j=1}^n (\delta_{ij} - \lambda \cdot C_j K(X_i, X_j)) y_j = f_i, \\ i = 1, 2, \dots, n, \end{cases} \quad (2.3)$$

де δ_{ij} – символ Кронекера, C_j – коефіцієнти відповідної КФ, $y_i = u(X_i)$, $f_i = f(X_i)$.

п.3. Розв'язуємо СЛАР (2.3) – отримуємо для $u(x)$ наближений сітковий розв'язок $\vec{y}$.

Різницевий метод без обмежень переноситься на :

- ✓ інтегральні рівняння Вольтерри другого роду;
- ✓ одновимірні нелінійні ІР (типу Урисона, Гаммерштейна);
- ✓ задачі на власні значення для ІР.

Алгоритм методу заміни ядра виродженням.

п.1. Наближення ядра $K(x, \theta)$ виродженням $K^0(x, \theta)$:

$$K^0(x, \theta) = \sum_{i=1}^m A_i(x) B_i(\theta), \quad (2.4)$$

($A_i(x)$ – лінійно-незалежні) здійснюється розвиненням $K(x, \theta)$ у певний ряд (степеневий, Фур'є і т.д.).

п.2. Розв'язуємо СЛАР відносно $\vec{c} = \{c_1, c_2, \dots, c_m\}$:

$$\begin{cases} c_i - \lambda \sum_{j=1}^m a_{ij} c_j = \varphi_i, \\ i = 1, 2, \dots, m, \end{cases} \quad (2.5)$$

де

$$a_{ij} = \int_a^b B_i(t) A_j(t) dt, \quad \varphi_i = \int_a^b B_i(t) f(t) dt. \quad (2.6)$$

п.3. Отримуємо для $u(x)$ наближений аналітичний розв'язок

$$y(x) = f(x) + \lambda \sum_{i=1}^m c_i A_i(x). \quad (2.7)$$

Чисельний варіант цього методу має наступну обчислювальну схему:

Після п.1 вводимо сітку, наприклад, (2.2). Обчислюємо значення функцій $f(x)$, $A_i(x)$, $B_i(x)$ на сітці $\bar{X}$. Використовуємо формули (2.4) і КФ з вузлами $\bar{X}$ і

будуємо СЛАР (2.5). Розв'язуємо її і отримуємо коефіцієнти $\bar{c}$. По формулі (2.7) обчислюємо $y(x)$ на сітці $\bar{X}$.

Методи заміни шуканої функції.

Алгоритм методу колокації.

п.1. Шукана функція $u(x)$ замінюється кусково-сталою функцією $S_{0,1}(x)$, $x \in [a, b]$ зі значеннями y_i в певних точках $\bar{x}_i$ (точках колокації), де $\bar{x}_i \in \Delta_i = [X_i, X_{i+1}]$, $h_i = X_{i+1} - X_i$, $[a, b] = \bigcup_{i=1}^N \Delta_i$, в результаті чого, (2.1) приводиться до системи співвідношень :

$$\begin{cases} y_i - \lambda \sum_{j=1}^N y_j \int_{X_j}^{X_{j+1}} K(\bar{x}_i, \theta) d\theta = f(\bar{x}_i), \\ i = 1, 2, \dots, N. \end{cases} \quad (2.8)$$

п.2. Апроксимуємо інтеграли в (2.8) з точністю $O(h_j)$ і отримуємо СЛАР відносно $\bar{y} = \{y_1, y_2, \dots, y_N\}$:

$$\begin{cases} \sum_{j=1}^N (\delta_{ij} - \lambda \cdot h_j K(\bar{x}_i, \bar{x}_j)) y_j = f(\bar{x}_i), \\ i = 1, 2, \dots, N. \end{cases} \quad (2.9)$$

п.3. Розв'язуємо (2.9) і маємо сплайнове наближення до функції $u(x)$.

Якщо точки колокацій вибрати наступним чином $\bar{x}_i = 0.5(X_i + X_{i+1})$, то інтеграли в (2.8) обчислюються по КФ центральних прямокутників і маємо точність $O(h_j^2)$.

Методи заміни функції $u(x)$ на сплайн-функції $S_{1,1}(x)$, $S_{2,1}(x)$, $S_{3,1}(x)$ див. [10, с. 107-111; 34, 35].

Алгоритм методу послідовних наближень.

п.1. Наближення розв'язку (2.1) $u(x) \approx y_m(x)$ шукається у вигляді ряду по степеням параметра λ :

$$y_m(x) = \varphi_0(x) + \lambda \varphi_1(x) + \lambda^2 \varphi_2(x) + \dots + \lambda^m \varphi_m(x), \quad (2.10)$$

який збігається при $|\lambda(b-a)|M < 1$, де $|K(x, \theta)| \leq M$, $M \in \text{Const}$.

п.2. Функції $\varphi_i(x)$ обчислюються за рекурентними формулами

$$\begin{cases} \varphi_0(x) = f(x), \\ \varphi_i(x) = \int_a^b K(x, \theta) \varphi_{i-1}(\theta) d\theta, \quad i = \overline{1, m}. \end{cases} \quad (2.11)$$

Розрахунки за формулами (2.11) закінчуються, якщо для заданого $0 < \varepsilon < 1$ виконується умова $\|\varphi_m(x)\| \leq \varepsilon$.

Чисельний варіант цього методу має наступний алгоритм :

п.1. Вводимо сітку, наприклад, (2.2). Обчислюємо вектори $\bar{\varphi}_0 = \{f(X_k)\}_{k=1}^n$, $\bar{y}_0 = \bar{\varphi}_0$ і матрицю $B = \lambda \{C_j K(X_k, X_j)\}_{k,j=1}^n$, де C_j – коефіцієнти певної КФ з вузлами $\bar{X}$.

п.2. Наближений сітковий розв'язок обчислюється за рекурентними формулами

$$\begin{cases} \bar{\varphi}_i = B \cdot \bar{\varphi}_{i-1}, \\ \bar{y}_i = \bar{y}_{i-1} + \bar{\varphi}_i, \quad i = \overline{1, m}. \end{cases} \quad (2.12)$$

Розрахунки за формулами (2.12) закінчуються, якщо для заданого $0 < \varepsilon < 1$ виконується умова $\|\vec{\varphi}_m\|_h \leq \varepsilon$.

Аудиторне завдання:

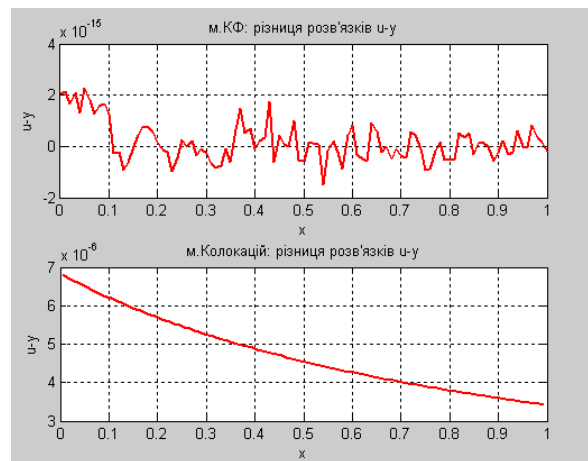
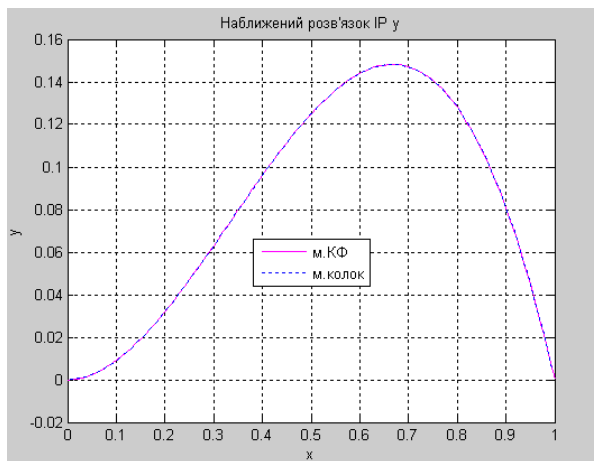
Розглянемо приклади розв'язання інтегральних рівнянь.

п.1. Знайти наближений розв'язок ІР Фредгольма II роду і порівняти з точним розв'язком $u(x) = x^2(1-x)$

$$u(x) - 12 \int_0^1 \frac{1}{1+x} u(\theta) d\theta = x^2(1-x) - \frac{1}{x+1}, \quad x \in [0, 1]$$

- а)** методом квадратурних формул (КФ Сімпсона);
б) методом колокацій (м.центральних прямокутників).

```
% М-файл p02a01.m
clear all, close all, clc
% вхідні дані
fr2mk_k = @(x,s) (1./(x+1)); % K(x,s)
fr2mk_f = @(x) (x.*x.*(1-x)-1./(x+1)); % f(x)
lamb = 12; a = 0; b = 1; n = 101;
X = linspace(a, b, n);
[x,YY] = fr2_mcol(lamb, X, fr2mk_k, fr2mk_f);
[X,Y,n] = fr2_mk(lamb, a, b, 1, n, fr2mk_k, fr2mk_f);
plot(X,Y,'m -',x,YY,'b :')
title('Наближений розв'язок ІР у')
legend('м.КФ', 'м.колок', 'Location', 'Best')
xlabel('x'), ylabel('y'), grid on, pause, close all
u = @(x) (x.*x.*(1-x)); % u(x) - точний розв'язок
subplot(2,1,1); plot(X,u(X)-Y,'r','LineWidth',2)
title('м.КФ: різниця розв'язків u-y')
xlabel('x'), ylabel('u-y'), grid on
subplot(2,1,2); plot(x,u(x)-YY,'r','LineWidth',2)
title('м.Колокацій: різниця розв'язків u-y')
xlabel('x'), ylabel('u-y'), grid on
```



п.2. Знайти наближений розв'язок ІР Фредгольма-Урисона

$$\int_0^{0.35} \exp(x-u(t)) dt = x - u(x), \quad x \in [0, 0.35]$$

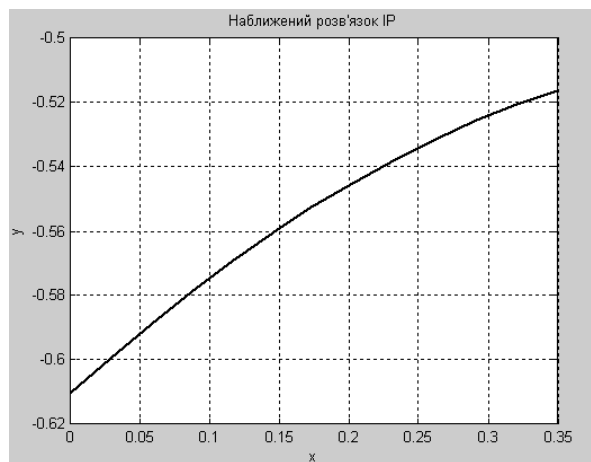
методом квадратурних формул.

```
function p02a02
clear all, close all, clc
% вхідні дані
```

```

a = 0; b = 0.35;
% параметри сітки і ітераційного процесу
n = 13; eps = 1e-8; mi = 500;
Y = 0.05.*ones(n,1); % початкове наближення
[X,Y] = frurm_k(a,b,1,n,@frurm_k,@frurm_f,Y,eps,mi,2);
plot(X,Y,'k','LineWidth',2)
title('Наближений розв'язок IP')
xlabel('x'), ylabel('y'), grid on
end
function [y, yp] = frurm_k(x, t, u)
    y = exp(x - u); % опис ядра K(x,t,u)
    yp = -y; % опис похідної dK(x,t,u)/du
end
function [y, yp] = frurm_f(x, u)
    y = x - u; % опис f(x,u)
    yp = -1; % опис похідної df(x,u)/du
end

```



п.3. Знайти наближений розв'язок IP

а) $u(x) - \frac{1}{3} \int_0^1 (xt^2 + x^2t)u(t)dt = \left(\frac{11}{12} - \frac{x}{9}\right)x, x \in [0,1] \quad (m=2)$ (модельна задача $u(x)=x$);

б) $u(x) - \int_0^2 (\exp(xt) - 1) * u(t)dt = x, \quad x \in [0,2] \quad (m=4, 8, 12, 16).$

Використати чисельний метод заміни ядра виродженням.

```

function p02a03
clear all, close all, clc
% а) модельна задача
global m
lamb = 1/3; a = 0; b = 1; % вхідні дані
n = 21; % кількість кв.вузлів м.Сімпсона
m = 2;
[X,Y,n] = fr2_mzjd(lamb,a,b,1,n,@fr2mzj_k,@fr2mzj_f);
plot(X,Y,'m','LineWidth',2)
title('Наближений розв'язок IP y')
xlabel('x'), ylabel('y'), grid on, pause, close all
u = @(x)(x); % u(x) - точний розв'язок
Z = u(X)-Y'; % Різниця між точним і набл.розв'язком
plot(X,Z,'r','LineWidth',2)
title('Різниця розв'язків u-y')
xlabel('x'), ylabel('u-y'), grid on, pause, close all
% б)
lamb = 1; a = 0; b = 2; % вхідні дані

```

```

cl = 'mbkr'; M = [4:4:16];
for j = 1:4;
    m = M(j); L{j} = sprintf('m=%u',m);
    [X,Y,n] = fr2_mzjd(lamb, a, b, 1, n, @mzj_k, @mzj_f);
    plot(X,Y,cl(j)), hold on
end
hold off, title('Наближений розв\'язок IP')
grid on, xlabel('x'), ylabel('y')
legend(L{:},'Location','Best')
end

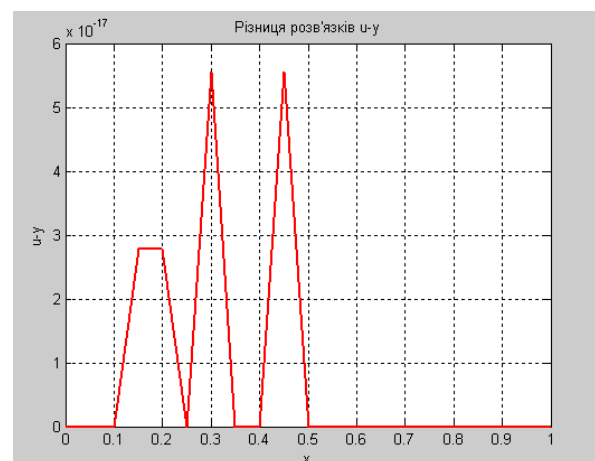
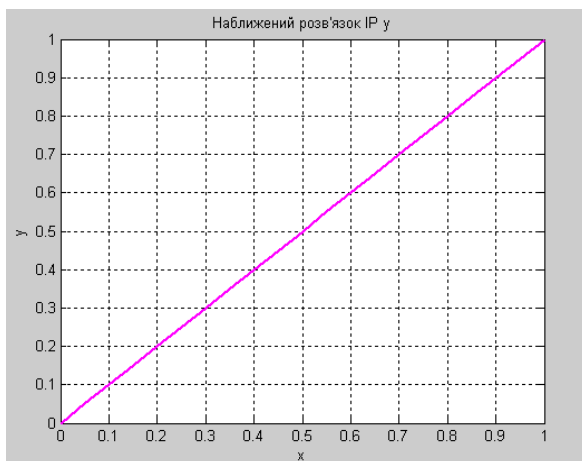
function [A, B] = fr2mzj_k(x)
    % ядро K(x,s)=SUM(k=1,m,Ak(x)*Bk(s))
    global m
    t = x.^2;
    A = [x; t];
    B = [t; x];
end

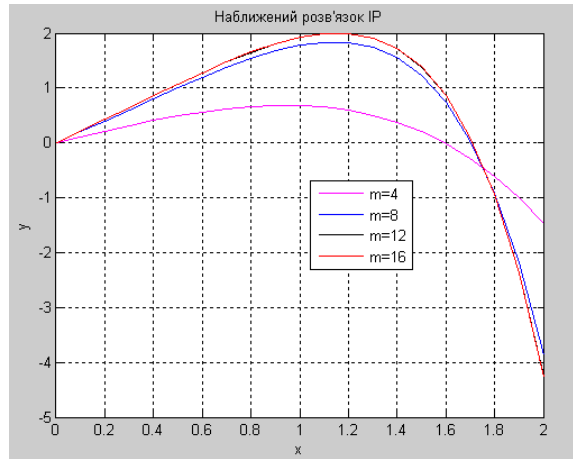
function yp = fr2mzj_f(x)
    yp = (11/12-1/9.*x).*x; % f(x)
end

function [A,B] = mzj_k(x)
    % опис ядра K(x,s)=SUM(k=1,m,Ak(x)*Bk(s))
    global m
    f = 1; z = 1; A = zeros(m,1); B = A;
    for i = 1:m
        z = z.*x; f = f.*i;
        A(i) = z; B(i) = z./f;
    end
end

function yp = mzj_f(x)
    yp = x; % f(x)
end

```





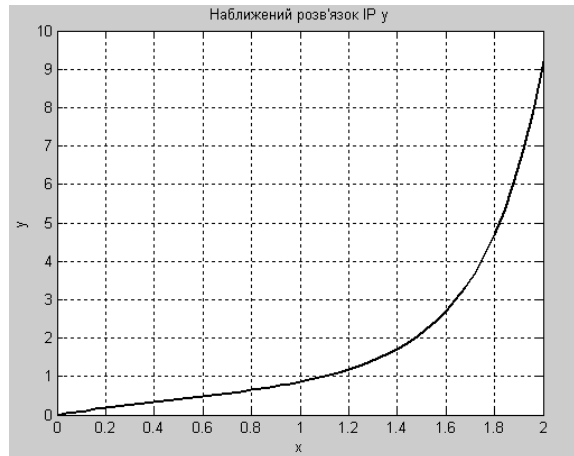
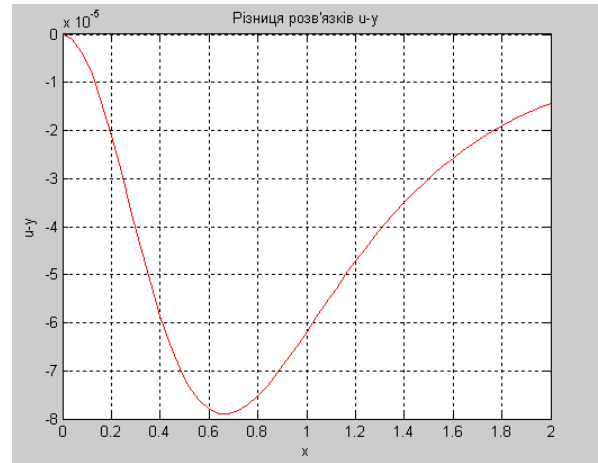
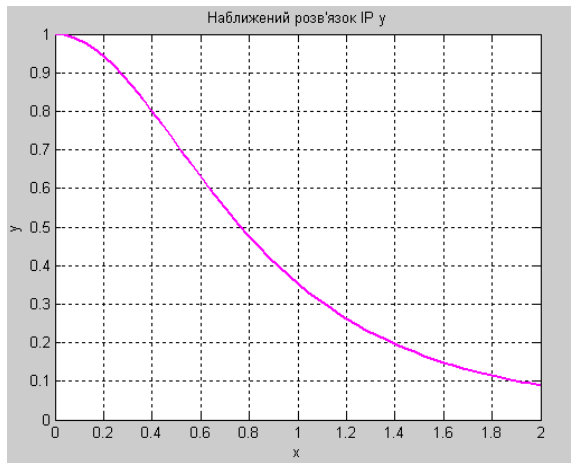
п.4. Знайти наближений розв'язок IP

а) $u(x) + \int_0^x \frac{t}{1+x^2} * u(t) dt = \frac{1}{1+x^2}, \quad x \in [0,2]$ (модельна задача $u(x) = \frac{1}{\sqrt{(1+x^2)^3}}$);

б) $u(x) - \int_0^x (xt-1) * u(t) dt = x, \quad x \in [0,2].$

Використати різницевий метод з кількістю вузлів сітки $n \geq 3$, а варіант квадратурної формули – формула трапецій.

```
function p02a04
    clear all, close all, clc
    % а) модельна задача
    lamb = -1; a = 0; b = 2;          % вхідні дані
    n = 51;
    [X,Y,n] = vo2_mk(lamb, a, b, n, @vo2mk_k, @vo2mk_f);
    plot(X,Y,'m','LineWidth',2)
    title('Наближений розв'язок IP у')
    xlabel('x'), ylabel('y'), grid on, pause, close all
    u = @(x) (1./sqrt((1+x.*x).^3)); % u(x) - точний розв'язок
    Z = u(X)-Y'; % Різниця між точним і набл.розв'язком
    plot(X,Z,'r'), title('Різниця розв'язків u-y')
    xlabel('x'), ylabel('u-y'), grid on, pause, close all
    % б)
    lamb = 1; a = 0; b = 2;          % вхідні дані
    [X,Y,n] = vo2_mk(lamb, a, b, n, @vo2_k, @vo2_f);
    plot(X,Y,'k','LineWidth',2), grid on
    title('Наближений розв'язок IP у')
    xlabel('x'), ylabel('y')
end
function yp = vo2mk_k(x,s)
    yp = s./(1+x.*x); % K(x,s)
end
function yp = vo2mk_f(x)
    yp = 1./(1+x.*x); % f(x)
end
function yp = vo2_k(x,s)
    yp = x.*s-1; % K(x,s)
end
function yp = vo2_f(x)
    yp = x; % f(x)
end
```

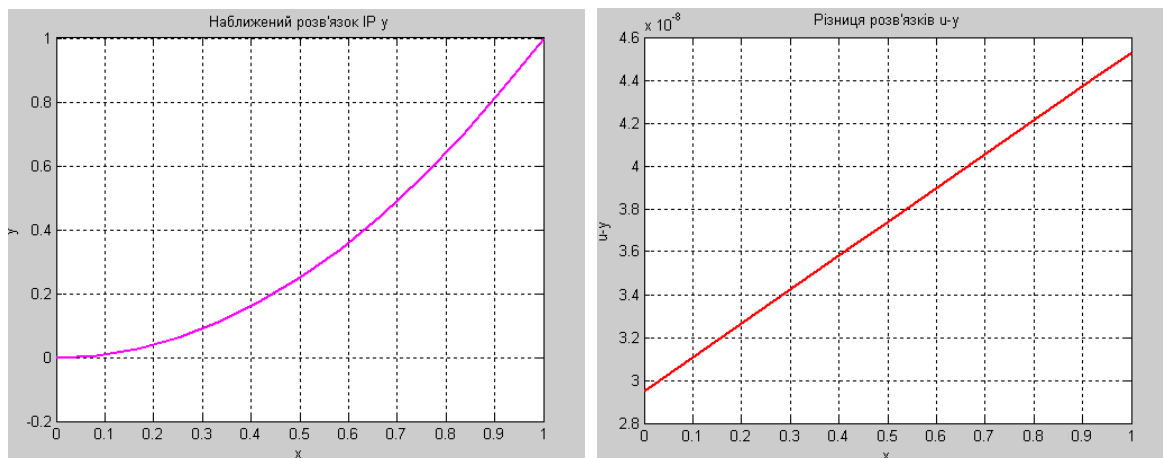


п.5. Знайти наближений розв'язок IP

$$u(x) - \frac{1}{4} \int_0^1 (xt+1) * u(t) dt = \left(x - \frac{1}{16}\right)x - \frac{1}{12}, \quad x \in [0,1] \quad (\text{модельна задача } u(x) = x^2);$$

Використати різницевий метод послідовних наближень ($\text{eps} = 10^{-7}$). Для числового інтегрування обрати КФ Сімпсона з кількістю вузлів сітки $n = 13$.

```
function p02a05
    clear all, close all, clc
    lamb = 0.25; a = 0; b = 1; % вхідні дані
    n = 13; eps = 1e-7; mi = 5000;
    [X,Y,n] = fr2_mpn(lamb,a,b,1,n,eps,mi,@fr2mpn_k,@fr2mpn_f);
    plot(X,Y,'m','LineWidth',2)
    title('Наближений розв'язок IP y')
    xlabel('x'), ylabel('y'), grid on, pause, close all
    u = @(x) (x.*x); % u(x) - точний розв'язок
    Z = u(X)-Y'; % Різниця між точним і набл.розв'язком
    plot(X,Z,'r','LineWidth',2)
    title('Різниця розв'язків u-y')
    xlabel('x'), ylabel('u-y'), grid on
end
function yp = fr2mpn_k(x,s)
    yp = x.*s+1; % K(x,s)
end
function yp = fr2mpn_f(x)
    yp = (x-1/16).*(x-1/12); % f(x)
end
```



Завдання для самостійної роботи:

п.1. Розв'язати задачу 1.**k**, використовуючи чисельний варіант методу заміни ядра виродженим (кількість доданків розвинення **m** ≥ 10) з урахуванням варіанту **V** = **mod(g+d,2)** і порівняти отриманий результат з результатом методу колокацій. Кількість точок сітки обрати = 21 (для обох методів).

Тут через **k** позначено номер за списком студента у групі, **g** – номер групи, **d** – день народження студента.

$$1.1 \quad u(x) + \int_0^{1+V} (x \exp(xt) - 1) \cdot u(t) dt = \exp(x) - x; \quad x \in [0, 1+V].$$

$$1.2 \quad u(x) + \int_0^1 (x \exp(xt) + V) \cdot u(t) dt = 1 - x^2; \quad x \in [0, 1].$$

$$1.3 \quad u(x) - 2 \int_0^{0.5\pi} \cos(2xt) \cdot u(t) dt = \sin(x) + V \cdot \exp(x); \quad x \in [0, 0.5\pi].$$

$$1.4 \quad u(x) + \int_0^2 sh(xt) u(t) dt = V - x + Vx^2; \quad x \in [0, 2].$$

$$1.5 \quad u(x) + \int_0^2 (t \exp(xt) + V) \cdot u(t) dt = -x^2; \quad x \in [0, 2].$$

$$1.6 \quad u(x) + \int_0^2 (1+x) \cdot ch(xt) u(t) dt = V - x; \quad x \in [0, 2].$$

$$1.7 \quad u(x) - \int_{-0.5}^{0.5} \left(\frac{1}{1-xt} \right) u(t) dt = V + x^2; \quad x \in [-0.5, 0.5].$$

$$1.8 \quad u(x) + \int_0^1 (1+t) tg(2xt) u(t) dt = V + x; \quad x \in [0, 1].$$

$$1.9 \quad u(x) - 3 \int_0^2 (1+x)(1+t) \cdot ctg(xt) u(t) dt = V - x; \quad x \in [0, 2].$$

$$1.10 \quad u(x) + 0.5 \int_0^{0.5\pi} \sin(2xt) u(t) dt = V - \sqrt{x}; \quad x \in [0, 0.5\pi].$$

$$1.11 \quad u(x) - 5 \int_{-0.5}^{0.5} (1+x^2) \cdot \arcsin(xt) \cdot u(t) dt = V + \exp(x); \quad x \in [-0.5, 0.5].$$

$$1.12 \quad u(x) - 2 \int_{-0.5}^{0.5} (1+x) \cdot \arccos(xt) \cdot u(t) dt = \sin(V+x); \quad x \in [-0.5, 0.5].$$

$$\begin{aligned}
1.13 \quad & u(x) + 3 \int_{-0.5}^{0.5} (1+t) \cdot \arctan(xt) \cdot u(t) dt = V + x; \quad x \in [-0.5, 0.5]. \\
1.14 \quad & u(x) + 3 \int_{-0.5}^{0.5} (1+x) \cdot \operatorname{arth}(xt) \cdot u(t) dt = V + \cos(x); \quad x \in [-0.5, 0.5]. \\
1.15 \quad & u(x) - \int_0^1 (1+t) \exp(-xt) \cdot u(t) dt = (1 + \cos(x^2))^2; \quad x \in [0, 1]. \\
1.16 \quad & u(x) - \int_0^1 (1+x-t)^4 \cdot u(t) dt = (1 + \cos(V+x))^2; \quad x \in [0, 1]. \\
1.17 \quad & u(x) + 0.1 \int_{-0.5}^{0.5} (2+t) \cdot \ln(1+xt) \cdot u(t) dt = V + x^2; \quad x \in [-0.5, 0.5]. \\
1.18 \quad & u(x) - 0.1 \int_{-0.4}^{0.4} (1+2t) \cdot \ln(1-xt) \cdot u(t) dt = V + x; \quad x \in [-0.4, 0.4]. \\
1.19 \quad & u(x) - \int_0^\pi (1+x)^2 \sin(xt) \cdot u(t) dt = \sin(x) + V; \quad x \in [0, \pi]. \\
1.20 \quad & u(x) + 3 \int_{-0.5}^{0.5} \ln(1-2xt) \cdot u(t) dt = V + \cos(x); \quad x \in [-0.5, 0.5]. \\
1.21 \quad & u(x) + \int_0^1 (2+t)^2 \cdot \operatorname{sh}(xt) \cdot u(t) dt = -Vx + x^2; \quad x \in [0, 1]. \\
1.22 \quad & u(x) - 2 \int_0^{0.5} \frac{1+t}{1-xt} u(t) dt = \sin(V+x); \quad x \in [0, 0.5]. \\
1.23 \quad & u(x) - \int_0^\pi \cos(x\sqrt{t}) \cdot u(t) dt = \sin(V+x); \quad x \in [0, \pi]. \\
1.24 \quad & u(x) + 3 \int_0^\pi \sin(t\sqrt{x}) \cdot u(t) dt = x - V; \quad x \in [0, \pi]. \\
1.25 \quad & u(x) - 2 \int_{-0.5}^{0.5} \frac{1+x}{1-xt} u(t) dt = \exp(V+x); \quad x \in [-0.5, 0.5].
\end{aligned}$$

п.2. Розв'язати задачу 2.k (нелінійне ІР Фредгольма-Урисона), використовуючи чисельний метод і КФ згідно варіанту $V = \text{mod}(g+d, 3)$ з кількістю вузлів $n = 7$, де

V	КФ
1	Гаусса
2	Маркова
3	Чебишева

Для розв'язання нелінійної системи використати метод Ньютона.

$$\begin{aligned}
2.1 \quad & u(x) - \int_{0.05}^{0.35} \exp(x - u(t)) dt = x; \quad x \in [0.05, 0.35]. \\
2.2 \quad & u(x) - \int_0^\pi \cos(x + u(t)) dt = x^2; \quad x \in [0, \pi]. \\
2.3 \quad & u(x) - \int_{0.05}^{0.15} \exp(-x^2 + u(t)) dt = 1 + x; \quad x \in [0.05, 0.15].
\end{aligned}$$

- 2.4 $\int_0^1 \sin(x^2 - u(t)) dt = u(x) + 2; \quad x \in [0,1].$
- 2.5 $\int_{-0.5}^{0.5} \cos(tx - u(t)) dt = u(x)^2 - x; \quad x \in [-0.5, 0.5].$
- 2.6 $\int_{-0.3}^{0.3} (1 - x - u(t))^2 dt = 2 + \sin(u(x)); \quad x \in [-0.3, 0.3].$
- 2.7 $u(x) - \int_0^1 (x - t \cdot u^2(t)) dt = (1 + x)^2; \quad x \in [0,1].$
- 2.8 $u(x) + \int_0^{0.35} (1 - x - u^2(t)) dt = x; \quad x \in [0, 0.35].$
- 2.9 $\int_0^{1.5} \sin(x - u \cdot t) dt = u(x) \exp(-x); \quad x \in [0, 1.5].$
- 2.10 $u(x) + 0.1 \int_0^{1.5} \sin(xu(t)) dt = 1; \quad x \in [0, 1.5].$
- 2.11 $u(x) - \int_0^{\pi} (x + \cos(u(t))) dt = \sin(x); \quad x \in [0, \pi].$
- 2.12 $\int_0^{\pi} \sin(x + u(t)) dt = u(x) - 1; \quad x \in [0, \pi].$
- 2.13 $\int_{-1.5}^{-1} \exp(x - u(t)) dt = 2 - xu(x); \quad x \in [-1.5, -1].$
- 2.14 $\sin(u(x)) - \int_0^{0.7} (x - u(t))^2 dt = x; \quad x \in [0, 0.7].$
- 2.15 $1 + \int_{0.5}^{1.5} (x^3 - u(t)) dt = (u(x) - x)^2; \quad x \in [0.5, 1.5].$
- 2.16 $\int_0^1 \frac{t - u^2(t)}{1 + x} dt = x - u(x); \quad x \in [0, 1].$
- 2.17 $\int_0^1 \frac{x - u^3(t)}{1 + t} dt = x + u(x); \quad x \in [0, 1].$
- 2.18 $\int_0^1 \frac{x - u^3(t)}{1 + t \cdot x} dt = \frac{1 + u(x)}{1 + x \cdot x}; \quad x \in [0, 1].$
- 2.19 $\int_0^1 (x - \sin(u(t))) dt = \frac{u(x)}{1 + x}; \quad x \in [0, 1].$
- 2.20 $u(x) + \int_{-0.3}^{0.7} \exp(x + t - u^2(t)) dt = x; \quad x \in [-0.3, 0.7].$
- 2.21 $\int_0^1 \exp(2xt - u^2(t)) dt = u(x) - 3; \quad x \in [0, 1].$
- 2.22 $u(x) + 0.01 \int_0^1 (1 - u(t))^3 dt = x^2; \quad x \in [0, 1].$
- 2.23 $\int_{0.2}^1 (1 - x^2 + u^2(t)) dt = (x - u(x))^2; \quad x \in [0.2, 1].$

$$2.24 \quad \int_0^1 (x^4 + u^2(t)) dt = x + u(x) - u^2(x); \quad x \in [0,1].$$

$$2.25 \quad \int_0^1 \sqrt{x+t+u^2(t)} dt = \frac{5-u(x)}{1+x^2}; \quad x \in [0,1].$$

Програмне забезпечення на MATLAB:

```
function [X Y n] = fr2_mk(lambda, a, b, nkf, n, kfun, ffun)
% Розв'язок лінійного неоднорідного інтегрального рівняння
% типу Фредгольма другого роду методом заміни інтеграла
% скінченною сумою
%
%          b
%      y(x)=f(x)+lambda* ∫ K(x,s)*y(s) ds
%          a
%
% вхідні дані :
% lambda - числовий параметр рівняння;
% a       - нижня границя інтеграла;
% b       - верхня границя інтеграла;
% nkf     - 0 = КФ методу трапецій,
%          1 = КФ методу Сімпсона;
% n       - число точок сітки. Повинно задовольняти умові:
%          n > 1,                при nkf=0,
%          n > 2 & mod(n,2) == 1, при nkf=1;
% kfun    - функція з описом ядра K(x,s);
% ffun    - функція з описом правої частини f(x);
% вихідні дані:
% X       - масив вузлів сітки;
% Y       - масив наближеного розв'язку;
% n       - розмірність масивів X, Y.

if nargin<7
    error('FR2_MK:NotEnoughInputs',...
        'Not enough input arguments. See FR2_MK. ');
    return
end
if a >= b
    error('FR2_MK:error a >= b.',...
        'a >= b. See FR2_MK. ');
    return
end
switch nkf
case 0
    if n < 2
        n = 2;
    end
case 1
    if n < 3
        n = 3;
    else
        if mod(n,2) == 0
            n = n+1;
        end
    end
otherwise
    error('FR2_MK:Unknown method.',...
        'Unknown method. See FR2_MK. ');
    return
end
```

```

% формування вузлів
X = linspace(a,b,n); h = X(2) - X(1); n1 = n-1;
% формування коефіцієнтів P(i), i=1,...,n КФ
P = zeros(n,1);
switch nkf
case 0
    P(1) = 0.5.*h;
    for i=2:n1
        P(i) = h;
    end
    P(n) = P(1);
case 1
    P(1) = h./3; c2 = 2.*P(1); c1 = 2.*c2;
    for i=2:2:n1
        P(i) = c1; P(i+1) = c2;
    end
    P(n) = P(1);
end
% формування матриці A(i,k) і вектора правої частини F(i) для СЛАР
%  $Y(i) - \lambda \sum_{k=1}^n P(k) * K(X(i), X(k)) * Y(k) = f(X(i))$ ,
% i=1,...,n
A = zeros(n, n); F = P;
for i = 1 : n
    x = X(i); F(i) = ffun(x);
    for k = 1 : n
        A(i, k) = -lambda .* P(k) .* kfun(x, X(k));
    end
    A(i, i) = 1 + A(i, i);
end
% розв'язок СЛАР A*Y=F
Y = A\F;
return

function [x Y] = fr2_mcol(lambda, X, kfun, ffun)
% Розв'язок лінійного неоднорідного інтегрального рівняння
% типу Фредгольма другого роду методом колокацій
%  $y(x) = f(x) + \lambda \int_a^b K(x,s) * y(s) ds$ 
%  $a$ 
% вхідні дані :
% lambda - числовий параметр рівняння;
% X - сплайнова сітка a=X(1)<X(2)<...<X(end-1)<X(end)=b,
% де [a,b] - інтервал інтегрування;
% kfun - функція з описом ядра K(x,s);
% ffun - функція з описом правої частини f(x);
% вихідні дані:
% x - масив колокаційної сітки x(i)=0.5*(X(i)+X(i+1)), i=1,...,n-1;
% Y - масив наближеного розв'язку в точках колокації;

if nargin<4
    error('FR2_MCOL:NotEnoughInputs',...
        'Not enough input arguments. See FR2_MCOL. ');
return
end
n = length(X);
if n < 2
    n = 2;
end

```

```

% формування колокаційної сітки
N = n-1; x = zeros(1, N); h = x; I = 1 : N;
for i = I
    j = i+1;
    x(i) = 0.5.*(X(i)+X(j));
    h(i) = X(j)-X(i);
end
% формування матриці A(i,j) і вектора правої частини F(i) для СЛАР
%  $Y(i) - \lambda \sum_{j=1, N} (X(j+1) - X(j)) * K(x(i), x(j)) * Y(j) = f(x(i))$ ,
%  $i=1, \dots, N$ 
A = zeros(N, N); F = zeros(N, 1);
for i = I
    xi = x(i); F(i) = ffun(xi);
    for j = I
        A(i, j) = -lambda.*h(j).*kfun(xi, x(j));
    end
    A(i, i) = 1 + A(i, i);
end
% розв'язок СЛАР A*Y=F
Y = A\F;
return

```

```

function [X Y n] = fr2_mpn(lambda, a, b, nkf, n, e, mki, kfun, ffun)

```

```

% Розв'язок лінійного неоднорідного інтегрального рівняння
% типу Фредгольма другого роду методом послідовних наближень
%
% 
$$y(x) = f(x) + \lambda \int_a^b K(x, s) * y(s) ds$$

%
% вхідні дані :
% lambda - числовий параметр рівняння;
% a - нижня границя інтеграла;
% b - верхня границя інтеграла;
% nkf - 0 = КФ методу трапецій,
%       1 = КФ методу Сімпсона;
% n - число точок сітки. Повинно задовольняти умові:
%     n > 1, при nkf=0,
%     n > 2 & mod(n,2) == 1, при nkf=1;
% e - точність (e>0);
% mki - максимальна кількість кроків ітерацій;
% kfun - функція з описом ядра K(x,s);
% ffun - функція з описом правої частини f(x);
% вихідні дані :
% X - масив вузлів сітки;
% Y - масив наближеного розв'язку;
% n - розмірність масивів X, Y.

```

```

if nargin<9
    error('FR2_MPN:NotEnoughInputs',...
        'Not enough input arguments. See FR2_MPN.');
```

```

    return
end
switch nkf
    case 0
        if n < 2
            n = 2;
        end
    case 1
        if n < 3
            n = 3;

```

```

        else
            if mod(n,2) == 0
                n = n + 1;
            end
        end
    otherwise
        error('FR2_MPN:Unknown method.','Unknown method. See FR2_MPN. ');
        return
    end
end
if a >= b
    error('FR2_MPN:error a >= b.','a >= b. See FR2_MPN. ');
    return
end
if e <= 0
    error('FR2_MPN:error e <= 0.','e <= 0. See FR2_MPN. ');
    return
end
% формування вузлів
X = linspace(a, b, n); h = X(2) - X(1); n1 = n - 1;
% формування коефіцієнтів P(i), i=1,...,n KФ
P = zeros(n, 1); FI = P; Y = FI; AK = zeros(n);
switch nkf
    case 0
        P(1) = 0.5 .* h;
        for i=2:n1
            P(i) = h;
        end
        P(n) = P(1);
    case 1
        P(1) = h ./ 3; c2 = 2 .* P(1); c1 = 2 .* c2;
        for i = 2 : 2 : n1
            P(i) = c1; P(i+1) = c2;
        end
        P(n) = P(1);
end
% формування матриці AK(i,j) і вектора FI(i),
% де AK(i,j)=lambda*P(j)*K(X(i),X(j)),
% FI(i) = f(X(i)), i=1,...,n; j=1,...,n
for i = 1 : n
    x = X(i); ff = ffun(x); FI(i) = ff; Y(i) = ff;
    for j = 1 : n
        AK(i, j) = lambda .* P(j) .* kfun(x,X(j));
    end
end
% послідовні наближення
k = 0; F = 1;
while F & k < mki
    k = k + 1; F = 0; FIi = AK * FI;
    for i = 1 : n
        ff = FIi(i); Y(i) = Y(i) + ff; FI(i) = ff;
        if F == 0
            F = abs(ff) > e;
        end
    end
end
end
return

```

function [X Y n] = fr2_mzjd(lambda,a,b,nkf,n,kfun,ffun)
 % Розв'язок лінійного неоднорідного інтегрального рівняння

```

% типу Фредгольма другого роду методом заміни ядра виродженим
%
%      
$$y(x) = f(x) + \lambda \int_a^b K(x,s) * y(s) ds,$$

%      де  $K(x,s) = \sum_{k=1,m} A_k(x) * B_k(s)$ 
% вхідні дані :
% lambda - числовий параметр рівняння;
% a - нижня границя інтеграла;
% b - верхня границя інтеграла;
% nkf - 0 = КФ методу трапецій,
%       1 = КФ методу Сімпсона
%       (для чисельного обчислення інтегралів);
% n - число точок сітки. Повинно задовольняти умові:
%     n > 1,                                при nkf=0,
%     n > 2 & mod(n,2) == 1,                при nkf=1;
% kfun - функція з описом ядра K(x,s);
% ffun - функція з описом правої частини f(x);
% вихідні дані:
% X      - масив вузлів сітки;
% Y      - масив наближеного розв'язку;
% n      - розмірність масивів X, Y.
global m
if nargin<7
    error('FR2_MZJ:NotEnoughInputs',...
        'Not enough input arguments. See FR2_MZJ.');
```

```

    return
end
if a >= b
    error('FR2_MZJ:error a >= b.', 'a >= b. See FR2_MZJ.');
```

```

    return
end
switch nkf
    case 0
        if n < 2
            n = 2;
        end
    case 1
        if n < 3
            n = 3;
        else
            if mod(n,2) == 0
                n = n + 1;
            end
        end
    otherwise
        error('FR2_MZJ:Unknown method.',...
            'Unknown method. See FR2_MZJ.');
```

```

    return
end
% формування вузлів
X = linspace(a, b, n); h = X(2) - X(1); n1 = n - 1;
% формування коефіцієнтів P(i), i=1,...,n КФ
P = zeros(n,1); FF = P; Y = P;
switch nkf
    case 0
        P(1) = 0.5.*h;
        for i = 2 : n1
            P(i) = h;
        end

```

```

        P(n) = P(1);
    case 1
        P(1) = h ./ 3; c2 = 2 .* P(1); c1 = 2 .* c2;
        for i = 2 : 2 : n1
            P(i) = c1; P(i+1) = c2;
        end
        P(n) = P(1);
    end
% формування допоміжних матриць AK, BK і вектора FF,
% де AK(k,i)=Ak(X(i)), BK(k,i)=Bk(X(i)),
% FF(i)=f(X(i)), i=1,...,n.
for i = 1 : n
    x = X(i);
    [AK(:, i), BK(:, i)] = kfun(x); FF(i) = ffun(x);
end
% формування матриці A і вектора F СЛАР A*C=F,
% де A(i,j)=if(i=j,1,0)-lambda*SUM(k=1,n, P(k)*AK(j,k)*BK(i,k) ),
% F(i) =SUM(k=1,n, P(k)*FF(k)*BK(i,k) ), i=1,...,n
F = zeros(m,1); A = zeros(m,m);
for i = 1 : m
    for j = 1 : m
        aa = 0;
        for k = 1 : n
            aa = aa + P(k).*AK(j, k).*BK(i, k);
        end
        aa = -lambda.*aa;
        if i == j
            aa = 1 + aa;
        end
        A(i,j) = aa;
    end
    aa = 0;
    for k = 1 : n
        aa = aa + P(k).*FF(k).*BK(i, k);
    end
    F(i) = aa;
end
% розв'язок СЛАР A*C=F
C = A\F;
% розв'язок у вигляді
% y(x)=f(x)+lambda*SUM(k=1,m, C(k)*Ak(x) )
for i = 1 : n
    Y(i) = FF(i) + lambda.*dot(C, AK(:, i));
end
return

```

```

function [X Y n] = vo2_mk(lambda, a, b, n, kfun, ffun)
% Розв'язок лінійного неоднорідного інтегрального рівняння
% типу Вольтерри другого роду методом заміни інтеграла
% скінченною сумою (квадратурні формули методу трапецій)
%
% 
$$y(x) = f(x) + \lambda \int_a^x K(x,s) * y(s) ds, \quad x \in [a,b].$$

%
% вхідні дані :
% lambda - числовий параметр рівняння;
% a,b - інтервал визначення функції y(x);
% n - число точок сітки (n>1)
% kfun - функція з описом ядра K(x,s);
% ffun - функція з описом правої частини f(x);

```

```

% вихідні дані:
% X      - масив вузлів сітки;
% Y      - масив наближеного розв'язку;
% n      - розмірність масивів X, Y.
if nargin<6
    error('VO2_MK:NotEnoughInputs',...
        'Not enough input arguments. See VO2_MK. ');
    return
end
if a >= b
    error('VO2_MK:error a >= b.', 'a >= b. See VO2_MK. ');
    return
end
if n < 2
    n = 2;
end
% формування вузлів
X = linspace(a, b, n); h = X(2) - X(1);
Y = zeros(n, 1); P = Y;
% Знаходження розв'язку (СЛАР з лівою трикутною матрицею):
% Y(i) = (f(X(i))+lambda*SUM(k=1,i-1, P(k)*K(X(i),X(k))*Y(k) )) /
% (1-lambda*P(i)*K(X(i),X(i))), i=1,...,n.
for i = 1 : n
% формування коефіцієнтів P(j), j=1,...,i КФ
    P(1) = 0.5.*h;
    for j = 2 : i - 1
        P(j) = h;
    end
    P(i) = P(1);
    x = X(i); c1 = 0;
    for k = 1 : i - 1
        c1 = c1 + P(k).*kfun(x, X(k)).*Y(k);
    end
    c1 = ffun(x) + lambda.*c1;
    c2 = 1 - lambda.*P(i).*kfun(x, x);
    Y(i) = c1./c2;
end
return

function [X Y] = frur_mk(a, b, nkf, n, k_dkfun, f_dffun, Y, e, mki, m)
% Розв'язок нелінійного інтегрального рівняння
% типу Фредгольма-Урисона методом заміни
% інтеграла скінченною сумою
% 
$$\int_a^b K(x,s,u(s))ds = f(x,u(x)), \quad x \in [a,b].$$

%
% вхідні дані :
% a,b - нижня і верхня границя інтеграла;
% nkf - 0 = КФ методу трапецій,
%       1 = КФ методу Сімпсона;
% n - число точок сітки. Повинно задовольняти умові:
%       n > 1, при nkf=0,
%       n > 2 & mod(n,2)==1, при nkf=1;
% k_dkfun - функція з описом ядра K(x,s,y) і dK(x,s,u)/du;
% f_dffun - функція з описом правої частини f(x,y) і df(x,u)/du;
% Y - початкове наближення до розв'язку;
% e - точність (e>0);
% mki - максимальна кількість кроків ітерацій;
% m - метод розв'язку нелінійної системи:

```

```

%           m = 1 - градієнта;
%           m = 2 - Ньютона;
%           m = 3 - перші 4 ітерації по градієнтному, далі по Ньютона;
% вихідні дані:
% X       - масив вузлів сітки;
% Y       - масив наближеного розв'язку.
if nargin<10
    error('FRUR_MK:NotEnoughInputs',...
        'Not enough input arguments. See FRUR_MK. ');
    return
end
if a >= b
    error('FRUR_MK:error a >= b.',...
        'a >= b. See FRUR_MK. ');
    return
end
switch nkf
    case 0
        if n < 2
            error('FRUR_MK:error nkf = 0 & n < 2.',...
                'nkf = 0 & n < 2. See FRUR_MK. ');
            return
        end
    case 1
        if n < 3
            error('FRUR_MK:error nkf = 1 & n < 3.',...
                'nkf = 1 & n < 3. See FRUR_MK. ');
            return
        else
            if mod(n,2) == 0
                error('FRUR_MK:error nkf = 1 & mod(n,2) == 0.',...
                    'nkf = 1 & mod(n,2) == 0. See FRUR_MK. ');
                return
            end
        end
    otherwise
        error('FRUR_MK:Unknown method.',...
            'Unknown method. See FRUR_MK. ');
        return
end
if m ~= 1 & m ~= 2 & m ~= 3
    error('FRUR_MK:error m ~= [1,2,3].',...
        'Unknown method m (m ~e [1,2,3]). See FRUR_MK. ');
    return
end
% формування вузлів сітки
X = linspace(a,b,n); h = X(2)-X(1); n1 = n-1;
d = Y; F = Y; J = zeros(n,n);
% формування коефіцієнтів P(i), i=1,...,n КФ
P = Y;
switch nkf
    case 0
        P(1) = 0.5.*h;
        for i=2:n1
            P(i) = h;
        end
        P(n) = P(1);
    case 1
        P(1) = h./3; c2 = 2.*P(1); c1 = 2.*c2;

```

```

        for i=2:2:n1
            P(i) = c1; P(i+1) = c2;
        end
        P(n) = P(1);
    end
    clear c1 c2 h
    s = 0; Fl = 1;
    while Fl & s < mki
        % знаходження розв'язку нелінійної системи :
        % формування матриці Якобі J і вектора F
        for i = 1:n
            xi = X(i); sk = 0;
            for j = 1:n
                p = P(j);
                [k,dk] = k_dkfun(xi,X(j),Y(j));
                J(i,j) = p.*dk; sk = sk + p.*k;
            end
            [f,df] = f_dffun(xi,Y(i));
            J(i,i) = J(i,i) - df;
            F(i) = sk - f;
        end
        if m == 1 | m == 3 & s < 3
            % метод градієнта
            d = J' * F; v = J * d;
            sk = dot(F,v)./dot(v,v);
            d = sk.*d;
        else
            % метод Ньютона
            d = J\F;
        end
        Y = Y - d; s = s + 1
        sk = norm(d); Fl = sk > e;
    end
    clear d J F sk P xi p k dk f df
    Y = Y';
    return

```

Практичне заняття № 03

ТЕМА № 3. Методи розв'язування задачі Коші для звичайних диференціальних рівнянь.

3.1. Різницеві методи та методи Рунге-Кутта (1 год).

Теоретична інформація:

Необхідно наближено (використовуючи чисельні методи) розв'язати задачу Коші для системи ЗДР

$$\begin{cases} \frac{d\vec{u}(x)}{dx} = \vec{f}(x, \vec{u}(x)), & x \in (x_0, x_{end}] \\ \vec{u}(x_0) = \vec{u}_0. \end{cases} \quad (3.1)$$

В залежності від стратегії побудови методу, найчастіше використовують:

- c) *різницеві (однокрокові і багатокрокові);*
- d) *типу Рунге-Кутта.*

Як перші так і другі бувають *явні і неявні.*

a) Різницеві однокрокові

Розглянемо спочатку скалярний варіант (3.1). Вводимо по змінній x сітку з певним кроком (рівномірним або нерівномірним). Виконаємо апроксимацію лівої і правої частини рівняння (3.1) в певній точці *шаблону* $S\{x_n, x_{n+1}\}$. Шаблон – сукупність вузлів сітки, які використовуються для запису різницевої схеми.

Алгоритм явного методу Ейлера. Для апроксимації беремо точку x_n , використовуємо *праву різницю* для $u'(x_n)$:

$$y_{n+1} = y_n + h_n \cdot f(x_n, y_n), \quad y_0 = u_0, \quad n = 0, 1, \dots \quad (3.2)$$

Алгоритм неявного методу Ейлера. Для апроксимації беремо точку x_{n+1} , використовуємо *ліву різницю* для $u'(x_{n+1})$:

$$y_{n+1} = y_n + h_n \cdot f(x_{n+1}, y_{n+1}), \quad y_0 = u_0, \quad n = 0, 1, \dots \quad (3.3)$$

Для розв'язання нелінійного рівняння

$$F(y_{n+1}) = 0, \quad (3.4)$$

де $F(\xi) = \xi - h_n f(x_{n+1}, \xi) - y_n$, використаємо ІМ Ньютона.

Алгоритм "симетрична схема". Для апроксимації беремо додаткову точку $x_{n+0.5} = 0.5(x_n + x_{n+1})$, використовуємо *центральну різницю* для $u'(x_{n+0.5})$ і середнє значення для $f(x_{n+0.5}, u_{n+0.5})$:

$$y_{n+1} = y_n + 0.5h_n \cdot (f(x_n, y_n) + f(x_{n+1}, y_{n+1})), \quad y_0 = u_0, \quad n = 0, 1, \dots \quad (3.5)$$

Тут теж маємо рівняння (3.4), де $F(\xi) = \xi - 0.5h_n (f(x_n, y_n) + f(x_{n+1}, \xi)) - y_n$.

Для автоматичного вибору кроку h_n використовують *метод Рунге*, при цьому враховуємо, що точність методів (3.2)-(3.3) – $O(h)$, а методу (3.5) – $O(h^2)$.

b) Методи типу Рунге-Кутта

Характерною особливістю цих методів є обчислення правої частини (3.1) не тільки в точках сітки, але і в деяких проміжних точках відрізка $[x_n, x_{n+1}]$, при цьому використовуються спеціальні квадратурні формули

$$y_{n+1} - y_n = \int_{x_n}^{x_{n+1}} f(\xi, u(\xi)) d\xi = h_n \sum_{i=1}^m A_i \cdot K_i, \quad y_0 = u_0, \quad n = 0, 1, \dots$$

$m = 2$ – сім'я методів Рунге-Кутта 2-го порядку точності:

$$A_1 = 1 - \gamma, \quad A_2 = \gamma, \\ K_1 = f(x_n, y_n), \quad K_2 = f\left(x_n + \frac{h_n}{2\gamma}, y_n + \frac{h_n}{2\gamma} K_1\right), \quad \gamma \in (0, 1]. \quad (3.6)$$

Найчастіше використовуються такі значення : $\gamma = 0.5, \quad \gamma = 1$.

$m = 4$ – найпопулярніший з методів Рунге-Кутта 4-го порядку точності:

$$A_1 = \frac{1}{6}, \quad A_2 = \frac{1}{3}, \quad A_3 = \frac{1}{3}, \quad A_4 = \frac{1}{6}, \\ K_1 = f(x_n, y_n), \quad K_2 = f(x_n + 0.5h_n, y_n + 0.5h_n K_1), \\ K_3 = f(x_n + 0.5h_n, y_n + 0.5h_n K_2), \quad K_4 = f(x_{n+1}, y_n + h_n K_3). \quad (3.7)$$

Аудиторне завдання:

п.1. Розв'язати задачу Коші

$$\begin{cases} u'(x) = \exp(-x) - u(x), \\ u(0) = 1. \end{cases} \quad (3.8)$$

на відрізку $(0, 5]$ різницеvim методом ($h=0.01$) без автоматичного вибору кроку, використовуючи:

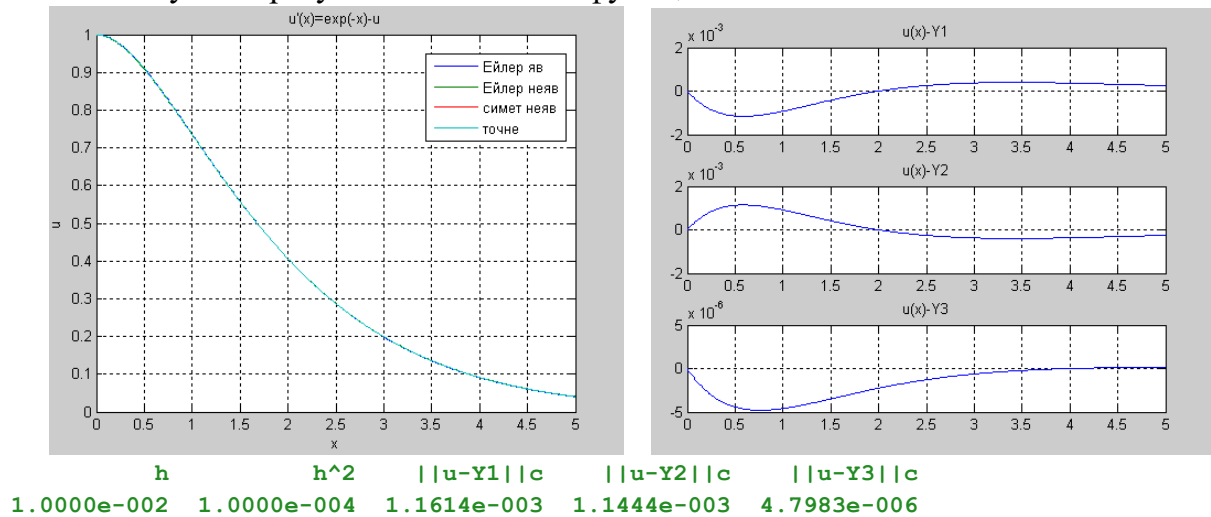
- ✓ явний метод Ейлера;
- ✓ неявний метод Ейлера;
- ✓ симетричну схему.

Порівняти точний $u(x) = \exp(-x)(x+1)$ і наближений розв'язок. Знайти теоретичні оцінки похибки і порівняти їх з практикою. Створимо відповідну М-функцію:

```
function P03a01
clear all, close all, clc
F = @(x,U) ([exp(-x)-U(1)]);
Y0 = [1]; % вектор початкових умов
X = linspace(0,5,501); % рівномірна сітка з h=0.01
% розв'язання системи :
Y1 = Euler_ev(F, X, Y0); % явний Ейлера
[Y2, efe] = Euler_nev(F, X, Y0); % неявний Ейлера
if efe ~= 1
    efe
    return
end
[Y3, efs] = symm_nev(F, X, Y0); % симетрична схема
if efs ~= 1
    efs
    return
end
u = @(x) (exp(-x).*(1+x)); Z = u(X);
plot(X, Y1, X, Y2, X, Y3, X, Z), grid on
title('u''(x)=exp(-x)-u'), xlabel('x'), ylabel('u'),
legend('Ейлер яв','Ейлер неяв',...
'sимет неяв','точне','Location','Best')
pause(5), close all
Z1 = Z-Y1; Z2 = Z-Y2; Z3 = Z-Y3;
subplot(3,1,1); plot(X, Z1); grid on; title('u(x)-Y1'); hold on;
subplot(3,1,2); plot(X, Z2); grid on; title('u(x)-Y2'); hold on;
subplot(3,1,3); plot(X, Z3); grid on; title('u(x)-Y3'); hold off;
h = X(2)-X(1); hh = h.*h;
d1 = norm(Z1,inf); d2 = norm(Z2,inf); d3 = norm(Z3,inf);
fr = '%12s %12s %12s %12s %12s\n';
fprintf(fr, 'h', 'h^2', '||u-Y1||c', '||u-Y2||c', '||u-Y3||c');
fr = '%12.4e %12.4e %12.4e %12.4e %12.4e\n';
fprintf(fr, h, hh, d1, d2, d3);
```

end

Маємо наступний результат виконання функції *P03a01* :



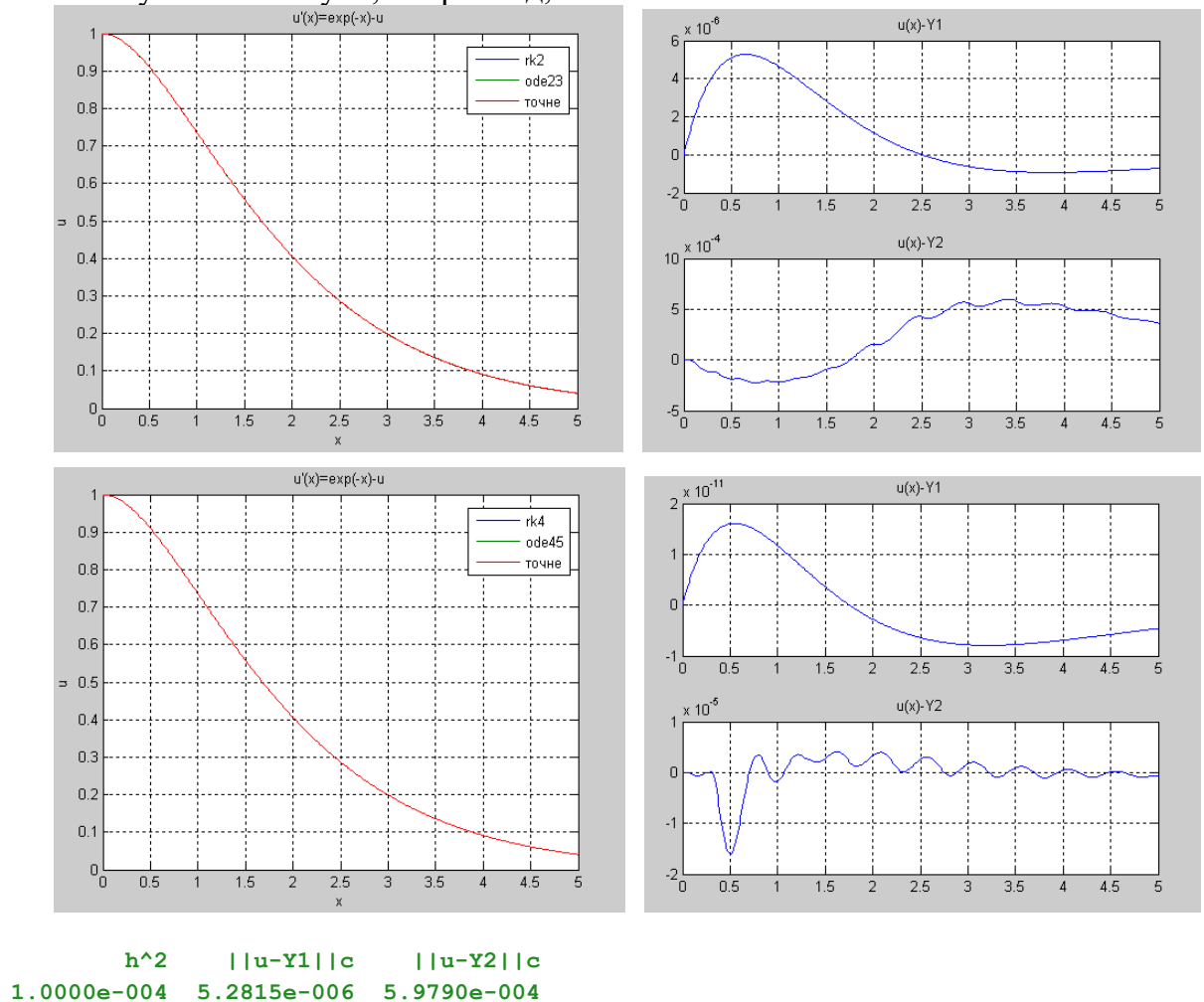
Завдання для самостійної роботи:

п.1. Розв'язати задачу Коші (3.8) на відрізку (0, 5] різницеvim методом ($h=0.01$) без автоматичного вибору кроку, використовуючи:

- ✓ метод Рунге-Кутта 2-го порядку точності (схема предиктор-коректор);
- ✓ метод Рунге-Кутта 4-го порядку точності;

Порівняти точний і наближені розв'язки отримані Вами, а також наближені розв'язки отримані за допомогою стандартних функцій MATLAB: *ode23*, *ode45* .

