

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

бакалавра

(назва освітньо-кваліфікаційного рівня)

студента *Козоріз Олександри Павлівни*

(ПІБ)

академічної групи *122-19-3*

(шифр)

спеціальності *122 Комп'ютерні науки*

(код і назва спеціальності)

освітньої програми *Комп'ютерні науки*

(назва освітньої програми)

на тему: *Розробка програмного забезпечення розрахунку*

температурних полів задачі Діріхле та їх візуалізація

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	<i>проф. Бердник М.Г.</i>			
розділів:				
спеціальний	<i>проф. Бердник М.Г.</i>			
економічний	<i>проф. Вагонова О.Г.</i>			
Рецензент				
Нормоконтролер	<i>доц. Гуліна І.Г.</i>			

Дніпро
2023

РЕФЕРАТ

Пояснювальна записка: 68 с., 54 рис., 0 табл., 3 дод., 29 джерел.
Об'єкт розробки: розподіл температурних полів при умовах Діріхле.

Мета кваліфікаційної роботи розробити програмне забезпечення для визначення температурних полів, що дозволить зменшити кількість витраченого часу та кількість помилок під час розрахунку.

У вступі розглядається аналіз і поточний стан проблеми, з'ясовується мета ідентифікаційної роботи та сфера її застосування, аргументується актуальність теми, уточнюється постановка завдання.

У першій частині аналізується предметна область, визначається релевантність завдання та мета розробки, розробляється формулювання завдання та визначаються вимоги до впровадження програмного забезпечення, методи та програмні засоби.

Другий розділ аналізує доступні рішення, вибирає платформу розробки, проектує та розробляє програму, описує роботу програми, алгоритм та її операційну структуру, визначає виклик і завантаження програми, визначає вхідні та вихідні дані, а також характеризує склад параметрів технічних засобів.

В економічному розділі визначається трудомісткість розробки інформаційної системи, розрахунок вартість робіт зі створення програми та час створення.

Практичне значення полягає у створенні додатка, що надає можливість підрахунку температури для різних поверхонь.

Актуальність інформаційної системи визначається великим попитом на подібні розробки, що оптимізують та спрощують дії щодо підрахунку, скорочують час; зменшують вірогідність помилки.

Список ключових слів: **ТЕМПЕРАТУРНЕ ПОЛЕ, ТЕПЛОВРОВІДНІСТЬ, МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ, МЕТОД СКІНЧЕНИХ ЕЛЕМЕНТІВ, МЕТОД ГАЛЕРКІНА**

ABSTRACT

Explanatory note: 68 p., 54 pictures, 0 tables, 3 add., 29 sources.

Object of development: software for calculating temperature fields

The goal of the diploma project: to create a program for determining temperature, thanks to which the amount of time spent and the number of errors during calculation will be reduced.

In the introduction, the analysis and current state of the problem is considered, the purpose of the identification work and the scope of its application are clarified, the topicality of the topic is argued, and the statement of the task is clarified.

In the first part, the subject area is analyzed, the relevance of the task and the purpose of development are determined, the formulation of the task is developed, and the requirements for software implementation, methods and software tools are determined.

The second section analyzes the available solutions, selects a development platform, designs and develops the program, describes the operation of the program, the algorithm and its operational structure, defines the call and download of the program, defines the input and output data, and also characterizes the composition of the parameters of the technical means.

In the economic section, the labor intensity of the development of the information system, the calculation of the cost of work on the creation of the program and the time of creation are determined.

The practical value is in creating an application that provides the ability to calculate the temperature for various surfaces.

The relevance of the information system is determined by the great demand for similar developments that optimize and simplify counting actions, reduce time; reduce the probability of error.

List of keywords: TEMPERATURE FIELD, THERMAL CONDUCTIVITY, MATHEMATICAL MODELING, FINITE ELEMENT METHOD, GALERKIN'S METHOD

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ПЗ-програмне забезпечення;

МСЕ- метод скінчених елементів;

ОЗП- оперативних запам'ятовуючий пристрій;

.m – формат файлу програми Matlab.

