МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Запорізький національний технічний університет

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторних робіт з дисципліни

"Моделювання систем"

для студентів спеціальності 123 "Комп’ютерна інженерія"
всіх форм навчання

2019

Методичні вказівки до виконання лабораторних робіт з дисципліни "Моделювання систем" для студентів спеціальності 123 "Комп’ютерна інженерія" всіх форм навчання / Укл. Р.К. Кудерметов, Н.В. Луценко, О.В. Польська – Запоріжжя: ЗНТУ, 2019. – 58 с.

Укладачі:
Р.К. Кудерметов, к.т.н., доцент
Н.В. Луценко, ст. викладач
О.В. Польська, ст. викладач

Рецензент:
М.Ю. Тягунова, к.т.н., доцент

Відповідальний за випуск
О.В. Польська, ст. викладач

Затверджено
на засіданні кафедри КСМ
Протокол № 11 від 20.05.2019

Рекомендовано до видання
на засіданні НМК факультету КНТ
Протокол № 10 від 31.05.2019

ЗМІСТ

ПОРЯДОК ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ

Перед виконанням лабораторної роботи студент повинен ознайомитися з теоретичними відомостями.

Практична частина складається з загального завдання та індивідуального завдання, вихідні дані до якого обираються за варіантами, наданими у відповідних таблицях. Номер варіанта – залишок від ділення номеру робочого місця на 10. Якщо залишок дорівнює 0, обрати варіант № 10.

Всі лабораторні роботи виконуються у пакеті наукових програм для чисельних обчислень Scilab (/ˈsaɪlæb/). Для кожного пункту лабораторного завдання розроблюється окремий файл сценарію, який зберігається у робочому каталозі.

Після виконання файлу сценарію отримані дані аналізуються та демонструються викладачу разом з вмістом цих файлів.

Перелік вбудованих функцій пакету Scilab, які використовуються в процесі виконання лабораторних завдань, а також їх синтаксис та опис, надано у додатку А.

До захисту лабораторної роботи допускаються студенти, які виконали усі пункти практичної частини за варіантом та за отриманими результатами склали звіт.

Звіт до кожної лабораторної роботи повинен містити:

- титульний лист, виконаний відповідно з СТП;
- мету роботи;
- лістинги та результати виконання sce-файлів у середовищі Scilab у вигляді графіків;
- результати розрахунків за необхідністю;
- відповіді на контрольні питання;
- висновки за результатами досліджень.

Захист лабораторної роботи складається з двох частин: практичної та теоретичної. Письмова (практична) частина передбачує наявність лекційних знань та знань, набутих у процесі виконання лабораторної роботи. Теоретична – усна відповідь на обрані довільно викладачем контрольні питання до відповідної лабораторної роботи.

ЗАГАЛЬНІ ВІДОМОСТІ

Моделювання – це заміщення одного об’єкту (оригіналу) іншим (моделлю) та фіксація або вивчення властивостей оригіналу шляхом дослідження властивостей моделі з метою спрощення, здешевлення, прискорення фіксації або вивчення властивостей оригіналу.

Модель – це система, що заміщує оригінал. Оригінал та модель схожі за деякими одними властивостями і параметрами та відрізняються за іншими.

Одна з *класифікацій моделей технічних систем*:

– *за призначенням*: імітаційні, оптимізаційні, прогнозуючі;

– *за ступенем збігу фізичних властивостей*: фізичні (натурні), математичні, напівнатурні;

– *за принципом побудови*: стохастичні, детерміновані;

– *за способом подання інформації*: дискретні, безперервні, дискретно-безперервні;

– *за темпом часу*: реального часу, прискорені та уповільнені;

– *за пристосуванням*: адаптивні, неадаптивні.

Система – це сукупність взаємопов’язаних об’єктів, що має чотири наступні властивості.

– *цілісність та частковість*. Система – це цілісна сукупність окремих елементів. Зібрані разом елементи можуть створювати систему, а можуть й не створювати. Крім того, система може зруйнуватися без якогось елемента;

– *наявність зв’язків*. Системі притаманна наявність суттєвих стійких зв’язків (відношень) між елементами або властивостями, які перевищують за потужністю (силою) зв’язки цих елементів з елементами, що не входять в дану систему. За фізичним наповненням зв’язки можна розділити на дійсні, енергетичні, інформаційні, змішані. За напрямком розрізняють зв’язки прямі, зворотні, нейтральні.

– *наявність визначеної організації*. Це формування зв’язків елементів, упорядковане розподілення зв’язків та елементів в часі та просторі. При цьому складається визначена структура системи, а властивості елементів трансформуються в функції (дії, поведінку).

Організація проявляється у зменшенні ентропії, у порівнянні з ентропією, **системоформуючих факторів** (*елементів* та *зв'язків*), які визначають можливість створення системи;

– *наявність інтегративних якостей*. Це якості, які притаманні системі в цілому, але не властиві жодному з її елементів окремо. Отже, властивості системи хоча й залежать від властивостей елементів системи, але не визначаються ними повністю.

Складну систему неможливо охопити, зрозуміти, вивчити повністю та детально. Для рішення цієї проблеми використовується **стратифікований підхід** до опису (представлення) системи.

Система задається сімейством моделей, кожна з яких описує систему на своєму рівні абстрагування. Для кожного рівня є ряд характерних особливостей, параметрів, законів та принципів. Слід намагатися найбільшої незалежності цих моделей. Таке представлення складних систем називається стратифікованим описом, а рівні абстрагування називають стратами.

У складних системах виокремлюють три рівні ієрархії:

– **страта** (*рівень опису, або абстрагування*). Мається на увазі, що властивості реального складного об'єкта описуються у формі сукупності, де окремі описи упорядковані за рівнем значущості. Такі ієрархічні системи називаються стратифікованими, тобто розділеними на страти за технологічними, інформаційними або економічними аспектами;

– **шар** (*рівень складності рішення, що приймається*). Механізм прийняття рішення охоплює дві проблеми: рішення повинне прийматися швидко, та рішення повинне прийматися після детального аналізу ситуації. Щоб задовольнити ці вимоги, складна проблема поділяється на сімейство підпроблем, вирішення яких призводить до вирішення всієї проблеми загалом;

– **ешелон** (*організаційний рівень*). Передбачається, що система має елементи, які знаходяться під керуванням інших вирішальних елементів.

1 ЛАБОРАТОРНА РОБОТА № 1
Моделювання динамічних систем

Мета роботи: оволодіння методами комп’ютерного моделювання динамічних систем, що описуються системами звичайних диференціальних рівнянь.

1.1 Теоретичні відомості

За допомогою моделювання можна відтворити поведінку будь-якої фізичної або технічної системи. Це дозволить отримати знання про характеристики оригіналу.

Одним з різновидів моделей є математична модель – математичне уявлення реальності для вилучення необхідної інформації щодо системи. При цьому дослідження базується на використанні диференціальних рівнянь, які допомагають описати стан системи на кожному етапі вивчення її поведінки в залежності від часу.

Звичайне диференціальне рівняння (ЗДР) першого порядку має вигляд (1.1):

$$y'(x) = f(x, y), \tag{1.1}$$

де $y'(x)$ – перша похідна змінної $y(x)$, x – незалежна змінна.

Розв’язати диференціальне рівняння – означає знайти множину функцій, які задовольняють даному рівнянню.

Найбільш поширеним підходом до розв’язання ЗДР є його чисельне інтегрування на комп’ютерах. Одним із найпростіших методів чисельного інтегрування є метод прямокутників (Ейлера), що реалізується за допомогою рекурентного співвідношення:

$$y_{n+1} = y_n + hf(x_n, y_n), \tag{1.2}$$

де y_n – значення змінної y на n-му кроці, x_n – значення змінної x на n-му кроці, h – крок інтегрування, отриманий шляхом поділу часової осі на рівні інтервали.

Для розв’язання рівняння на практиці часто використовують метод чисельного інтегрування *Рунге-Кутта* четвертого порядку $O(h^4)$. Порядок вказує на точність апроксимації розв’язку.

Оскільки розв'язок рівняння – сукупність точок, що графічно формують криву, яка відповідає поведінці системи, то для кожного значення x необхідно обчислити y.

Для обчислення y_{n+1} використовується формула:

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4), \tag{1.3}$$

де значення k_1, k_2, k_3, k_4 обчислюються за допомогою формул:

$$k_1 = f(x_n, y_n),$$
$$k_2 = f(x_n + \frac{h}{2},\ y_n + \frac{h}{2}k_1),$$
$$k_3 = f(x_n + \frac{h}{2},\ y_n + \frac{h}{2}k_2),$$
$$k_4 = f(x_n + h,\ y_n + hk_3).$$

1.2 Дослідження екологічної системи

Для деяких екологічних систем характерним є наявність хижаків та жертв. Наприклад, досліджується компонент біосфери – ліс. Він є системою, оскільки містить багато складових: рослини різних ярусів, тварини. Розглядається закрите середовище, де існують тільки два види тварин: вовки та зайці.

Тварини і рослини взаємодіють між собою. Рослини (редуценти) – їжа для рослиноїдних тварин (консументів першого порядку, наприклад, зайців). Рослиноїдні тварини – їжа для хижаків (консументів другого порядку, наприклад, вовків).

Таким чином, зайці – жертви, а вовки – хижаки. Введемо наступні позначення: x – кількість зайців в системі у момент часу t; y – кількість вовків в системі у момент часу t; x', y' – похідні за часом змінних, відповідно, x і y, тобто швидкості зміни цих видів.

За умови відсутності вовків приріст чисельності зайців описується рівнянням (1.4), де a_{11} – коефіцієнт приросту зайців:

$$x' = a_{11}x. \tag{1.4}$$

У випадку, якщо зайців немає, вовки вимирають і процес зменшення чисельності вовків відповідає рівнянню (1.5), де a_{22} – коефіцієнт зменшення вовків за відсутності зустрічей із жертвами:

$$y' = a_{22}y \,. \tag{1.5}$$

Якщо припустити, що в початковий момент спостереження за системою $t = 0$ число зайців дорівнює $x(0) = x_0$, а число вовків дорівнює $y(0) = y_0$, то аналітичні рішення рівнянь (1.4) і (1.5) матимуть вигляд:

$$x = x_0 e^{a_{11}t}\,,\ \ y = y_0 e^{-a_{22}t}\,.$$

Це означає, що число зайців зростає за експоненціальним законом, а число вовків убуває, також по експоненті.

Змоделюємо систему, в якій територіальні ресурси розподілені між зайцями та вовками. При зустрічі з хижаками кількість зайців буде зменшуватися, а кількість вовків – навпаки, збільшуватися. Цей процес описує система рівнянь Лотки-Вольтери:

$$\begin{cases} x' = a_{11}x - a_{12}xy, \\ y' = -a_{22}x + a_{21}xy. \end{cases} \tag{1.6}$$

Перший член кожного рівняння системи (1.6) відповідає стану моделі за рівняннями (1.4) та (1.5). Елемент $a_{12}xy$ системи (1.6) описує зменшення зайців при зустрічі з вовками, а елемент $a_{21}xy$ – збільшення вовків, якщо зайців для цього достатньо. Таким чином, можна сказати, що a_{12} і a_{21} - коефіцієнти, що характеризують частоту зустрічей жертв з хижаками і, очевидно, що вони рівні.

1.3 Практичне завдання (перша частина)

1.3.1 Моделювання екологічної системи

Для моделювання екологічної системи виконайте наступні дії:

а) зробіть наступні заміни в формулах (1.6): $x = x_1$, $y = x_2$. Після заміни отримаєте наступну систему рівнянь:

$$\begin{cases} x_1' = a_{11}x_1 - a_{12}x_1x_2, \\ x_2' = a_{21}x_1x_2 - a_{22}x_2. \end{cases} \quad (1.7)$$

б) запустіть програму Scilab. У пункті меню "Инструменты" оберіть "Текстовый редактор SciNotes". У середовищі Scilab створіть файл сценарію lab1_1.sce, який містить розв'язання системи (1.7). Задайте необхідні дані у відповідності до лістингу 1.1:

- коефіцієнти взаємодії між видами: `a11=0.06, a12=0.0006`, `a21=0.0006`, `a22=0.24`;
- вектор початкових умов з початковими значеннями кількості зайців ($x_1 = 500$) та вовків ($x_2 = 300$): `array=[500;300]`;
- час початку спостереження за системою – `t_begin=0`, крок інтегрування `step=0.1` дня, тривалість спостереження в днях – `t_end=150`;
- масив, який містить значення часу на інтервалі [`t_begin; t_end`] з кроком `step`: `time=t_begin:step:t_end`;
- функцію `syst_1()`, що містить систему рівнянь (1.7);
- функцію `ode()`, яка використана для обчислення розв'язку системи рівнянь (1.7) методом Рунге-Кутта четвертого порядку.