Результат має бути, наприклад, таким:



h^4 $||u-Y1||_C$ $||u-Y2||_C$
 1.0000e-008 1.6083e-011 1.5981e-005

3.2. Методи Адамса. Стійкість методів. Поняття жорстких СДР (1 год).

Теоретична інформація:

Продовжимо розглядати різницеві методи розв'язання задачі Коші для скалярного варіанту ЗДР (3.1), але тепер багатокрокові.

Різницеві m -крокові методи отримують із апроксимації $\{u'(x)|f(x,u)|u'(x)\&f(x,u)\}$ на шаблоні $S\{x_{n-m}, \dots, x_{n-1}, x_n\}$. В загальному випадку маємо:

$$\begin{cases} \frac{a_0 y_n + a_1 y_{n-1} + \dots + a_m y_{n-m}}{h} = \sum_{k=0}^m b_k f_{n-k}, \\ y_{n-m} = \mathcal{Y}_1, y_{n-m+1} = \mathcal{Y}_2, \dots, y_{n-1} = \mathcal{Y}_m, \end{cases} \quad (3.9)$$

де $a_0 \neq 0$, $\sum_{i=0}^m b_i = 1$ – умова апроксимації правої частини $f(x, u)$, $\mathcal{Y}_i$ – відомі константи.

Якщо $b_0 \neq 0$, то методи *явні*, якщо ж $b_0 = 0$, то *неявні*. У випадку $a_0 = -a_1 = 1$, $a_k = 0, k = 2, 3, \dots, m$ – *методи Адамса*.

Алгоритми явного/неявного методу Адамса. Інтегруємо (3.1) на інтервалі $[x_{n-1}, x_n]$:

$$y_n - y_{n-1} = \int_{x_{n-1}}^{x_n} f(\xi, u(\xi)) d\xi \equiv \int_{x_{n-1}}^{x_n} F(\xi) d\xi \quad (3.10)$$

Якщо нам відомі значення $\{y_i\}$ у вузлах сітки $\{x_{n-m}, \dots, x_{n-1}, x_n\}$, то відомі і значення функції $F(\xi)$ в цих вузлах, а отже можна побудувати інтерполяційний поліном (в формі Лагранжа чи Ньютона) за значенням $\{x_{n-j}, F(x_{n-j})\}_{j=0}^m$, відповідно інтеграл в (3.10) можна легко обчислити. Враховуючи ці обставини, запишемо інформацію про методи Адамса для **двох** точок в шаблоні S (рівномірна сітка з кроком h) в наступній таблиці.

Таблиця 3.1

Шаблон; Назва	$S[x_{n-2}, x_{n-1}];$ Явний (екстраполяційний)	$S[x_{n-1}, x_n];$ Неявний (інтерполяційний)
ІП для F	$F(\xi) = F_{n-1} + (\xi - x_{n-1})F_{n-1;n-2}$	$F(\xi) = F_n + (\xi - x_n)F_{n;n-1}$
Розрахункова формула	$y_n = y_{n-1} + 0.5h(-f_{n-2} + 3f_{n-1})$	$y_n = y_{n-1} + 0.5h(f_{n-1} + f_n)$
Похибка; Порядок через значення m , враховуючи $\sum_{k=0}^m b_k f_{n-k}$	$O(h^2);$ $2 = m$	$O(h^2);$ $2 = m+1$

Стійкість і збіжність різницевих m -крокових методів

Розглянемо для (3.9) однорідне рівняння

$$a_0 v_n + a_1 v_{n-1} + \dots + a_m v_{n-m} = 0. \quad (3.11)$$

Його розв'язок будемо шукати у вигляді $v_k = q^k$ (якщо деякі корені є кратні, то розв'язок буде $v_k = k^j q^k, j = 0, 1, \dots, r_k - 1$, де r_k – кратність кореня). Для q отримаємо *характеристичне рівняння*:

$$a_0 q^m + a_1 q^{m-1} + \dots + a_{m-1} q + a_m = 0. \quad (3.12)$$

Озн.1 Метод (3.9) задовольняє умові коренів, якщо $|q_i| \leq 1, i = 1, 2, \dots, m$, причому $|q_{i_0}| = 1$ не є кратним.

Озн.2 Метод (3.9), при виконанні умови коренів є *стійким*.

Стійкість поділяють на *стійкість за початковими даними* і *стійкість за правою частиною*, але умова коренів є основним чинником відповідних означень [33].

Якщо виконуються умови Теор.2 [33], то $|z_j| = |y_j - u(x_j)| \rightarrow 0$ при $h \rightarrow 0$ і *порядок точності* методу (3.9) співпадає з порядком апроксимації на розв'язку вихідного рівняння.

Стійкість умовна і абсолютна. Жорсткість систем ЗДР

Якщо розглядати модельну задачу Коші для (3.1) з $f(x, u(x)) = \lambda \cdot u(x)$, $x > 0$, $\lambda < 0$, то стійкість означає виконання нерівності $|u(x+h)| \leq |u(x)|$, $h > 0$.

Використовуючи різницевий метод, потрібно вимагати, щоб для його розв'язку виконувалась нерівність, аналогічна диференціальному розв'язку, тобто була *асимптотична стійкість*.

Озн.3 Різницевий метод є *абсолютно стійкий*, якщо він стійкий при $\forall h > 0$, і *умовно стійкий*, якщо він стійкий при певних обмеженнях на крок h .

Розглянемо Озн.3 для розглянутих нами різницеви методів з урахуванням модельної задачі Коші.

а) Для явного методу Ейлера (3.2): $|y_{n+1}| \leq |1 + \lambda h| |y_n|$ і отримаємо:

$$0 < h \leq \frac{2}{|\lambda|}. \quad (3.13)$$

б) Аналогічне обмеження на крок можна отримати для методу Рунге-Кутта 2-го порядку точності ($\gamma=1$), а для 4-го – $|\lambda| \cdot h \leq 2.78$.

в) Для неявного методу Ейлера (3.3): $|y_{n+1}| \leq \left| \frac{1}{1 - \lambda h} \right| |y_n|$ і маємо завжди $\frac{1}{1 - \lambda h} < 1$ при $0 < h \cdot \lambda < 1$.

Із наведених прикладів бачимо, що явні методи умовно стійкі, а неявні – безумовно стійкі, але вимагають розв'язання нелінійних рівнянь.

Розглянуті методи без зусиль переносяться на системи ЗДР, але виникає проблема зі стійкістю (обмеження типу (3.13)), пов'язана з різномасштабністю процесів, які описуються системою ЗДР. Простий приклад :

$$\begin{cases} \frac{du_1(x)}{dx} + a_1 u_1(x) = 0, \\ \frac{du_2(x)}{dx} + a_2 u_2(x) = 0. \end{cases}$$

$x > 0; \quad a_{1,2} > 0, \quad a_2 \gg a_1.$

Або у векторній формі запису :

$$\frac{d\vec{u}(x)}{dx} = A \cdot \vec{u}(x), \quad A = \begin{pmatrix} -a_1 & 0 \\ 0 & -a_2 \end{pmatrix}, \quad \vec{u}(x) = \begin{pmatrix} u_1(x) \\ u_2(x) \end{pmatrix}. \quad (3.14)$$

Враховуючи, що $a_{1,2} > 0$, $a_2 \gg a_1$, то починаючи з деякого x , ми повинні мати (аналітично) $\vec{u}(x) = \begin{pmatrix} u_1(x) \\ u_2(x) \end{pmatrix} \approx \begin{pmatrix} u_1(x) \\ 0 \end{pmatrix}$. Проте чисельно, якщо використовувати явний метод Ейлера, ми маємо обмеження на крок у вигляді нерівності (3.13), що приводить до $h \cdot a_2 \leq 2$.

Озн.4 (A – стала, квадратна матриця розмірності N) Система (3.14) називається *жорсткою*, якщо:

3) $\text{Re}(\lambda_k) < 0, k = 1, 2, \dots, N$ – система асимптотично стійка за Ляпуновим;

$$4) \quad S = \frac{\max_k |\operatorname{Re}(\lambda_k)|}{\min_k |\operatorname{Re}(\lambda_k)|} - \text{число жорсткості системи велике.}$$

Означення жорсткості можна поширити на лінійні системи ЗДР, в яких матриця A залежить від x , а також на нелінійні системи (3.1). Крім цього можуть визначатись спеціальні умови стійкості: A – *стійкість* і $A(\alpha)$ – *стійкість* [33].

Аудиторне завдання:

п.2. Розв'язати задачу Коші (3.8) на відрізку $(0, 20]$ з $h=0.01$ без автоматичного вибору кроку, використовуючи:

✓ Симетричну явну схему порядку $O(h^2)$

$$\begin{cases} \frac{y_n - y_{n-2}}{2h} = f(x_{n-1}, y_{n-1}), \\ y_1 \equiv u_0, \quad y_2; \quad n = 3, 4, \dots, N. \end{cases}$$

✓ явний метод Адамса порядку $O(h^2)$.

Для знаходження y_2 використати метод Рунге-Кутта порядку $O(h^2)$.

Порівняти точний $u(x) = \exp(-x)(x+1)$ і наближений розв'язок. Знайти теоретичні оцінки похибки і порівняти їх з практикою.

п.3. Перевірити виконання умови коренів для різницевого методу порядку $O(h^2)$

$$\begin{cases} \frac{3y_n - 4y_{n-1} + y_{n-2}}{2h} = f(x_n, y_n), \\ y_1 \equiv u_0, \quad y_2; \quad n = 3, 4, \dots, N. \end{cases}$$

Створимо відповідну М-функцію :

```
function P03a023
clear all, close all, clc

% п.2
F = @(x,U) ([exp(-x)-U(1)]);
Y0 = 1; % вектор початкових умов
h = 0.01; X = [0, h];
Y1 = rk2_ev(F, X, Y0, 1); % 2-стадійний Р-К на сітці X
N = 2001; X = linspace(0, 20, N); % рівномірна сітка з h=0.01
% розв'язання :
Y1 = [Y1, X(3:N)]; Y2 = Y1; % значення в т. X(1), X(2) + "заготовки"
fn1 = F(X(1), Y2(1)); h2 = 2.*h; h05 = 0.5.*h;
for n = 3 : N
    n2 = n - 2; n1 = n - 1; x = X(n1);
    Y1(n) = Y1(n2) + h2.*F(x, Y1(n1)); % симетр. явний O(h^2)
    fn2 = fn1; fn1 = F(x, Y2(n1));
    Y2(n) = Y2(n1) + h05.*(3.*fn2 - fn1); % явний м. Адамса O(h^2)
end
u = @(x) (exp(-x).*(1+x)); Z = u(X);
plot(X, Y1, X, Y2, X, Z), grid on
title('u'(x)=exp(-x)-u'), xlabel('x'), ylabel('u'),
legend('симетр яв', 'Адамс2 яв', 'точне', 'Location', 'Best')
pause(5), close all
Z1 = Z-Y1; Z2 = Z-Y2;
subplot(2,1,1); plot(X, Z1); grid on; title('u(x)-Y1'); hold on;
subplot(2,1,2); plot(X, Z2); grid on; title('u(x)-Y2'); hold off;
hh = h.*h;
d1 = norm(Z1, inf); d2 = norm(Z2, inf);
```

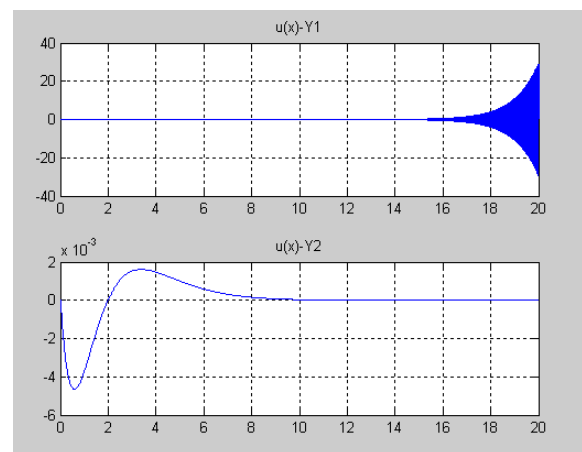
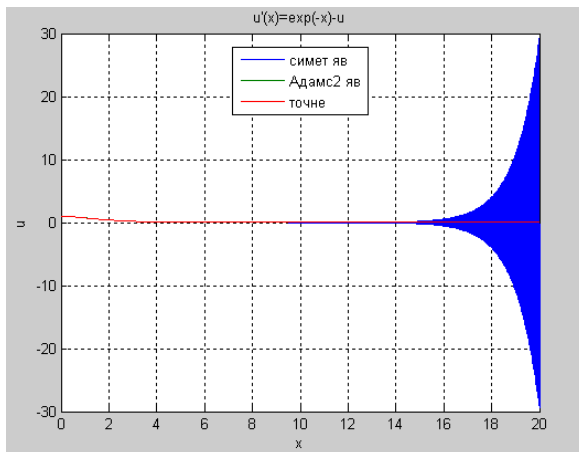
```

fr = '%12s %12s %12s\n';
fprintf(fr,'h^2','||u-Y1||c','||u-Y2||c');
fr = '%12.4e %12.4e %12.4e\n';
fprintf(fr,hh,d1,d2);
pause(5), close all

% п.3
clear all
syms q
Q = solve('3/2*q^2-2*q+1/2',q);
Q
End

```

Маємо наступний результат виконання функції P03a023:



```

      h^2      ||u-Y1||c      ||u-Y2||c
1.0000e-004  2.9554e+001  4.6511e-003
Q =
     1
    1/3

```

Завдання для самостійної роботи:

п.2. Отримати інформацію про методи Адамса для кількості **трьох** точок в шаблоні S (рівномірна сітка з кроком h) у вигляді таблиці, аналогічній Таб.3.1.

п.3. Розв'язати задачу Коші (3.8) на відрізку $(0, 20]$ з $h=0.01$ без автоматичного вибору кроку, використовуючи неявний метод Адамса порядку $O(h^2)$. Порівняти точний і отриманий Вами наближений розв'язок.

п.4. Перевірити виконання умови коренів для різницевого методу порядку $O(h^3)$

$$\begin{cases} \frac{11y_n - 18y_{n-1} + 9y_{n-2} - 2y_{n-3}}{6h} = f(x_n, y_n), \\ y_1 \equiv u_0, y_2, y_3; \quad n = 4, 5, \dots, N. \end{cases}$$

п.5. Розв'язати задачу Коші

$$\begin{cases} \frac{d\bar{u}(x)}{dx} = \bar{f}(x, \bar{u}), \quad x \in (0, 55], \\ \bar{u}(0) = [1, 0, 0]^T. \end{cases}$$

де

$$\bar{f}(x, \bar{u}) = [B - A, A - B - C, C]^T;$$

$$A = (0.1 + 0.01/k) * u_1(x), \quad B = (100 + 1/g) * u_2(x) * u_3(x), \quad C = (1000 + 1/d) * u_2(x).$$

Програмне забезпечення на MATLAB:

function Y = Euler_ev(F, X, U0)

```
% Розв'язок  $U'(x) = F(x, U(x))$ ,  $x \in X$ ,  
%  $U(X(1))=U0$ ,  
% явним м.Ейлера на фіксованій сітці X.  
nX = length(X); N = nX-1;  
[n, m] = size(U0);  
if n > m  
    Y = [U0, zeros(n, N)];  
else  
    Y = [U0', zeros(m, N)];  
end  
for k = 1 : N  
    kp = k+1; x = X(k); h = X(kp) - x;  
    z = Y(:, k); Y(:, kp) = z + h.*F(x, z);  
end  
end
```

function [Y, exitf] = Euler_nev(F, X, U0)

```
% Розв'язок  $U'(x) = F(x, U(x))$ ,  $x \in X$ ,  
%  $U(X(1))=U0$ ,  
% неявним м.Ейлера на фіксованій сітці X.  
nX = length(X); N = nX-1;  
[n, m] = size(U0);  
if n > m  
    Y = [U0, zeros(n, N)];  
else  
    Y = [U0', zeros(m, N)];  
end  
opt = optimset('Display', 'off', 'TolX', 1e-12, 'TolFun', 1e-12);  
for k = 1 : N  
    kp = k+1; x = X(kp); h = x - X(k);  
    z = Y(:, k); Y0 = z;  
    [YY, FX, exitf, output] = fsolve(@(T) (T-z-h.*F(x, T)), Y0, opt);  
    if exitf ~= 1  
        [k, x, YY, FX, exitf, output.iterations]  
        return  
    end  
    Y(:, kp) = YY;  
end  
end
```

function [Y, exitf] = symm_nev(F, X, U0)

```
% Розв'язок  $U'(x) = F(x, U(x))$ ,  $x \in X$ ,  
%  $U(X(1))=U0$ ,  
% симетричним методом на фіксованій сітці X.  
nX = length(X); N = nX-1;  
[n, m] = size(U0);  
if n > m  
    Y = [U0, zeros(n, N)];  
else  
    Y = [U0', zeros(m, N)];  
end  
opt = optimset('Display', 'off', 'TolX', 1e-12, 'TolFun', 1e-12);  
for k = 1 : N
```

```

    kp = k+1; x = X(kp); h = 0.5.*(x - X(k));
    z = Y(:,k); Y0 = z; fz = h.*F(X(k),z);
    [YY,FX,exitf,output] = fsolve(@ (T) (T-z-fz-h.*F(x,T)),Y0,opt);
    if exitf ~= 1
        [k, x, YY, FX, exitf, output.iterations]
        return
    end
    Y(:,kp) = YY;
end
end

```

Практичне заняття № 04

ТЕМА № 4. Розв'язування крайових задач для ЗДР.

4.1. Метод редукції крайової задачі для ЗДР до задачі Коші (2 год).

Теоретична інформація:

Розглянемо загальну постановку лінійної крайової задачі для системи двох ЗДР першого порядку:

$$\bar{U}'(x) = A(x) * \bar{U}(x) + \bar{F}(x), \quad x \in (a, b), \quad (4.1a)$$

$$(D_1 \ D_2) * \bar{U}(a) = D_3, \quad (4.1б)$$

$$(D_4 \ D_5) * \bar{U}(b) = D_6. \quad (4.1в)$$

де уведено позначення

$$\bar{U}(x) = \begin{pmatrix} u_1(x) \\ u_2(x) \end{pmatrix}, \quad A(x) = \begin{pmatrix} a_{1,1}(x) & a_{1,2}(x) \\ a_{2,1}(x) & a_{2,2}(x) \end{pmatrix}, \quad \bar{F}(x) = \begin{pmatrix} f_1(x) \\ f_2(x) \end{pmatrix}.$$

Розглянемо числовий метод, який дозволяє звести розв'язання крайової задачі (4.1) до розв'язання серії задач Коші, тому він і має назву *метод редукції до задачі Коші*. По аналогії з артилерійською стрільбою за участю корегувальника вогню, він також має найменування *метод стрільби* або *балістичний* [6, 7, 11].

Алгоритм методу стрільби.

«Прямий» хід :

1. Покладемо в (4.1б) величину $u_1(a)$ (або $u_2(a)$) рівною деякій константі η , тоді можемо визначити значення $u_2(a)$ (або $u_1(a)$) і таким чином отримуємо задачу Коші для системи ЗДР (4.1а) з початковими умовами (4.2) :

$$\bar{U}(a) = \begin{pmatrix} \eta \\ (D_3 - D_1\eta)/D_2 \end{pmatrix} \text{ або } \bar{U}(a) = \begin{pmatrix} (D_3 - D_2\eta)/D_1 \\ \eta \end{pmatrix}. \quad (4.2)$$

2. Користуючись деяким чисельним методом, наприклад Рунге-Кутта, розв'язуємо задачу Коші (4.1а), (4.2) і знаходимо розв'язок $\bar{U}(x; \eta)$ – як функцію від x при параметрі η , а отже і значення $\bar{U}(b; \eta)$.
3. Підставимо знайдене значення $\bar{U}(b; \eta)$ в умову (4.1в), отримаємо

$$\psi(\eta) \equiv (D_4 \ D_5) * \bar{U}(b; \eta) - D_6 \neq 0.$$

Отже, потрібно знайти таке $\tilde{\eta}$, що є коренем алгебраїчного рівняння $\psi(\eta) = 0$, тобто задовольняє як крайову умову (4.2), так і умову (4.1в). Враховуючи лінійність задачі (4.1), будемо мати лінійну залежність її розв'язку від параметра η , а отже функція $\psi(\eta)$ лінійна. Для знаходження $\tilde{\eta}$ – кореня лінійного рівняння $\psi(\eta) = 0$ достатньо мати дві точки $(\eta_1, \psi(\eta_1))$, $(\eta_2, \psi(\eta_2))$, де $\eta_1 \neq \eta_2$.

4. Аналогічно п.1, задаємо значення $\eta = \eta_2$, розв'язуємо ще одну задачу Коші і отримуємо нове значення $\psi(\eta_2)$.

«Зворотній» хід :

1. Знаходимо $\tilde{\eta}$ по формулі (метод хорд для рівняння $\psi(\eta) = 0$)

$$\tilde{\eta} = \eta_2 - \frac{(\eta_2 - \eta_1)\psi(\eta_2)}{\psi(\eta_2) - \psi(\eta_1)}. \quad (4.3)$$

2. Будуємо початкові умови типу (4.2) з $\eta = \tilde{\eta}$.
3. Знаходимо розв'язок $\bar{U}(x; \tilde{\eta})$ ще однієї задачі Коші, який задовольняє умови (4.1б) і (4.1в).

Для лінійної задачі (4.1), серію з трьох розв'язків задачі Коші можна скоротити до двох розв'язків і деякого алгебраїчного перерахунку, якщо скористатись теоретичними знаннями з курсу "Диференціальні рівняння". Дійсно, розв'язок $\bar{U}(x)$ можна подати як суму розв'язків $\bar{W}(x)$ – загального розв'язку однорідної задачі, і $\bar{V}(x)$ – частинного розв'язку неоднорідної задачі:

$$\bar{U}(x) = c * \bar{W}(x) + \bar{V}(x). \quad (4.4)$$

Тут $\bar{W}(x)$ – розв'язок однорідної задачі при $\eta = \eta_1 = 1$

$$\begin{cases} \bar{W}'(x) = A(x) * \bar{W}(x), & x \in (a, b], \\ \bar{W}(a) = \begin{pmatrix} 1 \\ -D_1/D_2 \end{pmatrix} \hat{=} \begin{pmatrix} -D_2/D_1 \\ 1 \end{pmatrix}, \end{cases} \quad (4.5)$$

а $\bar{V}(x)$ – розв'язок неоднорідної задачі при $\eta = \eta_2 = 0$

$$\begin{cases} \bar{V}'(x) = A(x) * \bar{V}(x) + \bar{F}(x), & x \in (a, b], \\ \bar{V}(a) = \begin{pmatrix} 0 \\ D_3/D_2 \end{pmatrix} \hat{=} \begin{pmatrix} D_3/D_1 \\ 0 \end{pmatrix}. \end{cases} \quad (4.6)$$

І в цьому випадку не потрібно розв'язувати (4.3), адже коефіцієнт c знаходимо із крайової умови в точці $x = b$

$$c = \frac{D_6 - D_5 V_1(b) - D_4 V_2(b)}{D_5 W_1(b) + D_4 W_2(b)}. \quad (4.7)$$

Для практичного застосування балістичного методу, з метою зменшення об'єму оперативної пам'яті, вводиться сітка $\bar{X} = (X_i)_{i=1}^N$ – точки виведення результатів, в яких запам'ятовуються вектори $\{\bar{W}(X_i), \bar{V}(X_i)\}$ при розв'язанні задач (4.5), (4.6) і розв'язок $\bar{U}(X_i)$ алгебраїчно розраховується по формулі (4.4) і значенню коефіцієнта c , отриманого в (4.7).

Розглянутий нами алгоритм балістичного методу реалізовано (див. програмне забезпечення) у вигляді М-функції

$$U = \text{firlin}(X, A, F, D)$$

де позначення формальних вхідних і вихідних параметрів функції співпадають з позначеннями постановки крайової задачі (4.1).

Схему застосування методу стрільби можна успішно використовувати і для систем виду (4.1а) з кількістю рівнянь $n > 2$. В цьому випадку крайові умови (4.1б), (4.1в) мають вид

$$L * \bar{U}(a) = \bar{I},$$

$$R * \bar{U}(b) = \bar{r},$$

де матриці L, R та вектори $\bar{I}, \bar{r}$ – відомі

$$L = (L_{i,j})_{i,j=1}^{m,n}, \quad \bar{I} = (I_i)_{i=1}^m, \quad R = (R_{i,j})_{i,j=1}^{n-m,n}, \quad \bar{r} = (r_i)_{i=1}^{n-m},$$

а також передбачається, що $\text{rank}(L) = m$, $\text{rank}(R) = n - m$ і вектори $\bar{F}(x), \bar{I}, \bar{r}$ не всі одночасно рівні $\bar{0}$. Якщо серед компонент вектора $\bar{U}(x)$ є сильно зростаючі при зміні x , то для підвищення стійкості по відношенню до накопичення обчислювальної похибки, використовують алгоритм *методу стрільби з дискретною ортогоналізацією*.

Балістичний метод застосовують і для системи двох нелінійних рівнянь, а також для систем нелінійних рівнянь з $n > 2$, але для випадку «пристрільки» по

одному параметру, тобто коли на одному з кінців відрізка $[a, b]$ маємо значення $m=1$ або $n-m=1$.

На практиці досить часто виникають прикладні задачі, які моделюються за допомогою крайової задачі для лінійного ЗДР другого порядку і записуються у вигляді [33]:

$$\begin{cases} \frac{d}{dx} \left(K(x) \frac{du(x)}{dx} \right) - p(x)u(x) = -f(x), \\ K(x) \geq K_0 > 0, \quad p(x) \geq 0, \quad x \in (a, b); \\ K(a) \frac{du(a)}{dx} = \beta_1 u(a) - \mu_1, \\ -K(b) \frac{du(b)}{dx} = \beta_2 u(b) - \mu_2; \\ (\beta_1 \geq 0) \& (\beta_2 \geq 0) \& (\beta_1 + \beta_2 > 0). \end{cases} \quad (4.8)$$

Вважатимемо, що функції $K(x)$, $p(x)$, $f(x)$ неперервні і мають необхідну гладкість.

Якщо ввести наступні позначення

$$u_1(x) = u(x), \quad u_2(x) = u'(x); \quad \bar{U}(x) = \begin{pmatrix} u_1(x) \\ u_2(x) \end{pmatrix};$$

$$A(x) = \begin{pmatrix} 0 & 1 \\ p(x)/K(x) & -K'(x)/K(x) \end{pmatrix}; \quad \bar{F}(x) = \begin{pmatrix} 0 \\ -f(x)/K(x) \end{pmatrix}; \quad D = [\beta_1, -K(a), \mu_1, \beta_2, K(b), \mu_2],$$

то (4.8) буде записано у векторно-матричній формі (4.1), а отже можна застосувати балістичний метод і відповідну функцію *firlin* для знаходження числового наближеного розв'язку.

Аудиторне завдання:

п.1. Розв'язати крайову задачу для лінійного ЗДР другого порядку методом редукції до задачі Коші

$$\begin{cases} u''(x) - \frac{1}{x} u'(x) = \frac{3}{x^3}, \quad x \in (1, 2), \\ u(1) = u'(1), \quad 3u(2) - 2u'(2) = 3. \end{cases} \quad (4.9)$$

з кроком виведення $h=0.02$. В функції **firlin** використати солвер **ode45** для розв'язання задачі Коші. Розглянути і інші варіанти комбінації крайових умов I, II, III роду. Побудувати відповідні графіки точного $u(x) = x^2 + \frac{1}{x} - 1$ і отриманого наближеного розв'язку (для функції і похідної), а також графіки абсолютної похибки.

Для використання програми **firlin** приведемо постановку задачі (4.9) до форми (4.1):

$$\begin{cases} \underbrace{\begin{pmatrix} u_1'(x) \\ u_2'(x) \end{pmatrix}}_{\bar{U}'(x)} = \underbrace{\begin{pmatrix} 0 & 1 \\ 0 & x^{-1} \end{pmatrix}}_{A(x)} * \underbrace{\begin{pmatrix} u_1(x) \\ u_2(x) \end{pmatrix}}_{\bar{U}(x)} + \underbrace{\begin{pmatrix} 0 \\ 3x^{-3} \end{pmatrix}}_{\bar{F}(x)}, \quad x \in (1, 2), \\ (1 \quad -1) * \bar{U}(1) = 0, \quad (3 \quad -2) * \bar{U}(2) = 3. \end{cases} \quad (4.9^*)$$

Тепер можна створити М-функцію *p04a01d* для розв'язання поставленої задачі (і передбачити завдання різних комбінацій крайових умов I, II, III роду):

```
function p04a01d
% Модельна КЗ для ЗДР:
% u''(x) - 1/x*u'(x) = 3/x^3, x в (1,2);
% точний розв'язок: u(x) = x^2+1/x-1.
% Варіанти крайових умов :
```

```

% u(1)= 1,          u(2)=3.5.          (B.1)
% u(1)-u'(1)=0, 3u(2)-2u'(2)=3. (B.2)
% u(1)= 1,          u'(2)=3.75.        (B.3)
% u'(1)=1,          u(2)=3.5.          (B.4)
% 3u(1)-u'(1)=2, u'(2)=3.75.          (B.5)
% u'(1)=1,          -u(2)+2u'(2)=4. (B.6)
% u'(1)=1,          u'(2)=3.75.        (B.7 !)

clear all, close all, clc

% вхідні дані
a = 1; b = 2;
%D = [1, 0, 1, 1, 0, 3.5];           % (B.1)
D = [1,-1, 0, 3,-2, 3];             % (B.2)
%D = [1, 0, 1, 0, 1, 3.75];         % (B.3)
%D = [0, 1, 1, 1, 0, 3.5];         % (B.4)
%D = [3,-1, 2, 0, 1, 3.75];         % (B.5)
%D = [0, 1, 1,-1, 2, 4];           % (B.6)
%D = [0, 1, 1, 0, 1, 3.75];         % (B.7 !)

% формування сітки виведення результатів
n = 51; X = linspace(a, b, n); I = [1:5:n];
U = firlin(X, @fA, @fF, D);

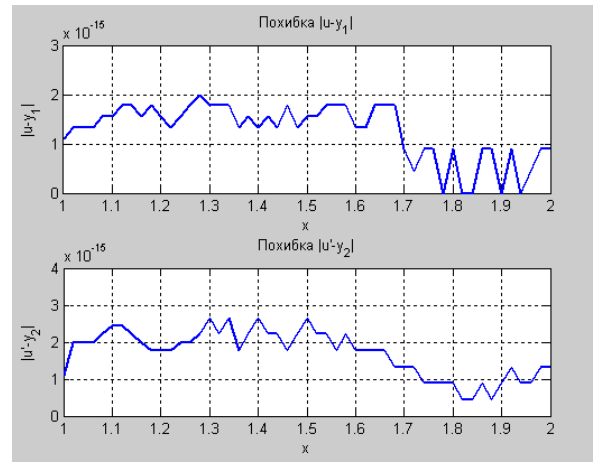
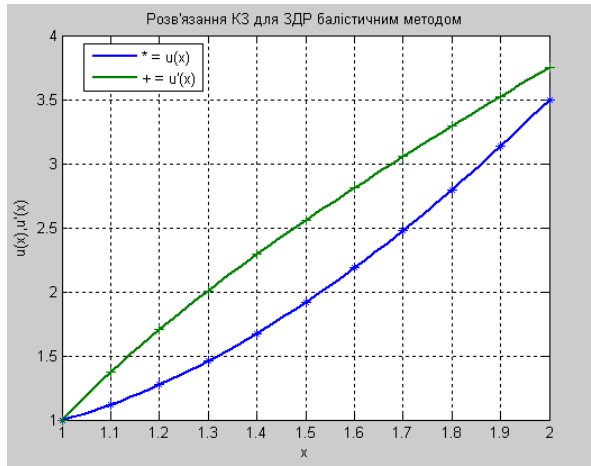
% порівняння наближеного і точного розв'язку
u = @(x) (x.*x+1./x-1);
du = @(x) (2.*x-1./x.^2);
Z = u(X); dZ = du(X);
plot(X, U, 'LineWidth', 2); hold on;
plot(X(I), Z(I), '*', X(I), dZ(I), '+'); hold off;
axis([1, 2, 1, 4]); grid on;
title('Розв'язання КЗ для ЗДР балістичним методом');
legend('* = u(x)', '+ = u''(x)', 'Location', 'Best');
xlabel('x'); ylabel('u(x),u''(x)');
pause(); close all;
subplot(2,1,1); plot(X,abs(Z-U(1,:)),'LineWidth',2);
title('Похибка |u-y_1|');
xlabel('x'), ylabel('|u-y_1|'); grid on;
subplot(2,1,2); plot(X,abs(dZ-U(2,:)),'LineWidth',2);
title('Похибка |u''-y_2|');
xlabel('x'); ylabel('|u''-y_2|'); grid on;
end

function P = fA(x)
% Опис матриці A(x)
P = [0, 1;...
     0, 1./x];
end

function P = fF(x)
% Опис вектор-функції F(x)
P = [0;...
     3./x.^3];
end

```

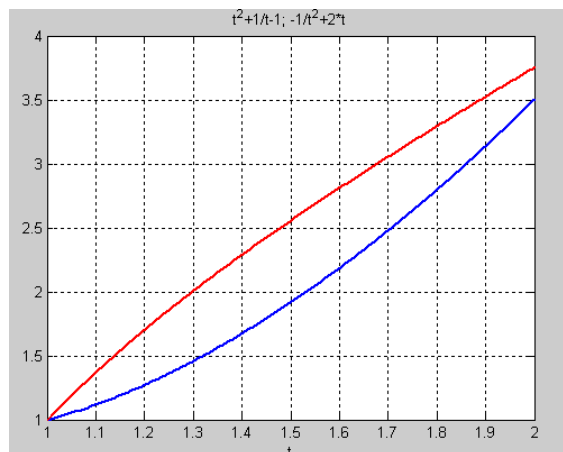
Результат виконання М-функції *P02a01d* (окрім варіанту 7 !) буде наступний:



Враховуючи, що комбінація КУ для варіанту 7 приводить до ситуації неоднозначного розв'язку (4.9), то розглянемо символічний розв'язок задачі (4.9) з КУ II роду в обох граничних точках. Створимо відповідний М-файл *p04a01s* :

```
% p04a01s.m
% Аналітичний розв'язок КЗ для ЗДР
%  $x''(t) - 1/t \cdot x'(t) = 3/t^3$ ,  $t \in (1,2)$ ,
%  $x'(1) = 1$ ,  $x'(2) = 15/4$ .
clear all, close all, clc
[x, y] = dsolve('Dx=y', 'Dy=y/t+3/t^3', 'y(1)=1', 'y(2)=15/4');
disp(['X(t)=', char(x)]);
x = subs(x, 'C1', '-1');
X = char(x);
y = expand(y); Y = char(y);
disp(['x(t)=', X]);
disp(['x''(t)=', Y]);
d = [1, 2, 1, 4];
f1 = ezplot(X, d); hold on;
f2 = ezplot(Y, d); grid on;
set(f1, 'color', 'blue', 'LineWidth', 2);
set(f2, 'color', 'red', 'LineWidth', 2);
title([X, ' ', Y]); hold off;
```

і отримаємо наступний результат його виконання :



```
x(t)=t^2+1/t+C1
x(t)=t^2+1/t-1
x'(t)=-1/t^2+2*t
```

Завдання для самостійної роботи:

п.1. Розв'язати крайову задачу (4.10) (параметри означені в Табл. 4.1)

$$\begin{cases} u''(x) - p(x)u(x) = -f(x), \\ p(x) \geq 0, \quad x \in (a, b); \\ D_1 u(a) + D_2 u'(a) = D_3, \\ D_4 u(b) + D_5 u'(b) = D_6. \end{cases} \quad (4.10)$$

методом редукції до задачі Коші з кроком виведення $h=0.02$. Використати функцію **firlin** і солвер MATLAB **ode45** для розв'язання задачі Коші з урахуванням варіанту індивідуального завдання ($k, p(x), f(x), a, b, D$). Тут через k позначено номер за списком студента у групі, g – номер групи, d – день народження студента. Порівняти отриманий наближений розв'язок з розв'язком (4.10) за допомогою солвера MATLAB **bvp4c**.

Таблиця 4.1 Варіанти індивідуального завдання.

k	$p(x)$	$f(x)$	a	b	D
1	$\sin^2(x+g)$	$1+gx+x^3$	0	1	[1,1,0,0,1,d]
2	$g \cos^2(x)$	$1+\exp(-gx)+2x^3$	-1	1	[1,0,d,1,1,1]
3	$(x+g)^2$	$x^3 \sin(1+gx)$	1	2	[1,1,d,0,1,d]
4	$g + \sin^2(\sqrt{2-x}-1)$	$1+g \exp(x+x^3)$	-1	0	[1,1,0,0,1,d]
5	$0.2 + \sin^2(1+gx)$	$x^3/(1+gx)$	1	2	[1,d,0,0,1,d]
6	$ch(x+g)$	$(1+gx)/(1-x^3)$	-2	0	[0,1,d,-1,1,0]
7	$th(x^2+g)$	$1+gx^3$	0	1	[0,1,d,1,1,d]
8	$g \cdot \arccos(x)$	$1+g \cdot x \cdot \sin(1+x^2)$	-1	1	[1,1,1,1,d,1]
9	$2g + \sin(x^2)$	$g \cdot x^2$	0	1	[0,1,d,1,1,1]
10	$\exp(g-x^2)$	$g \cdot th(1+x^2)$	1	2	[1,0,0,1,1,d]
11	$\sin^2(1+gx)$	$g \sin(x)+x^2$	-1	1	[1,1,d,1,1,d]
12	$1+gx+x^3$	$\sin^2(x+g)$	0	2	[0,1,1,1,0,d]
13	$1+th(gx+x^3)$	$\sin(x+g)$	0	1	[1,1,d,1,0,d]
14	$g \cdot ch(x)$	$g-x+x^3$	-1	1	[0,1,d,2,3,d]
15	$-th(1+g \cdot x+x^2)$	$g \cdot x^3$	-2	-1	[1,1,d,1,1,2]
16	$g+x^2$	$1+gx \exp(x^3)$	-1	0	[2,-1,d,0,1,d]
17	$g + \cos(x)$	$1+gx^2$	0	1	[1,1,d,1,1,d]
18	$(x^2+1)/(x^4+g)$	$g+x^3$	-1	1	[d,1,d,0,1,d]
19	$(x+g)/(x^2+g)$	$\sin(gx+1)$	1	2	[1,d,1,1,0,d]
20	$(x \cdot g)/(x^2+g)$	$gx+th(x^2)$	1	3	[0,d,1,d,0,1]
21	$\sin^2(x/g)$	$(1+gx)/(1-x^3)$	-1	0	[1,d,0,0,1,2]
22	$\cos^2(g \cdot x)$	$1+g(1+x)x$	0	1	[d,1,0,d,-1,2]
23	$g+x^2$	$1+g \cdot th(x)$	-1	1	[0,1,d,-d,1,1]
24	$ch(x+g)$	$g \cdot x(1+\sin(x))$	-1	2	[1,d,0,1,1,d]
25	$\sin^2(g+x^2)$	$g+x+\sqrt{x^3}$	0	2	[1,1,1,d,1,2]

26	$g \cdot x^2$	$1 + g\sqrt{x^2 - x}$	-2	0	[1,2,d,0,1,1]
27	$ch(g - x^2)$	$1 + gx^2$	0	1	[1,1,2,1,1,d]
28	$x + g$	$(1+x)/(1+gx+x^3)$	0	2	[1,0,2,1,0,d]
29	$x^4 + g$	$ch(g + x^3)$	-1	1	[d,1,1,1,0,1]
30	$g \cdot \cos(x)$	$(1+x)/(1+g \cdot x^2)$	-1	1	[0,d,1,2,-1,d]

Програмне забезпечення на MATLAB:

function U = firlin(X, A, F, D)

```
% Розв'язування методом "стрільби"
% системи 2-х лінійних ЗДР 1-го порядку
%  $U'(x) = A(x) \cdot U(x) + F(x)$ ,  $x \in (a, b)$ ,
% для якої в точках  $a = X(1)$ ,  $b = X(n)$ , ( $n = \text{length}(X)$ )
% виконуються крайові умови
%  $(D(1) \cdot U(1, x) + D(2) \cdot U(2, x))|_{x=a} = D(3)$ ,
%  $(D(4) \cdot U(1, x) + D(5) \cdot U(2, x))|_{x=b} = D(6)$ .
% Для  $D(1), D(2)$  і  $D(4), D(5)$  мають виконуватись умови :
%  $(D(1) \sim 0 \mid D(2) \sim 0) \ \& \ (D(4) \sim 0 \mid D(5) \sim 0)$ .
%
% Вхідні аргументи :
% X - сітка виведення результатів;
% A - функція з описом матриці  $A(x)$ ;
% F - функція з описом  $F(x)$ .
% D - масив значень коефіцієнтів КУ.
% Вихідні аргументи :
% U - (2,n)-масив наближеного розв'язку  $U(X) = [u_1(X); u_2(X)]$ .
%
```

```
function yp = fir_fo(t, Y)
```

```
% опис правої частини для однорідної системи
yp = A(t) * Y;
end
```

```
function yp = fir_fno(t, Y)
```

```
% опис правої частини для неоднорідної системи
yp = A(t) * Y + F(t);
end
```

```
if nargin < 2
    error('FIRLIN:NotEnoughInputs',...
        'Not enough input arguments. See FIRLIN. ');
    return
end
if length(D) < 6
    error('FIRLIN:NotEnoughInputs',...
        'Not enough input arguments D. See FIRLIN. ');
    return
end
if D(1) == 0 & D(2) == 0
    error('FIRLIN:error: D(1),D(2) =0.',...
        'D(1)==0 & D(2)==0. See FIRLIN. ');
    return
end
if D(4) == 0 & D(5) == 0
    error('FIRLIN:error: D(4),D(5) =0.',...
        'D(4)==0 & D(5)==0. See FIRLIN. ');
end
```

```

        'D(4)==0 & D(5)==0. See FIRLIN. ');
    return
end

n = length(X); U = zeros(2,n-1);

% Розв'язок однорідної системи ЗДР
if D(2)
    y1 = [1; -D(1)./D(2)];
else
    y1 = [-D(2)./D(1); 1];
end
U0 = [y1, U];
for i = 2 : n
    [x, y] = ode45(@fir_fo, [X(i-1) X(i)], y1);
    [m, c] = size(y); y1 = y(m,:); U0(:,i) = y1;
end

% Розв'язок неоднорідної системи ЗДР
if D(2) ~= 0
    y1 = [0; D(3)./D(2)];
else
    y1 = [D(3)./D(1); 0];
end
U = [y1, U];
for i = 2 : n
    [x, y] = ode45(@fir_fno, [X(i-1) X(i)], y1);
    [m, c] = size(y); y1 = y(m,:); U(:,i) = y1;
end

% обчислення параметра c
c = D(4).*U0(1,n) + D(5).*U0(2,n);
c = (D(6) - D(4).*U(1,n) - D(5).*U(2,n))./c;

% знаходження розв'язку в точках виведення
U = U + c.*U0;
end

```

Практичне заняття № 05

ТЕМА № 4. Розв'язування крайових задач для ЗДР.

4.2. Проекційні методи: колокацій, Бубнова-Гальоркіна.

Варіаційні методи: Рітца, найменших квадратів (2 год).

Теоретична інформація:

Розглянемо *першу крайову задачу* для $Lu(x)$ – лінійного диференціального оператора другого порядку:

$$\begin{cases} Lu(x) \equiv u''(x) - p(x)u(x) = -f(x), \\ p(x) \geq 0, \quad x \in (a, b); \\ u(a) = d_1, \quad u(b) = d_2. \end{cases} \quad (4.11)$$

До групи чисельно-аналітичних методів, де наближений розв'язок (4.11) точно задовольняє крайові умови, а диференціальне рівняння наближено, відносять *проекційні і варіаційні методи*.

п.1 У проекційному методі (методі моментів) розв'язок $u(x)$ шукаємо у вигляді

$$u(x) \approx y_n(x) = \varphi_0(x) + \sum_{i=1}^n c_i \varphi_i(x), \quad (4.12)$$

де $\varphi_0(x)$ задовольняє неоднорідні крайові умови, наприклад (у варіанті КУ I роду), $\varphi_0(x) = d_1 + \frac{d_2 - d_1}{b - a}(x - a)$, а *координатна система* функцій $\{\varphi_i(x)\}_{i=1}^n$ – лінійно-незалежна і задовольняє однорідні крайові умови, наприклад, $\varphi_i(x) = (x - a)^i(b - x)$. Крім цього, для координатної системи повинні виконуватись додаткові умови:

- 1) $\varphi_i(x) \in C^2([a, b])$;
- 2) $\{\varphi_i(x)\}_{i=1}^n$ – *повна система* ($\forall \varepsilon > 0 \quad \forall x \in [a, b] \quad \exists n \ \& \ \exists \{c_i\}_{i=1}^n : \|u(x) - y_n(x)\| \leq \varepsilon$).

В цьому методі вводиться ще одна система лінійно-незалежних функцій $\{\psi_i(x)\}_{i=1}^n$ – *проекційна*, для якої повинні виконуватись наступні умови:

- 1) $\psi_i(x) \in C([a, b])$;
- 2) $\{\psi_i(x)\}_{i=1}^n$ – *замкнена система* ($\forall f(x) \in C([a, b]) \ \& \ (f, \psi_i) = 0 \Rightarrow f(x) \equiv 0$).

Враховуючи замкненість проекційної системи, отримаємо для *моментів* μ_i умови:

$$\begin{cases} \mu_i \equiv (Ly_n + f, \psi_i) = \int_a^b (Ly_n(x) + f(x))\psi_i(x)dx = 0, \quad i = \overline{1, n} \end{cases}$$

звідки для коефіцієнтів $\vec{c} = (c_i)_{i=1}^n$ наближення (4.12) маємо наступну СЛАР $A\vec{c} = \vec{f}$.

Елементи матриці $A = (a_{i,j})_{i,j=1}^n$ і вектора $\vec{f} = (f_i)_{i=1}^n$ обчислюються по формулам

$$a_{i,j} = \int_a^b \varphi_j'(x)\psi_i'(x) + p(x)\varphi_j(x)\psi_i(x)dx, \quad f_i = \int_a^b (f(x) + L\varphi_0(x))\psi_i(x)dx. \quad (4.13)$$

В залежності від виду проекційної системи, можливі наступні варіанти обчислення цих коефіцієнтів.

Метод колокації: визначаємо проекційні функції у вигляді функцій Дірака $\psi_i(x) = \delta(x - x_i)$, де x_i це точки колокації ($a < x_1 < \dots < x_n < b$), тоді формули (4.13) виглядатимуть таким чином

$$a_{i,j} = \varphi_j''(x_i) - p(x_i)\varphi_j(x_i), \quad f_i = -f(x_i) - \varphi_0''(x_i) + p(x_i)\varphi_0(x_i). \quad (4.14)$$

Метод Бубнова-Гальоркіна: обидві системи функції співпадають: $\psi_i(x) \equiv \varphi_i(x)$, і формули (4.13) запишуться у вигляді:

$$a_{i,j} = \int_a^b (\varphi_j'(x)\varphi_i'(x) + p(x)\varphi_j(x)\varphi_i(x))dx, \quad (a_{i,j} = a_{j,i}); \quad f_i = \int_a^b (f(x) + L\varphi_0(x))\varphi_i(x)dx. \quad (4.15)$$

Для систем (4.14), (4.15) доведено існування та єдиність розв'язку. Аналіз збіжності проекційних методів достатньо складний і залежить від виду крайової задачі, координатної і проекційної системи, метрики. Відомо, що метод Гальоркіна збіжний, коли оператор L – симетричний та додатно визначений. Для випадку задачі (4.11) виконується оцінка $\|y_n(x) - u(x)\| = O(n^{-3/2})$. Швидкість збіжності також залежить від вибору системи $\{\varphi_i(x)\}_{i=1}^n$ (краще обирати ортонормовану систему).

Для практичного вибору параметра n оберем $\varepsilon > 0$ – задану точність і побудову (4.12) виконуємо до виконання умови $\|y_n(x) - y_{n-1}(x)\| \leq \varepsilon$.

п.2 Варіаційні методи розв'язування крайових задач для ЗДР.

Розглянемо лінійне диференціальне рівняння другого порядку (4.11) як операторне $Lu(x) = -f(x)$, де $L: H \rightarrow F$ – лінійний оператор, H – гільбертів простір, $u \in D(L) \subset H$ – область визначення L , $f \in F = R(L) \subset H$ – область значень L . Про (4.11) відомо, що воно має єдиний розв'язок $u^{exact}(x)$. Для цього рівняння побудуємо певний функціонал $J(u)$, такий, що $D(J) = D(L)$, $J(u) \geq c > -\infty$ і $\min_{D(J)} J(u) = J_0 = J(\tilde{u}(x))$ досягається тільки на одній функції $\tilde{u}(x)$. Ця функція і буде розв'язком (4.11), тобто $u^{exact}(x) = \tilde{u}(x)$. Звичайно, знайти $\tilde{u}(x)$ в явному вигляді для $J(u)$ буває складно, тому будується збіжна послідовність функцій $y_n(x)$, така, що $\lim_{n \rightarrow \infty} J(y_n(x)) = J(\tilde{u}(x))$. Для побудови функцій $y_n(x)$ використовують *процес Рітца*:

- 1) Обираємо деяку систему лінійно-незалежних функцій $\{\varphi_i(x)\}_{i=1}^\infty$, повну в H і кожна функція $\varphi_i(x)$ задовольняє однорідні крайові умови в (4.11).
- 2) Будуємо функції $y_n(x)$ аналогічно формулі (4.12) $y_n(x) = \varphi_0(x) + \sum_{i=1}^n c_i \varphi_i(x)$, де функція $\varphi_0(x)$ задовольняє неоднорідні крайові умови в (4.11), а $y_n(x) \in H_n \subset H$, H_n – скінченно-вимірний підпростір простору H , і при $n \rightarrow \infty$ маємо послідовність $H_1 \subset H_2 \subset \dots \subset H_n \subset \dots \subset H$. Звуження функціоналу J на підпростір H_n є функцією $\phi(c_1, c_2, \dots, c_n)$, що містить n параметрів $\{c_i\}_{i=1}^n$, і функція $\phi(\bar{c}) = J(y_n(x))$ досягає екстремальних значень на розв'язку системи

$$\left\{ \frac{\partial \phi(c_1, c_2, \dots, c_n)}{\partial c_i} = 0, \quad i = \overline{1, n}. \right. \quad (4.16)$$

З повноти системи $\{\varphi_i(x)\}_{i=1}^\infty$ випливає: $y_n(x) \xrightarrow{n \rightarrow \infty} \tilde{u}(x) \quad \left(\|y_n(x) - \tilde{u}(x)\| \rightarrow 0 \right)_{n \rightarrow \infty}$.

Різні варіаційні методи (ВМ) відрізняються вибором функціоналу J , який залежить від властивостей оператора L . Розглянемо конкретні варіанти ВМ.

Метод Рітца: необхідні такі властивості оператора L

1. $L = L^*$ – самоспряжений, $(Lu, u) = (u, Lu)$;
2. $L > 0$ – додатно визначений, $(Lu, u) \geq \mu_L \|u\|^2$, $\mu_L > 0$, $\forall u \in D(L)$.

Тоді *функціонал енергії*

$$J(u) = (Lu, u) + 2(f, u) = \int_a^b [(u')^2 + p * u^2 - 2f * u] dx = \int_a^b \Psi(x, u, u') dx \quad (4.17)$$

буде обмеженим знизу, а рівняння Ейлера $\frac{d}{dx} \Psi_{u'} - \Psi_u = 0$ для нього співпадає з рівнянням (4.11). Підставимо $y_n(x)$ в функціонал $J(u)$, отримаємо

$$J(u) = C + 2 \sum_{j=1}^n B_j c_j + \sum_{i=1}^n \sum_{j=1}^n A_{i,j} c_i c_j,$$

де

$$C = \int_a^b [(\varphi'_0)^2 + p \varphi_0^2 - 2f \varphi_0] dx, \\ B_j = \int_a^b [\varphi'_0 \varphi'_j + (p \varphi_0 - f) \varphi_j] dx, \quad A_{i,j} = A_{j,i} = \int_a^b [\varphi'_i \varphi'_j + p \varphi_i \varphi_j] dx, \quad i, j = \overline{1, n}. \quad (4.18)$$

Отже при застосуванні процесу Рітца, система (4.16) буде, відносно коефіцієнтів $\bar{c}$, СЛАР наступного виду:

$$\begin{cases} \sum_{j=1}^n A_{i,j} c_j = -B_i, \\ i = \overline{1, n}. \end{cases} \quad (4.19)$$

Метод найменших квадратів (МНК): від оператора L вимагається тільки лінійність, а при нашій домовленості про існування єдиного розв'язку (4.11) – існування оберненого оператора L^{-1} . Тоді функціонал запишемо у вигляді

$$J(u) = \|Lu + f\|^2 = (Lu, Lu) + 2(Lu, f) + (f, f).$$

Застосування процесу Рітца приводить до розв'язання СЛАР (4.19), де $A_{i,j}$, B_i обчислюються по формулам (4.20):

$$B_i = - \int_a^b [f L(\varphi_i)] dx, \quad A_{i,j} = A_{j,i} = \int_a^b [L(\varphi_j) L(\varphi_i)] dx, \quad i, j = \overline{1, n}. \quad (4.20)$$

Зауваження

1. Метод Рітца для рівняння $L^* Lu(x) = L^* f(x)$ еквівалентний МНК.
2. Метод Рітца і МНК можна розглядати як окремі схеми методу моментів. Так, якщо в методі моментів взяти:

$\psi_i(x) = \varphi_i(x)$, то отримаємо метод Рітца;

$\psi_i(x) = L \varphi_i(x)$ – отримаємо МНК.

3. Якщо маємо КУ II, III роду, то оберемо $\varphi_0(x) = (x-a) * (\tilde{A} + \tilde{B} * (b-x))$ і коефіцієнти $\tilde{A}$, $\tilde{B}$ визначаємо з КУ. Щодо системи координатних функцій, то для виду завдання $\varphi_i(x) = (x-a)^i (b-x)$ необхідно збільшити на 1 показник степеня при $(x-a)$ або при $(b-x)$, в залежності від того, в якій точці границі маємо КУ II, III роду.

Аудиторне завдання:

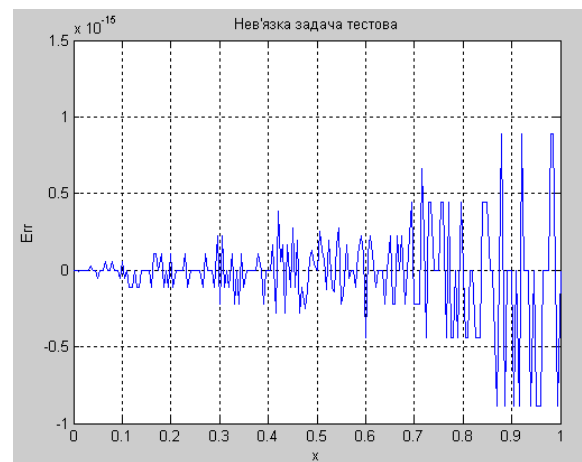
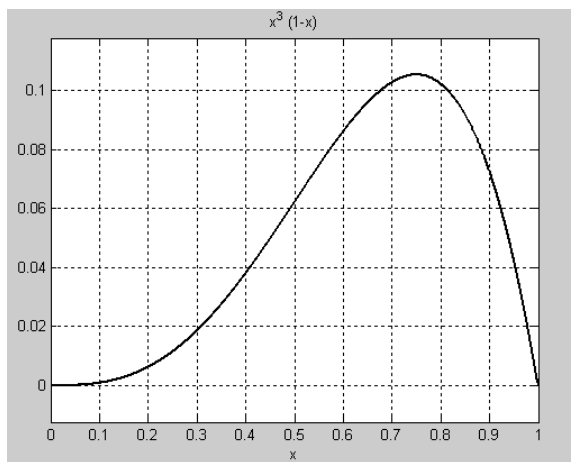
п.1. За допомогою методу колокації (функція **m_colocs**) з точністю $eps = 1e-5$ знайти $y_n(x)$ – чисельно-аналітичний розв'язок крайової задачі (4.11) при наступних даних $p(x) = 0$, $f(x) = 6x(2x-1)$, $a = 0$, $b = 1$, $d_1 = 0$, $d_2 = 0$. За допомогою М-функції користувача **exact_sol** легко знайти $u(x)$ – точний розв'язок задачі (4.11). Побудувати графік $y_n(x)$ і графік нев'язки $\delta_n(x) = L(y_n(x)) + f(x)$. Обчислити значення $\|\delta_n(x)\|_C$ і похибки $\|y_n(x) - u(x)\|_C$.

Створимо відповідну М-функцію *p05a01* :

```
function p05a01
clear all, close all, clc
% тестова задача :
a = 0; b = 1; D = [0, 0];
p = @(t) (0);
f = @(t) (6*t*(2*t-1));
disp('Точний розв'язок :');
u = exact_sol(p, f, a, b, D);
ep = 1e-5; nmax = 12;
Y = m_colocs(p, f, a, b, D, ep, nmax, 1);
% візуалізація наближеного розв'язку
disp(['Y=',char(Y)]);
G1 = ezplot(Y,[a, b]);
set(G1,'color',[0 0 0],'LineWidth', 1.5);
grid on; pause(); close all;
% обчислення і візуалізація нев'язки
N = 201;
defect(a, b, N, Y, p, f, D, 'задача тестова');
X = linspace(a, b, N);
E = simplify(Y-u); E = subs(E,'x',X(:));
fm = '%s%14.6e\n';
fprintf(fm,'Похибка ||Yn-u(x)||c=',norm(E,inf));
end
```

і отримаємо такий результат її виконання (показано тільки кінцеві результати):

```
Точний розв'язок :
u(x)=-x^4+x^3
Y=x^3*(1-x)
Нев'язка задача тестова= 8.881784e-016
Похибка ||Yn-u(x)||c= 0.000000e+000
```



п.2. За допомогою методу колокації (функція *m_colocs*) з точністю $eps = 1e-5$ отримати чисельно-аналітичний розв'язок крайової задачі (4.11) при наступних даних $p(x)=1+x$, $f(x)=2x(x-1)$, $a=0$, $b=2$, $d_1=1$, $d_2=2$. Побудувати графіки наближеного розв'язку і нев'язки $\delta_n(x)$. Обчислити $\|\delta_n(x)\|_C$.