ЗМІСТ

РЕФЕРАТ.....	3
ABSTRACT.....	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	5
ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ	
1.1. Загальні відомості з предметної галузі.....	10
1.2. Призначення розробки та галузь застосування.....	11
1.3. Підстава для розробки.....	12
1.4. Постановка завдання.....	12
1.5. Вимоги до програми або програмного виробу.....	13
1.5.1. Вимоги до функціональних характеристик.....	13
1.5.2. Вимоги до інформаційної безпеки.....	14
1.5.3. Вимоги до складу та параметрів технічних засобів.....	14
1.5.4. Вимоги до інформаційної та програмної сумісності	14
РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ	
2.1. Функціональне призначення системи.....	16
2.2. Опис застосованих математичних методів.....	16
2.2.1 Метод скінчених елементів.....	16
2.2.2 Метод Галеркіна.....	17
2.2.3 Метод Ейлера.....	20
2.2.4 Рівняння теплопровідності.....	20
2.3. Опис використаних технологій та мов програмування.....	21
2.4. Опис структури системи та алгоритмів її функціонування	26
2.5. Обґрунтування та організація вхідних та вихідних даних програми.....	27
2.6. Опис розробленої системи	29
2.6.1. Використані технічні засоби.....	30
2.6.2. Використані програмні засоби.....	32
2.6.3. Виклик та завантаження програми.....	36
2.6.4. Опис інтерфейсу користувача.....	38

РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ

3.1. Розрахунок трудомісткості та вартості розробки програмного продукту	63
3.2. Рахунок витрат на створення програми.....	68
ВИСНОВКИ.....	70
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	72
Додаток А. Код програми.....	75
Додаток Б. Відгук керівника економічного розділу.....	95
Додаток В. Перелік файлів на диску.....	96

ВСТУП

Вивчення процесу теплообміну відіграє важливу роль у розвитку науки і техніки та природознавства. На початку минулого століття дослідження в цій галузі переважно надихалися потребами теплоенергетики того часу. Розвиток авіації, ядерної енергетики, ракетобудування та аерокосмічних технологій надав нові формулювання проблеми теплообміну, поряд з новими, суворішими теоріями, які вимагали цілісності та надійності прогностичних можливостей.

Поглиблене вивчення та діапазон застосування явищ теплопередачі значно розширилися за останні кілька десятиліть. Вона включає у собі як передові технологічні напрями (хімічна технологія, металургія, будівництво, розробка родовищ нафти, машинобудування, сільськогосподарське машинобудування та інших), а й фундаментальні природничі напрями (біологія, геологія, атмосферна і морська фізика та інших.).

Теоретичні дослідження процесів теплообміну нині переважно засновані на чисельному моделюванні з використанням ЕОМ. Це стало можливим завдяки великим досягненням у розвитку обчислювальних методів розв'язання задач рівнянь у приватних похідних, а також збільшеним можливостям сучасних комп'ютерів.

Особливу увагу слід приділити таким ситуаціям. Чисельне моделювання процесів теплообміну в даний час набуває все більшого значення, оскільки сучасна наука і техніка вимагають надійного прогнозування цих процесів, а експериментальні дослідження в лабораторних або натурних умовах дуже складні і дорогі, а в ряді випадків просто неможливі. процесів теплообміну дедалі успішніше впроваджується у практику різних науково-дослідних, проектних і виробничих організацій.

Добре відомо, що існує три механізми передачі тепла: теплопровідність, конвекція та випромінювання. У цій роботі ми зосередимося на основному механізмі теплопередачі – теплопровідності.

Теплопровідність – це молекулярне перенесення тепла в суцільному середовищі. Цей процес відбувається при нерівномірному розподілі

температури. У цьому випадку тепло передається за рахунок прямого контакту частинок із різними речовинами.

Температура, внаслідок якої відбувається обмін енергією між молекулами, атомами чи вільними електронами.

Теплопровідність залежить від агрегатного стану, складу, чистоти, температури, тиску та інших характеристик речовини, тому в більшості випадків теплопровідність рідкої речовини приблизно в 10 разів більша, ніж у рідкому стані. газоподібний стан. Для твердих тіл вона значно вища, ніж для рідин поблизу точки плавлення (крім рідкого вісмуту, олова, телуру).

