

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики

(інститут)

Факультет інформаційних технологій

(факультет)

Кафедра Програмного забезпечення комп'ютерних систем

(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня

магістра

(назва освітньо-кваліфікаційного рівня)

| | | | |
|--------------------|---|--|--|
| студента | <i>Яшкіна Ростислава Ігоровича</i> | | |
| | (ПІБ) | | |
| академічної групи | <i>122М-20-1</i> | | |
| | (шифр) | | |
| спеціальності | <i>122 Комп'ютерні науки</i> | | |
| | (код і назва спеціальності) | | |
| освітньої програми | <i>«122 Комп'ютерні науки»</i> | | |
| | (назва освітньої програми) | | |
| на тему: | <i>Розробка та дослідження ефективності впровадження програмного забезпечення знаходження власних чисел та функцій еліптичного диференціального оператора для тіл обертання</i> | | |

Р.І. Яшкін

| Керівники | Прізвище, ініціали | Оцінка за шкалою | | Підпис |
|---------------------------------------|-----------------------------|------------------|---------------|--------|
| | | рейтинг овою | інституційною | |
| розділів кваліфікаційної роботи | | | | |
| спеціальний | <i>Проф. Бердник М.Г.</i> | | | |
| економічний | <i>Доц. Касьяненко Л.В.</i> | | | |

| | | | | |
|-----------|--|--|--|--|
| Рецензент | | | | |
|-----------|--|--|--|--|

| | | | | |
|----------------|------------------------|--|--|--|
| Нормоконтролер | <i>Доц. Реута О.В.</i> | | | |
|----------------|------------------------|--|--|--|

Дніпро
2022

**Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»**

ЗАТВЕРДЖЕНО:

Завідувач кафедри
Програмного забезпечення комп'ютерних систем
_____ (повна назва)

_____ **І.М. Удовик** _____

(підпис)

« » _____ 20 ____ року

ЗАВДАННЯ

на виконання кваліфікаційної роботи магістра

спеціальності _____ *122 Комп'ютерні науки* _____
(код і назва спеціальності)

студенту 122м-20-1 _____ Яшкіну Ростиславу Ігоровичу _____
(група) (прізвище та ініціали)

Тема кваліфікаційної роботи _____ *Розробка та дослідження ефективності
впровадження програмного забезпечення знаходження власних чисел і функцій
еліптичного диференціального оператора для тіл обертання* _____

1 ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Наказ ректора НТУ «Дніпровська політехніка» від 10.12.2021 р. № 1036 -с

2 МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Об'єкт досліджень – процес чисельного обчислення власних чисел та функцій.

Предмет досліджень – програмні засоби та алгоритми для чисельного обчислення власних чисел та функцій еліптичного диференціального оператора.

Мета НДР – підвищення точності власних чисел та функцій еліптичного оператора для тіл обертання знайдених за допомогою чисельних методів.

Вихідні дані для проведення роботи – теоретичні та експериментальні дослідження, методи скінчених елементів і Гальоркіна, методи обробки експериментальних даних.

3 ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

Наукова новизна результатів дипломної роботи полягає в побудові математичної моделі знаходження власних чисел та функцій еліптичного диференціального оператора для тіл обертання, на основі якої була розроблена комп'ютерна модель.

Завдяки результатам отриманих під час моделювання удосконалено процедуру пошуку власних чисел та функцій для тіл обертання, що дозволяє збільшити точність розрахунків.

Практична цінність: розроблене програмне забезпечення дозволяє легко та зручно вирішувати задачу пошуку власних чисел та функцій, а одержані за його допомогою дані дозволяють обрати необхідну кількість елементів триангуляції для знаходження рішення із заданою точністю.

4 ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Результати досліджень мають бути подані у вигляді, що дозволяє побачити та оцінити безпосереднє використання запропонованих методів. В результаті роботи повинен бути розроблений програмний комплекс для розрахування власних чисел та функцій еліптичного диференціального оператора для тіл обертання.

5 ЕТАПИ ВИКОНАННЯ РОБІТ

| Найменування етапів робіт | Строки виконання робіт (початок – кінець) |
|---|---|
| Аналіз теми та постановка задачі | 12.09.2021-30.09.2021 |
| Комп'ютерне моделювання власних чисел та функцій еліптичного диференціального оператора | 01.10.2021-31.10.2021 |
| Використання програми та аналіз отриманих результатів | 01.11.2021-20.12.2021 |

6 РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ ТА ЕФЕКТИВНІСТЬ

Економічний ефект від реалізації результатів роботи очікується позитивним завдяки розробці алгоритмів та програмного забезпечення для розрахунку власних чисел та функцій для тіл обертання, що дозволяє зменшити витрати для їх знаходження із заданою точністю.

Соціальний ефект від реалізації результатів роботи очікується позитивним, так як завдяки цьому буде полегшена робота науковців і інженерів, що дозволить прискорити та здешевити розробки і дослідження в даній області.

Завдання видав

_____ (підпис)

Бердник М.Г.

_____ (прізвище, ініціали)

Завдання прийняв до виконання

_____ (підпис)

Яшкін Р.І.

_____ (прізвище, ініціали)

Дата видачі завдання: 10.09.2021 р.

Термін подання дипломного проекту до ЕК 25.01.2022 р.

РЕФЕРАТ

Пояснювальна записка: 78 стор., 49 рис., 4 таблиці, 3 додатка, 27 джерел.
Об'єкт дослідження: процес чисельного обчислення власних чисел та функцій.

Предмет дослідження: програмні засоби та алгоритми для чисельного обчислення власних чисел та функцій еліптичного диференціального оператора.

Мета роботи: підвищення точності власних чисел та функцій еліптичного оператора для тіл обертання знайдених за допомогою чисельних методів.

Методи дослідження. Для вирішення поставлених задач були використані методи скінчених елементів і Гальоркіна, метод розділення змінних, методи обробки експериментальних даних.

Наукова новизна отриманих результатів визначається тим, що була побудована математична моделі знаходження власних чисел та функцій еліптичного диференціального оператора для тіл обертання, на основі якої була розроблена комп'ютерна модель. На основі отриманих під час моделювання результатів було удосконалено процедуру пошуку власних чисел та функцій для тіл обертання, що дозволило збільшити точність розрахунків.

Практична цінність результатів полягає в тому, що розроблене програмне забезпечення полегшує процес вирішення задач пошуку власних чисел та функцій, а отримані за його допомогою дані дозволяють обрати необхідну кількість елементів триангуляції, що використовуватимуться для знаходження рішення із заданою точністю.

Область застосування. Розроблене програмне забезпечення може використовуватися науковцями та інженерами при моделюванні чи розрахунках реальних фізичних процесів.

Значення роботи та висновки. Удосконалена методика дозволяє проектувати тіла обертання зі значним скороченням матеріальних та часових витрат та підвищити їх надійність та ефективність.

Прогнози щодо розвитку досліджень. Розробити WEB-додаток для розрахунку власних чисел та функцій еліптичного диференціального оператора для тіл обертання.

У розділі «Економіка» були проведені розрахунки трудомісткості розробки програмного забезпечення, витрат на створення ПЗ і тривалості його розробки та маркетингові дослідження ринку збуту створеного програмного продукту.

Список ключових слів: метод скінчених елементів, власне значення, власна функція, математичне моделювання, диференціальні рівняння у частинних похідних.

ABSTRACT

Explanatory note: 78 pages, 49 figures, 4 tables, 3 applications, 27 sources.

Object of research: the process of numerical calculation of eigenvalues and functions.

Subject of research: software tools and algorithms for numerical calculation of eigenvalues and functions of an elliptical differential operator.

Purpose of Master's thesis: improving the accuracy of eigenvalues and functions of the elliptical operator for bodies of rotation found using numerical methods.

Research methods. Finite element and Galerkin methods, variable separation method, experimental data processing methods were used to solve the set problems.

Originality of research is determined by the fact that a mathematical model of finding the eigenvalues and functions of an elliptical differential operator for bodies of rotation was built, on the basis of which a computer model was developed. Based on the results obtained during the simulation, the procedure for finding eigenvalues and functions for rotating bodies was improved, which allowed to increase the accuracy of calculations.

Practical value of results is that the developed software simplifies the process of solving problems of finding eigenvalues and functions, and the data obtained with it help to select the required number of triangulation elements that will be used to find a solution with a given accuracy.

Scope of application. Developed software can be used by scientists and engineers in modeling or calculating real physical processes.

The value of the work and conclusions. The improved technique allows to design bodies of revolution with a significant reduction in material and time costs and increase their reliability and efficiency.

Research development forecasts. Develop a WEB-application for calculating eigenvalues and eigenfunctions of an elliptical differential operator for bodies of rotation.

In the Economics section, calculations of the labor intensity of software development, the cost of creating software and the duration of its development, and marketing research of the sales market for the created software product were made.

Keywords: finite element method, eigenvalue, eigenfunction, mathematical modeling, partial differential equation.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ВВ – власний вектор;

ВЗ – власне значення;

ВЧ – власне число;

ВФ – власна функція;

МРЗ – метод розділення змінних;

МСЕ – метод скінчених елементів;

ПЗ – програмне забезпечення;

СС – ступені свободи;

PDE Tool - Partial Differential Equation Toolbox.

ЗМІСТ

| | |
|--|----|
| ВСТУП..... | 8 |
| РОЗДІЛ 1. АНАЛІЗ ТЕМИ ТА ПОСТАНОВКА ЗАДАЧІ | 10 |
| 1.1 Загальні відомості з предметної області..... | 10 |
| 1.2 Постановки фізичних задач для рівнянь еліптичного типу..... | 13 |
| 1.3 Диференціальні оператори у частинних похідних | 14 |
| 1.4 Порівняння методів розв’язання ДРЧП другого порядку..... | 17 |
| 1.5 Аналіз існуючих математичних пакетів | 19 |
| 1.6 Висновки | 23 |
| РОЗДІЛ 2. МЕТОДИ РІШЕННЯ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ ЧАСТИННИХ ПОХІДНИХ ДРУГОГО ПОРЯДКУ | 25 |
| 2.1 Класифікація методів вирішення ДРЧП | 25 |
| 2.2 Метод відокремлення змінних..... | 26 |
| 2.3 Метод скінчених елементів..... | 31 |
| 2.4 Програмна реалізація метода скінчених елементів | 39 |
| 2.5 Висновки | 47 |
| РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ЗНАХОДЖЕННЯ ВЛАСНИХ ЧИСЕЛ ТА ФУНКЦІЙ ЕЛІПТИЧНОГО ДИФЕРЕЦІАЛЬНОГО ОПЕРАТОРА | 48 |
| 3.1 Опис розробленої системи | 48 |
| 3.2 Аналіз ефективності програмного забезпечення та отриманих результатів..... | 54 |
| 3.3 Висновки | 62 |
| РОЗДІЛ 4. ЕКОНОМІКА..... | 63 |
| 4.1 Визначення трудомісткості розробки програмного забезпечення | 63 |
| 4.2 Витрати на створення програмного забезпечення..... | 66 |
| 4.3 Маркетингові дослідження ринку збуту розробленого програмного продукту | 68 |
| 4.4 Оцінка економічної ефективності впровадження програмного забезпечення | 69 |
| ВИСНОВКИ..... | 70 |
| ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ..... | 71 |
| Додаток А. КОД ПРОГРАМИ..... | 74 |
| Додаток Б. ВІДГУК КЕРІВНИКА ЕКОНОМІЧНОГО РОЗДІЛУ | 77 |
| Додаток В. ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ | 78 |

ВСТУП

Значна кількість прикладних задач, пов'язаних із дослідженням фізичних явищ або штучних систем описуються за допомогою функцій двох або більше змінних. В свою чергу ці функції являють собою розв'язки диференціальних рівнянь частинних похідних.

При моделюванні природних явищ, що пов'язані з динамікою рідин та газів, теплообміном різних тіл, електродинамікою та іншими, використовуються диференціальні рівняння з частинними похідними (ДРЧП) першого та другого порядку або системи рівнянь.

Процес вирішення прикладних задач інженера нерідко зводиться до розв'язання ДРЧП. Неможливість вирішення деяких ДРЧП аналітичним шляхом формує потребу у використанні чисельних методів таких як інтегральний метод, метод скінчених елементів (МСЕ) тощо. З появою комп'ютерів з високою обчислювальною потужністю з'явився попит на програмне забезпечення (ПЗ), що може прорахувати складні структури і розв'язати сотні ДРЧП за короткий проміжок часу. З розвитком математичної фізики великий попит на ПЗ для вирішення задач на ВЗ та ВФ зберігається актуальним і на сьогоднішній день.

Метою кваліфікаційної роботи є розробка ПЗ призначеного для вирішення ДРЧП другого порядку еліптичного диференціального оператора для тіл обертання та порівняння результатів роботи програми з рішеннями отриманими за допомогою аналітичних методів, в залежності від щільності розбиття області та визначити ефективність впровадження ПЗ.

Кваліфікаційна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел і чотирьох додатків. У першому розділі проведений аналіз предметної області та здійснена постановка задачі. У другому розділі розглядаються аналітичні та чисельні методи знаходження ВЧ та ВФ. У третьому розділі наведений опис розробленої системи, проведений аналіз ефективності ПЗ, та отриманих результатів. У четвертому розділі було визначено трудомісткість

розробки ПЗ, розраховано витрати на створення ПЗ, та проведені маркетингові дослідження ринку збуту розробленого програмного продукту.

РОЗДІЛ 1

АНАЛІЗ ТЕМИ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Загальні відомості з предметної області

Завдання знаходження власних значень (ВЗ) та власних векторів (ВВ) матриць є одним із основних завдань для багатьох розділів фізики. З обчислювальною проблемою такого роду можна стикнутись, наприклад, при дослідженні власних коливань різних механічних систем, коливальних та електронних спектрів молекул і кристалів. Цілком принципіальне значення ця проблема набула після створення у тридцятих роках минулого століття квантової механіки, яка стала базовою дисципліною дослідження мікросвіту.

ВЗ та ВВ можуть здаватися надуманими абстракціями для людей, що поверхнево знайомі з ними з курсу лінійної алгебри та в більшості випадків справа вивчення ВЗ та ВВ зводиться до думки про те, що насправді представляють власні пари.

ВВ матриці A – це ненульовий вектор, який при множенні на деяку квадратну матрицю A , перетворюється на самого себе з числовим коефіцієнтом λ , та для якого виконується співвідношення:

$$Av = \lambda v,$$

де

λ – це певний скаляр, дійсне або комплексне число;

v – ненульовий вектор.

Число λ називають ВЗ або власним числом (ВЧ) матриці A [1].

ВВ – це осі або напрямки, в яких діє матричне перетворення, а відповідні ВЗ дають нам міру того, наскільки “значущим” є цей ефект. ВВ і ВЗ не залежать від вибору базису, бо вони означені без застосування координат, через це подібні матриці мають однакові ВЗ.

Оскільки кожній квадратній матриці відповідає певне лінійне перетворення (у деякому базисі), то, виходячи з змістовного сенсу, часто говорять про ВЗ та ВВ лінійного перетворення.

Розглянемо геометричний приклад, що ілюструє поняття, що розглядаються – на репродукції Джоконди (рис. 1.1). Синій вектор не змінюється в результаті перекошу площини, а значить, є ВВ даного лінійного перетворення з коефіцієнтом $\lambda = 1$. Будь-який паралельний (колінеарний) синьому вектор – теж буде ВВ даного лінійного перетворення.

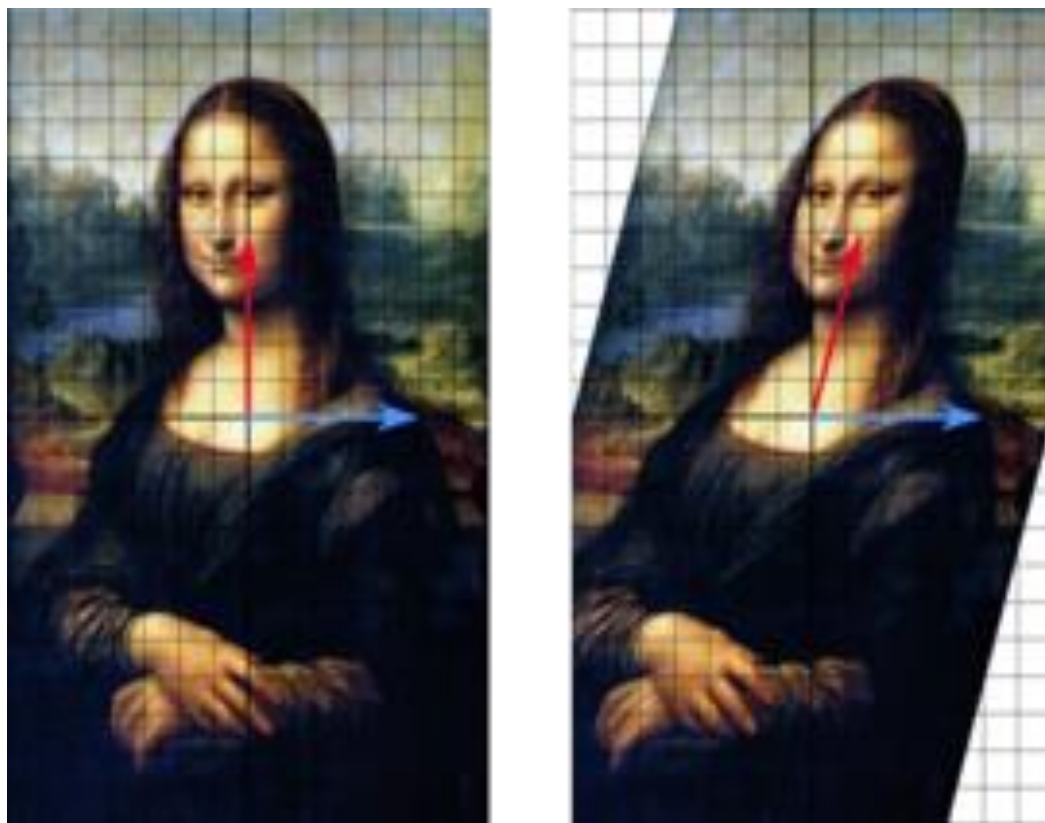


Рис.1.1. Геометричний приклад лінійного перетворення

Пам'ятаючи, що матриці є лінійними операторами, будь-які ідеї, наприклад проектування інженерних споруд та обчислення коливань, що на них впливають можна застосувати до інших лінійних операторів, таких як диференціальні оператори. Скажімо, нас цікавить стаціонарне рівняння

Шредінгера (1.1) в деякій області, зокрема розв'язки цього рівняння які є хвильовими функціями.

$$H\Psi = E\Psi, \quad (1.1)$$

де

H – оператор Гамільтона;

Ψ – хвильова функція;

E – ВЗ енергії.

Ця функція надає нам важливу інформацію щодо різних ймовірностей щодо електронів та інших елементарних частинок. Дане рівняння Шредінгера ставиться як задача на ВЗ, тому знаходження Ψ дорівнює знаходженню ВФ оператора Гамільтона [2].

Завдання на ВЗ дуже часто може не мати нетривіальних рішень або їх нескінченно багато. Нижче наведено приклад крайової задачі на ВЗ (1.2)

$$y'' + p(x)y' + \lambda q(x)y = 0; y(a) = 0, y(b) = 0, \quad (1.2)$$

де

$$p(x) \equiv 0, q(x) \equiv 1, a = 0 \text{ і } b = \pi;$$

число λ — параметр задачі.

Якщо взяти $\lambda = 3$, то буде отримано рівняння (1.3)

$$y'' + 3y = 0; y(0) = 0, y(\pi) = 0. \quad (1.3)$$

При $\lambda = 4$ буде отримано рівняння (1.4)

$$y'' + 4y = 0; y(0) = 0, y(\pi) = 0. \quad (1.4)$$

Приклади 1.3 та 1.4 показують, що ситуація у крайовому завданні з параметром може сильно залежати від конкретного чисельного значення параметра.

У задачі на ВЗ ставиться питання:

при яких (дійсних) значеннях параметра існує нетривіальне (тобто ненульове) вирішення крайового завдання [3]? Таке значення λ називається власним, або характеристичним числом, або значенням даної задачі. Тому, можна сказати, що у прикладі (1.4) ми переконалися, що $\lambda = 4$ є ВЧ крайової задачі

$$y'' + \lambda y = 0; y(0) = 0, y(\pi) = 0. \quad (1.5)$$

Припустимо, що λ_* — ВЧ задачі (1.2), а y_* — нетривіальне рішення задачі, що виходить після підстановки цього ВЧ, тобто задачі (1.6)

$$y_*'' + p(x)y_*' + \lambda_* q(x)y_* = 0 \text{ и } y_*(a) = 0, y_*(b) = 0. \quad (1.6)$$

Тоді y_* називається ВФ, що відповідає ВЧ λ_* .