Збережіть створений файл за ім'ям lab1_1.sce у робочому каталозі;

в) у пункті меню "Выполнение" оберіть "Загрузить в Scilab". У командному вікні Scilab відобразиться виконання програми або, якщо була допущена помилка, буде виведено рядок з попередженням про помилку в певному рядку коду програми. Збережіть побудовані за допомогою функції `plot()` графіки залежностей $x_1(t)$ і $x_2(t)$ та графік $x_2(x_1)$. Призначення та синтаксис функцій `subplot()` та `plot()` наведено у додатку А. Збережіть файл lab1_1.sce.

В результаті виконання файлу сценарію lab1_1.sce (здійснюється натиском клавіші **F5**) буде отримано графіки, що наведені на рис. 1.1.

На правому графіку (залежності хижаків від жертв), зображеному на рис. 1.1, вісь y – популяція хижаків, x – популяція жертв. Спочатку кількість зайців та вовків задана співвідношенням 500:300 (x:y). Вовки та зайці починають розмножуватися, змінюючи це співвідношення. Але, коли вовків стає дуже багато, чисельність зайців не встигає відновлюватися. Як наслідок, у вовків не вистачає їжі, і вони гинуть.

Лістинг 1.1 – Вміст файлу lab1_1.sce

```
//Коефіцієнти взаємодії
a11=0.06; a12=0.0006;
a22=0.24; a21=0.0006;
//Вектор початкових умов системи
array=[500;300];
//Моменти спостереження
t_begin=0; step=0.1; t_end=150;
time=t_begin:step:t_end;
//Функція syst_1 для обчислення
//x1=solution(1)
//x2=solution(2)
function solution=syst_1(t,v)
solution(1)=a11*v(1)-a12*v(1)*v(2);
solution(2)=a21*v(1)*v(2)-a22*v(2);
endfunction
result=ode("rk",array,t_begin,time,syst_1);
//Графік залежності x1 та x2 від часу
subplot(1,2,1);
plot(time,result);
//Графік залежності кількості
//хижаків від кількості жертв
subplot(1,2,2);
plot(result(1,:),result(2,:));
```

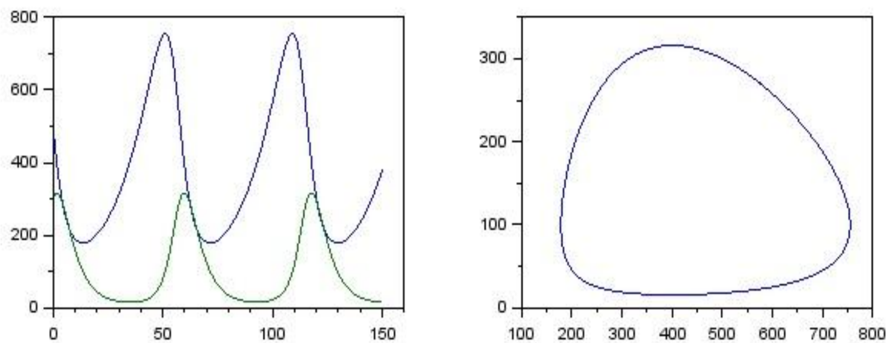

Рисунок 1.1 – Результат виконання файлу lab1_1.sce

1.3.2 Стаціонарний стан екологічної системи

Знайдіть стаціонарний стан системи, тобто таке співвідношення зайців та вовків, за яким чисельність зайців встигає відновлюватися, не дозволяючи вовкам гинути від нестачі їжі. Виконайте наступне:

а) для побудови стаціонарної точки системи на графіку необхідно знайти рішення x_1 та x_2 системи рівнянь (1.7) за умови $x_1' = 0$, $x_2' = 0$. Після підстановки цих значень в (1.7) система приймає вигляд:

$$\begin{cases} a_{11}x_1 - a_{12}x_1x_2 = 0, \\ a_{21}x_1x_2 - a_{22}x_2 = 0. \end{cases} \tag{1.8}$$

Виконайте необхідні перетворення системи рівнянь (1.8) та обчисліть координати стаціонарної точки (x_1, x_2);

б) створіть файл lab1_2.sce, в який скопіюйте вміст файлу lab1_1.sce. Визначте вектор `array_1` з обчисленими значеннями точки стаціонарного стану і обчисліть значення `result1` за допомогою функції `ode()`. Побудуйте графіки залежності $x_1(t)$ і $x_2(t)$ (функція `plot(time,result1(1,:),time,result1(2,:));`) та графік $x_2(x_1)$ (функція `plot(result1(1,:),result1(2,:),"rx");`). Збережіть зміни у файлі lab1_2.sce.

1.3.3 Екологічна система, в якій відсутні зустрічі між хижаками та жертвами

Для моделювання системи, в якій відсутні зустрічі між хижаками та жертвами, створіть файл lab1_3.sce, в який скопіюйте вміст lab1_1.sce. В файлі lab1_3.sce замініть систему рівнянь (1.7) у функції `syst_1` на рівняння (1.4) та (1.5). Задайте тривалість спостереження в днях – `t_end=20`. Побудуйте графіки залежностей $x_1(t)$ та $x_2(t)$.

Примітка. Для відображення назви графіку та підписування осей використайте функцію `xtitle("Name", "Signatureaxis X", "Signatureaxis Y");` Для відображення легенди – інструкцію `legend("Function 1", "Function 2")`. Налаштувати властивості графічного об’єкта та спосіб його відображення можна у вікні

`FigureEditor`, якщо на панелі меню графічного вікна обрати "Правка" – "Свойства осей…" або "Свойства графического окна…".

1.4 Дослідження процесу розповсюдження епідемії

Нехай є віддалений населений пункт, у якому знаходиться H людей, та почалася епідемія. Позначимо через x – кількість людей, які ще здорові, через y – кількість людей, які захворіли, а через z – кількість людей, які вже перехворіли і в них виробився імунітет.

У момент виявлення епідемії $y(0)$ людей уже хворіли, будучи носіями інфекції; $x(0)$ – людей були ще здорові, але потенційно схильні до захворювання; а кількість перехворілих людей обчислюється з співвідношення: $z(0)=H-x(0)-y(0)$.

Відповідно до гіпотези захворювання передається «контактно», тобто при зустрічі, тоді швидкість захворювання здорових людей x пропорційна частоті їх зустрічей із хворими y. Характер захворювання й умови життя в населеному пункті такі, що кожний хворий щодня в середньому передає інфекцію β здоровим людям із кількості H. Захворіла людина через γ днів видужує, набуваючи імунітет.

Тоді диференціальні рівняння розглянутої системи мають вид:

$$\begin{cases} x' = -\beta/H \cdot xy, \\ y' = \beta/H \cdot xy - 1/\gamma \cdot y, \\ z' = 1/\gamma \cdot y. \end{cases} \tag{1.9}$$

1.5 Практичне завдання (друга частина)

1.5.1 Моделювання процесу розповсюдження епідемії

Для моделювання процесу розповсюдження епідемії виконайте наступні дії:

а) для моделювання такої системи зробіть наступні заміни в формулі (1.9): $x_1 = x$, $x_2 = y$, $x_3 = z$. В результаті отримаєте наступну систему рівнянь:

$$\begin{cases} x_1' = -\beta/H \cdot x_1 x_2\,, \\ x_2' = \beta/H \cdot x_1 x_2 - 1/\gamma \cdot x_2\,, \\ x_3' = 1/\gamma \cdot x_2\,. \end{cases} \tag{1.10}$$

б) створіть файл сценарію lab1_4.sce. Розмістить в ньому відповідний до лістингу 1.2 вміст:

- кількість людей в населеному пункті `H=1000`, інтенсивність розповсюдження епідемії 1-а людина за день передає інфекцію 1-й здоровій людині `B=1`, кількість днів, необхідних на одужання `Y=10`;
- вектор початкових умов (`array=[x,y,z]`), елементи якого: `x=900` – кількість здорових людей, `y=90` – кількість хворих, `z=H-x-y` – кількість людей, які одужали;
- вектор часу спостереження: `time=t_begin:step:t_end`; де `t_begin=0` – початок спостереження за системою, `step=0.1` – крок інтегрування, `t_end=40` – тривалість спостереження в днях;
- функція `syst_2()`, яка містить систему рівнянь (1.10);
- функція `ode()` для обчислення `result`;

Лістинг 1.2 – Вміст файлу lab1_4.sce

```
H=1000; //Кількість людей в населеному пункті
B=1;  //Інтенсивність захворювання
Y=10; //Кількість днів на одужання
//Вектор початкових умов системи
x=900; y=90; z=H-x-y;
array=[x; y; z];
//Моменти спостереження
t_begin=0; step=0.1; t_end=40;
time=t_begin:step:t_end;

//Функція syst_2 для обчислення
function solution=syst_2(time_value,value)
 solution(1)=-B/H*value(1)*value(2);
 solution(2)=B/H*value(1)*value(2)-1/Y*value(2);
 solution(3)=1/Y*value(2);
endfunction

result=ode(array,t_begin,time,syst_2);
```

в) доповніть файл lab1_4.sce функцією `plot(time, result)` для побудови графіка залежності `result(time)` (рис. 1.2). Збережіть зміни у файлі та виконайте його.

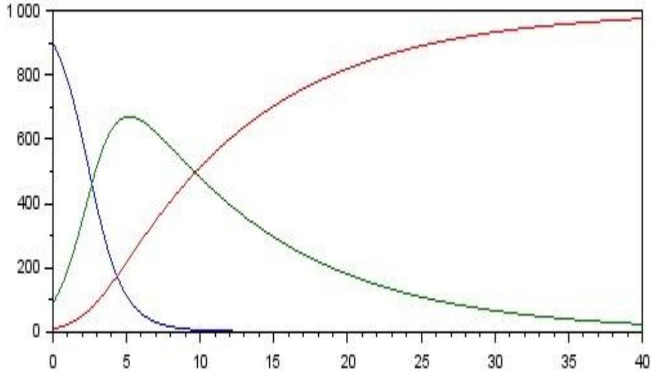

Рисунок 1.2 – Графік залежності кількості здорових, хворих та людей, що одужали, від часу

1.5.2 Моделювання процесу розповсюдження епідемії при ізоляції хворих

Модифікуйте систему рівнянь (1.10) таким чином, щоб в зазначений день k хвору частину населення ізолювали, тобто вони не будуть інфікувати здорових людей. Тоді кількість захворілих поступово почне зменшуватися (одужувати). Найпростіший спосіб моделювання такого стану системи – використання функції Хевісайда (лістинг 1.4). Ця функція дорівнює одиниці, якщо аргумент додатний, або дорівнює нулю, якщо аргумент від’ємний.

Тоді система рівнянь матиме вигляд:

$$\begin{cases} x_1' = -\beta/H \cdot x_1 x_2 \Phi(k-t), \\ x_2' = \beta/H \cdot x_1 x_2 \Phi(k-t) - 1/\gamma \cdot x_2, \\ x_3' = 1/\gamma \cdot x_2. \end{cases} \qquad (1.11)$$

До дня ізоляції хворих, функція Хевісайда не впливає на розповсюдження епідемії, оскільки дорівнює одиниці, але на k-й день хворих людей ізолюють, та функція повертає значення 0.

Для моделювання такої системи визначте окрему функцію `Heaviside()`, в яку в якості аргументів передайте поточне значення часу дослідження та k – день, в який вдалося ізолювати хворих людей. Для цього створіть новий файл lab1_5.sce та розмістить у ньому код, в якому внесіть зміни у функцію `syst_2()` згідно системи рівнянь (1.11), доповнивши її викликом функції `heaviside(time_value,k)` (лістинг 1.3). Збережіть зміни та виконайте програму.

Лістинг 1.3 – Приклад вмісту функції Хевісайда

```
//Функція Хевісайда:
function Heaviside=heaviside(cur_day,spec_day)
 if(cur_day<spec_day)
 Heaviside=1
 else Heaviside=0
 end
endfunction
```

1.6 Контрольні питання

1. Що таке моделювання? Його призначення?
2. Що таке модель? Наведіть деякі типи моделей.
3. Дайте визначення поняттю система та її чотирьом ознакам.
4. Які методи інтегрування Ви знаєте? Чим вони відрізняються?
5. Які допущення в описах систем дозволяли виділити їх із навколишнього середовища?

1.7 Зміст звіту з лабораторної роботи №1

1. Титульний лист, виконаний відповідно з СТП.
2. Мета роботи.
3. Лістинг sce-файлів до п.п.1.3, 1.5.
4. Результати виконання sce-файлів у середовищі Scilab у вигляді графіків.
5. Відповіді на контрольні питання.
6. Висновки за результатами досліджень.

2 ЛАБОРАТОРНА РОБОТА №2
Моделювання лінійних безперервних систем

Мета роботи: опанування навичок моделювання лінійних систем, що описуються звичайними диференціальними рівняннями n-го порядку, дослідження основних властивостей лінійних систем.

2.1 Теоретичні відомості

Для розв’язання звичайного диференціального рівняння (ЗДР) n-го порядку чисельними методами, його необхідно привести до системи, що складається з n диференціальних рівнянь першого порядку.