Створимо відповідну М-функцію *p05a02* :

```
function p05a02
clear all, close all, clc
% Опис задачі
a = 0; b = 2; D = [1, 2];
p = @(t) (1+t);
f = @(t) (2*t*(t-1));
ep = 1e-5; nmax = 17;
```

```

Y = m_colocs(p, f, a, b, D, ep, nmax, 1);
% візуалізація наближеного розв'язку
disp(['Y=',char(vpa(Y,6))]);
G1 = ezplot(Y,[a, b]); %hold on;
set(G1,'color',[0 0 0],'LineWidth', 1.5);
grid on; pause(); close all;
% обчислення і візуалізація похибки
defect(a, b, 201, Y, p, f, D, 'м.колокації/sym');
end

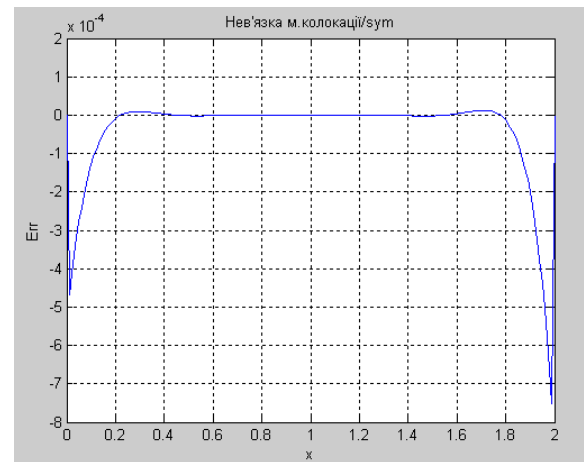
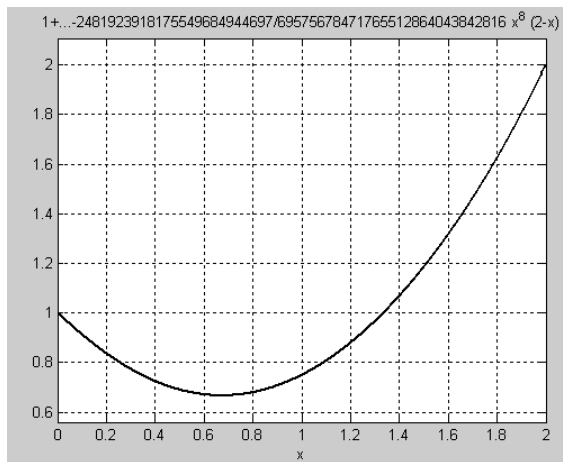
```

і отримаємо наступний результат її виконання (показано тільки кінцеві результати):

```

Y=1+.500000*x-.717491*x*(2.-1.*x)-.108878*x^2*(2.-1.*x)+.118173*x^3*(2.-1.*x)
-.436161e-1*x^4*(2.-1.*x)+.111461e-2*x^5*(2.-1.*x)+.123327e-2*x^6*(2.-1.*x)
-.353488e-3*x^7*(2.-1.*x)-.356723e-4*x^8*(2.-1.*x)
Нев'язка м.колокації/sym= 7.514623e-004

```



п.3. Знайти чисельно-аналітичний розв'язок крайової задачі (4.11) з точністю $\text{eps}=1e-5$ при наступних даних $p(x)=1+x$, $f(x)=2x(x-1)$, $a=0$, $b=2$, $d_1=1$, $d_2=2$ за допомогою методів :

1. колокації (функція **m_coloc**);
2. Бубнова-Гальоркіна (функція **m_galer**);
3. Рітца (функція **m_ritza**).

Побудувати графіки цих наближень на одному рисунку (для візуального порівняння). Для кожного з методів побудувати графік нев'язки $\delta_n(x)$ і обчислити $\|\delta_n(x)\|_C$.

Зауваження. При реалізації функцій **m_coloc**, **m_galer**, **m_ritza** використано гібридний варіант обчислення інтегралів, а саме: інтеграли, які містять тільки $\varphi_0(x)$, $\varphi_j(x)$ та їх похідні – обчислено точно, а інтеграли, в які входять $p(x)$, $f(x)$ – обчислено чисельно. Програмна реалізація – функція **i_simpsym**, в якій використано адаптивний алгоритм КФ м.Сімпсона.

Створимо відповідну М-функцію **p05a03** :

```

function p05a03
clear all, close all, clc
a = 0; b = 2; D = [1, 2];
% опис функцій p, f (для числових обчислень)
p = @(t) (1+t);
fd = @(t) (2.*t.*(t-1));
% опис функцій p, f (для символьних обчислень)
fs = @(t) (2*t*(t-1));
% точність наближення Yn і тах значення n :
ep = 1e-5; nmax = 17;
% точність числового інтегрування і тах число

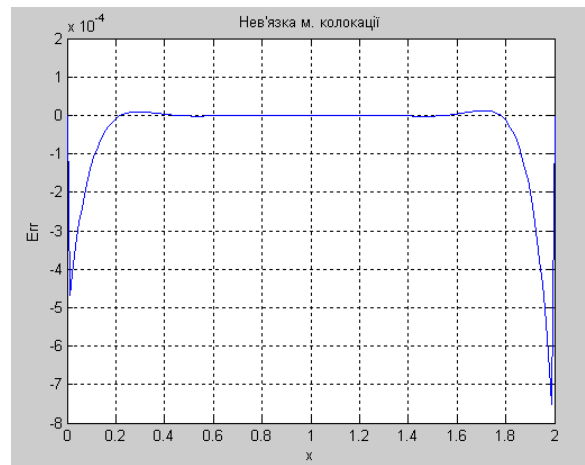
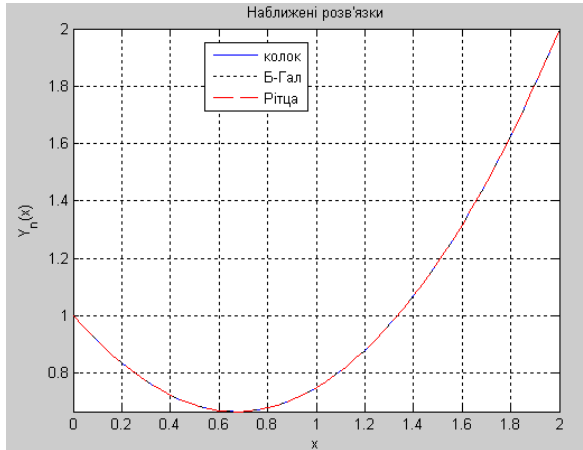
```

```

% поділу [a,b] в адапт.алгоритмі інтегрування :
epsi = 1e-10; maxd = 100;
YC = m_coloc(p, fd, a, b, D, ep, nmax);
disp(['Yколок=',char(vpa(YC,6))]);
YG = m_galer(p, fs, a, b, D, ep, nmax, epsi, maxd);
disp(['YБ-Гал=',char(vpa(YG,6))]);
YR = m_ritza(p, fs, a, b, D, ep, nmax, epsi, maxd);
disp(['YРітца=',char(vpa(YR,6))]);
nx = 51; X = linspace(a, b, nx);
m = inf; M = -inf;
for i = 1 : nx
    t = X(i); z = subs(YC,'x',t); m = min(m,z); M = max(M,z);
    yc(i) = z; yg(i) = subs(YG,'x',t); yr(i) = subs(YR,'x',t);
end
plot(X,yc,'b -',X,yg,'k :',X,yr,'r --'); grid on;
axis([a, b, m, M]); title('Наближені розв'язки');
legend('колок','Б-Гал','Рітца','Location','Best');
xlabel('x'); ylabel('Y_n(x)');
pause(); close all;
% обчислення і візуалізація похибки м.колокації
defect(a, b, 201, YC, p, fd, D, 'м. колокації');
pause(); close all;
% обчислення і візуалізація похибки м.Б-Гальоркіна
defect(a, b, 201, YG, p, fs, D, 'м.Гальоркіна');
pause(); close all;
% обчислення і візуалізація похибки м.Рітца
defect(a, b, 201, YR, p, fs, D, 'м. Рітца ');
end

```

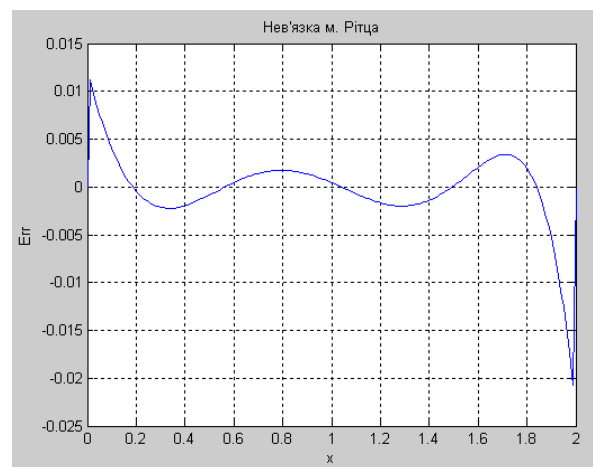
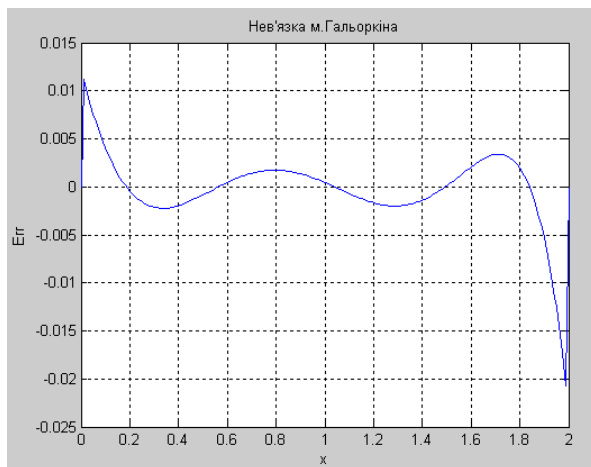
і отримаємо наступний результат її виконання:



```

Yколок=1.+500000*x-.717491*x*(2.-1.*x)-.108878*x^2*(2.-1.*x)+.118173*x^3*(2.-1.*x)
-.436161e-1*x^4*(2.-1.*x)+.111461e-2*x^5*(2.-1.*x)+.123327e-2*x^6*(2.-1.*x)
-.353488e-3*x^7*(2.-1.*x)-.356723e-4*x^8*(2.-1.*x)
YБ-Гал=1.+500000*x-.717830*x*(2.-1.*x)-.105869*x^2*(2.-1.*x)+.110534*x^3*(2.-1.*x)
-.367127e-1*x^4*(2.-1.*x)+.157124e-4*x^5*(2.-1.*x)
YРітца=1.+500000*x-.717830*x*(2.-1.*x)-.105869*x^2*(2.-1.*x)+.110534*x^3*(2.-1.*x)
-.367127e-1*x^4*(2.-1.*x)+.157118e-4*x^5*(2.-1.*x)
Нев'язка м. колокації= 7.514623e-004
Нев'язка м.Гальоркіна= 2.069101e-002
Нев'язка м. Рітца = 2.069101e-002

```



Завдання для самостійної роботи:

п.1. Отримати з точністю $\text{eps}=1e-4$ чисельно-аналітичний розв'язок крайової задачі (4.11) за допомогою методу: колокації, Бубнова-Гальоркіна, Рітца. Побудувати графіки наближених розв'язків на одному рисунку (для візуального порівняння). Для кожного з методів побудувати графік нев'язки $\delta_n(x)$ і обчислити значення $\|\delta_n(x)\|_C$, $\|y_n(x)-u(x)\|_C$, де $u(x)$ – точний розв'язок (4.11). Для його знаходження написати функцію **$u = \text{exact_sol}(p, f, a, b, D)$** , яка за допомогою солвера **dsolve** знаходить $u(x)$, а також будує графік $u(x)$, використовуючи **ezplot** .

Таблиця 4.2 Варіанти індивідуального завдання.

K	$p(x)$	$f(x)$	a	b	D
1	4	$2x(1-x)$	0	2	$[-2, -2]$
2	π	$-\sin(x)$	0	π	$[1, 0]$
3	1	$-1+x+x^2$	0	$\pi/2$	$[0, 0]$
4	1	$-x^2$	0	$\pi/2$	$[1, 0]$
5	1	$-\sin(x)$	0	$\pi/2$	$[1, 0]$
6	1	$-\cos(x)$	0	$\pi/2$	$[1, 0]$
7	1	$-x+\cos(x)$	0	$\pi/2$	$[1, 0]$
8	2	$-x^3$	0	1	$[1, 0]$
9	4	$-x-x^3$	0	1	$[1, 0]$
10	4	$-1+x+x^3$	0	1	$[1, 0]$
11	3	$x-3$	0	1	$[0, 1]$
12	1	x^2-x-2	0	1	$[1, 2]$
13	1/2	$\sin(x-2)$	0	1	$[0, 2]$
14	2	$-4x-1$	0	1	$[1, 0]$
15	4	$-4(x-1)$	0	1	$[0, 0]$
16	3	$2x-4$	0	2	$[-1, -2]$
17	2	$\cos(x)$	0	$\pi/2$	$[0, 0]$
18	4	$1-\sin(x)$	1/2	π	$[0, 1]$
19	2	$1-x^2$	0	1	$[0, 1]$
20	3	$2+x$	0	2	$[0, 0]$

Програмне забезпечення на MATLAB:

$\text{function fi} = \text{build_fi}(k, a, b)$

```

% побудова елементу  $\phi_i(k) = (x-a)^k(b-x)$ 
x = sym('x');
fi = (x-a)^k*(b-x);
end

function defect(a, b, N, Y, pfun, ffun, D, txt)
R = ['Нев' 'язка ',txt];
X = linspace(a,b,N); I = 2 : N-1;
% підставимо наближений розв'язок в крайову задачу
x = sym('x'); E = diff(Y,x,2) - pfun(x)*Y + ffun(x);
% враховуємо крайові умови
Err = [subs(Y,x,a)-D(1), X(I), subs(Y,x,b)-D(2)];
for k = I(:)
    Err(k) = subs(E,x,X(k));
end
plot(X, Err); xlabel('x'); ylabel('Err');
grid on; title(R);
fprintf('%s=%14.6e\n',R,norm(Err,inf));
end

function s = i_simpsym(a, b, f, ep, mki)
% s - значення визначеного інтеграла на [a,b]
% від символічної функції f(x) з точністю ep
% за допомогою адаптивного алгоритма м.Сімпсона
% з максимальною кількістю згущення сітки mki.
x = sym('x');
A = a; B = b; z = 1;
if a > b
    A = b; B = a; z = -1;
end
k = 0; h = B - A; h = 0.5.*h;
s1 = subs(f,x,A) + subs(f,x,B);
s2 = 0; s4 = subs(f,x,0.5.*(B+A));
s = h.*(s1 + 4.*s4)./3; F = 0;
while ~F & k < mki
    k = k+1; ho = h; h = 0.5.*h;
    s2 = s2 + s4;
    s4 = 0; t = A + h;
    while t < B
        s4 = s4 + subs(f,x,t); t = t + ho;
    end
    ss = h.*(s1 + 4.*s4 + 2.*s2)./3;
    F = abs(ss - s)./15 < ep; s = ss;
end
s = z.*s;
return

function Y = m_colocs(pfun, ffun, a, b, D, ep, N, vw)
% Розв'язок крайової задачі
%  $u''(x) - p(x)u(x) = -f(x)$ ,
%  $p(x) > 0$ ,  $x \in (a,b)$ ,
%  $u(a)=D(1)$ ,  $u(b)=D(2)$ .
% наближеним проекційним методом - м.колокації
% (символьна реалізація)
% Вхідні дані :
% pfun - функція з описом p(x);
% ffun - функція з описом f(x);
% a, b - граничні точки області;
% D - вектор крайових умов;

```

```

% ер - точність (0<ер<1);
% N - max значення n - кількості елементів наближення Yn;
% vw =1 - візуалізація наближення на ітераціях.
% Вихідні дані :
% Y - Yn-те наближення до розв'язку.
syms x
n = (D(2)-D(1))./(b-a); F = D(1)-n.*a;
fi0 = F + n*x; % fi(0)
n = 1; F = 0; Y = fi0;
while ~F & n <= N
    YY = Y; n = n + 1;
    % обчислюємо рівномірні точки колокації
    X = linspace(a,b,n+2); X = X(2:end-1);
    % формуємо наближення u~Y(n):
    % Y(n)=fi(0)+Сума(k=1,n,C(k)*fi(k))
    Y = fi0;
    for k = 1 : n
        C(k) = sym(['C',int2str(k)]);
        Y = Y + C(k)*build_fi(k, a, b);
    end
    % формуємо нев'язку
    E = diff(Y,x,2) - pfun(x)*Y + ffun(x);
    % формуємо СЛАР в точках колокації
    for k = 1 : n
        Ер{k} = subs(E,x,X(k));
    end
    clear E X;
    % розв'язуємо СЛАР
    CC = solve(Eр{1:end});
    % підставляємо знайдені константи C(k)
    % в наближений розв'язок
    CC = struct2cell(CC);
    for k = 1 : n
        Y = subs(Y,C(k),CC{k});
    end
    % порівнюємо сусідні наближення ||Y-YY||<ер
    F = abs(eval(int(Y-YY,x,a,b))) < ер;
    if vw
        % візуалізація сусідніх наближень
        G1 = ezplot(YY,[a,b]);
        set(G1,'color',[0 0 0],'LineWidth',1); hold on;
        G2 = ezplot(Y,[a,b]); grid on;
        set(G2,'color',[0 0 1],'LineWidth', 1.5);
        title(['n=',int2str(n)],'FontName','Courier','FontSize',10);
        hold off; pause(1); close all;
    end
    clear CC C
end
clear YY
return

function Y = m_coloc(pfun, ffun, a, b, D, ер, N)
% Розв'язок крайової задачі
% u''(x) - p(x)*u(x) = -f(x),
% p(x)>0, x є (a,b),
% u(a)=D(1), u(b)=D(2).
% наближеним проекційним методом - м.колокації
% (гібридний метод обчислення інтегралів)
% Вхідні дані :

```

```

% pfun - функція з описом  $p(x)$ ;
% ffun - функція з описом  $f(x)$ ;
% a, b - граничні точки області;
% D - вектор крайових умов;
% ep - точність ( $0 < ep < 1$ );
% N - max значення n - кількості елементів наближення  $Y_n$ .
% Вихідні дані :
% Y -  $Y_n$ -те наближення до розв'язку.
syms x;
n = (D(2)-D(1))./(b-a); F = D(1)-n.*a;
fi0 = F + n*x; %  $\phi_i(0)$ 
n = 1; F = 0; Y = fi0;
while ~F & n <= N
    YY = Y; n = n + 1;
    % обчислюємо рівномірні точки колокації
    X = linspace(a,b,n+2); X = X(2:end-1);
    % формуємо СЛАР: A,B
    A = zeros(n, n); B = zeros(n, 1);
    for i = 1 : n
        xi = X(i); px = pfun(xi);
        for j = 1 : n
            fj = build_fi(j, a, b); %  $\phi_i(j)$ 
            aij = diff(fj,x,2) - px*fj; %  $L(\phi_i(j))$ 
            A(i, j) = subs(aij, x, xi);
            clear fj aij
        end
        B(i) = px*subs(fi0, x, xi) - ffun(xi);
    end
    c = A \ B; % розв'язуємо СЛАР
    clear A B
    % формуємо наближення  $u \sim Y(n)$ :
    Y = fi0;
    for i = 1 : n
        Y = Y + c(i)*build_fi(i, a, b);
    end
    clear c
    % порівнюємо сусідні наближення  $||Y-YY|| < ep$ 
    F = abs(eval(int(Y-YY, x, a, b))) < ep;
end
clear YY fi0
return
end

```

```

function Y = m_galer(pfun, ffun, a, b, D, ep, N, epsi, maxd)
% Розв'язок крайової задачі
%  $u''(x) - p(x)*u(x) = -f(x)$ ,
%  $p(x) > 0, x \in (a,b)$ ,
%  $u(a)=D(1), u(b)=D(2)$ .
% наближеним проекційним методом - м.Гальоркіна
% (гібридний метод обчислення інтегралів)
% Вхідні дані :
% pfun - функція з описом  $p(x)$ ;
% ffun - функція з описом  $f(x)$ ;
% a, b - граничні точки області;
% D - вектор крайових умов;
% ep - точність ( $0 < ep < 1$ );
% N - max значення n - кількості елементів наближення  $Y_n$ ;
% epsi - точність для числового інтегрування;
% maxd - max кількість поділу  $[a,b]$  в адапт.алгоритмі.

```

```

% Вихідні дані :
% Y      - Yn-те наближення до розв'язку.
syms x;
n = (D(2)-D(1))./(b-a); F = D(1)-n.*a;
fi0 = F + n*x; %  $\phi_i(0)$ 
px = pfun(x); f_p = ffun(x)-px*fi0;
n = 1; F = 0; Y = fi0;
while ~F & n <= N
    YY = Y; n = n + 1;
    % формуємо СЛАР: A,B
    A = zeros(n, n); B = zeros(n, 1);
    for i = 1 : n
        fi = build_fi(i, a, b); %  $\phi_i(i)$ 
        dfi = diff(fi,x); %  $\phi_i'(i)$ 
        for j = i : n
            fj = build_fi(j, a, b); %  $\phi_i(j)$ 
            dfj = diff(fj,x); %  $\phi_i'(j)$ 
            %1-а частина A(i,j)  $\text{int}(a,b,\phi_i'(j)*\phi_i'(i))$  - точно :
            aij = int(dfi*dfj, x, a, b);
            %2-а частина A(i,j)  $\text{int}(a,b,p*\phi_i(j)*\phi_i(i))$  - наближено :
            z = px*fi*fj;
            aij = aij + i_simpsym(a, b, z, epsi, maxd);
            A(i, j) = aij;
            if i ~= j
                A(j, i) = aij;
            end
            clear fj dfj z
        end
        z = f_p*fi;
        B(i) = i_simpsym(a, b, z, epsi, maxd);
        clear fi dfi z
    end
    c = A \ B; % розв'язуємо СЛАР
    clear A B aij
    % формуємо наближення  $u \sim Y(n)$ :
    Y = fi0;
    for i = 1 : n
        Y = Y + c(i)*build_fi(i, a, b);
    end
    clear c
    % порівнюємо сусідні наближення  $||Y-YY|| < \epsilon$ 
    F = abs(eval(int(Y-YY, x, a, b))) < ep;
end
clear YY fio x px f_p
return

```

function Y = m_ritza(pfun, ffun, a, b, D, ep, N, epsi, maxd)

```

% Розв'язок крайової задачі
%  $u''(x) - p(x)*u(x) = -f(x)$ ,
%  $p(x) > 0, x \in (a,b)$ ,
%  $u(a)=D(1), u(b)=D(2)$ .
% наближеним варіаційним методом - м.Рітца
% (гібридний метод обчислення інтегралів)
% Вхідні дані :
% pfun - функція з описом  $p(x)$ ;
% ffun - функція з описом  $f(x)$ ;
% a, b - граничні точки області;
% D - вектор крайових умов;

```

```

% ер - точність (0<ер<1);
% N - max значення n - кількості елементів наближення Yn;
% ерси - точність для числового інтегрування;
% maxd - max кількості поділу [a,b] в адапт.алгоритмі.
% Вихідні дані :
% Y - Yn-те наближення до розв'язку.
syms x;
n = (D(2)-D(1))./(b-a); F = D(1)-n.*a;
fi0 = F + n*x; %  $\phi_i(0)$ 
dfi0 = diff(fi0,x); %  $\phi_i'(0)$ 
px = pfun(x); f_p = ffun(x) - px*fi0;
n = 1; F = 0; Y = fi0;
while ~F & n <= N
    YY = Y; n = n + 1;
    % формуємо СЛАР: A,B
    A = zeros(n, n); B = zeros(n, 1);
    for i = 1 : n
        fi = build_fi(i, a, b); %  $\phi_i(i)$ 
        dfi = diff(fi,x); %  $\phi_i'(i)$ 
        for j = i : n
            fj = build_fi(j, a, b); %  $\phi_i(j)$ 
            dfj = diff(fj,x); %  $\phi_i'(j)$ 
            %1-а частина A(i,j)  $\text{int}(a,b,\phi_i'(j)*\phi_i'(i))$  - точно :
            aij = int(dfi*dfj, x, a, b);
            %2-а частина A(i,j)  $\text{int}(a,b,p*\phi_i(j)*\phi_i(i))$  - наближено :
            z = px*fi*fj;
            aij = aij + i_simpsym(a, b, z, ерси, maxd);
            A(i, j) = aij;
            if i ~= j
                A(j, i) = aij;
            end
            clear fj dfj z
        end
        z = f_p*fi - dfi0*dfi;
        B(i) = i_simpsym(a, b, z, ерси, maxd);
        clear fi dfi z
    end
    c = A \ B; % розв'язуємо СЛАР
    clear A B aij
    % формуємо наближення u~Y(n):
    Y = fi0;
    for i = 1 : n
        Y = Y + c(i)*build_fi(i, a, b);
    end
    clear c
    % порівнюємо сусідні наближення ||Y-YY||<ер
    F = abs(eval(int(Y-YY, x, a, b))) < ер;
end
clear YY fio x px f_p
return

```

Практичне заняття № 06

ТЕМА № 4. Розв'язування крайових задач для ЗДР.

4.3. Різницеві схеми для лінійних і квазілінійних задач (2 год).

Теоретична інформація:

Розглянемо основні ідеї методу різницевих схем (МРС) на прикладі лінійної крайової задачі (4.11) :

$$\begin{cases} Lu(x) \equiv u''(x) - p(x)u(x) = -f(x), & x \in D \equiv (a, b); \\ p(x) \geq 0; & \bar{D} = D \cup \partial D, \quad \partial D = a \cup b; \\ u(a) = d_1, & u(b) = d_2. \end{cases}$$

Будемо вважати, що задані функції $p(x)$, $f(x)$ мають достатню гладкість.

Схема застосування МРС наступна.

А) Виконуємо дискретизацію області розв'язку задачі D та її межі ∂D – отримуємо сіткову область $\bar{D}_h = D_h \cup \partial D_h$. Для спрощення викладок введемо рівномірну сітку

$$\bar{D}_h = \{x_k = a + (k-1)h, \quad k = 1, 2, \dots, N, \quad b - a = (N-1)h\}.$$

Б) Означимо на сітці $\bar{D}_h$ дискретні функції і введемо позначення $y_k = u(x_k)$.

В) Побудуємо різницеву задачу (РЗ), використовуючи певний метод заміни диференціальних виразів на сіткові. Якщо скористаємося *методом безпосередньої заміни диференціальних операторів на формули числового диференціювання*, то на триточковому шаблоні $S\{x_{k-1}, x_k, x_{k+1}\}$ для (4.11) отримаємо РЗ, записану в формі СЛАР :

$$\begin{cases} y_1 = d_1, \\ y_{k-1} - (2 + h^2 p_k) y_k + y_{k+1} = -h^2 f_k, & k = \overline{2, N-1}; \\ y_N = d_2. \end{cases} \quad (4.21)$$

Г) Для дослідження складових частин РЗ (похибка апроксимації диференціального оператора різницеvim (4.22а), порядку апроксимації локальної і на сітці), а також в цілому РЗ (похибка і порядок апроксимації РЗ на розв'язку диференціальної (4.22б), коректність РС а) - б), збіжність (4.23) і точність РЗ) застосовують різні варіанти сіткових норм.

Використовуючи розвинення $\psi(x)$, $\psi_h(x)$ в ряд Тейлора

$$\psi(x) = L_h u - (Lu)_h = \frac{y_{k-1} - 2y_k + y_{k+1}}{h^2} - u''(x) = \frac{h^2}{12} u^{(4)}(x) + O(h^4) = O(h^2); \quad (4.22a)$$

$$\begin{aligned} \psi_h(x) &= [(y_{\bar{x},k} - p_k y_k) - (u''(x_k) - p(x_k)u(x_k))] + [f_k - f(x_k)] = \\ &= \psi_{h,1}(x) + \psi_{h,2}(x) = \frac{h^2}{12} u^{(4)}(x) = O(h^2); \end{aligned} \quad (4.22b)$$

отримаємо другий порядок апроксимації.

а) Система (4.21) – це СЛАР з тридіагональною матрицею і враховуючи $p(x) \geq 0$, маємо діагональну перевагу її елементів, а отже розв'язок існує і єдиний. Для його знаходження використовують стійкий *метод монотонної лівої/правої прогонки*.

б) Для отримання *априорних оцінок*, які використовуються при аналізі *стійкості* РС, скористаємося принципом максимуму. З урахуванням того, що маємо наступні співвідношення

$$\|y\|_C = \max_{1 \leq k \leq N} |y_k| = \max \left\{ |d_1|, |d_2|, \max_{1 < k < N} |y_k| \right\} = \max \{ |d_1|, |d_2|, |y_i| \}, \quad (1 < i < N);$$

виконаємо перетворення в (4.21) $(2 + h^2 p_i) y_i = y_{i-1} + y_{i+1} + h^2 f_i$; проведемо оцінки

$$(2 + h^2 p_i) |y_i| \leq |y_{i-1}| + |y_{i+1}| + h^2 |f_i| \leq 2|y_i| + h^2 |f_i|, \quad |y_i| \leq \left| \frac{f_i}{p_i} \right|;$$

і отримаємо $\|y_k\|_C \leq \left\| \frac{f_k}{p_k} \right\|_C + \max(|d_1|, |d_2|)$.

Скориставшись теоремою Лакса, для сіткової функції похибки $z(x) = y_k - (u)_h$, $x \in \bar{D}_h$ маємо оцінку

$$\|z_k\|_C \leq \frac{h^2}{12} \left\| \frac{u^{(4)}(x_k)}{p_k} \right\|_C,$$

а отже точність РС (4.21) є другого порядку.

В розглянутій задачі ми мали крайові умови першого роду, тому вони апроксимувались точно. У випадку крайових умов II, III роду, використовують певні методи їх апроксимації для отримання того-ж порядку, що і для диференціального рівняння. Розглянемо найбільш уживані.

Метод підвищення порядку апроксимації на розв'язку. Нехай в лівій граничній точці задано умову третього роду

$$d_1 u'(a) + d_2 u(a) = -d_3, \quad |d_1| + |d_2| \neq 0. \quad (4.23)$$

Спочатку замінімо $u'(a) \approx y_{x,1}$ і отримаємо апроксимацію цієї умови з порядком $O(h)$

$$d_1 \frac{y_2 - y_1}{h} + d_2 y_1 = -d_3.$$

Для отримання порядку $O(h^2)$ виконаємо розклад y_2 в ряд Тейлора

$$d_1 \frac{[y_1 + h u'(a) + 0.5 h^2 u''(a) + O(h^3)] - y_1}{h} + d_2 y_1 = -d_3,$$

а для обчислення $u''(a)$ скористаємось неперервним продовженням диференціального рівняння на межу $x = a$, в результаті отримаємо

$$0.5 h d_1 [p(a) u(a) - f(a)] + O(h^2) + \underbrace{d_1 u'(a) + d_2 u(a)}_{(4.23)} = -d_3.$$

Порівнюючи цей вираз з (4.23) видно, що для їх $O(h^2)$ -рівності, нам необхідно "прибрати" величину $0.5 h d_1 [p(a) u(a) - f(a)]$, яка є порядку $O(h)$. Отже, різницева крайова умова точності $O(h^2)$ для (4.23), буде наступна:

$$-(d_1/h - d_2) y_1 + (d_1/h) y_2 = -d_3 + 0.5 h d_1 (p_1 y_1 - f_1),$$

яку, помноживши на $2h$, перетворимо до форми

$$-\eta_1 y_1 + \theta_1 y_2 = -\mu_1, \quad (4.23^*)$$

де уведено позначення

$$\eta_1 = 2d_1 - 2hd_2 + h^2 d_1 p_1, \quad \theta_1 = 2d_1, \quad \mu_1 = 2hd_3 + h^2 d_1 f_1.$$

Метод фіктивної (законтурної) точки. Нехай в правій граничній точці задано умову третього роду

$$d_4 u'(b) + d_5 u(b) = d_6, \quad |d_4| + |d_5| \neq 0. \quad (4.24)$$

Апроксимуємо $u'(b)$ виразом $0.5 h^{-1} (y_{N+1} - y_{N-1}) + O(h^2)$, а також використовуємо рівняння $y_{\bar{x},N} - p_N y_N = -f_N$, маємо :

$$\begin{cases} d_4 (y_{N+1} - y_{N-1}) + 2hd_5 y_N = 2hd_6, \\ d_4 (y_{N-1} - 2y_N + y_{N+1}) - h^2 d_4 p_N y_N = -h^2 d_4 f_N. \end{cases}$$

Після виключення y_{N+1} з цієї системи рівнянь, отримаємо різницеву умову, яка апроксимує крайову умову (4.24) з другим порядком точності

$$2d_4 y_{N-1} - (2d_4 + 2hd_5 + h^2 d_4 p_N) y_N = -2hd_6 - h^2 d_4 f_N,$$

і перетворимо до форми

$$\eta_2 y_{N-1} - \theta_2 y_N = -\mu_2, \quad (4.24^*)$$

де уведено позначення

$$\eta_2 = 2d_4, \quad \theta_2 = 2d_4 + 2hd_5 + h^2 d_4 p_N, \quad \mu_2 = 2hd_6 + h^2 d_4 f_N.$$

Розглянемо застосування МРС до розв'язання наступної квазілінійної КЗ :

$$\begin{cases} u''(x) = -f(x, u), & x \in (a, b); \\ u(a) = d_1, & u(b) = d_2. \end{cases} \quad (4.25)$$

Аналогічно побудові РС (4.21) для лінійної задачі, на рівномірній сітці з триточковим шаблоном отримаємо нелінійну різницеву задачу, записану в формі системи нелінійних рівнянь :

$$\bar{\mathbf{F}}(\bar{\mathbf{y}}) = \bar{\mathbf{0}} \Leftrightarrow \begin{cases} y_1 - d_1 = 0, \\ y_{k-1} - 2y_k + h^2 f(x_k, y_k) + y_{k+1} = 0, & k = \overline{2, N-1}; \\ y_N - d_2 = 0. \end{cases} \quad (4.26)$$

Для РЗ (4.26) можна довести: порядок похибки $O(h^2)$; коректність; збіжність РС з другим порядком при умові, що існують похідні $u_x^{(4)}(x)$, $f'_u(x, u)$ і виконуються обмеження $|u_x^{(4)}| \leq M_4$, $0 < m_1 \leq |f'_u| \leq M_1$.

Виникає питання, як знайти розв'язок отриманої нелінійної системи (4.26) ? Враховуючи, що можливі варіанти методу послідовних наближень або не збігаються, або збіжність лінійна, то використовують ітераційний метод Ньютона.

Алгоритм

1. Покладемо $y_k^{(0)} = d_1 + \frac{d_2 - d_1}{b - a}(x_k - a)$, $k = \overline{1, N}$;

Для ітерацій $s = 0, 1, \dots$,

2. Представимо $y_k^{(s+1)} = y_k^{(s)} + \Delta_k$, $k = \overline{1, N}$;
3. Підставимо $y_k^{(s+1)}$ в (4.26), виконаємо лінеаризацію, отримаємо СЛАР для знаходження $\bar{\Delta}$

$$\begin{cases} \Delta_1 = 0, \\ \Delta_{\bar{x}x,k} + f'_u(x_k, y_k^{(s)}) \Delta_k = -\left(f(x_k, y_k^{(s)}) + y_{\bar{x}x,k}^{(s)}\right), & k = \overline{2, N-1}; \\ \Delta_N = 0. \end{cases} \quad (4.27)$$

яка ефективно розв'язується *методом немонотонної прогонки*.

4. Пункти 2, 3 повторюємо до виконання $\|\Delta_k\|_C \leq \varepsilon \vee s > \max s$ – умова закінчення ітераційного процесу. Тут $0 < \varepsilon < 1$ – задана точність, а $\max s > 0$ – максимальна кількість кроків ітерацій (страховка від зациклювання);

Умова збіжності розглянутого ітераційного процесу $\left\| \frac{f''_{uu}}{2f'_u} \right\|_C < 1$ і збіжність квадратична.

Аудиторне завдання:

п.1. Використовуючи РС з апроксимацією порядку $O(h^2)$ на рівномірній сітці $\bar{X}$ розв'язати модельну крайову задачу :

$$\begin{cases} u''(x) - p(x)u(x) = -f(x), & x \in (a, b); & p(x) \geq 0; \\ d_1 u'(a) + d_2 u(a) = -d_3, \\ d_4 u'(b) + d_5 u(b) = d_6. \end{cases} \quad (A1)$$

Вхідні дані наступні: $p(x) = 4 + x$, $f(x) = 6 + x(5 + x) - (8 + x)\cos^2(x)$; $[a, b] = [0, 3]$; $D = [2, -1, -2, 3, 2, 2, 8.201583218752855]$, а $u(x) = x + \sin^2(x)$ – точний розв’язок. Побудувати графік наближеного числового розв’язку $\bar{y}$ і графік абсолютної похибки $|\bar{y} - u(\bar{X})|$. Дослідити поведінку похибки $\|\bar{y} - u(\bar{X})\|_C$ для $h = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$ і порівняти отримані результати з теоретичними оцінками.

Різницєва задача для КЗ (A1) побудується на основі РЗ (4.21), в якій потрібно замінити крайові умови в точках $x = a, b$ на розрахункові формули, відповідно (4.23*), (4.24*). Програмна реалізація для цієї РЗ виконана в функції **m_grid2**, а отже створимо відповідну М-функцію *p06a01* :

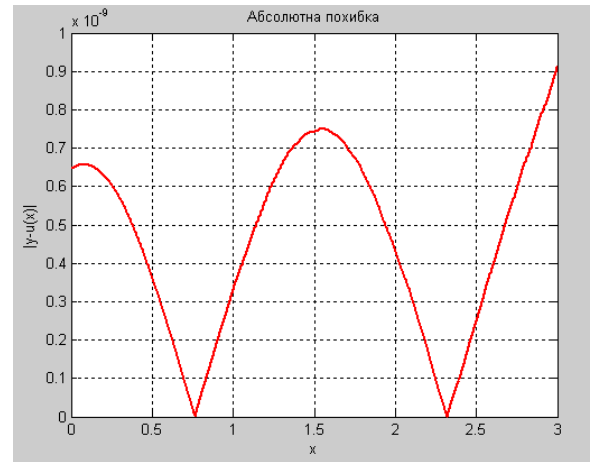
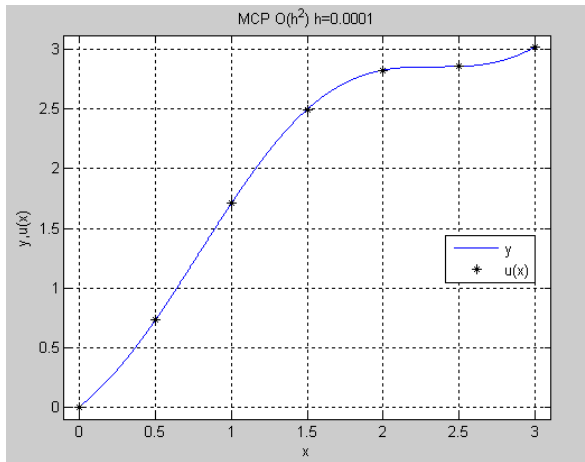
```
function p06a01
% Розв'язання модельної задачі для КЗ :
%   U''(x) - p(x)U(x) = -f(x),
%   p(x) >= 0,   x ∈ (a,b);
%   D(1)*U'(a) + D(2)*U(a) = -D(3),
%   D(4)*U'(b) + D(5)*U(b) = D(6).
% Для p(x)=4+x, f(x)=6+x(5+x)-(8+x)cos(x)^2, x∈[0,3] і
%   D=[2,-1,-2,3,2,2,8.201583218752855]
% розв'язок буде u(x)=x+sin(x)^2.
clear all, close all, clc
% Вхідні дані :
a = 0; b = 3;
p = @(x) (4 + x);
f = @(x) (6 + x.*(5 + x) - (8 + x).*cos(x).^2);
u = @(x) (x + sin(x).^2); % точний розв'язок
D = [2,-1,-2,...
     3, 2, 8.201583218752855];
fprintf(' h      ||y-u(X)||c\n');
fm = '%7.4f %14.6e\n'; k = 5;
for n = [31,301,3001,30001]
    % Знаходження наближеного розв'язку :
    [X, Y] = m_grid2(p, f, a, b, D, n);
    h = X(2)-X(1);
    U = u(X); % проекція u(x) на сітку X
    m = min(U); M = max(U); I = [1:k:n]; k = 10*k;
    close all;
    plot(X, Y, 'b -', X(I), U(I), 'k*'); grid on;
    s = sprintf('MCP O(h^2) h=%6.4f', h); title(s);
    de = 1e-1; axis([a-de, b+de, m-de, M+de]);
    xlabel('x'); ylabel('y,u(x)');
    legend('y','u(x)', 'Location', 'Best');
    pause(5); close all;
    % Візуалізація абсолютної похибки :
    U = abs(Y - U); M = max(U);
    plot(X, U, 'r', 'LineWidth', 1.5); grid on;
    title('Абсолютна похибка');
    xlabel('x'); ylabel('|y-u(x)|');
    fprintf(fm, h, M);
    pause(3);
end
end
```

і отримаємо наступний результат її виконання (показано тільки кінцеві результати):

```

h          ||y-u(x)||c
0.1000    9.241145e-004
0.0100    9.272511e-006
0.0010    9.272341e-008
0.0001    9.094894e-010

```



п.2. Для крайової задачі (A1) з $p(x)=P$, $P \in \text{Const}$ при $P=4$, $f(x)=6+4x-8\cos^2(x)$, а решта параметрів залишається без зміни, знайти на рівномірній сітці $\bar{X}$ розв'язки $\bar{y}_1, \bar{y}_2$ з порядком апроксимації, відповідно $O(h^2), O(h^4)$. Виведення результатів організувати аналогічно вимог задачі **п.1**, але тільки для кроку $h=10^{-2}$.

Щоб чисельно розв'язати КЗ (A1) з порядком $O(h^2)$ використаємо функцію **m_grid2**, а для розв'язку з порядком $O(h^4)$ – створимо функцію **m_grid4**.

Для написання М-функції **m_grid4** необхідно спочатку побудувати різницеву задачу, яка з порядком $O(h^4)$ апроксимує вихідну КЗ. Скористаємось теоретичною інформацією з п.2 Лекції №04. Там, для диференціального рівняння $u''(x) - Pu(x) = -f(x)$, на шаблоні $S\{x_{k-1}, x_k, x_{k+1}\}$, було отримано різницеве рівняння

$$y_{\bar{x}x,k} - \tilde{\eta} P y_k = -\tilde{\eta} f_k - \tilde{h} f_k'', \quad k=2, N-1, \quad (4.21^{**})$$

(позначено $\tilde{h}=h^2/12$, $\tilde{\eta}=1+\tilde{h}P$) з необхідним нам порядком $O(h^4)$. Тепер потрібно отримати апроксимацію крайових умов (4.23), (4.24) також з порядком $O(h^4)$. Використаємо для цього метод підвищення порядку апроксимації на розв'язку.

Маємо наступний ланцюжок перетворень при отриманні апроксимації КУ III-роду (4.23) з порядком $O(h^4)$:

1	$d_1 \frac{y_2 - y_1}{h} + d_2 y_1 = -d_3;$
2	$d_1 \frac{(y_1 + h u'(a) + \frac{1}{2} h^2 u^{(2)}(a) + \frac{1}{6} h^3 u^{(3)}(a) + \frac{1}{24} h^4 u^{(4)}(a) + O(h^5)) - y_1}{h} + d_2 y_1 = -d_3;$
3	$d_1 \left\{ \frac{1}{2} h [P u(a) - f(a)] + \frac{1}{6} h^2 [P u'(a) - f'(a)] + \frac{1}{24} h^3 [P * (P u(a) - f(a)) - f''(a)] \right\} + O(h^4) + d_1 u'(a) + d_2 y_1 = -d_3;$
4	$d_1 \left\{ \frac{1}{2} h \left(1 + \frac{1}{12} h^2 P \right) P u(a) - \frac{1}{2} h \left(1 + \frac{1}{12} h^2 P \right) f(a) - \frac{1}{6} h^2 (f'(a) + \frac{1}{4} h f''(a)) \right\} + \frac{1}{6} h^2 P [d_1 u'(a) \pm d_2 u(a) \pm d_3] + O(h^4) = 0;$

5	$\left[\frac{1}{2} h P d_1 \left(1 + \frac{1}{12} h^2 P \right) - \frac{1}{6} h^2 P d_2 \right] \mu(a) - \frac{1}{6} h^2 P d_3 - \frac{1}{2} h d_1 \left(1 + \frac{1}{12} h^2 P \right) f(a) -$ $- \frac{1}{6} h^2 d_1 \left(f'(a) + \frac{1}{4} h f''(a) \right) + O(h^4) = 0;$
---	--

З останнього рівняння видно, який вираз порядку $O(h^4)$ необхідно з нього "прибрати", щоб мати похибку необхідного нам порядку. Отже, різницева крайова умова точності $O(h^4)$ для (4.23) буде така :

$$-(d_1/h - d_2)y_1 + (d_1/h)y_2 = -d_3 +$$

$$+ \left[\frac{1}{2} h P d_1 \left(1 + \frac{1}{12} h^2 P \right) - \frac{1}{6} h^2 P d_2 \right] y_1 - \frac{1}{6} h^2 P d_3 - \frac{1}{2} h d_1 \left(1 + \frac{1}{12} h^2 P \right) f_1 - \frac{1}{6} h^2 d_1 \left(f'_1 + \frac{1}{4} h f''_1 \right),$$

яку перетворимо до форми

$$-\tilde{\eta}_1 y_1 + \tilde{\theta}_1 y_2 = -\tilde{\mu}_1, \quad (4.23^{**})$$

де уведено позначення

$$\tilde{\eta}_1 = d_1/h + \frac{1}{2} h P d_1 \left(1 + \frac{1}{12} h^2 P \right) - d_2 \left(1 + \frac{1}{6} h^2 P \right), \quad \tilde{\theta}_1 = d_1/h,$$

$$\tilde{\mu}_1 = d_3 \left(1 + \frac{1}{6} h^2 P \right) + \frac{1}{2} h d_1 \left(1 + \frac{1}{12} h^2 P \right) f_1 + \frac{1}{6} h^2 d_1 \left(f'_1 + \frac{1}{4} h f''_1 \right).$$

Отримаємо різницеве рівняння для апроксимації КУ III-роду (4.24). Маємо наступний ланцюжок перетворень :

1	$d_4 \frac{y_N - y_{N-1}}{h} + d_5 y_N = d_6;$
2	$d_4 \frac{y_N - \left(y_N - h u'(b) + \frac{1}{2} h^2 u^{(2)}(b) - \frac{1}{6} h^3 u^{(3)}(b) + \frac{1}{24} h^4 u^{(4)}(b) + O(h^5) \right)}{h} + d_5 y_N = d_6;$
3	$-d_4 \left\{ \frac{1}{2} h [P u(b) - f(b)] - \frac{1}{6} h^2 [P u'(b) - f'(b)] + \frac{1}{24} h^3 [P * (P u(b) - f(b)) - f''(b)] \right\} + O(h^4) = 0;$
4	$- \frac{1}{2} h d_4 P \left(1 + \frac{1}{12} h^2 P \right) \mu(b) + \frac{1}{2} h d_4 \left(1 + \frac{1}{12} h^2 P \right) f(b) - \frac{1}{6} h^2 d_4 \left(f'(b) - \frac{1}{4} h f''(b) \right) +$ $+ \frac{1}{6} h^2 P (d_4 u'(b) \pm d_5 u(b) \mp d_6) + O(h^4) = 0;$
5	$- \left[\frac{1}{2} h d_4 P \left(1 + \frac{1}{12} h^2 P \right) + \frac{1}{6} h^2 P d_5 \right] \mu(b) + \frac{1}{6} h^2 P d_6 + \frac{1}{2} h d_4 \left(1 + \frac{1}{12} h^2 P \right) f(b) -$ $- \frac{1}{6} h^2 d_4 \left(f'(b) - \frac{1}{4} h f''(b) \right) + O(h^4) = 0;$

Визначились, що потрібно "прибирати", отже маємо вираз для різницевої КУ точності $O(h^4)$ для (4.24) :

$$-(d_4/h)y_{N-1} + (d_4/h + d_5)y_N = d_6 -$$

$$- \left[\frac{1}{2} h d_4 P \left(1 + \frac{1}{12} h^2 P \right) + \frac{1}{6} h^2 P d_5 \right] y_N + \frac{1}{6} h^2 P d_6 + \frac{1}{2} h d_4 \left(1 + \frac{1}{12} h^2 P \right) f_N - \frac{1}{6} h^2 d_4 \left(f'_N - \frac{1}{4} h f''_N \right)$$

Отриману різницеву КУ представимо у формі

$$\tilde{\eta}_2 y_{N-1} - \tilde{\theta}_2 y_N = -\tilde{\mu}_2, \quad (4.24^{**})$$

де уведено позначення

$$\tilde{\eta}_2 = d_4/h, \quad \tilde{\theta}_2 = d_4/h + \frac{1}{2} h d_4 P \left(1 + \frac{1}{12} h^2 P \right) + \left(1 + \frac{1}{6} h^2 P \right) d_5,$$

$$\tilde{\mu}_2 = \left(1 + \frac{1}{6} h^2 P \right) d_6 + \frac{1}{2} h d_4 \left(1 + \frac{1}{12} h^2 P \right) f_N - \frac{1}{6} h^2 d_4 \left(f'_N - \frac{1}{4} h f''_N \right).$$

Реалізуємо отримані компоненти (4.21**), (4.23**), (4.24**) РЗ в М-функції **m_grid4** і тепер можна написати М-функцію **p06a02**. В цій функції залишено фрагмент коду для тестування поведінки абсолютної похибки для усіх можливих комбінацій крайових умов I, II, III роду. При бажанні можна вибрати необхідний варіант тесту і порівняти результати.

function p06a02

% Розв'язання модельної задачі для КЗ :

*% $U''(x) - p * U(x) = -f(x)$,*

% $p \in Const, p \geq 0, x \in (a, b)$;

*% $D(1) * U'(a) + D(2) * U(a) = -D(3)$,*

```

% D(4)*U'(b) + D(5)*U(b) = D(6).
% Для p=4, f(x)=6+4x-8cos(x)^2, x∈[0,3] і
% D=[2,-1,-2,3,2,8.201583218752855]
% розв'язок буде u(x)=x+sin(x)^2.
clear all, close all, clc
% Вхідні дані :
a = 0; b = 3;
p = @(x) (4); P = 4;
f = @(x) (6+4*x-8*cos(x).^2);
u = @(x) (x + sin(x).^2); % точний розв'язок
%D = [0, 1, 0, 0, 1, u(b)]; % I p, I p
%D = [0, 1, 0, 1, 0, 1+sin(2*b)]; % I p, II p
%D = [0, 1, 0, 3, 2, 8.201583218752855]; % I p, III p
%D = [1, 0,-1, 0, 1, u(b)]; % II p, I p
%D = [1, 0,-1, 1, 0, 1+sin(2*b)]; % II p, II p
%D = [1, 0,-1, 3, 2, 8.201583218752855]; % II p, III p
%D = [2, 1,-2, 0, 1, u(b)]; % III p, I p
%D = [2,-1,-2, 1, 0, 1+sin(2*b)]; % III p, II p
D = [2,-1,-2, 3, 2, 8.201583218752855]; % III p, III p
n = 301; k = 50;
% Знаходження наближеного розв'язку :
[X, Y1] = m_grid2(p, f, a, b, D, n);
h = X(2)-X(1); U = u(X); % проєкція u(x) на сітку X
m = min(U); M = max(U); I = [1:k:n];
de = 1e-1; de = [a-de, b+de, m-de, M+de];
plot(X, Y1, 'b', X(I), U(I), 'k*', 'LineWidth', 2);
grid on; title('MCP O(h^2)'); axis(de);
xlabel('x'); ylabel('Y1,u(x)');
legend('Y1','u(x)','Location','Best');
pause(5); close all;
E1 = abs(Y1 - U); mE = max(E1); % абсолютна похибка
fprintf(' h ||Y-u(X)||c\n');
fm = '%7.4f %14.6e\n';
fprintf(fm, h, mE);

[X, Y2] = m_grid4(P, @fd2f, @df1, a, b, D, n);
plot(X, Y2, 'r', X(I), U(I), 'k*', 'LineWidth', 2);
grid on; title('MCP O(h^4)'); axis(de);
xlabel('x'); ylabel('Y2,u(x)');
legend('Y2','u(x)','Location','Best');
pause(5); close all;
% Візуалізація абсолютних похибок :
E2 = abs(Y2 - U); mE = max(E2);
fprintf(fm, h, mE);
plot(X, E1, 'b', 'LineWidth', 1.5); hold on;
plot(X, 1e5.*E2, 'r', 'LineWidth', 1.5); grid on;
fm = '\{\psi\}';
title(['Абсолютна похибка ',fm,':=|Y-u(x)|']); hold off;
xlabel('x'); ylabel(fm);
fm = [fm,' - O(h^')];
legend([fm,'2'], ['10^5*',fm,'4'],'Location','Best');
format long e
disp(' U Y1 Y2');
disp([U(I)',Y1(I)',Y2(I)']);
end

function [f, f2] = fd2f(x)
% Опис f(x), f''(x)
f = 6+4*x-8*cos(x).^2;

```

```

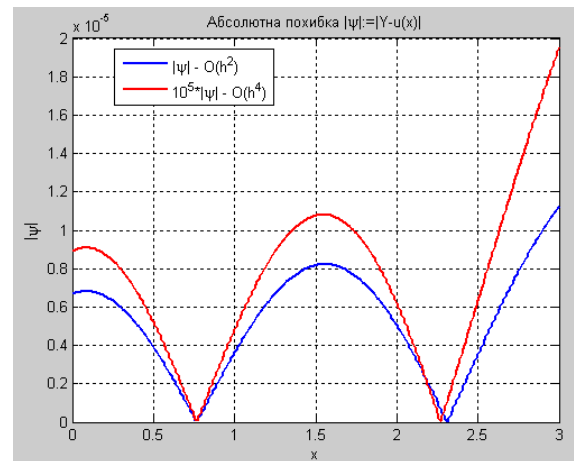
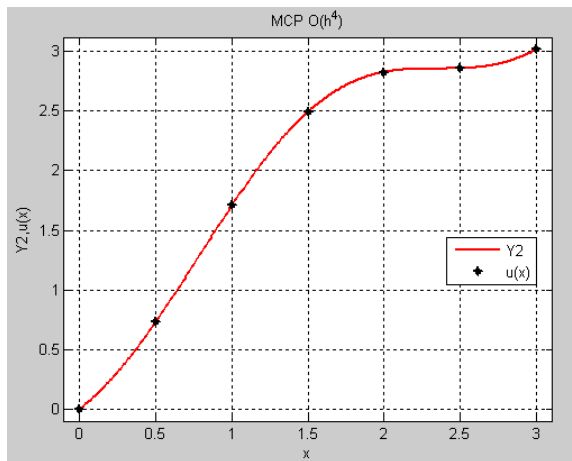
f2 = 16*cos(2*x);
end

function df = df1(x)
% Опис f'(x)
df = 4+8*sin(2*x);
end

```

Отримано такий результат виконання *p06a02* (показано графіки $\bar{y}_2$ та похибки – $|\psi(x_k)|$ для $\bar{y}_1$ і $10^5 * |\psi(x_k)|$ для $\bar{y}_2$):

h	Y-u(X) c	Y1	Y2
0.0100	1.123379e-005		
0.0100	1.954299e-010		
U		Y1	Y2
0		-6.679592742803511e-006	8.897953901465527e-011
7.298488470659301e-001		7.298449340719683e-001	7.298488471180069e-001
1.708073418273571e+000		1.708077051800217e+000	1.708073418225266e+000
2.494996248300223e+000		2.495004419862950e+000	2.494996248192618e+000
2.826821810431806e+000		2.826826849896179e+000	2.826821810369851e+000
2.858168907268387e+000		2.858165363957070e+000	2.858168907331716e+000
3.019914856674817e+000		3.019903622880140e+000	3.019914856870247e+000



Завдання для самостійної роботи:

п.1. Методом скінченних різниць з рівномірним кроком $h=0.01$, в поєднанні з методом лінеаризації, розв'язати квазілінійну крайову задачу

$$\begin{cases} u''(x) - p(x)u(x) = -f(x, u(x)), & p(x) \geq 0, \quad x \in (a, b); \\ u(a) = d_1, \quad u(b) = d_2. \end{cases} \quad (4.28)$$

Тут позначено :

- ♦ $f(x, u) = \exp(u/g - x) - \sin(x * u)$;
- ♦ $p(x), a, b$ – беремо з Таблиці 4.1;
- ♦ $d_1 = d/10 - g, \quad d_2 = d/10 + g, \quad \mathbf{D} = [d_1, d_2]$

для значень $\mathbf{k}, \mathbf{g}, \mathbf{d}$ – варіанту індивідуального завдання.

Вказівка

1. Очевидно, що для побудови РЗ, яка з порядком $O(h^2)$ апроксимує (4.28), можна взяти РЗ (4.26), яку необхідно доповнити елементом $-p_k u_k$, тоді замість (4.27) отримаємо:

$$\begin{cases} \Delta_1 = 0, \\ \Delta_{k-1} - \left(2 + h^2 p_k + h^2 f'_u(x_k, y_k^{(s)})\right) \Delta_k + \Delta_{k+1} = -\left(y_{k-1}^{(s)} - \left(2 + h^2 p_k\right) y_k^{(s)} + y_{k+1}^{(s)} + h^2 f(x_k, y_k^{(s)})\right), \\ k = \overline{2, N-1}; \\ \Delta_N = 0. \end{cases}$$

Це частково модифікує алгоритм, розглянутий в задачі (4.25)-(4.27).

2. Написати М-функцію **m_grid2n**

`function [X,Y,exf,s] = m_grid2n(pfun,fdfun,a,b,D,n,ep,maxs)`

Тут вихідні/вхідні аргументи співпадають з параметрами функції **m_grid2**, окрім *fdfun* – опис пари $[f(x,u), f'_u(x,u)]$, а також в список вхідних параметрів добавлено: *ep* – задана точність ($0 < \varepsilon < 1$); *maxs* – максимальна кількість кроків ітерацій. В список вихідних, природно, добавлено: *exf* – код закінчення (0=нормальне | 1=використано ітерацій $> \text{maxs}$), *s* – затрачено ітерацій.