1.2 Постановки фізичних задач для рівнянь еліптичного типу

Одним із основних розділів вищої математики є рівняння математичної фізики, що присвячені вивченню диференціальних рівнянь частинних похідних. Найбільш розповсюдженими є диференціальні рівняння 2-го порядку. Інженерам часто доводиться розв'язувати задачі, в яких шукана величина залежить від декількох змінних. В такому випадку такі рівняння, що містять частинні похідні називають диференціальними рівняннями з частинними похідними (ДРЧП). На практиці дані рівняння виступають у вигляді математичних моделей існуючих фізичних явищ та процесів.

Загалом, диференціальні рівняння з частинними похідними мають нескінченну кількість розв'язань, що залежать від довільних функцій. Під час вивчення реальних фізичних процесів, таких як теплопровідність, дифузія, рівновага та інші потрібно знати повністю визначений розв'язок рівняння, а це

можливо за наявності додаткових умов [4]. Такі умови можуть бути визначені з постановки обраної фізичної задачі.

Існує два основних типи умов:

- початкові умови, що відповідають стану досліджуваного фізичного процесу в початковий момент часу;
- граничні (крайові) умови, визначені на границях досліджуваної області.

Підсумовуючи вищенаведене, щоб скласти постановку задачі математичної фізики необхідно:

- 1) обрати функцію, що характеризує фізичне явище чи процес;
- 2) вивести відповідне обраному процесу рівняння;
- 3) сформулювати початкові та встановити граничні умови.

1.3 Диференціальні оператори у частинних похідних

Диференціальні оператори є узагальнення операції диференціювання. Найпростіший диференціальний оператор D , діючи на функцію y , "повертає" першу похідну цієї функції (1.7):

$$Dy(x) = y'(x). \quad (1.7)$$

Дворазове застосування операції D дозволяє отримати другу похідну функції $y(x)$ (1.8)

$$D^2y(x) = D(Dy(x)) = Dy'(x) = y''(x). \quad (1.8)$$

Аналогічно, n -а ступінь оператора D призводить до n -ої похідної (1.9)

$$D^n y(x) = y^{(n)}(x). \quad (1.9)$$

Тут ми припускаємо, що функція $u(x)$ є n разів диференційованою та визначеною на множині дійсних чисел. Сама функція $u(x)$ може набувати комплексних значень. Диференціальні оператори можуть мати і більш складний вид - в залежності від утворюючих їх диференціальних виразів.

Диференціальний оператор називається еліптичним, гіперболічним або параболічним, якщо він визначається диференціальним виразом відповідного типу [5]. Диференціальні рівняння з частинними похідними класифікують в залежності від їх математичної природи, такі як гіперболічні, еліптичні, параболічні або від розв'язуваних фізичних задач, наприклад хвильові рівняння, рівняння дифузії, рівняння теплопровідності (табл. 1.1).

Таблиця 1.1

| Тип рівняння | Математична форма | Практична задача |
|--------------------------------|--|---------------------------------|
| Лапласа (еліптичне) | $\Delta U = 0$ | Стаціонарні теплові поля |
| Пуассона (еліптичне) | $\Delta U = -k$ | Теплопередача від джерела тепла |
| Хвильове (гіперболічне) | $\Delta U = \frac{1}{a^2} \frac{\partial u}{\partial t}$ | Розповсюдження звукових хвиль |
| Теплопровідності (параболічне) | $\Delta U = \frac{1}{c^2} \frac{\partial^2 u}{\partial t^2}$ | Нестаціонарна теплопровідність |

До рівнянь еліптичного типу приводить вивчення стаціонарних процесів різної фізичної природи. Саме під час вивчення та вирішення еліптичних рівнянь використовують еліптичні оператори. У загальному вигляді еліптичне рівняння може бути записано як $Lu = f$. Рівняння Пуассона є одним з найпростіших рівнянь еліптичного типу. У випадку коли $f(x) = 0$, рівняння Пуассона називається рівнянням Лапласа [6], або потенційним рівнянням (1.10).

$$\Delta U(x_1, x_2, \dots, x_n) = 0. \quad (1.10)$$

Рівняння в частинних похідних другого порядку гіперболічного типу найчастіше зустрічаються у задачах пов'язаних з процесами коливань. Найпростіше рівняння гіперболічного типу виглядає так (1.11)

$$u_{xx} - u_{yy} = 0 \quad (1.11)$$

та зазвичай називається рівнянням коливань струни.

Рівняння частинних похідних другого порядку параболічного типу найчастіше зустрічається при вивченні процесів теплопровідності та дифузії. Прикладом рівняння параболічного типу можна називати рівнянням виду (1.12)

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + \frac{\partial^2 U}{\partial z^2} + f(t, x, y, z) \quad (1.12)$$

та описати ним такі процеси:

1) поширення тепла в однорідному ізотропному тілі. Тоді $U(t, x, y, z)$ є температурою точок тіла в різні моменти часу (1.13)

$$a^2 = k(cp)^{-1}, \quad (1.13)$$

де $k > 0$ – коефіцієнт внутрішньої теплопровідності тіла;

c – його питома теплоємність;

ρ – густина;

$$f(t, x, y, z) = F\rho^{-1},$$

де $f(t, x, y, z)$ – інтенсивність внутрішніх джерел тепла.

2) дифузія рідини або газу в однорідному середовищі. Тоді $U(t, x, y, z)$ є концентрація речовини в точках середовища в різні моменти часу (1.14),

$$a^2 = dc^{-1}, \quad (1.14)$$

де d – коефіцієнт дифузії;

c – коефіцієнт пористості середовища;

$$f(t, x, y, z) = Fc^{-1},$$

де $F(t, x, y, z)$ – інтенсивність внутрішніх джерел тепла.

1.4 Порівняння методів розв'язання ДРЧП другого порядку

Завдання на ВЗ та ВФ точно вирішені лише для небагатьох випадків. Для їхнього наближеного вирішення застосовуються різні чисельні методи. Один із найпоширеніших наближених методів вирішення ДРЧП частинних похідних є МСЕ. Популярність метода зумовлена розвитком математичного апарату побудови та аналізу МСЕ, а також простотою алгоритмізації метода [7].

МСЕ був розроблений інженерами в 50-х роках двадцятого століття. Основним призначенням метода було вирішення задач аеродинаміки. Однак, у 70-х роках метод отримав популярність в інженерних кругах серед математиків.

Основними етапами застосування МСЕ є:

1. Інтегральне (варіаційне) формулювання задачі математичної фізики;
2. Розбиття області на елементи, наприклад трикутники (триангуляція), чотирикутники;
3. Визначення залежності між напругою та деформацією у вузлах;
4. Виведення рівнянь для системи в цілому;
5. Розв'язання отриманої системи алгебраїчних рівнянь;
6. Обчислення значень інших вершин.

Метод кінцевих різниць (або метод сіток) – є одним із універсальних методів, що широко відомий та використовується для вирішення крайових задач [8]. Його популярність можна пояснити тим, що метод надає відносно простий підхід для дискретизації диференціальних рівнянь. Сутність метода полягає в наступному.

Область нескінченої зміни аргументу заміняють дискретною кількістю точок (вузлів), що називається сіткою. Замість функцій нескінченого аргументу розглядають функції, що визначені тільки у вузлах сітки – сіткові функції.

Порівняймо МСЕ з іншим популярним методом вирішення диференціальних рівнянь – методом кінцевих різниць (табл. 1.2).

Таблиця 1.2

| Метод скінчених елементів | Метод кінцевих різниць |
|---|---|
| Наближений метод вирішення задач, описуваних диференціальними рівняннями | Наближений метод вирішення диференціальних рівнянь |
| Включає вирішення великих «розріджених» систем рівнянь | Включає вирішення великих «розріджених» систем рівнянь |
| Постановка може бути складною, з численними вхідними і вихідними параметрами | Постановка задачі проста, містить мало вхідних і вихідних параметрів |
| Широкий вибір типів, форм та розмірів елементів | Зазвичай використовують правильні сітки |
| Дозволяє вирішувати задачі з неоднорідним й анізотропним розподіленням параметрів | Неприспосований для вирішення задач з неоднорідним та анізотропним розподіленням параметрів |
| Дозволяє створювати універсальні програми | Для кожної конкретної задачі створюється окрема програма |
| Представлення граничних умов - просте | Представлення граничних умов пов'язано з труднощами |

Виходячи з порівняльної таблиці двох методів, можна зробити висновок, що МСЕ є більш універсальним та підходить для написання програм, що вирішуватимуть різні задачі.

1.5 Аналіз існуючих математичних пакетів

Незважаючи на універсальність МСЕ, складність вирішення задач з високим ступенем точності в короткий термін залишається нагальною [9]. Вирішити цю проблему можливо використовуючи універсальні математичні пакети, за допомогою яких можна швидко розв'язати багато математичних задач або провести попередні розрахунки, що в подальшому будуть використовуватись у проєктованій комп'ютерній системі.

Найвідомішими математичними пакетами є: MatLab, Mathcad, Maple, Mathematica, R.

Пакет MatLab – це мова програмування високого рівня та інтерактивна серед розробки алгоритмів і програм, а також інструмент аналізу даних. Пакет орієнтований на вирішення інженерних задач, а також моделювання складних технічних систем. Ефективність MatLab зумовлена передусім орієнтацією на математичні обчислення з програмною емуляцією паралельних обчислень і спрощеними засобами задання циклів [10]. Окрім того, у MatLab є велика кількість пакетів розширення, що багатократно збільшують ефективність використання системи. Одним із таких пакетів, призначених для вирішення ДРЧП та їх систем є PDE TOOLBOX (рис. 1.2). Також, пакет включає в себе бібліотеку Maple, для роботи з аналітичними перетвореннями.

Окремо треба зазначити, що пакет MatLab, має як англійську так російськомовну документацію, що постійно оновлюється. PDE TOOLBOX, також має свою документацію, крім того на сайті розробника MathWorks можна задавати питання іншим користувачам пакета або звернутись до Help Center з питанням пов'язаним з роботою в MatLab.

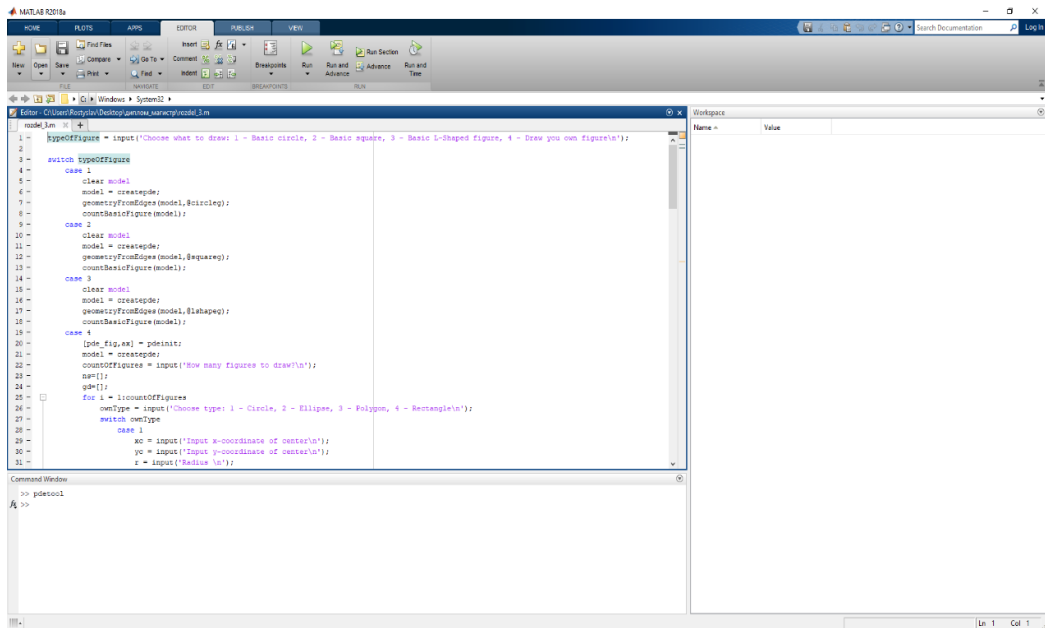


Рис. 1.2. Інтерфейс Matlab

Пакет Mathematica – популярна система комп'ютерної алгебри, є лідером серед програмних продуктів. Серед можливостей пакету можна виділити проведення символічних (аналітичних) перетворень, що допомагає ефективно розв'язувати різні математичні задачі (рис. 1.3). В пакеті доступна мова програмування високого рівня, однак, більшість задач розв'язуються з використанням інтерфейсу [11].

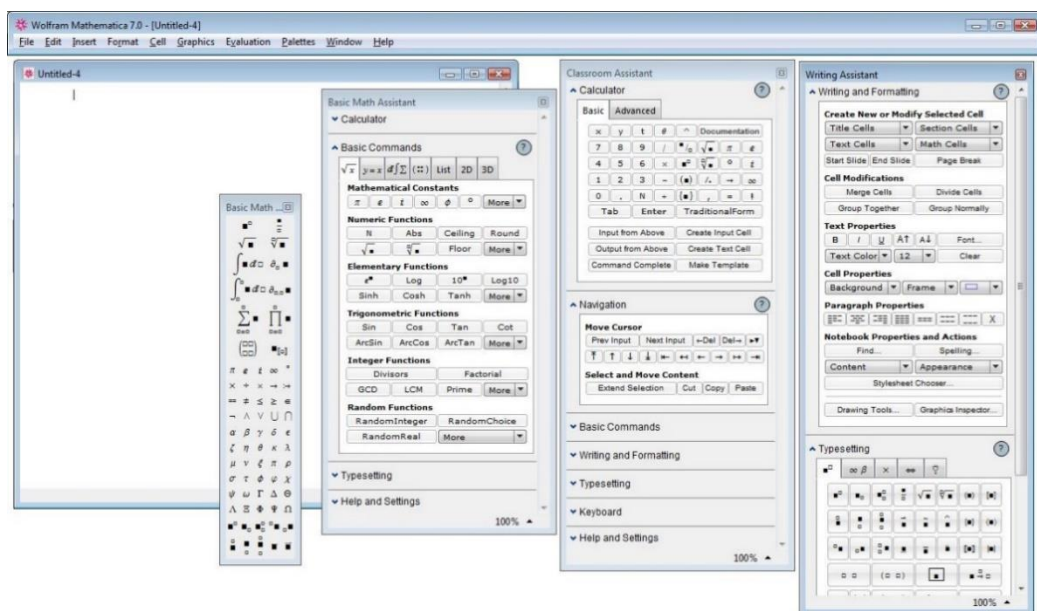


Рис. 1.3. Інтерфейс Mathematica

Maple – це пакет для аналітичних обчислень на комп'ютері, що містить більше двох тисяч команд, що дозволяють вирішувати задачі алгебри, геометрії, математичного аналізу, диференціальних рівнянь, статистики та математичної фізики, знаходити інтеграли (рис. 1.4). Окрім аналітичних обчислень, задачі є можливість робити чисельні розрахунки. У Maple є символічні, чисельні та гібридні алгоритми, алгоритмічне ядро системи містить методи, недоступні іншим платформам [12]. Крім цього, система володіє функціоналом для створення 2D- та 3D-візуалізації та анімації, і також пропонує користувачам ефективні алгоритми для високопродуктивних обчислень.

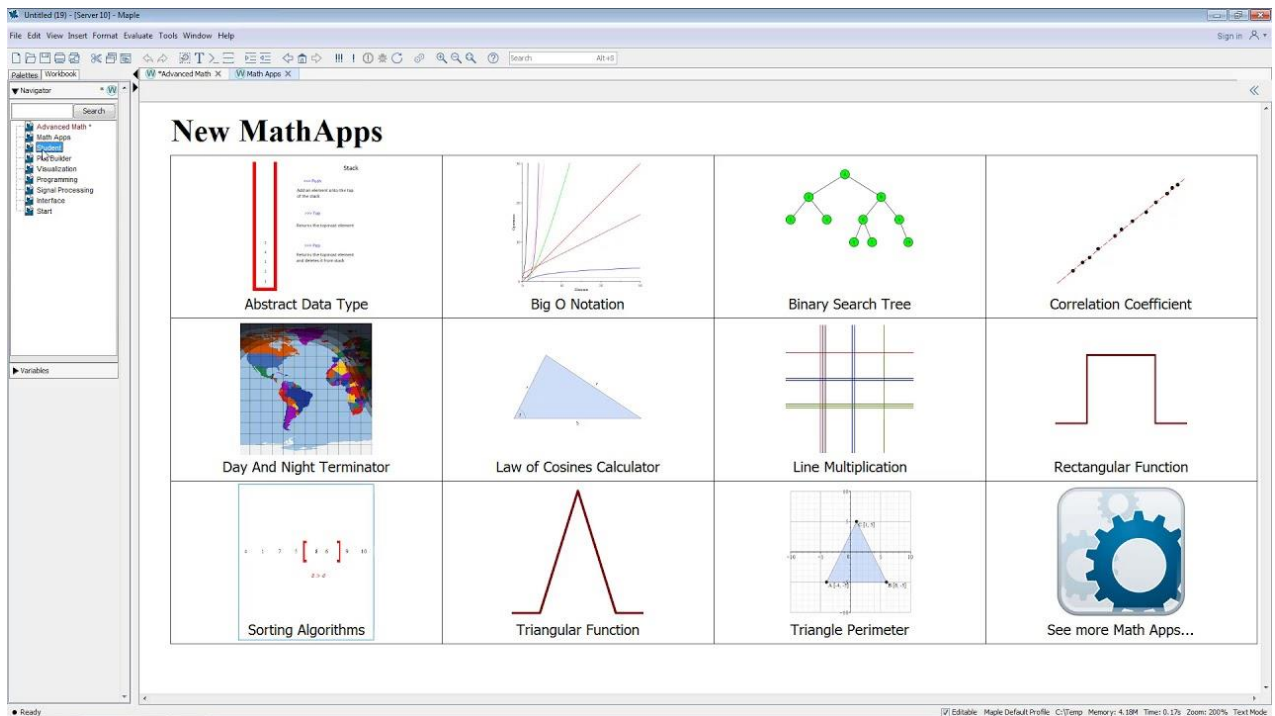


Рис. 1.4. Інтерфейс Maple

Mathcad – є математичним редактором (рис. 1.5), що дозволяє проводити різноманітні наукові та інженерні розрахунки, починаючи від елементарної арифметики та закінчуючи складними реалізаціями чисельних методів. Хоча програма орієнтована на аудиторію користувачів, що не мають кваліфікацію програміста, завдяки простоті застосування, наочності математичних дій, великій бібліотеці вбудованих функцій та чисельних методів, можливості символічних

обчислень, а також чудовому апарату представлення результатів, Mathcad використовується в складних проектах [13]. Також пакет має вбудовану інтеграцію з MatLab та дозволяє зв'язувати створені ним документи із документами пакета MatLab.