Методику такого приведення розглянемо на прикладі ЗДР другого порядку:

$$y''(t) + a_1 y'(t) + a_2 y(t) = 0, \tag{2.1}$$

з наступними початковими умовами $y'(0) = y'_0$, $y(0) = y_0$.

Розв’яжемо рівняння (2.1) щодо старшої похідної:

$$\ddot{y}(t) = -a_1 \dot{y}(t) - a_2 y(t). \tag{2.2}$$

Для ЗДР другого порядку введемо дві нові змінні x_1, x_2:

$$y(t) = x_1\ ,\ \ y'(t) = x'_1 = x_2,\ \ y''(t) = x'_2\ . \tag{2.3}$$

Зробивши підстановку нових змінних (2.3) в (2.2), одержимо систему з двох ЗДР першого порядку:

$$\begin{cases} x'_1 = x_2\,, \\ x'_2 = -a_1 x_2 - a_2 x_1, \end{cases} \tag{2.4}$$

$$x_1(0) = y(0),\ \ x_2(0) = y'_0(0)\ .$$

Зверніть увагу, що у правих частинах рівнянь відсутні похідні змінних і система не містить початкової змінної y.

Систему рівнянь (2.4) можна записати у векторній формі:

$$X' = F(X),\quad X(0) = X_0\ , \tag{2.5}$$

де $X = [x_0,\ x_1]^T$, $F = [x_1,\ -a_1 x_1 - a_2 x_0]^T$.

Для дослідження багатьох технічних і природних систем їх можна представити шляхом лінеаризації і допущень як лінійні системи. При цьому рівняння мають лінійний вигляд, тобто в них відсутні нелінійні функції від змінних, а методи дослідження простіші.

Лінійні системи мають властивості однорідності й адитивності. Якщо система лінійна, то вона описується за допомогою рівнянь:

$$\begin{cases} \dfrac{dz(t)}{dt} = a(t)x(t) + b(t)z(t), \\ y(t) = c(t)x(t) + d(t)z(t), \end{cases} \tag{2.6}$$

де $z(t)$ – вектор стану системи, $x(t)$ – вектор зовнішніх впливів на систему, $y(t)$ – вектор виходів системи, $a(t)$, $b(t)$, $c(t)$ і $d(t)$ – матриці відповідних розмірів.

Матриці $a(t)$ і $c(t)$ визначають перехід системи в кінцевий стан та виходи системи в процесі переходу при нульовому початковому стані і одиничному векторі вхідних впливів. Матриці $b(t)$ і $d(t)$ визначають перехід системі з начального стану в кінцевий та вихід системі при нульовому векторі вхідних впливів.

Властивість адитивності виражається таким співвідношенням:

$$Z_1\big(z_1(0), X_1(t)\big) = Z_2\big(z_2(0), X_2(t)\big) + Z_3\big(z_3(0), X_3(t)\big), \tag{2.7}$$

де $z_1(0),\ z_2(0),\ z_3(0)$ – початкові стани систем, причому $z_1(0) = z_2(0) + z_3(0)$; $X_1(t),\ X_2(t),\ X_3(t)$ – зовнішні впливи на систему, причому $X_1(t) = X_2(t) + X_3(t)$.

Властивості однорідності задовольняє таке співвідношення:

$$kZ_1\big(z_1(0), X_1(t)\big) = Z_4\big(z_4(0), X_4(t)\big), \tag{2.8}$$

або $kZ_1\big(z_1(0), X_1(t)\big) = Z_4\big(kz_1(0), kX_1(t)\big)$,

де $z_1(0),\ z_4(0)$ – початкові стани систем, причому $z_4(0) = kz_1(0)$, $X_1(t),\ X_4(t)$ – зовнішні впливи на систему, причому $X_4(t) = kX_1(t)$, k – коефіцієнт пропорційності.

З наведених співвідношень (2.7) та (2.8) очевидна властивість декомпозиції:

$$Z_1\big(z_1(0), X_1(t)\big) = Z_2\big(0, X_1(t)\big) + Z_3\big(z_1(0), 0\big),$$

яку можна використовувати для дослідження лінійності систем.

Таким чином, лінійну систему можна досліджувати частинами. Наприклад, досліджувати її при нульових початкових умовах, з тестовими впливами, потім при нульових вхідних впливах із різними початковими умовами. Або одержати рішення при одиничних вхідних впливах, а будь-які інші рішення отримати шляхом множення його на відповідний коефіцієнт k.

2.2 Практичне завдання

2.2.1 Моделювання коливальної системи

Змоделюйте лінійну коливальну систему, що описується рівнянням (2.1), перетворену в систему рівнянь (2.2).

Для цього створіть файл сценарію lab2_1.sce, в якому задайте такі дані:

- змінні, що відповідають коефіцієнтам a_1 і a_2, тобто `a1=0.5`, `a2=39.4784176`;
- вектор початкових умов `Vector`, значення елементів якого `x1`= $y(0)$ та `x2` = $y'(0)$ оберіть з таблиці 2.1;
- час початку спостереження за системою – `t_begin=0`, крок інтегрування – `step=0.02` та тривалість моделювання – `t_end` (див. табл. 2.1);
- вектор, елементи якого належать інтервалу часу спостереження: `time=t_begin:step:t_end`;
- функцію `syst()`, яка містить систему рівнянь (2.4);
- функцію `ode()` для знаходження розв'язку системи `syst()`, при цьому значення, що повертає функція, збережіть у змінній (матриці) `result`;
- функцію `plot2d()` для побудови графіків залежностей змінної $x_1(t)$, її похідної $x_2(t)$ (`plot2d(time,result')`) та графік залежності $x_2(x_1)$ (`plot2d(result(1,:),result(2,:))`).

Збережіть створений файл за ім'ям lab2_1.sce у робочому каталозі.

Виконайте програму.

Збережіть отримані графіки.

У лістингу 2.1 наведено приклад вмісту файлу lab2_1.sce.

Лістинг 2.1 – Приклад вмісту файлу lab2_1.sce

```
a1=0.5; a2=39.4784176;
//Вектор початкових умов системи
Vector=[4;0.3];
//Моменти спостереження
t_begin=0; step=0.02; t_end=9;
time=t_begin:step:t_end;
//Функція syst для обчислення
function dx=syst(t,x)
dx(1)=x(2);
dx(2)=-a1*x(2)-a2*x(1);
endfunction
result=ode("rk",Vector,t_begin,time,syst);
subplot(1,2,1);
plot2d(time, result');//'-транспонування
subplot(1,2,2);
plot2d(result(1,:),result(2,:));
```

Таблиця 2.1 – Варіанти індивідуальних завдань

№	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
$y(0)$	8	-4	10	3	6	-10	-5	6	-8	5
$y'(0)$	0.4	1.0	-12	0.2	0.7	1.5	0.8	-0.6	1.2	0.5
t_end, c	10	8	12	10	12	8	6	10	12	8

2.2.2 Моделювання системи, що описується диференціальним рівнянням четвертого порядку

У цій частині роботи треба змоделювати систему, яка описується диференціальним рівнянням четвертого порядку:

$$y''''(t) + a_1 y'''(t) + a_2 y''(t) + a_3 y'(t) + a_4 y(t) = f(t), \qquad (2.9)$$

де вхідний вплив описується функцією:

$$f(t) = A\sin(\omega t + \varphi).$$

Для моделювання заданого диференціального рівняння зробіть наступні дії:

а) розв’яжіть диференціальне рівняння четвертого порядку (2.9) відносно старшої похідної:

$$y''''(t) = f(t) - a_1 y'''(t) - a_2 y''(t) - a_3 y'(t) - a_4 y(t)\,.$$

Перетворіть диференціальне рівняння четвертого порядку в систему з чотирьох рівнянь першого порядку, для чого здійсніть наступну заміну змінних:

$$y(t) = x_1\ ,\ y'(t) = x_1' = x_2\,,\ y''(t) = x_2' = x_3\,,$$

$$y'''(t) = x_3' = x_4\ ,\ y''''(t) = x_4'\,.$$

Після перетворення диференціального рівняння четвертого порядку отримаємо систему з чотирьох ЗДР першого порядку:

$$\begin{cases} x_1' = x_2\,, \\ x_2' = x_3\,, \\ x_3' = x_4\,, \\ x_4' = A\sin(\omega t + \varphi) - a_1 x_4 - a_2 x_3 - a_3 x_2 - a_4 x_1\,. \end{cases} \tag{2.10}$$

б) для моделювання в середовищі Scilab створіть файл lab2_2.sce, в якому задайте такі дані (див. табл. 2.2):

- змінні `a1`, `a2`, `a3`, `a4`;
- змінні `A`, `w`, `fi`, які відповідають параметрам A, ω, φ;
- вектор початкових умов `Vector`, що містить $y(0)$, $y'(0)$, $y''(0)$, $y'''(0)$;
- час початку спостереження за системою – `t_begin=0`, крок інтегрування – `step=0.1` та тривалість моделювання – `t_end`;
- вектор часу спостереження: `time=t_begin:step:t_end`;

в) визначте функцію `syst_2()`, яка містить праві частини системи рівнянь (2.10) та розв’яжіть систему диференціальних рівнянь, скориставшись функцією `ode()`;

г) побудуйте за допомогою функції `plot()` графіки змінної $x_1(t)$ (`result(1,:)`) та її похідної $x_2(t)$ (`result(2,:)`);

д) Збережіть зміни у файлі lab2_2.sce.

Примітка. Для виконання завдання за п.п.2.2.2-2.2.4 дані для моделювання обирати згідно варіанту з таблиці 2.2.

Лістинг 2.2 – Приклад вмісту файлу lab2_2.sce

```
a1=0.8; a2=2.0; a3=0.6; a4=0.4;
A=2; w=%pi/4; fi=%pi/8;
vector=[-5; 8; 1; 2];
t_begin=0; step=0.1; t_end=80;
time=t_begin:step:t_end;
function dy=syst_2(t,y)
 dy(1)=y(2);
 dy(2)=y(3);
 dy(3)=y(4);
 dy(4)=A*sin(w*t+fi)-a1*y(4)-a2*y(3)-a3*y(2)-a4*y(1);
endfunction
result=ode("rk",vector,t_begin,time,syst_2);
subplot(121);
plot2d(time,result');
subplot(122);
plot(time,result(1,:),"b",time,result(2,:),"r");
```

Таблиця 2.2 – Варіанти індивідуальних завдань

№	1	2	3	4	5	6	7	8	9	10
a_1	0.5	0.2	0.25	0.75	0.6	0.3	0.06	0.5	0.4	0.2
a_2	20	0.16	1	0.15	1.2	0.8	64	14	0.6	0.4
a_3	4	0.01	0.01	0.025	0.08	0.1	32	0.8	0.012	0.02
a_4	5	0.002	0.04	0.002	0.04	0.12	24	6	0.025	0.015
A	3	6	20	10	4	16	100	40	0.2	1.5
ω	π/2	π/8	π/4	π/3	π/4	π/2	π/2	π	π/6	π/4
φ	π/4	π/6	π/8	π/12	π/8	π/4	–π/8	π/6	π/12	π/10
$y(0)$	5	300	20	–40	–5	–20	–1	1.5	2.5	–4.5
$y'(0)$	0.5	–2	–5	10	8	40	4	2.5	1.5	4
$y''(0)$	1	2	–8	–10	0	–5	–2	– 1	–0.8	–0.2
$y'''(0)$	0	0	0	0	0	0	0	0	0	0
k	5	4	3	2	6	7	4	3	2	5
t_end	50	120	150	100	80	50	10	15	150	120

2.2.3 Перевірка властивості адитивності системи

Для перевірки властивості адитивності системи зробіть наступне:

а) створіть новий файл lab2_3.sce, в який скопіюйте вміст файлу lab2_2.sce. Доповніть файл, в якому за допомогою функції `ode()` обчисліть значення `res_ad_1` (лістинг 2.3) та значення `res_ad_2` (лістинг 2.4). Знайдіть суму значень `res_ad=res_ad_1+res_ad_2`;

б) побудуйте графіки залежностей змінних $x_1(t)$ та її похідних $x_2(t)$ для `res_ad_1` (перше вікно), `res_ad_2` (друге вікно), `res_ad` (третє вікно).

Примітка. Для кращої наочності підберіть різні кольори та вигляд ліній графіків. Виведіть назви графіків;

Лістинг 2.3 – Дослідження системи при нульових початкових умовах, з заданими впливами

```
vector=[0; 0; 0; 0]; A=2;
res_ad_1=ode("rk",vector,t_begin,time,syst_2);
scf();
subplot(221);
plot(time,res_ad_1(1,:),"--m",time,res_ad_1(2,:),"--c");
```

Лістинг 2.4 – Дослідження системи при нульових вхідних впливах із заданими початковими умовами

```
vector=[-5; 8; 1; 2]; A=0;
res_ad_2=ode("rk",vector,t_begin,time,syst_2);
subplot(222);
plot(time,res_ad_1(1,:),"-.m",time,res_ad_1(2,:),"-.c");
```

г) порівняйте графіки `res_ad` з графіками `result`. Для цього виведіть їх у четверте вікно разом (`subplot(224);`). Якщо ці графіки співпадають, то умова адитивності виконується.