Програмне забезпечення на MATLAB:

```
function [X, Y] = m_grid2(pfun, ffun, a, b, D, n)
% Наближений розв'язок лінійного диференціального рівняння
%   u''(x) - p(x)*u(x) = - f(x),
%   p(x) >= 0,   x є (a,b),
% для якого виконуються крайові умови
%   D(1)*u'(a) + D(2)*u(a) = -D(3),
%   D(4)*u'(b) + D(5)*u(b) = D(6).
% (різницева схемах 2-го порядку точності).
% Для параметрів D(1),D(2) і D(4),D(5) виконуються умова :
% abs(D(1)) + abs(D(2)) <> 0 & abs(D(4)) + abs(D(5)) <> 0.
% Вхідні дані :
%   pfun   - функція з описом p(x);
%   ffun   - функція з описом правої частини f(x);
%   a, b   - границі відрізка [a,b];
%   D      - масив значень коефіцієнтів крайових умов;
%   n      - число точок сітки різницевої схеми.
% Вихідні дані:
%   X      - масив значень вузлів сітки;
%   Y      - масив значень наближеного розв'язку.
if nargin<6
    error('M_GRID2:NotEnoughInputs',...
        'Not enough input arguments. See M_GRID2. ');
    return
end
if length(D)<6
    error('M_GRID2:NotEnoughInputs',...
        'Not enough input arguments D. See M_GRID2. ');
    return
end
if abs(D(1)) + abs(D(2)) == 0 | abs(D(4)) + abs(D(5)) == 0
    error('M_GRID2:NotEnoughInputs',...
        'Not enough input arguments D. See M_GRID2. ');
    return
end
if a >= b
    error('M_GRID2:error a>=b.', 'a>=b. See M_GRID2. ');
    return
end
% формування вузлів
```

```

X = linspace(a, b, n); h = X(2) - X(1);
% допоміжні змінні
hh = h.*h; h2 = 2.*h;
% формування коефіцієнтів
% A(k), C(k), B(k), F(k), k=1,...,n для СЛАР
A = ones(n, 1); B = A; C = A; F = A;
A(1) = 0; B(1) = 2.*D(1); x = D(1).*hh;
C(1) = B(1) - D(2).*h2 + x.*pfun(a);
F(1) = D(3).*h2 + x.*ffun(a);
for k = 2 : n-1
    x = X(k); C(k) = 2 + hh.*pfun(x); F(k) = hh.*ffun(x);
end
A(n) = 2.*D(4); B(n) = 0; x = D(4).*hh;
C(n) = A(n) + D(5).*h2 + x.*pfun(b);
F(n) = D(6).*h2 + x.*ffun(b);
% знаходження сіткової функції при розв'язанні СЛАР
% методом монотонної (лівої) прогонки, а у випадку
% невиконання умов стійкості - методом немонотонної прогонки
[Y AL er] = m_progm(A, C, B, F);
if er
    [Y AL] = m_progn(A, C, B, F);
end
return

function [y, alfa, err] = m_progm(a, c, b, f)
% метод монотонної (правої) прогонки для розв'язання
% СЛАР з тридіагональною матрицею
% a(k)*y(k-1) - c(k)*y(k) + b(k)*y(k+1) = - f(k),
% a(1) = b(n) = 0,
% при виконанні умов :
% |c(k)| >= |a(k)| + |b(k)|,
% k= 1,2,...,n.
% вхідні дані:
% a, -c, b - піддіагональ, діагональ і наддіагональ матриці СЛАР;
% -f - права частина.
% вихідні дані:
% y - розв'язок;
% alfa - прогоночні коефіцієнти;
% err - 1= умови застосування методу не виконуються і
% розв'язок не знайдено,
% 0= розв'язок знайдено.
n = length(c); err = 0; k = 0;
while k < n & ~err
    k = k + 1;
    err = err | abs(c(k)) < abs(a(k)) + abs(b(k));
end
alfa = zeros(1, n); y = alfa;
if err
    return
end
% пряма прогонка :
nm = n - 1; beta = alfa;
for k = 1 : nm
    j = k + 1; z = c(k) - a(k).*alfa(k);
    alfa(j) = b(k)./z; beta(j) = (f(k) + a(k).*beta(k))./z;
end
% зворотна прогонка :
y(n) = (f(n) + a(n).*beta(n))./(c(n) - a(n).*alfa(n));
for k = nm : -1 : 1

```

```

    j = k + 1; y(k) = alfa(j).*y(j) + beta(j);
end
return

function [y, alfa] = m_progn(a, c, b, f)
% метод немонотонної прогонки для розв'язання СЛАР
%   з тридіагональною матрицею
% (метод Гауса з вибором провідного елемента по рядку)
%   a(k)*y(k-1) - c(k)*y(k) + b(k)*y(k+1) = - f(k)
%   k= 1,2,...,n;   a(1)=b(n)=0.
%   вхідні дані:
%   a, -c, b - піддіагональ, діагональ і наддіагональ матриці СЛАР;
%   -f - права частина.
%   вихідні дані:
%   y - розв'язок;
%   alfa - прогоночні коефіцієнти.
n = length(c); nm = n - 1;
% пряма прогонка:
alfa = zeros(1, n); y = alfa;
beta = alfa; ka = ones(1, n); te = alfa;
C = c(1); A = a(2); F = f(1); Q = f(2);
for k = 1 : nm
    j = k + 1;
    if abs(C) >= abs(b(k))
        alfa(j) = b(k)./C; beta(j) = F./C;
        C = c(j) - A.*alfa(j); F = Q + A.*beta(j);
        te(j) = ka(k); ka(j) = j;
        if k ~= nm
            j = j + 1; A = a(j); Q = f(j);
        end
    else
        alfa(j) = C./b(k); beta(j) = -F./b(k);
        C = c(j).*alfa(j) - A; F = Q - c(j).*beta(j);
        te(j) = j; ka(j) = ka(k);
        if k ~= nm
            j1 = j; j = j + 1;
            A = a(j).*alfa(j1); Q = f(j) + a(j).*beta(j1);
        end
    end
end
end
% зворотня прогонка:
y(ka(n)) = F./C;
for k = nm : -1 : 1
    j = k + 1; y(te(j)) = alfa(j).*y(ka(j)) + beta(j);
end
return

```

```

function [X, Y] = m_grid4(p, ffun, dffun, a, b, D, n)
% Наближений розв'язок лінійного диференціального рівняння
%   u''(x) - p*u(x) = - f(x),
%   p є Const, p>0, x є (a,b),
% для якого виконуються крайові умови
%   D(1)*u(a) + D(2)*u'(a) = -D(3),
%   D(4)*u(b) + D(5)*u'(b) = D(6).
% (різницева схема 4-го порядку точності).
% Для параметрів D(1),D(2) і D(4),D(5) виконуються умова :
% abs(D(1)) + abs(D(2)) <> 0 & abs(D(4)) + abs(D(5)) <> 0.
% Вхідні дані :
%   p - коефіцієнт в рівнянні;

```

```

% ffun - функція для обчислення  $[f(x), f''(x)]$ ;
% dffun - функція для обчислення  $f'(x)$ ;
% a, b - границі відрізка  $[a, b]$ ;
% D - масив значень коефіцієнтів крайових умов;
% n - число точок сітки різницевої схеми.
% Вихідні дані:
% X - масив значень вузлів сітки;
% Y - масив значень наближеного розв'язку.
if nargin<7
    error('M_GRID4:NotEnoughInputs',...
        'Not enough input arguments. See M_GRID4.');
```

```

    return
end
if length(D)<6
    error('M_GRID4:NotEnoughInputs',...
        'Not enough input arguments D. See M_GRID4.');
```

```

    return
end
if abs(D(1)) + abs(D(2)) == 0 | abs(D(4)) + abs(D(5)) == 0
    error('M_GRID4:NotEnoughInputs',...
        'Not enough input arguments D. See M_GRID4.');
```

```

    return
end
if a >= b
    error('M_GRID4:error a>=b.', 'a>=b. See M_GRID4.');
```

```

    return
end
% формування вузлів
X = linspace(a, b, n); h = X(2) - X(1);
% допоміжні змінні
hh = h.*h; hh6 = hh./6; hh12 = hh./12;
h4 = 0.25.*h; h2 = 0.5.*h;
ph6 = 1 + p.*hh6; ph12 = 1 + p.*hh12;
e1 = ph12.*hh; hv = hh12.*hh; x = 2 + p.*e1;
% формування коефіцієнтів
% A(k), C(k), B(k), F(k), k=1,...,n для СЛАР
A = ones(1, n); B = A; C = x.*A; F = A;
% враховуємо крайові умови при x = a
A(1) = 0; B(1) = D(1)./h;
C(1) = B(1) + h2.*p.*D(1).*ph12 - D(2).*ph6;
[f, f2] = ffun(a); f1 = dffun(a);
F(1) = D(3).*ph6 + D(1).*(h2.*ph12.*f + hh6.*(f1 + h4.*f2));
% обчислюємо праву частину для i=2:n-1 :
for i = 2 : n-1
    x = X(i); [f, f2] = ffun(x); F(i) = e1.*f + hv.*f2;
end
% враховуємо крайові умови при x = b
A(n) = D(4)./h; B(n) = 0;
C(n) = A(n) + h2.*p.*ph12.*D(4) + ph6.*D(5);
[f, f2] = ffun(b); f1 = dffun(b);
F(n) = ph6.*D(6) + D(4).*(h2.*ph12.*f - hh6.*(f1 - h4.*f2));
% розв'язок СЛАР методом немонотонної прогонки :
[Y, alfa] = m_progn(A, C, B, F);
return

```

Практичне заняття № 07

ТЕМА № 4. Розв'язування крайових задач для ЗДР.

4.4. Інтегро-інтерполяційний метод побудови різницевих схем для лінійних задач з неперервними або кусково-неперервними коефіцієнтами (2 год).

Теоретична інформація:

В Лекції №05 ми розглянули використання інтегро-інтерполяційного методу (ІМ) для побудови різницевих схем відповідної якості на прикладі лінійної крайової задачі, яка описує стаціонарний розподіл тепла в стержні довжини $b - a$. Такий процес моделюється ЗДР другого порядку

$$W'(x) - p(x)u(x) = -f(x), \quad x \in D \equiv (a, b);$$
$$\bar{D} = D \cup \partial D, \quad \partial D = a \cup b \quad (4.29)$$

де функція $-W(x) = -K(x)u'(x)$ – тепловий потік, $K(x) \geq K_0 > 0$ – коефіцієнт теплопровідності, а параметри $p(x) \geq 0$, $f(x)$ – характеризують щільність теплових джерел.

В залежності від прикладної задачі, рівняння (4.29) доповнюються відповідними крайовими умовами

- ◆ I роду – в граничній точці підтримується стала температура;
- ◆ II роду – в граничній точці задано тепловий потік;
- ◆ III роду – в граничній точці є теплообмін із зовнішнім середовищем за законом Ньютона.

З точки зору математичної моделі крайових умов, маємо

Гранична точка	КУ I роду	КУ II, III роду
$x = a$	$u(a) = d_1$	$-K(a)u'(a) + g_1u(a) = d_1, \quad g_1 \geq 0$
$x = b$	$u(b) = d_2$	$K(b)u'(b) + g_2u(b) = d_2, \quad g_2 \geq 0$

Звичайно, в постановці конкретної задачі можливі різні комбінації умов I, II і III роду, наприклад, в $x = a$ маємо КУ III роду, а в $x = b$ – КУ I роду (саме таку комбінацію КУ ми розглядали в Лекції №05).

В лекційній частині ми також розглянули теоретичні обґрунтування ІМ:

- апроксимацію $O(h^2)$ рівняння (4.29) і отримали *достатні умови апроксимації другого порядку*

$$\frac{A_{k+1} - A_k}{h} = K'(x_k) + O(h^2), \quad \frac{A_{k+1} + A_k}{2} = K(x_k) + O(h^2),$$

а для коефіцієнтів $P_k = p(x_k) + O(h^2)$, $F_k = f(x_k) + O(h^2)$ – це обчислення з аналогічною точністю;

- апроксимацію $O(h^2)$ крайової умови III роду, на прикладі КУ в точці $x = a$;
- для випадку неперервних коефіцієнтів $K(x)$, $p(x)$, $f(x)$, *методом енергетичних нерівностей* отримали апріорні оцінки, які гарантують *стійкість* різницевої схеми;
- виконали доведення збіжності розв'язку різницевої задачі до розв'язку диференціальної задачі. У випадку неперервних коефіцієнтів $K(x)$, $p(x)$, $f(x)$ маємо *другий порядок*.

Тепер зосередимось на практичних аспектах розв'язання крайових задач, які будемо наближено розв'язувати з використанням різницевих схем, побудованих ІМ.

При створенні відповідного програмного забезпечення (ПЗ) для рівняння (4.29) додамо КУ і будемо користуватися такими формами їх запису :

$$-\delta_1 K(a)u'(a) + g_1 u(a) = d_1, \quad (\delta_1 = 0 \& g_1 = 1) | (\delta_1 = 1 \& g_1 \geq 0), \quad (4.29a)$$

$$\delta_2 K(b)u'(b) + g_2 u(b) = d_2, \quad (\delta_2 = 0 \& g_2 = 1) | (\delta_2 = 1 \& g_2 \geq 0). \quad (4.29b)$$

З урахуванням обмежень на значення (δ_1, g_1) , (δ_2, g_2) , вказаних в (4.29a), (4.29b), маємо при $\delta = 0$ КУ I роду, а при $\delta = 1$ КУ II, III роду.

Для розробки ПЗ «зберемо» всі частини різницевої задачі, розглянутої в Лекції №05 – основне рівняння, КУ і представимо в формі СЛАР з конкретними формулами обчислення відомих коефіцієнтів системи.

Будемо використовувати рівномірну сітку $X = (x_k)_{k=1}^n$, означену наступним чином

$$\bar{D}_h = \{x_k = a + (k-1)h, \quad k = \overline{1, n}, \quad b - a = (n-1)h\},$$

а також позначення $x_{k \pm 0.5} = x_k \pm 0.5h$, $y_k = u(x_k)$, $y_{k \pm 0.5} = u(x_{k \pm 0.5})$.

Систему рівнянь різницевої задачі, яка ставиться у відповідність КЗ (4.29), (4.29a), (4.29b), можна записати в формі, зручній для застосування методу монотонної лівої|правої прогонки

$$\begin{cases} -c_1 y_1 + b_1 y_2 = -\varphi_1, & k = 1; \\ a_k y_{k-1} - c_k y_k + b_k y_{k+1} = -\varphi_k, & k = \overline{2, n-1}; \\ a_n y_{n-1} - c_n y_n = -\varphi_n, & k = n. \end{cases} \quad (4.30)$$

Тут введено позначення

$$k = 1: a_1 = 0; c_1 = \begin{cases} 1, & \delta_1 = 0 \\ A_2 + h(g_1 + 0.5hP_1), & \delta_1 = 1 \end{cases}; b_1 = \begin{cases} 0, & \delta_1 = 0 \\ A_2, & \delta_1 = 1 \end{cases}; \varphi_1 = \begin{cases} d_1, & \delta_1 = 0 \\ h(d_1 + 0.5hF_1), & \delta_1 = 1 \end{cases};$$

$$P_1 = \frac{1}{0.5h} \int_a^{a+0.5h} p(x) dx; \quad F_1 = \frac{1}{0.5h} \int_a^{a+0.5h} f(x) dx;$$

$$k = \overline{2, n-1}: a_k = A_k; \quad b_k = A_{k+1}; \quad c_k = a_k + b_k + h^2 P_k; \quad \varphi_k = h^2 F_k;$$

$$A_k = \left(\frac{1}{h} \int_{x_{k-1}}^{x_k} \frac{dx}{K(x)} \right)^{-1}; \quad P_k = h^{-1} \int_{x_k-0.5h}^{x_k+0.5h} p(x) dx; \quad F_k = h^{-1} \int_{x_k-0.5h}^{x_k+0.5h} f(x) dx; \quad (4.31)$$

$$k = n: a_n = \begin{cases} 0, & \delta_2 = 0 \\ A_n, & \delta_2 = 1 \end{cases}; c_n = \begin{cases} 1, & \delta_2 = 0 \\ A_n + h(g_2 + 0.5hP_n), & \delta_2 = 1 \end{cases}; b_n = 0; \varphi_n = \begin{cases} d_2, & \delta_2 = 0 \\ h(d_2 + 0.5hF_n), & \delta_2 = 1 \end{cases};$$

$$P_n = \frac{1}{0.5h} \int_{b-0.5h}^b p(x) dx; \quad F_n = \frac{1}{0.5h} \int_{b-0.5h}^b f(x) dx.$$

Рекомендується при обчислення коефіцієнтів A_k, P_k, F_k в (4.31) дотримуватися такої стратегії :

для неперервних коефіцієнтів $K(x), p(x), f(x)$ можна

1) Безпосередньо (точно) обчислювати інтеграли.

2) Використовувати наближене обчислення інтегралів з порядком $O(h^2)$, наприклад, квадратурні формули *центральных прямоугольников*

$$A_k = K(x_{k-0.5}), \quad P_k = p(x_k), \quad F_k = f(x_k); \quad (4.32.1)$$

або *трапецій*

$$A_k = \frac{2K(x_{k-1})K(x_k)}{K(x_{k-1}) + K(x_k)}, \quad P_k = \frac{p(x_{k-0.5}) + p(x_{k+0.5})}{2}, \quad F_k = \frac{f(x_{k-0.5}) + f(x_{k+0.5})}{2}. \quad (4.32.2)$$

для коефіцієнтів $K(x), p(x), f(x)$ з інтегрованими особливостями потрібно

1) Будувати сітку $\bar{D}_h$ потрібно так, щоб розриви $K(x)$ були в півцілих точках D_h ,

а $p(x)$, $f(x)$ в цілих точках D_h .

2) Обчислення A_k, P_k, F_k проводити по наступним формулам

$$\begin{aligned} A_k &= 0.5(K(x_{k-0.5} - 0) + K(x_{k-0.5} + 0)), \\ P_k &= 0.5(p(x_k - 0) + p(x_k + 0)), \quad F_k = 0.5(f(x_k - 0) + f(x_k + 0)). \end{aligned} \quad (4.33)$$

При створенні ПЗ потрібно врахувати, що маємо працювати не тільки з неперервними функціями $K(x)$, $p(x)$, $f(x)$, а також з функціями, які мають інтегровані особливості, означені в певних індексах j сітки X . В загальному випадку, це мають бути відповідні множини індексів (індексні масиви в MATLAB) для кожної з функцій $K(x)$, $p(x)$, $f(x)$:

- множина $TrK = \{1 \leq j_K < n\}$ з відповідністю обчислення значень $X(j_K) + 0.5h$;
- множини $Trp = \{1 < j_p < n\}$, $Trf = \{1 < j_f < n\}$ – обчислення значень $X(j_p)$, $X(j_f)$.

З огляду цієї ситуації, звичний функціонал $K(x)$, $p(x)$, $f(x)$ опису функції (маємо на увазі реалізацію в СКМ MATLAB), наприклад, функції $f(x)$

```
function e = f(x)
    e = обчислення f(x)
end
```

краще замінити наступною схемою реалізації (за наявності не пустої множини Trf)

```
function e = f(j)
    global X n h25 Trf nf
    % X-вектор сітки, n-його розмірність, h25-значення h/4
    % Trf-вектор з номерами індексів X, де розташована точка розриву
    % nf-розмірність вектора Trf
    x = X(j);
    k = find(j == Trf(:));
    if isempty(k)
        e = обчислення f(x) в неперервній точці по формулам (4.32)
    elseif k == 1
        e = обчислення f(x) в точці 1-го розриву по формулі (4.33)
    elseif k == 2
        e = обчислення f(x) в точці 2-го розриву по формулі (4.33)
    . . .
    elseif k == nf
        e = обчислення f(x) в останній точці розриву по формулі (4.33)
    end
end
```

а у разі пустої множини Trf – маємо відповідно реалізацію

```
function e = f(j)
    global X n
    x = X(j);
    e = обчислення f(x) в неперервній точці по формулам (4.32)
end
```

Використовуючи (4.30)–(4.31), рекомендації (4.32)–(4.33) і міркування відносно універсальності функціоналу опису функцій K , p , f можемо створити М-функцію

```
function Y = m_iim(Kfun, pfun, ffun, D)
```

де вхідні аргументи $Kfun, pfun, ffun$ – це завдання функцій K, p, f ; D – вектор-рядок зі значеннями констант КУ (4.29a), (4.29b), тобто $D = [\delta_1, g_1, d_1, \delta_2, g_2, d_2]$. Вихідний параметр Y – наближений розв’язок крайової задачі в вузлах масива X .

Враховуючи, що інформація про $X, n, h, h/2, h/4$ нам необхідна не тільки в **m_iim**, а і в описах фактичних функцій K, p, f , тому для сумісного доступу до цих даних використаємо опис *global*. При наявності точок розриву в K, p, f – аналогічно вчинимо з індексними масивами і їх розмірностями: TrK, nK ; Trp, np ; Trf, nf .

Аудиторне завдання:

Розглянемо розв’язання деяких модельних задач, в яких продемонструємо переваги ПМ при побудові РС з неперервними і розривними коефіцієнтами.

п.1. Розв’язати модельну крайову задачу (4.29), (4.29a), (4.29b) з неперервними функціями K, p, f :

$$K(x)=1+x, \quad p(x)=\exp(-x), \quad f(x)=\sin(x)*(1+x+\exp(-x))-\cos(x)$$

на $[a, b]=[1, 3]$ для всіх можливих комбінацій КУ I, II, III роду. При таких даних маємо точний розв’язок $u(x)=\sin(x)$. Побудувати графіки: наближеного $\bar{Y}$ і точного розв’язку $u(\bar{X})$; абсолютної похибки $|Y-u(\bar{X})|$. Обчислити значення $\|Y-u(\bar{X})\|_C$ і порівняти його з теоретичними оцінками.

Програмна реалізація для цієї РЗ виконана в функції **m_iim**, а отже створимо відповідну М-функцію *p07a01*, в якій розмістимо підфункції *Kf, pf, ff*:

```
function p07a01
% п.1 Вхідні дані неперервні :
% Модельна задача: при
% K(x)=1+x, p(x)=exp(-x),
% f(x)=sin(x)*(1+x+exp(-x))-cos(x),
% a=1, b=3;
% U(a)=sin(a), U(b)=sin(b);
% розв'язок буде U(x)=sin(x).
clear all, close all, clc
global X n h h05 h25
% Вхідні дані :
a = 1; b = 3;
D=[0,1,sin(a),0,1,sin(b)]; % Ip, Ip
D=[1,0,-(1+a).*cos(a),0,1,sin(b)]; % IIp, Ip
D=[1,2,-(1+a).*cos(a)+2*sin(a),0,1,sin(b)]; % IIIP, Ip
D=[0,1,sin(a),1,0,(1+b).*cos(b)]; % Ip, IIp
D=[1,0,-(1+a).*cos(a),1,0,(1+b).*cos(b)]; % IIp, IIp
D=[1,2,-(1+a).*cos(a)+2*sin(a),1,0,(1+b).*cos(b)]; % IIIP, IIp
D=[0,1,sin(a),1,3,(1+b).*cos(b)+3*sin(b)]; % Ip, IIIP
D=[1,0,-(1+a).*cos(a),1,3,(1+b).*cos(b)+3*sin(b)]; % IIp, IIIP
D=[1,2,-(1+a).*cos(a)+2*sin(a),1,3,(1+b).*cos(b)+3*sin(b)]; % IIIP, IIIP
% Наближений розв'язок :
n = 201; X = linspace(a, b, n);
h = X(2)-X(1); h05 = 0.5*h; h25 = 0.5*h05;
Y = m_iim(@Kf, @pf, @ff, D);
% Точний розв'язок :
u = @(x)(sin(x)); U = u(X);
t = 'K,p,f-неперервні';
view_solu(Y, U, 7, t, 5);
end

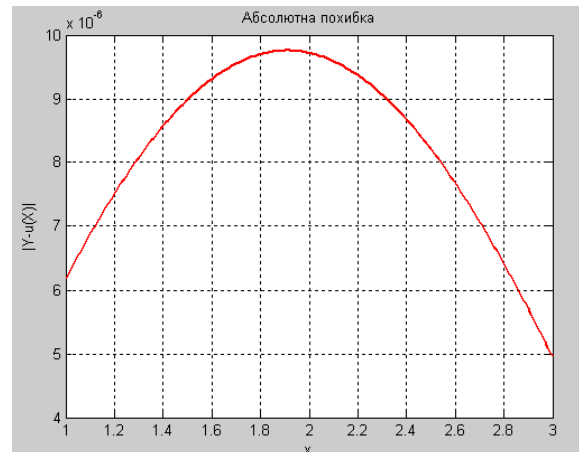
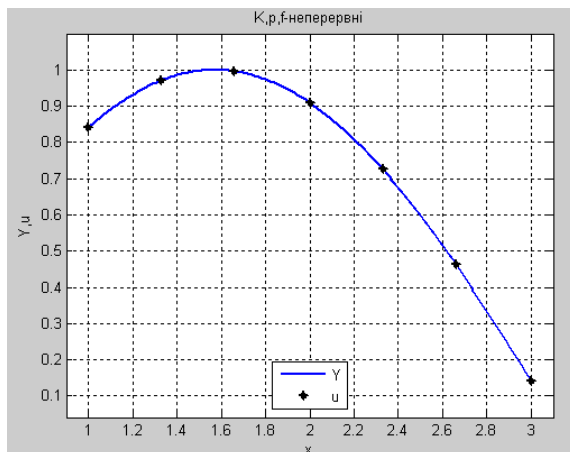
function e = Kf(j)
% Функція K(x)=1+x
global X h05
x = X(j) - h05;
e = 1 + x;
```

end

```
function e = pf(j)
% Функція p(x)=exp(-x)
global X n h25
x = X(j);
if j == 1
    x = x + h25;
elseif j == n
    x = x - h25;
end
e = exp(-x);
end
```

```
function e = ff(j)
% Функція f(x)=sin(x)*(1+x+exp(-x))-cos(x)
global X n h25
x = X(j);
if j == 1
    x = x + h25;
elseif j == n
    x = x - h25;
end
e = sin(x).*(1 + x + exp(-x)) - cos(x);
end
```

і отримаємо наступний результат її виконання (показано тільки результати для КУ III роду):



$h^2 = 1.000e-004$ Похибка в нормі C = $9.75581e-006$

п.2. Розв'язати модельну крайову задачу (4.29), (4.29a), (4.29b) з неперервними функціями K, p і розривною функцією f :

$$K(x)=1, \quad p(x)=0, \quad f(x)=\begin{cases} 100\text{sign}(x)+\exp(x), & x \neq 0; \\ 0, & x=0. \end{cases}$$

на $[a,b]=[-1,1]$ для КУ I роду $u(-1)=0, u(1)=1$. При таких даних маємо точний розв'язок $u(x)=0.5(1-99x)+50x^2\text{sign}(x)+\exp(x)-0.5((1+x)\exp(1)+(1-x)\exp(-1))$. Виведення інформації реалізувати аналогічно **п.1**.

Створимо відповідну М-функцію `p07a02`, в якій розмістимо анонімні функції `Kf`, `pf` і підфункцію `ff`:

```
function p07a02
% п.2 K,p-неперервні, f-розривна :
% Модельна задача: при
% K(x)=1, p(x)=0,
```

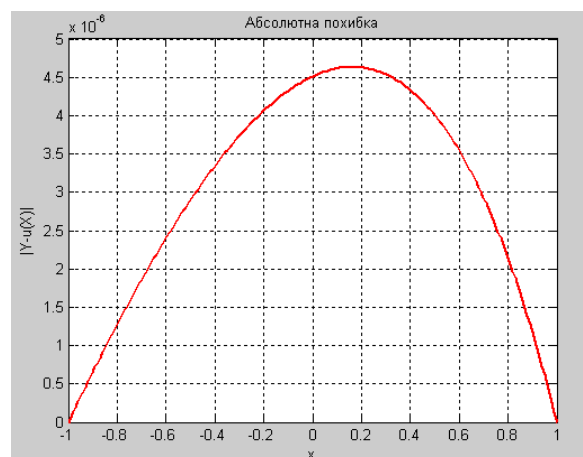
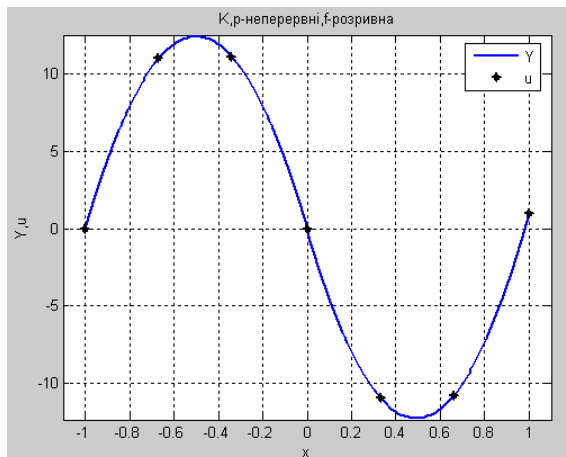
```

% f(x)=-iif(x=0,0,100*sign(x)+exp(-x)),
% a=-1, b=1; U(a)=0, U(b)=1;
% розв'язок буде
% U(x)=0.5*(1-99x)+50x^2*sign(x)+exp(x)
% -0.5*((1+x)*exp(1)+(1-x)*exp(-1))
clear all, close all, clc
global X n h h05 h25 Trf nf
% Вхідні дані :
Kf = @(j)(1);
pf = @(j)(0);
a = -1; b = 1;
D = [0,1,0,0,1,1];
% Наближений розв'язок :
n = 201; X = linspace(a, b, n);
h = X(2)-X(1); h05 = 0.5*h; h25 = 0.5*h05;
% опис розміщення точки розриву f(x) на сітці X :
Trf = [101]; nf = 1;
Y = m_iim(Kf, pf, @ff, D);
% Точний розв'язок :
u = @(x) (0.5*(1-99*x)+50*x.*x.*sign(x)+exp(x)...
-0.5*((1+x).*exp(1)+(1-x).*exp(-1)));
U = u(X);
t = 'К,р-неперервні,f-розривна';
view_solu(Y, U, 7, t, 5);
end

function e = ff(j)
% Функція f(x)=-iif(x=0,0,100*sign(x)+exp(x))
global X n h25 Trf
x = X(j); k = find(j == Trf(:));
if isempty(k)
%обчислення f(x) в неперервній точці по формулам (4.32)
if j == 1
x = x + h25;
elseif j == n
x = x - h25;
end
e = -100*sign(x) - exp(x);
elseif k == 1
%обчислення f(x) в точці 1-го розриву по формулі (4.33)
e = -0.5*(exp(x-h25) + exp(x+h25));
end
end
end

```

і отримаємо наступний результат її виконання



$h^2 = 1.000e-004$ Похибка в нормі $C = 4.63353e-006$

п.3. Розв'язати крайову задачу (4.29), (4.29a), (4.29b) з неперервними функціями K , f і розривною функцією p :

$$K(x)=1, \quad p(x)=\begin{cases} p_1, & a \leq x < x_m, \\ p_2, & x_m < x \leq b. \end{cases} \quad p_1=2, p_2=10, 1 < m < n, \quad f(x)=\sin(x)$$

на $[a, b]=[0, 1]$ для КУ $u(0)=1, u'(1)+3u(1)=4$. Побудувати графік наближеного розв'язку.

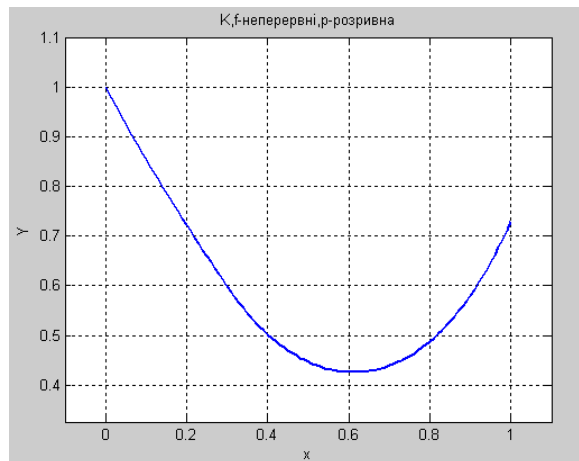
Створимо відповідну М-функцію `p07a03`, в якій розмістимо анонімну функцію `Kf` і підфункції `pf`, `ff` :

```
function p07a03
% п.3 K,f-неперервні, p-розривна :
% Крайова задача :
% u''(x)-p(x)*u(x)=-sin(pi*x), x в (0,1)
% p(x)=p1, 0<=x<c,
%      =p2, c<x<=1; c = X(m), 1<m<n;
% U(0)=1, U'(1)+3*U(1)=4.
clear all, close all, clc
global X n h h05 h25 Trp np p1 p2
% Вхідні дані :
Kf = @(j)(1);
p1 = 2; p2 = 10;
a = 0; b = 1;
D = [0,1,1,1,3,4];
% Наближений розв'язок :
n = 101; X = linspace(a, b, n);
h = X(2)-X(1); h05 = 0.5*h; h25 = 0.5*h05;
% опис розміщення точки розриву p(x) на сітці X :
Trp = [31]; np = 1;
Y = m_iim(Kf, @pf, @ff, D);
t = 'K,f-неперервні,p-розривна';
view_sol(Y, t);
end

function e = pf(j)
% Функція p(x)=p1, 0 <=x < c,
%      =p2, c < x <=1; c = X(m), 1<m<n
global X n h25 Trp p1 p2
x = X(j); k = find(j == Trp(:));
if isempty(k)
%обчислення p(x) в неперервній точці по формулам (4.32)
if j == 1
x = x + h25;
elseif j == n
x = x - h25;
end
if j < Trp(1)
e = p1;
else
e = p2;
end
elseif k == 1
%обчислення p(x) в точці 1-го розриву по формулі (4.33)
e = 0.5*(p1 + p2);
end
end
```

```
function e = ff(j)
% Функція f(x)=sin(pi*x)
global X n h25
x = X(j);
if j == 1
    x = x + h25;
elseif j == n
    x = x - h25;
end
e = sin(pi.*x);
end
```

і отримаємо наступний результат її виконання



п.4. Розв'язати модельну крайову задачу (4.29), (4.29a), (4.29b) з неперервними функціями p , f і розривною K :

$$K(x) = \begin{cases} K_1, & 0 < x < c, \\ K_2, & c < x < 1 \end{cases}, \quad K_1 \neq K_2, \quad c = x_m + 0.5h, \quad 1 < m < n; \quad p(x) = 0, \quad f(x) = 0 \quad (4.34)$$

на $[a, b] = [0, 1]$ для КУ $u(0) = 1, u(1) = 0$. Виведення інформації реалізувати аналогічно

п.1.

Зауваження. Якщо функція $K(x)$ в (4.29) має розрив першого роду в точці $x = c$, тобто

$$[K(c)] = K(c+0) - K(c-0) \neq 0,$$

то поставлена задача буде мати розв'язок, в тому випадку, коли в точці $x = c$ ставляться додаткові умови неперервності $u(x)$ і $W(x)$

$$[u(c)] = 0, \quad [K(c)u'(c)] = 0. \quad (4.35)$$

Отже, якщо крайову задачу (4.29), (4.29a), (4.29b) з параметрами (4.34) доповнити умовами (4.35), то будемо мати точний розв'язок

$$u(x) = \begin{cases} 1 - \alpha * x, & 0 \leq x \leq c, \\ \beta * (1 - x), & c \leq x \leq 1 \end{cases}, \quad \alpha = \frac{1}{\gamma + c * (1 - \gamma)}, \quad \beta = \alpha * \gamma, \quad \gamma = \frac{K_1}{K_2}.$$

Створимо відповідну М-функцію `p07a04`, в якій розмістимо анонімні функції `pf`, `ff` і підфункції `Kf`, `u` :

```
function p07a04
% п.4 p,f-неперервні, K-розрив 1-го роду :
% Крайова залача :
% (K(x)*u'(x))' = 0, x в (0,1)
% K(x)=K1, 0<x<c або =K2, c<x<1; c = X(m)+0.5h, 1<m<n;
% U(0)=1, U(1)=0.
clear all, close all, clc
global X n h h05 h25 TrK nK K1 K2 c al be
% Вхідні дані :
```

```

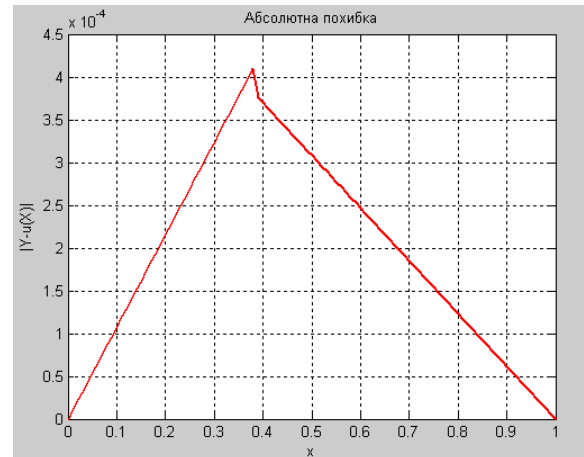
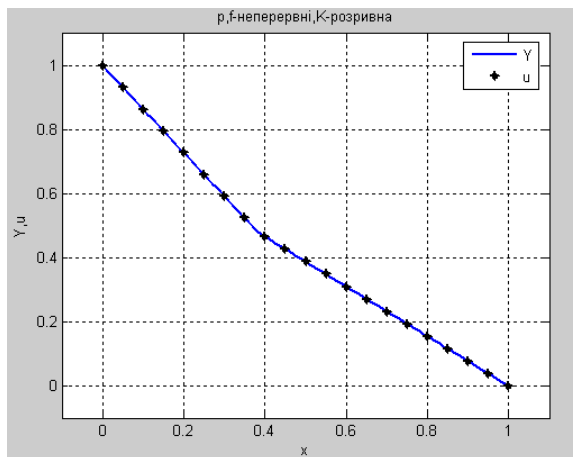
pf = @(j) (0);
ff = @(j) (0);
K1 = 2; K2 = 3.5; a = 0; b = 1; D = [0,1,1,0,1,0];
% Наближений розв'язок :
n = 101; X = linspace(a, b, n);
h = X(2)-X(1); h05 = 0.5*h; h25 = 0.5*h05;
% опис точки розриву K(x) на сітці X :
TrK = [39]; nK = 1; c = X(TrK(1)) + h05; t = K1./K2;
al = 1/(t + (1-t).*c); be = al.*t;
Y = m_iim(@Kf, pf, ff, D);
U = X;
for k = 1:n
    U(k) = u(X(k)); % точний розв'язок
end
t = 'p,f-неперервні,K-розривна';
view_solu(Y, U, 21, t, 5);
end

function e = Kf(j)
% Функція K(x)=K1, 0<x<c,
%                =K2, c<x<1; c = X(m)+h05, 1<m<n
global TrK K1 K2
m = TrK(1)+1;
if j < m
    e = K1;
elseif j == m
    e = 0.5*(K1 + K2);
else
    e = K2;
end
end

function e = u(x)
% Функція u(x)
global c al be
if x <= c
    e = 1 - al.*x;
else
    e = be.*(1 - x);
end
end

```

і отримаємо наступний результат її виконання:



$h^2 = 1.000e-004$ Похибка в нормі $C = 4.09816e-004$

Завдання для самостійної роботи:

п.1. Довести, що достатні умови апроксимації другого порядку виконуються для наступних формул обчислення коефіцієнта A_k .

Таблиця 4.3 Варіанти індивідуального завдання.

k	Формула
1	$A_k = K_{k-0.5}$
2	$A_k = 2K_k K_{k-1} / (K_k + K_{k-1})$
3	$A_k = 0.5(K_k + K_{k-1})$
4	$A_k = \sqrt{K_k K_{k-1}}$

п.2. Знайти наближений розв'язок задачі

$$\begin{cases} \frac{d}{dx}(K(x)u'(x)) - p(x)u(x) = -f(x), \\ K(x) \geq K_0 > 0, \quad p(x) \geq 0, \quad x \in (a, b); \\ u(a) = d_1, \quad u(b) = d_2, \\ [u(c)] = 0, \quad [K(c)u'(c)] = 0, \quad c \in (a, b). \end{cases}$$

з рівномірним кроком $h=0.01$, де функція $K(x)$ – кусково-стала :

$$K(x) = \begin{cases} k + 0.2, & a < x < c, \\ k - 0.5, & c < x < b, \end{cases}$$

а решта параметрів ($k, c, p(x), f(x), a, b, \check{G} = [d_1, d_2]$) означена в Таблиці 4.4.

Таблиця 4.4 Варіанти індивідуального завдання.

k	c	$p(x)$	$f(x)$	a	b	$\check{G}$
1	0.5	$\sin^2(x+g)$	$(200+x)*(g+x^2)$	0	1	[1,d]
2	0.2	$g \cos^2(x)$	$100 \exp(-gx) + 20x^3$	-1	1	[d,1]
3	1.7	$(x+g)^2$	$100x^3 \sin(1+gx)$	1	2	[3,d]
4	-0.3	$g + \sin^2(\sqrt{2-x}-1)$	$200 * g * \exp(x+x^3)$	-1	0	[d,0]
5	-0.5	$g * \cos(x+0.5)$	$(x-x^3)/(1+gx^2)$	-1	1	[d,d]
6	-0.8	$ch(x+g)$	$10(1+gx)/(1-x^3)$	-2	0	[1,1/d]
7	0.7	$th(x^2+g)$	$100 \exp(0.5-gx)$	0	1	[0,d/10]
8	0.6	$g * \arccos(x)$	$100(1+gx) * \sin(1+x^2)$	-1	1	[2,d]
9	0.7	$2g + \sin(x^2)$	$300g * x^2$	0	1	[d,2]
10	1.4	$\exp(g-x^2)$	$70g * tg(5-x^2)$	1	2	[0,d]
11	0.6	$\sin^2(1+gx)$	$50g * \sin(x) + x^2$	-1	1	[d,0]
12	0.8	$1 + gx + x^3$	$300 * \sin^2(x+g)$	0	2	[-3,d]
13	0.6	$1 + th(gx + x^3)$	$300 \sin(x+g)$	0	1	[d,-3]
14	-0.4	$g * ch(x)$	$50(g-x+x^3)$	-1	1	[0,-d]

15	-1.4	$th(1+g \cdot x+x^2)$	$50g \cdot x^3$	-2	-1	[1,-d]
16	-0.6	$g+x^2$	$(1+200g \cdot x) \cdot \exp(x^3)$	-1	0	[2,-d]
17	0.5	$g+\cos(x)$	$300(1+gx^2)$	0	1	[-d,d]
18	0.1	$(x^2+1)/(x^4+g)$	$100(g+x^3)$	-1	1	[1-d,0]
19	1.6	$(x+g)/(x^2+g)$	$x+300 \cdot \sin(gx)$	1	2	[2-d,0]
20	1.8	$(x \cdot g)/(x^2+g)$	$gx+50 \cdot th(x)$	1	3	[0,d-1]
21	-0.6	$\sin^2(x/g)$	$2000(1+gx)/(1-x^3)$	-1	0	[d-1,0]
22	0.7	$\cos^2(g \cdot x)$	$300g(1+x)x$	0	1	[0,d-1]
23	0.4	$g+x^2$	$1+50g \cdot th(x)$	-1	1	[5-d,0]
24	0.3	$ch(x+g)$	$300g \cdot x(1+\sin(x))$	-1	1	[1-d,d]
25	1.1	$\sin^2(g+x^2)$	$20(g+x+\sqrt{x^3})$	0	2	[1,2-d]
26	-1.1	$g \cdot x^2$	$20(1+g\sqrt{x^2-x})$	-2	0	[2-d,0]
27	0.7	$ch(g-x^2)$	$300(1+gx^2)$	0	1	[1,1-d]
28	1.2	$x+g$	$200(1+x)/(1+gx+x^3)$	0	2	[0,2-d]
29	-0.2	x^4+g	$50 \cdot ch(g+x^3)$	-1	1	[d-1,2]
30	0.2	$g \cdot \cos(x)$	$200(1+x)/(1+g \cdot x^2)$	-1	1	[d-1,1-d]

Програмне забезпечення на MATLAB:

```

function Y = m_iim(Kfun, pfun, ffun, D)
% Наближений розв'язок крайової задачі
% (K(x)*u'(x))' - p(x)*u(x) = -f(x),
% K(x)>=K0>0, p(x)>=0, x є (a,b),
% K(x), p(x), f(x) можуть бути розривні
% (-D(1)K(x)*u'(x)+D(2)u(x))|x=a = D(3),
% (D(4)K(x)*u'(x)+D(5)u(x))|x=b = D(6).
% РС побудована інтегро-інтерполяційним методом.
% Вхідні дані :
% Kfun - функція з описом K(X(k));
% pfun - функція з описом p(X(k));
% ffun - функція з описом f(X(k));
% D - масив значень коефіцієнтів КУ:
% D(1),D(4)=0 & D(2),D(5)=1 - КУ I роду,
% D(1),D(4)=1 & D(2),D(5)>=0 - КУ II,III роду.
% Вихідні дані:
% Y - масив значень наближеного розв'язку.
if nargin < 4
    error('M_IIM:NotEnoughInputs',...
        'Not enough input arguments. See M_IIM. ');
    return
end
if length(D) < 6
    error('M_IIM:NotEnoughInputs',...
        'Not enough input arguments D. See M_IIM. ');

```

```

        return
    end
    if D(1) ~= 0 & D(1) ~= 1
        error('M_IIM:NotEnoughInputs',...
            'Not enough input arguments D(1). See M_IIM.');
```

```

        return
    end
    if D(4) ~= 0 & D(4) ~= 1
        error('M_IIM:NotEnoughInputs',...
            'Not enough input arguments D(4). See M_IIM.');
```

```

        return
    end
    global X n h h05
    hh = h.*h; % допоміжні змінні
    % формування коефіцієнтів
    % A(k), C(k), B(k), F(k), k=1,...,n для СЛАР
    A = zeros(1, n); B = A; C = A; F = A;
    akp = Kfun(2);
    if D(1)
        C(1) = akp + h.*(D(2) + h05.*pfun(1));
        B(1) = akp; F(1) = h.*(D(3) + h05.*ffun(1));
    else
        C(1) = 1; F(1) = D(3);
    end
    for k = 2 : n-1
        ak = akp ; akp = Kfun(k+1);
        A(k) = ak; B(k) = akp;
        C(k) = ak + akp + hh.*pfun(k);
        F(k) = hh.*ffun(k);
    end
    if D(4)
        C(n) = akp + h.*(D(5) + h05.*pfun(n));
        A(n) = akp; F(n) = h.*(D(6) + h05.*ffun(n));
    else
        C(n) = 1; F(n) = D(6);
    end
    % знах-ня Y з розв'язку СЛАР м.монотонної (правої) прогонки
    [Y AL er] = m_progmm(A, C, B, F);
    return

```

function view_solu(Y, U, ni, txt, s)

```

global X n h
m = U(1); M = m;
for i = 2:n
    de = U(i);
    if de < m
        m = de;
    elseif de > M
        M = de;
    end
end
i = fix(linspace(1, n, ni));
de = 1e-1; de = [X(1)-de, X(n)+de, m-de, M+de];
plot(X,Y,'b',X(i),U(i),'k*','LineWidth',2);
axis(de); grid on; title(txt);
xlabel('x'), ylabel('Y,u');
legend('Y','u','Location','Best');
if s, pause(s)

```

```

else, pause
end
close all
U = abs(Y-U);
fprintf('h^2=%11.3e Похибка в нормі C=%13.5e\n',h.*h,max(U));
plot(X,U,'r','LineWidth',1.5); grid on;
title('Абсолютна похибка');
xlabel('x'), ylabel('|Y-u(X)|');
end

```

```

function view_sol(Y, txt)
global X n
m = Y(1); M = m;
for i = 2:n
    de = Y(i);
    if de < m
        m = de;
    elseif de > M
        M = de;
    end
end
de = 1e-1; de = [X(1)-de, X(n)+de, m-de, M+de];
plot(X,Y,'b','LineWidth',2);
axis(de); grid on; title(txt);
xlabel('x'), ylabel('Y');
end

```

Практичне заняття № 08

ТЕМА № 5. Метод прямих (Роте) розв'язання крайових задач для нестационарних рівнянь і систем рівнянь різних типів (2 год).

Теоретична інформація:

Метод прямих (Роте) належить до групи наближених методів розв'язання крайових задач, основна ідея яких пов'язана з пониженням розмірності поставленої задачі [7, с. 93–94]. В цьому методі розмірність понижують за рахунок апроксимації тільки певної частини диференціальних операторів, що приводить до системи звичайних диференціальних рівнянь. Такий підхід досить часто застосовують до систем, що складаються з рівнянь математичної фізики різних типів, особливо у випадку квазілінійних та нелінійних рівнянь. Узагальнення цього методу – *метод інтегральних співвідношень*.

Розглянемо схему застосування м.Роте на прикладі наступної мішаної задачі. Нехай маємо систему квазілінійних рівнянь

$$\begin{cases} \frac{\partial u(x,t)}{\partial t} = f_1(x,t,u(x,t),v(x,t)), & (x,t) \in (a,b) \times (0,T], \\ \frac{\partial v(x,t)}{\partial t} = c \frac{\partial^2 v(x,t)}{\partial x^2} + f_2(x,t,u(x,t),v(x,t)), & c > 0, \end{cases} \quad (5.1)$$

для якої виконуються початкові умови

$$u(x,0) = u_0(x), \quad v(x,0) = v_0(x), \quad x \in [a,b] \quad (5.2)$$

а також крайові умови (КУ) в граничних точках a, b , в нашому випадку для параболічного рівняння, а в загальному – для тих функції, де є частинні похідні за змінною x . Звичайно, крайові умови в точках границі a, b сукупно можуть бути певною комбінацією КУ I, II, III роду.

Враховуючи, що вид КУ впливає на технічну сторону реалізації м.Роте, то будемо розглядати основні комбінації КУ. Отримані знання і досвід при реалізації цих варіантів допоможе легко впоритись з іншими комбінаціям КУ.

Основними комбінаціями КУ, з огляду варіантів розрахункових формул при реалізації м.Роте, є наступні пари:

1 {I (a), I (b)}; **2a** {III (a); I (b)}; **2b** {I (a); III (b)}; **3** {III (a); III (b)}.

Враховуючи, що пари **2a, 2b** подібні в технології реалізації, то зосередимося на трьох комбінаціях КУ на різних границях :

1. в лівій і правій граничній точці задано КУ I роду

$$v(a,t) = \mu_1(t), \quad v(b,t) = \mu_2(t), \quad t \in [0,T], \quad (5.3.1)$$

2. в точці $x=a$ задано КУ III роду, а в правій граничній точці задано КУ I роду (при $g_1(t) \equiv 0$ будемо мати в точці $x=a$ КУ II роду)

$$\frac{\partial v(a,t)}{\partial x} + g_1(t) * v(a,t) = \mu_1(t), \quad v(b,t) = \mu_2(t), \quad t \in [0,T], \quad (5.3.2)$$

3. в обох граничних точках задано КУ III роду

$$\frac{\partial v(a,t)}{\partial x} + g_1(t) * v(a,t) = \mu_1(t), \quad \frac{\partial v(b,t)}{\partial x} + g_2(t) * v(b,t) = \mu_2(t), \quad t \in [0,T], \quad (5.3.3)$$

При розв'язанні задачі (5.1)–(5.3) вважатимемо, що:

- початкові і крайові дані мають достатню гладкість і для них виконуються необхідні умови узгодження в точках $(a;0), (b;0)$;
- функції f_1, f_2 неперервно диференційовані по всім змінним;
- шукані функції u, v також достатньо гладкі;
- виконано дослідження КЗ щодо коректності, а також розв'язок існує і єдиний.

Для нашої задачі зручно застосувати м.Роте за змінною x . Уведемо рівномірну сітку

$$\bar{\omega}_h = \{x_k = (k-1)h, \quad k=1,2,\dots,n, \quad h=1/(n-1)\} = \bar{X},$$

і позначимо на прямих $x = x_k$ невідомі функції

$$u(x_k, t) = U_k(t), \quad v(x_k, t) = V_k(t).$$

Виконаємо в (5.1), (5.3.2), (5.3.3) різницеву апроксимацію порядку $O(h^2)$ всіх диференціальних операторів, що містять x і для цього будемо використовувати три-точковий шаблон $S\{x_{k-1}, x_k, x_{k+1}\}$ – у результаті отримаємо деяку систему рівнянь. Для кожного варіанту **1–3** комбінації КУ на різних границях, запишемо окремі системи.

Система для варіанту **1** – мішана задача (5.1), (5.2), (5.3.1) :

$$\left\{ \begin{array}{l} \frac{dU_1(t)}{dt} = f_1(x_1, t, U_1(t), \mu_1(t)), \quad k=1, \\ \frac{dU_k(t)}{dt} = f_1(x_k, t, U_k(t), V_k(t)), \quad k=\overline{2, n-1}, \\ \frac{dU_n(t)}{dt} = f_1(x_n, t, U_n(t), \mu_2(t)), \quad k=n, \\ V_1(t) = \mu_1(t), \quad k=1, \\ \frac{dV_2(t)}{dt} = c \frac{\mu_1(t) - 2V_2(t) + V_3(t)}{h^2} + f_2(x_2, t, U_2(t), V_2(t)), \quad k=2, \\ \frac{dV_k(t)}{dt} = c \frac{V_{k-1}(t) - 2V_k(t) + V_{k+1}(t)}{h^2} + f_2(x_k, t, U_k(t), V_k(t)), \quad k=\overline{3, n-2}, \\ \frac{dV_{n-1}(t)}{dt} = c \frac{V_{n-2}(t) - V_{n-1}(t) + \mu_2(t)}{h^2} + f_2(x_{n-1}, t, U_{n-1}(t), V_{n-1}(t)), \quad k=n-1, \\ V_n(t) = \mu_2(t), \quad k=n. \end{array} \right. \quad (5.4.1)$$

В системі (5.4.1), для рівнянь $\frac{dU_1(t)}{dt}, \frac{dU_n(t)}{dt}, \frac{dV_2(t)}{dt}, \frac{dV_{n-1}(t)}{dt}$, в які входять відомі з КУ (5.3.1) функції $V_1(t), V_n(t)$, ми використали їх означення $\mu_1(t), \mu_2(t)$. Самі крайові умови I роду ми також залишили в системі (5.4.1). Це пов'язано з тим, що функції $\{(U_k(t))_{k=1}^n, (U_k(t))_{k=2}^{n-1}\}$ ми будемо знаходити як розв'язок деякої задачі Коші і запам'ятовувати в точках $\bar{T} = \{0 = t_1 < t_2 < \dots < t_{m-1} < t_m = T\}$ – сітки виведення результату по змінній t . Після цього обчислимо $\{V_1(t), V_n(t)\}_{t \in \bar{T}} = \{\mu_1(t), \mu_n(t)\}_{t \in \bar{T}}$ і будемо мати повну інформацію про наближений розв'язок $\{u(x_k, t_j), v(x_k, t_j)\}$, $x_k \in \bar{X}$, $t_j \in \bar{T}$.

Система для варіанту **2** – мішана задача (5.1), (5.2), (5.3.2) :

$$\left\{ \begin{array}{l} \frac{dU_k(t)}{dt} = f_1(x_k, t, U_k(t), V_k(t)), \quad k=\overline{1, n-1}, \\ \frac{dU_n(t)}{dt} = f_1(x_n, t, U_n(t), \mu_2(t)), \quad k=n, \\ \frac{dV_1(t)}{dt} = 2c \frac{V_2(t) + (hg_1(t) - 1)V_1(t) - h\mu_1(t)}{h^2} + f_2(x_1, t, U_1(t), V_1(t)), \quad k=1, \\ \frac{dV_k(t)}{dt} = c \frac{V_{k-1}(t) - 2V_k(t) + V_{k+1}(t)}{h^2} + f_2(x_k, t, U_k(t), V_k(t)), \quad k=\overline{2, n-2}, \\ \frac{dV_{n-1}(t)}{dt} = c \frac{V_{n-2}(t) - V_{n-1}(t) + \mu_2(t)}{h^2} + f_2(x_{n-1}, t, U_{n-1}(t), V_{n-1}(t)), \quad k=n-1, \\ V_n(t) = \mu_2(t), \quad k=n. \end{array} \right. \quad (5.4.2)$$

В системі (5.4.2), для КУ I роду в точці $x=b$ ми поступили аналогічно, як і в системі (5.4.1). Щодо рівняння для $\frac{dV_1(t)}{dt}$, то тут ми скористались *методом законтурної точки* для отримання апроксимації КУ III роду в точці $x_1=a$ з порядком $O(h^2)$, а саме, з системи двох рівнянь :

$$\begin{cases} \frac{V_2(t)-V_0(t)}{2h} + g_1(t)V_1(t) = \mu_1(t), \\ \frac{dV_1(t)}{dt} = c \frac{V_0(t) - 2V_1(t) + V_2(t)}{h^2} + f_2(x_1, t, U_1(t), V_1(t)), \quad k=1. \end{cases}$$

виключаємо $V_0(t)$. Всі інші моменти, які ми розглянули для системи (5.4.1), модифікуються з урахуванням реалій системи (5.4.2).

Система для варіанту 3 – мішана задача (5.1), (5.2), (5.3.3) :

$$\begin{cases} \frac{dU_k(t)}{dt} = f_1(x_k, t, U_k(t), V_k(t)), \quad k=\overline{1, n}, \\ \frac{dV_1(t)}{dt} = 2c \frac{V_2(t) + (hg_1(t) - 1)V_1(t) - h\mu_1(t)}{h^2} + f_2(x_1, t, U_1(t), V_1(t)), \quad k=1, \\ \frac{dV_k(t)}{dt} = c \frac{V_{k-1}(t) - 2V_k(t) + V_{k+1}(t)}{h^2} + f_2(x_k, t, U_k(t), V_k(t)), \quad k=\overline{2, n-1}, \\ \frac{dV_n(t)}{dt} = 2c \frac{V_{n-1}(t) - (1 + hg_2(t))V_n(t) + h\mu_2(t)}{h^2} + f_2(x_n, t, U_n(t), V_n(t)), \quad k=n. \end{cases} \quad (5.4.3)$$

Останнє рівняння (5.4.3) також отримане з урахуванням апроксимації КУ III роду в точці $x_n=b$ з порядком $O(h^2)$ методом законтурної точки. Для цього із системи двох рівнянь

$$\begin{cases} \frac{V_{n+1}(t) - V_{n-1}(t)}{2h} + g_2(t)V_n(t) = \mu_2(t), \\ \frac{dV_n(t)}{dt} = c \frac{V_{n-1}(t) - 2V_n(t) + V_{n+1}(t)}{h^2} + f_2(x_n, t, U_n(t), V_n(t)), \quad k=n. \end{cases}$$

ми виключили $V_{n+1}(t)$. Зауважимо, що система рівнянь (5.4.3), на відміну від систем (5.4.1)–(5.4.2) не містить додаткових операцій, пов'язаних з «довизначенням» на сітці $\bar{T}$ деяких компонент наближеного розв'язку задачі (5.1), (5.2), (5.3.3).

Відмітимо також, що отримані системи (5.4.1)–(5.4.3) і їм подібні для варіантів комбінацій КУ на різних границях, які ми не розглядали, складаються з декількох груп рівнянь :

А) Рівняння, записані в граничній точці і пов'язують невідомі функції, означені тільки в цій точці. Ця група буде присутня у випадку КУ I роду в даній граничній точці і може складатися як з дифрівняння відносно невідомої функції, так і з відомої КУ з постановки задачі.

Б) Рівняння, записані в граничній точці і пов'язують невідомі функції, означені в граничній та приграничних точках. Ця група буде присутня у випадку КУ II, III роду в даній граничній точці і завжди є дифрівнянням відносно невідомої функції.

В) Рівняння, записані в приграничній точці, яка безпосередньо межує з граничною точкою (іншими словами рівнянь – записаних на шаблоні $S\{x_1=a, x_2, x_3\}$ або на $S\{x_{n-2}, x_{n-1}, x_n=b\}$). Ця група буде присутня при КУ I роду в даній граничній точці і наявності диференціального оператора другого порядку по змінній x .

Г) рівняння, які записані тільки у внутрішніх точках сітки $\bar{X}$ (на шаблоні $S\{x_{k-1}, x_k, x_{k+1}\}$).

Для систем рівнянь (5.4) потрібно додати початкові умови (5.3) у вигляді

$$U_k(0)=u_0(x_k), \quad V_k(0)=v_0(x_k), \quad k=\overline{1,n}. \quad (5.5)$$

Якщо в системах рівнянь (5.4) перенумерувати тільки невідомі функції і ввести для них єдине позначення $\{Y_k(t)\}_{k=1}^N$, а саме

для системи (5.4.1) $N=2n-2$

$$\begin{array}{c|c|c|c|c|c|c|c} U_1(t) & \cdots & U_n(t) & V_1(t) & V_2(t) & \cdots & V_{n-1}(t) & V_n(t) \\ \hline Y_1(t) & \cdots & Y_n(t) & \sim & Y_{n+1}(t) & \cdots & Y_N(t) & \sim \end{array}$$

$$\vec{Y0}=(U_1(0),\dots,U_n(0),V_2(0),\dots,V_{n-1}(0))^T$$

для системи (5.4.2) $N=2n-1$

$$\begin{array}{c|c|c|c|c|c|c|c} U_1(t) & \cdots & U_n(t) & V_1(t) & V_2(t) & \cdots & V_{n-1}(t) & V_n(t) \\ \hline Y_1(t) & \cdots & Y_n(t) & Y_{n+1}(t) & Y_{n+2}(t) & \cdots & Y_N(t) & \sim \end{array}$$

$$\vec{Y0}=(U_1(0),\dots,U_n(0),V_1(0),\dots,V_{n-1}(0))^T$$

для системи (5.4.3) $N=2n$

$$\begin{array}{c|c|c|c|c|c|c|c} U_1(t) & \cdots & U_n(t) & V_1(t) & V_2(t) & \cdots & V_{n-1}(t) & V_n(t) \\ \hline Y_1(t) & \cdots & Y_n(t) & Y_{n+1}(t) & Y_{n+2}(t) & \cdots & Y_{N-1}(t) & Y_N(t) \end{array}$$

$$\vec{Y0}=(U_1(0),\dots,U_n(0),V_1(0),\dots,V_n(0))^T$$

то для вектор-функції $\vec{Y}(t)=(Y_1(t),Y_2(t),\dots,Y_N(t))^T$, отримаємо задачу Коші для системи звичайних диференціальних рівнянь

$$\begin{cases} \frac{d\vec{Y}(t)}{dt} = \vec{F}(t, \vec{Y}(t)), \\ \vec{Y}(0) = \vec{Y0}. \end{cases} \quad (5.6)$$

Для числового розв'язку (5.6) можна застосувати *ode*-солвери СКМ MATLAB: **ode45** (задача не жорстка) або **ode23s** (задача жорстка). Після знаходження розв'язку $\vec{Y}(t)$, а за наявності КУ I роду виконання дій по довизначенню відомих (граничних) функцій в точках $t=\bar{T}$, будемо мати повну інформацію про наближений розв'язок $\{u(x_k, t_j), v(x_k, t_j)\}$, $x_k \in \bar{X}$, $t_j \in \bar{T}$.

Розглянемо ще етап трансформації математичного опису вектор-функції $\vec{F}(t, \vec{Y}(t))$ в програмний опис відповідної М-функції. Згідно правила використання *ode*-солверів це має бути М-функція (оберемо для неї ім'я, наприклад *namef*) з таким заголовком

$$function F = namef(t, y)$$

де F, y – вектор-стовпці, а t – скалярна змінна. Тобто, обчислення F – правої частини системи ЗДР має залежати тільки від двох формальних параметрів – t, y . Наведемо приклад деталізації $\vec{F}(t, \vec{Y}(t)) \Rightarrow namef(t, y)$ для системи (5.4.1). Уведемо додаткові константи $n1=n-1$, $np=n+1$, $n2=n+2$, $d=c*h^{-2}$, тоді маємо наступний перелік "операторів" для запису тіла цієї М-функції

$$\begin{aligned} z1 &= \mu_1(t); \quad F_1 = f_1(x_1, t, y_1, z1); \\ F_k &= f_1(x_k, t, y_k, y_{k+n1}), \quad k=\overline{2, n1}; \\ z2 &= \mu_2(t); \quad F_n = f_1(x_n, t, y_n, z2); \\ F_{np} &= d * (z1 - 2 * y_{np} + y_{n2}) + f_2(x_2, t, y_2, y_{np}); \\ F_j &= d * (y_{j-1} - 2 * y_j + y_{j+1}) + f_2(x_k, t, y_k, y_j), \quad k=\overline{3, n1-1}, \quad j=k+n1; \\ F_N &= d * (y_{N-1} - 2 * y_N + z2) + f_2(x_{n1}, t, y_{n1}, y_N); \end{aligned} \quad (5.7)$$

який потрібно доповнити оператором передачі необхідних глобальних значень

$$global \quad n \quad n1 \quad np \quad n2 \quad N \quad X \quad d$$

(X – вектор задання рівномірної сітки $\bar{\omega}_h$).