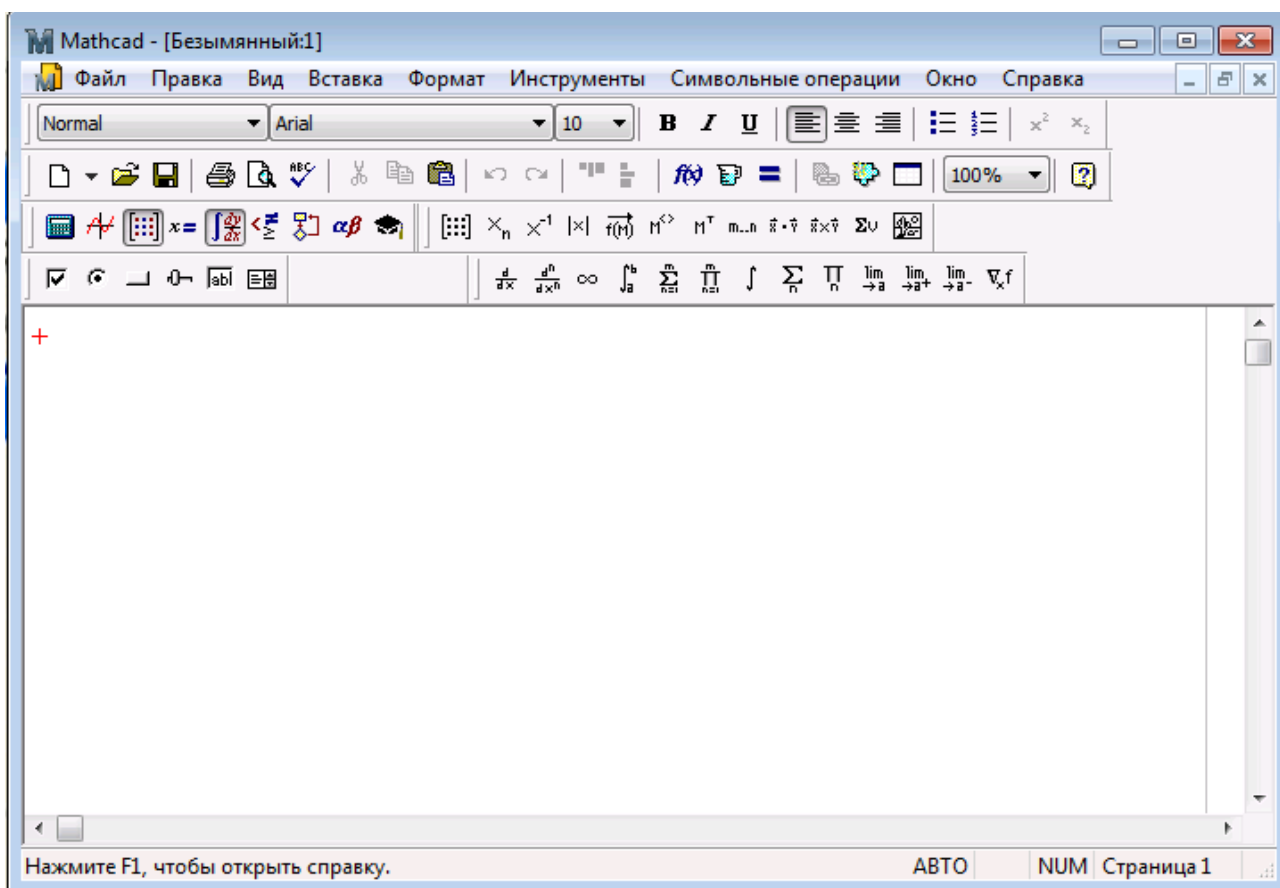


Рис. 1.5. Інтерфейс Mathcad

Пакет R – це сукупність R функцій та даних, а також мова програмування для аналітики. Пакет дозволяє розширити функціональність мови програмування R, пакети без яких важко було б уявити роботу з R, включені в базову збірку автоматично доступні після встановлення R на комп'ютер (ядро R). Перевагою R серед інших пакетів є велика кількість безкоштовних пакетів з відкритим вихідним кодом. R значною мірою використовується у проектах пов'язаних з статистичними дослідженнями. Для роботи з R, рекомендовано користуватись середою розробки RStudio (рис. 1.6).

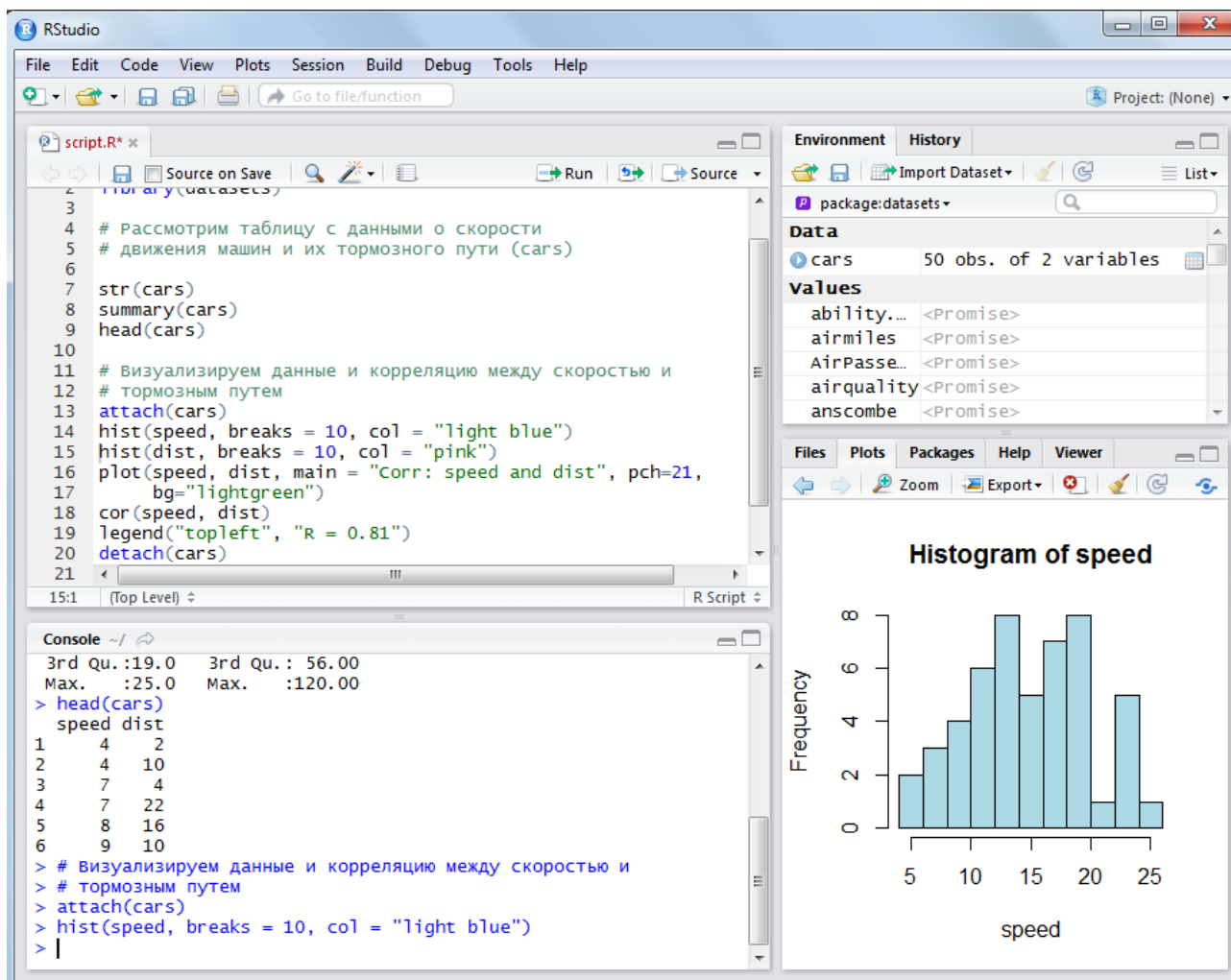


Рис. 1.6. Інтерфейс RStudio

Хоча усі перелічені математичні пакети мають свою сильні сторони, та можуть використовуватися для вирішення ДРЧП, для розробки ПЗ у рамках даної кваліфікаційної роботи було убрано MatLab, через його сумісність з іншими платформами, сучасну документацію та потужні вбудовані бібліотеки, такі як PDE Tool.

1.6 Висновок

Метою роботи є розробка ПЗ призначеного для вирішення ДРЧП другого порядку, які задовольняють крайовим умовам на кордоні області в двовимірних

областях, що дає змогу підвищити точність розрахунку ВЗ і ВФ еліптичного диференціального оператора для тіл обертання.

Для досягнення поставленої мети було поставлено наступні завдання:

- розробити ПЗ для автоматизації процесу вирішення ДРЧП другого порядку на ВЗ та ВФ;
- ПЗ має обробляти вхідні дані, що вводить користувач, такі як діапазон значень, коефіцієнти ДРЧП, область в якій шукається рішення, щільність дискредитації області, граничні умови;
- вся вхідна інформація має перевірятися на коректність та відповідність очікуваному типу;
- розв’язання задачі має припинятись, якщо знайдено велику кількість ВЗ у вказаному користувачем діапазоні;
- вихідною інформацією ПЗ є вектор знайдених ВЗ та матриця ВФ;
- отримане рішення має бути візуалізовано, з можливістю імпорту у файл;
- ПЗ має обробляти виняткові ситуації.

Отримані результати необхідно проаналізувати на предмет залежності відхилення отриманого рішення від аналітичного, в залежності від щільності розбиття області.

РОЗДІЛ 2

МЕТОДИ РІШЕННЯ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ ЧАСТИННИХ ПОХІДНИХ ДРУГОГО ПОРЯДКУ

2.1 Класифікація методів вирішення ДРЧП

Для розв'язування ДРЧП можливо використовувати аналітичні методи, що дають точні результати. Однак, розв'язки аналітичним методами ДРЧП існують лише для деяких, ідеалізованих, їх класів, тому на разі при вирішенні прикладних задач інженери віддають перевагу числовим методам, з використанням комп'ютерів.

Не дивлячись на те, що числові методи розв'язання ДРЧП дуже різноманітні, можна виділити риси, спільні ним [14 - 16] :

- неперервну область зміни аргументів шуканої функції замінюють різницевою сіткою (дискретна множина вузлів);
- на різницевій сітці розглядають функції дискретного аргументу – сіткові функції, що були визначені у вузлах сітки, або використовують їх локальну апроксимацію за допомогою функцій, визначених елементів, що обмежені вузлами;
- ДР для кожного моменту модельного часу замінюють різницевими рівняннями, початкові і граничні умови також замінюють різницевими початковими і крайовими умовами.

Методи розв'язання диференціальних рівнянь частинних похідних:

- 1) метод відокремлення змінних;
- 2) метод інтегральних перетворень;
- 3) метод перетворення координат;
- 4) метод інтегральних рівнянь;
- 5) чисельні методи;
- 6) варіаційні методи;
- 7) метод скінчених елементів.

Перші чотири методи є точними методами вирішення ДРЧП, інші – наближеними методами.

У результаті використання чисельного метода отримуємо різницеву крайову задачу у вигляді однієї або декількох систем алгебраїчних рівнянь. В залежності від обраного методу, результатом розв'язку різницевої задачі може бути кусково-неперервна функція (для МСЕ), або сіткова функція.

Обчислювальним методам ДРЧП відрізняються один від одного значенням таких притаманних їм характеристик як:

- при збільшенні щільності дискретизації сітки спостерігається збіжність отриманого наближеного розв'язку до аналітичного;
- точність апроксимації диференціальної задачі різницевою;
- стійкість розв'язку до випадкового відхилення вхідних даних;
- консервативність, тобто виконання законів збереження енергії, маси;
- монотонність, а саме відповідність наближеного розв'язку фізичному явищу, що моделюється, з точки зору кількості локальних екстремумів та його осциляцій;
- економічність, а саме лінійна залежність кількості операцій, необхідних для розв'язання алгебраїчних рівнянь, від кількості вузлів сітки.

2.2 Метод відокремлення змінних

Одним з найбільш поширених способів аналітичного вирішення рівнянь у приватних похідних є метод поділу змінних, званий також методом Фур'є.

Цей метод вперше запропонував Ж.Л. Д'аламбер в середині XVIII століття для вирішення хвильового рівняння з початковими і крайовими умовами, а на початку XIX століття був розвинений Ж.Б.Ж. Фур'є та обґрунтований М.В. Остроградським та П.Г. Діріхле.

Основна ідея цього методу полягає в тому, що розв'язання задачі для рівняння з частинними похідними зводиться до вирішення допоміжних задач для

рівнянь із меншою кількістю незалежних змінних. Зокрема, якщо задане рівняння містить дві незалежні змінні, то допоміжні завдання вже залежатимуть лише від однієї змінної. Таким чином, рішення ДРЧП зводиться до вирішення звичайних диференціальних рівнянь (2.1).

Нижче представлено завдання на ВЗ та ВФ для оператора Лапласа [17] у прямокутній області (2.1) $G = \{(x, y): 0 < x < a, 0 < y < b\}$ (рис. 2.1):

$$\begin{cases} -\Delta u = \lambda u, (x, y) \in G, \\ u|_{\partial G} = 0, \\ u \neq 0, \quad \|u\| = 1. \end{cases} \quad (2.1)$$

де

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}, \quad (2.2)$$

$$u = u(x, y) \in U \subset L_2(G), \quad (2.3)$$

u - невідома функція,

λ - невідоме число,

∂G - границя прямокутника G , $\|u\| = \sqrt{\int_G |u|^2 dG}$.

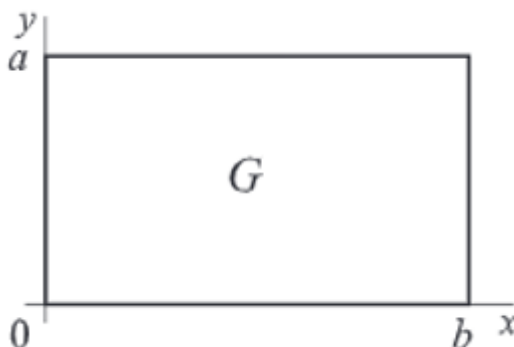


Рис. 2.1. Прямокутна область для рівняння (2.1)

Для розв'язання задачі (2.1) застосуємо метод відокремлення змінних із застосуванням рядів Фур'є. Основне його припущення у тому, що невідома функція $u(x, y)$ шукається як (2.4)

$$u(x, y) = X(x)Y(y). \quad (2.4)$$

Для застосування цього методу необхідно виконання наступних умов:

а) область G має бути представлена у вигляді прямого добутку одновимірних областей;

б) граничні умови повинні розділятися на граничні умови для кожної із змінних;

в) у рівнянні змінні мають розділятися.

У задачі, що розглядається, перша умова виконується: $G = (0, a) \otimes (0, b)$. Далі, необхідно підставити, наприклад, граничні умови на відрізьку $y = 0, x \in (0, a)$ у функцію (2.4):

$$u|_{\substack{y=0 \\ x \in (0, a)}} = X(x)Y(0) = 0. \quad (2.5)$$

Це можливо, якщо $Y(0) = 0$ або $X(x) \equiv 0$. В останньому випадку $u \equiv 0$ це відповідає тривіальному рішення, що неприпустимо. Із (2.5) отримано граничну умову для функції $Y(y) : Y(0) = 0$. Аналогічно, розглядаючи відрізьки $y = b, x = 0, x = a$, одержано $Y(b) = 0, X(0) = 0, X(a) = 0$. Підставивши в оператор Лапласа вираз (2.4) одержано рівняння (2.6)

$$\begin{aligned} & - \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) - X(x)Y(y) = \\ & = -X''(x)Y(y) - X(x)Y''(y) = \lambda X(x)Y(y). \end{aligned} \quad (2.6)$$

Розділивши рівняння (2.6) було одержано (2.7)

$$-\frac{X''(x)}{X(x)} = \lambda + \frac{Y''(y)}{Y(y)}. \quad (2.7)$$

Тотожність (2.7) має виконуватися при всіх значеннях незалежних змінних x та y . Це можливо тільки якщо обидві частини дорівнюють певній константі (2.8)

$$-\frac{X''(x)}{X(x)} = \mu = \lambda + \frac{Y''(y)}{Y(y)}. \quad (2.8)$$

Постійна μ називається константою поділу змінних. Отже, виконуються всі три умови а – в. У результаті одержано дві одновимірні задачі (2.9)

$$\begin{cases} -X''(x) = \mu X(x), \\ X(0) = 0, \\ X(a) = 0, \\ X \neq 0, \|X\| = 1; \end{cases} \quad \begin{cases} -Y''(y) = \nu Y(y), \\ Y(0) = 0, \\ Y(b) = 0, \\ Y \neq 0, \|Y\| = 1; \end{cases} \quad (2.9)$$

Тут $\nu = \lambda - \mu$. Умови нормування функцій $X(x)$ і $Y(y)$ достатні для отримання нормованої функції $u(x,y)$, оскільки

$$\begin{aligned} \|u\| &= \sqrt{\int_0^a \int_0^b |u(x,y)|^2 dx dy} = \\ &= \sqrt{\int_0^a |X(x)|^2 dx} \sqrt{\int_0^b |Y(y)|^2 dy} = \|X\| \|Y\| \end{aligned} \quad (2.10)$$

ВЗ та ВФ одномірних завдань (2.9) мають вигляд (2.11)

$$\begin{aligned} \mu_m &= \left(\frac{\pi m}{a}\right)^2, & X_m(x) &= \sqrt{\frac{2}{a}} \sin \frac{\pi m x}{a}, m = 1, 2, \dots; \\ \nu_n &= \left(\frac{\pi n}{b}\right)^2, & Y_n(y) &= \sqrt{\frac{2}{b}} \sin \frac{\pi n y}{b}, n = 1, 2, \dots; \end{aligned} \quad (2.11)$$

Це дозволяє записати рішення початкової задачі у вигляді (2.12)

$$\begin{aligned} \lambda_{mn} &= \mu_m + \nu_n = \left(\frac{\pi m}{a}\right)^2 + \left(\frac{\pi n}{b}\right)^2, \quad m, n = 1, 2, \dots; \\ u_{mn}(x, y) &= X_m(x)Y_n(y) = \left(\frac{\pi m}{a}\right)^2 + \left(\frac{\pi n}{b}\right)^2 = \\ &= \frac{2}{\sqrt{ab}} \sin \frac{\pi m x}{a} \sin \frac{\pi n y}{b}. \end{aligned} \quad (2.12)$$

Необхідно зазначити, що ВЗ для різних m і n можуть збігатися і кількість ВФ для даного ВЗ, або кратність виродження, може бути більшою за одиницю. Наприклад, у разі квадрата ($a=b$) кратність виродження для λ_{11} дорівнює 1, для $\lambda = \lambda_{12} = \lambda_{21}$, для $\lambda = \lambda_{47} = \lambda_{74} = \lambda_{18} = \lambda_{81}$.

Залишається відкритим питання, чи всі рішення задачі (2.1), при виконанні умов а – в можуть бути знайдені методом Фур'є, тобто в припущенні (2.4). У теорії доводиться, що безліч рішень $u_{mn}(x, y)$ (2.12) утворює базис $L_2(G)$ і, таким чином, знайдено всі можливі рішення.

Але МРЗ не без недоліків. Коли граничні умови в одному напрямку залежать від іншого — подібно до границі в напрямку x залежить від y — тоді той самий набір функцій $f_j(x)$ не може бути розширений для кожного y . Візьмемо для прикладу округлену квадратну область (b), показану на рис. 2.2, МРЗ для вирішення задачі з такою областю не підійде.

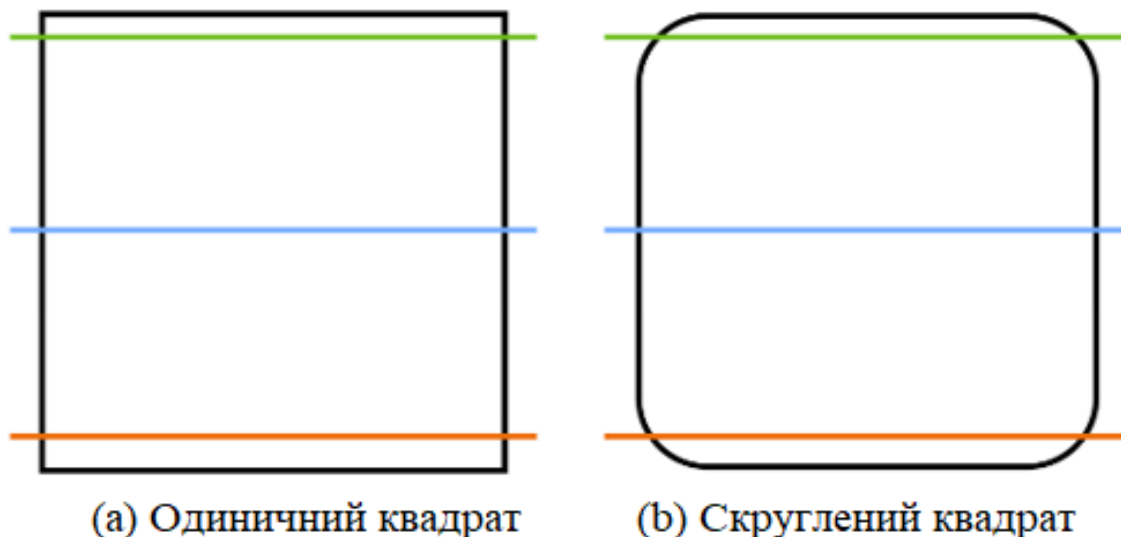


Рис. 2.2. Дві області для (a) МРЗ працюватиме, для (b) не працюватиме

Три кольорові лінії розміщені на трьох різних значеннях y . На фігурі одиничному квадраті (рис. 2.2) можна побачити, що ліворуч і праворуч межі залишаються незмінними для кожної з трьох ліній, тоді як на скругленому квадраті (рис. 2.2) ті самі межі змінюються.

Якщо задача розглядається для довільної області, то стає набагато менш імовірно, що знайти сумісну систему координат буде можливо, щоб відобразити симетрії області та використати МРЗ.

Отже, якщо (2.1) не можна розв'язати в округлій області (рис. 2.2), за допомогою МРЗ, то можливо використати інший чисельний метод, наприклад МСЕ.