2.2.4 Перевірка властивості однорідності системи

Для перевірки властивості однорідності системи зробіть наступне:

а) створіть новий файл lab2_4.sce, в який скопіюйте вміст файлу lab2_2.sce. У файлі обчисліть значення `res_un` з вектором початкових умов та з вхідним впливом, кожний з яких помножений на змінну k (лістинг 2.5) та побудуйте графік (перше вікно);

Лістинг 2.5 – Дослідження системи при початкових умовах та з вхідними впливами, що помножені на k

```
vector=[-5;8;1;2]; A=2;
k=6; //коефіцієнт пропорційності
A=A*k; vector=vector*k;
res_un=ode("rk",vector,t_begin,time,syst_2);
scf(); subplot(211);
plot(time,res_un(1,:),"--m",time,res_un(2,:),"--c");
```

б) матрицю `result` помножте на k (`result_k=result*k`) та побудуйте графіки залежностей $x_1(t)$, $x_2(t)$. Порівняйте графіки `result_k` з графіками `res_un`. Для цього виведіть їх у друге вікно разом (`subplot(212);`). Якщо ці графіки співпадають, то умова однорідності виконується.

2.3 Контрольні питання

1. Які з розглянутих раніше моделей систем є лінійними, нелінійними? Відповідь обґрунтувати.
2. Які властивості має лінійна система? Дайте визначення цих властивостей.
3. Які моделі систем є безперервними та дискретними, детермінованими та стохастичними, стаціонарними та нестаціонарними?
4. Як здійснюється перетворення диференціального рівняння n-го порядку до системи з n рівнянь першого порядку? Наведіть приклад перетворення заданого диференціального рівняння четвертого порядку.

2.4 Зміст звіту з лабораторної роботи № 2

1. Титульний лист, виконаний відповідно з СТП.
2. Мета роботи.
3. Лістинг sce-файлів до п.п.2.2.3, 2.2.4.
4. Результати виконання sce-файлів у середовищі Scilab у вигляді графіків.
5. Відповіді на контрольні питання.
6. Висновки за результатами досліджень.

3 ЛАБОРАТОРНА РОБОТА №3
Моделювання просторово-розподілених процесів

Мета роботи: одержати уяву про моделі просторово розподілених процесів та про їхні властивості, навчитися відтворювати і досліджувати ці моделі на комп'ютері.

3.1 Теоретичні відомості

Більшість відомих у природі явищ і процесів поширюються в часі та просторі. Фізичні закони, що визначають їхню поведінку, мають безупинний характер та описуються диференціальними рівняннями в частинних похідних. Це електричні і магнітні поля, поширення тепла і дифузійні процеси, теорія пружності й ін.

Прикладами таких диференціальних рівнянь є:

- рівняння Пуассона:

$$\nabla^2 V = \frac{\partial V^2}{\partial x^2} + \frac{\partial V^2}{\partial y^2} = f(x, y); \tag{3.1}$$

- рівняння теплового потоку і дифузії:

$$\nabla^2 V = K \frac{\partial V}{\partial t}; \tag{3.2}$$

- хвильове рівняння:

$$\nabla^2 V = \alpha^2 \frac{\partial^2 V}{\partial t^2}. \tag{3.3}$$

Роздивимось одну методику розв'язання рівняння Лапласа, яке є частковим випадком рівняння Пуассона та має вигляд:

$$\nabla^2 V = \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} = 0. \tag{3.4}$$

Формулюємо задачу таким чином. Необхідно знайти в якійсь області (ділянці) Ω на площині xy безперервну функцію $V(x, y)$, яка задовольняє рівнянню (3.4) та приймає на межі Γ області задані значення $V_\Gamma = \varphi(x, y)$.

Така задача відома під назвою задачі Діріхле для рівняння Лапласа. Ця задача є класичною та часто використовується для демонстрації переваг її розв’язування паралельними методами в дисципліні “Паралельні та розподілені обчислення”.

В загальному випадку межа Γ може бути довільною, однак будемо розглядати задачу Діріхле на прямокутній ділянці Ω, сторони якої дорівнюють a і b.

Для розв’язування диференціальних рівнянь у частинних похідних частіше використовують кінцево-різницеві методи, в яких частинні похідні апроксимуються різницевими операторами.

Отже, розглянемо прямокутну ділянку Ω на площині xy з рівномірною (для спрощення) сіткою з кроком h (рис. 3.1).

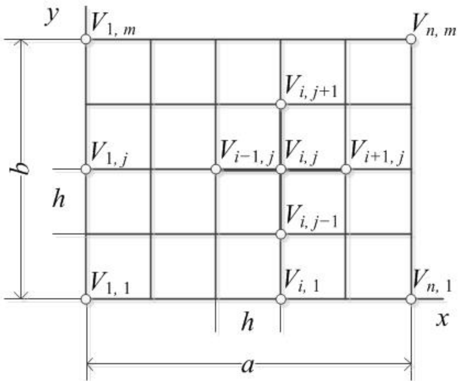

Рисунок 3.1 – Прямокутна ділянка Ω та позначення функції $V_{i,j}$ в вузлах рівномірної сітки

Позначимо значення функції в вузлах сітки через $V_{i,j}$. Тоді часткові похідні в рівнянні Лапласа можна апроксимувати центральними кінцевими різницями другого порядку:

$$\frac{\partial^2 V}{\partial x^2} \approx \frac{V_{i+1,j} - 2V_{i,j} + V_{i-1,j}}{h^2} + O(h^2), \qquad (3.5)$$

$$\frac{\partial^2 V}{\partial y^2} \approx \frac{V_{j,i+1} - 2V_{i,j} + V_{i,j-1}}{h^2} + O(h^2), \qquad (3.6)$$

де $O\left(h^2\right)$ – члени другого порядку малості; $i = 1, 2, \ldots, n$ – індекси точок уздовж осі x; $j = 1, 2, \ldots, m$ – індекси точок уздовж осі y.

Вузли сітки (i, j) внутрішньої області Ω будемо називати внутрішніми, вузли на межі Γ – граничними. Таким чином, індекси внутрішніх вузлів сітки мають значення $i = 2, \ldots, n-1$ та $j = 2, \ldots, m-1$. Якщо вирази (3.5) та (3.6) підставити в рівняння (3.4), то, нехтуючи членами другого порядку малості, рівняння Лапласа для кожного внутрішнього вузла можна записати у вигляді:

$$V_{i,j} = \frac{1}{4}\left(V_{i+1,j} + V_{i-1,j} + V_{i,j+1} + V_{i,j-1}\right). \qquad (3.7)$$

Позначимо задану граничну функцію нижньої грані області Ω як φ_{1x} на межі [0, a], верхньої грані – φ_{2x} на межі [0, a], лівої грані – φ_{1y} на межі [0, b], правої грані – φ_{2y} на межі [0, b]. При цьому повинні виконуватися *граничні умови* $\varphi_{1x}(0) = \varphi_{1y}(0)$, $\varphi_{1x}(a) = \varphi_{2y}(0)$, $\varphi_{2x}(0) = \varphi_{1y}(b)$, $\varphi_{2x}(a) = \varphi_{2y}(b)$.

Таким чином, нам відомі значення функції $V_{i,j}$ на граничних вузлах, тобто відомі значення $V_{i,1}$, $V_{i,m}$, $V_{1,j}$ та $V_{n,j}$ для усіх i та j.

Після введених вище визначень та позначень можна для кожного внутрішнього вузла сітки записати рівняння виду (3.7). Отримаємо систему з $(n-2)\times(m-2)$ лінійних алгебраїчних рівнянь. Приклад такого рівняння для вузла (2,2):

$$V_{2,2} = \frac{1}{4}\left(V_{3,2} + V_{1,2} + V_{2,3} + V_{2,1}\right).$$

Значення функції $V_{i,j}$ в кожному вузлі є середнім врівноваженим чотирьох сусідніх вузлів. Така форма апроксимації виникла завдяки обраному способу апроксимації других похідних рівняння Лапласа центральними різницями другого порядку та зветься чотирьох точковим (або хрестоподібним) шаблоном (рис. 3.1). В залежності від способу апроксимації можна отримати інші шаблони, наприклад, 6-точковий, 8-точковий.

Одним з методів знаходження значень $V_{i,j}$ на внутрішніх вузлах сітки області Ω є ітераційний метод Якобі, який для кожного рівняння (3.7) може бути записаний у вигляді:

$$V_{i,j}^{k+1} = \frac{1}{4}\left(V_{i+1,j}^{k} + V_{i-1,j}^{k} + V_{i,j+1}^{k} + V_{i,j-1}^{k}\right), \tag{3.8}$$

де $k = 0, 1, 2, \ldots$ – номер ітерації.

Щоб розпочати ітераційний процес (першу ітерацію) необхідно задати початкові значення функції $V_{i,j}^{0}$ на внутрішніх вузлах сітки. Якщо нам нічого невідомо про передбачуваний розв’язок, то ці начальні значення можна обирати довільними.

Ітераційний процес закінчується, коли результати обчислення значень функції $V_{i,j}$ у всіх внутрішніх вузлах сітки на даній ітерації відрізняються від значень функції попередньої ітерації не більш ніж припустима похибка ε. Умова закінчення ітераційного процесу визначається за формулою:

$$\max\left|V_{i,j}^{k+1} - V_{i,j}^{k}\right| < \varepsilon, \tag{3.9}$$

де ε - деяке мале, попередньо задане, додатне число.

3.2 Практичне завдання

3.2.1 Розв’язання контрольної задачі

Змоделюйте рівняння Лапласа, що описується рівнянням (3.4), за допомогою ітераційного методу Якобі з граничними функціями області Ω:

$$\varphi_{1x} = \sin(x) - x^3 \sin(1),\ \varphi_{2x} = x^2,\ \varphi_{1y} = 0,\ \varphi_{2y} = y^3. \tag{3.10}$$

Для розв’язання контрольної задачі виконайте наступні дії:

а) створіть файл сценарію lab3_1.sce, в якому задайте такі дані та визначте масиви та змінні:

– висота і ширина прямокутної області Ω $a = 1$, $b = 1$;
– крок дискретизації по осях x та y $h = 0{,}1$;
– кількість точок по осі x $n = a/h + 1$;
– кількість точок по осі y $m = b/h + 1$;

– максимальна припустима похибка $\varepsilon = 10^{-4}$, `eps`=0.0001;

– одновимірний масив x розміром n для зберігання значень по осі x та одновимірний масив y розміром m для зберігання значень по осі y;

– двовимірний масив `V` розміром $n \times m$ і двовимірний допоміжний масив `V1` розміром $n \times m$;

– змінна `dif` – абсолютне значення похибки обчислення функції V в кожному вузлі сітки;

– змінна `err` – максимальна похибка розв'язку;

– змінна `k` – кількість ітерацій;

б) заповніть випадковими числами двовимірний масив `V`, обчисліть значення елементів і заповніть масив `V` на граничних вузлах сітки за формулами (3.10). Привласніть всім елементам масиву `V1` значення елементів масиву `V`;

в) організуйте цикл за умовою (3.9) для обчислення значень елементів матриці `V` на внутрішніх вузлах сітки за формулою (3.8);

г) виведіть на екран обчислені значення матриці `V` та кількість виконаних ітерацій та побудуйте тривимірні графіки вихідної функції $V(x, y)$ (див. рис. 3.2), значення якої є випадкові числа, та її значення після ітераційного процесу. Зробіть спробу інтерпретації розташування осей на графіку та значень розв'язання на граничних вузлах;

д) зробіть оцінку середнього значення кількості необхідних ітерацій, для чого декілька раз запустіть задачу на виконання. Змініть точність розв'язання `eps` на вищу і нижчу за 10^{-4}. Проаналізуйте результати.