Зауваження

1. За наявності у крайових умовах похідних рекомендуємо для отримання порядку апроксимації $O(h^2)$ використовувати *метод законтурних точок*.
2. Перехід від позначень невідомих функцій в (5.4), (5.5) до позначень векторів в (5.6) і їх використання в (5.7) необхідно виписувати поступово та дуже уважно. При цьому ретельно перевіряти відповідність за індексами «старих» і «нових» позначень.

Аудиторне завдання:

п.1. Розв'язати модельну крайову задачу

$$\begin{cases} u'_t(x,t) = u^2 + (v-x)*v - 1, \\ v'_t(x,t) = v''_x(x,t) + x*u + t^2*v, \\ (x,t) \in (0,1) \times (0,2]; \\ u(x,0) = 1, \quad v(x,0) = 0, \quad x \in [0,1]; \end{cases} \quad (5.8)$$

для крайових умов

$$v(0,t) = 0, \quad v(1,t) = \sin(t), \quad t \in [0,2]. \quad (5.9)$$

Точний розв'язок (5.8), (5.9) $u(x,t) = \cos(x*t)$, $v(x,t) = \sin(x*t)$. Побудувати графіки: наближеного $(Y_k(\bar{T}))_{k=1}^n$, $(Y_k(\bar{T}))_{k=n+1}^{2n}$; точного розв'язку $u(\bar{X}, \bar{T})$, $v(\bar{X}, \bar{T})$; похибок $(Y_k(\bar{T}))_{k=1}^n - u(\bar{X}, \bar{T})$, $(Y_k(\bar{T}))_{k=n+1}^{2n} - v(\bar{X}, \bar{T})$. Обчислити значення похибок в $\|\circ\|_{Ch}$.

Створимо відповідну М-функцію *p08a01* :

```
function p08a01
clear all, close all, clc, warning off
global n n1 np n2 N d X
% параметри крайової задачі :
c = 1; a = 0; b = 1; T = 2;
% завдання просторової сітки :
n = 51; X = linspace(a, b, n);
% допоміжні величини :
n1 = n-1; np = n+1; n2 = n+2; N = 2*n1;
h = X(2)-X(1); d = c./(h.*h);
% завдання сітки отримання розв'язку :
M = 51; Tt = linspace(0, T, M);
u0 = @(x)(1); % функція початкових умов u0(x)
v0 = @(x)(0); % функція початкових умов v0(x)
u = @(x,t)(cos(x.*t)); % точний u
v = @(x,t)(sin(x.*t)); % точний v
% U,V,Uu,Vv - масиви для побудови графіка:
U = zeros(M,n); V = U; Uu = U; Vv = V;
for i = 1:n
    x = X(i); U(1,i) = u0(x); V(1,i) = v0(x);
    for j = 1:M
        t = Tt(j); Uu(j,i) = u(x,t); Vv(j,i) = v(x,t);
    end
end
% вектор початкових умов для задачі Коші:
Y0 = [U(1,:), V(1,2:n1)]';
for m = 2:M
    m1 = m-1; tn = Tt(m1); tk = Tt(m);
    % розв'язання задачі на відрізьку t ∈ [Tt(m-1), Tt(m)]:
    [te, Y] = ode23s(@namef, [tn tk], Y0);
    Y0 = Y(length(te,:),:); % розв'язок в t=te
    U(m,:) = Y0(1:n); % зберігаємо U-частину набл.роз-ку
```

```

        % "доповнюємо" відомими значеннями V з КУ I роду
        V(m,:) = [mj1(tk),Y0(nr:N),mj2(tk)];
    end
    [Mx, Mt] = meshgrid(X, Tt); % побудова сіток для 3D-графіків
    % виведення графіків: розв'язку U і точного u(x,t)
    de = [a, b, 0, T, -0.5, 1.1];
    mesh(Mx, Mt, U); axis(de);
    shading interp, colormap cool%, view(60,30)
    xlabel('x'), ylabel('t'), zlabel('U')
    title('U - наближений');
    pause(), close all
    mesh(Mx, Mt, Uu); axis(de);
    shading interp; colormap cool; %, view(60,30)
    xlabel('x'); ylabel('t'); zlabel('u');
    title('u(x,t) - точний');
    pause(), close all
    % виведення графіків: розв'язку V і точного v(x,t)
    mesh(Mx, Mt, V); axis(de);
    shading interp, colormap jet %, view(80,20)
    xlabel('x'), ylabel('t'), zlabel('V');
    title('V - наближений');
    pause(), close all
    mesh(Mx, Mt, Vv); axis(de);
    shading interp, colormap jet %, view(80,20)
    xlabel('x'), ylabel('t'), zlabel('v');
    title('v(x,t) - точний');
    pause(), close all
    % виведення графіків похибки U-Uu, V-Vv
    Uu = U-Uu; C = norm(Uu,inf);
    fprintf('Похибка U-u =%13.5e\n',C);
    mesh(Mx, Mt, Uu);
    shading interp; colormap cool; %, view(60,30)
    xlabel('x'); ylabel('t'); zlabel('u');
    title('Похибка U-u(x,t)');
    pause(), close all
    Vv = V-Vv; C = norm(Vv,inf);
    fprintf('Похибка V-v =%13.5e\n',C);
    mesh(Mx, Mt, Vv);
    shading interp, colormap jet %, view(80,20)
    xlabel('x'), ylabel('t'), zlabel('v');
    title('Похибка V-v(x,t)');
    pause(), close all
end

function s = f1(x,t,u,v)
    % функція f1
    s = u.*u+v.*(v-x)-1;
end

function s = f2(x,t,u,v)
    % функція f2
    s = t.*t; s = x.*u+s.*v;
end

function s = mj1(t)
    % функція mj1
    s = 0;
end

```

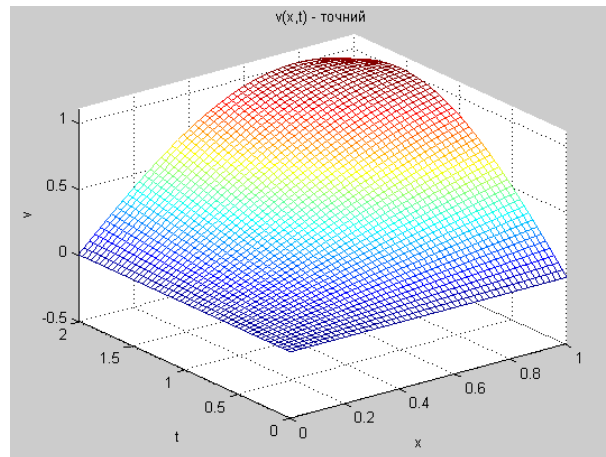
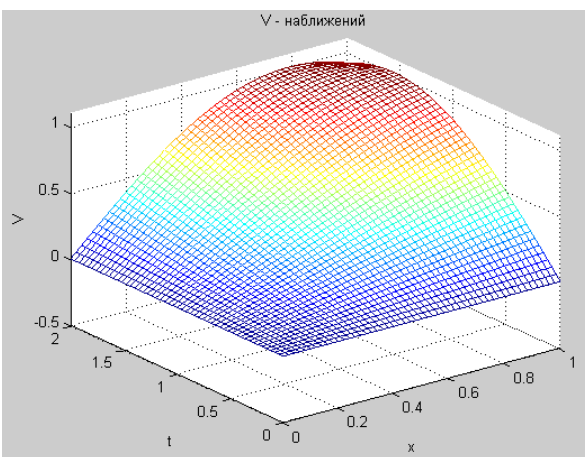
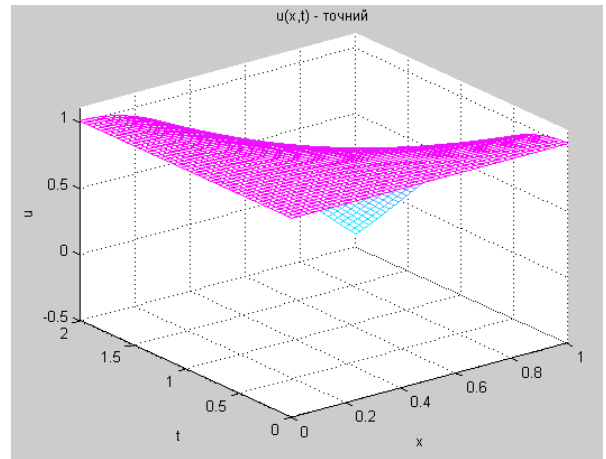
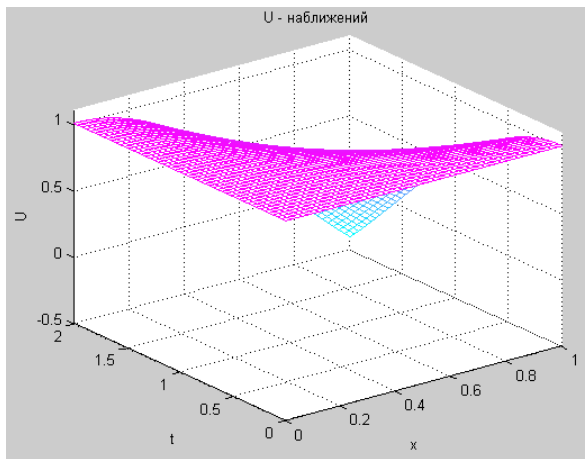
```

function s = mj2(t)
    % функція mj2
    s = sin(t);
end

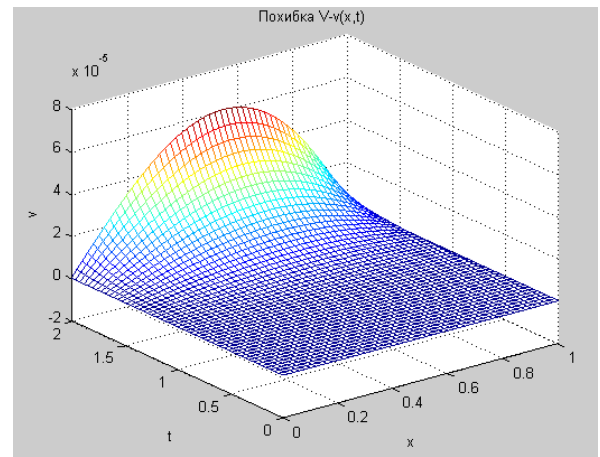
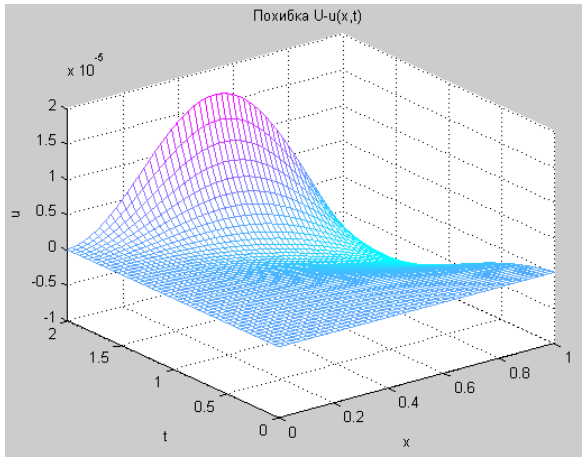
function F = namef(t, y)
    % функція правої частини системи
    global n n1 np n2 N d X
    F = zeros(N, 1);
    z1 = mj1(t); F(1) = f1(X(1), t, y(1), z1);
    for k = 2 : n1
        F(k) = f1(X(k), t, y(k), y(k+n1));
    end
    z2 = mj2(t); F(n) = f1(X(n), t, y(n), z2);
    F(np) = d.*(z1-2.*y(np)+y(n2))+f2(X(2), t, y(2), y(np));
    for k = 3 : n1-1
        j = k+n1;
        F(j) = d.*(y(j-1)-2.*y(j)+y(j+1))+f2(X(k), t, y(k), y(j));
    end
    F(N) = d.*(y(N-1)-2.*y(N)+z2)+f2(X(n1), t, y(n1), y(N));
end

```

і отримаємо наступний результат її виконання



Помилка $U-u = 4.24207e-004$
 Помилка $V-v = 1.95321e-003$



Враховуючи, що апроксимація операторів за змінною x виконана з $O(h^2)$ і розрахунки проведено з $h=0.02$, то отримано досить гарний наближений розв'язок.

п.2. Розв'язати крайову задачу (5.1), (5.2), (5.3.3)

$$\begin{cases} \frac{\partial u(x,t)}{\partial t} = x * v - t^2 * u^2, \\ \frac{\partial v(x,t)}{\partial t} = \frac{\partial^2 v(x,t)}{\partial x^2} + \cos(2t * u) + 2t * v, & (x,t) \in (0,1) \times (0,2]; \\ u(x,0)=0, \quad v(x,0)=x(1-x), \quad x \in [0,1]; \\ \frac{\partial v(0,t)}{\partial x} - 10^{-3}t * v(0,t)=1, \quad \frac{\partial v(1,t)}{\partial x} + 0.05t * v(1,t)=-1, \quad t \in [0,2]. \end{cases}$$

Побудувати графіки наближеного $(Y_k(\bar{T}))_{k=1}^n, (Y_k(\bar{T}))_{k=n+1}^{2*n}$ розв'язку.

Створимо відповідну М-функцію *p08a02* :

```
function p08a02
clear all, close all, clc, warning off
global n n1 np n2 N d d2 h X
% параметри крайової задачі :
c = 1; a = 0; b = 1; T = 2;
% завдання просторової сітки :
n = 51; X = linspace(a, b, n);
% допоміжні величини :
n1 = n-1; np = n+1; n2 = n+2; N = 2*n;
h = X(2)-X(1); d = c./(h.*h); d2 = 2.*d;
% завдання сітки отримання розв'язку :
M = 51; Tt = linspace(0, T, M);
u0 = @(x)(0); % функція початкових умов u0(x)
v0 = @(x)((x-a).*(b-x)); % функція початкових умов v0(x)
% U,V - масиви для побудови графіка:
U = zeros(M,n); V = U;
% вектор початкових умов для задачі Коші:
for i = 1:n
    x = X(i); U(1,i) = u0(x); V(1,i) = v0(x);
end
Y0 = [U(1,:), V(1,:)]';
for m = 2 : M
    m1 = m-1; tn = Tt(m1); tk = Tt(m);
    % розв'язання задачі на відрітку t ∈ [Tt(m-1), Tt(m)]:
    [te, Y] = ode23s(@nmf, [tn tk], Y0);
    Y0 = Y(length(te),:); % розв'язок в t=te
    U(m,:) = Y0(1:n); % зберігаємо U-частину набл.роз-ку
    V(m,:) = Y0(np:N); % зберігаємо V-частину набл.роз-ку
end
```

```

[Mx, Mt] = meshgrid(X, Tt); % побудова сіток для 3D-графіків
% виведення графіків: розв'язку U
mesh(Mx, Mt, U);
shading interp; colormap cool; view(15,15)
xlabel('x'); ylabel('t'); zlabel('U')
title('U - наближений');
pause(); close all
% виведення графіків: розв'язку V
mesh(Mx, Mt, V);
shading interp; colormap jet; view(-38,13)
xlabel('x'); ylabel('t'); zlabel('V');
title('V - наближений');
pause(); close all
end

function s = f1(x,t,u,v)
    s = t.*u;
    s = x.*v-s.*s;
end

function s = f2(x,t,u,v)
    s = 2.*t; s = cos(s.*u)+s.*v;
end

function s = g1(t)
    s = -1e-3.*t;
end

function s = mj1(t)
    s = 1;
end

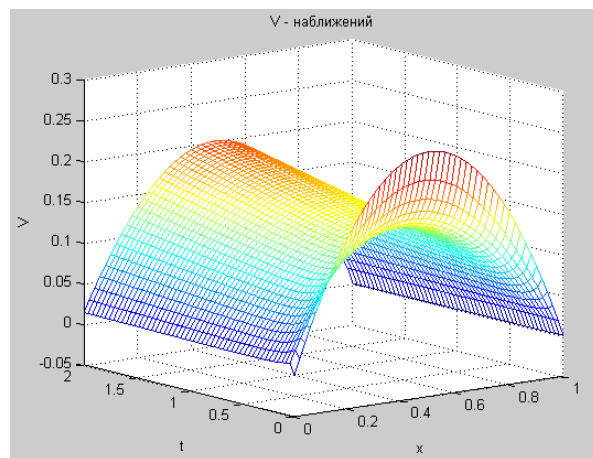
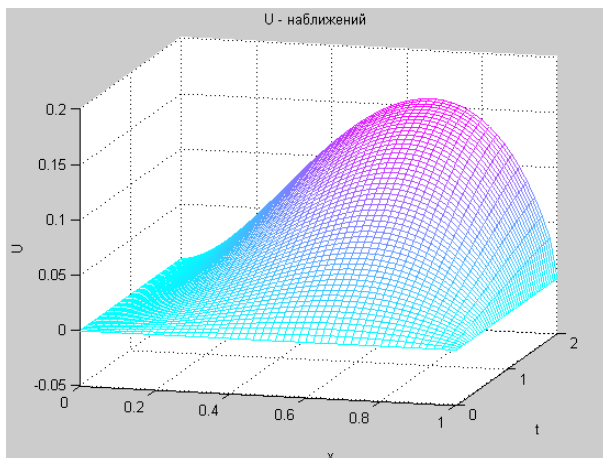
function s = g2(t)
    s = 0.05.*t;
end

function s = mj2(t)
    s = -1;
end

function F = nmf(t, y)
    % функція правої частини системи
    global n n1 np n2 N d d2 h X
    F = zeros(N, 1);
    for k = 1 : n
        F(k) = f1(X(k), t, y(k), y(k+n));
    end
    z1 = h.*g1(t)-1; z2 = h.*mj1(t);
    F(np) = d2.*(y(n2)+z1.*y(np)-z2)+f2(X(1), t, y(1), y(np));
    for k = 2 : n1
        j = k+n1;
        F(j) = d.*(y(j-1)-2.*y(j)+y(j+1))+f2(X(k), t, y(k), y(j));
    end
    z1 = h.*g2(t)+1; z2 = h.*mj2(t);
    F(N) = d2.*(y(N-1)-z1.*y(N)+z2)+f2(X(n), t, y(n), y(N));
End

```

і отримаємо наступний результат її виконання



Завдання для самостійної роботи:

п.1. Розв'язати модельну крайову задачу (5.8) при заданій комбінації КУ для функції $v(x,t)$ згідно індивідуального варіанту k Таблиці 5.1. Виведення результатів організувати аналогічно п.1 аудиторного завдання.

Таблиця 5.1 Варіанти індивідуального завдання.

k	КУ в точці $x=0$	КУ в точці $x=1$
1	$v(0,t)=0$	$v'_x(1,t)=t \cos(t)$
2	$v(0,t)=0$	$v'_x(1,t)+t * v(1,t)=t(\cos(t)+\sin(t))$
3	$v(0,t)=0$	$v'_x(1,t)-t * v(1,t)=t(\cos(t)-\sin(t))$
4	$v'_x(0,t)=t$	$v(1,t)=\sin(t)$
5	$v'_x(0,t)=t$	$v'_x(1,t)=t \cos(t)$
6	$v'_x(0,t)=t$	$v'_x(1,t)+0.5t * v(1,t)=t(\cos(t)+0.5\sin(t))$
7	$v'_x(0,t)=t$	$v'_x(1,t)-0.5t * v(1,t)=t(\cos(t)-0.5\sin(t))$
8	$v'_x(0,t)-t * v(0,t)=t$	$v(1,t)=\sin(t)$
9	$v'_x(0,t)-v(0,t)=t$	$v'_x(1,t)=t \cos(t)$
10	$v'_x(0,t)-0.2t * v(0,t)=t$	$v'_x(1,t)+v(1,t)=t \cos(t)+\sin(t)$
11	$v'_x(0,t)+0.1t * v(0,t)=t$	$v'_x(1,t)-v(1,t)=t \cos(t)-\sin(t)$
12	$v'_x(0,t)+v(0,t)=t$	$v'_x(1,t)=t \cos(t)$
13	$v'_x(0,t)+2v(0,t)=t$	$v'_x(1,t)+t * v(1,t)=t(\cos(t)+\sin(t))$
14	$v'_x(0,t)-2v(0,t)=t$	$v'_x(1,t)-t * v(1,t)=t(\cos(t)-\sin(t))$
15	$v'_x(0,t)+(t+2) * v(0,t)=t$	$v(1,t)=\sin(t)$
16	$v'_x(0,t)+(t-1) * v(0,t)=t$	$v'_x(1,t)=t \cos(t)$

Практичне заняття № 09

ТЕМА № 6. Різницевий метод розв'язання крайових задач для рівнянь параболічного типу.

6.1. Різницеві схеми для лінійних одновимірних задач (2 год).

Теоретична інформація:

Використовуючи метод різницевого схем, побудувати відповідну різницеву задачу і знайти наближений розв'язок мішаної задачі для рівняння параболічного типу

$$\frac{\partial u(x,t)}{\partial t} = \frac{\partial^2 u(x,t)}{\partial x^2} - p(x,t)u(x,t) + f(x,t), \quad (x,t) \in D, \quad (6.1)$$
$$p(x,t) \geq 0, \quad \bar{D} = D \cup \partial D \equiv [a,b] \times [0,T]$$

з початковою умовою

$$u(x,0) = u_0(x), \quad x \in [a,b], \quad (6.2)$$

а крайові умови в точках границі a, b сукупно можуть бути певною комбінацією КУ I і II-III роду. Враховуючи, що різницева задача в подальшому буде реалізована у вигляді відповідного програмного забезпечення, введемо параметри va, vb для ідентифікації варіанта КУ. Будемо вважати, що

$va = 0$ відповідає варіанту КУ I роду в точці $x = a$:

$$u(a,t) = \mu(t), \quad t \in [0,T] \quad (6.3.1a)$$

$va = 1$ відповідає варіанту КУ II-III роду в точці $x = a$:

$$g(t)u(a,t) - \frac{\partial u(a,t)}{\partial x} = \mu(t), \quad t \in [0,T]; \quad g(t) \geq 0 \quad (6.3.2a)$$

$vb = 0$ відповідає варіанту КУ I роду в точці $x = b$:

$$u(b,t) = \eta(t), \quad t \in [0,T] \quad (6.3.1b)$$

$vb = 1$ відповідає варіанту КУ II-III роду в точці $x = b$:

$$q(t)u(b,t) + \frac{\partial u(b,t)}{\partial x} = \eta(t), \quad t \in [0,T]; \quad q(t) \geq 0 \quad (6.3.2b)$$

Відомо, що при певних передбаченнях про гладкість функцій $p, f, u_0, g, \mu, q, \eta$, виконанню умов узгодження початкових і крайових умов в точках $(a;0), (b;0)$ розв'язок поставленої задачі існує, єдиний і неперервно залежить від вхідних даних. При отриманні різницевого апроксимації диференціальних операторів також будемо вважати, що шукана функція u також має необхідну гладкість.

Побудова сімейства різницевого схем. Уведемо рівномірні сітки

$$\bar{\omega}_h = \{x_k = a + (k-1)h, \quad k = \overline{1,n}, \quad b-a = (n-1)h\}; \quad (6.4)$$
$$\bar{\omega}_\tau = \{t_j = j\tau, \quad j = \overline{0,M}, \quad T = M\tau\},$$

і замінімо область $\bar{D}$ сітковою областю $\bar{\omega}_{h\tau} = \bar{\omega}_h \times \bar{\omega}_\tau$, а також будемо використовувати наступні позначення

$$\dot{u} = \frac{\partial u}{\partial t}, \quad Lu = \frac{\partial^2 u}{\partial x^2}; \quad Lu = \dot{u} + pu - f, \quad \dot{u} = Lu - pu + f;$$
$$\hat{y} \equiv \hat{y}_k = y_k^{j+1}, \quad y \equiv y_k = y_k^j, \quad \bar{y} \equiv \bar{y}_k = y_k^{j+0.5}; \quad \bar{t} \equiv \bar{t}_j = t_{j+0.5} = t_j + 0.5\tau; \quad \hat{t} = t_{j+1}; \quad (6.5)$$
$$y_{t,j} = \frac{\hat{y}_k - y_k}{\tau}; \quad \Lambda y = y_{\bar{x}\bar{x},k} \equiv \frac{y_{k-1} - 2y_k + y_{k+1}}{h};$$
$$0 \leq \sigma \leq 1, \quad y^{(\sigma)} \equiv Y = \sigma \hat{y} + (1-\sigma)y = 0.5(\hat{y} + y) + \tau(\sigma - 0.5)y_{t,j}.$$

Тут параметр σ відповідає за узагальнений варіант побудови однопараметричної сім'ї двошарових різницевих схем – схем з вагою на шеститочковому шаблоні $S\{(x_{k-1}, t_j), (x_k, t_j), (x_{k+1}, t_j), (x_{k-1}, \bar{t}), (x_k, \bar{t}), (x_{k+1}, \bar{t})\}$. З лекційного матеріалу ми знаємо, що від його значення залежить *порядок похибки апроксимації, стійкість і збіжність РЗ* до розв'язку диференціальної задачі. Враховуючи, що реалізація числових розрахунків також суттєво залежить від цього параметру, а саме, визначення $(y_k^{j+1})_{k=1}^n$ на сітковому шарі $j+1$:

- ❖ при значенні $\sigma=0$ (*явна схема*) – маємо формули алгебраїчного розрахунку за відомими значеннями $(y_k^j)_{k=1}^n$ з сіткового шару j ;
- ❖ при значеннях $0 < \sigma \leq 1$ (*неявна схема*) – маємо необхідність у розв'язанні СЛАР, тому із загальної РЗ з $0 \leq \sigma \leq 1$ отримаємо розрахункові формули окремо для варіанту $\sigma=0$ і варіанту $0 < \sigma \leq 1$.

При $0 \leq \sigma \leq 1$ стандартним чином будемо для (6.1) різницеве рівняння

$$\sigma \gamma * y_{k-1}^{j+1} - (1 + 2\sigma\gamma + \sigma\tau * \bar{p}_k) y_k^{j+1} + \sigma \gamma * y_{k+1}^{j+1} = -F_k^j, \quad (6.6)$$

$$k = \overline{2, n-1};$$

з порядком апроксимації $O(h^2 + \tau^m)$. Тут уведені позначення

$$F_k^j = (1 - \sigma)\gamma * (y_{k-1}^j + y_{k+1}^j) + (1 - 2(1 - \sigma)\gamma - (1 - \sigma)\tau * \bar{p}_k) * y_k^j + \tau * \bar{f}_k;$$

$$\gamma = \tau/h^2; \quad \bar{p}_k = p(x_k, \bar{t}); \quad \bar{f}_k = f(x_k, \bar{t});$$

$$m = \begin{cases} 1, & \sigma \in [0, \frac{1}{2}) \cup (\frac{1}{2}, 1], \\ 2, & \sigma = \frac{1}{2}. \end{cases}$$

Крайові умови I роду (6.3.1a)–(6.3.1b) апроксимуємо точно

$$y_1^{j+1} = \mu(t_{j+1}), \quad (6.7.1a)$$

$$y_n^{j+1} = \eta(t_{j+1}). \quad (6.7.1b)$$

Розглянемо, як здійснити апроксимацію крайових умов II-III роду, означених рівняннями (6.3.2a)–(6.3.2b), також з порядком $O(h^2 + \tau^m)$.

Різницеву КУ для (6.3.2a) в точці $x=a$ будемо записувати на шаблоні з чотирьох точок $S\{(x_1, t_j), (x_2, t_j), (x_1, t_{j+1}), (x_2, t_{j+1})\}$. Покажемо, що різницеве рівняння

$$\sigma(\hat{y}_{x,1} - \bar{g} * \hat{y}_1) + (1 - \sigma)(y_{x,1} - \bar{g} * y_1) = \frac{1}{2} h [(y_1)_{t,j} + \frac{1}{2} \bar{p}_1 (\hat{y}_1 + y_1) - \bar{f}_1] - \bar{\mu}, \quad (6.7.2a)$$

апроксимує (6.3.2a) на розв'язку рівняння (6.1) з порядком $O(h^2 + \tau^m)$. Тут ми позначили $\bar{g} = g(\bar{t}_j)$, $\bar{\mu} = \mu(\bar{t}_j)$. За відомою з лекційного матеріалу схемою, отримуємо рівняння для похибки $z = y - u(x_k, t_j)$, а саме, підставимо $y = z + u(x_k, t_j) \equiv z + (u)_k^j$ в різницеве рівняння (6.7.2a):

$$\sigma(z_{x,1}^{j+1} - \bar{g} * z_1^{j+1}) + (1 - \sigma)(z_{x,1}^j - \bar{g} * z_1^j) = 0.5h[(z_1)_{t,j} + \frac{1}{2} \bar{p}_1 (z_1^{j+1} + z_1^j)] - \beta_a,$$

де

$$\beta_a = \sigma((u)_{x,1}^{j+1} - \bar{g} * (u)_1^{j+1}) + (1 - \sigma)((u)_{x,1}^j - \bar{g} * (u)_1^j) - 0.5h[(u)_1^j]_{t,j} + \frac{1}{2} \bar{p}_1 ((u)_1^{j+1} + (u)_1^j) - \bar{f}_1 + \bar{\mu}$$

похибка апроксимації КУ (6.3.2a) різницеvim рівнянням (6.7.2a) на розв'язку u . Наступний технологічний етап – виконуємо розклад u в околі вузла $(x_1, \bar{t}_j)$ по формулі Тейлора. Для ясності деталізуємо елементи виразу β_a :

$$\underline{1.} \quad \sigma(u)_{x,1}^{j+1} + (1 - \sigma)(u)_{x,1}^j =$$

$$= \sigma/h \left[\left[(\bar{u})_1 + h(\bar{u}')_1 + 0.5h^2(\bar{u}'')_1 \right] - (\bar{u})_1 \right] + 0.5\tau \left[\left[(\bar{\dot{u}})_1 + h(\bar{\dot{u}})_1 \right] - (\bar{\dot{u}})_1 \right] +$$

$$\begin{aligned}
& + (1-\sigma)/h \left\{ \left(\overline{(u)}_1 + h \overline{(u')}_1 + 0.5h^2 \overline{(u'')}_1 \right) - \overline{(u)}_1 \right\} - 0.5\tau \left\{ \left(\overline{(\dot{u})}_1 + h \overline{(\dot{u})}_1 \right) - \overline{(\dot{u})}_1 \right\} + O(h^2 + \tau^2) = \\
& = \overline{(u')}_1 + 0.5h \overline{(u'')}_1 + (\sigma - 0.5)\tau \overline{(\dot{u}')}_1 + O(h^2 + \tau^2); \\
2. \quad & -\bar{g} * (\sigma(u)_1^{j+1} + (1-\sigma)(u)_1^j) = -\bar{g} * (\overline{(u)}_1 + (\sigma - 0.5)\tau \overline{(\dot{u})}_1) + O(\tau^2); \\
3. \quad & (u)_1^j_{t,j} = \overline{(\dot{u})}_1 + O(\tau^2); \\
4. \quad & \frac{1}{2}((u)_1^{j+1} + (u)_1^j) = \overline{(u)}_1 + O(\tau^2),
\end{aligned}$$

а для всього виразу β_a , маємо

$$\beta_a = \underbrace{[\bar{u}' - \bar{g} * \bar{u} + \bar{\mu}]_1}_{=0} + (\sigma - 0.5)\tau \underbrace{(\bar{u}' - \bar{g} * \bar{u})_1}_{=0} + 0.5h * \underbrace{[\bar{u}'' - \bar{u} - \bar{p} * \bar{u} + \bar{f}]_1}_{=0} + O(h^2 + \tau^2).$$

Як бачимо, в виразі β_a , перша квадратна дужка є КУ (6.3.1a) в точці $\bar{t}_j$, друга квадратна дужка є рівняння (6.1) в точці $\bar{t}_j$, а отже, при $\sigma=1/2$ отримаємо $\beta_a = O(h^2 + \tau^2)$, інакше маємо $\beta_a = O(h^2 + \tau)$ для значень $\sigma \in [0, 1] \setminus \{1/2\}$.

Різницеву крайову умову для (6.3.2b) в точці $x=b$ будемо також записувати на шаблоні з чотирьох точок $S\{(x_{n-1}, t_j), (x_n, t_j), (x_{n-1}, \hat{t}), (x_n, \hat{t})\}$ і, по аналогії з (6.7.2a), маємо таке різницеве рівняння

$$\sigma(\hat{y}_{\bar{x},n} + \bar{q} * \hat{y}_n) + (1-\sigma)(y_{\bar{x},n} + \bar{q} * y_n) = -\frac{1}{2}h[(y_n)_{t,j} + \frac{1}{2}\bar{p}_n(\hat{y}_n + y_n) - \bar{f}_n] + \bar{\eta}, \quad (6.7.2b)$$

де введено позначення $\bar{q} = q(\bar{t})$, $\bar{\eta} = \eta(\bar{t})$. Можна довести (виконати це самостійно!), що умова (6.7.2b) апроксимує (6.3.2a) на розв'язку рівняння (6.1) з порядком $O(h^2 + \tau^m)$.

Тепер маємо все необхідне, щоб перейти до запису розрахункових формул.

Явна схема. При $\sigma=0$ на шаблоні $S\{(x_{k-1}, t_j), (x_k, t_j), (x_{k+1}, t_j), (x_k, \hat{t})\}$ різницеве рівняння (6.6) запишемо у формі, зручній для проведення розрахунків

$$y_k^{j+1} = \gamma * (y_{k-1}^j + y_{k+1}^j) + (1-2\gamma - \tau * \bar{p}_k) * y_k^j + \tau * \bar{f}_k, \quad k = \overline{2, n-1}. \quad (6.8)$$

Обчислення значень y_1^{j+1} , y_n^{j+1} здійснюємо в залежності від КУ, зокрема для КУ II-III роду ми використаємо умови (6.7.2a), (6.7.2b) при $\sigma=0$:

$$y_1^{j+1} = \begin{cases} \mu(t_{j+1}), & va = 0, \\ P_1 * (G_1 * y_1^j + 2\gamma * y_2^j + \Phi_1), & va = 1 \end{cases} \quad (6.9a)$$

$$y_n^{j+1} = \begin{cases} \eta(t_{j+1}), & vb = 0, \\ P_n * (2\gamma * y_{n-1}^j + G_n * y_n^j + \Phi_n), & vb = 1 \end{cases} \quad (6.9b)$$

де

$$P_1 = (1 + 0.5\tau * \bar{p}_1)^{-1}; \quad G_1 = (1 - 2\gamma - 0.5\tau * \bar{p}_1 - \tilde{\gamma} * \bar{g}); \quad \Phi_1 = \tau * \bar{f}_1 + \tilde{\gamma} * \bar{\mu}; \quad \tilde{\gamma} = 2\tau * h^{-1}.$$

$$P_n = (1 + 0.5\tau * \bar{p}_n)^{-1}; \quad G_n = (1 - 2\gamma - 0.5\tau * \bar{p}_n - \tilde{\gamma} * \bar{q}); \quad \Phi_n = \tau * \bar{f}_n + \tilde{\gamma} * \bar{\eta};$$

Враховуючи, що для явної схеми маємо умовну стійкість, то при означенні параметрів схеми τ , h обираємо їх так, щоб виконувалась нерівність $2\tau \leq h^2$.

Неявна схема. При значеннях $0 < \sigma \leq 1$ ми уже маємо розрахункову формулу (6.6), яка описує для n невідомих $(y_k^{j+1})_{k=1}^n$ частину СЛАР – $(n-2)$ рівняння з номерами $k = 2, 3, \dots, n-1$. Необхідно доповнити СЛАР ще двома рівняннями. В залежності від комбінації КУ, будемо мати:

$$\begin{aligned}
& y_1^{j+1} = \mu(t_{j+1}), \quad va = 0, \\
& -C_1 * y_1^{j+1} + B_1 * y_2^{j+1} = -F_1, \quad va = 1;
\end{aligned} \quad \text{для рівняння з номером } k=1 \quad (6.10a)$$

$$\begin{aligned} y_n^{j+1} &= \eta(t_{j+1}), \quad vb=0, \\ A_n * y_{n-1}^{j+1} - C_n * y_n^{j+1} &= -F_n, \quad vb=1. \end{aligned} \quad \text{для рівняння з номером } k=n \quad (6.10b)$$

де

$$\begin{aligned} C_1 &= 1 + 2\sigma\gamma + 0.5\tau * \bar{p}_1 + \sigma\tilde{\gamma} * \bar{g}; \quad B_1 = 2\sigma\gamma; \\ F_1 &= [1 - 2(1-\sigma)\gamma - 0.5\tau * \bar{p}_1 - (1-\sigma)\tilde{\gamma} * \bar{g}]y_1 + 2(1-\sigma)\gamma * y_2 + \tau * \bar{f}_1 + \tilde{\gamma} * \bar{\mu}; \end{aligned}$$

$$\begin{aligned} A_n &= 2\sigma\gamma; \quad C_n = 1 + 2\sigma\gamma + 0.5\tau * \bar{p}_n + \sigma\tilde{\gamma} * \bar{q}; \\ F_n &= 2(1-\sigma)\gamma * y_{n-1} + [1 - 2(1-\sigma)\gamma - 0.5\tau * \bar{p}_n - (1-\sigma)\tilde{\gamma} * \bar{q}]y_n + \tau * \bar{f}_n + \tilde{\gamma} * \bar{\eta}. \end{aligned}$$

Отриману СЛАР (6.6), (6.10) розв'язують *монотонною прогонкою*, яка є стійка, враховуючи умови $p \geq 0, g \geq 0, q \geq 0$ і діагональну перевагу коефіцієнтів СЛАР.

Незалежно від виду схеми (явна | неявна), розрахунок починаємо з визначення $(y_k^0)_{k=1}^n$ за допомогою початкової умови (6.2) :

$$y_k^0 = u0(x_k), \quad k = \overline{1, n}. \quad (6.11)$$

Після цього, послідовно для шарів $j = \overline{1, M}$ знаходимо сіткову функцію для всієї області $\bar{\omega}_{h\tau}$.

Аудиторне завдання:

Розглянемо розв'язання модельної задачі, на базі якої протестуємо створене програмне забезпечення для побудованої явної і неявної РС з комбінаціями КУ I, II-III роду.

п.1. Розв'язати задачу (6.1), (6.2) з різними комбінаціями КУ I, II, III роду, використавши явну схему. Побудувати графіки наближеного розв'язку $Y(\bar{X}, \bar{J})$ і похибки $\delta(\bar{X}, \bar{J}) = Y(\bar{X}, \bar{J}) - u(\bar{X}, \bar{J})$. Обчислити значення $\|\delta(\bar{X}, \bar{J})\|_{Ch}$.

За наступних вхідних даних :

$$\begin{aligned} a &= 0, \quad b = 1; \quad T = 3.12, \\ p(x, t) &= (2 + x)t, \\ f(x, t) &= (2 + x(x - 2) + xt(4 - x^2))\exp(-t); \\ u0(x) &= x(2 - x); \\ g(t) &= 0.1t, \quad \mu(t) = -2\exp(-t); \\ q(t) &= 1, \quad \eta(t) = \exp(-t) \end{aligned} \quad (6.12)$$

маємо точний розв'язок $u(x, t) = x(2 - x)\exp(-t)$.

Зауваження. Якщо обрати кроки сітки $\bar{\omega}_{h\tau}$ зі значеннями $h = 10^{-2}$, $\tau = 4 * 10^{-5}$ (згадаємо умову стійкості явної схеми!), то розмірність вектор-рядка $\bar{X}$ – рівномірної сітки по x буде $n = 101$, а кількість шарів $M = 78000$. Зрозуміло, що для побудови 3D-графіка сіткового розв'язку $Y_{k,j}$ не потрібно зберігати значення для УСІХ M шарів, набагато раціональніше ввести $\bar{J}$ – сітку номерів j -шарів виведення результатів по змінній t_j .

Створимо власну М-функцію **fdm_P10** – солвер для розв'язання задачі (6.1)–(6.3) з використанням явної різницевої схеми (6.8)–(6.9), (6.11). Також напишемо відповідну М-функцію *p09a01*, в якій означимо анонімні функції $p, f, u0$ та підфункції Ga, Gb і зробимо виклик цього солвера :

```
function p09a01
clear all, close all, clc
% п.1 модельна задача
% параметри задачі :
a = 0; b = 1;
```

```

p = @(x,t)((2+x).*t);
f = @(x,t)((2+x.*(x-2)+(4-x.*x).*x.*t).*exp(-t));
u0 = @(x)(x.*(2-x));
v = [1, 1];
% параметри різницевої схеми :
n = 101; X = linspace(a, b, n);
tau = 4e-5; M = 78000; J = [0 : 1200 : M];
% розв'язання задачі :
U = fdm_P10(p, f, u0, @Ga, @Gb, v, X, J, tau);
J = tau.*J; m = length(J); T = J(m);
% точний розв'язок :
u = @(x,t)(x.*(2-x).*exp(-t));
Uu = U;
for i = 1 : n
    x = X(i);
    for k = 0 : m-1
        j = k+1; t = J(j); Uu(j,i) = u(x,t);
    end
end
[Mx, Mt] = meshgrid(X, J); % побудова сіток для 3D-графіків
% виведення графіків: розв'язку U і точного u(x,t)
de = [a, b, 0, T, -0.2, 1.2];
mesh(Mx, Mt, U); axis(de);
shading interp, colormap cool;
xlabel('x'), ylabel('t'), zlabel('U')
title('U - наближений');
pause(), close all
mesh(Mx, Mt, Uu); axis(de);
shading interp, colormap cool;
xlabel('x'); ylabel('t'); zlabel('u');
title('u(x,t) - точний');
pause(), close all
% виведення графіків похибки U-Uu
Uu = U-Uu;
C = norm(Uu,inf);
fprintf('Похибка ||U-u||c=%13.5e\n',C);
C = max(max(abs(Uu)));
fprintf('max(max(|U-u|)) =%13.5e\n',C);
mesh(Mx, Mt, Uu);
shading interp, colormap jet;
xlabel('x'); ylabel('t'); zlabel('U-u');
title('Похибка U-u(x,t)');
end

function gr = Ga(t)
% Опис [g(t), мю(t)] для x=a
% КУ I   роду va=0 : 1*u(a,t) = мю(t)
% КУ II  роду va=1 : -du(a,t)/dx + 0*u(a,t) = мю(t)
% КУ III роду va=1 : -du(a,t)/dx + g(t)*u(a,t) = мю(t)
gr = [ 1, -2*exp(-t)]; % III p
%gr = [ 0, -2*exp(-t)]; % II p
%gr = [ 1, 0]; % I p
end

function qr = Gb(t)
% Опис [q(t), ню(t)] для x=b
% КУ I   роду vb=0 : 1*u(b,t) = ню(t)
% КУ II  роду vb=1 : du(b,t)/dx + 0*u(b,t) = ню(t)
% КУ III роду vb=1 : du(b,t)/dx + q(t)*u(b,t) = ню(t)

```

```

qr = [ 2, 2.*exp(-t) ];          % III p
%qr = [ 1, exp(-t) ];           % II p
%qr = [ 1, 0 ];                 % I p
end

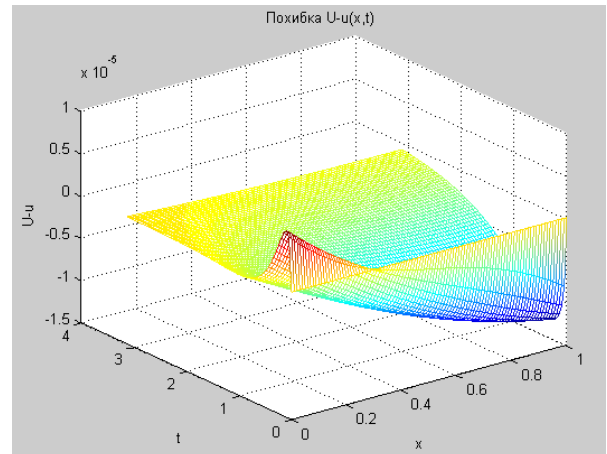
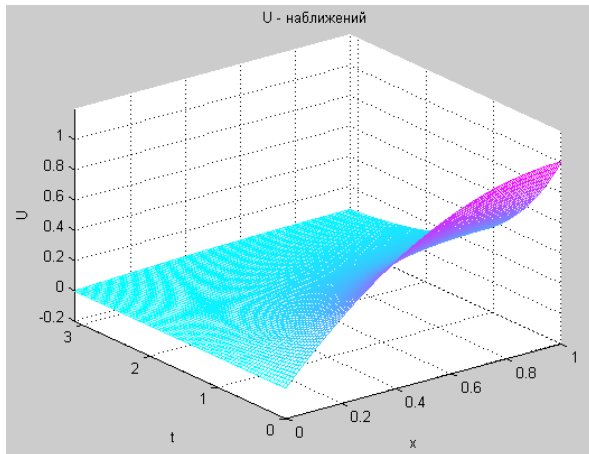
```

Запустимо її на виконання і отримаємо наступний результат (як бачимо значення похибки в $\|\circ\|_{Ch}$ узгоджується з теоретичною оцінкою $O(h^2 + \tau)$, враховуючи, що $h^2 + \tau = 1.4 \cdot 10^{-4}$):

```

Похибка ||U-u||c= 7.58614e-004
max(max(|U-u|)) = 1.27336e-005

```



Змінюючи значення вектора $v = [va, vb]$ – опис виду КУ в точках $x=a$, $x=b$ і відповідних функції $Ga(t)$ – опис $[g(t), \mu(t)]$ та $Gb(t)$ – опис $[q(t), \eta(t)]$, будемо отримувати наближений розв'язок (6.12) для різних варіантів комбінації КУ I, II-III роду.

п.2. Задачу (6.12), при умові збереження усіх вхідних параметрів і вимог візуалізації результату, розв'язати за допомогою неявної схеми.

Створимо власну М-функцію **fdm_P1s** – солвер для розв'язання задачі (6.1)–(6.3) з використанням неявної різницевої схеми (6.6), (6.10)–(6.11) і відповідну М-функцію **p09a02**, яка буде відрізнятися від **p09a01** тільки двома рядками:

```

tau = 1e-2; M = 315; J = [0 : 5 : M];

```

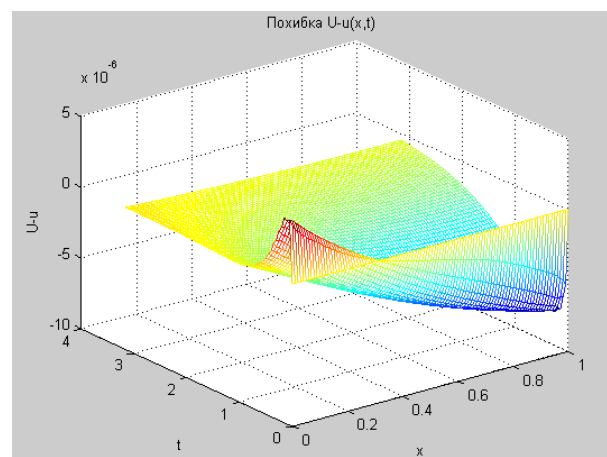
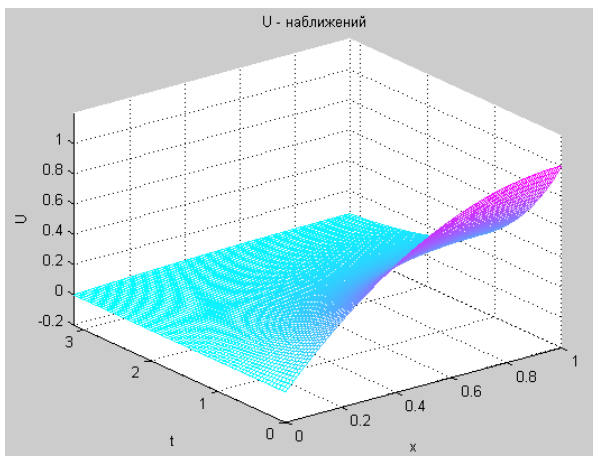
```

U = fdm_P1s(p, f, u0, @Ga, @Gb, v, X, J, tau, 0.5);

```

(тому повний її текст тут не приводимо).

Отримано наступний результат її виконання (у варіанті симетричної схеми)



```

Похибка ||U-u||c= 4.28959e-004
max(max(|U-u|)) = 7.67481e-006

```

Тут також значення похибки в $\|\circ\|_{Ch}$ узгоджується з теоретичною оцінкою $O(h^2 + \tau^2)$, враховуючи, що $h^2 + \tau^2 = 2 \cdot 10^{-4}$. Якщо порівняти з результатами явної схеми, то симетрична неявна схема значно ефективніша і по точності і по часу виконання.

Завдання для самотійної роботи:

п.1. Знайти наближений розв'язок задачі (6.1)–(6.3) за допомогою явної різницевої схеми, побудувати 3D-графік наближеного розв'язку. Належним чином обрати кроки h, τ для забезпечення точності $\approx 10^{-4}$ і стійкості різницевої задачі. Параметри задачі : $[a, b] = [0, 1]$, $T = 2.5$, решта – означені в Таблиці 6.1.

Таблиця 6.1 Варіанти індивідуального завдання.

k	$p(x, t)$	$f(x, t)$	$u_0(x)$	$[g(t), \mu(t)]$	$[q(t), \eta(t)]$	v
1	$\sin^2(x + gt)$	$(50 + xt)(g + x^2)$	x	$[0, \cos(t)]$	$[1, 1 - 0.1t]$	$[1, 0]$
2	$g \cos^2(x + t)$	$30 \exp(-gx) + 10x^3t$	$1 - x$	$[1, 1 - 0.2t]$	$[0, t - 1]$	$[0, 1]$
3	$(gx - t)^2$	$50x^3 \sin(t + gx)$	$1 + x$	$[0, 1 - \sin(t)]$	$[1, 2 - 0.5t]$	$[1, 0]$
4	$\sin^2(2/g - xt)$	$50gt - \exp(x - t^2)$	x	$[1, 1 + 0.2t]$	$[1, 1 - 0.1t]$	$[1, 1]$
5	$gt \cos(x + 0.5)$	$(x - t^2)/(1 + gx^2)$	$x^2 + 1$	$[1, \cos(2t)]$	$[1, 2 - 0.4t]$	$[0, 0]$
6	$ch(x/(t + g))$	$10(t + gx)/(1 + x)$	$1 - x$	$[2, 1 - 0.1t]$	$[0, t - 1]$	$[1, 1]$
7	$th(x^2 - t/g)$	$50 \exp(t - gx)$	$1 + x$	$[2, 3 + \sin(t)]$	$[1, 2 - 0.5t]$	$[1, 1]$
8	$\cos(x^2 + t/g)$	$5(t + gx)\sin(t + x^2)$	$1 - x^2$	$[1, \cos(2t)]$	$[1, t]$	$[0, 0]$
9	$2g \sin^2(x + t)$	$200g(t + x^2)$	$x^3 + 1$	$[1, 1 + 0.5t]$	$[1, 2 - t]$	$[0, 0]$
10	$\exp(gt - x^2)$	$70g * tg(2t - x^2)$	x	$[1, 0.3t]$	$[1, 1 - 0.2t]$	$[0, 1]$
11	$\sin^2(t + gx)$	$50\sin(t + x) + gx^2$	$1 - x$	$[1, 1 - 0.5t]$	$[1, 0.3t^2]$	$[0, 0]$
12	$1 + gt + x^3$	$300 * \sin^2(x + gt)$	$1 - x^3$	$[1, 1 + 0.5t]$	$[1, 0.2t^2]$	$[0, 0]$
13	$g + t \sin^2(2x)$	$300 \sin(x + gt)$	$x^3 + 1$	$[0, 1 + 0.2t]$	$[1, 4 + 0.1t]$	$[1, 1]$
14	$g + t + ch(x)$	$50(g - t + x^3)$	$x^3 - 1$	$[0, 0.2t - 1]$	$[2, 2 + 0.1t]$	$[1, 1]$
15	$th(t + gx^2)$	$\sqrt{t} + 50g * x^3$	$1 + x$	$[0, 1 + \sin(t)]$	$[0, 1 + 0.5t]$	$[1, 1]$
16	$g\sqrt{t} + x^2$	$(t + 50gx)\exp(x^3)$	$1 - x^2$	$[0, \sin(0.5t)]$	$[0, t]$	$[1, 1]$
17	$gt + \cos(x)$	$100(t + gx^2)$	$1 - x$	$[1, 1 - 0.2t]$	$[1, t(t + 1)]$	$[1, 0]$
18	$(t^2 + 1)/(x^2 + g)$	$50(gt + x^3)$	$x^3 + 1$	$[0, 0.2t]$	$[0, 3 + 0.2t]$	$[1, 1]$
19	$(t + g)/(x + g)$	$xt + 50 \sin(gx)$	$1 - x^3$	$[0, 0.4t]$	$[0, 0.6t^2 - 3]$	$[1, 1]$
20	$(x + g)/(t + g)$	$gxt + 50 * th(x)$	$x^3 - 1$	$[0, 0.2t - 1]$	$[0, 3 - 0.1t]$	$[1, 1]$
21	$\sin^2(x + t/g)$	$1e3(t + gx)/(1 + x^3)$	x	$[0, 1 + 0.5t]$	$[0, 1 + 0.4t]$	$[1, 1]$
22	$\cos^2(g - xt)$	$300g(t + x)x$	$1 - x$	$[0, -1 + 0.2t]$	$[0, 1 + 0.1t^2]$	$[1, 1]$
23	$g + \sqrt{t^3} + x$	$t^2 + 50g * th(x)$	$x^2 + 1$	$[0, t]$	$[0, 1 + 0.6\sqrt{t}]$	$[1, 1]$
24	$ch(1 + xt/g)$	$100gx(t + \sin(x))$	$\sqrt{x^3} + 1$	$[1, \cos(0.5t)]$	$[1, 2 - 0.4t]$	$[0, 0]$

25	$\sin^2(gt + x^2)$	$20(g + tx + \sqrt{x^3})$	$(1+x)/(10-x)$	$[1, 0.1+t]$	$[1, 2/9+t]$	$[0, 0]$
26	$g + x^2t$	$20(t + g\sqrt{x^2 - x})$	$(5+x)/(10-x)$	$[1, 0.2+t]$	$[1, 2/3 + \sin(t)]$	$[0, 0]$
27	$ch(gt - x^2)$	$300(t + gx^2)$	$e^{-x(1-x)}$	$[1, 1 + 0.2t]$	$[1, \cos(2t)]$	$[0, 0]$
28	$gx + \sin(t)$	$50(t+x)/(1+gt+x^3)$	x	$[0, \cos(t)]$	$[1, 1 + 0.3t]$	$[1, 1]$
29	$g + \sqrt{t^2 + x^4}$	$50ch(gt + x^3)$	$\sqrt{x^3} + 1$	$[0, 0.2t^2]$	$[1, 2 - 0.1t]$	$[1, 0]$
30	$g + t \cos^2(x)$	$200(t+x)/(1+gx^2)$	$1+x$	$[0, 1 + \sin(t)]$	$[0, 1 + 0.8t]$	$[1, 1]$

п.2. Знайти наближений розв'язок задачі (6.1)–(6.3) за допомогою чисто неявної різницевої схеми, побудувати 3D-графік наближеного розв'язку. Належним чином обрати кроки h, τ для забезпечення точності $\approx 10^{-4}$ розв'язку РЗ. Порівняти знайдений розв'язок з розв'язком, отриманим за допомогою **pdepe** – вбудованого солвера MATLAB. Параметри задачі означені в Таблиці 6.1.

Програмне забезпечення на MATLAB:

```
function U = fdm_P10(p, f, u0, Ga, Gb, v, X, J, tau)
% Наближений розв'язок крайової задачі :
% du(x,t)/dt = d^2u(x,t)/dx^2 - p(x,t)*u + f(x,t),
% p(x)>=0, x є (a,b),
% u(x,0) = u0(x), x є [a,b],
% КУ I роду va=0 : u(a,t) = мя(t)
% КУ II-III роду va=1 : -du(a,t)/dx + g(t)u(a,t) = мя(t)
% КУ I роду vb=0 : u(b,t) = ню(t)
% КУ II-III роду vb=1 : du(b,t)/dx + q(t)u(b,t) = ню(t)
% (явна схема sigm = 0)
% Вхідні дані :
% p - функція з описом p(x,t)
% f - функція з описом f(x,t)
% u0 - функція з описом u0(x)
% Ga - вектор-функція [ 1, мя(t) ] при va=0
% [g(t), мя(t)] при va=1
% Gb - вектор-функція [ 1, ню(t) ] при vb=0
% [q(t), ню(t)] при vb=1
% v - [va, vb] - масив значень виду КУ в x=a,b
% X - масив рівномірної сітки по x a=X(1),b=X(end)
% J - масив з номерами шарів для запам'ятовування розв'язку
% max(J(:)) = M - гранична точка сітки по t, T=M*tau
% tau - крок сітки по змінній t
% Вихідні дані:
% U - масив значень наближеного розв'язку
% U(j,k) = u(X(k),tau*J(j))

% параметри різницевої схеми :
n = length(X); nm = n-1; h = X(2)-X(1); hh = h.*h;
M = length(J); Y = X; U = zeros(M,n); M = max(J);
if 2*tau >= hh
    warning('FDM_P10: не виконується умова стійкості!');
end
% обчислення додаткових констант
tau05 = 0.5.*tau;
gam = tau./hh; gam2 = 2.*gam; gamt = 2.*tau./h; g12 = 1-gam2;
```

```

% формування початкових умов при t = 0 :
for k = 1 : n
    Y(k) = u0(X(k));
end
U(1,:) = Y; nJ = 2; % запам'ятовуємо для побудови 3D-графіку
for j = 1 : M
    t = (j-1).*tau; tt = t + tau05; tp = t+tau;
% знаходження наближеного розв'язку Y(:) на шарі t=t(j):
    yc = Y(1);
    if v(1) == 0
        gr = Ga(tp); Y(1) = gr(2);
    else
        gr = Ga(tt);
        x = X(1); pt = tau05.*p(x,tt); P = 1 + pt;
        F = tau.*f(x,tt) + gamt.*gr(2);
        x = g12 - pt - gamt.*gr(1);
        Y(1) = (x.*yc + gam2.*Y(2) + F)./P;
    end
    for k = 2 : nm
        x = X(k); yl = yc; yc = Y(k);
        P = tau.*p(x,tt); F = tau.*f(x,tt);
        Y(k) = gam.*(yl+Y(k+1)) + (g12-P).*yc + F;
    end
    if v(2) == 0
        qr = Gb(tp); Y(n) = qr(2);
    else
        qr = Gb(tt);
        x = X(n); pt = tau05.*p(x,tt); P = 1 + pt;
        F = tau.*f(x,tt) + gamt.*qr(2);
        x = g12 - pt - gamt.*qr(1);
        Y(n) = (gam2.*yc + x.*Y(n) + F)./P;
    end
    if j == J(nJ)
        U(nJ,:) = Y; nJ = nJ+1; % для побудови 3D-графіку
    end
end
end

function U = fdm_Pls(p, f, u0, Ga, Gb, v, X, J, tau, sigm)
% Наближений розв'язок крайової задачі :
%  $du(x,t)/dt = d^2u(x,t)/dx^2 - p(x,t)*u + f(x,t)$ ,
%  $p(x) \geq 0$ ,  $x \in (a,b)$ ,
%  $u(x,0) = u0(x)$ ,  $x \in [a,b]$ ,
% КУ I роду  $va=0$  :  $u(a,t) = mю(t)$ 
% КУ II-III роду  $va=1$  :  $-du(a,t)/dx + g(t)u(a,t) = mю(t)$ 
% КУ I роду  $vb=0$  :  $u(b,t) = ню(t)$ 
% КУ II-III роду  $vb=1$  :  $du(b,t)/dx + q(t)u(b,t) = ню(t)$ 
% (неявна схема  $0 < sigm \leq 1$ )
% Вхідні дані :
% p - функція з описом p(x,t)
% f - функція з описом f(x,t)
% u0 - функція з описом u0(x)
% Ga - вектор-функція [ 1, mю(t) ] при va=0
% [g(t), mю(t)] при va=1
% Gb - вектор-функція [ 1, ню(t) ] при vb=0
% [q(t), ню(t)] при vb=1
% v - [va, vb] - масив значень виду КУ в x=a,b
% X - масив рівномірної сітки по x a=X(1),b=X(end)
% J - масив з номерами шарів для запам'ятовування розв'язку

```

```

%          max(J(:)) = M - гранична точка сітки по t, T=M*tau
%      tau - крок сітки по змінній t
%      sigm - вага PC (0<sigm<=1)
% Вихідні дані:
%      U - масив значень наближеного розв'язку
%          U(j,k) = u(X(k),tau*J(j))

% параметри різницевої схеми :
n = length(X); nm = n-1; h = X(2)-X(1); hh = h.*h;
M = length(J); Y = X; U = zeros(M,n); M = max(J);
if sigm <= 0 | sigm > 1
    sigm = 0.5;
    warning('FDM_P1S: значення ваги sigm змінено на 0.5!');
end
% обчислення додаткових констант
tau05 = 0.5.*tau; s1 = 1-sigm;
gam = tau./hh; gam2 = 2.*gam; gamt = 2.*tau./h;
gams = s1.*gam; gam2s = s1.*gam2;
gam = sigm.*gam; gam2 = sigm.*gam2;
g12p = 1+gam2; g12m = 1-gam2s;
sgt = sigm.*gamt; slgt = s1.*gamt;
st = sigm.*tau; slt = s1.*tau;
% формування початкових умов при t = 0 :
for k = 1 : n
    Y(k) = u0(X(k));
end
U(1,:) = Y; nJ = 2; % запам'ятовуємо для побудови 3D-графіку
% формуємо A,C,B,F - елементи СЛАР
A = ones(1,n); C = A; F = A; A = gam.*A; B = A;
A(1) = 0;
if v(1) == 0
    B(1) = 0;
else
    B(1) = gam2;
end
if v(2) == 0
    A(n) = 0;
else
    A(n) = gam2;
end
B(n) = 0;

for j = 1 : M
    t = (j-1).*tau; tt = t + tau05; tp = t+tau;
% формуємо коефіцієнти СЛАР для шару t=t(j):
    if v(1) == 0
        gr = Ga(tp); F(1) = gr(2);
    else
        gr = Ga(tt); x = X(1); pt = tau05.*p(x,tt);
        C(1) = g12p+pt+sgt.*gr(1);
        F(1) = (g12m-pt-slgt.*gr(1)).*Y(1) + gam2s.*Y(2) ...
            + tau.*f(x,tt) + gamt.*gr(2);
    end
    for k = 2 : nm
        x = X(k); pt = p(x,tt);
        C(k) = g12p + st.*pt;
        F(k) = gams.*(Y(k-1)+Y(k+1)) + (g12m-slt.*pt).*Y(k) ...
            + tau.*f(x,tt);
    end
end

```

```

if v(2) == 0
    qr = Gb(tp); F(n) = qr(2);
else
    qr = Gb(tt); x = X(n); pt = tau05.*p(x,tt);
    C(n) = gl2p + pt + sgt.*qr(1);
    F(n) = gam2s.*Y(nm) + (gl2m-pt-s1gt.*qr(1)).*Y(n) ...
        + tau.*f(x,tt) + gamt.*qr(2);
end
% знаходження наближеного розв'язку Y(:) на шарі t=t(j):
[Y AL, er] = m_progM(A, C, B, F);
if j == J(nJ)
    U(nJ,:) = Y; nJ = nJ+1; % для побудови 3D-графіку
end
end
end

```

Практичне заняття № 10

ТЕМА № 6. Різницевий метод розв'язання крайових задач для рівнянь параболічного типу.