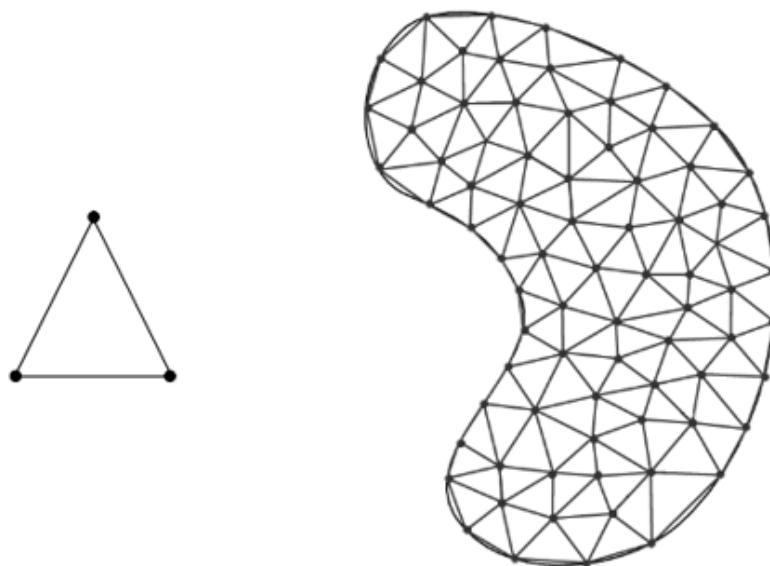
2.3 Метод скінчених елементів

Основна ідея МСЕ, що базується на методах Бубнова, Галеркіна і Рітца, була запропонована Р. Курантом 1943 р., але залишилася непоміченою, випередивши потреби практики. У 50-х роках минулого століття з появою перших комп'ютерів виникла потреба в розробці нових інженерних підходів до

чисельного вирішення задач зі складною геометрією, в яких області інтегрування розбивалися на підобласті. Такі підобласті і отримали назву кінцевих елементів.

Переважає більшість фізичних явищ у нашому світі виражається мовою математики через ДРЧП [18]. Як можна побачити з прикладу вище, за винятком дуже ідеалізованих ситуацій, задачі не можуть бути вирішені аналітичним шляхом. Основна концепція МСЕ полягає в тому, щоб розбити нетипову просторову область на кілька простих елементів, зазвичай трикутників або прямокутників. Ці елементи можуть бути зібрані різними способами і в результаті можуть представляти дуже складні форми. Елементи з'єднуються разом у вузлах, конкретних точках елемента, у яких значення польової змінної — залежної змінної у диференціальному рівнянні — буде явно обчислено. За допомогою вузлових значень можна апроксимувати невузлові значення шляхом інтерполяції.

Процес моделювання або апроксимації вашої вихідної задачі таким чином називається дискретизація. Разом елементи та вузли відомі як сітка. Дискретизація має вирішальне значення, оскільки вимоги задачі тепер мають бути задоволені лише на кінцевій кількості дискретних елементів, на відміну від безперервної кількості вздовж усієї області.



(c) Трикутний елемент із виділеними вузлами

(d) Сітка вузлів у довільній області

Рис. 2.3. Дискретизація області на сітку

Щоб розглянути принцип роботи МСЕ необхідно представити задачу, наприклад оператора Лапласа на ВЗ (2.1), у наступному вигляді (2.13)

$$\begin{cases} -\Delta u = \lambda u, \\ u = 0. \end{cases} \quad (2.13)$$

на довільній області Ω .

Очевидно, метою цієї (2.13) задачі є знаходження власних пар $-(u, \lambda)$. Для цього ми повинні зіткнутися із завданням дискретизації нашого безперервного ДРЧП на області Ω на скінченну кількість дискретних елементів.

Першим етапом метода є переформування граничної задачі, що називається слабкою (варіаційною) формою. Те, що записано в (2.13), відоме як сильна форма, оскільки воно вимагає, щоб твердження виконувалося для всіх точок Ω . Найкращим варіантом було би, щоб слабка форма послабила цю вимогу таким чином, щоб нова задача все ще була еквівалентною, але тепер її можна легко наблизити за допомогою чисельних методів.

Загалом, простір Гільберта ($V := H(\Omega)$) – це узагальнення поняття евклідового простору на нескінченновимірний функціональний простір [19].

Одна з властивостей є внутрішній добуток, визначений як (2.14)

$$\langle v, f \rangle = \int_{\Omega} v f dA. \quad (2.14)$$

Для тестової функції $\varphi \in V$ із (2.14) можна отримати:

$$\begin{aligned} \int_{\Omega} -\Delta u \varphi dA &= \int_{\Omega} \lambda u \varphi A, \\ \int_{\Omega} \nabla \cdot (-\nabla u) \varphi dA &= \int_{\Omega} \lambda u \varphi dA. \end{aligned} \quad (2.15)$$

Проінтегрувавши по частинам (2.15), одержано (2.16)

$$\int_{\Omega} \nabla u \cdot \nabla \varphi dA + \int_{\partial\Omega} (-\nabla u) \cdot \vec{n} \varphi dS = \int_{\Omega} \lambda u \varphi A, \quad (2.16)$$

враховуючи граничні умови (2.16) приймає вигляд:

$$\int_{\Omega} \nabla u \cdot \nabla \varphi dA = \int_{\Omega} \lambda u \varphi dA. \quad (2.17)$$

Рівняння (2.17) є інтегральною формою ДРЧП, яка відома як слабка форма [20]. Форму називають слабкою, оскільки вона послаблює необхідну гладкість шуканого рішення порівняно з класичним рішенням. Тепер задачу було дещо змінено на знаходження $u \in V$ так, що для (2.11) виконується $\forall \varphi \in V$.

Наразі лишається проблема, задача все ще сформульована у нескінченновимірному сенсі. Щоб використовувати потужність сучасних комп'ютерів, необхідно зробити задані функціональні простори скінченновимірними. Для цього потрібно спочатку провести просторову дискретизацію фізичної області Ω в сітку. Сітка — це сукупність елементів, показаних на рис. 2.3. Суть ідеї полягає в тому, щоб створити схему, яка буде апроксимувати рішення для багатьох дрібних частин окремо, а потім згодом інтерполювати частини назад в одне безперервне рішення [21]. Для цього необхідно визначити скінченний елемент як K , P_K , Σ , де K — деякий елемент, що належить сітці, P — скінченновимірний функціональний простір до K , а Σ — набір ступенів свободи (СС). Хоча ступені свободи відіграють вирішальну роль у проблемі інтерполяції та в основі P_K , математика є досить громіздкою, і ту саму інформацію можна краще передати візуально.

Необхідно переформулювати проблему таким чином, щоб було можливо наблизити «значення» елементу з його вузлів. Для цього потрібно визначити базисну функцію (функцію змішування, у числовому аналізі та теорії наближення). Функції змішування – це заздалегідь визначені функції, які будуть використовуватися для опису варіації залежної змінної в межах кінцевого елемента.

У роботі розглядаються лише кусково-лінійні функції, які характеризують будь-яку зміну залежної змінної як лінійну всередині елемента. Існує багато інших варіантів функцій форми, які описують нелінійні варіації, тому цей вільний вибір робить МСЕ дуже потужним методом, застосовним до широкого кола проблем, а не тільки для пошуку власних пар. Оскільки всі три вузлові значення повинні бути відомі для формування апроксимації (наближення), кажуть, що ці елементи мають три ступені свободи – самі вузлові точки. Функції тріангуляції (трикутна функція) – це широко використовуване сімейство функцій форми, що становлять область визначення. Для трикутного

елемента з трьома вузлами залежна змінна може бути апроксимована у вигляді
(2.18)

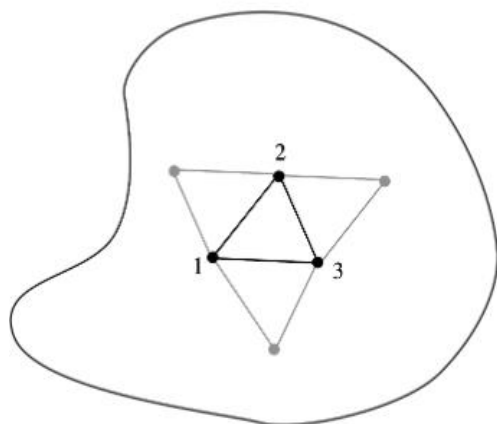
$$\alpha_1\psi_1 + \alpha_2\psi_2 + \alpha_3\psi_3, \quad (2.18)$$

де

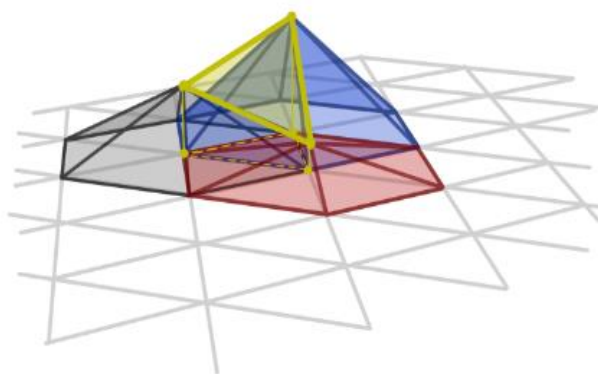
α_j — вузлові значення;

ψ_j — функції форми.

На рис. 2.4 можна побачити, як лінійна комбінація базисних функцій формує частину наближення.



(e) Ступені свободи для трикутного елемента



(f) Базисні функції апроксимують залежну змінну

Рис. 2.4. Наближення значення елемента з його вузлів

P_K - скінченновимірний функціональний простір деякого підпростору V . P_K складається з наших функцій триангуляції і має базис $\{\psi_1, \psi_2, \dots, \psi_j\}$. Для кожного елемента вибирається функція триангуляції з P_K , яка має властивість, що функція i -тої форми має значення 1 на i -тому ступені свободи і дорівнює нулю для всіх інших ступенів свободи (СС). Тобто,

$$P_K \subset V := \{p \in C(\Omega) \mid p|_{K_i} \in P_K, \text{ для кожного } K_i, p|_{\partial\Omega} = 0\} \quad (2.19)$$

$$\psi_j(CC_i) \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \quad (2.20)$$

З визначення P_K відомо, що будь-який $p \in P_K$ можна однозначно виразити за допомогою (2.21)

$$p = \sum_i p_j \psi_j, \quad (2.21)$$

де

p_j - коефіцієнти.

Треба зауважити, що ця форма ідентична апроксимації залежної змінної для трикутного елемента з трьома вузлами, який розглядався раніше, лише записаний більш формально [22 - 24].

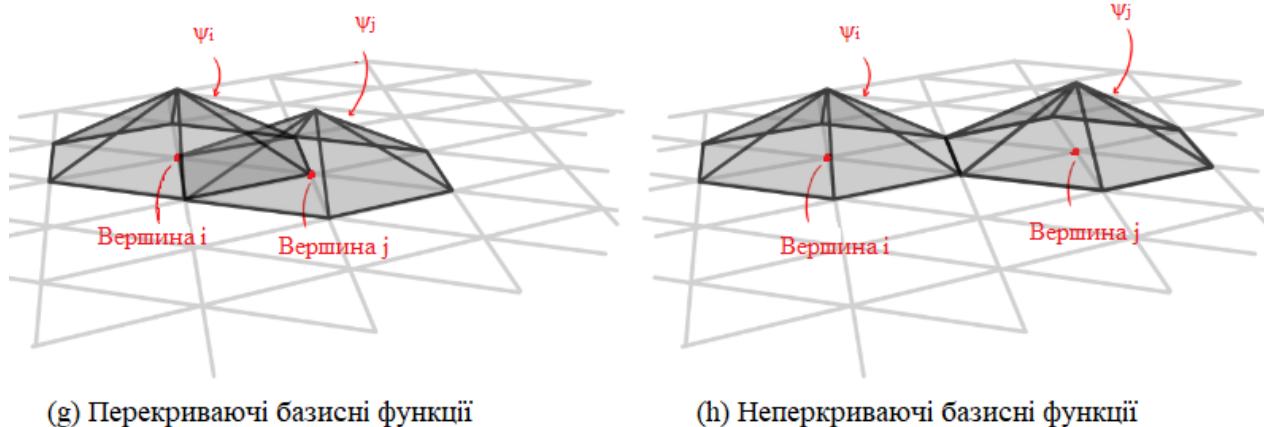


Рис. 2.5. Базисні функції (функції форми)

Ступінь перекриття визначає значення внутрішнього добутку між двома функціями форми. Результуюча проблема ВЗ стає дуже розрідженою, оскільки існує набагато більше комбінацій функцій форми (рис. 2.5), які не перекриваються, ніж тих, які перекриваються.

Таким чином, розв'язок задачі зводиться до знаходження рішення $u_p \in P_K$, по всім елементам u_p (2.22):

$$u_p = \sum_i u_j \psi_j. \quad (2.22)$$

Завдяки цьому опису u_p стає можливим виразити слабку форму задачі у скінченновимірній формі. Тестові функції φ як базисні функції P_K вибираються у вигляді:

$$\int_{\Omega} \nabla u \cdot \nabla \varphi dA = \int_{\Omega} \lambda u \varphi dA, \quad (2.23)$$

$$\int_{\Omega} \nabla \left(\sum_i u_j \psi_j \right) \cdot \nabla \varphi dA = \int_{\Omega} \lambda \left(\sum_i u_j \psi_j \right) \varphi dA, \quad (2.24)$$

$$\int_{\Omega} \nabla \left(\sum_i u_j \psi_j \right) \cdot \nabla \psi_i dA = \int_{\Omega} \lambda \left(\sum_i u_j \psi_j \right) \psi_i dA, \quad (2.25)$$

$$\sum_i u_j \int_{\Omega} \nabla \psi_j \cdot \nabla \psi_i dA = \lambda \sum_i u_j \int_{\Omega} \psi_j \psi_i dA, \quad (2.26)$$

$$\sum_i u_j \langle \nabla \psi_j, \nabla \psi_i \rangle = \lambda \sum_i u_j \langle \psi_j, \psi_i \rangle. \quad (2.27)$$

Тепер задача зводиться до знаходження $u_p \in P_K$ такого, що для (2.27) виконується $\forall \psi_i \in P_K$. Слід зауважити (2.27) утворює лінійну систему рівнянь з тим же розміром, що і P_K , це рівняння можна записати в більш звичний спосіб

$$A \vec{u}_p = \lambda M \vec{u}_p, \quad (2.28)$$

де

$$\vec{u}_p = \{u_1, u_2, \dots, u_i\};$$

$$A = \{a_{ij}\}_{i,j=1}^n, \quad a_{ij} = \langle \nabla \psi_j, \nabla \psi_i \rangle;$$

$$M = \{m_{ij}\}_{i,j=1}^n, \quad m_{ij} = \langle \nabla \psi_j, \nabla \psi_i \rangle.$$

A – називають матрицею жорсткості, а M — матрицею мас. Тут важливо зазначити, що ця форма проблеми все ще знаходиться в локальному масштабі окремого елемента. Матриці жорсткості та маси існуватимуть для кожного елемента в сітці, і багато сусідніх елементів матимуть якусь взаємодію чи вплив на сусідів, що потрібно відобразити в усій апроксимації задачі. Це робиться шляхом побудови глобальних матриць жорсткості та маси завдяки математичним схемам. Вони відрізняються в кожному конкретному випадку, залежно від обраної дискретизації. Вважається, що форма, наведена в (2.28), є глобальною формою.

Отже, після дискретизації була отримана задача на ВЗ та ВФ матриці. Таку форму задачі називають узагальненою матричною задачею на ВЗ і ВФ. Існує багато числових схем, для розв’язання цієї задачі, оскільки розріджені матричні задачі містять деякі унікальні застереження — наприклад, алгоритм Арнольда. Використовуючи цю ітерацію, стає можливо обчислити власні пари (λ, \vec{u}_p) чисельно. Знайшовши \vec{u}_p , можна знайти значення функцій в усіх вузлових точках сітки. Тоді, інтерполюючи ці точки, стає можливо апроксимувати вихідне u та його графічне зображення.

МСЕ — це в основному систематичний спосіб перетворення нескінченновимірних функцій, та будь-яких пов’язаних з ними проблем, у скінченновимірні, а потім у звичайні вектори у векторному просторі, створюючи наближено еквівалентну розв’язувану проблему, яку можна розв’язати чисельними методами.

2.4 Програмна реалізація метода скінчених елементів

У першому розділі був проведений аналіз сучасних математичних пакетів, що можуть бути використані для вирішення ДРЧП. В результаті було

обрано математичний пакет MatLab, адже в цьому пакеті є велика кількість пакетів розширення, що багаторазово збільшують ефективність використання системи. Одним з таких пакетів є PDE Toolbox, призначений для вирішення диференціальних рівнянь у частинних похідних та їх систем. Наведемо алгоритм розв'язання задачі МСЕ використовуючи Matlab та PDE Toolbox.

Пакет складається з набору функцій, що автоматизують реалізацію МСЕ для вирішення ДРЧП другого порядку та їх систем різних типів, таких як : гіперболічні, еліптичні, параболічні. Також, у пакеті доступний графічний інтерфейс, що дозволяє користувачу простіше користуватись усіма функціями PDE Toolbox та MatLab [25]. Для прикладу (2.29) буде розглянуто вирішення еліптичного ДЧРП використовуючи функції PDE Toolbox, в основі яких лежатиме МСЕ

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -20 \cdot \sin \pi x \cdot \sin 2\pi y, \quad (2.29)$$

де

Ω – це прямокутник, розмірами 2 x 1 з центром на початку координат, з нульовими граничними умовами $u|_{\Gamma} = 0$.

Розв'язання задачі складається з наступних етапів:

1. У командному вікні MatLab необхідно набрати pdetool (рис. 2.6)

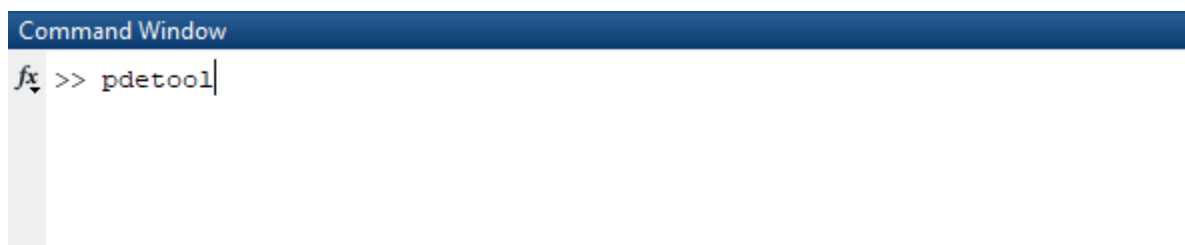


Рис. 2.6. Команда запуску інтерфейсу

Відкривається графічний інтерфейс (рис. 2.7) PDE Tool.

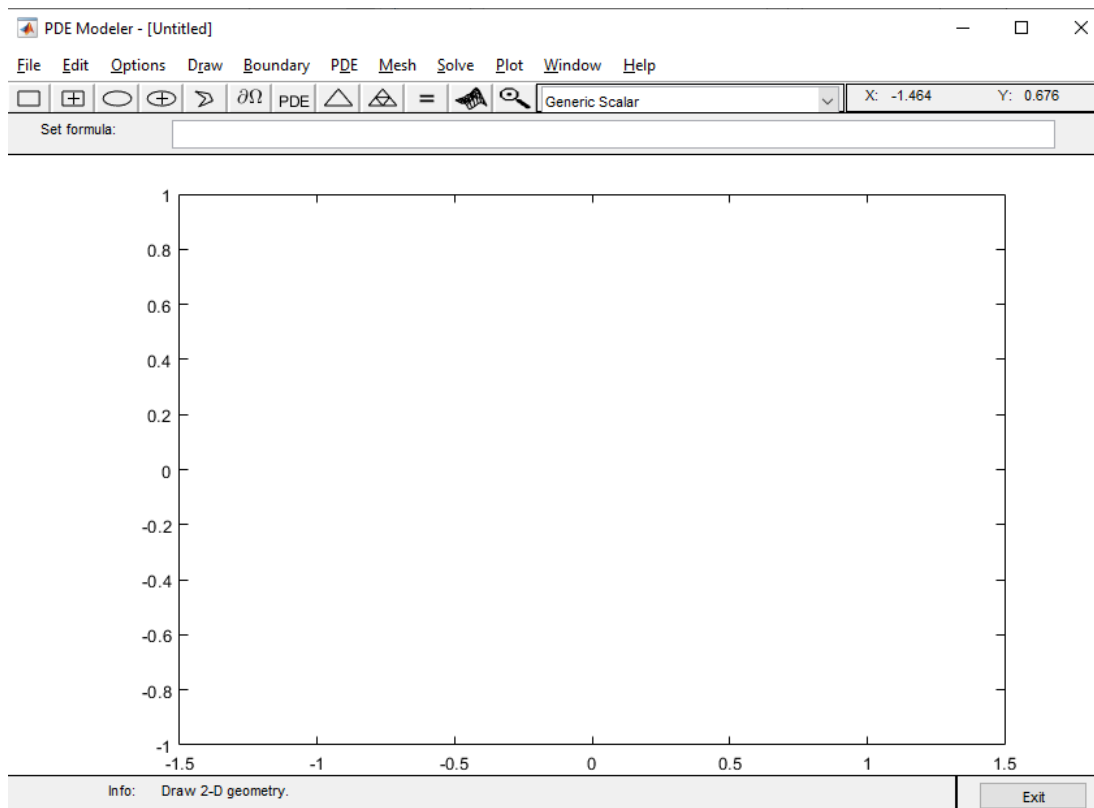


Рис. 2.7. Графічний інтерфейс Matlab PDE Tool

2. Задати двовимірну область Ω . Це можливо зробити у графічному інтерфейсі, натиснувши кнопку для створення прямокутника (рис. 2.8), що дає можливість відредагувати параметри створеного прямокутника (рис. 2.9).