Лістинг 3.1 – Код Scilab для розв'язку контрольної задачі

```
h=0.1; a=1; b=1;
x=[0.0:h:a];
y=[0.0:h:b];
n=a/h+1;
m=b/h+1;
eps=0.0001;
V=rand(n,m); // випадкові значення
```

```
 // граничні значення
 V(:,1)=(sin(x)-x.^3*sin(1.0))';   // fi_1x
 V(:,m)=(x.^2)';  // fi_2x
 V(1,:)=0.0;      // fi_1y
 V(n,:)=y.^3;     // fi_2y
for i=1:m
  for j=1:n
    printf(" %5.2f ", V(i,j));
  end
  printf("\n");
end
 subplot(121);
 plot3d(x,y,V,250,30,'x@y@V',[0,2,4]);
 V1=V; //допоміжний масив
 k=0; // кількість ітерацій
 err=1.0;
 while err>eps
   err=0.0;
   for i=2:n-1
     for j=2:m-1
V(i,j)=(V1(i+1,j)+V1(i-1,j)+V1(i,j+1)+V1(i,j-1))/4;
       dif=abs(V(i,j)-V1(i,j))
       if dif>err then err=dif;
       end
     end
  end
  V1=V;
  k=k+1;
end
printf("\n\n k = %d\n", k);
for i=1:m
  for j=1:n
    printf(" %5.2f ", V(i,j));
  end
  printf("\n");
end
subplot(122);
plot3d(x,y,V,250,30,'x@y@V',[0,2,4]);
```

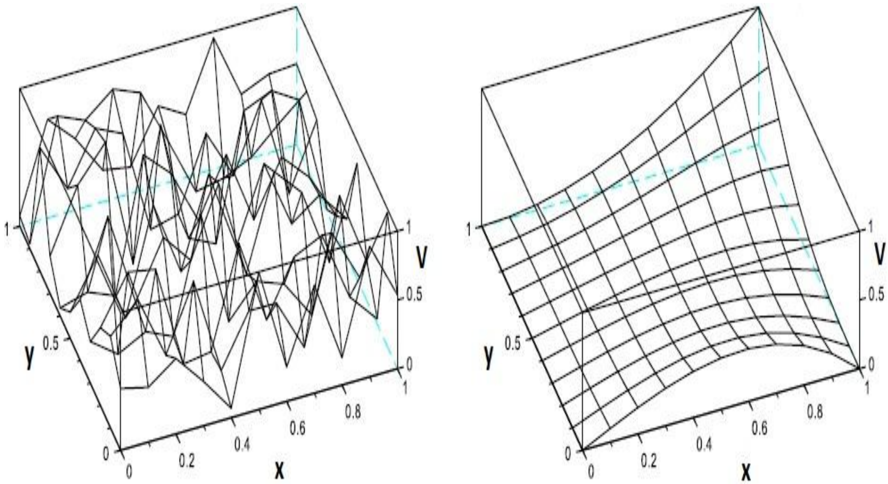

Рисунок 3.2 – Графічний розв’язок контрольної задачі

3.2.2 Індивідуальне завдання

Для виконання індивідуального завдання виконайте наступне:

а) перевірте виконання граничних умов для заданих варіантів задачі згідно з даними, наведеними в табл. 3.1;

б) створіть нові файли lab3_2.sce та lab3_3.sce, в які скопіюйте вміст файлу lab3_1.sce;

в) задайте значення елементів масиву `V` на граничних вузлах сітки за формулами згідно з вихідними даними, наведеними в табл. 3.1. Обчисліть значення елементів матриці `V` на внутрішніх вузлах сітки за формулою (3.8) та побудуйте тривимірні графіки вихідної функції $V(x,y)$.

Примітка. Написання у коді системних змінних, таких як число π, експонента e, починається з символу `%`. Наприклад, `%pi`, `%e`.

Таблиця 3.1 – Варіанти індивідуальних завдань

№		φ_{1x}	φ_{2x}	φ_{1y}	φ_{2y}
1	*a*	x^3	$x+1$	y^2	$\cos y+(2-\cos y)y$
	b	0	$30(1-x^2)$	$30y$	0
2	*a*	$1-x^3$	x^2	$e^y-e^1y^2$	y
	b	$50-\sin\pi x$	0	$50y(1-y^2)$	0
3	*a*	$\sin x-(1+\sin 1)x^3$	$x-1$	$-y^2$	$y-1$
	b	$50x(1-x)$	20	$20y$	$20y^2$
4	*a*	1	$x+1$	$e^y+(1-e^1)y^2$	$y+1$
	b	$50x(1-x)$	$50x(1-x)$	0	$50y(1-y^2)$
5	*a*	$\sin x-x^3\sin 1$	x^2	0	y
	b	$30(1-x)$	$20\sqrt{x}$	$30(1-y)$	$20y$
6	*a*	$2-x^3$	$x-1$	$2e^y-(1+2e^1)y^2$	$1-y$
	b	$50\sin\pi x$	$30\sqrt{x}$	$50\sin\pi y$	$30y^2$
7	*a*	$9x^2+7x+6$	$9x^2-15x-12$	$-10y^2-8y+6$	$-10y^2-30y+22$
	b	0	$40(1-x)$	$40\sqrt{y}$	$20y(1-y)$
8	*a*	$6x^2+4x+3$	$6x^2-12x-9$	$-7y^2-5y+3$	$-7y^2-21y+13$
	b	$e^x-(1+e^1)x^2$	0	$1-y^3$	$y-1$
9	*a*	$e^{x\cdot x}$	$e^{x\cdot x-1}$	$e^{-y\cdot y}$	$e^{1-y\cdot y}$
	b	$-x^3$	$1-x^3$	y^2	y^2-1
10	*a*	x^2	x^2-1	$-y^3$	$1-y^3$
	b	0	x	$\sin y-y^3\sin 1$	y^2

3.3 Контрольні питання

1. Наведіть приклади диференціальних рівнянь, які описуються частинними похідними .
2. Наведіть формули апроксимації частинних похідних різницевими операторами.
3. Що таке шаблон апроксимації? Які шаблони бувають? Який шаблон застосовується в даній задачі
4. Чим визначається кількість ітерацій при розв'язуванні рівнянь у частинних похідних методом Якобі?
5. Що є джерелом похибки розв'язку рівнянь у частинних похідних кінцево-різницевими методами?
6. Що таке граничні умови? Які особливості граничних умов даної задачі?

3.4 Зміст звіту з лабораторної роботи № 3

1. Титульний лист.
2. Мета роботи.
3. Розрахунок перевірки граничних умов для заданих варіантів.
4. Лістинг sce-файлів до п.3.2.2
5. Результати виконання sce-файлів у середовищі Scilab у вигляді графіків.
6. Відповіді на контрольні питання.
7. Висновки за результатами досліджень.

4 ЛАБОРАТОРНА РОБОТА №4
Моделювання систем, що описуються рівняннями теплопровідності

Мета роботи: опанування навичок моделювання систем, що описуються диференціальними рівняннями в частинних похідних, методом приведення до системи звичайних диференціальних рівнянь.

4.1 Теоретичні відомості

У практичних додатках важливу роль відіграє моделювання розповсюдження тепла в деякому середовищі в часі. Швидкість зміни температури в точках середовища описується за допомогою рівняння теплопровідності, яке для одновимірного випадку має вигляд:

$$\frac{\partial T}{\partial t} = a\frac{\partial^2 T}{\partial y^2}, \tag{4.1}$$

де $a = \frac{\lambda}{c\rho}$ – коефіцієнт температуропровідності речовини, з якої складається середовище, та який характеризує швидкість зміни температури в середовищі, м^2/с;

λ – коефіцієнт теплопровідності речовини, Вт/(м·град);

c – питома теплоємність, Дж /(кг·град); (для довідки: Дж=Вт·с);

ρ – щільність речовини, кг/м^3.

Розроблено безліч методів розв’язування рівняння теплопровідності, таких як метод Фур’є, метод розділення змінних, метод кінцевих різниць, аналогічні викладеному в лабораторній роботі № 3. В останньому випадку, одновимірна задача розв’язується на двовимірній сітці, один вимір якої являє товщину простору поширення тепла, а інший - час. У даній лабораторній роботі використовується метод, в якому зміна температури в часі обчислюється за допомогою чисельного інтегрування.

Розглянемо стінку, що складається з однорідної речовини, яка розділяє два середовища з різними температурами. Стінка має товщину L (рис. 4.1). Припустимо, що ширина і висота стінки нескінченні. Температура середовища зліва від стінки може змінюватися в часі за деяким законом $\varphi_1(t)$, а справа – за законом

$\varphi_2(t)$. Функції $\varphi_1(t)$ та $\varphi_2(t)$ назвемо *граничними умовами*. Початкове розподілення температури в точках стінки задається функцією $\varphi(y)$, яку будемо називати *початковими умовами*.

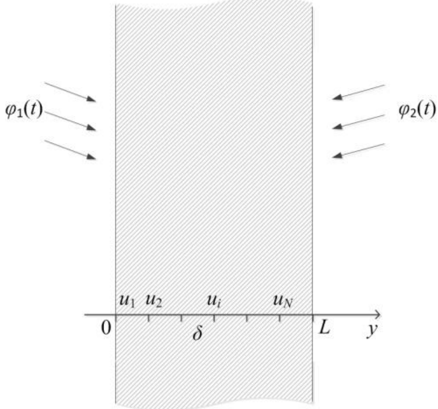

Рисунок 4.1 – Схема розбиття товщини стінки на шари

Будемо позначати температуру в точках стінки через $u(t, y)$. Припускаємо, що площа стінки нескінченна, а речовина стінки однорідна, отже точки стінки, що знаходяться на одній вертикалі, мають однакову температуру. З урахуванням введених позначень, процес поширення температури в стінці буде описуватися рівнянням теплопровідності у вигляді

$$\frac{\partial u}{\partial t} = a\frac{\partial^2 u}{\partial y^2} \tag{4.2}$$

та умовами

$$u(0, y) = \varphi(y)\,,\; u(t,0) = \varphi_1(t)\,,\; u(t, L) = \varphi_2(t)\,. \tag{4.3}$$

Для отримання системи звичайних диференціальних рівнянь, що апроксимують рівняння теплопровідності, розділимо стінку вертикальними лініями на N шарів однакової товщини δ. На перетині цих прямих з віссю y утвориться ряд точок, які пронумеруємо від $i=1$ до $N+1$. Тепер для кожної з позначених точок можна записати рівняння теплопровідності:

$$\frac{\partial u(t,y_i)}{\partial t}=a\frac{\partial^2 u(t,y_i)}{\partial {y_i}^2}\ . \qquad (4.4)$$

Позначимо $u(t,y_i)$ через $u_i(t)$. Частинну похідну $\frac{\partial^2 u(t,y_i)}{\partial {y_i}^2}$ в кожній i-й точці можна апроксимувати за допомогою центральної різниці другого порядку:

$$\frac{\partial^2 u(t,y_i)}{\partial y_i^2}\approx\frac{u_{i+1}(t)-2u_i(t)+u_{i-1}(t)}{\delta^2}+O(\delta^2)\,. \qquad (4.5)$$

Введемо позначення $\mu=\frac{a}{\delta^2}$. Нехтуючи членами другого порядку малості $O(\delta^2)$ та враховуючи введені вище позначення, отримаємо звичайні диференціальні рівняння для кожної внутрішньої точки осі y_i:

$$\frac{du_i(t)}{dt}=\mu\big(u_{i+1}(t)-2u_i(t)+u_{i-1}(t)\big)\ . \qquad (4.6)$$

Таких рівнянь буде стільки, скільки точок було визначено на осі y. Якщо кількість точок N, то всі рівняння разом складуть систему з N диференціальних рівнянь. Наприклад, для чотирьох точок система рівнянь буде виглядати наступним чином:

$$\begin{aligned}
u_1(t)&=\mu\big(u_2(t)-2u_1(t)+\varphi_1(t)\big),\\
u_2'(t)&=\mu\big(u_3(t)-2u_2(t)+u_1(t)\big),\\
u_3'(t)&=\mu\big(u_4(t)-2u_3(t)+u_2(t)\big),\\
u_4'(t)&=\mu\big(\varphi_2(t)-2u_4(t)+\nu_3(t)\big)
\end{aligned} \qquad (4.7)$$

при початкових умовах $u_i(0)=\varphi(y_i)$ та граничних умовах $u(t,0)=\varphi_1(t),\ u(t,L)=\varphi_2(t)$.

Таку систему рівнянь можна розв’язати за допомогою будь-яких методів чисельного інтегрування (наприклад, Рунге-Кутта).

4.2 Дослідження процесів зміни температури в стінці

4.2.1 Розв'язання контрольної задачі (стінка із заліза)

Змоделюйте процес зміни температури в залізній стінці методом приведення до системи звичайних диференціальних рівнянь. Нехай в початковий момент температура у внутрішніх точках стінки розподілена за законом $\varphi(y)=5\sin(2\pi y)$, тобто на границі стінки температура дорівнює нулю, а всередині стінки температура максимальна і дорівнює 5 ºС. Зліва від стінки температура навколишнього середовища змінюється згідно з умовою $\varphi_1(t)=2\cos\left(\pi\frac{t}{T}-\frac{\pi}{2}\right)$, де $T=60$ хв. – час спостереження за процесом (час моделювання). Температура середовища справа від стінки дорівнює нулю, тобто $\varphi_2(t)=0$ ºС.