Насправді часто буває, що теплопровідність всередині об'єкта і поблизу його кордонів різна. зміцнення, старіння, знос та ін.). Зовнішні фактори, такі як опромінення, перепади тиску, магнітні поля можуть істотно впливати на теплопровідність. У напівпрозорих середовищах теплопровідність супроводжується радіаційним теплообміном. Ефективна теплопровідність таких середовищ, що спостерігається, являє собою суму фактичної теплопровідності і променистого теплообміну.

Частка випромінювання у складному теплообміні збільшується з підвищенням температури і стає значною при температурах сотні градусів Цельсія.

Метою кваліфікаційної роботи є розробка програмного забезпечення для розрахунку температурних полів при умовах Діріхле.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1. Загальні відомості з предметної області

Термічні процеси найпоширеніші у виробництві продуктів харчування. До них відносяться нагрівання та охолодження, випаровування та конденсація. Середовище, що випромінює тепло за більш високої температури, називається теплоносієм. Низькотемпературне середовище, що поглинає тепло, називається холодоагентом. Теплопередача завжди відбувається у бік зменшення температури.

Теплопередача – це мимовільний та незворотний процес передачі тепла від гарячішого тіла (або частини тіла) до холоднішого тіла. Теплота (теплотворна здатність) є енергетичною властивістю процесу теплообміну та визначається кількістю енергії, яку тіло віддає або одержує при теплообміні.

Відомі три способи передачі тепла:

- теплопровідність (у твердих тілах та рідинах);
- конвекція (рух рідин та газів);
- випромінювання або випромінювання (між двома тілами через проміжне середовище за допомогою електромагнітних коливань).

Зазвичай спостерігаються складні теплообміни, що передбачають одночасне перенесення тепла різними шляхами (конвективно-провідний, конвективно-радіаційний). Складний теплообмін між двома середовищами, що рухаються, розділеними контактом фаз або поверхнею нагріву (твердою стінкою), називається теплообміном. Теплопровідність – це процес передачі теплової енергії.

Передача тепла від гарячішої частини тіла до холоднішої частини тіла та взаємодії мікрочастинок. Температура тіла підвищується за рахунок теплопровідності.

Вирівнює поверхню тіла, де всі точки мають однакову температуру називається ізотермічною поверхнею.

Температура в тілі (середовищі) змінюється в напрямку від однієї ізотермічної поверхні до іншої.

Зміна відбувається в напрямку від найбільшої температури до найменшої вздовж напрямку, нормального до ізотермічної поверхні. Ізотермічна поверхня у напрямку нормалі називається градієнтом температури.

Конвекція - перенесення теплоти під час переміщення об'єму рідини або газу з більш нагрітого простору в менш нагрітий.

Теплопередача - це процес теплообміну між поверхнею тіла і навколишнім середовищем.

Інтенсивність теплообміну характеризується коефіцієнтом тепловіддачі α . Він дорівнює відношенню густини теплового потоку на межі розділу до температурного тиску між поверхнею теплопередачі та середовищем (теплоносієм). При конвективному теплообміні тепло в потоці рідини або газу одночасно передається від (або до) поверхні твердого тіла шляхом конвекції та провідності.

Вільна конвекція (природна конвекція) - це рух частинок рідини або газу в об'ємі пристрою. Вона виникає через різницю в густині між нагрітими та охолодженими частинками рідини або газу. Швидкість природної конвекції залежить від фізичних властивостей рідини або газу, різниці температур між гарячими та холодними частинками та об'єму, в якому відбувається природна конвекція.

Під впливом насоса, або вентилятора відбувається вимушена конвекція яка визначається фізичними властивостями середовища.

За вимушеної конвекції теплообмін відбувається набагато інтенсивніше, ніж за природної конвекції.

1.2. Призначення розробки та галузь застосування

Явище теплопередачі має велике значення в багатьох галузях сучасної техніки. Процеси теплопередачі відіграють важливу роль в енергетиці, авіації, суднобудуванні, ракетобудуванні, промисловості та газотранспортних системах.

Теорія теплопровідності використовується в усіх розрахунках енергетичних установок, робочих процесів і теплового захисту.