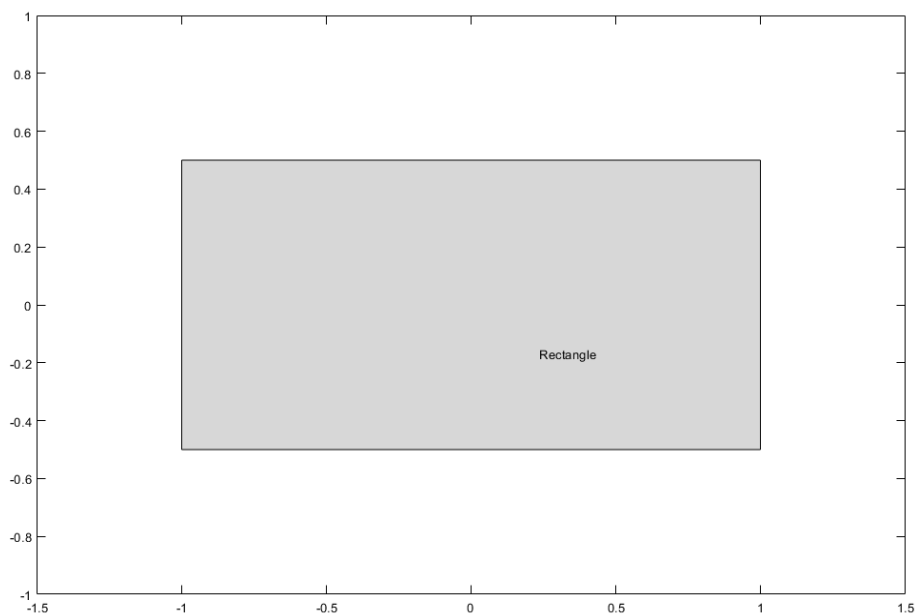


Рис. 2.8. Намальований довільний прямокутник

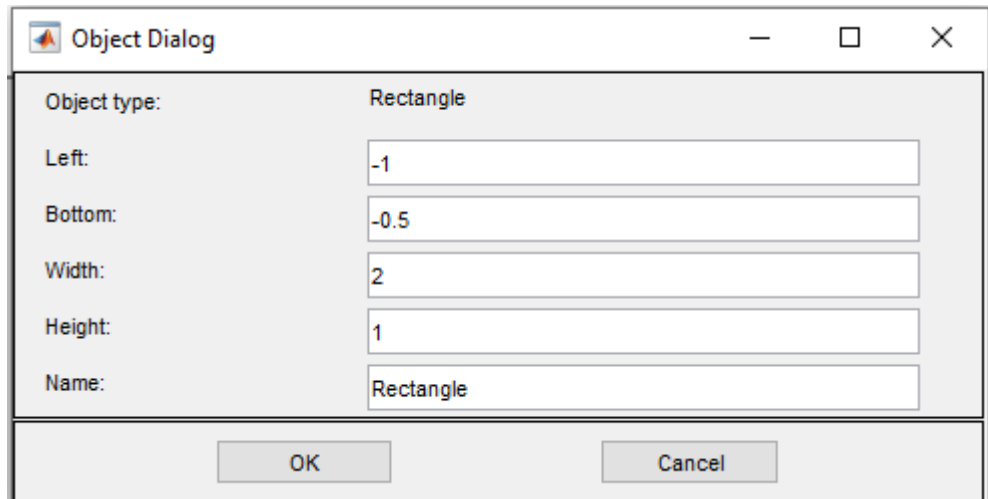


Рис. 2.9. Редагування параметрів прямокутника

3. Натиснувши кнопку Boundary та пункт Boundary mode (рис. 2.10), можна задати граничні умови, границі прямокутника виділяються червоним кольором.

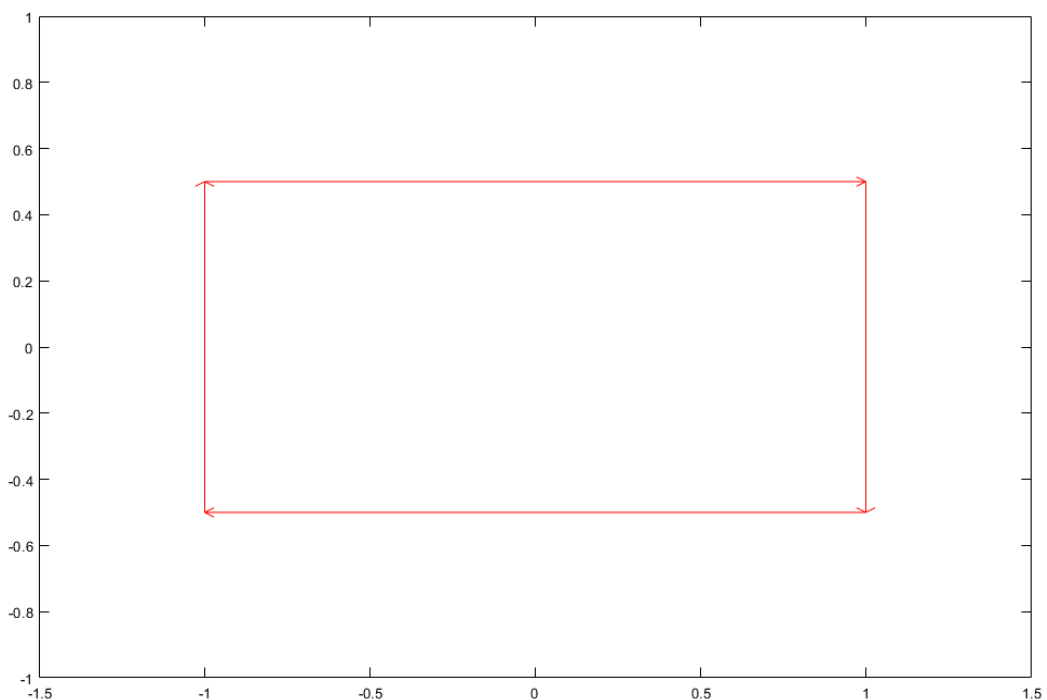


Рис. 2.10. Boundary Mode

Для уточнення граничних умов, необхідно у меню Boundary обрати пункт Specify Boundary Conditions (рис. 2.11).

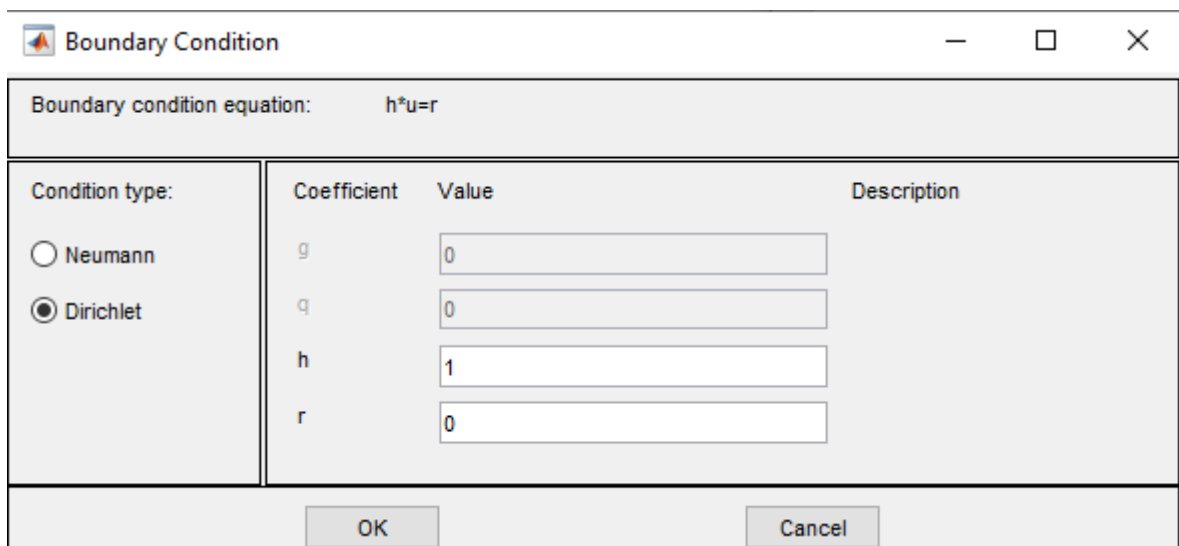


Рис. 2.11. Вікно налаштування граничних умов

У вікні (рис. 2.11) перемикач *Dirichlet* вказує на задання граничних умов Діріхле. Для даної задачі параметр $r = 1$.

4. Наступним етапом вирішення задачі буде генерація кінцево-елементної сітки. Натиснувши кнопку *Generate mesh* сітка генерується і відображається на прямокутнику (рис. 2.12), це виконуються завдяки функції автоматичної триангуляції області.

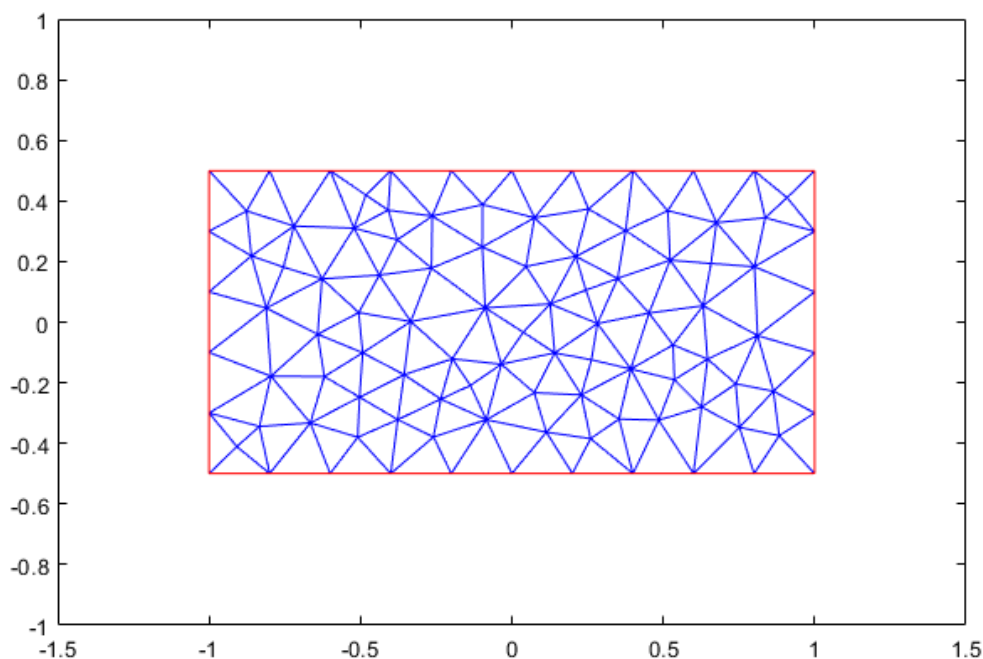


Рис. 2.12. Згенерована сітка у прямокутнику

Для підвищення точності обчислень, використовується опція Refine Mesh. Меню Mesh дозволить вказати додаткові параметри для зміни сітки. Після натискання сітка автоматично збільшить щільність розбиття (рис. 2.13).

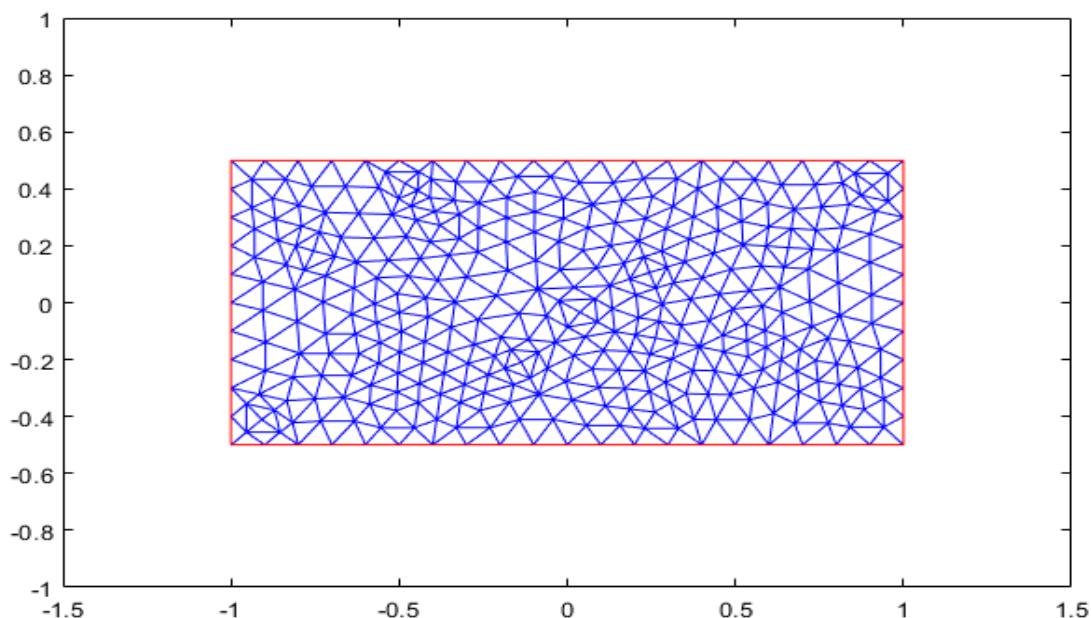


Рис. 2.13. Щільна сітка після Refine Mesh

5. Далі необхідно задати рівняння (2.29). Щоб це зробити, необхідно у меню PDE обрати пункт PDE Specification. З'являється вікно вибору виду ДРЧП. У строку f треба ввести значення правої частини рівняння (рис. 2.14)

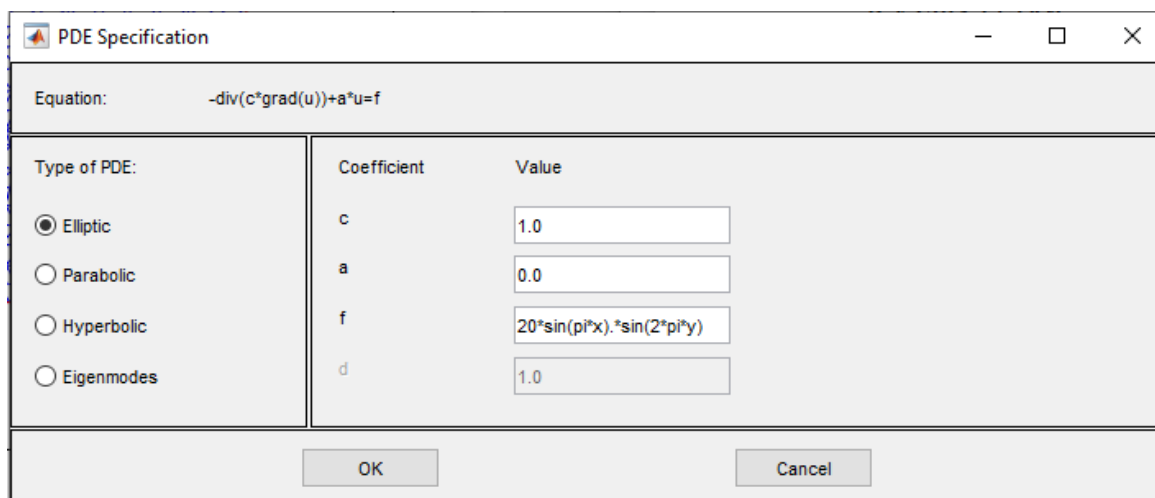


Рис. 2.14. Вікно налаштування рівняння

6. Задавши усі умови задачі необхідно натиснути кнопку Solve, тоді рівняння буде вирішено. В отриманому вікні отримано кольоровий двовимірний графік (рис. 2.15) наближеного рішення, на якому колір точок відповідає значенню рішення $u(x, y)$. У меню налаштування графіків (рис. 2.16), можна обрати пункт 3-D plot та побудувати тривимірний графік (рис. 2.17)