Для моделювання виконайте наступні дії:

а) створіть файл сценарію lab4_1.sce, в якому задайте наступні дані та визначте змінні:

– товщину стінки $L=0{,}5\,м$;

– температуропровідність заліза $a=23\cdot10^{-6}\,м^2/с$;

– кількість шарів розбиття $n=100$;

– крок інтегрування $h=1$ хвилина;

– час моделювання $T=1$ година;

– вектор часу спостереження: `t=t0:h:T`;

– змінну `dy` для обчислення товщини шару (δ);

– змінну `mu` для обчислення параметра $\mu=a/\delta^2$;

– одновимірний масив `u0` розміром $n+1$ для збереження початкових значень по осі y та $n+1$ масивів `u(i)` для збереження обчислених значень за часом;

б) визначте функцію `wall()`, яка містить праві частини системи рівнянь (4.6), та розв'яжіть систему диференціальних рівнянь (2.8), скориставшись функцією `ode()`. Побудуйте за допомогою функції `mesh(t,x,z)` тривимірний графік змінної `z(t,x)` (див. лістинг 4.1);

Лістинг 4.1 - Код Scilab для розв’язку контрольної задачі

```
   //стінка із заліза
funcprot(0) //
L = 0.5;    //товщина стінки 0,5 м.
n = 100;     //число шарів розбиття
T = 60;     //час моделювання, хв.
h = 1;      //шаг інтегрування, хв.
t0 = 0;
t = t0:h:T;
dy = L/n ;  // товщина шару
a =23.0E-6;//температуропровідність заліза [м*м/с]
s = 60;    //кількість хвилин у годині
a = a*s;   //перерахунок на хвилини
mu = a/dy^2;//параметр
             //завдання початкової умови
y=0;
for i = 1:n+1
   u0(i)= 5*sin(2*%pi*y);
   y=y+dy;
end
       //ліва гранична умова
function y=fi1(t)
  y =2*cos(-%pi/2+%pi*t/T);
endfunction
       //права гранична умова
function y=fi2(t)
  y=0;
endfunction
       //функція обчислення похідних
function du=wall(t,u)
  du(1)= mu*(u(2)-2*u(1)+fi1(t));
  for i=2:n
    du(i)= mu*(u(i+1)-2*u(i)+u(i-1));
  end
  du(n+1)= mu*(fi2(t)-2*u(n+1)+u(n));
endfunction
```

```
z=ode(u0,t0,t,wall);

x=[1:n+1]; // вектор точок розбиття

mesh(t,x,z);

title('IRON WALL');
xlabel('time, min');
ylabel('thickness, 0,5 m');
zlabel('T, deg. C.');
```

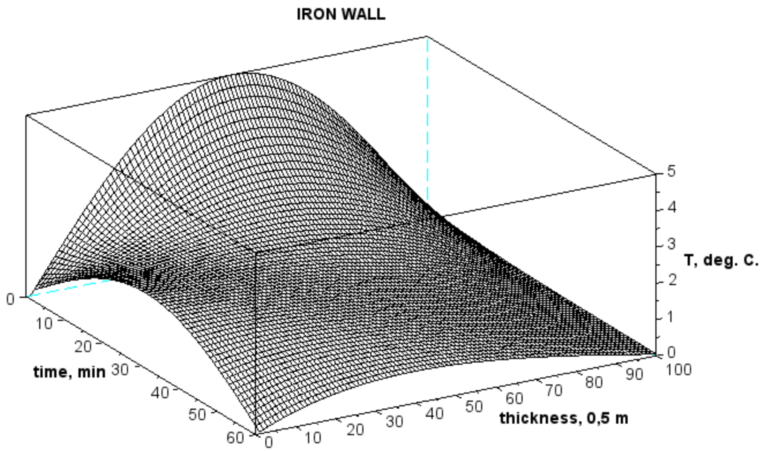

Рисунок 4.2 – Процес зміни температури в стінці із заліза

в) для подальшого дослідження процесів зміни температури в залізній стінці створіть новий файл lab4_2.sce, в який скопіюйте вміст файлу lab4_1.sce, задайте нові дані і отримайте графіки для наступних варіантів (див. табл.4.1).

4.2.2 Індивідуальне завдання

Створіть новий файл lab4_3.sce, в який скопіюйте вміст файлу lab4_1.sce. Задайте дані для розв'язання варіанту задачі згідно з вихідними даними, наведеними в табл. 4.2, 4.3. Побудуйте за допомогою функції `mesh(t,y,z)` тривимірний графік змінної `z(t,y)` за варіантом (див. рис. 4.2-4.12).

Таблиця 4.1 - Дослідження процесів зміни температури в стінці

Варіанти процесів	Початкова температура стінки	Температура середовища	
		зліва від стінки	справа від стінки
Вільне охолодження	$\varphi(y)=5$	$\varphi_1(t)=0$	$\varphi_2(t)=0$
Одностороннє примусове охолодження	$\varphi(y)=5$	$\varphi_1(t)=5$	$\varphi_2(t)=-3$
Одностороннє примусове нагрівання	$\varphi(y)=0$	$\varphi_1(t)=5$	$\varphi_2(t)=0$
Двостороннє примусове нагрівання	$\varphi(y)=0$	$\varphi_1(t)=5$	$\varphi_2(t)=5$
Нагрівання з охолодженням	$\varphi(y)=2$	$\varphi_1(t)=5$	$\varphi_2(t)=-1$

Таблиця 4.2 - Початкові та граничні умови (варіанти 1-10)

№	$\varphi(y)$	$\varphi_1(t)$	$\varphi_2(t)$
1	$6+e^{8,8y}$	$8-e^{0,045t}$	20
2	$10e^{1,1756y}$	$10\cos(\pi t/60)$	$18+7\cos(3\pi/2+\pi t/120)$
3	$10e^{1,946y}$	$10(1+\sin(\pi^2 t/62))$	$70(1+\cos(\pi^2 t/62))$
4	$50(1-\cos(7,1437\pi y))$	$10\sin(\pi^2 t/50)$	$10(4+\cos(\pi^3 t/120))$
5	$20y^2$	$0,2(e^{0,03t}-1)$	$20(1-0,1\sqrt{t})$
6	$100(0,1+y^3)$	$10+\sqrt{t}$	$52,2e^{-0,01t}$
7	$100y$	$1-\cos(\pi t)$	$2e^{-1,5t}$
8	$37y^2$	$3,7\cdot10^{-5}(1-\cos(\pi t/2))$	$3,7\cdot10^{-5}e^{-2t}$
9	$12\sqrt[3]{y}$	$e^{-0,25t}\sin(\pi t)$	$3,5e^{-2t}$
10	$-100\,y$	$15\sin(\pi t/25)$	$-60(1+\sin(\pi t/30))$

Таблиця 4.3 - Вихідні дані для моделювання (варіанти 1-10)

№	**Матеріал**	***a*** 10^{-6}, **м^2/с**	***L*, м**	***T***	***N***	***h***
1	Дерево	0,082	0,3	3 доби	100	3 год.
2	Цегла	0,52	0,5	5 діб	100	1 год.
3	Скло	0,34	0,01	1 год.	100	1 хв.
4	Мідь	111,0	0,07	1 год.	100	1 хв.
5	Повітря	19,0	1,0	3 год.	100	5 хв.
6	Вода	0,143	0,75	10 год.	100	20 хв.
7	Алюміній	84,18	0,02	5 хв.	100	0,1 хв.
8	Нейлон	0,09	0,001	5 хв.	100	0,2 хв.
9	Гума	0,13	0,025	5 хв.	100	0,1 хв.
10	Свинець	23,6	0.6	1 год.	100	1 хв.

4.3 Контрольні питання

1. Наведіть формули апроксимації рівняння теплопровідності.
2. Чим визначається порядок системи звичайних диференціальних рівнянь для моделювання задачі теплопровідності за наведеною методикою?
3. Назвіть основні джерела похибок, які впливають на результат дослідження.
4. Назвіть переваги й недоліки даної методики моделювання.

4.4 Зміст звіту з лабораторної роботи № 4

1. Титульний лист.
2. Мета роботи.
3. Лістинг sce-файлів до п.4.2.2.
4. Результати виконання sce-файлів у середовищі Scilab у вигляді графіків.
5. Відповіді на контрольні питання.
6. Висновки за результатами досліджень.

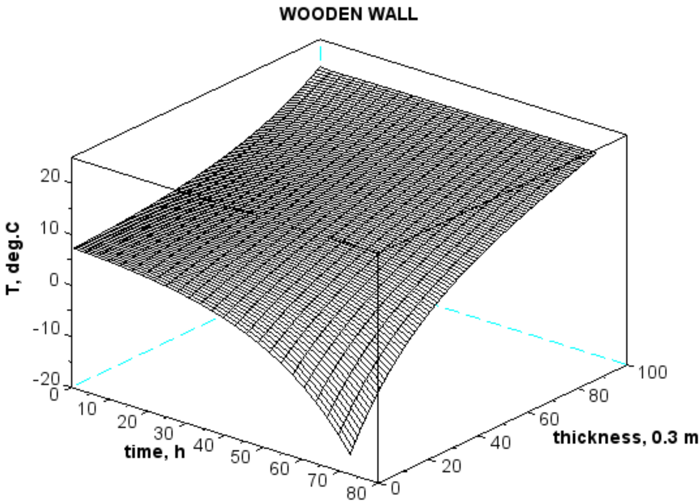

Рисунок 4.3 – Процес зміни температури в стінці з дерева (варіант 1)

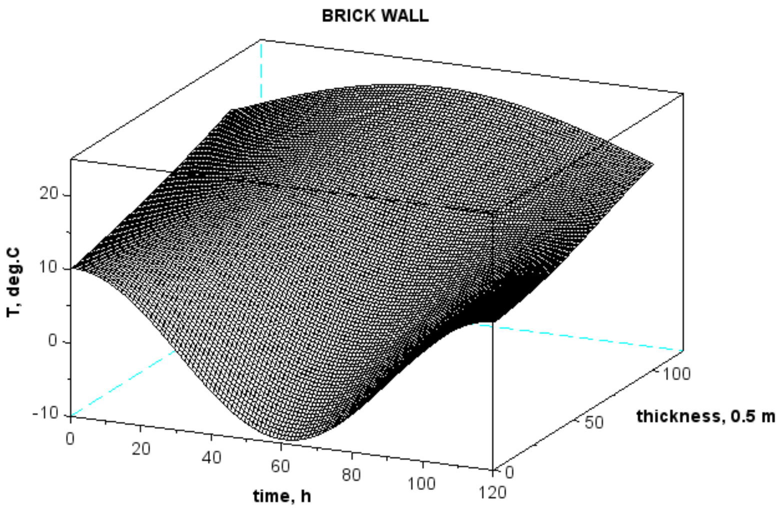

Рисунок 4.4 – Процес зміни температури в стінці з цегли (варіант 2)

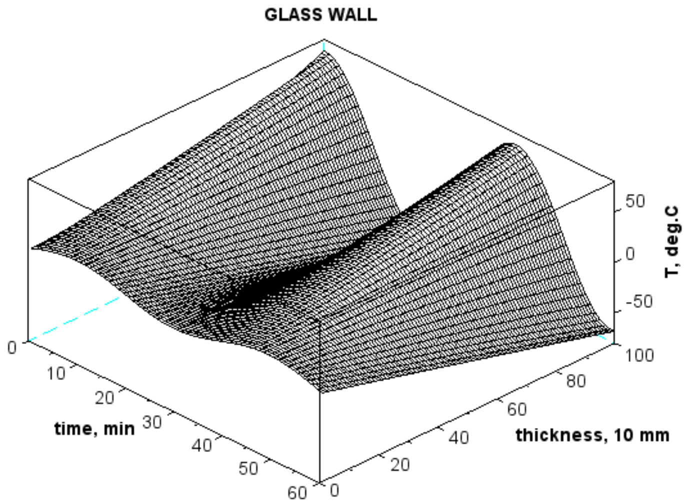

Рисунок 4.5 – Процес зміни температури в стінці зі скла (варіант 3)

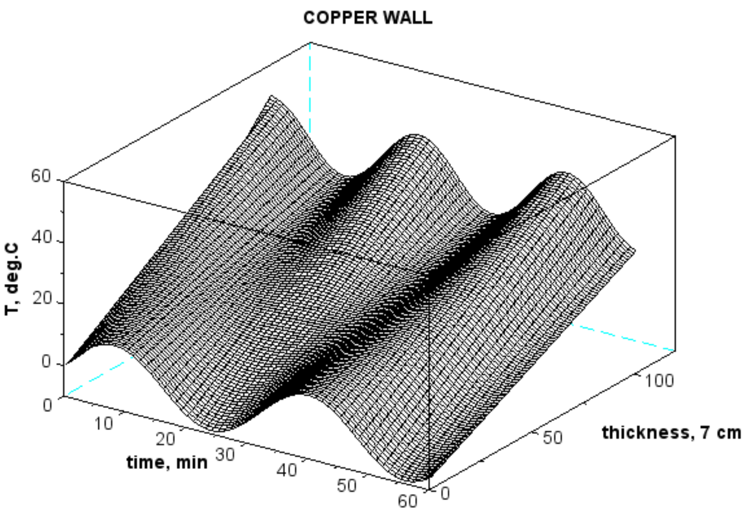

Рисунок 4.6 – Процес зміни температури в стінці з міді (варіант 4)