6.2. Квазілінійні крайові задачі для одновимірного рівняння (2 год).

Теоретична інформація:

Розглянемо застосування різницевого методу для наближеного розв'язку квазілінійних крайових задач для одновимірного рівняння параболічного типу в області $\bar{D} = D \cup \partial D = [a, b] \times [0, T]$. Якщо розглядати цю математичну модель як опис високотемпературних процесів з виділенням тепла в результаті хімічних реакцій, то маємо таке рівняння

$$\frac{\partial \varphi(x, t, u)}{\partial t} = \frac{\partial W(x, t, u, u')}{\partial x} + f(x, t, u). \quad (6.13)$$

У випадку, коли тепловий потік $W = W(x, t, u, u')$ лінійно залежить від $u' = \partial u / \partial x$ і виконується закон Фур'є $W = -K(x, t, u) * u'$, то рівняння (6.13) трансформується в таку форму запису

$$c(x, t, u) \frac{\partial u(x, t)}{\partial t} = \frac{\partial}{\partial x} \left(K(x, t, u) \frac{\partial u}{\partial x} \right) + f(x, t, u), \quad (6.14)$$
$$c(x, t, u) > 0, \quad K(x, t, u) > 0.$$

Типовим випадком практичних задач є варіант, коли параметри c , K , f залежать тільки від температури u і введенням нової змінної, рівняння (6.14) приводиться до виду

$$\frac{\partial \tilde{\varphi}(v)}{\partial t} = \frac{\partial^2 v}{\partial x^2} + \tilde{f}(v), \quad \left(v = \int_0^u K(\xi) d\xi \right), \quad (6.15)$$

або

$$\frac{\partial v}{\partial t} = \frac{\partial}{\partial x} \left(\kappa(v) \frac{\partial v}{\partial x} \right) + \tilde{f}(v), \quad \left(v = \int_0^u c(\xi) d\xi \right). \quad (6.16)$$

Враховуючи, що використання явних різницевих схем для задач (6.13)–(6.16) не має сенсу з огляду на значні обмеження на крок τ в умові стійкості, тому застосовують безумовно стійкі неявні схеми. Другою особливістю (6.13)–(6.16) є необхідність використання ітераційних процесів для розв'язання нелінійних систем рівнянь, які отримуємо після різницевої апроксимації рівнянь (6.13)–(6.16) і відповідних крайових умов. В залежності від форми рівняння (6.13)–(6.16) використовують методи: *послідовних наближень*, *Ньютона*.

Для отримання практичних навиків в побудові РС для крайових задач з квазілінійним рівнянням параболічного типу і в створенні відповідного програмного забезпечення, обмежимося розглядом варіанту крайових умов I роду, а отже рівняння (6.13)–(6.16) доповнимо крайовими умовами

$$u(a, t) = \mu(t), \quad u(b, t) = \eta(t), \quad t \in [0, T], \quad (6.17)$$

і початковою умовою

$$u(x, 0) = u_0(x), \quad x \in [a, b]. \quad (6.18)$$

Введемо в області $\bar{D}$ рівномірні сітки (6.4) і позначення (6.5).

6.2.1 Рівняння (6.14) апроксимуємо з порядком $O(h^2 + \tau)$ таким різницеvim рівнянням

$$\hat{c}_i * (\hat{y}_i - y_i) = \gamma * \left[\hat{K}_i \hat{y}_{i-1} - (\hat{K}_i + \hat{K}_{i+1}) \hat{y}_i + \hat{K}_{i+1} \hat{y}_{i+1} \right] + \tau * \hat{f}_i, \quad (6.19)$$
$$i = \overline{2, n-1}, \quad j = \overline{0, M-1};$$

де уведено позначення

$$\begin{aligned}\hat{c}_i &= c(x_i, \hat{t}, \hat{y}_i), \quad \hat{f}_i = f(x_i, \hat{t}, \hat{y}_i), \quad \gamma = \tau/h^2; \\ \hat{K}_i &= \frac{1}{2}(K(x_i, \hat{t}, \hat{y}_i) + K(x_i, \hat{t}, \hat{y}_{i-1})).\end{aligned}\quad (6.20)$$

Крайові умови (6.17) і початкові умови (6.18) апроксимуються точно

$$y_1^j = \mu(t_j), \quad y_n^j = \eta(t_j), \quad j = \overline{1, M}; \quad (6.21)$$

$$y_i^0 = u_0(x_i), \quad i = \overline{1, n}. \quad (6.22)$$

Для знаходження функції $\tilde{\mathbf{y}} = (y_k^{j+1})_{k=1}^n$ на кожному сітковому шарі $j = \overline{0, M-1}$, будуємо ітераційний процес (ІП) за наступним **алгоритмом**:

1. Початкове наближення $\tilde{\mathbf{y}}^{(0)}$ беремо з попереднього шару $\bar{\mathbf{y}}$ ($\tilde{\mathbf{y}}^{(0)} = \bar{\mathbf{y}}$).

2. Послідовно, для ітерацій $s = 0, \text{smax} - 1$

2.1 знаходимо розв'язок наступної СЛАР

$$\begin{cases} \hat{y}_1^{(s+1)} = \mu(t_{j+1}), \\ A_i \hat{y}_{i-1}^{(s+1)} - C_i \hat{y}_i^{(s+1)} + B_i \hat{y}_{i+1}^{(s+1)} = -F_i, \quad i = \overline{2, n-1}, \\ \hat{y}_n^{(s+1)} = \eta(t_{j+1}), \end{cases} \quad (6.23)$$

де $A_i = \gamma * \hat{K}_i^{(s)}$, $B_i = A_{i+1}$, $C_i = A_i + B_i + \hat{c}_i^{(s)}$, $F_i = \hat{c}_i^{(s)} * y_i + \tau * \hat{f}_i^{(s)}$.

2.2 перевіряємо умову

$$\max_i |\hat{y}_i^{(s+1)} - \hat{y}_i^{(s)}| \leq \varepsilon, \quad (6.24)$$

і при виконання (6.24) ітерації завершуємо.

Описаний ІП для більшості параметри c , K , f , які зустрічаються на практиці, при $\text{smax} = 5$ збігається.

Створимо власну М-функцію **fdm_P1n1** – солвер для розв'язання задачі (6.14), (6.17)–(6.18), який реалізує різницеву задачу (6.19)–(6.22) і ІП (6.23)–(6.24).

6.2.2 Перепозначимо в рівнянні (6.15) $v \rightarrow u$, $\tilde{\varphi} \rightarrow \varphi$, $\tilde{f} \rightarrow f$, після чого використаємо для апроксимації схему з $\sigma=1$ і отримаємо таке різницеве рівняння (порядок $O(h^2 + \tau)$)

$$\begin{aligned}\varphi(\hat{y}_i) - \varphi(y_i) &= \gamma * (\hat{y}_{i-1} - 2\hat{y}_i + \hat{y}_{i+1}) + \tau * f(\hat{y}_i), \\ i &= \overline{2, n-1}, \quad j = \overline{0, M-1}.\end{aligned} \quad (6.25)$$

Приєднуючи до (6.25) крайові і початкові умови (6.17)–(6.18) маємо нелінійну різницеву задачу. Для знаходження сіткової функції $\tilde{\mathbf{y}} = (y_k^{j+1})_{k=1}^n$ на кожному шарі $j = \overline{0, M-1}$ використаємо ІП – метод Ньютона. В цьому випадку виконаємо деякі зміни в **алгоритмі**, розглянутому вище, а саме, систему (6.23) замінимо на таку

$$\begin{cases} \Delta_1 = \mu(t_{j+1}) - \hat{y}_1^{(s)}, \\ A_i \Delta_{i-1} - C_i \Delta_i + B_i \Delta_{i+1} = -F_i, \quad i = \overline{2, n-1}, \\ \Delta_n = \eta(t_{j+1}) - \hat{y}_n^{(s)}, \end{cases} \quad (6.26)$$

де

$$\begin{aligned}\hat{y}_i^{(s+1)} &= \hat{y}_i^{(s)} + \Delta_i, \\ A_i &= \gamma, \quad B_i = \gamma, \quad C_i = 2\gamma + \varphi'(\hat{y}_i^{(s)}) - \tau * f'(\hat{y}_i^{(s)}), \\ F_i &= \gamma * (\hat{y}_{i-1}^{(s)} - 2\hat{y}_i^{(s)} + \hat{y}_{i+1}^{(s)}) + \tau * f(\hat{y}_i^{(s)}) + \varphi(y_i) - \varphi(\hat{y}_i^{(s)}).\end{aligned}$$

Створимо власну М-функцію **fdm_P1n2** – солвер для розв'язання задачі (6.15), (6.17)–(6.18), який реалізує різницеву задачу (6.25), (6.21)–(6.22) і ІП (6.26).

6.2.3 Рівняння (6.14) з параметрами $c \equiv 1, K = K(u), f = f(u)$ або рівняння (6.16) при $v \rightarrow u, \kappa = K, \tilde{f} \rightarrow f$ приймають форму

$$\frac{\partial u(x,t)}{\partial t} = \frac{\partial}{\partial x} \left(K(u) \frac{\partial u}{\partial x} \right) + f(u). \quad (6.27)$$

Якщо для різницевої апроксимації рівняння (6.27) скористатись схемою з вагою $\sigma = 1/2$, то отримаємо РС з порядком $O(h^2 + \tau^2)$, але такі схеми мають недолік – вони *немонотонні*, що може привести до появи «брижів» в чисельному розв’язку. Тому краще обмежитись чисто неявною схемою з подальшим використання метода Ньютона або ІІ виду (6.23) для розв’язання нелінійної системи різницевих рівнянь.

Для числового розв’язання нелінійної крайової задачі (6.27), (6.17)–(6.18) також використовують схему *предиктор-коректор*. Вона, по-перше, має точність $O(h^2 + \tau^2)$, а по-друге, не вимагає розв’язання нелінійної системи при переході з сіткового шару j на шар $j+1$, і записується наступним чином :

$$\begin{cases} \bar{y}_1 = \mu(\bar{t}_j), \\ \frac{\bar{y}_i - y_i}{0.5\tau} = (K_i^j * \bar{y}_{\bar{x}})_{x,i} + f(y_i), \quad i = \overline{2, n-1}, \\ \bar{y}_n = \eta(\bar{t}_j), \\ \hat{y}_1 = \mu(t_{j+1}), \\ \frac{\hat{y}_i - y_i}{\tau} = \frac{1}{2} \left[(K_i^{j+0.5} * \hat{y}_{\bar{x}})_{x,i} + (K_i^{j+0.5} * y_{\bar{x}})_{x,i} \right] + f(\bar{y}_i), \quad i = \overline{2, n-1}, \\ \hat{y}_n = \eta(t_{j+1}), \end{cases} \quad (6.28)$$

де $K_i^j = \frac{1}{2} (K(y_i^j) + K(y_{i-1}^j))$.

Для розв’язування СЛАР (6.23), (6.28) використовуємо монотонну прогонку, а для (6.26) – немонотонну прогонку.

Створимо власну М-функцію **fdm_P1n3** – солвер для розв’язання задачі (6.27), (6.17)–(6.18), який реалізує різницеву задачу (6.28), (6.21)–(6.22).

Аудиторне завдання:

Розглянемо чисельне розв’язання модельної задачі, точний розв’язок якої є функція $U(x,t) = tg(x+t)$

$$\begin{cases} U'_t(x,t) = U''_{xx}(x,t) + (1+U^2)(1-2U), \quad (x,t) \in (0,1) \times (0,T]; \\ U(x,0) = tg(x), \quad x \in [0,1]; \\ U(0,t) = tg(t), \quad U(1,t) = tg(1+t), \quad t \in [0,T], \quad T = 0.4. \end{cases} \quad (6.29)$$

На задачі (6.29) протестуємо створене ПЗ – солвери **fdm_P1n1**, **fdm_P1n2**, **fdm_P1n3** і побудуємо графіки наближеного розв’язку $\vec{Y}(\vec{X}, \vec{J})$ і похибки $\delta(\vec{X}, \vec{J}) = \vec{Y}(\vec{X}, \vec{J}) - u(\vec{X}, \vec{J})$, а також обчислимо значення $\|\delta(\vec{X}, \vec{J})\|_{Ch}$.

п.1. Розв’язати задачу (6.29) за допомогою солвера **fdm_P1n1**.

Напишемо відповідну М-функцію *p10a01*, в якій означимо анонімні функції $K, u0$ та підфункції *cf, mn* і зробимо виклик солвера **fdm_P1n1** :

```
function p10a01
clear all, close all, clc
% п.1 модельна задача
% параметри задачі :
a = 0; b = 1;
K = @(x,t,u) (1.0);
u0 = @(x) (tan(x));
```

```

% параметри різницевої схеми :
n = 201; X = linspace(a, b, n);
tau = 2.5e-5; M = 16000; J = [0 : 320 : M];
ep = 1e-10; sm = 10;
% розв'язання задачі :
Y = fdm_P1n1(K, @cf, u0, @mn, X, J, tau, ep, sm);
J = tau.*J; m = length(J); T = J(m);
% точний розв'язок :
u = @(x,t) (tan(x+t)); U = Y;
for i = 1 : n
    x = X(i);
    for j = 1 : m
        U(j,i) = u(x,J(j));
    end
end
end
ti1 = 'Y - наближений'; z1 = 'Y';
ti2 = 'U - точний'; z2 = 'U';
tp = 'Похибка'; z3 = 'Y-U'; ti3 = [tp, ' ', z3];
view_solu(X, J, Y, U, ti1, z1, ti2, z2, tp, ti3, z3, 5);
end

function e = cf(x,t,u)
% Опис [c(x,t,u), f(x,t,u)]
e = [ 1, (1+u.*u).*(1-2.*u)];
end

function e = mn(t)
% Опис [мю(t), ню(t)] для x=a, x=b
e = [ tan(t), tan(1+t)];
end

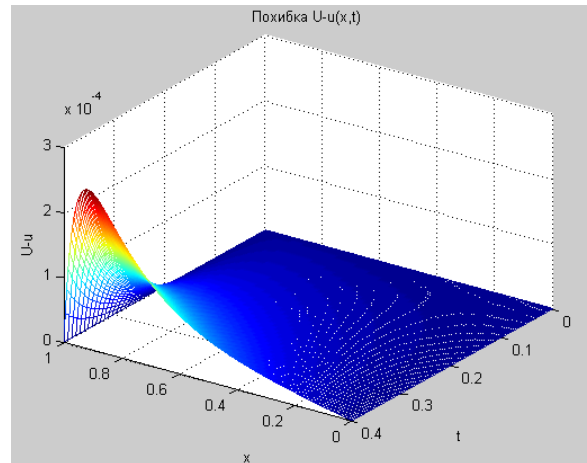
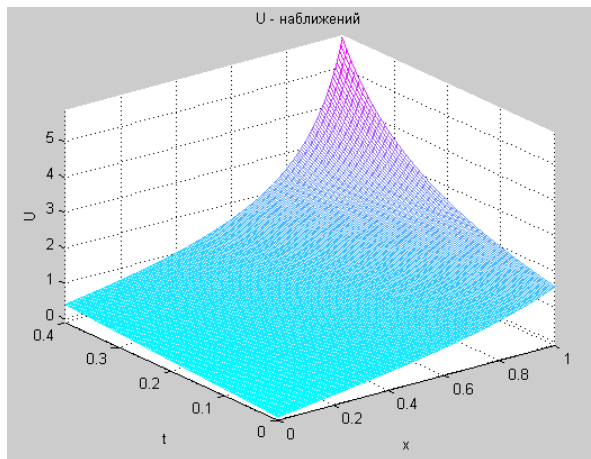
```

Запустимо її на виконання і отримаємо наступний результат :

```

Похибка ||U-u||с= 1.75616e-002
max(max(|U-u|)) = 2.43765e-004

```



п.2. Розв'язати задачу (6.29) за допомогою солвера **fdm_P1n2**.

Напишемо відповідну М-функцію *p10a02*, в якій означимо анонімні функції *fi*, *u0* та підфункції *fdf*, *mn* і зробимо виклик солвера **fdm_P1n2**. Враховуючи, що вона буде мало відрізнятися від *p10a01*, тому повний її текст не приводимо, а відобразимо тільки відмінні рядки :

```

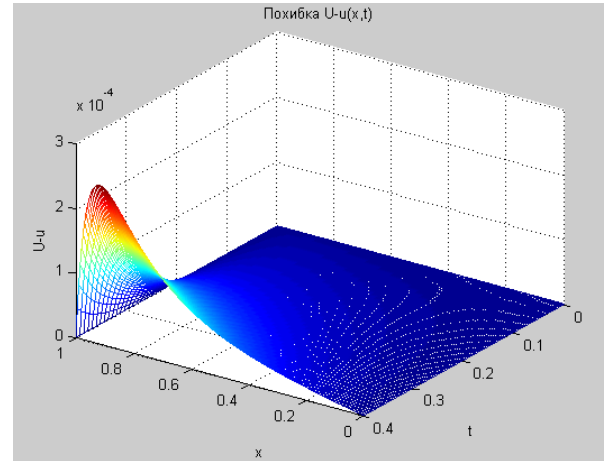
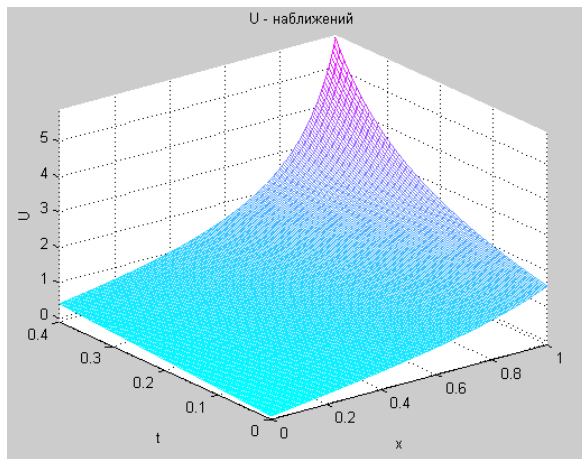
fi = @(u) (u);
...
U = fdm_P1n2(fi, @fdf, u0, @mn, X, J, tau, ep, sm);
...
function e = fdf(u)

```

```
% Опис [f(u), d(f(u))/du, d(phi(u))/du]
v = 1 + u.*u; z = 2.*u; w = 1 - z;
e = [ v.*w, z.*w-2.*v, 1];
end
```

Отримано наступний результат її виконання (як бачимо, аналогічної точності з результатами п.1.):

```
Похибка ||U-u||с= 1.75814e-002
max(max(|U-u|)) = 2.44120e-004
```



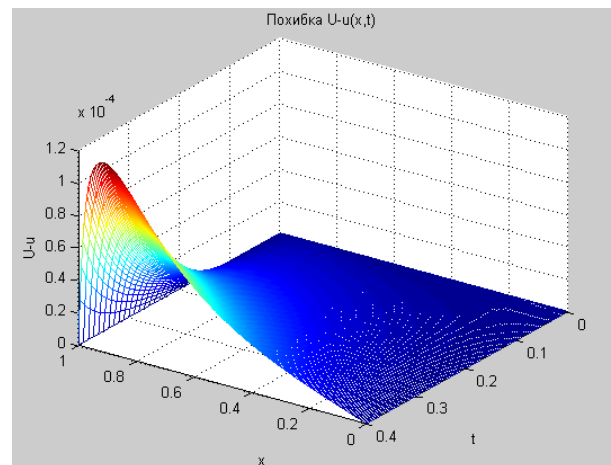
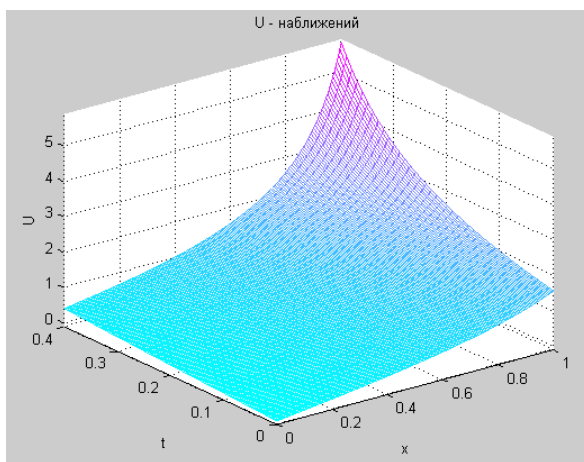
п.3. Розв'язати задачу (6.29) за допомогою солвера **fdm_P1n3**.

Напишемо відповідну М-функцію *p10a03*, в якій означимо анонімні функції *K, f, u0* та підфункцію *mn* і зробимо виклик солвера **fdm_P1n3**. Враховуючи, що вона буде мало відрізнятися від *p10a01*, тому повний її текст не приводимо, а відобразимо тільки відмінні рядки:

```
K = @(u) (1.0);
f = @(u) ((1+u.*u).*(1-2.*u));
...
tau = 1e-3; M = 400; J = [0 : 10 : M];
% розв'язання задачі :
U = fdm_P1n3(K, f, u0, @mn, X, J, tau);
```

Отримано наступний результат її виконання:

```
Похибка ||U-u||с= 9.76108e-003
max(max(|U-u|)) = 1.16073e-004
```



Як бачимо, при збільшенні кроку з $\tau = 2.5e-5$ (див. п.1, п.2) до $\tau = 1e-3$ і суттєвому зменшенні часу виконання, ми отримали результат $\approx$ в 2 рази кращий, якщо порівнювати похибки.

Завдання для самостійної роботи:

п.1. Знайти $Y(\bar{X}, \bar{J})$ – наближений розв’язок задачі (6.27), (6.17)–(6.18) за допомогою солверів **fdm_P1n1**, **fdm_P1n3**, а також $U(\bar{X}, \bar{J})$ – розв’язок, отриманий солвером **MATLABa pdepe**. Побудувати графіки: $Y(\bar{X}, \bar{J})$, $U(\bar{X}, \bar{J})$; різниці $\Delta(\bar{X}, \bar{J}) = Y(\bar{X}, \bar{J}) - U(\bar{X}, \bar{J})$, а також обчислити значення $\|\Delta(\bar{X}, \bar{J})\|_{Ch}$ і $\max_{i,j} |\Delta(\bar{X}, \bar{J})|$. Параметри

задачі: $[a, b] = [0, 1]$; $T = \frac{1}{2}$; $K(u) = \exp\left(1 - \frac{u^2}{30 + u^4}\right)$; $u0(x) = d(x - a)(b - x)$; $\mu(t) = g * \sin(g * t)$;

$\eta(t) = 1 - (1 + g * t)^{-1}$, решта – означені в Таблиці 6.2.

Таблиця 6.2 Варіанти індивідуального завдання.

<i>k</i>	<i>f(u)</i>	<i>k</i>	<i>f(u)</i>	<i>k</i>	<i>f(u)</i>
1	$\exp(-u)$	11	$u^2/(1+u^2)$	21	$1 - \operatorname{ch}(u)$
2	$u/(1+u^2)$	12	$\exp(-u^2)$	22	$u^2/(1+u^4)$
3	$u \sin(u)$	13	$u * \cos(u)$	23	$u^2/\sqrt{1+u^4}$
4	$\cos(u)/(1+u^2)$	14	$\sqrt{1+u^2}$	24	$u - \sin(u)$
5	$th(u)$	15	$sh(u)$	25	$2u + \cos(u)$
6	$u/(1+u^4)$	16	$u * \sin(u^2)$	26	$\sin(u)/\sqrt{1+u^2}$
7	$u/\sqrt{1+u^2}$	17	$u \exp(-u^2)$	27	$\exp(1-u)$
8	$u + \cos(u)$	18	$u - sh(u)$	28	$\sin(\pi * u)/\sqrt{1+u^2}$
9	$u * \sin(u)/\sqrt{1+u^4}$	19	$\exp(-u) \cos(u)$	29	$\exp(1-u^2)$
10	$\exp(-u) \sin(u)$	20	$\sin(u)/(1+u^2)$	30	$u * \cos(\pi * u)$

Програмне забезпечення на MATLAB:

```
function U = fdm_P1n1(K, cf, u0, mn, X, J, tau, ep, smax)
% Наближений розв'язок крайової задачі :
% c(x,t,u)*du(x,t)/dt = d(K(x,t,u)*u')/dx + f(x,t,u),
% c, K > 0, x ∈ (a,b),
% u(x,0) = u0(x), x ∈ [a,b],
% u(a,t) = мю(t), u(b,t) = ню(t).
% Вхідні дані :
% K - функція з описом K(x,t,u)
% cf - вектор-функція з описом [c(x,t,u), f(x,t,u)]
% u0 - функція з описом u0(x)
% mn - вектор-функція з описом [мю(t), ню(t)]
% X - масив рівномірної сітки по x a=X(1),b=X(end)
% J - масив з номерами шарів для запам'ятовування розв'язку
% тах(J(:)) = M - гранична точка сітки по t, T=M*tau
% tau - крок сітки по змінній t
% ep - точність (0<ep<1)
% smax - тах кількість ітерацій
% Вихідні дані :
% U - масив значень наближеного розв'язку
% U(j,k) = u(X(k),tau*J(j))

% параметри різницевої схеми :
n = length(X); nm = n-1; h = X(2)-X(1); hh = h.*h;
```

```

M = length(J); Y = X; U = zeros(M,n); M = max(J);
% обчислення додаткових констант
gam = tau./hh; gam05 = 0.5.*gam;
% формування початкових умов при t = 0 :
for i = 1 : n
    Y(i) = u0(X(i));
end
U(1,:) = Y; nJ = 2; % запам'ятовуємо для побудови 3D-графіку
% A,C,B,F - елементи СЛАР
A = zeros(1,n); B = A; F = A; C = ones(1,n);
for j = 1 : M
    t = j.*tau; Yp = Y;
    % формуємо F(1),F(n) для СЛАР на шарі j:
    m_n = mn(t); F(1) = m_n(1); F(n) = m_n(2);
    for s = 1 : smax
        Ys = Y;
        % формуємо решту коефіцієнтів СЛАР на шарі j:
        ki = K(X(1),t,Y(1)); kip = K(X(2),t,Y(2));
        for i = 2 : nm
            ip = i+1; x = X(i); y = Y(i); c_f = cf(x,t,y);
            kim = ki; ki = kip; kip = K(X(ip),t,Y(ip));
            ai = gam05.*(kim + ki); A(i) = ai;
            bi = gam05.*(ki + kip); B(i) = bi;
            C(i) = ai + bi + c_f(1);
            F(i) = c_f(1).*Yp(i) + tau.*c_f(2);
        end
        % знаходження Y на s-й ітерації для шару j:
        [Y AL, er] = m_prog(A, C, B, F);
        if max(abs(Y-Ys)) <= er % закінчуємо ітерацію по s
            break;
        end
    end
    if j == J(nJ)
        U(nJ,:) = Y; nJ = nJ+1; % для побудови 3D-графіку
    end
end
end
end

```

```

function U = fdm_P1n2(fi, fdf, u0, mn, X, J, tau, ep, smax)
% Наближений розв'язок крайової задачі :
%  $d(\phi(u))/dt = u''(x,t) + f(u)$ ,
%  $x \in (a,b)$ ,
%  $u(x,0) = u0(x)$ ,  $x \in [a,b]$ ,
%  $u(a,t) = \text{мю}(t)$ ,  $u(b,t) = \text{ню}(t)$ .
% Вхідні дані :
% fi - функція з описом  $\phi(u)$ 
% fdf - функція з описом  $[f(u), d(f(u))/du, d(\phi(u))/du]$ 
% u0 - функція з описом  $u0(x)$ 
% mn - вектор-функція з описом  $[\text{мю}(t), \text{ню}(t)]$ 
% X - масив рівномірної сітки по x  $a=X(1), b=X(\text{end})$ 
% J - масив з номерами шарів для запам'ятовування розв'язку
%  $\text{max}(J(:)) = M$  - гранична точка сітки по t,  $T=M*\text{tau}$ 
% tau - крок сітки по змінній t
% ep - точність ( $0 < \text{ep} < 1$ )
% smax - тах кількість ітерацій
% Вихідні дані :
% U - масив значень наближеного розв'язку
%  $U(j,k) = u(X(k), \text{tau}*J(j))$ 

```

```

% параметри різницевої схеми :
n = length(X); nm = n-1; h = X(2)-X(1); hh = h.*h;
M = length(J); Y = X; U = zeros(M,n); M = max(J);
% обчислення додаткових констант
gam = tau./hh; gam2 = 2.*gam;
% формування початкових умов при t = 0 :
for i = 1 : n
    Y(i) = u0(X(i));
end
U(1,:) = Y; nJ = 2; % запам'ятовуємо для побудови 3D-графіку
% A,C,B,F - елементи СЛАР
A = ones(1,n); F = A; C = A;
A(1) = 0; A(n) = 0; A = gam.*A; B = A;
for j = 1 : M
    t = j.*tau; Yp = Y; m_n = mn(t);
    for s = 1 : smax
        Ys = Y;
        % формуємо решту коефіцієнтів СЛАР на шарі j:
        F(1) = m_n(1) - Ys(1); F(n) = m_n(2) - Ys(n);
        for i = 2 : nm
            x = X(i); y = Ys(i); f_df = fdf(y);
            C(i) = gam2 - tau.*f_df(2) + f_df(3);
            F(i) = gam.*(Ys(i-1)-2.*y+Ys(i+1)) + ...
                tau.*f_df(1) + fi(Yp(i)) - fi(y);
        end
        % знаходження Y на s-й ітерації для шару j:
        [Y AL] = m_progn(A, C, B, F);
        x = max(abs(Y)); Y = Ys + Y;
        if x <= ep % закінчуємо ітерацію по s
            break;
        end
    end
    end
    if j == J(nJ)
        U(nJ,:) = Y; nJ = nJ+1; % для побудови 3D-графіку
    end
end
end
end

```

function U = fdm_P1n3(K, f, u0, mn, X, J, tau)

```

% Наближений розв'язок крайової задачі :
%  $du(x,t)/dt = d(K(u)*u')/dx + f(u)$ ,
%  $K > 0, x \in (a,b)$ ,
%  $u(x,0) = u0(x), x \in [a,b]$ ,
%  $u(a,t) = mю(t), u(b,t) = ню(t)$ .
% Вхідні дані :
% K - функція з описом K(u)
% f - функція з описом f(u)
% u0 - функція з описом u0(x)
% mn - вектор-функція з описом [мю(t), ню(t)]
% X - масив рівномірної сітки по x a=X(1),b=X(end)
% J - масив з номерами шарів для запам'ятовування розв'язку
% max(J(:)) = M - гранична точка сітки по t, T=M*tau
% tau - крок сітки по змінній t
% Вихідні дані :
% U - масив значень наближеного розв'язку
% U(j,k) = u(X(k),tau*J(j))

```

```

% параметри різницевої схеми :
n = length(X); nm = n-1; h = X(2)-X(1); hh = h.*h;
M = length(J); Y = X; U = zeros(M,n); M = max(J);
% обчислення додаткових констант
gam = tau./hh; gam05 = 0.5.*gam;
tau05 = 0.5.*tau; tau2 = 2.*tau;
% формування початкових умов при t = 0 :
for i = 1 : n
    Y(i) = u0(X(i));
end
U(1,:) = Y; nJ = 2; % запам'ятовуємо для побудови 3D-графіку
% A,C,B,F - елементи СЛАР
A = zeros(1,n); B = A; F = A; C = ones(1,n);
for j = 1 : M
    t = (j-0.5).*tau;
    % формуємо СЛАР на шарі j+0.5:
    m_n = mn(t); F(1) = m_n(1); F(n) = m_n(2);
    ki = K(Y(1)); kip = K(Y(2));
    for i = 2 : nm
        ip = i+1; y = Y(i);
        kim = ki; ki = kip; kip = K(Y(ip));
        ai = gam05.*(kim + ki); A(i) = ai;
        bi = gam05.*(ki + kip); B(i) = bi;
        C(i) = ai + bi + 2;
        F(i) = 2.*y + tau.*f(y);
    end
    % знаходження Y для шару j+0.5:
    [Yt, AL, er] = m_progM(A, C, B, F);
    % формуємо СЛАР на шарі j+1:
    t = t + tau05;
    m_n = mn(t); F(1) = m_n(1); F(n) = m_n(2);
    ki = K(Yt(1)); kip = K(Yt(2));
    for i = 2 : nm
        ip = i+1; y = Yt(i);
        kim = ki; ki = kip; kip = K(Yt(ip));
        ai = gam05.*(kim + ki); A(i) = ai;
        bi = gam05.*(ki + kip); B(i) = bi;
        x = ai + bi; C(i) = 2 + x; x = 2 - x;
        F(i) = ai.*Y(i-1)+x.*Y(i)+bi.*Y(ip) + tau2.*f(y);
    end
    % знаходження Y для шару j+1:
    [Y, AL, er] = m_progM(A, C, B, F);
    if j == J(nJ)
        U(nJ,:) = Y; nJ = nJ+1; % для побудови 3D-графіку
    end
end
end
end

function view_solu(X, J, Y, U, ti1, z1, ti2, z2, txtp, ti3, z3, s)
n = length(X); m = length(J);
mU = inf; MU = -inf;
for i = 1 : n
    x = X(i);
    for j = 1 : m
        de = U(j,i);
        if de > MU
            MU = de;
        elseif de < mU
            mU = de;
        end
    end
end

```

```

        mU = de;
    end
end
end
[Mx, Mt] = meshgrid(X, J); % побудова сіток для 3D-графіків
% виведення графіка розв'язку Y
de = [X(1), X(end), J(1), J(m), mU-0.1, MU+0.1];
mesh(Mx, Mt, Y); axis(de);
shading interp, colormap cool;
xlabel('x'), ylabel('t'), zlabel(z1);
title(ti1);
if s, pause(s);
else, pause;
end
close all
% виведення графіка U
mesh(Mx, Mt, U); axis(de);
shading interp, colormap cool;
xlabel('x'); ylabel('t'); zlabel(z2);
title(ti2);
if s, pause(s);
else, pause;
end
close all
% виведення графіка Y - U
YU = Y - U;
C = norm(YU,inf);
fprintf('%s ||Y-U||c=%13.5e\n',txtp,C);
C = max(max(abs(YU)));
fprintf('max(max(|Y-U|)) =%13.5e\n',C);
mesh(Mx, Mt, YU); view(-145,35);
shading interp, colormap jet;
xlabel('x'); ylabel('t'); zlabel(z3);
title(ti3);
end

```

Практичне заняття № 11

ТЕМА № 6. Різницевий метод розв'язання крайових задач для рівнянь параболічного типу.

6.3. Лінійні крайові задачі для багатовимірних рівнянь.

Локально-одновимірні різницеві схеми (2 год).

Теоретична інформація:

Розглянемо розв'язання крайової задачі :

$$\begin{cases} Pu \equiv \frac{\partial u(x,t)}{\partial t} - Lu - f(x,t) = 0, \\ (x,t) \in G \times (0,T]; \\ u(x,0) = u_0(x), \quad x \in \bar{G}; \\ u(x,t) = \mu(x,t), \quad x \in \partial G, \quad t \in [0,T]. \end{cases} \quad (6.30)$$

де

$$x = (x_1, x_2, \dots, x_p), \quad x_\alpha \in \bar{G}_\alpha, \quad \bar{G}_\alpha = G_\alpha \cup \partial G_\alpha = [0, a_\alpha], \quad \bar{G} = \bar{G}_1 \times \bar{G}_2 \times \dots \times \bar{G}_p;$$

$$L = L_1 + L_2 + \dots + L_p, \quad L_\alpha u = \frac{\partial^2 u(x,t)}{\partial x_\alpha^2}, \quad \alpha = 1, 2, \dots, p, \quad p \geq 2.$$

6.3.1 Побудова ЛОС у варіанті першого ланцюга [33]. Поставимо рівнянню $Pu = 0$ у відповідність ланцюг «одновимірних» рівнянь, а саме :

$$\sum_{\alpha=1}^p P_\alpha u = 0, \quad P_\alpha u = \frac{1}{p} \frac{\partial u(x,t)}{\partial t} - L_\alpha u - f_\alpha(x,t),$$

де $f_\alpha(x,t)$, $\alpha = 1, 2, \dots, p$ – довільні функції тієї-ж гладкості, що і $f(x,t)$, при цьому повинна виконуватись умова нормування $f(x,t) = f_1 + f_2 + \dots + f_p$.

На відрізку $0 \leq t \leq T$ введемо рівномірну сітку $\bar{\omega}_\tau = \{t_j = j\tau, j = 0, 1, \dots, M\}$ і кожен інтервал $(t_j, t_{j+1}]$ розіб'ємо на p частин точками $t_{j+\alpha/p} = t_j + \alpha * \tau/p$, $\alpha = 1, 2, \dots, p$.

Введемо позначення $\Delta_\alpha = (t_{j+(\alpha-1)/p}, t_{j+\alpha/p}]$, тоді $(t_j, t_{j+1}] = \bigcup_{\alpha=1}^p \Delta_\alpha$, а також позначимо

$u(x,t) = v_{(\alpha)}(x,t)$, $t \in \Delta_\alpha$. Будемо послідовно, починаючи з $\alpha = 1, 2, \dots$, розв'язувати рівняння

$$P_\alpha v_\alpha = 0, \quad x \in G, \quad t \in \Delta_\alpha, \quad \alpha = 1, 2, \dots, p, \quad (6.31)$$

додаючи при цьому початкові і граничні умови

$$\begin{aligned} v_{(1)}(x,0) &= u_0(x), \quad v_{(\alpha)}(x, t_{j+(\alpha-1)/p}) = v_{(\alpha-1)}(x, t_{j+(\alpha-1)/p}), \\ v_{(\alpha)}(x, t_{j+\alpha/p}) &= \mu(x, t_{j+\alpha/p}), \quad x \in \partial G, \quad \alpha = 1, 2, \dots, p. \end{aligned} \quad (6.32)$$

Розв'язком цієї задачі назвемо $u(x, t_j) = v_{(p)}(x, t_j)$, $j = 1, \dots, M$.

Апроксимуючи кожне з рівнянь (6.31) двошаровою схемою з вагою σ_α , отримуємо ланцюг p одновимірних схем, які і назвемо *локально-одновимірними схемами* (в подальшому будемо іменувати ЛОС-1) :

$$\begin{cases} \frac{1}{p} \frac{y^{j+\alpha/p} - y^{j+(\alpha-1)/p}}{\tau/p} = \Lambda_\alpha (\sigma_\alpha y^{j+\alpha/p} + (1-\sigma_\alpha) y^{j+(\alpha-1)/p}) + f_\alpha(x, t_j + 0.5\tau), \\ y^{j+\alpha/p} = \mu(x, t_{j+\alpha/p}), \quad x \in \partial G_{h,\alpha}, \quad \alpha = 1, 2, \dots, p; \\ j = 0, 1, \dots, M-1; \\ y^0 = u_0(x), \quad x \in \bar{G}_{h,\alpha}. \end{cases} \quad (6.33)$$

Тут триточковий різницевий оператор $\Lambda_\alpha \sim L_\alpha$,
 $\Lambda_\alpha u \equiv u_{\bar{x}_\alpha, k_\alpha} = h_\alpha^{-2} * (u_{k_\alpha-1} - 2u_{k_\alpha} + u_{k_\alpha+1})$, і залишені тільки ті індекси, які змінюються.

6.3.2 Побудова ЛОС у варіанті другого ланцюга [33]. Поставимо рівнянню $Pu = 0$ у відповідність інший ланцюг «одновимірних» рівнянь (в подальшому будемо іменувати ЛОС-2), а саме :

$$Pu = \sum_{\alpha=1}^p P_\alpha w_{(\alpha)} = 0, \quad (6.34)$$

$$P_\alpha w_{(\alpha)} = \frac{\partial w_{(\alpha)}(x, t)}{\partial t} - L_\alpha w_{(\alpha)} - f_\alpha(x, t), \quad x \in G, \quad t \in (t_j, t_{j+1}]$$

який доповнимо крайовими

$$w_{(\alpha)}(x, t) = \mu(x, t), \quad x \in \partial G, \quad \alpha = 1, 2, \dots, p; \quad (6.35)$$

і початковими умовами

$$w_{(\alpha)}(x, t_j) = \begin{cases} w_{(p)}(x, t_{j+1}), & j > 0 \\ u_0(x), & j = 0 \end{cases}, \quad \alpha = 1; \quad (6.36)$$

$$w_{(\alpha-1)}(x, t_{j+1}), \quad \alpha = 2, 3, \dots, p.$$

Розв'язком цієї задачі назвемо функцію $u(x, t_j) = w_{(p)}(x, t_j)$, $j = 1, \dots, M$. На відміну від варіанта ЛОС-1, кожне із рівнянь номера α розв'язується на всьому інтервалі $t \in (t_j, t_{j+1}]$. Різницеву задачу для (6.34)–(6.36) при просторовій розмірності $p = 2$ і варіанту схеми з вагами $\sigma_\alpha = 1$, $\alpha = \overline{1, p}$ ми записували на Лекції №10 (див. схему (5.54*)), тому не будемо повторюватися.

В регулярних вузлах сітки ЛОС має сумарну апроксимацію, як мінімум, $O(\tau + |h|^2)$, $|h|^2 = h_1^2 + h_1^2 + \dots + h_p^2$. За допомогою *принципу максимуму* можна отримати апіорні оцінки для стійкості ЛОС у метриці **C** і рівномірну збіжність зі швидкістю $O(\tau + h^2)$, $h = \max_{1 \leq \alpha \leq p} (h_\alpha)$ [33].

Тепер зупинимось на деяких особливостях ЛОС.

- ❖ В загальному випадку функції $v_{(p)}(x, t_j)$ ЛОС-1 і $w_{(p)}(x, t_j)$ ЛОС-2 не співпадають. За умови, що оператори L_α переставні, функція $f(x, t) \equiv 0$ & $(f_\alpha \equiv 0, \alpha = \overline{1, p})$ і крайові умови I роду однорідні – маємо співпадіння цих розв'язків.
- ❖ Який варіант (ЛОС-1 | ЛОС-2) обрати для чисельної реалізації задач типу (6.30) і більш складних ситуацій : $L_\alpha u = \frac{\partial}{\partial x_\alpha} \left(K_\alpha(x, t) \frac{\partial u}{\partial x_\alpha} \right)$, $K_\alpha(x, t, u)$, $f(x, t, u)$, чи наявність КУ II-III роду? Практика створення програмного забезпечення з використанням ЛОС і прості тести числових розрахунків показують перевагу варіанта ЛОС-1.
- ❖ Яке значення обрати для параметрів схеми $0 < \sigma_\alpha \leq 1$, $\alpha = \overline{1, p}$? Якщо взяти $\sigma_\alpha = \frac{1}{2}$, то відповідні різницеві рівняння для (6.31) мають сумарну апроксимацію порядку $O(\tau^2 + |h|^2)$, але це тільки рівняння! Для отримання похибки апроксимації $O(\tau^2)$ в крайових умовах необхідно їх записувати з певними «поправками». Наприклад, для КУ I роду, у вигляді типу (5.33) див. Лекцію №09. З огляду на цю проблему перевагу віддають чисто неявним ЛОС.

Якщо врахувати ці нюанси ЛОС, то можна переходити до створення програмного забезпечення, в якому реалізуємо ЛОС-1 при просторовій розмірності $p=2$ і варіанту чисто неявної схеми для мішаної крайової задачі (6.30) в прямокутній області $\bar{G} = \{(x_1, x_2) \in [a1, b1] \times [a2, b2]\}$.

Спочатку запишемо (6.33) у вигляді, зручному для алгоритмізації :

$$\begin{aligned} \bar{X1} &= a1 + (k1-1)h1, \quad k1=1, \bar{N1}, \quad b1 - a1 = (N1-1)*h1; & - \text{сітка по змінній } x_1; \\ \bar{X2} &= a2 + (k2-1)h2, \quad k2=1, \bar{N2}, \quad b2 - a2 = (N2-1)*h2; & - \text{сітка по змінній } x_2; \\ t_j &= j * \tau, \quad j=0, \bar{M}, \quad \tau = M * T; & - \text{сітка по змінній } t; \\ f_1 &= f_2 = \frac{1}{2} f; & - \text{представлення функції } f \text{ з урахуванням умови нормування;} \\ y_{k1, k2}^0 &= u0(x1, x2), \quad x1 \in \bar{X1}, \quad x2 \in \bar{X2}. \end{aligned} \quad (6.37)$$

$$\begin{aligned} (6.38.1) \quad & y_{k1,1}^{j+0.5} = \mu(x1_{k1}, a2, t_{j+0.5}), \quad y_{k1,1}^{j+0.5} = \mu(x1_{k1}, b2, t_{j+0.5}), \quad k1=1, \bar{N1}; \\ (6.38.2) \quad & \begin{cases} y_{1,k2}^{j+0.5} = \mu(a1, x2_{k2}, t_{j+0.5}) \\ \frac{y_{k1,k2}^{j+0.5} - y_{k1,k2}^j}{\tau} = \frac{1}{h1^2} (y_{k1-1,k2}^{j+0.5} - 2y_{k1,k2}^{j+0.5} + y_{k1+1,k2}^{j+0.5}) + f_1(x1_{k1}, x2_{k2}, t_{j+0.5}), \\ k1=2, \bar{N1-1}, \\ y_{N1,k2}^{j+0.5} = \mu(b1, x2_{k2}, t_{j+0.5}); \end{cases} \\ & k2=2, \bar{N2-1}; \\ (6.38.3) \quad & y_{1,k2}^{j+1} = \mu(a1, x2_{k2}, t_{j+1}), \quad y_{N1,k2}^{j+1} = \mu(b1, x2_{k2}, t_{j+1}), \quad k2=1, \bar{N2}; \\ (6.38.4) \quad & \begin{cases} y_{k1,1}^{j+1} = \mu(x1_{k1}, a2, t_{j+1}) \\ \frac{y_{k1,k2}^{j+1} - y_{k1,k2}^{j+0.5}}{\tau} = \frac{1}{h2^2} (y_{k1,k2-1}^{j+1} - 2y_{k1,k2}^{j+1} + y_{k1,k2+1}^{j+1}) + f_2(x1_{k1}, x2_{k2}, t_{j+1}), \\ k2=2, \bar{N2-1}, \\ y_{k1,N2}^{j+1} = \mu(x1_{k1}, b2, t_{j+1}); \end{cases} \\ & k1=2, \bar{N1-1}; \\ & j = 0, 1, \dots, M-1 \end{aligned} \quad (6.38)$$

Після отримання детальних формул (6.37)–(6.38), числове розв'язання крайової задачі (6.30) з використанням ЛОС-1 (6.33) для $p=2$ і $\sigma_{1,2}=1$ можна представити у вигляді наступного **алгоритма**

- 1.** Підготовча робота. Формуємо рівномірні сітки $\bar{X1}, \bar{X2}$ по змінним x_1, x_2 з кроками $h1, h2$. Обираємо крок τ і загальну кількість шарів M для значення T . Формуємо сітку (може бути і пуста!) $\bar{J}$ з номерами шарів j_{out} – виведення на МД наближеного розв'язку по змінній t . Формуємо матрицю СЛАР – масиви, елементи яких не залежать від значення номера шару :

$$\begin{aligned} A1_1 &= B1_1 = 0, \quad C1_1 = 1, \quad k1=1, \\ A1_{k1} &= B1_{k1} = 1, \quad C1_{k1} = 2 + \gamma1, \quad k1=2, \bar{N1-1} \\ A1_{N1} &= B1_{N1} = 0, \quad C1_{N1} = 1, \quad k1=N1; \quad \gamma1 = \frac{\tilde{h}1}{\tau}; \quad \tilde{h}1 = h1^2; \\ A2_1 &= B2_1 = 0, \quad C2_1 = 1, \quad k2=1, \\ A2_{k2} &= B2_{k2} = 1, \quad C2_{k2} = 2 + \gamma2, \quad k2=2, \bar{N2-1} \\ A2_{N2} &= B2_{N2} = 0, \quad C2_{N2} = 1, \quad k2=N2; \quad \gamma2 = \frac{\tilde{h}2}{\tau}; \quad \tilde{h}2 = h2^2. \end{aligned}$$

- 2.** За допомогою (6.37) обчислюємо сіткову функцію для $j=0$.
- 3.** Послідовно, для сіткових шарів $j=1, 2, \dots, M$ виконуємо наступні дії :

3.1 Обчислюємо сіткову функцію на горизонтальних граничних прямих

$x_2 = x_{2_1} \equiv a_2, x_2 = x_{2_{N_2}} \equiv b_2$ для $t = t_{j+0.5}$ по формулам (6.38.1);

Для кожної внутрішньої прямої $x_2 = x_{2_{k_2}}, k_2 = \overline{2, N_2 - 1}$, формуємо вектор правої частини СЛАР (6.38.2) :

$$F1_1 = \mu(a_1, x_{2_{k_2}}, t_{j+0.5}),$$

$$F1_{k_1} = \gamma_1 * y_{k_1, k_2}^j + \tilde{h}_1 * f1_{k_1, k_2}^{j+0.5}, \quad k_1 = \overline{2, N_1 - 1};$$

$$F1_{N_1} = \mu(b_1, x_{2_{k_2}}, t_{j+0.5}).$$

і знаходимо методом монотонної прогонки розв'язок наступної СЛАР :

$$\begin{cases} A1_{k_1} * y_{k_1-1, k_2}^{j+0.5} - C1_{k_1} * y_{k_1, k_2}^{j+0.5} + B1_{k_1} * y_{k_1+1, k_2}^{j+0.5} = -F1_{k_1}, \\ k_1 = \overline{1, N_1}. \end{cases}$$

3.2 Обчислюємо сіткову функцію на вертикальних граничних прямих

$x_1 = x_{1_1} \equiv a_1, x_1 = x_{1_{N_1}} \equiv b_1$ для $t = t_{j+1}$ по формулам (6.38.3);

Для кожної внутрішньої прямої $x_1 = x_{1_{k_1}}, k_1 = \overline{2, N_1 - 1}$ формуємо вектор правої частини СЛАР (6.38.4) :

$$F2_1 = \mu(x_{1_{k_1}}, a_2, t_{j+1}),$$

$$F2_{k_2} = \gamma_2 * y_{k_1, k_2}^{j+0.5} + \tilde{h}_2 * f2_{k_1, k_2}^{j+1}, \quad k_2 = \overline{2, N_2 - 1};$$

$$F2_{N_2} = \mu(x_{1_{k_1}}, b_2, t_{j+1}).$$

і знаходимо методом монотонної прогонки розв'язок наступної СЛАР :

$$\begin{cases} A2_{k_2} * y_{k_1, k_2-1}^{j+1} - C2_{k_2} * y_{k_1, k_2}^{j+1} + B2_{k_2} * y_{k_1, k_2+1}^{j+1} = -F2_{k_2}, \\ k_2 = \overline{1, N_2}. \end{cases}$$

Реалізуємо **алгоритм** у вигляді М-функції (опис аргументів див. Програмне забезпечення на MATLAB):

function Y = fmd_P21s(f1, f2, muh, muv, u0, X1, X2, tau, M, J)

Аудиторне завдання:

п.1. Виконати тестування М-функції **fmd_P21s** на розв'язанні модельної задачі (6.30) при $p = 2$ і наступних вхідних даних :

$$f(x_1, x_2, t) = \exp(-t) (4 + x_1^2 + x_2^2) - 7; \quad a_1 = a_2 = 0, \quad b_1 = 1, \quad b_2 = 2, \quad T = 5;$$

$$u0(x_1, x_2) = x_1^2 + x_2^2;$$

$$u(0, x_2, t) = \mu(0, x_2, t) \equiv t + x_2^2 (2 - \exp(-t)),$$

$$u(1, x_2, t) = \mu(1, x_2, t) \equiv t + (1 + x_2^2) (2 - \exp(-t));$$

$$u(x_1, 0, t) = \mu(x_1, 0, t) \equiv t + x_1^2 (2 - \exp(-t)),$$

$$u(x_1, 2, t) = \mu(x_1, 2, t) \equiv t + (x_1^2 + 4) (2 - \exp(-t)).$$

(6.39)

З урахуванням (6.39), маємо для мішаної крайової задачі (6.30) точний розв'язок $u(x_1, x_2, t) = t + (x_1^2 + x_2^2) (2 - \exp(-t))$. Побудувати графіки : наближеного, точного і похибки для значення $t = T$.

Напишемо відповідну М-функцію *p11a01*, в якій означимо анонімні функції f (це опис f_1, f_2 !), $u0, u$ та підфункції :

$\mu h = [\mu(x_1, a_2, t), \mu(x_1, b_2, t)]$ – опис крайових умов I роду «по горизонталі»;

$\mu v = [\mu(a_1, x_2, t), \mu(b_1, x_2, t)]$ – опис крайових умов I роду «по вертикалі»

і зробимо виклик солвера **fdm_P21s** :

function p11a01
clear all, close all, clc

```

% п.1 модельна задача ЛОС-1 (1-й ланцюг)
% параметри задачі :
a1 = 0; b1 = 1; a2 = 0; b2 = 2;
f = @(x1,x2,t) (0.5.*exp(-t).*(4+x1.*x1+x2.*x2)-3.5);
u0 = @(x1,x2) (x1.*x1+x2.*x2);
% параметри різницевої схеми :
N1 = 21; X1 = linspace(a1, b1, N1);
N2 = 41; X2 = linspace(a2, b2, N2);
tau = 25e-4; M = 2000; J = [];
% розв'язання задачі :
Y = fmd_P2ls(f, f, @muh, @muv, u0, X1, X2, tau, M, J);
% точний розв'язок :
u = @(x1,x2,t) (t+(x1.*x1+x2.*x2).*(2-exp(-t)));
U = zeros(N2,N1); T = tau.*M;
for k1 = 1 : N1
    x1 = X1(k1);
    for k2 = 1 : N2
        U(k2,k1) = u(x1,X2(k2),T);
    end
end
end

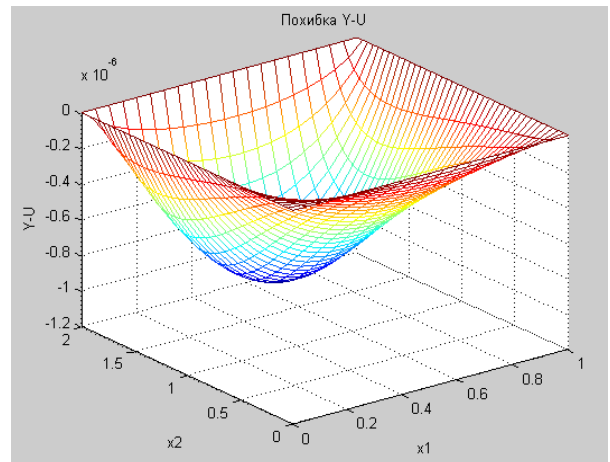
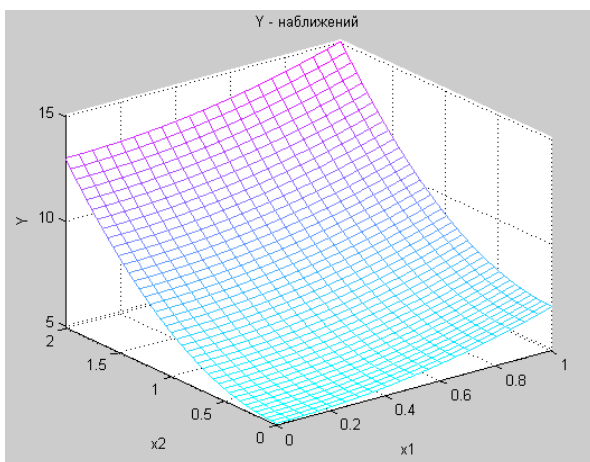
til = 'Y - наближений'; z1 = 'Y';
ti2 = 'U - точний'; z2 = 'U';
tp = 'Похибка'; z3 = 'Y-U'; ti3 = [tp, ' ', z3];
view_solu(X1, X2, Y, z1, ti1, 0, 5, U, z2, ti2, tp, z3, ti3);
end

function [ga, gb] = muh(x1, t)
% Опис [мю(x1,a2,t), мю(x1,b2,t)]
xx = x1.*x1; e = 2 - exp(-t);
ga = t + xx.*e; gb = t + (xx + 4).*e;
end

function [ga, gb] = muv(x2, t)
% Опис [мю(a1,x2,t), мю(b1,x2,t)]
xx = x2.*x2; e = 2 - exp(-t);
ga = t + xx.*e; gb = t + (1 + xx).*e;
end

```

Отримано наступний результат її виконання (точний розв'язок тут не приводимо, з огляду на величину похибки) :



Похибка $\|Y-U\|_c = 1.37740e-005$
 $\max(\max(|Y-U|)) = 1.01012e-006$

Завдання для самостійної роботи:

п.1. Знайти $Y(\overline{X1}, \overline{X2})$ – наближений розв’язок задачі (6.30) при $p = 2$ за допомогою М-функції **fmd_P21s**. Побудувати графіки наближеного розв’язку. Параметри задачі :

$$a1 = a2 = 0, b1 = b2 = 1; T = 2;$$

$$u0(x_1, x_2) = g * x_1(b1 - x_1)x_2(b2 - x_2) \exp\left(5\left(x_1 - \frac{1}{4}\right)^2 + 2\left(x_2 - \frac{3}{4}\right)^2\right);$$

$$\mu(a1, x_2, t) = \mu(b1, x_2, t) = 0.1 * t * (x_2 - a2)(b2 - x_2);$$

$$\mu(x_1, a2, t) = \mu(x_1, b2, t) = 0.1 * t * (x_1 - a1)(b1 - x_1),$$

решта – означені в Таблиці 6.3.