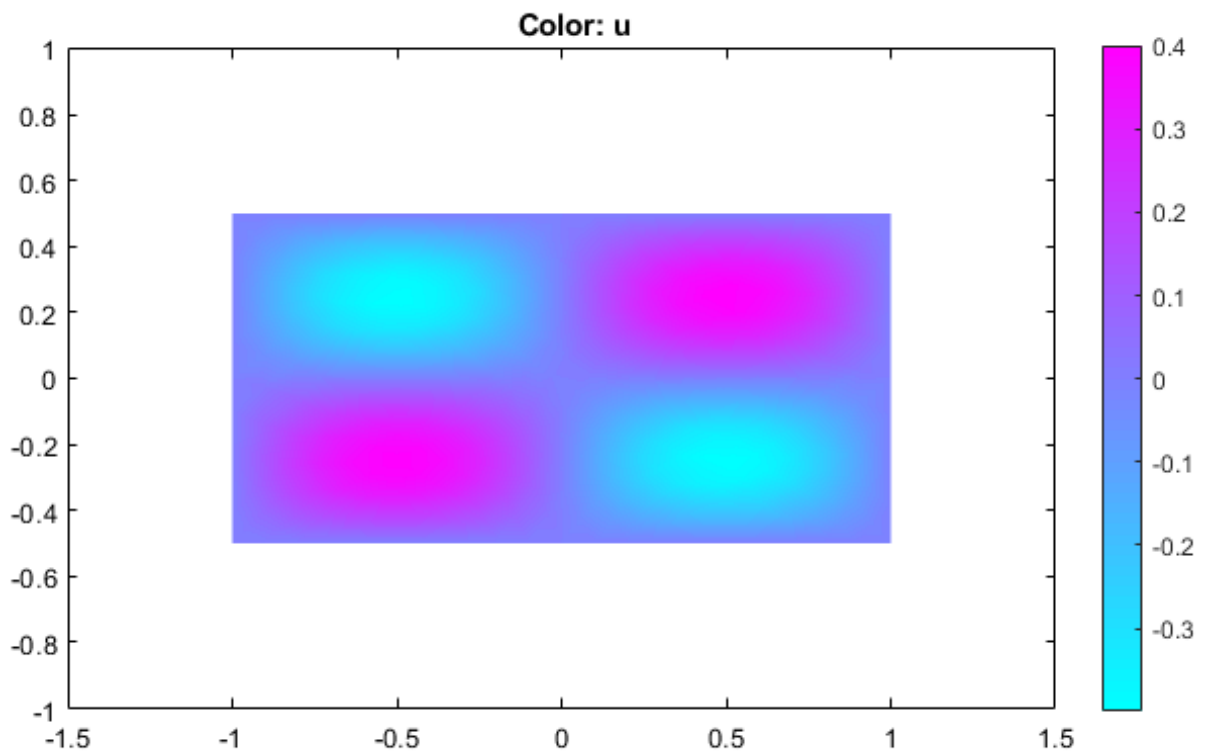


Рис. 2.15. Двовимірний графік отриманого рішення

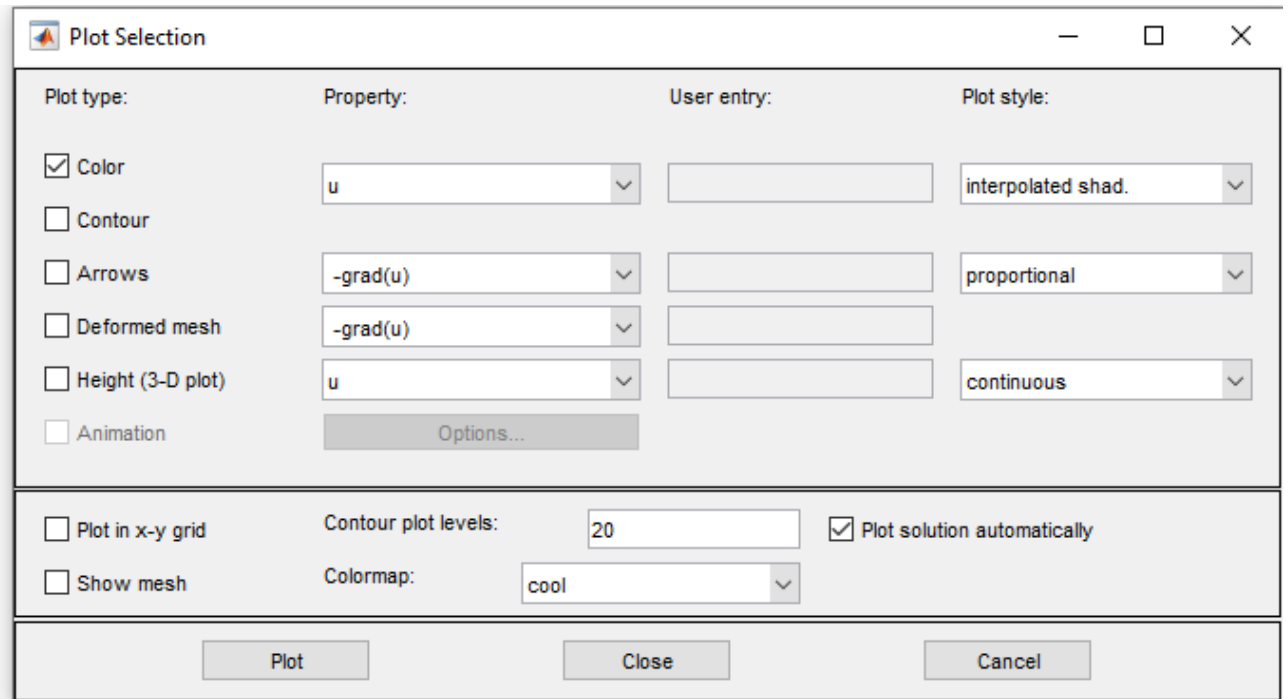


Рис. 2.16. Вікно налаштування інструменту побудови графіків

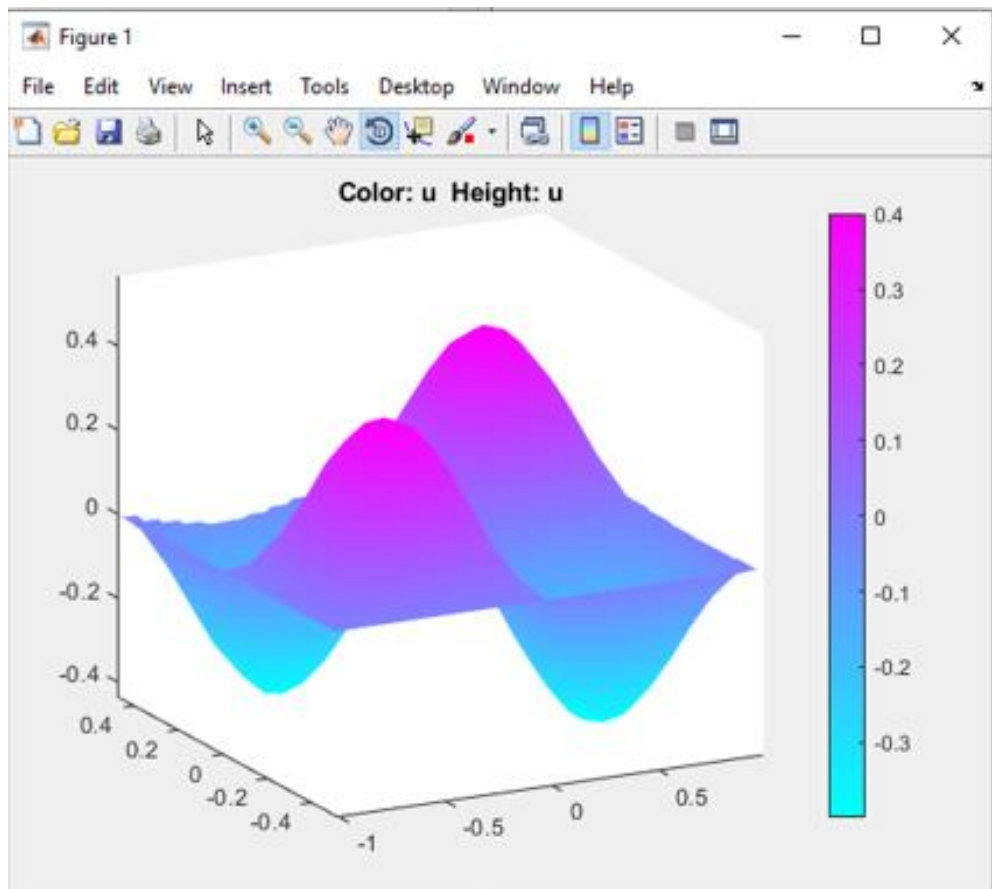


Рис. 2.17. Тривимірний графік отриманого рішення

2.5 Висновок

МСЕ дозволяє побудувати зручну схему формування системи алгебраїчних рівнянь щодо вузлових значень бажаної функції. Наближена апроксимація рішення за допомогою простих поліноміальних функцій та всі необхідні операції виконуються на кожному типовому елементі. В результаті елементи об'єднуються, що до бажаної системи алгебраїчних рівнянь. Такий алгоритм переходу від окремого елемента до їх сукупності, особливо зручний для геометрично та фізично складних систем. Вирішення подібних задач дуже зручно оформляти у формі програм, отримуючи необхідну інформацію від користувача через консоль.

РОЗДІЛ 3

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ЗНАХОДЖЕННЯ ВЛАСНИХ ЧИСЕЛ ТА ФУНКЦІЙ ЕЛІПТИЧНОГО ДИФЕРЕЦІАЛЬНОГО ОПЕРАТОРА

3.1 Опис розробленої системи

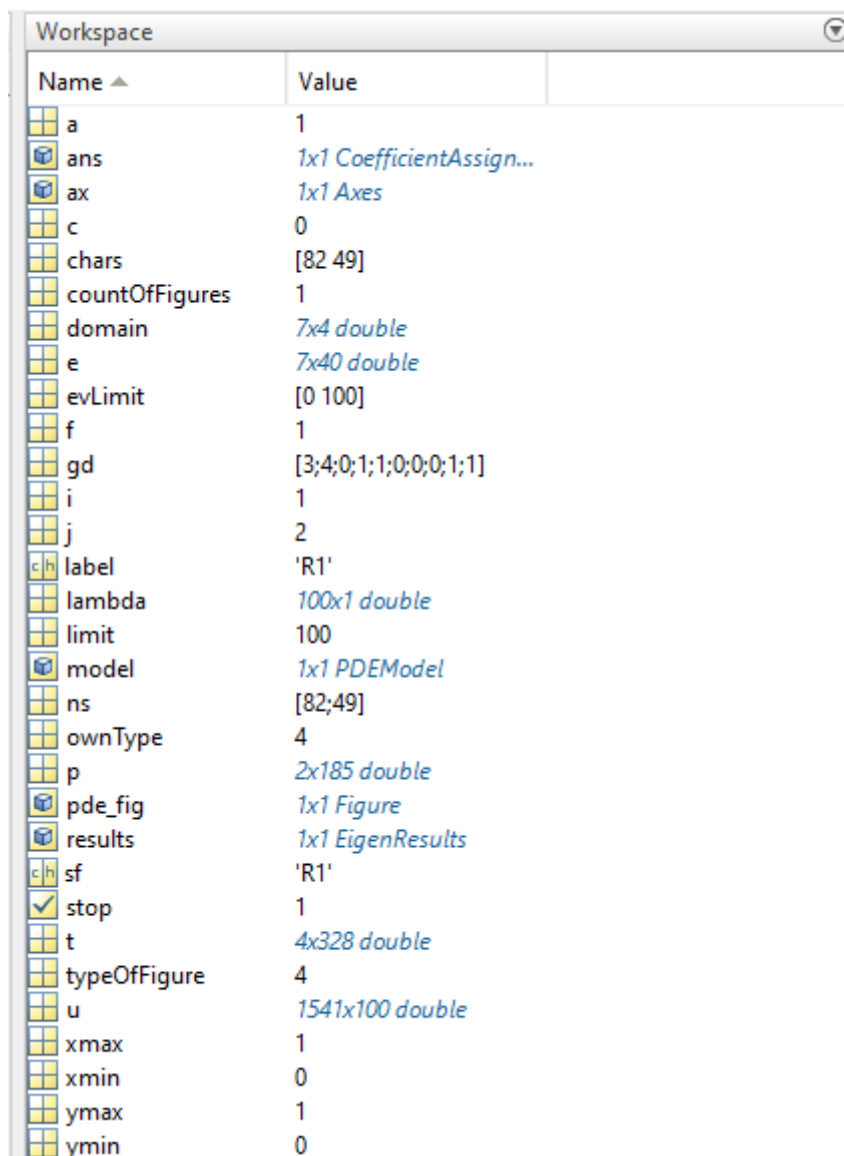
Оскільки Matlab є комплексним програмним середовищем, усі функції, що були представлені у другому розділі, доступні для використання за допомогою інтерпретованої мови програмування, що має ту саму назву – Matlab.

У даному розділі розглянуто основні методи, що використовуються у розробленій програмі, а також зроблено порівняння прикладу з другого розділу з результатами роботи програми.

Як видно з розглянутого прикладу вирішення ДРЧП у другому розділі, процес налагодження складається з великої кількості етапів. Щоб зменшити час на виконання однотипних обчислень зручно використовувати файли-сценарії [26]. Програми на мові MatLab мають два різновиди – script-файли (файли-сценарії, або керуючі програми) та файли-функції (процедури). Обидва різновиди повинні мати розширення імені файлу .m. За допомогою сценаріїв оформляють основні програми, керуючі від початку до кінця організацією всього обчислювального процесу, а також окремі частини основних програм.

Файл-функції служать для оформлення окремих процедур та функцій (тобто частин програми, розрахованих на неодноразове використання script-файлами при змінених значеннях вихідних параметрів і не можуть бути виконані без попереднього завдання значень вхідних змінних).

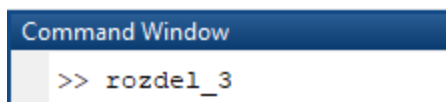
Сценарії оперують даними з робочої області (рис. 3.1) і можуть генерувати нові дані для подальшої обробки цього ж файлі. Дані, які використовуються у сценарії, зберігаються в робочій області після завершення сценарію та можуть бути використані для подальших обчислень.



| Name | Value |
|----------------|--------------------------|
| a | 1 |
| ans | 1x1 CoefficientAssign... |
| ax | 1x1 Axes |
| c | 0 |
| chars | [82 49] |
| countOfFigures | 1 |
| domain | 7x4 double |
| e | 7x40 double |
| evLimit | [0 100] |
| f | 1 |
| gd | [3;4;0;1;1;0;0;0;1;1] |
| i | 1 |
| j | 2 |
| label | 'R1' |
| lambda | 100x1 double |
| limit | 100 |
| model | 1x1 PDEModel |
| ns | [82;49] |
| ownType | 4 |
| p | 2x185 double |
| pde_fig | 1x1 Figure |
| results | 1x1 EigenResults |
| sf | 'R1' |
| stop | 1 |
| t | 4x328 double |
| typeOfFigure | 4 |
| u | 1541x100 double |
| xmax | 1 |
| xmin | 0 |
| ymax | 1 |
| ymin | 0 |

Рис.3.1. Приклад заповненого Workspace

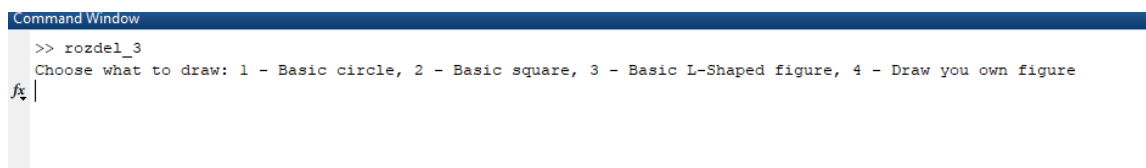
Нижче розглянуто приклад розв'язання еліптичного ДРЧП на ВЗ та ВФ за написаним сценарієм. Щоб запустити сценарій необхідно відкрити його файл у MatLab та натиснути кнопку меню Editor -> Run (), або прописати назву файлу у Command Window (рис. 3.2).



```
Command Window
>> rozdel_3
```

Рис. 3.2. Запуск програми за назвою в командному вікні

У консолі з'являється запит до користувача, необхідно обрати (рис. 3.3), яку фігуру намалювати, для користувача, що хоче ознайомитись з роботою програми, є на вибір три базові фігури. Четверта опція дає можливість налаштувати іншу фігуру, так само як це робиться у графічному інтерфейсі.



```
Command Window
>> rozdel_3
Choose what to draw: 1 - Basic circle, 2 - Basic square, 3 - Basic L-Shaped figure, 4 - Draw you own figure
fx |
```

Рис. 3.3. Запит до користувача обрати фігуру

Щоб намалювати коло, необхідно обрати пункт 1 – Basic circle (рис. 3.3). За побудову базової фігури відповідає функція `geometryFromEdges(model,@circleg)`, де

- `model` – спеціальний об'єкт моделі, зазначений як об'єкт `PDEModel`, цей об'єкт зберігає усі параметри, що використовується для роботи з функціями `PDETool`. Створити модель можливо командує `model = createpde()`;

- `@circleg` – шаблонна функція геометрії.

Наступним етапом буде налаштування сітки (рис. 3.4). Виконати триангуляцію області можна за допомогою команди `generateMesh(model, name, value)`. Для того, щоб зменшити крок сітки за замовченням, можна вказати ім'я (`name`), що надає `PDETool`, наприклад `Hmax`. `Hmax` – максимальна довжина ребра трикутника сітки.

```
Do you want to setup mesh? 1 - NO; 2 - YES;
2
Input mesh Name in quotes
'Hmax'
Input mesh Value
0.05
```

Рис. 3.4. Приклад налаштування сітки

Команда `applyBoundaryCondition` встановлює граничні умови Діріхле. Далі, необхідно вказати коефіцієнти рівняння c , a , f , (рис. 3.5) це можна зробити за допомогою функції `specifyCoefficients(mod,'m',0,'d',1,'c',c,'a',a,'f',f)`.

```
Input c coefficient
1
Input a coefficient
0
Input f coefficient
0
```

Рис. 3.5. Приклад коефіцієнтів рівняння

Фінальним етапом перед вирішенням на ВЗ, буде запит до користувача, вказати верхню границю для шуканих ВЗ, наприклад 100 (рис. 3.6).

```
Input limit for counted results:
100
```

| | | | | |
|------------|-------|-------|---------------|---|
| Basis= 10, | Time= | 0.30, | New conv eig= | 0 |
| Basis= 11, | Time= | 0.30, | New conv eig= | 0 |
| Basis= 12, | Time= | 0.36, | New conv eig= | 0 |
| Basis= 13, | Time= | 0.36, | New conv eig= | 0 |
| Basis= 14, | Time= | 0.36, | New conv eig= | 0 |
| Basis= 15, | Time= | 0.36, | New conv eig= | 0 |
| Basis= 16, | Time= | 0.36, | New conv eig= | 0 |
| Basis= 17, | Time= | 0.36, | New conv eig= | 0 |
| Basis= 18, | Time= | 0.36, | New conv eig= | 0 |
| Basis= 19, | Time= | 0.36, | New conv eig= | 0 |
| Basis= 20, | Time= | 0.36, | New conv eig= | 2 |
| Basis= 21, | Time= | 0.36, | New conv eig= | 2 |
| Basis= 22, | Time= | 0.36, | New conv eig= | 2 |
| Basis= 23, | Time= | 0.36, | New conv eig= | 2 |
| Basis= 24, | Time= | 0.39, | New conv eig= | 4 |
| Basis= 25, | Time= | 0.39, | New conv eig= | 4 |
| Basis= 26, | Time= | 0.39, | New conv eig= | 4 |
| Basis= 27, | Time= | 0.39, | New conv eig= | 4 |
| Basis= 28, | Time= | 0.39, | New conv eig= | 4 |
| Basis= 29, | Time= | 0.39, | New conv eig= | 4 |
| Basis= 30, | Time= | 0.39, | New conv eig= | 5 |

Рис. 3.6. Приклад вказаної границі шуканих ВЗ

Наближеним вирішенням задачі є вектор значень у вузлах сітки. Для їх вирішення використовуються спеціальні функції – солвери (solver) (табл. 3.1).

Таблиця 3.1

| Ім'я функції | Опис типу вирішуваних задач |
|--------------|---|
| asempde | Еліптичні рівняння та системи |
| parabolic | Параболічні рівняння та системи |
| hyperbolic | Гіперболічні рівняння та системи |
| pdenonlin | Нелінійні стаціонарні рівняння |
| adaptmesh | Адаптивна генерація сітки та вирішення еліптичних рівнянь та систем із заданою точністю |
| solverdeeig | Вирішення еліптичних задач на ВЗ |

Перед викликом солверу має бути створена матриця або файл-функція граничних умов b , задані триангуляція в масивах та задані коефіцієнти рівняння. Якщо вирішується нестационарне завдання, необхідно створити функції, що визначають рішення в початковий момент часу і вектор зі значеннями часу, в які слід знайти рішення. Потім ці дані передаються функцій солвер. Вони знаходять наближене вирішення завдання у вузлах сітки.

Для вирішення еліптичної задачі на ВЗ використовується солвер – solverdeeig. Вона вирішує проблему ВЗ ДРЧП в моделі для ВЗ у вказаному діапазоні. Якщо діапазон не містить жодних ВЗ, solverdeeig повертає об'єкт EigenResults з порожніми властивостями EigenVectors, EigenValues та Mesh. Дана функція є заміною застарілої функції pdeeig.

Завершенням програми буде, запит на побудову графіку (рис. 3.7) для одного з ВЗ, за замовчуванням це останнє знайдене ВЗ (рис. 3.8). Побудова графіка реалізована за допомогою функції `rdeplot`, що дозволяє будувати двовимірні та тривимірні графіки (рис. 3.9).

```
Do you want to display graphic for eigenvalue? 1 - NO; 2 - YES;
2
Do you want to display graphic for eigenvalue? 1 - NO; 2 - YES;
1
Do you want to export results to file (in c:/matlab-results)? 1 - NO; 2 - YES;
```