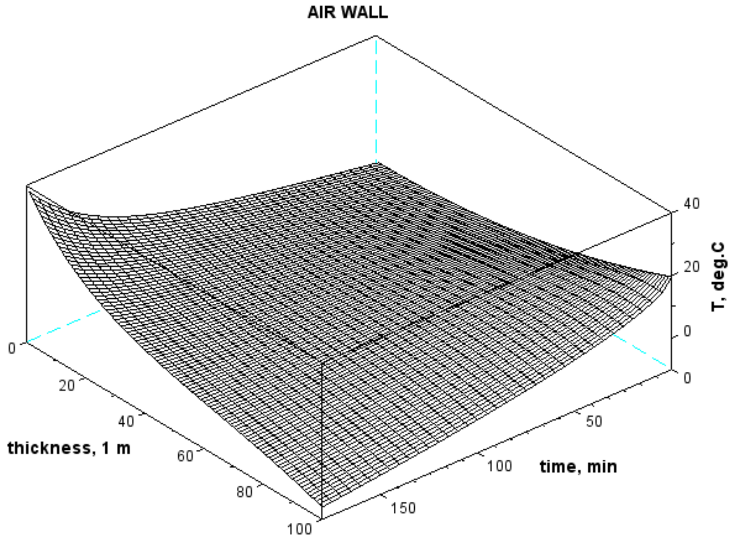

Рисунок 4.7 – Процес зміни температури в стінці з повітря (варіант 5)

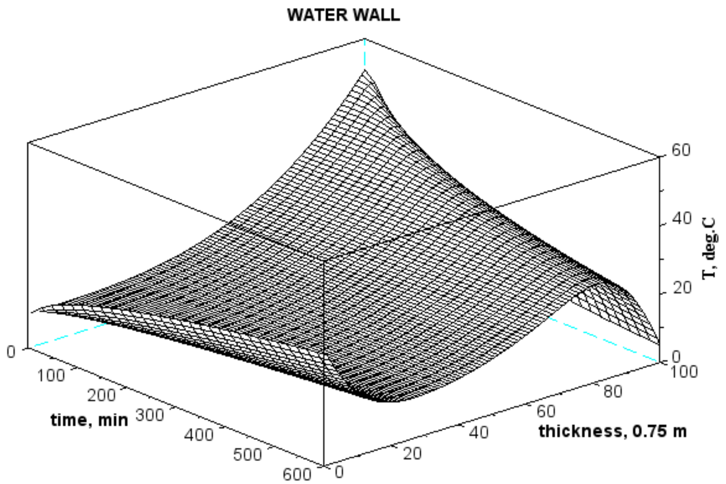

Рисунок 4.8 – Процес зміни температури в стінці з води (варіант 6)

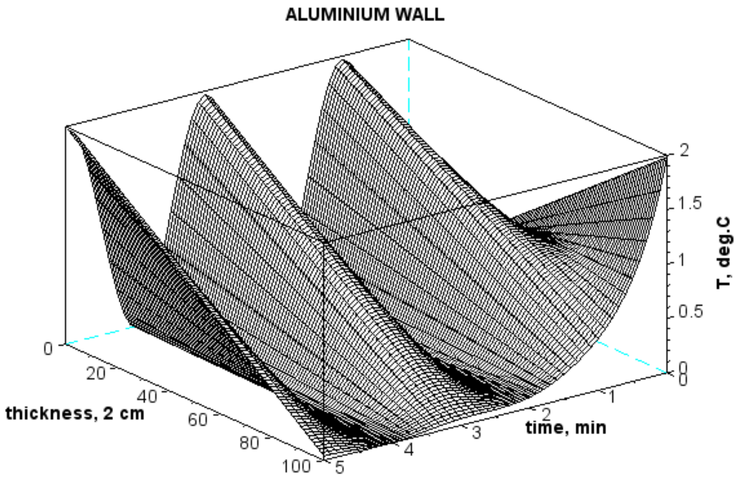

Рисунок 4.9 – Процес зміни температури в стінці з алюмінію (варіант 7)

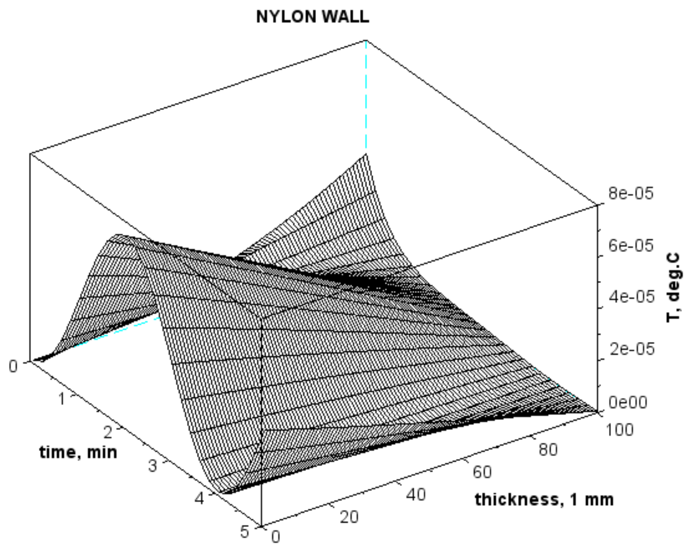

Рисунок 4.10 – Процес зміни температури в стінці з нейлону (варіант 8)

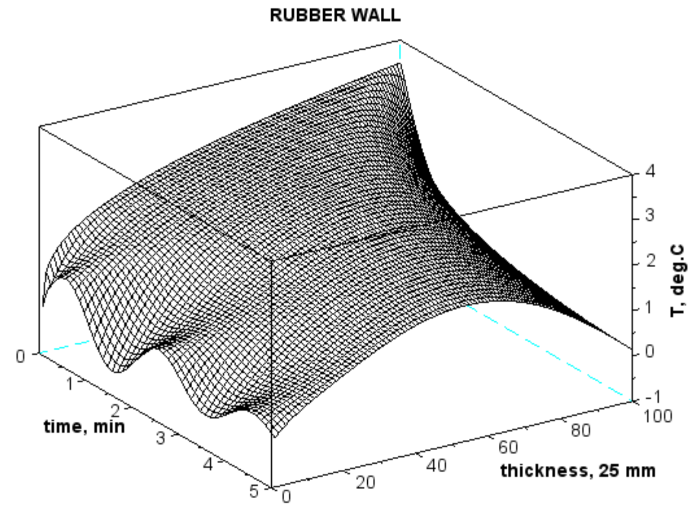

Рисунок 4.11 – Процес зміни температури в стінці з гуми (варіант 9)

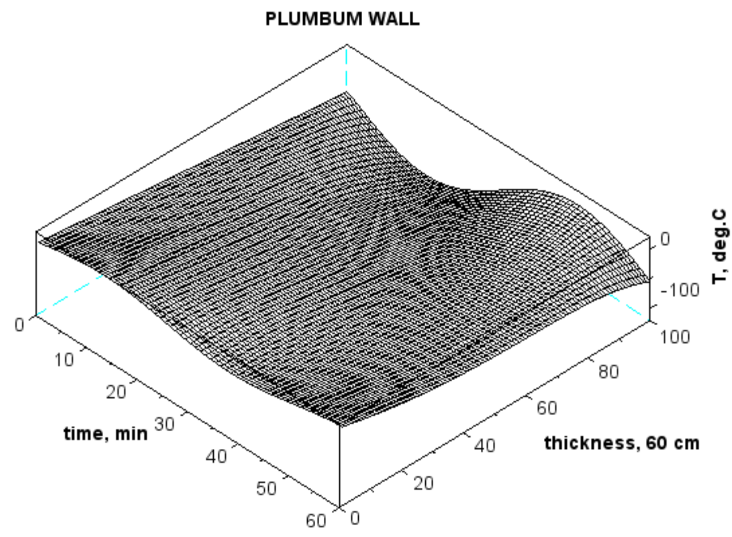

Рисунок 4.12 – Процес зміни температури в стінці з свинцю (варіант 10)

5 ЛАБОРАТОРНА РОБОТА №5
Моделювання систем масового обслуговування

Мета роботи: вивчення методики моделювання та дослідження систем масового обслуговування (СМО) на прикладі моделювання найпростіших систем.

5.1 Теоретичні відомості

5.1.1 Поняття потоків в СМО

Системи масового обслуговування (СМО) – це системи, функціонування яких можна представити як процес обслуговування деякого потоку вимог, подій, заявок одного типу.

Потік вимог – послідовність вимог, які необхідно обслужити.

Вхідний потік – потік вимог, які надходять до обслуговуючої системи. *Вихідний потік* – потік вимог, які виходять з обслуговуючої системи. *Джерело вимог* – першопричина виникнення вимог, незалежно від їх фізичної природи.

Інтенсивність потоку – це середнє число вимог, що виникають в одиницю часу.

При масовому надходженні вимог в систему обслуговування в СМО можуть виникнути *черги*.

Стаціонарний потік вимог – це потік, інтенсивність якого не залежить від часу.

Ординарний потік вимог – це потік, в якому вимоги виникають по одному.

Потік без післядії – це потік вимог, кількість вимог якого, за умови їх існування в інтервалах часу, що не перетинаються, не залежить одна від одної. Відсутність післядій значить що вимоги надходять до системи незалежно одна від одної.

Найпростіший потік вимог – це потік, що одночасно стаціонарний, ординарний та без післядії.

Рекурентний потік - це такий потік вимог, якщо він стаціонарний, ординарний, а інтервали часу між вимогами є випадковими величинами з однаковим (довільним) розподілом.

Окремі випадки рекурентного потоку: *найпростіший* потік, коли інтервали часу між подіями розподілені за експоненціальним законом, та *нормальний* потік, коли інтервали часу між подіями розподілені за нормальним (гаусовим) розподілом.

5.1.2 Алгоритм роботи одноканальної СМО з обмеженим часом очікування

Розглянемо одноканальну систему, в яку на обслуговування надходять заявки.

Інтервал часу між двома послідовними заявками τ є випадковою величиною з заданим законом розподілу $f(\tau)$. Час зайнятості каналу (час обслуговування) τ^F також є випадковою величиною із законом розподілу $f\left(\tau^F\right)$.

Якщо наступна заявка, що надійшла, застає канал зайнятим, то вона очікує звільнення каналу протягом часового інтервалу, не більш ніж τ^W, який теж є випадковою величиною розподіленою за законом $f\left(\tau^W\right)$. Якщо за цей час канал звільняється, то він відразу починає обслуговувати заявку, інакше – заявка залишає систему (отримує відмову). Така система зветься *СМО з обмеженим часом очікування*.

В результаті моделювання такої системи необхідно отримати характеристики якості обслуговування:

- середню кількість обслугованих заявок;
- середню кількість заявок, які отримали відмову;
- середній час очікування;
- середній час обслуговування.

Алгоритм моделювання СМО та організацію роботи системи надано на рис. 5.1.

Процес функціонування системи розглядається в інтервалі часу $[0,T]$.

Зробимо деякі *позначення* моделі системи:

а) t_i^E – момент надходження i -ої заявки. Заявки, що надійшли в момент часу $t_i^E \geq T$ в моделі не розглядаються;

б) t^B – момент часу початку обслуговування. Якщо канал вільний, то обслуговування чергової заявки починається в момент її появи, тобто $t^B = t^E$, канал вважається зайнятим;

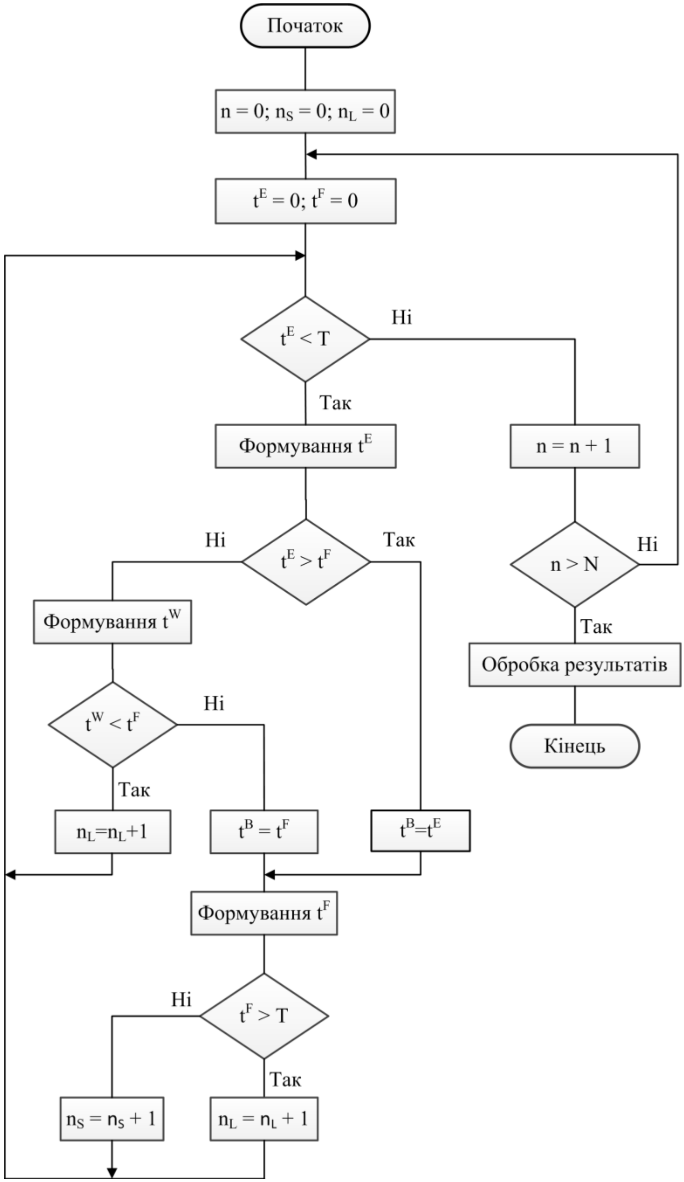

Рисунок 5.1 – Блок-схема алгоритму моделювання СМО з обмеженим часом очікування

в) t^F – момент часу звільнення каналу (закінчення обслуговування). Якщо момент закінчення обслуговування $t^F > T$, вважаємо що заявка отримала відмову;

г) t^F – момент часу закінчення очікування поточної заявки. Якщо $t^W > t^F$, то заявка починає обслуговуватись, причому $t^B = t^F$ і система знову вважається зайнятою;

д) заявка вважається обслуженою, якщо час закінчення обслуговування $t^F < T$;

е) заявка отримала відмову, якщо $t^W < t^F$;

ж) заявка отримала відмову, якщо для даної заявки час початку обслуговування $t^B < T$, а час закінчення обслуговування $t^F < T$.