Таблиця 6.3 Варіанти індивідуального завдання.

k	$f(x_1, x_2, t)$	k	$f(x_1, x_2, t)$	k	$f(x_1, x_2, t)$
1	$10d * t * \exp(-x_1 x_2)$	11	$\frac{10d(t + x_1)^2}{1 + x_1^2 + x_2^2}$	21	$\frac{10d(t + x_1 + x_2)}{3 + \cos^2(x_1 + x_2)}$
2	$\frac{50d * x_1 x_2}{t + (x_1 + x_2)^2}$	12	$20d \frac{x_1 + x_2}{1 + t^2}$	22	$\frac{10d(t + x_1 x_2)^2}{1 + t + x_1 x_2}$
3	$\frac{10d * t}{5 + \sin(\pi(x_1 + x_2))}$	13	$\frac{20d * t(x_1 + x_2)}{2 + t + \cos(\pi * x_1 x_2)}$	23	$\frac{10d(x_1 + 3x_2)^2}{\sqrt{1 + t^4}}$
4	$10d \frac{x_1 + x_2}{1 + t^2}$	14	$30d \sqrt{1 + t(x_1 + x_2)}$	24	$10d(t - \sin(\pi(x_1 + x_2)))$
5	$\frac{20d(1 + t)}{d + x_1^2 + x_2^2}$	15	$20d * t(x_1 - x_2)^2$	25	$\frac{2t + 20d(x_1 + x_2)}{2 + \cos(\pi(x_1 + x_2))}$
6	$5 \frac{dx_1^2 + t}{1 + x_2^4}$	16	$\frac{20d * (t + 0.5)}{3 + \sin(2x_1 + 3x_2)}$	26	$\frac{20d \sin(t + x_1 + x_2)}{\sqrt{1 + t^2}}$
7	$\frac{10d * x_1 x_2}{\sqrt{1 + t + (x_1 x_2)^2}}$	17	$\frac{30d \exp(-t^2)}{1 + 4(x_1 + x_2)^3}$	27	$\frac{10d(t + x_1 + x_2)}{\exp(t + x_1 + x_2)}$
8	$\frac{d * x_1 x_2 + 5t}{4 + \cos(x_1 - x_2)}$	18	$10d * t - \sin(x_1 + 2x_2)$	28	$\frac{10d \sin^2(t)}{\sqrt{1 + (x_1 + x_2)^2}}$
9	$\frac{10d * \sin(t + x_1)}{\sqrt{1 + (x_1 - x_2)^4}}$	19	$\frac{10d \exp(-t)}{2 + \cos^2(x_1 - x_2)}$	29	$\frac{10d \exp(1 - t^2)}{3 + (x_1 + x_2)^2}$
10	$\frac{10d \exp(-t)}{4 - \sin(x_1 - x_2)}$	20	$\frac{20d \sin(x_1 - x_2)}{1 + t^2 + \sqrt{x_1 x_2}}$	30	$\frac{12d * (t + x_1 + x_2)}{4 + \cos(\pi(x_1 + x_2))}$

Програмне забезпечення на MATLAB:

function Y = fmd_P21s(f1, f2, muh, muv, u0, X1, X2, tau, M, J)

% Розв'язування лінійної мішаної задачі для двовимірного

% рівняння параболічного типу з КУ I роду за допомогою

% ЛОС-1 (1-й ланцюг) у варіанті чисто неявної схеми.

% Вхідні дані :

% f1, f2 - функції з описом f1, f2, де $f1 + f2 = f(x_1, x_2, t)$

% muh - функція мю на "горизонталі" [мю(x1, X2(1), t), мю(x1, X2(N2), t)]

% muv - функція мю на "вертикалі" [мю(X1(1), x2, t), мю(X1(N1), x2, t)]

% u0 - функція з описом $u0(x_1, x_2)$

% X1, X2 - масив рівномірної сітки по змінним x_1, x_2

```

%      tau - крок сітки по змінній t
%      M - гранична точка сітки по t (T=M*tau)
%      J - масив з номерами шарів для запису розв'язку на МД.
%      Має бути J(1)<J(2)<...<J(end) або пустий.
% Вихідні дані :
%      Y - масив значень наближеного розв'язку
%      Y(k2,k1) = u(X1(:),X2(:)) для j = M

% параметри різницевої схеми :
N1 = length(X1); N1m = N1 - 1; h1 = X1(2) - X1(1);
N2 = length(X2); N2m = N2 - 1; h2 = X2(2) - X2(1);
% обчислення додаткових констант :
tau05 = 0.5.*tau;
h12 = h1.*h1; ga1 = h12./tau; c1 = 2 + ga1;
h22 = h2.*h2; ga2 = h22./tau; c2 = 2 + ga2;
% формування елементів матриці СЛАР незалежних від j :
C1 = ones(1,N1); F1 = C1; A1 = [0,C1(2:N1m),0];
B1 = A1; C1(2:N1m) = c1;
C2 = ones(1,N2); F2 = C2; A2 = [0,C2(2:N2m),0];
B2 = A2; C2(2:N2m) = c2;
% формування початкових умов при t=0 :
Y = zeros(N2,N1);
for k2 = 1 : N2
    x2 = X2(k2);
    for k1 = 1 : N1
        Y(k2,k1) = u0(X1(k1),x2);
    end
end
nempJ = ~isempty(J);
if nempJ
    jout = 1; name = 'P2ls_';
    if J(jout) == 0
        namef = sprintf('%s%u.mat', name, J(jout));
        save(namef, 'Y'); jout = jout + 1;
    end
end
for j = 1 : M
% знаходження наближеного розв'язку Y на шарі j-1/2 :
    t = j.*tau; tt = t - tau05;
% обчислення крайових умов в точках X2(1),X2(N2) :
    for k1 = 1 : N1
        [c1, c2] = muh(X1(k1),tt); Y(1,k1) = c1; Y(N2,k1) = c2;
    end
    for k2 = 2 : N2m
        x2 = X2(k2);
% формування F1 :
        [c1, c2] = muv(x2,tt); F1(1) = c1; F1(N1) = c2;
        for k1 = 2 : N1m
            F1(k1) = ga1.*Y(k2,k1) + h12.*f1(X1(k1),x2,tt);
        end
% розв'язок СЛАР, знаходження Y(k2,:) :
        [y, alfa, err] = m_prog(A1, C1, B1, F1);
        Y(k2,:) = y;
    end
    clear y alfa
% знаходження наближеного розв'язку Y на шарі j :
% обчислення крайових умов в точках X1(1),X1(N1) :
    for k2 = 1 : N2
        [c1, c2] = muv(X2(k2),t); Y(k2,1) = c1; Y(k2,N1) = c2;
    end
end

```

```

    end
    for k1 = 2 : N1m
        x1 = X1(k1);
% формування F2 :
        [c1, c2] = muh(x1,t); F2(1) = c1; F2(N2) = c2;
        for k2 = 2 : N2m
            F2(k2) = ga2.*Y(k2,k1) + h22.*f2(x1,X2(k2),t);
        end
% розв'язок СЛАР, знаходження Y(:,k1) :
        [y, alfa, err] = m_progm(A2, C2, B2, F2);
        Y(:,k1) = y';
    end
    clear y alfa
    if nempJ
        if J(jout) == j
            namef = sprintf('%s%u.mat', name, J(jout));
            save(namef, 'Y'); jout = jout + 1;
        end
    end
end
end
end

```

function view_solu(X1, X2, Y, z1, ti1, pnt, s, U, z2, ti2, txtp, z3, ti3)

```

N1 = length(X1); N2 = length(X2);
mY = inf; MY = -inf; % mY=min(Y), MY=max(Y)
for k2 = 1 : N2
    for k1 = 1 : N1
        de = Y(k2,k1);
        if de > MY
            MY = de;
        elseif de < mY
            mY = de;
        end
    end
end

if pnt % при необхідності - друк
    for k2 = N2:-1:1
        fprintf('%11.2e',Y(k2,:));
        fprintf('\n');
    end
end

[Mx1, Mx2] = meshgrid(X1, X2); % побудова сіток для 3D-графіків
% виведення графіка розв'язку Y
de = [X1(1), X1(N1), X2(1), X2(N2), mY-0.1, MY+0.1];
mesh(Mx1, Mx2, Y); axis(de);
shading interp, colormap cool;
xlabel('x1'), ylabel('x2'), zlabel(z1); title(ti1);
if s, pause(s);
else, pause;
end
close all

if nargin == 7
    return
end

```

```

% виведення графіка U
mesh(Mx1, Mx2, U); axis(de);
shading interp, colormap cool;
xlabel('x1'); ylabel('x2'); zlabel(z2); title(ti2);
if s, pause(s);
else, pause
end
close all
% виведення графіка Y - U
YU = Y - U;
C = norm(YU,inf);
fprintf('%s ||Y-U||c=%13.5e\n',txtp,C);
C = max(max(abs(YU)));
fprintf('max(max(|Y-U|)) =%13.5e\n',C);
mesh(Mx1, Mx2, YU); %view(-145,35);
shading interp, colormap jet;
xlabel('x1'); ylabel('x2'); zlabel(z3); title(ti3);
end

```

Практичне заняття № 12

ТЕМА № 7. Різницький метод розв'язання крайових задач для рівнянь еліптичного типу.

7.1. Наближений розв'язок лінійних КЗ для рівняння Пуассона в областях складених з прямокутників, трапецій, трикутників (1 год).

Теоретична інформація:

Досить часто прикладні крайові задачі для двовимірного рівняння Пуассона приходить розв'язувати в областях, які компонуються з геометричних фігур – прямокутників, трапецій, трикутників, причому всередині такої області можливі «вирізи» з таких-же фігур. Крім цього, крайові умови на певних частинах зовнішньої (можливо і внутрішньої) границі можуть буди різного роду – I, II і III. Розглянемо приклад такої задачі, як з точки зору побудови РЗ, так і зі сторони знаходження розв'язку за допомогою програми, яка реалізує ітераційний метод релаксації.

Знайти неперервну в області $\bar{D} = Sg_1 \setminus Sg_2 \setminus Tr$, $Sg_1 = \{(x_1, x_2) \in [0,1] \times [0,1]\}$, $Sg_2 = \{(x_1, x_2) \in [0, \frac{3}{10}] \times [0, \frac{3}{10}]\}$, $Tr = \{(x_1, x_2) \in (x_2 > 2 - 2x_1) \& (x_1 \leq 1) \& (x_2 \leq 1)\}$ функцію $u(x_1, x_2)$, яка справджує рівняння

$$\Delta u(x_1, x_2) = -f(x_1, x_2), \quad x \equiv (x_1, x_2) \in D, \quad (7.1)$$

і крайові умови на границі $\partial D = \partial D_1 \cup \partial D_2 \cup \partial D_3 \cup \partial D_4 \cup \partial D_5 \cup \partial D_6$:

$$\begin{aligned} u(x_1, x_2) &= \mu_{x11}(x_1, x_2), \quad (x_1, x_2) \in \partial D_1, \quad \partial D_1 = \{0 \leq x_1 < 0.3 \& x_2 = 0.3\}; \\ \frac{\partial u(x_1, x_2)}{\partial x_1} &= \mu_{y12}(x_1, x_2), \quad (x_1, x_2) \in \partial D_2, \quad \partial D_2 = \{x_1 = 0.3 \& 0 < x_2 < 0.3\}; \\ u(x_1, x_2) &= \mu_{x12}(x_1, x_2), \quad (x_1, x_2) \in \partial D_3, \quad \partial D_3 = \{0.3 \leq x_1 \leq 1 \& x_2 = 0\}; \\ u(x_1, x_2) &= \mu_{xy}(x_1, x_2), \quad (x_1, x_2) \in \partial D_4, \quad \partial D_4 = \{0.5 \leq x_1 \leq 1 \& x_2 = 2 - 2x_1\}; \\ u(x_1, x_2) &= \mu_{x21}(x_1, x_2), \quad (x_1, x_2) \in \partial D_5, \quad \partial D_5 = \{0 \leq x_1 < 0.5 \& x_2 = 1\}; \\ \frac{\partial u(x_1, x_2)}{\partial x_1} &= \mu_{y11}(x_1, x_2), \quad (x_1, x_2) \in \partial D_6, \quad \partial D_6 = \{x_1 = 0 \& 0.3 < x_2 < 1\}. \end{aligned} \quad (7.2)$$

Тут введено позначення $\Delta u = L_1 u + L_2 u$, $L_\alpha u = \frac{\partial^2 u}{\partial x_\alpha^2}$, $\alpha = 1, 2$,

а область D з границею ∂D зображена на Рис. 7.1

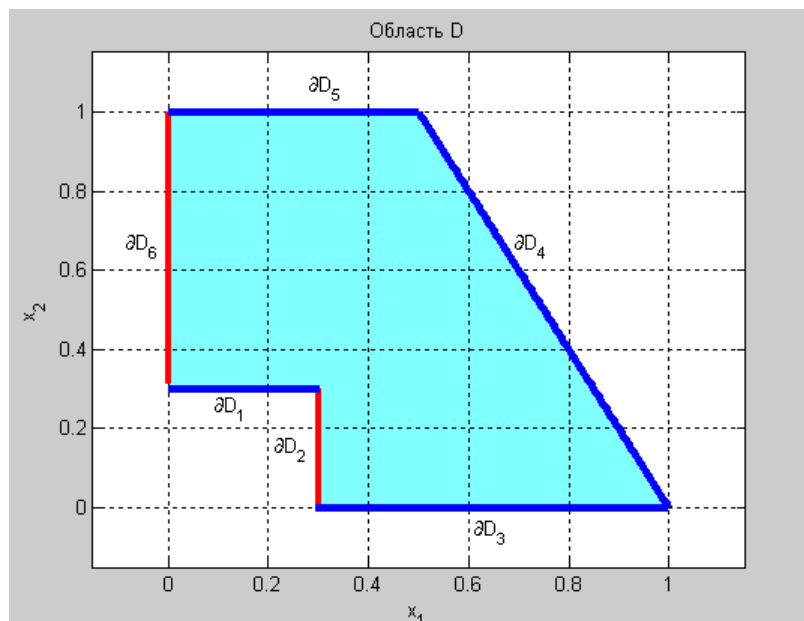


Рис.7.1 Область D з границею ∂D (на "**синій**" границі – КУ I роду, на "**червоній**" – КУ II роду).

Вважаємо, що при заданих функціях $f(x)$, $\mu(x)$, розв'язок поставленої задачі існує, єдиний і достатньо гладкий.

Введемо в $\bar{D}$ прямокутну рівномірну сітку $\bar{\omega}_h$

$$\bar{\omega}_h = \omega_h \cup \partial\omega_h = \bar{\omega}_{h_1} \times \bar{\omega}_{h_2}, \quad \bar{\omega}_{h_\alpha} = \{x_\alpha^{(k_\alpha)} = (k_\alpha - 1)h_\alpha, \quad k_\alpha = \overline{1, N_\alpha}, \quad l_\alpha = (N_\alpha - 1)h_\alpha\}, \quad \alpha = 1, 2,$$

таким чином, щоб на ∂D_4 точно розміщались вузли обох сіток $\bar{\omega}_{h_1}$ і $\bar{\omega}_{h_2}$ (для цього має виконуватись співвідношення $2h_1 = h_2$). Ця вимога стосується і границь ∂D_1 , ∂D_2 , адже вони розташовані «всередині» квадрата Sg_1 . Враховуючи, що $l_{1,2} = 1$, то оберемо $Nx \equiv N_1 = 201$, $Ny \equiv N_2 = 101$ і при значеннях $Hx \equiv h_1 = 5 \cdot 10^{-3}$, $Hu \equiv h_2 = 10^{-2}$ всі ці вимоги будуть виконані. Тут Nx, Ny, Hx, Hu – імена змінних, означених в програмі *p12a01*, аналогічно і в подальшому будемо використовувати відповідні позначення, наприклад, $Nx1 = N_1 - 1$, $Ny1 = N_2 - 1$, $i = k_1$, $j = k_2$, що стосується програмної реалізації цієї задачі.

На п'ятиточковому шаблоні $S = S(x) \cup S'(x)$, $S(x) = \{x_1^{(k_1)}, x_2^{(k_2)}\}$, $S'(x) = \{x_1^{(k_1 \pm 1)}, x_2^{(k_2 \pm 1)}\}$, задачі (7.1)–(7.2) поставимо у відповідність наступну різницеву задачу:

$$\Delta y(x) = -\phi(x), \quad x \in \omega_h; \quad (7.3)$$

$$y(x) = \mu(x), \quad x \in \partial D_1, \partial D_3, \partial D_4, \partial D_5; \quad (7.4)$$

$$\frac{y_{2,k_2} - y_{0,k_2}}{2h_1} = \mu((x_1)_1, (x_2)_{k_2}), \quad k_2 = \overline{Ny11, Ny-1}; \quad (7.5)$$

$$\frac{y_{Nx11+1,k_2} - y_{Nx11-1,k_2}}{2h_1} = \mu((x_1)_{Nx11}, (x_2)_{k_2}), \quad k_2 = \overline{2, Ny11-1}; \quad (7.6)$$

де позначено

$$\Delta y = \Delta_1 y + \Delta_2 y, \quad \Delta_\alpha y = y_{(\bar{x}_\alpha x_\alpha)k_\alpha}, \quad \alpha = 1, 2;$$

$$\phi(x) - f(x) = O(|h|^2), \quad |h|^2 = h_1^2 + h_2^2;$$

$$(Ny11-1) \cdot h_2 = 0.3; \quad (Nx11-1) \cdot h_1 = 0.3.$$

Сіткові функції y_{0,k_2} , y_{Nx11-1,k_2} в законтурних точках виключаються за допомогою рівняння (7.3), яке записуємо в точках границі, відповідно, ∂D_6 , ∂D_2 . Остаточний вид сіткових рівнянь (7.5), (7.6), після виключення законтурних точок, отримати і записати самостійно.

Якщо $f \in C^{(2)}(D)$, то $\phi(x) = f(x)$, $x \in \omega_h$ і для похибки апроксимації (7.3) отримаємо, що він співпадає з похибкою апроксимації диференціального оператора різнице-вим і має порядок $O(h_1^2 + h_2^2)$. Всі крайові умови I роду апроксимовані точно (враховуючи узгодженість сіток), а КУ II роду – апроксимовані з порядком $O(h_1^2)$ на розв'язку диференціальної задачі. Таким чином, порядок апроксимації РЗ (7.3)–(7.6), на розв'язку диференціальної задачі буде $O(h_1^2 + h_2^2)$.

Для лінійної РЗ (7.3)–(7.6) будемо мати стійкість по початковим даним і по правій частині, а також збіжність з порядком, який відповідає порядку апроксимації.

Для знаходження розв'язку різницевої задачі (7.3)–(7.6) використаємо ІМ верхньої релаксації. Ми уже використовували даний метод для модельної задачі Пуассона з КУ Діріхле в області $[0,1] \times [0,1]$ (див. Лекцію №11). Всі розглянуті там

деталі, як самого алгоритму, так і його програмної реалізації, будуть справедливі і для нашої задачі (7.3)–(7.6). Звичайно, дещо потрібно врахувати додатково.

По-перше, це геометрія області (див. Рис.7.1). Ця особливість (виріз у вигляді квадрату Sg_2 і трикутника Tr), легко вирішується при програмній реалізації. А саме, вибір початкового і кінцевого значення індекса i у вигляді функцій $i = Beg(j)$, $i = End(j)$ при зміні індекса $j = \overline{2, Ny1}$. Ми увели ці функції формально, реально в програмі *p12a01* їх опису не буде, враховуючи, що

$$Beg(j) = \begin{cases} Nx11, & j = \overline{2, Ny11}; \\ 1, & j = \overline{Ny11+1, Ny1}. \end{cases} \quad End(j) = Nx - (j - 1), \quad j = \overline{2, Ny1}.$$

По-друге, це наявність КУ I роду на границі ∂D_1 і КУ II роду на границях ∂D_2 , ∂D_6 (див. Рис.7.1). Ця особливість також вирішується в процесі програмної реалізації. Весь діапазон $\overline{2, Ny1}$ неперервної зміни індекса j (згідно алгоритму, розглянутому в Лекції №11) розділимо на три частини, з урахуванням границь $\partial D_2, \partial D_1, \partial D_6$:

1. $j = \overline{2, Ny11-1}$, з окремим розрахунком крайової умови (7.6) при $i = Nx11$ і виконанням розрахунків для $i = [Beg(j), End(j)]$;
2. $j = Ny11$, з виконанням розрахунків для $i = [Beg(j), End(j)]$;
3. $j = \overline{Ny11+1, Ny1}$, з окремим розрахунком крайової умови (7.5) при $i = 1$ і виконанням розрахунків для $i = [Beg(j), End(j)]$;

По-третє, розрахунок початкового наближення в ІМ. Замінімо лінійну інтерполяцію на присвоєння значення 0 усім внутрішнім точкам сіткової функції, тобто, $y^{(0)}(x) = 0$, $x \in \omega_{\bar{h}}$ (при бажанні, можна реалізувати інтерполювання по «вертикалям»).

І останнє зауваження. Враховуючи, що для зберігання даних будемо використовувати матрицю $U = (U_{j,i})_{j,i=1}^{Ny, Nx}$, то ту її частину, яка по індексам (j,i) належить областям Sg_2 і Tr , заповнимо значенням 0 і залишимо без змін (див. графік результату).

Аудиторне завдання:

п.1. Знайти чисельний розв'язок задачі (7.1)–(7.2) за наступних значень вхідних даних:

$$\begin{aligned} f(x_1, x_2) &= 30(x_1(1-x_1) + x_2(1-x_2)), \quad \mu_{x11}(x_1) = 3.15x_1(1-x_1), \quad \mu_{y12}(x_2) = 6x_2(1-x_2), \\ \mu_{x12}(x_1) &= 0, \quad \mu_{xy}(x_1) = 30x_1(1-x_1)^2(2x_1-1), \quad \mu_{x21}(x_1) = 0, \quad \mu_{y11}(x_2) = 15x_2(1-x_2) \end{aligned} \quad (7.7)$$

і порівняти наближений розв'язок із точним розв'язком $u(x_1, x_2) = 15x_1(1-x_1)x_2(1-x_2)$.

Вивести графіки:

- $(U_{j,i})$ – наближеного розв'язку;
- $(UU_{j,i}) = u(x_{1i}, x_{2j})$ – точного розв'язку;
- $(\delta_{j,i}) = U - UU$ – похибки;

і значення характеристик розрахунку:

- кількості затрачених ітерацій;
- час розрахунку;
- $\|(\delta_{j,i})\|_{hC}$ – похибка в сітковій нормі C ;
- $\max_{i,j} \|(\delta_{j,i})\|$ – максимальне відхилення наближеного від точного розв'язку.

Враховуючи геометрію області $\bar{D}$ і крайових умов (7.2), немає смислу писати універсальне ПЗ, тому створимо відповідну М-функцію *p12a01* «під цю» задачу :

```
function p12a01
clear all, close all, clc
% п.1 модельна задача :
% Розв'язування крайової задачі для рівняння Пуассона
% в області складеної з класичних геометричних елементів
% (прямокутників, трапецій, трикутників)

% параметри задачі і "опис" геометрії області :
a = 1; b = 1;
Nx = 201; Ny = 101; % кількість вузлів узгодженої сітки
Nx1 = Nx-1; Ny1 = Ny-1;
X = linspace(0, a, Nx); Y = linspace(0, b, Ny);
Hx = X(2)-X(1); Hy = Y(2)-Y(1);
Nx11 = 61; Nx21 = 101; Ny11 = 31;
% ітераційні параметри м.релаксації :
ro = 1.965; ep = 1e-8; itmax = 30000;
% допоміжні величини :
HxH = 1./(Hx.*Hx); HyH = 1./(Hy.*Hy);
C = ro.*0.5./(HxH+HyH); A = HxH.*C; B = HyH.*C; D = 1-ro;
A2 = 2.*A;
% формування крайових умов та початкового наближення :
el = clock();
U = zeros(Ny, Nx);
for i = 1 : Nx11 % D1 = {0<=x<0.3&0.3=y}
    U(Ny11,i) = mux11(X(i));
end
for i = Nx11 : Nx % D3 = {0.3<=x<=1&0=y}
    U(1,i) = mux12(X(i));
end
for i = 1 : Nx21 % D5 = {0<=x<0.5&1=y}
    U(Ny,i) = mux21(X(i));
end
for j = 2 : Ny1 % D4 = {0.5<=x<=1&y=2-2*x}
    i = Nx-j+1; U(j,i) = muxy(X(i));
end

it = 0; Fla = 0;
while ~Fla & it < itmax % ітер.процес
    Fla = 1; it = it+1;

    for j = 2 : Ny11-1 % x=0.3&0<y<0.3
        i = Nx11; u = U(j,i);
        UU = A2.*(U(j,i+1)-Hx.*muy12(Y(j))) + ...
            B.*(U(j-1,i)+U(j+1,i)) + D.*u + C.*f(X(i),Y(j));
        if Fla
            Fla = abs(UU-u) < ep;
        end
        U(j,i) = UU;
        for i = Nx11+1 : Nx-j % 0.3<x<1&0<y<0.3&y=2-2x
            u = U(j,i);
            UU = A.*(U(j,i-1)+U(j,i+1)) + ...
                B.*(U(j-1,i)+U(j+1,i)) + D.*u + C.*f(X(i),Y(j));
            if Fla
                Fla = abs(UU-u) < ep;
            end
            U(j,i) = UU;
        end
    end
end
```

```

    end
end

j = Ny11;
for i = Nx11+1 : Nx-j          % 0.3<x<1&y=0.3&y=2-2x
    u = U(j,i);
    UU = A.*(U(j,i-1)+U(j,i+1)) + ...
        B.*(U(j-1,i)+U(j+1,i)) + D.*u + C.*f(X(i),Y(j));
    if Fla
        Fla = abs(UU-u) < ep;
    end
    U(j,i) = UU;
end

for j = Ny11+1 : Ny1;          % 0=x&0.3<y<1&y=2-2x
    i = 1; u = U(j,i);
    UU = A2.*(U(j,i+1) - Hx.*muy11(Y(j))) + ...
        B.*(U(j-1,i)+U(j+1,i)) + D.*u + C.*f(X(i),Y(j));
    if Fla
        Fla = abs(UU-u) < ep;
    end
    U(j,i) = UU;
    for i = 2 : Nx-j          % 0<x<1&0.3<y<1&y=2-2x
        u = U(j,i);
        UU = A.*(U(j,i-1)+U(j,i+1)) + ...
            B.*(U(j-1,i)+U(j+1,i)) + D.*u + C.*f(X(i),Y(j));
        if Fla
            Fla = abs(UU-u) < ep;
        end
        U(j,i) = UU;
    end
end

end

e2 = clock(); u = etime(e2,e1);
fprintf('Виконано ітерацій =%5u\n', it);
fprintf('Тривалість розрах.=%9.3f сек\n', u);

% побудова графіку розв'язку :
u = @(x,y) (15.*x.*(1-x).*y.*(1-y)); % точний розв'язок
UU = zeros(Ny,Nx);
for j = 1 : Ny
    for i = 1 : Nx
        if (j >= Ny11 | i >= Nx11) & i <= Nx-j+1
            UU(j,i) = u(X(i),Y(j));
        end
    end
end

end
[Mx,My] = meshgrid(X,Y); clear X Y u
u = 5e-4; e1 = [-u,a+u,-u,b+u,-u,l+u];
surfl(Mx, My, U); axis(e1);
e2 = 'Різнищевий розв'язок';
title(e2); xlabel('x'); ylabel('y'); zlabel('U');
shading interp; colormap(pink); view(40,20);
pause; close all;
contour(Mx, My, U, 15);
xlabel('x'); ylabel('y'); title(e2); colormap hsv;
pause; close all;

```

```

surfl(Mx, My, UU); axis(e1);
e2 = 'Точний розв'язок';
title(e2); xlabel('x'); ylabel('y'); zlabel('u');
shading interp; colormap(pink); view(40,20);
pause; close all;
contour(Mx, My, UU, 15);
xlabel('x'); ylabel('y'); title(e2); colormap hsv;
pause; close all;

UU = U - UU;
surfl(Mx, My, UU);
e2 = 'Похибка U-u';
title(e2); xlabel('x'); ylabel('y'); zlabel('U-u');
shading interp; colormap(pink); view(40,20);
e1 = norm(UU,inf); fprintf([e2,' =%13.5e\n'], e1);
e1 = max(max(abs(UU))); fprintf('max(max|U-u|)=%13.5e\n', e1);
end

function ff = f(x, y)
% функція f(x,y)
ff = 30.*(x.*(1 - x) + y.*(1 - y));
end

function mu = mux11(x)
% крайові умови на D1 = {(0<=x<=0.3) & (y=0.3)}
mu = 3.15.*x.*(1 - x);
end

function mu = mux12(x)
% крайові умови на D3 = {(0.3<=x<=1) & (y=0)}
mu = 0;
end

function mu = mux21(x)
% крайові умови на D5 = {(0<=x<=0.5) & (y=1)}
mu = 0;
end

function mu = muxy(x)
% крайові умови на D4 = {(0.5<=x<=1) & (y=2-2x)}
xx = 2.*x;
mu = 15.*xx.*(1 - x).^2.*(xx - 1);
end

function mu = muy11(y)
% крайові умови на D6 = {(0.3<y<1) & (x=0)}
mu = 15.*y.*(1 - y);
end

function mu = muy12(y)
% крайові умови на D2 = {(0<y<0.3) & (x=0.3)}
mu = 6.*y.*(1 - y);
end

```

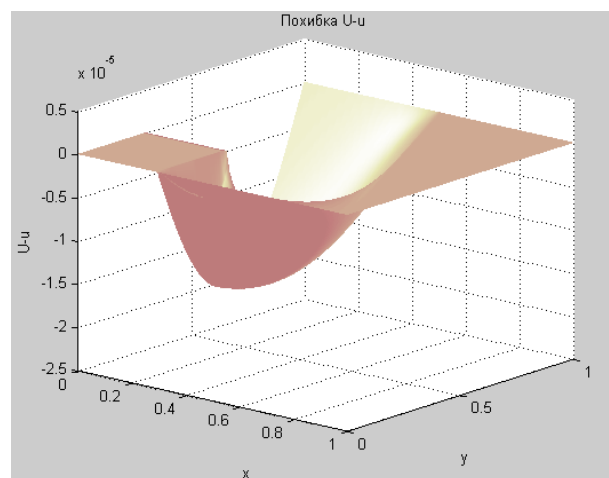
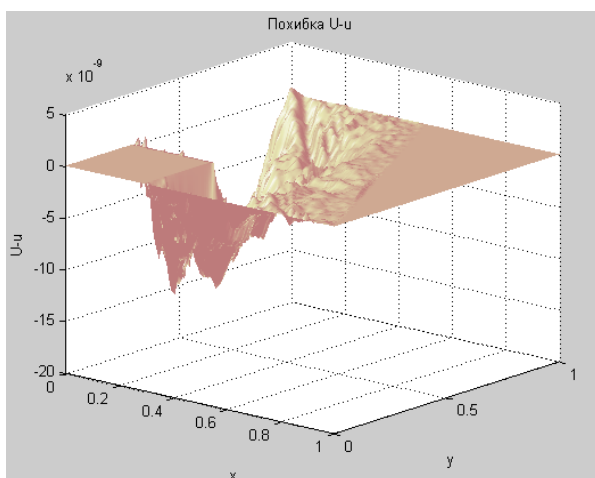
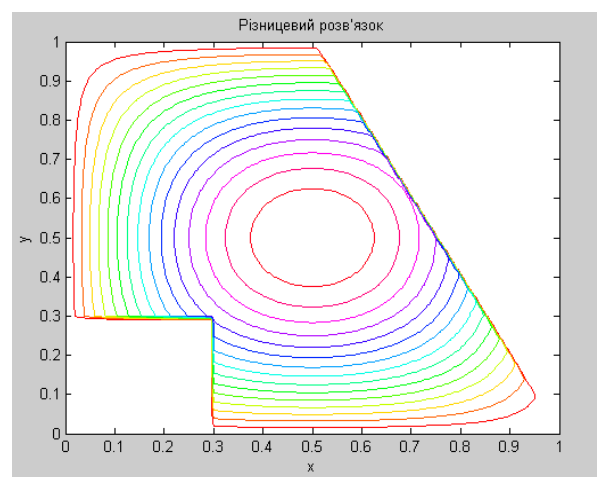
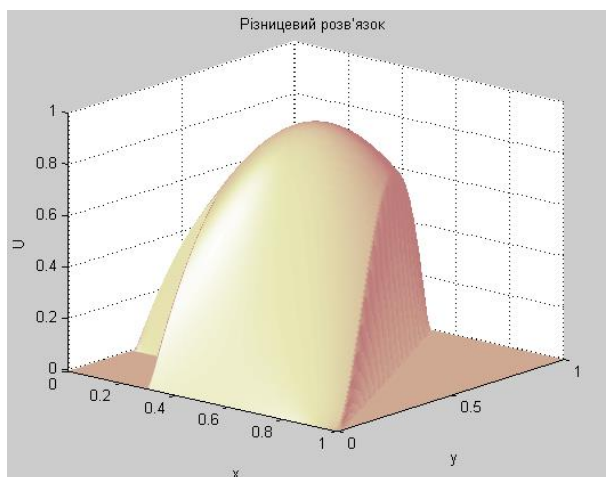
В результаті виконання цієї функції отримано наступний результат (точний розв'язок тут не приводимо з огляду на величину похибки /лівий рисунок/):

```

Виконано ітерацій = 562
Тривалість розрах.= 29.719 сек

```

Похибка $U-u = 7.52836e-007$
 $\max(\max|U-u|) = 1.66109e-008$



м.релаксації

м.Зейделя

Враховуючи, що $\max(h_1, h_2) = h_2 = 10^{-2}$, а отже, теоретично, має бути похибка $\approx 10^{-4}$, то ми отримали досить непоганий результат. Якщо використати ітераційний метод Зейделя, то маємо такі значення характеристик розрахунку

Виконано ітерацій = 22111
 Тривалість розрах. = 1231.844 сек
 Похибка $U-u = 1.86601e-003$
 $\max(\max|U-u|) = 2.06408e-005$

і графік похибки /правий рисунок/. Зрозуміло, що графік наближеного розв'язку візуально не відрізняється від отриманого методом релаксації.

Завдання для самостійної роботи:

п.1. Знайти чисельний розв'язок задачі (7.1)–(7.2) за наступних значень вхідних даних:

$$\begin{aligned} \mu_{x11}(x_1) &= 0, \quad \mu_{y12}(x_2) = 0, \\ \mu_{x12}(x_1) &= 0, \quad \mu_{xy}(x_1, x_2) = 0, \quad \mu_{x21}(x_1) = 0, \quad \mu_{y11}(x_2) = 0 \end{aligned} \quad (7.8)$$

Функція $f(x_1, x_2)$ означена в Таблиці 7.1.

Вивести графік наближеного розв'язку і значення характеристик розрахунку :

- кількість затрачених ітерацій;
- час розрахунку.

7.2. Ітераційні методи розв'язання різницьових задач Діріхле в прямокутнику для рівняння Пуассона (1 год).

Теоретична інформація:

З матеріалів Лекції №12 ми знаємо, що крім простих ІМ з повільною збіжністю існують і більш ефективні ІМ, побудовані на *обчисленнях на усталювання*, коли за розв'язок крайової задачі для еліптичного рівняння беруть розв'язок при $t \rightarrow \infty$ мішаної задачі для параболічного рівняння з тим же просторовим оператором, співпадаючими крайовими умовами і довільною початковою умовою.

Маємо рівняння Пуассона, задане в деякій прямокутній області :

$$\Delta u(x_1, x_2) \equiv \frac{\partial^2 u(x_1, x_2)}{\partial x_1^2} + \frac{\partial^2 u(x_1, x_2)}{\partial x_2^2} = -f(x_1, x_2), \quad x \equiv (x_1, x_2) \in D; \quad (7.9)$$

$$\bar{D} = \{0 \leq x_\alpha \leq l_\alpha, \alpha = 1, 2\}$$

де на «горизонтальних» і «вертикальних» границях прямокутника $\partial D = \partial D_H \cup \partial D_V$ задані крайові умови Діріхле

$$\begin{aligned} u(x_1, x_2) &= \mu_H(x_1) \equiv [\mu_H(x_1, 0), \mu_H(x_1, l_2)], \quad x \in \partial D_H = \{0 \leq x_1 \leq l_1 \text{ \& } (x_2 = 0 \mid x_2 = l_2)\}; \\ u(x_1, x_2) &= \mu_V(x_2) \equiv [\mu_V(0, x_2), \mu_V(l_1, x_2)], \quad x \in \partial D_V = \{0 < x_2 < l_2 \text{ \& } (x_1 = 0 \mid x_1 = l_1)\}. \end{aligned} \quad (7.10)$$

Для наближеного розв'язання (7.9), (7.10) використовуємо метод обчислення на усталювання, який складається з декількох етапів.

На першому етапі перехід до наступної диференціальної задачі :

$$\begin{cases} \frac{\partial w(x_1, x_2, t)}{\partial t} = \Delta w(x_1, x_2, t) + f(x_1, x_2), \quad x \equiv (x_1, x_2) \in D, \quad t > 0; \\ w(x, t) = \mu_H(x), \quad x \in \partial D_H; \\ w(x, t) = \mu_V(x), \quad x \in \partial D_V; \\ w(x, 0) = w_0(x), \quad x \in D, \quad \forall w_0(x). \end{cases} \quad (7.11)$$

На другому етапі будуємо для (7.11) різницеву задачу. В залежності від того ітераційного процесу (ІП), який в подальшому буде застосовано, РЗ буде мати свої особливості, окрім "*початкової умови*" :

$$y^{(0)}(x) = \begin{cases} \mu(x), \quad x \in \partial \omega_h^-; \\ \forall \text{ value}, \quad x \in \omega_h^-. \end{cases} \quad (7.12)$$

Для застосування *методу змінних напрямків* маємо дві різницьові задачі, які реалізують перехід $(j) \rightarrow (j + \frac{1}{2}) \rightarrow (j + 1)$:

$$\begin{cases} \frac{\bar{y} - y^{(j)}}{\tau_{j+1}^{(1)}} = \Lambda_1 \bar{y} + \Lambda_2 y^{(j)} + f(x), \quad x \in \omega_h^-, \\ \bar{y}(x) = \mu_H(x), \quad \bar{y}(x) = \mu_V(x), \quad x \in \partial \omega_h^-; \end{cases} \quad (7.13)$$

$$\begin{cases} \frac{\hat{y} - \bar{y}}{\tau_{j+1}^{(2)}} = \Lambda_1 \bar{y} + \Lambda_2 \hat{y} + f(x), \quad x \in \omega_h^-, \\ \hat{y}(x) = \mu_H(x), \quad \hat{y}(x) = \mu_V(x), \quad x \in \partial \omega_{h_2}^-; \end{cases} \quad (7.14)$$

Для застосування *позмінно-трикутного методу* будуємо спочатку чисто неявну різницеву схему і отримуємо таку РЗ, яка реалізує перехід $(j) \rightarrow (j + 1)$:

$$\begin{cases} \frac{\hat{y} - y}{\tau} = (\Lambda_1 + \Lambda_2) \hat{y} + f(x), \quad x \in \omega_h^-, \\ \hat{y}(x) = \mu_H(x), \quad \hat{y}(x) = \mu_V(x), \quad x \in \partial \omega_{h_2}^-, \end{cases} \quad (7.15)$$

а після нескладних перетворень в різницевому рівнянні, направлених на заміну «складної» для розрахунків матриці задачі (7.15) на факторизовану матрицю, отримуємо таку РЗ :

$$\begin{cases} \mathbf{B} \frac{\hat{y} - y}{\tau} = \Lambda y + f(x), & \mathbf{B} = (E + \tau R_1)(E + \tau R_2), \quad x \in \omega_{\bar{h}}, \\ \hat{y}(x) = \mu_H(x), \quad \hat{y}(x) = \mu_V(x), & x \in \partial\omega_{\bar{h}_2}; \end{cases} \quad (7.16)$$

де

$$R_1 y = \frac{1}{h_1^2} (y_{k_1, k_2} - y_{k_1-1, k_2}) + \frac{1}{h_2^2} (y_{k_1, k_2} - y_{k_1, k_2-1}), \quad R_2 y = \frac{1}{h_1^2} (y_{k_1, k_2} - y_{k_1+1, k_2}) + \frac{1}{h_2^2} (y_{k_1, k_2} - y_{k_1, k_2+1}).$$

На третьому етапі будемо відповідні ІМ :

- ✓ МЗН з вибором оптимальних ітераційних параметрів (по Жордану) для (7.13)-(7.14);
- ✓ ПТМ з чебишовським набором параметрів для (7.16).

Опис алгоритмів цих ІМ ми розглядали на лекції, тому не будемо його повторювати ще раз, а програмна реалізація – М-функції *ADM_optJ*, *ATM_Ch_ps* (див. Програмне забезпечення на MATLAB).

Аудиторне завдання:

п.2. Знайти розв'язок модельної задачі (7.9), (7.10) для таких вхідних даних:

$$(x_1, x_2) \in \bar{D} \equiv [0, 2] \times [0, 1];$$

$$f(x_1, x_2) = \pi^2 (\sin(\pi * x_1) + 4 \cos(2\pi * x_2));$$

$$\mu_H(x) = 4 + \sin(\pi * x_1), \quad x_1 \in [0, 2] \text{ \& } (x_2 = 0 | x_2 = 2),$$

$$\mu_V(x) = 3 + \cos(2\pi * x_2), \quad x_2 \in [0, 1] \text{ \& } (x_1 = 0 | x_1 = 1)$$

і порівняти наближений розв'язок із точним розв'язком

$$u(x_1, x_2) = 3 + \sin(\pi * x_1) + \cos(2\pi * x_2).$$

Виведення результатів організувати аналогічно п.1.

Створимо відповідну М-функцію *pr12a02*, в якій використовуються функції *u*, *f* – анонімні (опис точного розв'язку і правої частини рівняння Пуассона), *muh*, *muh* – внутрішні (опис крайових умов); *AMD_optJ*, *ATM_Ch_ps*, *viewU* – зовнішні (реалізації ІМ і візуалізація результатів наближеного, точного розв'язку, похибки і числових характеристик ІМ):

```
function pr12a02
clear all, close all, clc
% п.2 модельна задача :
% Задача Діріхле для р-я Пуассона в прямокутній області.

a = 2; b = 1;
u = @(x,y) (3+sin(pi.*x)+cos(2.*pi.*y));
f = @(x,y) (pi^2.*(sin(pi.*x)+4*cos(2*pi.*y)));
ep = 1e-6;
Nx = 201; Ny = 101; % кількість вузлів сітки
Nx1 = Nx-1; Ny1 = Ny-1;
X = linspace(0, a, Nx); Y = linspace(0, b, Ny);
Nx = X(2)-X(1); Ny = Y(2)-Y(1);

UU = zeros(Ny,Nx);
for j = 1 : Ny
    for i = 1 : Nx
        UU(j,i) = u(X(i),Y(j));
    end
end
end
```

```

t1 = clock();
[U, it] = ADM_optJ(f, @muh, @muv, X, Y, ep);
t = clock(); t = etime(t,t1);
fprintf('м.Змін.Напр. Nx=%4u Ny=%4u Hx=%7.5f Hy=%7.5f\n',Nx,Ny,Hx,Hx);
fprintf('Ітер. Час(сек) max(max|U-u|)\n');
t1 = max(max(abs(U-UU)));
fprintf('%5u%9.3f %13.5e\n', it, t, t1);
viewU(3, X, Y, U, 15, UU);

t1 = clock();
[U, it] = ATM_Ch_ps(f, @muh, @muv, X, Y, ep);
t = clock(); t = etime(t,t1);
fprintf('ПТМ з ЧНП Nx=%4u Ny=%4u Hx=%7.5f Hy=%7.5f\n',Nx,Ny,Hx,Hx);
fprintf('Ітер. Час(сек) max(max|U-u|)\n');
t1 = max(max(abs(U-UU)));
fprintf('%5u%9.3f %13.5e\n', it, t, t1);
viewU(3, X, Y, U, 15, UU);
end

function [g1, g2] = muh(x)
    g1 = 4+sin(pi*x); % y=0
    g2 = g1;          % y=1
end

function [g1, g2] = muv(y)
    g1 = 3+cos(2*pi*y); % x=0
    g2 = g1;          % x=2
end

```

Отримано наступний результат її виконання (показано графіки наближеного розв'язку, а точний розв'язок тут не приводимо з огляду на величину похибки) :

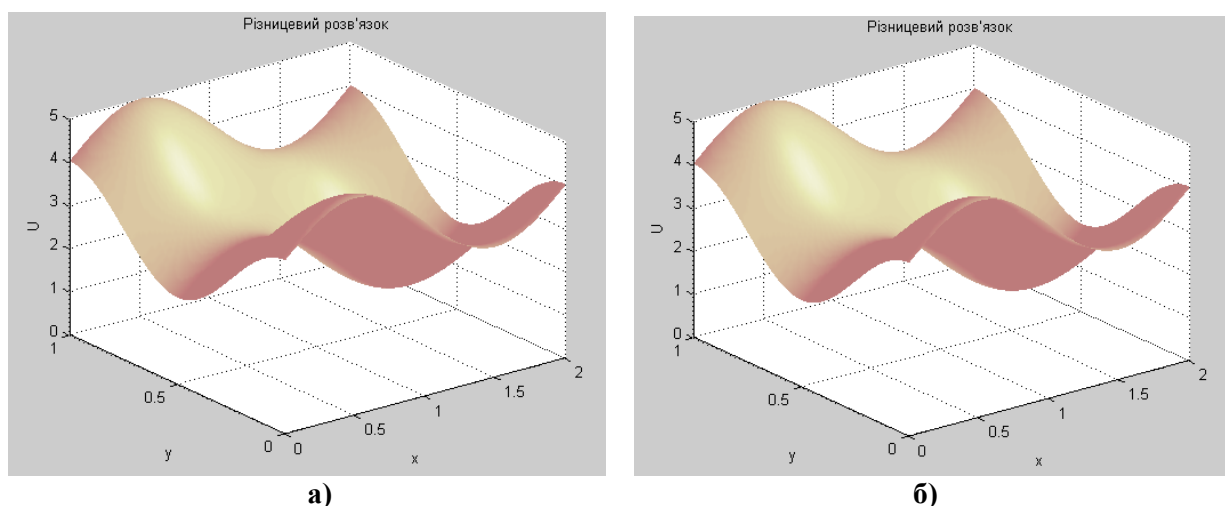


Рис.7.2 Наближені розв'язки модельної задачі: **а)** – МЗП; **б)** – ПТМ.

```

м.Змін.Напр. Nx= 201 Ny= 101 Hx=0.01000 Hy=0.01000
Ітер. Час(сек) max(max|U-u|)
    15   18.063   6.27228e-004
ПТМ з ЧНП Nx= 201 Ny= 101 Hx=0.01000 Hy=0.01000
Ітер. Час(сек) max(max|U-u|)
    116    7.469   6.29396e-004

```

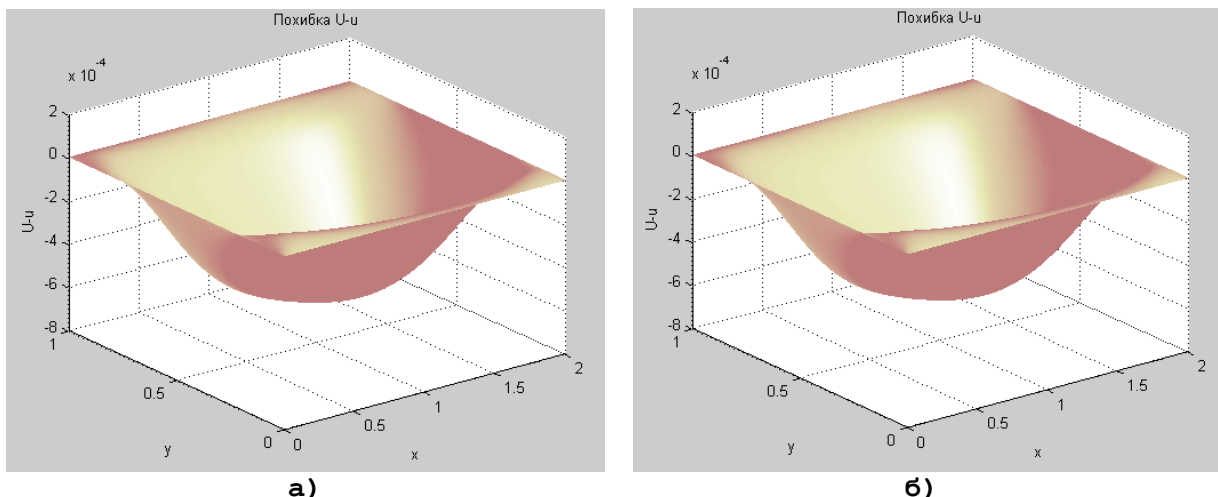


Рис.7.3 Похибка наближеного розв'язку модельної задачі: а) – МЗП; б) – ПТМ.

Графіки наближених розв'язків і похибок візуально співпадають для обох методів, аналогічно і числові характеристики значень $\max(\max |U-u|)$.

Завдання для самостійної роботи:

п.2. Використовуючи МЗП, ПТМ, знайти наближений розв'язок задачі (7.9), (7.10) для вхідних параметрів, указаних в Таблиці 7.1. Виведення результатів організувати аналогічно п.1.

Таблиця 7.1 Варіанти індивідуального завдання.

K	l_1	l_2	$f(x_1, x_2)$	$\mu_H(x_1) =$ $[\mu_H(x_1, 0), \mu_H(x_1, l_2)]$	$\mu_V(x_2) =$ $[\mu_V(0, x_2), \mu_V(l_1, x_2)]$
1	1	1	$10 \sin(\pi x_1 x_2 / 2)$	$[0, x_1]$	$[0, x_2]$
2	1	1	$10 \cos(\pi x_1 x_2 / 2)$	$[0, x_1]$	$[0, x_2]$
3	2	1	$8 \sin(2\pi x_1 x_2)$	$[x_1, 0]$	$[0, 2 - 2x_2]$
4	2	1	$8 \cos(2\pi x_1 x_2)$	$[x_1, 0]$	$[0, 2 - 2x_2]$
5	1	1	$\exp(-x_1 - x_2)$	$[0, x_1]$	$[0, x_2]$
6	1	1	$\exp(x_1 + x_2)$	$[0, x_1]$	$[0, x_2]$
7	2	1	$x_1 \exp(x_1 - x_2)$	$[x_1, 0]$	$[0, 2 - 2x_2]$
8	2	1	$x_2 \exp(x_1 + x_2)$	$[x_1, 0]$	$[0, 2 - 2x_2]$
9	1	2	$5 \sin(1.5\pi x_1 x_2)$	$[2 - x_1, 0]$	$[2 - x_2, 1 - x_2 / 2]$
10	1	2	$5 \cos(1.5\pi x_1 x_2)$	$[2 - x_1, 0]$	$[2 - x_2, 1 - x_2 / 2]$
11	1	2	$\exp(-x_1 - 2x_2)$	$[2 - x_1, 0]$	$[2 - x_2, 1 - x_2 / 2]$
12	1	2	$\exp(2x_1 + x_2)$	$[2 - x_1, 0]$	$[2 - x_2, 1 - x_2 / 2]$
13	1	2	$x_1 + \exp(x_1 - x_2)$	$[2 - x_1, 0]$	$[2 - x_2, 1 - x_2 / 2]$
14	1	2	$x_2 + \exp(x_1 + x_2)$	$[2 - x_1, 0]$	$[2 - x_2, 1 - x_2 / 2]$
15	2	2	$x_1 \exp(x_1 x_2)$	$[0, 2 - x_1]$	$[x_2, 0]$
16	2	2	$x_2 \exp(2 - x_1 x_2)$	$[0, 2 - x_1]$	$[x_2, 0]$
17	2	2	$2x_1 - \exp(x_2)$	$[0, 2 - x_1]$	$[x_2, 0]$
18	2	2	$\sqrt{x_1} + 2x_2$	$[0, 2 - x_1]$	$[x_2, 0]$
19	2	2	$x_2 \sqrt{10 - x_1 x_2}$	$[0, 2 - x_1]$	$[x_2, 0]$
20	2	2	$\sqrt{20 - x_1^2 - 2x_2^2}$	$[0, 2 - x_1]$	$[x_2, 0]$
21	1	1	$x_1 + 3\exp(-x_2)$	$[0, x_1]$	$[0, x_2]$
22	1	1	$\sqrt{x_1^2 + 2x_2 + 1}$	$[0, x_1]$	$[0, x_2]$

23	2	1	$5 \sin(\pi/2 * x_1) - x_2$	$[x_1, 0]$	$[0, 2 - 2x_2]$
24	2	1	$3 \cos(\pi/2 * x_2) + x_1 x_2$	$[x_1, 0]$	$[0, 2 - 2x_2]$
25	2	1	$x_1^2 + 2x_2 + \sqrt{2 + x_1}$	$[x_1, 0]$	$[0, 2 - 2x_2]$
26	2	1	$x_1 x_2 / \sqrt{10 - x_1 x_2}$	$[x_1, 0]$	$[0, 2 - 2x_2]$
27	2	2	$x_1 + x_2 + x_1 x_2$	$[0, 2 - x_1]$	$[x_2, 0]$
28	2	2	$x_1^2 + \sqrt{x_2} + x_1 x_2$	$[0, 2 - x_1]$	$[x_2, 0]$
29	1	2	$x_1 + x_2 + \sin(x_1 x_2)$	$[2 - x_1, 0]$	$[2 - x_2, 1 - x_2/2]$
30	1	2	$x_1 + x_2^3 + \exp(-x_1 x_2)$	$[2 - x_1, 0]$	$[2 - x_2, 1 - x_2/2]$

Програмне забезпечення на MATLAB:

```

function [U, n] = ADM_optJ(f, muh, muv, X, Y, ep)
% Розв'язування задачі Діріхле в прямокутнику
% D U dD = [0, X(end)] * [0, Y(end)]
% для р-я Пуассона за допомогою РС з використанням
% м. змінних напрямків (ADM) і вибором оптим.парам. по Жордану.
% u"xx + u"yy = -f(x,y), (x,y) є D,
% u(x,y) = muh(x), (x,y) є dD "по горизонталі"
% u(x,y) = muv(y), (x,y) є dD "по вертикалі"
% Вхідні дані :
% f - функція з описом f(x,y)
% muh - функція з описом muh(x)
% muv - функція з описом muv(y)
% X - сітка по x;
% Y - сітка по y;
% ep - точність (0<ep<1) ітераційного процесу
% Вихідні дані :
% U - сіткова функція наближеного розв'язку u(X,Y)
% n - кількість кроків ітераційного процесу

% параметри різницевої схеми:
Nx = length(X); Nx1 = Nx - 1; Hx = X(2) - X(1);
Ny = length(Y); Ny1 = Ny - 1; Hy = Y(2) - Y(1);
% формування граничних умов і початкового наближення :
U = zeros(Ny, Nx);
for k1 = 1 : Nx % КУ на "горизонталях" y=0, y=Y(Ny)
    [c1, c2] = muh(X(k1)); U(1,k1) = c1; U(Ny,k1) = c2;
end
for k2 = 2 : Ny1 % КУ на "вертикалях" x=0, x=X(Nx)
    [c1, c2] = muv(Y(k2));
    % + лінійна інтерполяція по горизонталі
    U(k2,:) = linspace(c1, c2, Nx);
end
clear c1 c2
% обчислення оптимальних ітераційних параметрів :
HHx = 1./(Hx.*Hx); HHy = 1./(Hy.*Hy);
p = 4.*HHx; ka = 4.*HHy; t = 0.5.*pi;
r = t.*Hx./X(Nx); t = t.*Hy./Y(Ny);
d1 = sin(r); d1 = p.*d1.*d1;
d2 = sin(t); d2 = ka.*d2.*d2;
D1 = cos(r); D1 = p.*D1.*D1;
D2 = cos(t); D2 = ka.*D2.*D2;
ka = (D1 - d1)./(D2 + d1); t = ka.*(D2 - d2);
t = sqrt(t./(D1 + d2)); eta = (1 - t)./(1 + t);

```

```

ka = ka.*D2./D1; p = (ka - t)./(ka + t);
r = D1 - D2 + (D1 + D2).*p;
r = 0.5.*r./(D1.*D2);
q = r + (1 - p)./D1;
n = fix(log(4./ep).*log(4./eta)./(pi.*pi));
tet = eta.*eta; tet = 0.0625.*(1 + 0.5.*tet).*tet;
ka = 2.*n; eta = 0.5 + tet;

% формування складових матриць СЛАР :
Ax = zeros(1,Nx); Cx = ones(1,Nx); Bx = Ax; Fx = Ax;
Ay = zeros(1,Ny); Cy = ones(1,Ny); By = Ay; Fy = Ay;

for j = 1 : n
% знаходження компоненти w ітераційних параметрів :
    t = (2.*j - 1)./ka; d1 = eta.*(1 + tet.^t);
    d2 = 1 + tet.^(1-t) + tet.^(1+t);
    d2 = tet.^(0.5.*t).*d2; w = d1./d2;
% знаходження наближеного розв'язку U для (j+1/2) :
    tau = (q.*w + r)./(1 + w.*p); UJ = U;
    d1 = tau.*HHx; d2 = 1 + 2.*d1;
    D1 = tau.*HHy; D2 = 1 - 2.*D1;
    for k1 = 2 : Nx1
        Ax(k1) = d1; Cx(k1) = d2; Bx(k1) = d1;
    end
    for k2 = 2 : Ny1
        y = Y(k2);
        Fx(1) = U(k2,1); Fx(Nx) = U(k2,Nx);
        for k1 = 2 : Nx1
            z = D1 .* (UJ(k2-1,k1) + UJ(k2+1,k1)) + D2.*UJ(k2,k1);
            Fx(k1) = z + tau.*f(X(k1),y);
        end
    end
% розв'язок СЛАР, знаходження U(k2,:) :
    [YY, alfa, err] = m_progm(Ax, Cx, Bx, Fx);
    U(k2,:) = YY;
end
clear YY alfa

% знаходження наближеного розв'язку U для (j+1) :
    tau = (q.*w - r)./(1 - w.*p); UJ = U;
    d1 = tau.*HHy; d2 = 1 + 2.*d1;
    D1 = tau.*HHx; D2 = 1 - 2.*D1;
    for k2 = 2 : Ny1
        Ay(k2) = d1; Cy(k2) = d2; By(k2) = d1;
    end
    for k1 = 2 : Nx1
        x = X(k1);
        Fy(1) = U(1,k1); Fy(Ny) = U(Ny,k1);
        for k2 = 2 : Ny1
            z = D1.*(UJ(k2,k1-1) + UJ(k2,k1+1)) + D2.*UJ(k2,k1);
            Fy(k2) = z + tau.*f(x,Y(k2));
        end
    end
% розв'язок СЛАР, знаходження U(:,k1) :
    [YY, alfa, err] = m_progm(Ay, Cy, By, Fy);
    U(:,k1) = YY';
end
clear YY alfa
end
return

```

```

function [U, n] = ATM_Ch_ps(f, muh, muv, X, Y, ep)
% Розв'язування задачі Діріхле в прямокутнику
%   D U dD = [0,X(end)]*[0,Y(end)]
% для р-я Пуассона за допомогою РС і використанням
% позмінно-трикутного м. з чебишовським набором параметрів.
%   u"xx + u"yy = -f(x,y), (x,y) ∈ D,
%   u(x,y) = muh(x), (x,y) ∈ dD "по горизонталі"
%   u(x,y) = muv(y), (x,y) ∈ dD "по вертикалі"
% Вхідні дані :
%   f - функція з описом f(x,y)
%   muh - функція з описом muh(x)
%   muv - функція з описом muv(y)
%   X - сітка по x;
%   Y - сітка по y;
%   ep - точність (0<ep<1) ітераційного процесу
% Вихідні дані :
%   U - сіткова функція наближеного розв'язку u(X,Y)
%   n - кількість кроків ітераційного процесу

% параметри різницевої схеми:
Nx = length(X); Nx1 = Nx - 1; Hx = X(2) - X(1);
Ny = length(Y); Ny1 = Ny - 1; Hy = Y(2) - Y(1);
% формування граничних умов і початкового наближення :
U = zeros(Ny, Nx); UJ = U;
for k1 = 1 : Nx           % КУ на "горизонталях" y=0, y=Y(Ny)
    [c1, c2] = muh(X(k1)); U(1,k1) = c1; U(Ny,k1) = c2;
end
for k2 = 2 : Ny1         % КУ на "вертикалях" x=0, x=X(Nx)
    [c1, c2] = muv(Y(k2));
    % + лінійна інтерполяція по горизонталі
    U(k2,:) = linspace(c1, c2, Nx);
end
clear c1 c2
% обчислення оптимальних ітераційних параметрів :
HHx = 1./(Hx.*Hx); HHy = 1./(Hy.*Hy);
p = 4.*HHx; ka = 4.*HHy; t = 0.5.*pi;
r = t.*Hx./X(Nx); t = t.*Hy./Y(Ny);
d = sin(r); d = p.*d.*d;
D = sin(t); D = ka.*D.*D; d = d + D; D = p + ka;
ka = d./D; d2 = sqrt(ka);
n = fix(0.5.*log(2./eps)./sqrt(2.*d2));
d1 = 0.5.*d./(1 + d2); d2 = 0.25.*d./d2;
w = 2./sqrt(d.*D); t = d1./d2;
r = (1 - t)./(1 + t); q = 2./(d1 + d2);
M = optchset(n); % стійкий набір множини ЧНП
d1 = w.*HHx; d2 = w.*HHy;
d = 1./(1 + d1 + d2); D = -2.*(HHx + HHy);

for j = 1 : n
    tau = q./(1 + r.*M(j)); % ітераційний параметр
    for k2 = 2 : Ny1
        y = Y(k2);
        for k1 = 2 : Nx1
            z = HHx.*(U(k2,k1-1) + U(k2,k1+1));
            p = HHy.*(U(k2-1,k1) + U(k2+1,k1));
            z = z + p + D.*U(k2,k1) + f(X(k1),y);
            p = d1.*UJ(k2,k1-1) + d2.*UJ(k2-1,k1);
            UJ(k2,k1) = (p + z).*d;
        end
    end
end

```

```

end
for k1 = Nx1 : -1 : 2
    x = X(k1);
    for k2 = Ny1 : -1 : 2
        z = d1.*UJ(k2,k1+1) + d2.*UJ(k2+1,k1);
        UJ(k2,k1) = (z + UJ(k2,k1)).*d;
    end
end
for k1 = 2 : Nx1
    for k2 = 2 : Ny1
        U(k2,k1) = U(k2,k1) + tau.*UJ(k2,k1);
    end
end
end
return

```

function T = optchset(n)

```

% Побудова множини оптимальних Чебишовських параметрів
% Вхідні дані:
% n - кількість чисел множини;
% Вихідні дані:
% T - оптимальний набір параметрів.