Рис. 3.7. Приклад запиту на відображення графіка для знайденого ВЗ

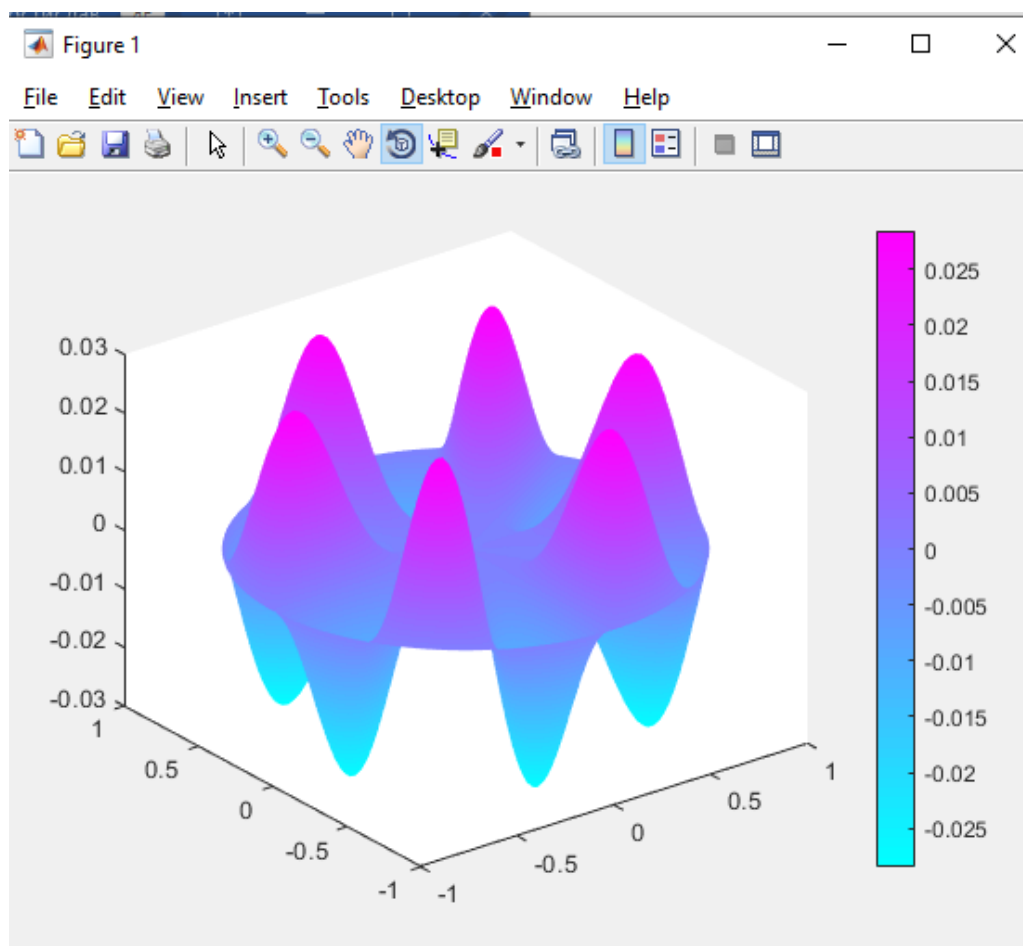


Рис.3.8. Тривимірний графік останнього знайденого λ_{21} ВЗ

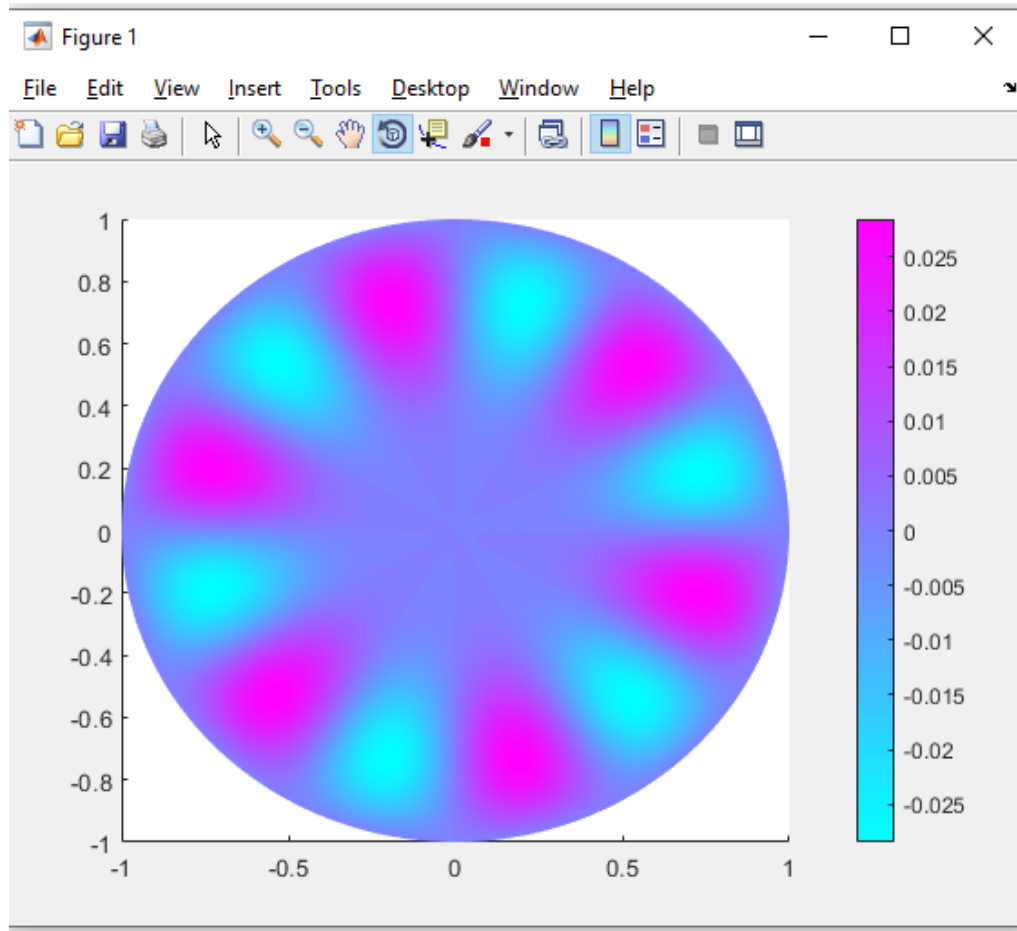


Рис. 3.9. Двовимірний графік останнього знайденого λ_{21} ВЗ

Також, доступна функція експорту отриманих ВЗ та ВВ у відповідні файли. Цю функцію продемонстровано в наступному прикладі.

3.2 Аналіз ефективності програмного забезпечення та отриманих результатів

У другому розділі, було розглянуто вирішення (2.10) рівняння методом розділення змінних.

Щоб порівняти результати, отримані за допомогою програми, для прямокутника зі сторонами $a = 1$, $b = 1$, було знайдено результат аналітичного рішення (2.10) з використанням програми Microsoft Excel (рис. 3.10) для $m = 1 \dots 6$, $n = 1 \dots 6$, та зафіксовано отримані перші 21 ВЗ (рис. 3.11),

| | A | B | C | D | E | F | G | H | I |
|---|---|---|---|----------|---|---|---|---|---|
| 1 | m | n | | pi | | | | | |
| 2 | 1 | 1 | | 3,141593 | | | | | |
| 3 | 2 | 2 | | | | | | | |
| 4 | 3 | 3 | | | | | | | |
| 5 | 4 | 4 | | | | | | | |
| 6 | 5 | 5 | | | | | | | |
| 7 | 6 | 6 | | | | | | | |

Рис. 3.10. Результат аналітичного рішення (2.10) у вигляді формули Microsoft Excel

| Lambdas | | | | | | |
|---------|----------|----------|----------|----------|----------|----------|
| m/n | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 19,73921 | 49,34802 | 98,69604 | 167,7833 | 256,6097 | 365,1754 |
| 2 | | 78,95684 | 128,3049 | 197,3921 | 286,2185 | 394,7842 |
| 3 | | | 177,6529 | 246,7401 | 335,5665 | 444,1322 |
| 4 | | | | 315,8273 | 404,6538 | 513,2194 |
| 5 | | | | | 493,4802 | 602,0459 |
| 6 | | | | | | 710,6115 |

Рис. 3.11. ВЗ знайдені за формулою (2.10)

Для перевірки роботи програми, та порівняння результатів її роботи з аналітичними, було запущено скрипт-сценарій, що дав можливість намалювати будь-яку область, обравши кількість елементів (рис. 3.12), та їх тип (рис. 3.13). Відкривається графічний інтерфейс, на якому можна побачити намальовану фігуру, після того як всі координати та назви були задані.

```

Command Window
>> rozdel_3
Choose what to draw: 1 - Basic circle, 2 - Basic square, 3 - Basic L-Shaped figure, 4 - Draw you own figure
4
How many figures to draw?
fx |

```

Рис. 3.12 Приклад побудови іншої фігури

Щоб намалювати один прямокутник, тому було введено цифру 1. Та обрано пункт 4 – прямокутник (рис. 3.13).

```
How many figures to draw?  
1  
Choose type: 1 - Circle, 2 - Ellipse, 3 - Polygon, 4 - Rectangle  
4
```

Рис. 3.13. Запит на обирання кількості фігур та типу

Задано чотири вершини для прямокутника по осі X та Y (рис. 3.14).

```
Input xmin of corner (vertice)  
0  
Input xmax of corner (vertice)  
1  
Input ymin of corner (vertice)  
0  
Input ymax of corner (vertice)  
1
```

Рис. 3.14. Встановлення вершин прямокутника

Задано назву прямокутника, наприклад R1 (рис. 3.15).

```
Input label  
'R1'|
```

Рис. 3.15. Встановлення назви прямокутника

Графічний інтерфейс автоматично оновився (рис. 3.16), демонструючи побудовану фігуру.

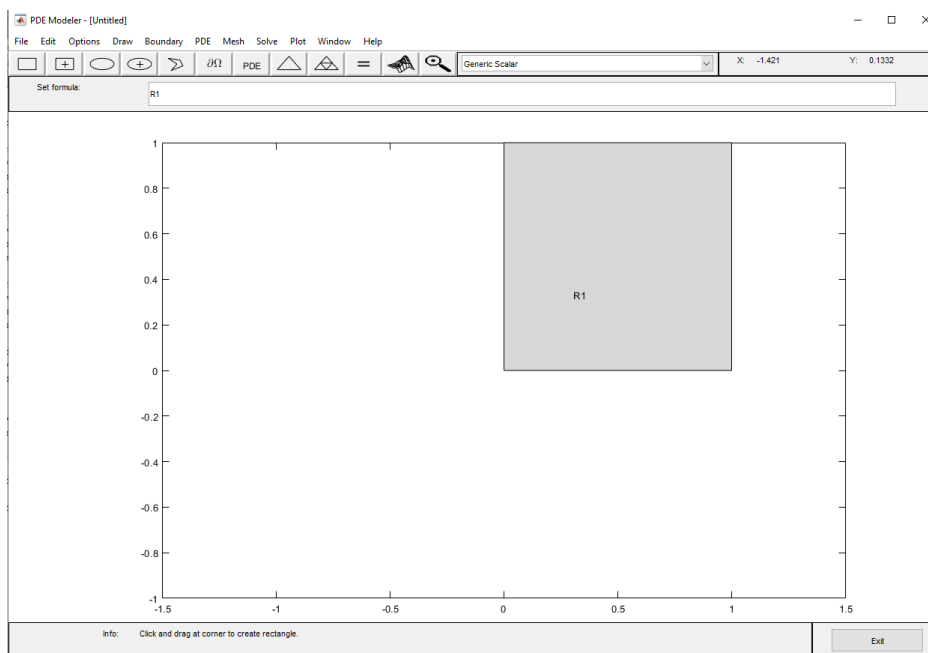


Рис. 3.16. Результат побудованої програмою фігури

Якщо фігур буде декілька, то усі кроки повторяться для нової фігури, після чого треба вказати формулу (рис. 3.18).

```
Set formula
R1
```

Рис. 3.18. Задання формули

Щоб порівняти значення в залежності від щільності сітки, в перший раз значення було залишено без змін (за замовчуванням) (рис. 3.19), після вводу значень з'явиться малюнок з сіткою (рис. 3.20).

```
Do you want to setup mesh? 1 - NO; 2 - YES;
1
```

Рис. 3.19. Встановлення сітки за замовчуванням

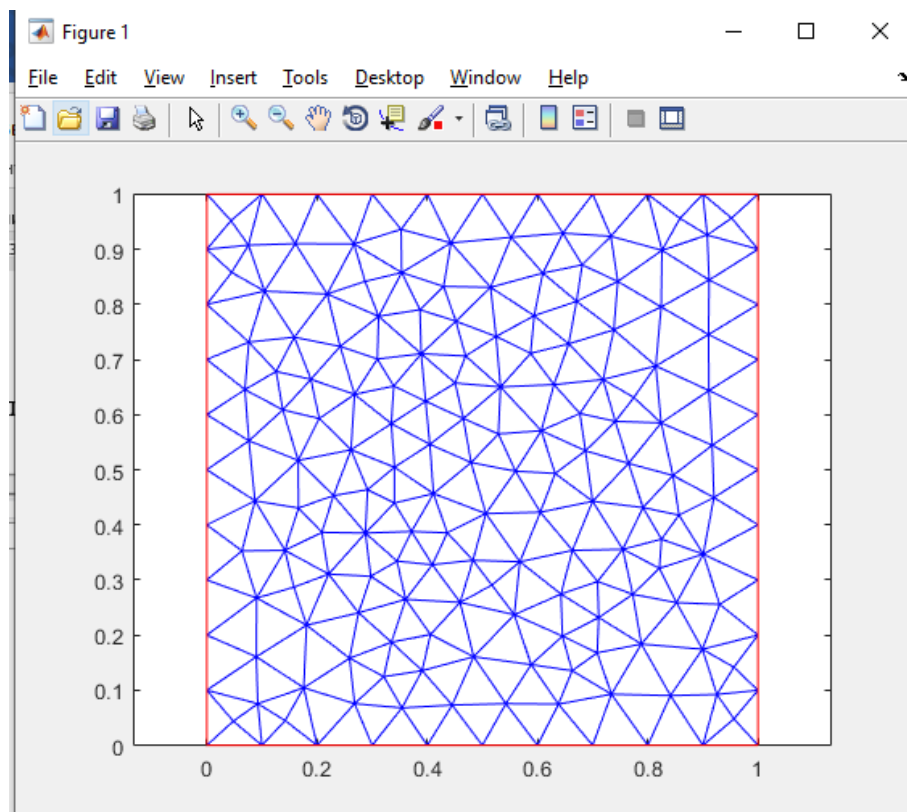


Рис. 3.20. Зображення сітки за замовчуванням

Необхідно вказати коефіцієнти рівняння та діапазон для ВЗ (рис. 3.21).

```

Input c coefficient
1
Input a coefficient
0
Input d coefficient
0
Input limit for counted results:
300

```

Рис. 3.21. Введено коефіцієнти та границю шуканих ВЗ

Підтвердивши експорт даних в файл, отримуємо два відповідних файли створених у директорії matlab-results (рис. 3.22), підтвердивши запит на малювання графіка, отримано відображення для останнього ВЗ (рис. 3.23).

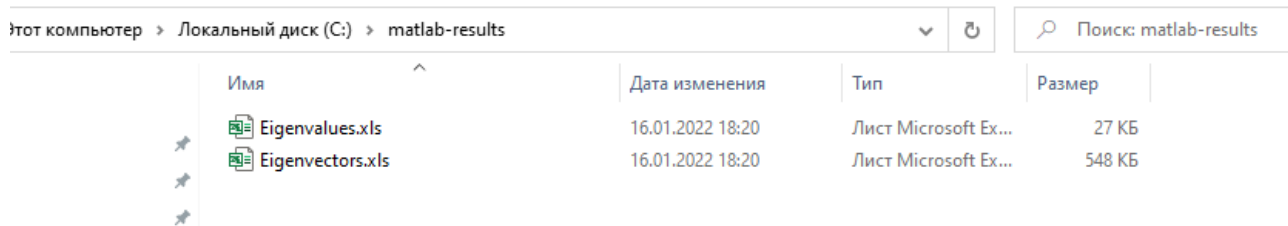


Рис. 3.22. Створені програмно файли з результатами

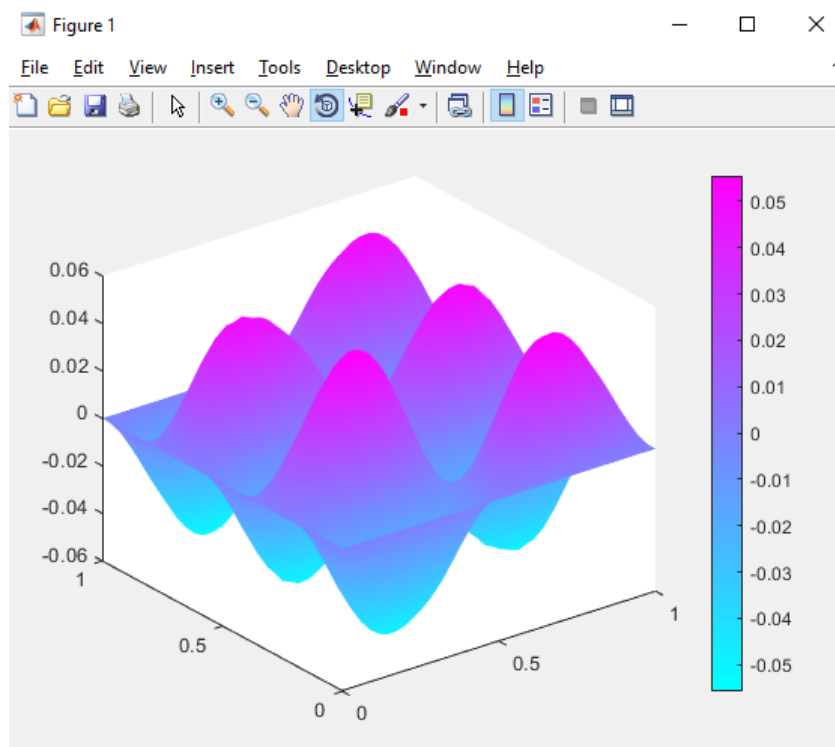


Рис. 3.23. Графік для 19-го знайденого ВЗ

Очевидно, що отримані результати (рис. 3.24) збігаються з аналітичними на рис. 3.11.

| | 1 | 2 | 3 |
|----|----------|---|---|
| 1 | 19.7393 | | |
| 2 | 49.3490 | | |
| 3 | 49.3490 | | |
| 4 | 78.9606 | | |
| 5 | 98.7036 | | |
| 6 | 98.7041 | | |
| 7 | 128.3211 | | |
| 8 | 128.3213 | | |
| 9 | 167.8188 | | |
| 10 | 167.8209 | | |
| 11 | 177.6952 | | |
| 12 | 197.4499 | | |
| 13 | 197.4513 | | |
| 14 | 246.8536 | | |
| 15 | 246.8554 | | |
| 16 | 256.7390 | | |
| 17 | 256.7466 | | |
| 18 | 286.3949 | | |
| 19 | 286.4000 | | |
| 20 | | | |

Рис. 3.24. ВЗ знайдені із сіткою за замовчуванням

Щоб, дослідити залежність точності отриманих результатів від щільності розбиття сітки, була вказана більша щільність сітки $H_{max} = 0.05$ (рис. 3.25).

```
>> rozdel_3
Choose what to draw: 1 - Basic circle, 2 - Basic square, 3 - Basic L-Shaped figure, 4 - Draw you own figure
4
How many figures to draw?
1
Choose type: 1 - Circle, 2 - Ellipse, 3 - Polygon, 4 - Rectangle
4
Input xmin of corner (vertice)
0
Input xmax of corner (vertice)
1
Input ymin of corner (vertice)
0
Input ymax of corner (vertice)
1
Input label
'R2'
Set formula
R2
Do you want to setup mesh? 1 - NO; 2 - YES;
2
Input mesh Name in quotes
'Hmax'
Input mesh Value
0.05
```

Рис. 3.25. Введені дані для налаштованої сітки

Отримано результати для більш щільної сітки (рис. 3.26), де N_{max} - це приблизна верхня межа на довжині краю сітки. Іноді generateMesh може створити сітку з деякими елементами, які перевищують N_{max} . Для generateMesh існують інші параметри налаштування, що дозволяють гнучко змінювати параметри сітки (табл. 3.2).

Таблиця 3.2

| Назва параметру | Опис параметру |
|-----------------|--|
| Hgrad | Темпи росту сітки, число, що більше або дорівнює 1 і менше або дорівнює 2. |
| Hmin | Приблизна нижня межа деякого ребра сітки |
| Hface | Вказати точну довжину граней, вказавши їх значення у масиві |
| Hedge | Вказати точну довжину ребер, вказавши їх значення у масиві |
| Hvertex | Вказати довжину навколо обраних вершин, вказавши їх значення у масиві |

| lambda | |
|-------------|----------|
| 19x1 double | |
| | 1 |
| 1 | 19.7392 |
| 2 | 49.3486 |
| 3 | 49.3487 |
| 4 | 78.9592 |
| 5 | 98.7010 |
| 6 | 98.7012 |
| 7 | 128.3155 |
| 8 | 128.3158 |
| 9 | 167.8076 |
| 10 | 167.8083 |
| 11 | 177.6806 |
| 12 | 197.4301 |
| 13 | 197.4307 |
| 14 | 246.8143 |
| 15 | 246.8157 |
| 16 | 256.6943 |
| 17 | 256.6969 |
| 18 | 286.3335 |
| 19 | 286.3364 |
| 20 | |

Рис. 3.26. ВЧ для налаштованої сітки

3.3 Висновок

Порівнявши результати, отримані аналітичним шляхом (рис. 3.11), результати для сітки за замовчуванням (рис. 3.24), та для більш щільної сітки (рис. 3.26), можна зробити висновок, що результати отримані за допомогою програми, збігаються з результатами отриманими методом Фур'є, що робить впровадження даного програмного забезпечення для вирішення інших задач на ВЗ та ВФ ДРЧП для тіл обертання ефективним. Також точність отриманих використовуючи ПЗ результатів, залежить від щільності дискретизації сітки, більш щільна сітка дає точніші результати.

РОЗДІЛ 4 ЕКОНОМІКА

Важливими етапами розробки програмного забезпечення є визначення трудомісткості розробки та розрахунок витрат на створення програмного продукту.

4.1 Визначення трудомісткості розробки програмного забезпечення

Задані дані:

1. передбачуване число операторів – 300;
2. коефіцієнт складності програми – 1,6;
3. коефіцієнт корекції програми в ході її розробки – 0,1;
4. годинна заробітна плата програміста, грн/год - 250;
5. коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі – 1,3;
6. коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності – 1;
7. вартість машино-години ЕОМ, грн/год - 10.