Нехай λ – інтенсивність потоку заявок, μ – інтенсивність обслуговування каналу, ξ – інтенсивність очікування заявок. Будемо вважати, що час між появою заявок, час обслуговування та час чекання для кожної заявки розподілені за експоненціальним законом:

$$f(\tau) = \lambda e^{-\lambda\tau}, \; f(\tau^F) = \mu e^{-\mu\tau}, \; f(\tau^W) = \xi e^{-\xi\tau}. \qquad (5.1)$$

Обчислити випадкові значення, розподілені за експоненціальним законом, можна за формулою:

$$z = -\frac{1}{\theta}\ln(1 - x), \qquad (5.2)$$

де x – випадкові *рівномірно* розподілені числа. Якщо діапазон розподілення $x \in (0,1)$, то таке співвідношення еквівалентне виразу

$$z = -\frac{1}{\theta}\ln x. \qquad (5.3)$$

Таким чином, випадкові моменти появи заявок, закінчення обслуговування та закінчення очікування можна обчислити за формулами:

$$t_{i+1}^E = t_i^E - \frac{1}{\lambda}\ln x, \; t^F = t^B - \frac{1}{\mu}\ln x, \; t^W = t^E - \frac{1}{\xi}\ln x. \qquad (5.4)$$

Відмітимо, що за допомогою даної моделі можна моделювати і інші типи СМО. При необмеженому збільшенні інтенсивності очікувань ξ, що означає те, що заявки інтенсивніше залишають систему, не дочекавшись обслуговування, система з обмеженим часом очікування за своїми характеристиками наближається до системи з відмовами. З іншого боку, при необмеженому зменшенні інтенсивності очікувань ця модель системи наближається до СМО з необмеженим очікуванням (з необмеженою чергою).

5.2 Практичне завдання - моделювання одноканальної СМО

Розробіть програму моделювання СМО з обмеженим часом очікування за допомогою пакету Scilab. Для цього зробіть наступне:

а) створіть файл сценарію lab5_1.sce, в якому визначте такі дані:

`T` – тривалість однієї реалізації $T = 1000$;

`N` – загальна кількість реалізацій $N = 100$;

`L` – інтенсивність потоку заявок $\lambda = 0{,}4$;

`M` – інтенсивність "обслуговувань" каналу $\mu = 0{,}6$;

`K` – інтенсивність очікувань заявок $\xi = 1{,}2$;

`n` – номер поточної реалізації;

`ne` – загальна кількість заявок;

`nS` – кількість обслугованих заявок;

`nL` – кількість заявок, що отримали відмову в обслуговуванні;

`te` – момент надходження i-ої заявки;

`tb` – момент часу початку обслуговування;

`tf` – момент часу звільнення каналу (закінчення обслуговування);

`tw` – момент часу закінчення очікування поточної заявки;

б) розробіть програму моделювання СМО з обмеженим часом очікування з заданими параметрами, блок-схема якої наведена на рис. 5.1. В результаті виконання коду лістингу 5.1 отримаємо приблизно такі результати: середня кількість заявок в одній реалізації – 400, середня кількість обслугованих заявок за одну реалізацію – 281, середня кількість заявок, що отримали відмову в обслуговуванні – 119;

Лістинг 5.1 – Реалізація СМО з обмеженим часом очікуванням

```
funcprot();
function z=f(nu,tc)  // nu –інтенсивність потоку
  z=tc-log(rand())/nu;
endfunction
L=0.4; M=0.6; K=1.2;//лямбда, мю, ксі
T = 1000;    //тривалість однієї реалізації
n=0;         //номер поточної реалізації
N = 100;     //загальна кількість реалізацій
ne = 0;      // кількість заявок, що надійшли
nS = 0;      //кількість обслугованих заявок
nL = 0; // кількість заявок, що отримали відмову
while n < N
te = 0.0;    //момент часу надходження заявки
tf = 0.0;    //момент часу звільнення каналу
while te<T
  te = f(L,te) ;
  ne = ne + 1; // кількість заявок
  if te>tf then
    tb = te;  //момент початку обслуговування
    tf=f(M,tb); //момент закінчення обслуговування
    if tf>T then
      nL = nL + 1; // кількість відмов
    else
      nS = nS + 1; // кількість обслуговувань
    end
  else
    tw=f(K,te);//tw – момент часу, який очікує заявка
    if tw<tf then
      nL = nL +1; // кількість відмов
    else
      tb = tf;
      tf=f(M,tb); // час виконання нової заявки
      if tf> T then
         nL = nL + 1; //кількість відмов
      else
```

```
          nS = nS + 1; // кількість обслуговувань
        end
      end
    end
  end
      n = n+1; // кількість реалізацій
  end
  printf("ne=%.2f\n",ne/N);
  printf("nS=%.2f nL=%.2f\n",nS/N,nL/N);
```

в) створіть новий файл сценарію lab5_2.sce, в якому змоделюйте систему з обмеженим часом очікуванням з параметрами λ, μ, ξ, які виберіть з табл. 5.1 згідно заданого варіанту;

Таблиця 5.1 – Значення інтенсивностей потоку

№	λ	μ	ξ
1	0,2	0,25	1,3
2	0,5	0,7	1,4
3	0,6	0,8	1,6
4	0,7	1,0	2,0
5	0,8	1,1	1,6
6	0,3	0,5	1,0
7	0,25	0,4	0,8
8	0,75	1,5	3,0
9	0,1	0,15	0,3
10	0,16	0,2	0,5

г) створіть новий файл сценарію lab5_3.sce, в якому змоделюйте систему з відмовами з параметрами згідно варіанту;

д) створіть новий файл lab5_4.sce, в якому змоделюйте систему з необмеженою чергою з параметрами згідно варіанту. В програмі підрахуйте час перебування всіх заявок в системі, визначте середню кількість заявок в системі і середній час перебування заявки в системі за формулами:

$$L = \frac{1}{T}\sum_i t_i \quad , \qquad W = \frac{1}{K}\sum_i t_i \, , \tag{5.4}$$

де t_i – час перебування i-ої заявки в системі `(tf-te)`;

L – середня кількість заявок в системі;

W – середній час перебування заявки в системі;

K – середнє число заявок, що надійшли під час моделювання T.

5.3 Контрольні питання

1. Дайте визначення потоку вимог, інтенсивності потоку.
2. Що таке найпростіший потік вимог? Приклади.
3. Як розподілені інтервали часу між вимогами в найпростішому потоці?
4. Складіть приклад розміченого графа та системи рівнянь Колмогорова для системи, у якій три можливих стани, причому, система може переходити з кожного стану в будь-який з двох інших.
5. Поясніть зміст фінальних ймовірностей станів СМО. У якому випадку вони існують? Як це визначити для конкретної СМО?

5.4 Зміст звіту з лабораторної роботи № 5

1. Титульний лист.
2. Мета роботи.
3. Лістинг sce-файлів до п.5.2 (в, г, д).
4. Результати виконання sce-файлів у середовищі Scilab для СМО з обмеженим часом очікування, з відмовами, з необмеженою чергою.
5. Відповіді на контрольні питання.
6. Висновки за результатами досліджень.

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Алексеев, Е.Р., Чеснокова, О.В., Рудченко, Е.А. Scilab: Решение инженерных и математических задач / Е.Р. Алексеев, О.В. Чеснокова, Е.А. Рудченко. – М.: ALT Linux; БИНОМ. Лаборатория знаний, 2008. – 260 с.

2. Campbell, S.L., Chancelier, J.-P., Nikoukhah, R. Modeling and Simulation in Scilab/Scicos / S.L. Campbell, J.-P. Chancelier, R. Nikoukhah. – NewYork: Springer, 2006. – 313 p.

3. Томашевський, В.М. Моделювання систем / В.М. Томашевський. - К.: Видавнича група BHV, 2005. – 352 с.

4. Кельтон, В., Лоу, А. Имитационное моделирование / В. Кельтон, А. Лоу. – 3-е изд., испр.; СПб: Питер; К.: Издат. группа BHV, 2004. – 847 с.

5. Клейнрок, Л. Теория массового обслуживания / Л. Клейнрок. – М.: Машиностроение, 1998. – 432 с.

6. Офіційний сайт Scilab [Електронний ресурс]. – Режим доступу: https://www.scilab.org/

7. Згуровський, М.З., Панкратова, Н.Д. Основи системного аналізу / М.З. Згуровський, Н.Д. Панкратова. – К.: Видавнича група BHV, 2007. – 544 с.

8. Самойленко, Н.И., Скоков, Б.Г. Исследование операций (Математическое программирование. Теория массового обслуживания) / Н.И. Самойленко, Б.Г. Скоков: Уч. пособие. – Харьков: ХНАГХ, 2005. – 176 с.

9. Писарук, Н.Н. Исследование операций / Н.Н. Писарук. – Минск: БГУ, 2015. – 292 с.

10. Чуйко, Г.П., Дворник, О.В., Яремчук, О.М. Математичне моделювання систем і процесів: навчальний посібник / Г.П. Чуйко, О.В. Дворник, О.М. Яремчук. – Миколаїв: Вид-во ЧДУ імені Петра Могили, 2015. – 244 с.

11. Литвинов, А.Л. Теорія систем масового обслуговування / А.Л. Литвинов: навч. посібник. – Харків. нац. ун-т міськ. госп-ва ім. О.М. Бекетова. – Харків: ХНУМГ ім. О.М. Бекетова, 2018. – 141 с.

ДОДАТОК А

Таблиця А.1 - Вбудовані функції Scilab

Назва функції	***Опис***	***Синтаксис***
1	***2***	***3***
ode	Функція для обчислення звичайних диференційних рівнянь	**ode**("*type*", y0, t0, t, f), **type** – тип методу, що використовується для обчислення, наприклад "rk" – метод Рунге-Кутта 4 порядку; **y0** –вектор початкових умов; **t** – вектор моментів часу; **f** – зовнішня функція (права частина рівняння)
plot	Функція побудови графіків, що описані рівняннями вигляду: ***y***=***f***(***x***)	**plot**(x, y, [*xcap*, *ycap*, *caption*]), **x –** масив абсцис; **y** – масив ординат; ***xcap, ycap, caption*** – підписи осей та графіка відповідно.
plot2d	Функція побудови двовимірних графіків	**plot2d** ([*flag*], x, y’, *key*1=*value*1,…]), **flag**–два символи: n– нормальна вісь; l – логарифмічна вісь; **x –** масив абсцис; **y** – масив ординат; ***keyi=valuei*** – властивості графіка
plot3d	Функція побудови тривимірного графіка	**plot3d**(x, y, z, *a*,*b*,‘*X@Y@Z*’,[*p*1,*p*2,*p*3]), **x, y** – масиви, що задають прямокутну сітку; **z** – матриця значень функції (апліката); ***a, b*** – сферичні координати кута огляду на графік; ‘***X@Y@Z***’ – підписи осей; ***p*1** – колір поверхні; ***p*2** – масштаб графіка; ***p*3** – наявність рамки навколо графіка

Кінець таблиці А.1

1	*2*	*3*
mesh	Побудова тривимірного графіка	**mesh**(x, y, z)**,** **x, y** – масиви, що задають прямокутну сітку; **z** – матриця значень функції (апліката)
subplot	Функція побудови декількох графіків в одному графічному вікні	**subplot**(m, n, p), **m –** кількість вікон по вертикалі; **n** – кількість вікон по горизонталі; **p** – номер вікна, в який буде виведено графік.
scf	Створення нового графічного вікна і вибір його поточним	**scf**( )