```

```

T = nchset(n); b = pi./(2.*n); T = -cos(b.*T);
return

```

function T = nchset(n)

```

% Побудова множини T(i) непарних чисел для
% оптимальних Чебишовських параметрів
% Вхідні дані:
% n - кількість чисел;
% Вихідні дані:
% T - числова множина.
P(1) = n; r = n; k = 1;
while r ~= 1
    k = k + 1;
    if mod(r, 2) == 0
        r = r./2;
    else
        r = r - 1;
    end
    P(k) = r;
end
T(1) = 1; kT = 1;
for j = k-1 : -1 : 1
    t = r; r = P(j);
    if 2.*t == r
        m = 2.*r;
        if j ~= 1
            if P(j-1) == r + 1
                m = m + 2;
            end
        end
    end
    for i = t : -1 : 1
        i2 = 2.*i; e = T(i);
        T(i2 - 1) = e; T(i2) = m - e;
    end
    kT = kT.*2;
else

```

```

    kT = kT + 1; T(kT) = r;
end
end
return

function viewU(s, X, Y, U, kc, UU)
[Mx, My] = meshgrid(X, Y); close all;
surfl(Mx, My, U);
e2 = 'Різницевий розв'язок';
title(e2); xlabel('x'); ylabel('y'); zlabel('U');
shading interp; colormap(pink); %view(40,20);
if s, pause(s);
else, pause;
end
close all;
if kc
    contour(Mx, My, U, kc);
    xlabel('x'); ylabel('y'); title(e2); colormap hsv;
    if s, pause(s);
    else, pause;
    end
    close all;
end
if nargin == 5
    return
end
surfl(Mx, My, UU);
e2 = 'Точний розв'язок';
title(e2); xlabel('x'); ylabel('y'); zlabel('u');
shading interp; colormap(pink); %view(40,20);
if s, pause(s);
else, pause;
end
close all;
if kc
    contour(Mx, My, UU, kc);
    xlabel('x'); ylabel('y'); title(e2); colormap hsv;
    if s, pause(s);
    else, pause;
    end
    close all;
end
Uu = U - UU;
surfl(Mx, My, Uu);
e2 = 'Похибка U-u';
title(e2); xlabel('x'); ylabel('y'); zlabel('U-u');
shading interp; colormap(pink); %view(40,20);
if s, pause(s);
else, pause;
end
end
end

```

Практичне заняття № 13

ТЕМА № 8. Різницевий метод розв'язання крайових задач для одновимірних лінійних рівнянь гіперболічного типу другого порядку (2 год).

Теоретична інформація:

Розглянемо в області $\bar{D} = D \cup \partial D = [a, b] \times [0, T]$ мішану задачу для рівняння гіперболічного типу :

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} - p(x, t)u(x, t) + f(x, t), \quad (x, t) \in D \equiv (a, b) \times (0, T], \quad (8.1)$$

з крайовими умовами I роду

$$u(a, t) = \mu_1(t), \quad u(b, t) = \mu_2(t), \quad t \in [0, T] \quad (8.2.1)$$

або II, III роду

$$-\frac{\partial u(a, t)}{\partial x} + \beta_1 u(a, t) = \mu_1(t), \quad \frac{\partial u(b, t)}{\partial x} + \beta_2 u(b, t) = \mu_2(t), \quad t \in [0, T] \quad (8.2.2)$$

і початковими умовами

$$u(x, 0) = u_0(x), \quad x \in [a, b], \quad (8.3)$$

$$\frac{\partial u(x, 0)}{\partial t} = u_1(x), \quad x \in [a, b] \quad (8.4)$$

Вважатимемо, що :

- початкові і крайові дані мають достатню гладкість і для них виконуються необхідні умови узгодження в точках $(a; 0)$, $(b; 0)$;
- функції p, f неперервні;
- шукана функція u також достатньо гладка,

а отже розв'язок (8.1)-(8.4) існує, єдиний і неперервно залежить від вхідних даних.

Побудова сімейства різницевих схем. Уведемо рівномірні сітки

$$\bar{\omega}_{h\tau} = \bar{\omega}_h \times \bar{\omega}_\tau; \quad \bar{\omega}_h = \{x_k = a + (k-1)h, \quad k = \overline{1, N}, \quad b - a = (N-1)h\};$$
$$\bar{\omega}_\tau = \{t_j = j\tau, \quad j = \overline{0, M}, \quad T = M\tau\},$$

і замінімо область $\bar{D}$ сітковою областю $\bar{\omega}_{h\tau}$, а також будемо використовувати наступні позначення

$$\dot{u} = \frac{\partial u}{\partial t}, \quad \ddot{u} = \frac{\partial^2 u}{\partial t^2}, \quad Lu = \frac{\partial^2 u}{\partial x^2}; \quad \dot{u} = Lu + f, \quad L\dot{u} = L^2 u + Lf, \quad L^2 u = L\ddot{u} - Lf;$$

$$y_{t,j} = \frac{\hat{y}_k - y_k}{\tau}, \quad y_{\bar{t},j} = \frac{y_{t,j} - y_{\bar{t},j}}{\tau} = \frac{\hat{y}_k - 2y_k - \check{y}_k}{\tau^2}; \quad \Lambda y = y_{\bar{x}x,k};$$

$$0 \leq \sigma \leq \frac{1}{2}; \quad y^{(\sigma)} \equiv Y = \sigma \hat{y} + (1-2\sigma)y + \sigma \check{y} = y + \sigma \tau^2 y_{\bar{t}t,j}.$$

На дев'ятиточковому шаблоні $S = \{(x_{k+1}, t_{j-1}), (x_k, t_{j-1}), (x_{k+1}, t_j), (x_k, t_j), (x_{k+1}, t_{j+1}), (x_k, t_{j+1})\}$ рівнянню (8.1) поставимо у відповідність однопараметричну сім'ю тришарових схем

$$y_{\bar{t}t,j} = \Lambda Y - p(x_k, t_j)Y + f(x_k, t_j), \quad (8.5)$$
$$k = \overline{2, N-1}; \quad j = \overline{1, M-1}.$$

Крайові умови (8.2.1) і початкова умова (8.3) апроксимуються точно

$$y_1^j = \mu_1(t_j), \quad y_N^j = \mu_2(t_j), \quad j = \overline{2, M}; \quad (8.6.1)$$

$$y_k^0 = u_0(x_k), \quad k = \overline{1, N}. \quad (8.7)$$

В Лекції №13 ми розглянули два варіанта отримання апроксимації умови (8.4) з другим порядком точності, наприклад, у вигляді наступного рівняння (порядок $O(\tau^2)$):

$$y_k^1 = y_k^0 + \tau * u_1(x_k) + 0.5\tau^2 (u_0''(x_k) - p(x_k, 0)y_k^0 + f(x_k, 0)), \quad k = \overline{1, N}. \quad (8.8)$$

Для апроксимації з другим порядком точності КУ умов II, III роду (8.2.2), використаємо метод законтурних точок. Спочатку отримаємо для точки $x=a$:

$$\begin{cases} -\frac{Y_2 - Y_0}{2h} + \beta_1 Y_1 = \mu_1(t_j), & O(h^2), \\ \frac{\hat{y}_1 - 2y_1 + \tilde{y}_1}{\tau^2} = \frac{Y_0 - 2Y_1 + Y_2}{h^2} - p(x_1, t_j)Y_1 + f(x_1, t_j), & O(\tau^2 + h^2). \end{cases}$$

Виключимо з цієї системи Y_0 і після ланцюжка нескладних перетворень

1. $Y_0 = Y_2 - 2h\beta_1 Y_1 + 2h\mu_1(t_j)$
2. $\frac{\hat{y}_1 - 2y_1 + \tilde{y}_1}{\tau^2} = \frac{(Y_2 - 2h\beta_1 Y_1 + 2h\mu_1(t_j)) - 2Y_1 + Y_2}{h^2} - p(x_1, t_j)Y_1 + f(x_1, t_j)$
3. $\hat{y}_1 - 2y_1 + \tilde{y}_1 = \tilde{\gamma} * (2Y_2 - (2 + 2h\beta_1)Y_1 + 2h\mu_1(t_j)) - \tau^2 p(x_1, t_j)Y_1 + \tau^2 f(x_1, t_j), \quad \tilde{\gamma} = (\tau/h)^2;$
4. $\hat{y}_1 - 2y_1 + \tilde{y}_1 = -[2\tilde{\gamma}(1 + h\beta_1) + \tau^2 p(x_1, t_j)]Y_1 + 2\tilde{\gamma}Y_2 + \tau^2 f(x_1, t_j) + 2h\tilde{\gamma}\mu_1(t_j),$

отримаємо різницеве рівняння

$$\begin{aligned} -\hat{y}_1 - (2\tilde{\gamma}(1 + h\beta_1) + \tau^2 p(x_1, t_j))Y_1 + 2\tilde{\gamma}Y_2 = \\ = -(2y_1 - \tilde{y}_1 + \tau^2 f(x_1, t_j) + 2h\tilde{\gamma}\mu_1(t_j)), \quad j = \overline{1, M-1} \end{aligned} \quad (8.6.2a)$$

Аналогічно отримаємо апроксимацію умови (8.2.2) для точки $x=b$:

$$\begin{cases} \frac{Y_{N+1} - Y_{N-1}}{2h} + \beta_2 Y_N = \mu_2(t_j), & O(h^2), \\ \frac{\hat{y}_N - 2y_N + \tilde{y}_N}{\tau^2} = \frac{Y_{N-1} - 2Y_N + Y_{N+1}}{h^2} - p(x_N, t_j)Y_N + f(x_N, t_j), & O(\tau^2 + h^2). \end{cases}$$

Виключимо з цієї системи Y_{N+1} і після ланцюжка нескладних перетворень :

1. $Y_{N+1} = Y_{N-1} - 2h\beta_2 Y_N + 2h\mu_2(t_j)$
2. $\frac{\hat{y}_N - 2y_N + \tilde{y}_N}{\tau^2} = \frac{Y_{N-1} - 2Y_N + (Y_{N-1} - 2h\beta_2 Y_N + 2h\mu_2(t_j))}{h^2} - p(x_N, t_j)Y_N + f(x_N, t_j)$
3. $\hat{y}_N - 2y_N + \tilde{y}_N = \tilde{\gamma} * [2Y_{N-1} - 2(1 + h\beta_2)Y_N + 2h\mu_2(t_j)] - \tau^2 p(x_N, t_j)Y_N + \tau^2 f(x_N, t_j)$
4. $\hat{y}_N - 2y_N + \tilde{y}_N = 2\tilde{\gamma} * Y_{N-1} - (2\tilde{\gamma}(1 + h\beta_2) + \tau^2 p(x_N, t_j))Y_N + \tau^2 f(x_N, t_j) + 2h\tilde{\gamma}\mu_2(t_j),$

отримаємо різницеве рівняння

$$\begin{aligned} 2\tilde{\gamma} * Y_{N-1} - \hat{y}_N - (2\tilde{\gamma}(1 + h\beta_2) + \tau^2 p(x_N, t_j))Y_N = \\ = -(2y_N - \tilde{y}_N + \tau^2 f(x_N, t_j) + 2h\tilde{\gamma}\mu_2(t_j)), \quad j = \overline{1, M-1} \end{aligned} \quad (8.6.2b)$$

Різницеві рівняння (8.5), (8.6.2) ще не мають виду розрахункових формул, адже вони залежать від вибору параметра σ , як і точність та стійкість отриманої РЗ (див. Лекцію №13).

З урахуванням частинних випадків: *явна схема* ($\sigma = 0$), *неявна схема* ($0 < \sigma \leq \frac{1}{2}$) опишемо розрахункові алгоритми, які і перетворимо у відповідні програмні реалізації.

Враховуючи, що при програмній реалізації потрібно розрізняти вид КУ, то введемо для цього параметри va , vb . Домовимось, що значення $va = 0$ відповідає варіанту КУ I роду в точці $x=a$ і $va = 1$ відповідає варіанту КУ II, III роду в точці $x=a$. Аналогічно зі значеннями параметра vb для точки $x=b$.

Алгоритм розрахунку для явної схеми :

4. Для $j=0$ визначаємо сіткову функцію $\{y_k^0\}_{k=1}^N$, згідно формули (8.7)
5. Для $j=1$ знаходимо $\{y_k^1\}_{k=1}^N$ по формулі (8.8).

6. Циклічно, по номерах $j = \overline{1, M-1}$ шарів сітки $\overline{\omega}_\tau$, виконуємо розрахунки

3.1. в граничній точці $x = x_1 \equiv a$ для $k = 1$

$$\begin{aligned} y_1^{j+1} &= \begin{cases} \mu_1(t_{j+1}), & va = 0; \\ \tilde{\beta}_1 * y_1^j + 2\tilde{\gamma} * y_2^j - y_1^{j-1} + \tau^2 f(x_1, t_j) + \tilde{\mu}_1, & va = 1 \end{cases} \\ \tilde{\beta}_1 &= \tilde{p}_1 - 2\tilde{\gamma}h\beta_1; \quad \tilde{\mu}_1 = 2\tilde{\gamma}h\mu_1(t_j) \\ \tilde{p}_k &= 2(1 - \tilde{\gamma}) - \tau^2 p(x_k, t_j) \end{aligned} \quad (8.9.a)$$

3.2. у внутрішніх точках сітки $\overline{\omega}_h$ для $k = \overline{2, N-1}$ по формулі

$$y_k^{j+1} = \tilde{\gamma} * (y_{k-1}^j + y_{k+1}^j) + \tilde{p}_k * y_k^j - y_k^{j-1} + \tau^2 f(x_k, t_j) \quad (8.9)$$

3.3. в граничній точці $x = x_N \equiv b$ для $k = N$

$$\begin{aligned} y_N^{j+1} &= \begin{cases} \mu_2(t_{j+1}), & vb = 0; \\ 2\tilde{\gamma} * y_{N-1}^j + \tilde{\beta}_2 * y_N^j - y_N^{j-1} + \tau^2 f(x_N, t_j) + \tilde{\mu}_2, & vb = 1 \end{cases} \\ \tilde{\beta}_2 &= \tilde{p}_N - 2\tilde{\gamma}h\beta_2; \quad \tilde{\mu}_2 = 2\tilde{\gamma}h\mu_2(t_j) \end{aligned} \quad (8.9.b)$$

Алгоритм розрахунку для неявної схеми :

1. Для $j = 0$ визначаємо сіткову функцію $\{y_k^0\}_{k=1}^N$, згідно формули (8.7)
2. Для $j = 1$ знаходимо $\{y_k^1\}_{k=1}^N$ по формулі (8.8).
3. Циклічно, по номерах $j = \overline{1, M-1}$ шарів сітки $\overline{\omega}_\tau$, формуємо СЛАР:

для $k = 1$

$$\begin{aligned} \begin{cases} y_1^{j+1} = \mu_1(t_{j+1}), & va = 0; \\ -C_1 * y_1^{j+1} + B_1 * y_2^{j+1} = -F_1, & va = 1; \end{cases} \\ \bar{\gamma} = \sigma\tilde{\gamma}; \quad \tilde{\sigma} = 1 - 2\sigma; \quad \tilde{p}_k = 2 - 2\tilde{\sigma}\tilde{\gamma} - \tilde{\sigma}\tau^2 p(x_k, t_j) \\ \bar{\beta}_1 = 2\tilde{\gamma}h\beta_1; \quad \tilde{\beta}_1 = 2\tilde{\sigma}\tilde{\gamma}h\beta_1; \quad \tilde{\mu}_1 = 2\tilde{\gamma}h\mu_1(t_j) \\ C_1 = 1 + 2\bar{\gamma} + \sigma\tau^2 p(x_1, t_j) + \bar{\beta}_1, \quad B_1 = 2\bar{\gamma}, \\ F_1 = 2\tilde{\sigma}\tilde{\gamma} * y_2^j + (\tilde{p}_1 - \tilde{\beta}_1) * y_1^j + 2\bar{\gamma} * y_2^{j-1} - C_1 * y_1^{j-1} + \tau^2 f(x_1, t_j) + \tilde{\mu}_1 \end{aligned} \quad (8.10)$$

для $k = \overline{2, N-1}$

$$\begin{aligned} A_k * y_{k-1}^{j+1} - C_k * y_k^{j+1} + B_k * y_{k+1}^{j+1} &= -F_k; \\ A_k = B_k = \bar{\gamma}, \quad C_k &= 1 + 2\bar{\gamma} + \sigma\tau^2 p(x_k, t_j) \\ F_k &= \bar{\gamma}(y_{k-1}^{j-1} + y_{k+1}^{j-1}) - C_k * y_k^{j-1} + \tilde{\sigma}\tilde{\gamma}(y_{k-1}^j + y_{k+1}^j) + \tilde{p}_k * y_k^j + \tau^2 f(x_k, t_j) \end{aligned} \quad (8.11)$$

для $k = N$

$$\begin{aligned} \begin{cases} y_N^{j+1} = \mu_2(t_{j+1}), & vb = 0; \\ A_N * y_{N-1}^{j+1} - C_N * y_N^{j+1} = -F_N, & vb = 1. \end{cases} \\ \bar{\beta}_2 = 2\tilde{\gamma}h\beta_2; \quad \tilde{\beta}_2 = 2\tilde{\sigma}\tilde{\gamma}h\beta_2; \quad \tilde{\mu}_2 = 2\tilde{\gamma}h\mu_2(t_j) \\ A_N = 2\bar{\gamma}, \quad C_N &= 1 + 2\bar{\gamma} + \sigma\tau^2 p(x_N, t_j) + \bar{\beta}_2, \\ F_N &= 2\tilde{\sigma}\tilde{\gamma} * y_{N-1}^j + (\tilde{p}_N - \tilde{\beta}_2) * y_N^j + 2\bar{\gamma} * y_{N-1}^{j-1} - C_N * y_N^{j-1} + \tau^2 f(x_N, t_j) + \tilde{\mu}_2 \end{aligned} \quad (8.12)$$

СЛАР (8.10)-(8.12) ефективно розв'язуються стійкою *монотонною прогонкою*.

Всі різницеві рівняння, які утворюють різницеві задачі для $\sigma = 0$, а також для $0 < \sigma \leq \frac{1}{2}$, мають апроксимацію порядку $O(\tau^2 + h^2)$.

В Лекції №13 ми отримали наступні умови стійкості однопараметричної сім'ї тришарових схем :

- ✓ явна схема $\sigma = 0$ стійка, при виконанні нерівності $\tau \leq h$ – умови Куранта;
- ✓ схема з $\sigma \in (0, \frac{1}{4})$ також умовно стійка ($\tau \leq h\sqrt{1-4\sigma}$);
- ✓ схема з $\sigma \in [\frac{1}{4}, \frac{1}{2}]$ абсолютно стійка.

Різницеві задачі (8.7)–(8.9) і (8.7),(8.8),(8.10)–(8.12), за умов виконання стійкості, відповідної гладкості вхідних даних і шуканої функції, збігається до розв’язку диференціальної задачі (8.1)–(8.4) у відповідних нормах із порядком $O(\tau^2 + h^2)$.

Аудиторне завдання:

п.1. Знайти розв’язок модельної крайової задачі :

$$\left\{ \begin{array}{l} \frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} - 5u(x, t) + 1 - 2t + 5x(1 + xt) - t \cos(2x), \\ (x, t) \in (0, \pi) \times (0, T], \quad T = 3; \\ u(x, 0) = x, \quad \frac{\partial u(x, 0)}{\partial t} = x^2, \quad x \in [0, \pi]; \\ \frac{\partial u(0, t)}{\partial x} = 1, \quad \frac{\partial u(\pi, t)}{\partial x} = 1 + 2\pi * t, \quad t \in [0, T]. \end{array} \right. \quad (8.13)$$

за допомогою явної схеми, де $u(x, t) = \frac{1}{5} (1 - \cos(t\sqrt{5})) + \frac{1}{27} (\sin(3t) - 3t) \cos(2x) + x(1 + xt)$.

Побудувати графіки наближеного розв’язку $Y(\bar{X}, \bar{J})$ і похибки $\delta(\bar{X}, \bar{J}) = Y(\bar{X}, \bar{J}) - u(\bar{X}, \bar{J})$.

Обчислити значення $\|\delta(\bar{X}, \bar{J})\|_{Ch}$ і $\max_j \left(\max_k |\delta(\bar{X}, \bar{J})| \right)$.

Створимо власну М-функцію **fdm_H10** – солвер, який реалізує алгоритм явної РС для розв’язання (8.1)–(8.4). Також напишемо М-функцію **p13a01**, в якій означимо анонімні функції p, f, u та підфункції $G0, Ga, Gb$ і зробимо виклик цього солвера :

```
function p13a01
% Розв'язування крайової задачі для одновимірного
% рівняння гіперболічного типу 2-го порядку
% різницеvim методом ( sigm=0 - явна схема )
clear all, close all, clc
global q5 dd
% параметри задачі :
p = @(x,t) (5); % функція p(x,t)
f = @(x,t) (1-2*t+5*x.*(1+x.*t)-t.*cos(2*x)); % функція f(x,t)
% точний розв'язок :
q5 = sqrt(5); dd = 1/27;
u = @(x,t) (0.2*(1-cos(q5.*t))+dd.*(sin(3*t)-3*t).*cos(2*x)+x.*(1+x.*t));
v = [1, 1]; a = 0; b = pi;
% параметри РЗ :
N = 101; X = linspace(a, b, N);
tau = 3e-2; M = 100; J = [0:2:M];
U = fdm_H20(p, f, @G0, @Ga, @Gb, v, X, J, tau);
J = tau.*J; nJ = length(J); T = J(nJ); Uu = U;
for k = 1 : N
    x = X(k);
    for j = 1 : nJ
        Uu(j,k) = u(x, J(j));
    end
end
end
viewU(3, X, J, U, 0, Uu);
end

function [u0, u1, d2u0] = G0(x)
```

```

u0 = x; % u(x,0) = u0(x)
u1 = x.*x; % du(x,0)/dt = u1(x)
d2u0 = 0; % u0''(x)
end

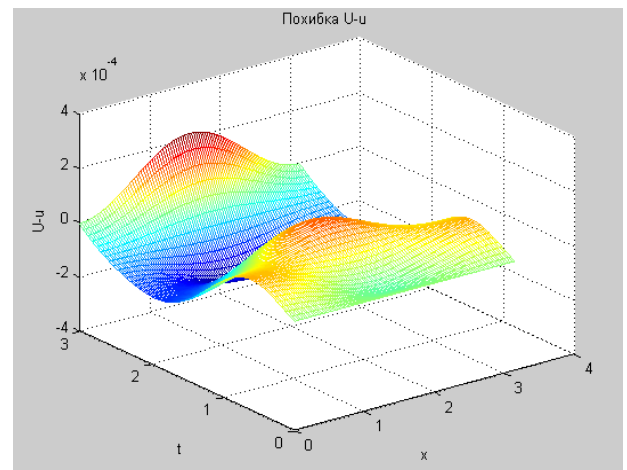
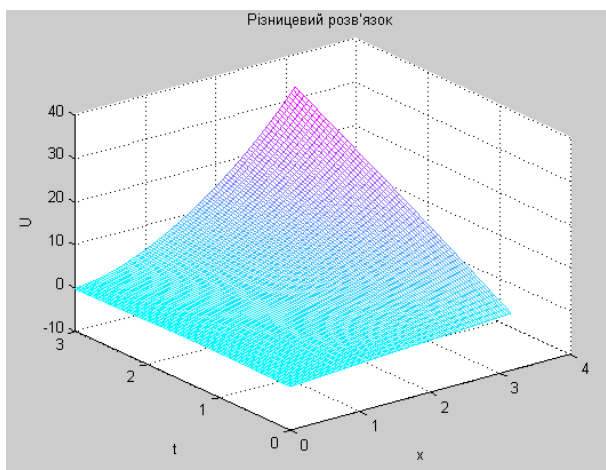
function gr = Ga(t)
% Опис [bet1, мю1(t)] для x=a
% КУ I   роду va=0 : 1*u(a,t) = мю1(t)
% КУ II  роду va=1 : -du(a,t)/dx + 0*u(a,t) = мю1(t)
% КУ III роду va=1 : -du(a,t)/dx + bet1*u(a,t) = мю1(t)
global q5 dd
%c = 0.2*cos(q5.*t); s = 3*t; s = dd.*(sin(s)-s);
%gr = [1,0.2+s-c]; % I p
gr = [0,-1]; % II p
%gr = [2,-0.6+2*(s-c)]; % III p
end

function qr = Gb(t)
% Опис [bet2, мю2(t)] для x=b
% КУ I   роду vb=0 : 1*u(b,t) = мю2(t)
% КУ II  роду vb=1 : du(b,t)/dx + 0*u(b,t) = мю2(t)
% КУ III роду vb=1 : du(b,t)/dx + bet2*u(b,t) = мю2(t)
global q5 dd
%c = 0.2*cos(q5.*t); s = 3*t; s = dd.*(sin(s)-s);
%pt = pi.*t; ppt = pi.*(1+pt);
%qr = [1,0.2+s-c+ppt]; % I p
qr = [0,1+2*pi*t]; % II p
%qr = [1,1.2+2*pt+s-c+ppt]; % III p
end

```

Запустимо її на виконання і отримаємо наступний результат (як бачимо значення похибки в $\|\cdot\|_{Ch}$ узгоджується з теоретичною оцінкою $O(h^2 + \tau^2)$, враховуючи, що $h^2 + \tau^2 = 1.887 \cdot 10^{-3}$):

Похибка $U-u = 1.81746e-002$
 $\max(\max|U-u|) = 2.21850e-004$



Змінюючи значення вектора $v = [va, vb]$ – опис виду КУ в точках $x=a$, $x=b$ і відповідних функції $Ga(t)$ – опис $[\beta_1, \mu_1(t)]$ та $Gb(t)$ – опис $[\beta_2, \mu_2(t)]$, будемо отримувати наближений розв'язок (8.13) для різних варіантів комбінації КУ I, II-III роду.

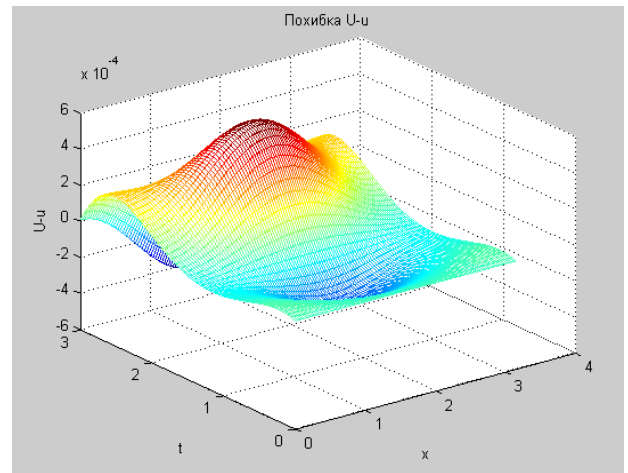
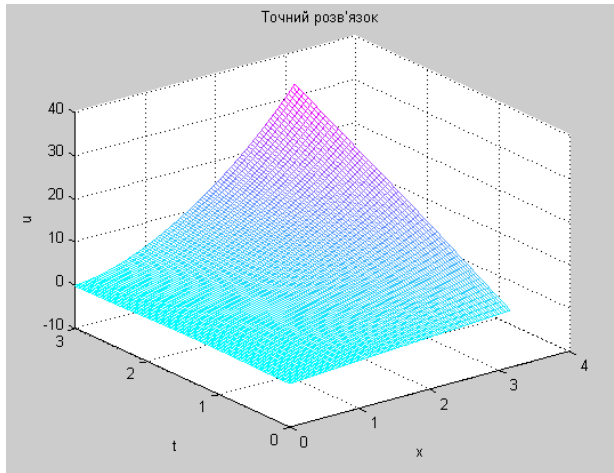
п.2. Знайти розв'язок задачі (8.13) за допомогою неявної схеми з $\sigma = \frac{1}{4}$. Виведення інформації про отриманий розв'язок організувати аналогічно **п.1.**

Створимо власну М-функцію ***fdm_H1s*** – солвер, який реалізує алгоритм неявної РС для розв’язання (8.1)–(8.4). Також напишемо М-функцію *p13a02*, яка буде відрізнятися від *p13a01* тільки двома рядками :

```
tau = 3e-2; M = 100; J = [0:2:M]; sigm = 0.25;
U = fdm_H2s(p, f, @G0, @Ga, @Gb, v, X, J, tau, sigm);
```

(тому повний її текст тут не приводимо).

Отримано наступний результат її виконання (у варіанті $\sigma = \frac{1}{4}$)



Похибка $U-u = 3.54863e-002$
 $\max(\max |U-u|) = 5.51768e-004$

Тут також значення похибки в $\|\cdot\|_{Ch}$ узгоджується з теоретичною оцінкою $O(h^2 + \tau^2)$, враховуючи, що $h^2 + \tau^2 = 1.887 \cdot 10^{-3}$.

Завдання для самостійної роботи:

п.1. Знайти розв’язок задачі (8.1)–(8.4) за допомогою явної схеми для вхідних параметрів $[a,b]=[0,1]$; $T=3$, а решта – указана в Таблиці 8.1. Побудувати графік наближеного розв’язку $Y(\bar{X}, \bar{J})$.

п.2. Знайти розв’язок задачі (8.1)–(8.4) за допомогою неявної схеми для вхідних параметрів $[a,b]=[0,1]$; $T=3$, а решта – указана в Таблиці 8.1. Побудувати графік наближеного розв’язку $Y(\bar{X}, \bar{J})$.

Таблиця 8.1 Варіанти індивідуального завдання.

k	$p(x, t)$	$f(x, t)$	$u0(x), u1(x)$	$[\beta_1, \mu_1(t)]$	$[\beta_2, \mu_2(t)]$	v
1	$x + gt$	$(5 + xt)(g + x^2)$	$x, 1 - x$	$[0, \cos(t)]$	$[1, 1 - 0.1t]$	$[1, 0]$
2	$gx + t$	$\exp(-gx) + 2x^3t$	$1 - x, x$	$[1, 1 - 0.2t]$	$[0, t - 1]$	$[0, 1]$
3	$gx - 0.2t$	$5x^3 \sin(t + gx)$	$1 + x, 2x$	$[0, 1 - \sin(t)]$	$[1, 2 - 0.5t]$	$[1, 0]$
4	$2/g + xt$	$gt - \exp(x - t^2)$	$x, 1 - x$	$[1, 1 + 0.2t]$	$[1, 1 - 0.1t]$	$[1, 1]$
5	$gt(x + 1)$	$(x - t^2)(1 + gx^2)^{-1}$	$x^2 + 1, x$	$[1, \cos(2t)]$	$[1, 2 - 0.4t]$	$[0, 0]$
6	$x/(t + g)$	$(t + gx^2)(1 + x)^{-1}$	$1 - x, x$	$[2, 1 - 0.1t]$	$[0, t - 1]$	$[1, 1]$
7	$(x + t)/g$	$2 \exp(t - gx)$	$1 + x, 0$	$[2, 3 + \sin(t)]$	$[1, 2 - 0.5t]$	$[1, 1]$
8	$x^2 + t/g$	$5(t + gx)(1 + tx^2)^{-1}$	$1 - x^2, x$	$[1, \cos(2t)]$	$[1, t]$	$[0, 0]$
9	$(x + t)^2/g$	$2g(t + x^2)$	$x^3 + 1, x$	$[1, 1 + 0.5t]$	$[1, 2 - t]$	$[0, 0]$
10	gtx	$5(t + x)(g + x^2)^{-1}$	$x, 1 - x^2$	$[1, 0.3t]$	$[1, 1 - 0.2t]$	$[0, 1]$

11	$t + gx^2$	$5(t+x) + gx^2$	$1-x, x$	$[1, 1-0.5t]$	$[1, 0.3t^2]$	$[0, 0]$
12	$1 + gtx^2$	$3\sin^2(x+gt)$	$1-x^3, x$	$[1, 1+0.5t]$	$[1, 0.2t^2]$	$[0, 0]$
13	$gt + x^2$	$3\sin(x+gt)$	x^3+1, x	$[0, 1+0.2t]$	$[1, 4+0.1t]$	$[1, 1]$
14	$gt + \sqrt{x}$	$50(g-t+x^3)$	$x^3-1, 0$	$[0, 0.2t-1]$	$[2, 2+0.1t]$	$[1, 1]$
15	$t + gx^2$	$\sqrt{t} + 50g * x^3$	$1+x, 0$	$[0, 1+\sin(t)]$	$[0, 1+0.5t]$	$[1, 1]$
16	$g\sqrt{t} + x^2$	$(t+5gx)\exp(x^3)$	$1-x^2, x$	$[0, \sin(0.5t)]$	$[0, t]$	$[1, 1]$
17	$gt + x$	$10(t+gx^2)$	$1-x, x$	$[1, 1-0.2t]$	$[1, t(t+1)]$	$[1, 0]$
18	$(xt+g)^{-1}$	$50(gt+x^3)$	x^3+1, x	$[0, 0.2t]$	$[0, 3+0.2t]$	$[1, 1]$
19	$t/(x+g)$	$xt + 50\sin(gx)$	$1-x^3, x$	$[0, 0.4t]$	$[0, t^2-3]$	$[1, 1]$
20	$x/(t+g)$	$gxt + 5\sin(x)$	$x^3-1, 0$	$[0, 0.2t-1]$	$[0, 3-0.1t]$	$[1, 1]$
21	$x+t/g$	$(t+gx)(1+x^3)^{-1}$	$x, 1-x^2$	$[0, 1+0.5t]$	$[0, 1+0.4t]$	$[1, 1]$
22	$(g-xt)^2$	$30g(t+x)x$	$1-x, 0.3x$	$[0, -1+0.2t]$	$[0, 1+0.1t^2]$	$[1, 1]$
23	$gx + \sqrt{t^3}$	$t^2 + 5gx^2$	$x^2+1, 0$	$[0, t]$	$[0, 1+0.6\sqrt{t}]$	$[1, 1]$
24	$1+xt/g$	$gx(t+\sin(x))$	$\sqrt{x^3}+1, x$	$[1, \cos(0.5t)]$	$[1, 2-0.4t]$	$[0, 0]$
25	$gt + x^2$	$gtx + \sqrt{x^3}$	$1+x, 1-x$	$[1, 0.1+t]$	$[1, 2/9+t]$	$[0, 0]$
26	$g + x^2t$	$t + g\sqrt{x^2-x}$	$2+x, 1-x$	$[1, 0.2+t]$	$[1, 2+\sin(t)]$	$[0, 0]$
27	$g(t-x)^2$	$3(t+gx^2)$	$e^{-x(1-x)}, 0$	$[1, 1+0.2t]$	$[1, \cos(2t)]$	$[0, 0]$
28	$gx\sin(t)$	$(t+5x)(1+gt+x^3)^{-1}$	$x, 1-x^2$	$[0, \cos(t)]$	$[1, 1+0.3t]$	$[1, 1]$
29	$g\sqrt{t+x^4}$	$5\sqrt{gt+x^3}$	$\sqrt{x^3}+1, 0$	$[0, 0.2t^2]$	$[1, 2-0.1t]$	$[1, 0]$
30	$t^2/(g+x)$	$2(t+x)(1+gx^2)^{-1}$	$1+x, \sqrt{x}$	$[0, 1+\sin(t)]$	$[0, 1+0.8t]$	$[1, 1]$

Програмне забезпечення на MATLAB:

```

function U = fdm_H20(p, f, G0, Ga, Gb, v, X, J, tau)
% Наближений розв'язок крайової задачі :
%  $d^2u(x,t)/dt^2 = d^2u(x,t)/dx^2 - p(x,t)*u + f(x,t)$ ,
%  $x \in (a,b), t \in (0,T]$ 
%  $u(x,0) = u0(x), x \in [a,b]$ ,
%  $du(x,0)/dt = u1(x), x \in [a,b]$ ,
% КУ I роду  $va=0$  :  $u(a,t) = mю1(t)$ 
% КУ II-III роду  $va=1$  :  $-du(a,t)/dx + bet1*u(a,t) = mю1(t)$ 
% КУ I роду  $vb=0$  :  $u(b,t) = mю2(t)$ 
% КУ II-III роду  $vb=1$  :  $du(b,t)/dx + bet2*u(b,t) = mю2(t)$ 
% явною різницевою схемою (sigm = 0)
% Вхідні дані :
% p, f - функції з описом p(x,t), f(x,t)
% G0 - функція з описом [u0(x), u1(x), u0''(x)]
% Ga - вектор-функція [ 1, mю1(t) ] при va=0
% [bet1, mю1(t)] при va=1
% Gb - вектор-функція [ 1, mю2(t) ] при vb=0
% [bet2, mю2(t)] при vb=1
% v - [va, vb] - масив значень виду КУ в x=a,b
% X - масив рівномірної сітки по x a=X(1), b=X(end)

```

```

%      J - масив з номерами шарів для запам'ятовування розв'язку
%      max(J(:)) = M - гранична точка сітки по t, T = M*tau
%      tau - крок сітки по змінній t
%      Вихідні дані :
%      U - масив значень наближеного розв'язку
%      U(j,k) = u(X(k),tau*J(j))

% параметри PЗ :
N = length(X); h = X(2)-X(1); N1 = N - 1;
nJ = length(J); M = J(nJ); U = zeros(nJ,N);
% обчислення додаткових констант :
gam = tau./h; gam = gam.*gam; tau2 = tau.*tau;
t2 = 0.5.*tau2; g2 = 2.*(1 - gam);
ga2 = 2.*gam; hg = ga2.*h;
Ym = zeros(1,N); Y = Ym; Yp = Y;
% формування початкових умов при t=0 :
t = 0;
for k = 1 : N
    x = X(k); [u0, u1, d2u0] = G0(x); Ym(k) = u0;
    Y(k) = u0 + tau.*u1 + t2.*(d2u0-p(x,t).*u0+f(x,t));
end
clear u0 u1 d2u0
jp = 1;
if J(jp) == 0 % запам'ятовуємо для j=0
    U(jp,:) = Ym; jp = jp + 1;
end
if J(jp) == 1 % запам'ятовуємо для j=1
    U(jp,:) = Y; jp = jp + 1;
end
for j = 2 : M % знаходження наближеного розв'язку Yp(:) на шарі j
    t = (j-1).*tau;
    for k = 2 : N1
        x = X(k); pp = g2-tau2.*p(x,t);
        Yp(k) = gam.*(Y(k-1)+Y(k+1))+pp.*Y(k)-Ym(k)+tau2.*f(x,t);
    end
    tt = t + tau;
    if v(1)
        x = X(1); w = Ga(t); pp = g2-tau2.*p(x,t); be = pp-hg.*w(1);
        Yp(1) = be.*Y(1)+ga2.*Y(2)-Ym(1)+tau2.*f(x,t)+hg.*w(2);
    else
        w = Ga(tt); Yp(1) = w(2);
    end
    if v(2)
        x = X(N); w = Gb(t); pp = g2-tau2.*p(x,t); be = pp-hg.*w(1);
        Yp(N) = ga2.*Y(N1)+be.*Y(N)-Ym(N)+tau2.*f(x,t)+hg.*w(2);
    else
        w = Gb(tt); Yp(N) = w(2);
    end
    if J(jp) == j % запам'ятовуємо для j
        U(jp,:) = Yp; jp = jp + 1;
    end
    for k = 1 : N
        Ym(k) = Y(k); Y(k) = Yp(k);
    end
end
end
end

function U = fdm_H2s(p, f, G0, Ga, Gb, v, X, J, tau, sigm)
% d^2u(x,t)/dt^2 = d^2u(x,t)/dx^2 - p(x,t)*u + f(x,t),

```

```

%           $x \in (a,b), \quad t \in (0,T]$ 
%           $u(x,0) = u_0(x), \quad x \in [a,b],$ 
%           $du(x,0)/dt = u_1(x), \quad x \in [a,b],$ 
% КУ I роду       $va=0 :$                                  $u(a,t) = mю1(t)$ 
% КУ II-III роду  $va=1 :$   $-du(a,t)/dx + bet1*u(a,t) = mю1(t)$ 
% КУ I роду       $vb=0 :$                                  $u(b,t) = mю2(t)$ 
% КУ II-III роду  $vb=1 :$   $du(b,t)/dx + bet2*u(b,t) = mю2(t)$ 
%          (неявна схема  $0 < \text{sigm} \leq 0.5$ )
% Вхідні дані :
%  $p, f$  - функції з описом  $p(x,t), f(x,t)$ 
%  $G0$  - функція з описом  $[u_0(x), u_1(x), u_0''(x)]$ 
%  $Ga$  - вектор-функція  $[1, mю1(t)]$  при  $va=0$ 
%                                $[bet1, mю1(t)]$  при  $va=1$ 
%  $Gb$  - вектор-функція  $[1, mю2(t)]$  при  $vb=0$ 
%                                $[bet2, mю2(t)]$  при  $vb=1$ 
%  $v = [va, vb]$  - масив значень виду КУ в  $x=a,b$ 
%  $X$  - масив рівномірної сітки по  $x$   $a=X(1), b=X(end)$ 
%  $J$  - масив з номерами шарів для запам'ятовування розв'язку
%           $J(end) = M$  - гранична точка сітки по  $t, T = M*tau$ 
%  $tau$  - крок сітки по змінній  $t$ 
%  $sigm$  - вага РС,  $sigm \in (0, 0.5]$ 
% Вихідні дані :
%  $U$  - масив значень наближеного розв'язку
%           $U(j,k) = u(X(k), tau*J(j))$ 
if sigm <= 0 | sigm > 0.5
    sigm = 0.25;
    warning('FDM_H1S: значення ваги sigm змінено на 0.25!');
end
% параметри РЗ :
N = length(X); h = X(2)-X(1); N1 = N - 1;
nJ = length(J); M = J(nJ); U = zeros(nJ,N);
% обчислення додаткових констант :
tau2 = tau.*tau; t2 = 0.5.*tau2; ss = 1 - 2.*sigm;
st = sigm.*tau2; st1 = ss.*tau2;
gam = tau./h; gam = gam.*gam; gs = gam.*sigm;
ggs = 2.*gs; cs = 1 + ggs; gs1 = gam.*ss;
sga2 = 2.*gs1; gs2 = 2 - sga2; ga2 = 2.*gam;
hg = ga2.*h; gsh = ggs.*h; shg = sga2.*h;
% формування коефіцієнтів  $A, C, B, F, k=1, \dots, N$  для СЛАР
A = ones(1,N); F = A; A(1) = 0; A(N) = 0;
B = gs.*A; A = B; C = F;
if v(1)
    B(1) = ggs;
end
if v(2)
    A(N) = ggs;
end
end
Ym = F; Y = Ym; Yp = Y;
% формування початкових умов при  $t=0$  :
t = 0;
for k = 1 : N
    x = X(k); [u0, u1, d2u0] = G0(x); Ym(k) = u0;
    Y(k) = u0 + tau.*u1 + t2.*(d2u0-p(x,t).*u0+f(x,t));
end
clear u0 u1 d2u0
jp = 1;
if J(jp) == 0 % запам'ятовуємо для  $j=0$ 
    U(jp,:) = Ym; jp = jp + 1;
end

```

```

if J(jp) == 1 % запам'ятовуємо для j=1
    U(jp,:) = Y; jp = jp + 1;
end
for j = 2 : M % знаходження наближеного розв'язку Yp(:) на шарі j
    t = (j-1).*tau;
    for k = 2 : N1
        x = X(k); km = k-1; kp = k+1; pp = p(x,t);
        c = cs + st.*pp; C(k) = c; pp = gs2-st1.*pp;
        pp = gs1.*(Y(km)+Y(kp)) + pp.*Y(k) + tau2.*f(x,t);
        F(k) = gs.*(Ym(km)+Ym(kp)) - c.*Ym(k) + pp;
    end
    tt = t + tau;
    if v(1)
        x = X(1); w = Ga(t); pp = p(x,t);
        be = gsh.*w(1); c = cs + st.*pp + be; C(1) = c;
        be = shg.*w(1); pp = gs2 - st1.*pp - be;
        c = -c.*Ym(1) + tau2.*f(x,t) + hg.*w(2);
        F(1) = sga2.*Y(2) + pp.*Y(1) + ggs.*Ym(2) + c;
    else
        w = Ga(tt); F(1) = w(2);
    end
    if v(2)
        x = X(N); w = Gb(t); pp = p(x,t);
        be = gsh.*w(1); c = cs + st.*pp + be; C(N) = c;
        be = shg.*w(1); pp = gs2 - st1.*pp - be;
        c = -c.*Ym(N) + tau2.*f(x,t) + hg.*w(2);
        F(N) = sga2.*Y(N1) + pp.*Y(N) + ggs.*Ym(N1) + c;
    else
        w = Gb(tt); F(N) = w(2);
    end
    [Yp, AL, er] = m_progm(A, C, B, F);
    if J(jp) == j % запам'ятовуємо для j
        U(jp,:) = Yp; jp = jp + 1;
    end
    for k = 1 : N
        Ym(k) = Y(k); Y(k) = Yp(k);
    end
end
end
end

```

```

function viewU(s, X, Y, U, kc, UU)
    [Mx, My] = meshgrid(X, Y); close all;
    mesh(Mx, My, U);
    e2 = 'Різнцевий розв'язок';
    title(e2); xlabel('x'); ylabel('t'); zlabel('U');
    shading interp; colormap cool; %view(40,20);
    if s, pause(s);
    else, pause;
    end
    close all;
    if kc
        contour(Mx, My, U, kc);
        xlabel('x'); ylabel('t'); title(e2); colormap hsv;
        if s, pause(s);
        else, pause;
        end
        close all;
    end
    if nargin == 5

```

```

        return
    end
    mesh(Mx, My, UU);
    e2 = 'Точный разв'язок';
    title(e2); xlabel('x'); ylabel('t'); zlabel('u');
    shading interp; colormap cool; %view(40,20);
    if s, pause(s);
    else, pause;
    end
    close all;
    if kc
        contour(Mx, My, UU, kc);
        xlabel('x'); ylabel('t'); title(e2); colormap hsv;
        if s, pause(s);
        else, pause;
        end
        close all;
    end
    Uu = U - UU;
    mesh(Mx, My, Uu);
    e2 = 'Похибка U-u';
    title(e2); xlabel('x'); ylabel('t'); zlabel('U-u');
    shading interp; colormap jet; %view(40,20);
    e1 = norm(Uu,inf); fprintf([e2,' =%13.5e\n'], e1);
    e1 = max(max(abs(Uu))); fprintf('max(max|U-u|)=%13.5e\n', e1);
    if s, pause(s);
    else, pause;
    end
end
end

```

СПИСОК ЛІТЕРАТУРИ

1. *Анджейчак І. А.* Практикум з обчислювальної математики част. 1 : навч. посіб. / І. А. Анжейчак, Є. М. Федюк. – Львів : ДУ «Львів. Політех.», 2000. – 100 с.
2. *Анджейчак І. А.* Практикум з обчислювальної математики. Лекції : навч. посіб. част. 2 / І. А. Анжейчак, Є. М. Федюк, В. Є. Анохін та ін. – Львів : ДУ «Львів. Політех», 2001. – 152 с.
3. *Бігун Я. Й.* Числові методи розв’язування нелінійних рівнянь і систем : навч. посіб. / Я. Й. Бігун, І. В. Березовська. – Чернівці : Чернівецький нац. ун-т, 2011. – 104 с.
4. *Вакал Є.С.* Класифікація рівнянь із частинними похідними з використанням системи MATLAB / Є.С. Вакал, Ю.Є. Вакал. – К.; Основа, 2017. – 104 с.
5. *Вакал Є.С.* Методи математичної фізики в прикладах і задачах : навч. посіб. для студ. мех.-мат. ф-ту / Є. С. Вакал, А. В. Ловеїкін. – К.: Вид. Кравченко Я.О., 2020. – 188 с.
6. Використання математичного пакета MATLAB для розв’язування прикладних задач : навч. посіб. / Б.П. Довгий, Є.С. Вакал, Ю.Є. Вакал, А.В. Попов. – К.: Фітосоціоцентр, 2012. – 78 с.
7. Використання системи комп’ютерної математики MATLAB для розв’язування прикладних задач: навч. посіб. / Б.П. Довгий, Є.С. Вакал, Ю.Є. Вакал, А.В. Попов. – К.: ВПЦ "Київський університет", 2016. – 143 с.
8. *Гаврилюк І. П.* Методи обчислень / І. П. Гаврилюк, В. Л. Макаров. – К. : Вища шк., 1995. – 367 с.
9. *Гап’як І. В.* Методичні вказівки до застосування математичних пакетів для чисельно-аналітичного розв’язування задач: для студ. мех.-мат. ф-ту [Електронний ресурс] / І.В. Гап’як, Б.П. Довгий, Є.С. Вакал, А.В. Ловеїкін. – Режим доступу: http://www.mechmat.univ.kiev.ua/wp-content/uploads/2021/04/hapiak_dovhyj_vakal_lovejkin_metodychni-vkazivky.pdf. – К., 2021. – 121 с.
10. *Довгий Б.П.* Сплайн-функції та їхнє застосування : навч. посіб. / Б.П. Довгий, А.В. Ловеїкін, Є.С. Вакал, Ю.Є. Вакал. – К.: ВПЦ "Київський університет", 2017. – 120 с.
11. *Довгий Б.П.* Методи обчислень: методичні вказівки до лабораторних робіт з використанням пакета MATLAB / Б.П. Довгий, Є.С. Вакал, Ю.Є. Вакал. – К.: ВПЦ "Київський університет", 2017. – 60 с.
12. *Довгий Б.П.* Інформаційні системи та технології : навч. посіб. для студ. мех. мат. ф-ту / Б.П. Довгий, Є.С. Вакал. – К.: Вид. Кравченко Я. А., 2021. – 111 с.
13. *Довгий Б.П.* Інформаційні системи та технології. Збірник задач для студ. мех.-мат. ф-ту [Електронний ресурс] / Б.П. Довгий, Є.С. Вакал. Режим доступу : http://www.mechmat.univ.kiev.ua/wp-content/uploads/2021/04/dovhyj_vakal_informatsijni-sysnemy-ta-tekhnohohii_zb_zadach.pdf. К., 2001. – 58 с.
14. *Довгий Б.П.* Учбові завдання для аудиторної та самостійної роботи з навчальної дисципліни “Методи обчислень”: навч. посіб. / Б. П. Довгий, А. В. Ловеїкін. – К. : ВПЦ «Київський університет», 2020. – 123 с.
15. *Задачин В. М.* Чисельні методи: навч. посіб. / В.М. Задачин, І.Г. Конюшенко. – Харків : Вид. ХНЕУ, 2014. – 180 с.
16. *Костюшко І.А.* Методи обчислень : підручник / І.А. Костюшко, Н.Д. Любашенко, В.В. Третиник. – Київ : КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2021. – 243 с.

17. *Лященко М. Я.* Чисельні методи / М. Я.Лященко, М.С.Головань. – К. : «Либідь», 1996. – 288 с.
18. *Охріменко М.Г.* Методи розв'язання некоректно-поставлених задач : навч. посіб. / М.Г. Охріменко, О.А. Жуковська, О.О. Купка. – К. : Центр учб. літер., 2008. – 166 с.
19. *Попов В.В.* Методи обчислень : конспект лекцій для студ. мех. мат. ф-ту / В.В. Попов. – К.:ВПЦ "Київський університет", 2012. – 303 с.
20. *Попов В.В.* Лабораторні роботи та домашні завдання для самостійної роботи з дисципліни "Методи обчислень" для студ. мех. мат. ф-ту / В.В. Попов, Б.П. Довгий, Є.С. Вакал. – К.: Укр. фітосоціолог. центр, 2006. – 32 с.
21. *Самборська О.М.* Числові методи : навч. посіб. / О.М.Самборська, Б.Г.Шелестовський. – Тернопіль : ТДТУ ім. Івана Пулюя, 2008. – 140 с.
22. *Цегелик Г. Г.* Чисельні методи : підручник / Г. Г. Цегелик. – Львів : Н. У., 2004. – 407 с.'
23. *Fokas, A. S.* A Unified Approach to Boundary Value Problems / A S. Fokas. – SIAM, 2008. – 336 p.
24. *Forsythe, G. E.* Computer Methods for Mathematical Computations / G. E. Forsythe, M. A. Malcolm, C. B. Moler, – Englewood Cliffs, New Jersey. Prentice Hall, Inc., 1977. – 270 p.
25. *Greenbaum, A.* Numerical Methods: Design, Analysis, and Computer Implementation of Algorithms /A. Greenbaum, T. P. Chartier.– Princeton University Press, Princeton, N.J., 2012. – 464 p.
26. *Hageman, L.A.* Applied iterative methods /L.A. Hageman, D.M. Yong. – New York, Academic Press, 1981. – 386 p.
27. *Hamming, R.W.* Numerical Methods for Scientists and Engineers / R. W. Hamming. – New York : Dover Publications, 1986. – 721 p.
28. *Isaacson, E.* Analysis of Numerical Methods /E. Isaacson, H. B.Keller. – New York : Wiley, 1966. – 360 p.
29. *Ortega, J. M.* Iterative Solution of Nonlinear Equations in Several Variables (Computer Science & Applied Mathematics Monograph) /J. M.Ortega, W. C. Rheinboldt . – New York and London, Academic Press, 2014. –572 p.
30. *Ortega, J.M.* Introduction to Numerical Methods for Differential Equations /J.M.Ortega, W.G.Poole. – London, Pitman, 1981. – 329 p.
31. *Samarskii, A.A.* Blow-up in quasilinear parabolic equations / A. A. Samarskii, V.A. Galaktionov, S.P. Kurdyumov, A.P. Mikhailov. – Walter de Gruyte Berlin, NY, 1995. – 534 p.
32. *Samarsky, A.A.* Computational Heat Transfer / A. A. Samarsky, P. N. Vabishchevich // Mathematical Modpomielling. – Vol.1. – Chichester, Wiley, 1995. – 406 p. // The Finite Difference Methodology – Vol.2. – Chichester, Wiley, 1995. – 422 p.
33. *Samarskii, A.A.* The theory of difference schemes / A.A. Samarskii. – New York – Basel. Marcel Dekker, Inc, 2001. –761 p.
34. *Schoenberg, I.J.* Selectedpapers. Contribution to the problem of approximation of equidistant data by analytic functions. Vol.2. / I.J. Schoenberg. – New York: Springer Science+Business Media, 1988. – P. 45-46.
35. *Schoenberg, I.J.* Cardinal spline interpolation. / I.J. Schoenberg. – Philadelphia: PA: Society of Industrial and Applied Mathematics, 1973. – P. 221-244.

ЗМІСТ

ВСТУП	3
ЛЕКЦІЯ №01	
ТЕМА № 1. Числове розв'язування інтегральних рівнянь.....	4
Метод квадратурних формул. Метод заміни ядра виродженим. Метод заміни шуканої функції. Метод послідовних наближень	
ЛЕКЦІЯ №02	
ТЕМА № 2. Методи розв'язування задачі Коші для звичайних диференціальних рівнянь.....	10
Різницеві однокрокові. Поняття збіжності, нев'язки, порядку апроксимації. Методи типу Рунге-Кутта. Деякі загальні аспекти використання чисельних методів.	
ЛЕКЦІЯ №03	
ТЕМА № 2. Методи розв'язування задачі Коші для звичайних диференціальних рівнянь.....	16
Різницеві багатокрокові. Стійкість і збіжність різницевих m-крокових методів. Стійкість умовна і абсолютна. Жорсткість систем ЗДР. Деякі методи розв'язування задачі Коші для ЖС ЗДР.	
ЛЕКЦІЯ №04	
ТЕМА № 3. Різницеві методи розв'язування крайових задач для ЗДР	22
Основні поняття методу скінченних різниць. Різницеві схеми для лінійних задач. Різницеві схеми для нелінійних задач.	
ЛЕКЦІЯ №05	
ТЕМА № 3. Різницеві методи розв'язування крайових задач для ЗДР	29
Інтегро-інтерполяційний метод побудови різницевих схем для лінійних задач з неперервними або кусково-неперервними коефіцієнтами.	
ЛЕКЦІЯ №06	
ТЕМА № 3. Різницеві методи розв'язування крайових задач для ЗДР	34
Проекційно-різницевий метод побудови різницевих схем.	
ЛЕКЦІЯ №07	
ТЕМА № 4. Різницевий метод розв'язання крайових задач для одновимірних лінійних рівнянь гіперболічного типу першого порядку.....	41
Побудова сімейства різницевих схем. Похибка апроксимації. Стійкість і збіжність різницевої схеми. Знаходження розв'язку. Консервативні схеми для квазілінійних рівнянь переносу.	
ЛЕКЦІЯ №08	
ТЕМА № 5. Різницевий метод розв'язання крайових задач для рівнянь параболічного типу порядку	51
5.1 Різницеві схеми для лінійних одновимірних задач. Побудова сімейства різницевих схем. Похибка апроксимації. Стійкість і збіжність різницевих схем. 5.2 Різницеві схеми для циліндрично- і сферично-симметричних задач теплопровідності	

ЛЕКЦІЯ №09

ТЕМА № 5. Різницевий метод розв'язання крайових задач для рівнянь параболічного типу	56
5.3. Економічні різницеві схеми. Економічні РС для двовимірних рівнянь. Передумови для написання економічних РС. Поздовжно-поперечна схема.	

ЛЕКЦІЯ №10

ТЕМА № 5. Різницевий метод розв'язання крайових задач для рівнянь параболічного типу	60
5.4. Лінійні крайові задачі для багатовимірних рівнянь. Поняття сумарної апроксимації, адитивності РС. Побудова для КЗ параболічного типу адитивних РС з сумарною апроксимацією на основі локально-одновимірного методу. Варіант ЛОС (перший ланцюг). Варіант ЛОС (другий ланцюг).	

ЛЕКЦІЯ №11

ТЕМА № 6. Різницеві схеми для рівняння Пуассона з крайовими умовами Діріхле	66
Побудова різницевої схеми. Похибка апроксимації. Стійкість і збіжність РС. Знаходження розв'язку. Порівняння ІМ Якобі, Зейделя, релаксації та їх програмна реалізація	

ЛЕКЦІЯ №12

ТЕМА № 6. Різницеві схеми для рівняння Пуассона з крайовими умовами Діріхле	77
Алгоритм методу змінних напрямків з вибором оптимальних ітераційних параметрів (по Жордану). Алгоритм ПТМ з чебишовським набором параметрів. Порівняння МЗН і ПТМ та їх програмна реалізація.	

ЛЕКЦІЯ №13

ТЕМА № 7. Різницевий метод розв'язання крайових задач для лінійних рівнянь гіперболічного типу другого порядку	83
РС для одновимірних рівнянь гіперболічного типу другого порядку. Побудова сімейства різницевих схем. Похибка апроксимації. Стійкість і збіжність РС. РС для багатовимірних рівнянь гіперболічного типу другого порядку.	

Практичне заняття № 01

ТЕМА № 1. Числове інтегрування	90
---	-----------

Практичне заняття № 02

ТЕМА № 2. Інтегральні рівняння	101
---	------------

Практичне заняття № 03

ТЕМА № 3. Методи розв'язування задачі Коші для звичайних диференціальних рівнянь	120
Різницеві методи та методи Рунге-Кутта. Методи Адамса. Стійкість методів. Поняття жорстких СДР	

Практичне заняття № 04

ТЕМА № 4. Розв'язування крайових задач для ЗДР	129
Метод редукції крайової задачі для ЗДР до задачі Коші	

Практичне заняття № 05

ТЕМА № 4. Розв’язування крайових задач для ЗДР 137

Проекційні методи: колокацій, Бубнова-Гальоркіна.

Варіаційні методи: Рітца, найменших квадратів.

Практичне заняття № 06

ТЕМА № 4. Розв’язування крайових задач для ЗДР 149

Різницеві схеми для лінійних і квазілінійних задач.

Практичне заняття № 07

ТЕМА № 4. Розв’язування крайових задач для ЗДР 161

Інтегро-інтерполяційний метод побудови різницевих схем для лінійних задач

з неперервними або кусково-неперервними коефіцієнтами

Практичне заняття № 08

ТЕМА № 5. Метод прямих (Роте) розв’язання крайових задач для нестационарних рівнянь і систем рівнянь різних типів 174

Практичне заняття № 09

ТЕМА № 6. Різницевий метод розв’язання крайових задач для рівнянь параболічного типу 184

Різницеві схеми для лінійних одновимірних задач

Практичне заняття № 10

ТЕМА № 6. Різницевий метод розв’язання крайових задач для рівнянь параболічного типу 195

Квазілінійні крайові задачі для одновимірного рівняння

Практичне заняття № 11

ТЕМА № 6. Різницевий метод розв’язання крайових задач для рівнянь параболічного типу 205

Лінійні крайові задачі для багатовимірних рівнянь.

Локально-одновимірні різницеві схеми

Практичне заняття № 12

ТЕМА № 7. Різницевий метод розв’язання крайових задач для рівнянь еліптичного типу 214

Наближений розв’язок лінійних КЗ для рівняння Пуассона в областях складених з прямокутників, трапецій, трикутників.

Ітераційні методи розв’язання різницевих задач Діріхле в прямокутнику для рівняння Пуассона (обчислення на усталювання)

Практичне заняття № 13

ТЕМА № 8. Різницевий метод розв’язання крайових задач для одновимірних лінійних рівнянь гіперболічного типу другого порядку 230

СПИСОК ЛІТЕРАТУРИ 241

Навчальне видання

**ДОВГИЙ Борис Павлович
БАКАЛ Євген Сергійович
ЛОВЕЙКІН Андрій Вячеславович**

МЕТОДИ ОБЧИСЛЕНЬ
Конспект лекцій та практичні заняття

Навчальний посібник
для студентів механіко-математичного-факультету