Нормування праці в процесі розробки програмного забезпечення суттєво ускладнено через творчий характер роботи програміста. Тому трудомісткість розробки ПЗ може бути розрахована на основі системи моделей з різною точністю оцінки за формулою:

$$t = t_o + t_u + t_a + t_n + t_{отл} + t_{\partial}, \text{ людино-годин, (4.1)}$$

де t_o – витрати праці на підготовку й опис поставленої задачі (приймається 50);

t_u - витрати праці на дослідження алгоритму рішення задачі;

t_a - витрати праці на розробку блок-схеми алгоритму;

t_n - витрати праці на програмування по готовій блок-схемі;

$t_{отл}$ - витрати праці на налагодження програми на ЕОМ;

t_{∂} - витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у ПЗ та розраховується за формулою:

$$Q = q \cdot C \cdot (1 + p),$$

де q - передбачуване число операторів;

C - коефіцієнт складності програми;

p - коефіцієнт корекції програми в ході її розробки.

$$Q = 300 \cdot 1,6 \cdot (1 + 0,1) = 528 \text{ , людино-годин.}$$

Витрати праці на вивчення опису задачі t_u визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t_u = \frac{Q \cdot B}{(75 \cdot 85) \cdot k} \text{ , людино-годин,}$$

де B - коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі;

k - коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності. При стажі роботи спеціаліста від 2 до 3 років коефіцієнт складає 1.

$$t_u = \frac{528 \cdot 1,3}{78 \cdot 1} = 8,8 \text{ , людино-годин.}$$

Витрати праці на розробку алгоритму рішення задачі визначаються за формулою:

$$t_a = \frac{Q}{(20 \dots 25) \cdot k}, \text{ людино-годин,}$$

де Q – умовне число операторів програми;

k – коефіцієнт кваліфікації програміста.

Підставимо у формулу відповідні значення, та отримаємо:

$$t_a = \frac{528}{20 \cdot 1} = 26,4, \text{ людино-годин.}$$

Витрати на складання програми по готовій блок-схемі:

$$t_n = \frac{Q}{(20 \dots 25) \cdot k}, \text{ людино-годин.}$$

$$t_n = \frac{528}{24 \cdot 1} = 22, \text{ людино-годин.}$$

Витрати праці на налагодження програми на ЕОМ:

- за умови автономного налагодження одного завдання:

$$t_{\text{отл}} = \frac{Q}{(4 \dots 5) \cdot k}, \text{ людино-годин.}$$

$$t_{\text{отл}} = \frac{528}{4 \cdot 1} = 132, \text{ людино-годин.}$$

- за умови комплексного налагодження завдання:

$$t_{\text{отл}}^k = 1,5 \cdot t_{\text{отл}}, \text{ людино-годин.}$$

$$t_{\text{отл}}^k = 1,5 \cdot 132 = 198, \text{ людино-годин.}$$

Витрати праці на підготовку документації визначаються за формулою:

$$t_{\partial} = t_{\partial p} + t_{\partial o}, \text{ людино-годин,}$$

де t_{dp} - трудомісткість підготовки матеріалів і рукопису:

$$t_{dp} = \frac{Q}{(15..20) \cdot k}, \text{ людино-годин,}$$

де t_{do} - трудомісткість редагування, печатки й оформлення документації:

$$t_{do} = 0,75 \cdot t_{dp}, \text{ людино-годин.}$$

Підставляючи відповідні значення, отримаємо:

$$t_{dp} = \frac{528}{16 \cdot 1} = 33 \text{ людино-годин.}$$

$$t_{do} = 0,75 \cdot 33 = 24,75 \text{ людино-годин.}$$

$$t_{\partial} = 33 + 24,75 = 57,75 \text{ людино-годин.}$$

Використовуючи формулу (4.1), отримано повну оцінку трудомісткості розробки програмного забезпечення:

$$t = 50 + 8,8 + 26,4 + 22 + 132 + 57,75 = 296,95 \approx 297 \text{ людино-годин.}$$

4.2 Витрати на створення програмного забезпечення

Витрати на створення ПЗ K_{no} включають витрати заробітну плату виконавця програми $Z_{3П}$ і витрат машинного часу, необхідного на налагодження програми на ЕОМ

$$K_{no} = Z_{3П} + Z_{MB}, \text{ грн.}$$

Заробітна плата виконавця визначається за формулою:

$$Z_{зп} = t \cdot C_{пр}, \text{ грн,}$$

де t – загальна трудомісткість, людино-годин;

$C_{пр}$ – середня годинна заробітна плата програміста, грн/година.

$$Z_{зп} = 297 \cdot 250 = 74250 \text{ грн.}$$

Вартість машинного часу, необхідного для налагодження програми на ЕОМ:

$$Z_{мв} = t_{омл} \cdot C_{мч}, \text{ грн,}$$

де $t_{омл}$ – трудомісткість налагодження програми на ЕОМ, год;

$C_{мч}$ – вартість машино-години ЕОМ, грн/год.

Визначені в такий спосіб витрати на створення програмного забезпечення є частиною одноразових капітальних витрат на створення АСУП.

$$Z_{мв} = 132 \cdot 10 = 1320 \text{ грн.}$$

Отже витрати на створення ПЗ $K_{по}$ дорівнюють:

$$K_{по} = 74250 + 1320 = 75570 \text{ грн.}$$

Очікуваний період створення ПЗ:

$$T = \frac{t}{B_k \cdot F_p}, \text{ міс,}$$

де B_k – число виконавців;

F_p - місячний фонд робочого класу (при 40 годинному робочому тижні $F_p = 176$ годин).

$$T = \frac{297}{1 \cdot 176} = 1,6, \text{ міс.}$$

Таким чином було розраховано наступні показники:

- Трудомісткість розробки ПЗ – 297 людино-годин;
- Витрати на створення – 75570 грн;
- Очікуваний період створення ПЗ – 1,6 міс.

4.3 Маркетингові дослідження ринку збуту розробленого програмного продукту

Доцільність розробки програмного продукту для вирішення задачі зі знаходження ВЧ та ВФ еліптичного диференціального оператора, обґрунтовується її важливістю для науковців та інженерів при розрахунках та моделюванні реальних фізичних процесів.

Практична цінність розробленого ПЗ полягає у ефективності його результатів. Отримані чисельні рішення задач є більш точними, що матиме значний вплив на реальні інженерні конструкції, наприклад прокатні котушки, рефлектори космічних апаратів, збільшить їх надійність та дозволить зменшити витрати. Також, використання розробленого програмного продукту дозволить зменшити витрати часу на вирішення прикладних задач знаходження ВЧ і функцій у порівнянні з аналогічними аналітичними методами, та не комп'ютеризованими чисельними методами.

Однак, розроблене ПЗ не є цілком універсальним, використовується для побудови довільних об'єктів лише для еліптичного диференціального оператора та потребує високої кваліфікації користувача у сферах математики та фізики.

4.4 Оцінка економічної ефективності впровадження програмного забезпечення

Оскільки кваліфікаційна робота має дослідницький характер, а розроблене програмне забезпечення створене з метою аналізу залежності відхилення отриманого рішення від аналітичного в залежності від щільності розбиття області, чисельний розрахунок економічного ефекту не може бути виконаний. Однак можливо виділити соціальний ефект від реалізації результатів роботи, а саме:

- прискорення обчислення ВЧ та функцій еліптичного диференціального оператора для тіл обертання;
- збільшення точності результатів у порівнянні з аналітичними методами.

У ході виконання роботи над даним економічним розділом, було визначено трудомісткість розробки програмного забезпечення (297 людино-годин), був проведений підрахунок витрат на створення програми (75570 гривень) та визначено очікуваний період створення ПЗ (1,6 місяці). Було проведено маркетинговий аналіз ринку збуту програмного продукту та визначено соціальний ефект від впровадження програмного продукту.

ВИСНОВКИ

Кваліфікаційна робота була присвячена розробці програмного забезпечення, для знаходження ВЧ та ВФ еліптичного диференціального оператора для тіл обертання, та дослідженню відхилення отриманого рішення від аналітичного, в залежності від щільності розбиття області.

Програмне забезпечення було створено з використанням Matlab PDE Tool для платформи Windows, алгоритм програми реалізує в своїй основі МСЕ та вирішує задачі на ВЗ та ВФ еліптичних диференціальних рівнянь. Аналітичне рішення було отримане за допомогою МРЗ. В результаті проведених досліджень були отримані ВЗ значення та ВФ, для квадрата зі сторонами, що дорівнюють одиниці, виявлена залежність точності отриманих результатів від щільності дискретизації області.

Отримані чисельні результати розглянутих задач є достатньо точними, тому програмне забезпечення може використовуватися науковцями та техніками при моделюванні реальних фізичних процесів та вирішенні прикладних задач математичної фізики і допоможе спростити та прискорити процес обчислення складних фігур на ВЗ та ВФ.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Собственные значения и собственные векторы. Примеры решений.
URL: http://mathprofi.ru/sobstvennyye_znachenija_i_sobstvennyye_vektory.html. – (дата звернення 15.11.2021).
2. Владимиров, В. С. Уравнения математической физики. Учебник для вузов. - 2-е изд., стереотип. / В. С. Владимиров, В. В. Жаринов – М.Ж ФИЗМАТЛИТ, 2004. – 400 с.
3. Буценко, Ю. П. Вступ до теорії рівнянь математичної фізики. Методичні вказівки. / Ю. П. Буценко, Н. М. Задерей, Г. Д. Нефьодова, Я. В. Симчук – К. НТУУ “КПІ”, 2012. – 56 с.
4. Маринець, В. В. Збірник задач з математичної фізики: навчальний посібник. – Вид. 2-ге виправлене і доповнене / В. В. Маринець, М. О. Перестюк, В. Л. Рего. – Кам’янець-Подільський: Аксіома, 2012. – 252 с.
5. Борковская, И. М. Уравнения математической физики : учебно-методическое пособие для студентов вузов, обучающихся по специальности "Автоматизация технологических процессов и производств" / И. М. Борковская, О. Н. Пыжкова. - Минск : БГТУ, 2010. – 75 с.
6. Макарьянц, Г. М. Основы метода конечных элементов: учебн. пособ. / Г. М. Макарьянц, А.Б. Прокофьев. - Самара: Изд-во Самар. гос. аэрокосм. ун-та, 2013. – 80 с.
7. Андреев, В. Б. Лекции по методу конечных элементов: Учебное пособие. / В. Б. Андреев – М.: Издательский отдел факультета ВМиК МГУ им. М.В. Ломоносова; МАКС Пресс, 2010. – 264 с.
8. Трудоношин, В. А. Системы автоматизированного проектирования: Учеб. пособие для вузов. / В. А. Трудоношин, Н. В. Пивоварова; Под ред. И.П.Норенкова. - Минск: Выш. шк., 1988. – 159 с.
9. Задачін, В. М. Чисельні методи. Навчальний посібник / В. М. Задачін, І. Г. Конюшенко – Харків. Вид. ХНЕУ ім. С. Кузнеця, 2014 р. – 180 с.

10. Дьяконов, В. П. MATLAB. Полный самоучитель. / В.П. Дьяконов – М.: ДМК Пресс, 2012. – 768 с.: ил.
11. Эдвардс, Ч. Г. Дифференциальные уравнения и краевые задачи: моделирование и вычисление с помощью Mathematica, Maple и MATLAB. 3-е издание / Ч. Г. Эдвардс, Д. Э. Пенни, – М.: Вильямс, 2008. — 1104 с. : ил.
12. Грудкіна, Н. С. Рівняння математичної фізики : посібник до практичних занять і самостійної роботи / Н. С. Грудкіна, С. О. Шевцов. – Краматорськ : ДДМА, 2019. – 47 с.
13. Ясев, О. Г. Моделювання динаміки технічних систем з використанням пакета програм Mathcad: Навч. посібник (рос. мовою) / О. Г. Ясев, В. Г. Расчубкін – Дніпропетровськ: НМетАУ, 2011. – 76 с.
14. Матях, С. В. Чисельно-аналітичне моделювання нелінійних процесів переносу зарядів в фотоелектричних та електрохімічних перетворювачах енергії : дис. ... канд. тех. наук: 05.14.08 / С. В. Матях – Інститут відновлюваної енергетики. – Київ, 2016 – 161 с. – Бібліогр. : С. 149 – 161.
15. Тихонов, А. Н. Уравнения математической физики, Учеб. Пособие. 6-изд. Испр. и доп. / А. Н. Тихонов, А. А. Самарский. – Москва : Изд-во МГУ, 1999. – 798 с.
16. Колесникова, С. И. Методы решения основных задач математической физики. – уч. пособие. / С. И. Колесникова – Москва: МФТИ, 2015. – 79 с.
17. Панов, Ю. Д. Математическая физика, Методы решения задач / Ю. Д. Панов, Р. Ф. Егоров. – Екатеринбург : Уральский государственный университет им. А. М. Горького, 2005. – 150 с.
18. Gedicke, J. Arnold-Winther MFEM for Stokes Eigenvalue Problems / J. Gedicke, A Khan, 2017. – 22 p.
19. Ayoub, T. J. Eigenvalue Problems on Atypical Domains The Finite Element Method / T. J. Ayoub, 2021. – 16 p.

20. Зенкевич, О. Конечные элементы и аппроксимация. Пер. с англ. / О. Зенкевич, К. Морган. – Москва : Мир, 1986. – 318 с.
21. Бахвалов, Н. С. Численные методы / Н. С. Бахвалов, Н. П. Жидков, Г. М. Кобельков. – Москва. : Бином, 2007. – 636 с.
22. Sun, J. Finite element methods for eigenvalue problems. Monographs and Research Notes in Mathematics / J. Sun, A. Zhou. – Boca Raton : CRC Press, 2017. – 367 p.
23. Штумпф, С. А. Математические методы компьютерных технологий в научных исследованиях. Учебное пособие / С. А. Штумпф, М. А. Бахтин. – Пермь : Пермский национальный исследовательский политехнический университет, 2015. – 21 с.
24. Gillman A., Fast and accurate numerical methods for solving elliptic difference equations defined on lattices / A. Gillman, P. G. – Martinsson Department of Applied Mathematics, 526 UCB; Journal of Computational Physics, 2010. – 24 p.
25. Борзенков, А. В. Дифференциальные уравнения в частных производных. МАТЛАБ: конспект лекций для студ. всех спец. БГУИР днев. формы обуч. / А. В. Борзенков. – Минск : БГУИР, 2009. – 120 с.
26. Коткин, Г. Л., Компьютерное моделирование физических процессов с использованием МАТЛАБ: Учеб. пособие / Г. Л. Коткин, В. С. Черкасский. – Новосибирск : Новосиб. ун-т., 2001. – 173 с.
27. Методичні рекомендації до виконання кваліфікаційних робіт здобувачами другого (магістерського) рівня вищої освіти спеціальностей 121 «Інженерія програмного забезпечення» та 122 «Комп'ютерні науки» / Б.І. Мороз, О.В. Іванченко, О.В. Реута, О.С. Шевцова; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро : НТУ «ДП», 2021.

Додаток А

КОД ПРОГРАМИ

```

rozdel_3.m
typeOfFigure = input('Choose what to draw: 1 - Basic circle, 2 - Basic square, 3
- Basic L-Shaped figure, 4 - Draw your own figure\n');

switch typeOfFigure
    case 1
        clear model
        model = createpde;
        geometryFromEdges(model,@circleg);
        countBasicFigure(model);
    case 2
        clear model
        model = createpde;
        geometryFromEdges(model,@squareg);
        countBasicFigure(model);
    case 3
        clear model
        model = createpde;
        geometryFromEdges(model,@lshapeg);
        countBasicFigure(model);
    case 4
        [pde_fig,ax] = pdeinit;
        model = createpde;
        countOfFigures = input('How many figures to draw?\n');
        ns=[];
        gd=[];
        for i = 1:countOfFigures
            ownType = input('Choose type: 1 - Circle, 2 - Ellipse, 3 - Polygon,
4 - Rectangle\n');
            switch ownType
                case 1
                    xc = input('Input x-coordinate of center\n');
                    yc = input('Input y-coordinate of center\n');
                    r = input('Radius \n');
                    label = input('Input label\n');
                    pdecirc(xc,yc,r,label);

gd=get(findobj(get(pde_fig,'Children'),'flat','Tag','PDEMeshMenu'),'UserData');
                case 2
                    xc = input('Input x-coordinate of center\n');
                    xy = input('Input y-coordinate of center\n');
                    a = input('Input semiaxe a\n');
                    b = input('Input semiaxe b\n');
                    phi = input('Input rotation phi in rads\n');
                    label = input('Input label\n');
                    pdeellip(xc,xy,a,b,phi,label);

gd=get(findobj(get(pde_fig,'Children'),'flat','Tag','PDEMeshMenu'),'UserData');
                case 3
                    x = input('Input x-coordinate of corner (vertice)\n');
                    y = input('Input y-coordinate of corner (vertice)\n');
                    label = input('Input label\n');
                    pdepoly(x,y,label)

gd=get(findobj(get(pde_fig,'Children'),'flat','Tag','PDEMeshMenu'),'UserData');
                case 4

```

```

        xmin = input('Input xmin of corner (vertice)\n');
        xmax = input('Input xmax of corner (vertice)\n');
        ymin = input('Input ymin of corner (vertice)\n');
        ymax = input('Input ymax of corner (vertice)\n');
        label = input('Input label\n');
        pderect([xmin xmax ymin ymax], label);

gd=get(findobj(get(pde_fig, 'Children'), 'flat', 'Tag', 'PDEMeshMenu'), 'UserData');
    end
    chars = double(char(label));
    for j = 1:length(chars)
        ns(j,i)=chars(j);
    end
end

sf=input('Set formula\n', 's');
domain = decsg(gd, sf, ns);
geometryFromEdges(model, domain);
[p,e,t]=initmesh(domain);
pdemesh(p,e,t); axis equal
countMesh(model);
c = input('Input c coefficient\n'); %1
a = input('Input a coefficient\n'); %0
d = input('Input d coefficient\n'); %1

applyBoundaryCondition(model, 'dirichlet', 'Edge', 1:model.Geometry.NumEdges, 'u', 0)
;
    specifyCoefficients(model, 'm', 0, ...
        'd', d, ...
        'c', c, ...
        'a', a, ...
        'f', 0);

    limit = input('Input limit for counted results: \n'); % 100
    evLimit = [0, limit];

    results = solvepdeeig(model, evLimit);
    u = results.Eigenvectors;
    lambda = results.Eigenvalues;
    stop = true;
    workaroundResultsPrompt(stop, model, u, lambda);
end

function countBasicFigure(mod)
    countMesh(mod);

applyBoundaryCondition(mod, 'dirichlet', 'Edge', 1:mod.Geometry.NumEdges, 'u', 0);

    c = input('Input c coefficient\n'); %1
    a = input('Input a coefficient\n'); %0
    f = input('Input f coefficient\n'); %0

    specifyCoefficients(mod, 'm', 0, ...
        'd', 1, ...
        'c', c, ...
        'a', a, ...
        'f', f);
    limit = input('Input limit for counted results: \n'); % 100

```

```

    evLimit = [0,limit];
    results = solvepdeeig(mod,evLimit);

    u = results.Eigenvectors;
    lambda = results.Eigenvalues;
    stop = true;
    workaroudResultsPrompt (stop,mod,u,lambda);
end

function countMesh(md)
    userChoice = input('Do you want to setup mesh? 1 - NO; 2 - YES; \n');
    switch userChoice
        case 1
            generateMesh(md);
        case 2
            meshName = input('Input mesh Name in quotes \n');
            meshValue = input('Input mesh Value \n');
            generateMesh(md,meshName,meshValue); % 'Hmax' 0.05
    end
    return
end

function workaroudResultsPrompt(st,md,value,lmb)
    while(st == true)
        keepGoing = input('Do you want to display graphic for eigenvalue? 1 -
NO; 2 - YES; \n');
        if keepGoing == 1
            exportFlag = input('Do you want to export results to file (in
c:/matlab-results)? 1 - NO; 2 - YES; \n');
            if exportFlag == 1
                st = false;
            else
                folder = 'c:/matlab-results';
                if ~exist(folder, 'dir')
                    mkdir(folder);
                end

                eigenvaluesFileName1 = fullfile(folder, 'Eigenvectors.xls');
                eigenvaluesFileName2 = fullfile(folder, 'Eigenvalues.xls');
                xlswrite(eigenvaluesFileName1, value);
                xlswrite(eigenvaluesFileName2, lmb);
                st = false;
            end
        end
        else
            l = length(lmb);
            pdeplot(md,'XYData',value(:,l),'ZData',value(:,l));
        end
    end
end
end
...
...
...

```


Додаток В

ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ

| Ім'я файлу | Опис |
|------------------------|--|
| Пояснювальні документи | |
| Yashkin.doc | Пояснювальна записка до кваліфікаційної роботи. Документ Word. |
| Vasylenko.pdf | Пояснювальна записка до кваліфікаційної роботи в форматі PDF. |
| Програма | |
| Yashkin.zip | Архів. Містить коди програми. |
| Презентація | |
| Yashkin.ppt | Презентація кваліфікаційної роботи. |