Усі вони ґрунтуються на теорії теплопровідності, тому теплопровідність є фундаментальним поняттям для майбутніх фахівців у галузі енергетичної системи.

1.3. Підстава для розробки

Підстави для розробки та виконання кваліфікаційної роботи:

- Освітня програма 122 «Комп'ютерні науки»;
- Навчальний план та графік навчального процесу;
- Наказ ректора Національного технічного університету «Дніпровська політехніка» №350-с від 16.05.2023 р;
- Завдання на кваліфікаційну роботу на тему «Розробка програмного забезпечення розрахунку температурних полів задачі Діріхле та їх візуалізація».

1.4. Постановка завдання

Розробити програмного забезпечення розрахунку температурних полів задачі Діріхле та їх візуалізація.

В програмному забезпеченні необхідно реалізувати:

- 1) Простий та зрозумілий інтерфейс.
- 2) Зручний спосіб введення даних.
- 3) Візуалізацію отриманого рішення.
- 4) Порівняння знайденого рішення з аналітичним розв'язком , якщо таке рішення існує.
- 5) Перевірка вхідних даних на їх коректність.
- 6) Обробка виняткових ситуацій.

Чисельне рішення характеризується наступними показниками:

- коефіцієнти рівняння;

- граничні умови;
- область, в якій шукається рішення;
- щільністю дискретизації області.

Вихідною інформацією програми є вектор знайдених температурних полів

Вся вхідна інформація повинна перевірятися на коректність та відповідність очікуваному типу.

1.5. Вимоги до програми або програмного виробу

Розглянемо особливості програмування під Android. Клас Activity - найважливіший клас, з якого будується програма Android. Цей клас представляє візуальну активність програми, і визначає дії, які може робити користувач. Майже всі екземпляри та різновиди activity безпосередньо взаємодіють з користувачем, за створення вікна в якому відповідає клас Activity, в якому можете розмістити свій візуальний інтерфейс.

1.5.1. Вимоги до функціональних характеристик

Вимогами при створенні програмного забезпечення були:

- Сайт повинен відкриватись на будь-яких комп'ютерах, які підключені до інтернету и мають браузер;
- простий та зрозумілий інтерфейс;
- зручний спосіб введення даних;
- візуалізацію отриманого рішення;
- можливість імпорту рішення у файл;
- порівняння знайденого рішення з аналітичним розв'язком, якщо таке рішення існує;
- перевірку вхідних даних на їх коректність;
- обробку виняткових ситуацій.

1.5.2. Вимоги до інформаційної безпеки

Так як програма буде розміщений на сервері загального користування, необхідно буде задати вхід адміністратора використовуючи логін і пароль на самому сервері, для безпеки було використано кіберагент.

1.5.3. Вимоги до складу та параметрів технічних засобів

Системні вимоги (мінімальні):

- процесор 2 ядра по 2ГГц;
- ОЗП 256 мб або більше;
- будь який браузер.

Рекомендовані системні вимоги:

- процесор 2 ядра 3ГГц;
- ОЗП 1024мб;
- будь який браузер.

1.5.4. Вимоги до інформаційної та програмної сумісності

Для передачі даних між activity існує intent – це об'єкт, у якому ми прописуємо, яку Activity необхідно викликати. Після чого ми передаємо цей Intent-об'єкт методу startActivity, який знаходить відповідну Activity та показує його.

Для отримання доступу до віджетів використовується метод findViewById(int). Для реагування кнопок натискання використовується метод setOnClickListener (View.OnClickListener l). На вхід подається об'єкт з інтерфейсом View.OnClickListener.

Компонент TextView призначений для відображення тексту без можливості редагування користувачем, що видно з його назви. Програмно задається за допомогою методу setText().

Компонент EditText — це текстове поле для введення користувача, яке використовується, якщо необхідно редагування тексту. Слід зазначити, що

EditText є спадкоємцем TextView.

Віджет ListView є список елементів, що прокручується. Компонент ListView більш складний у застосуванні, порівняно з TextView та іншими простими елементами. Список використання складається з двох частин. Спочатку ми додаємо сам ListView до форми, а потім заповнюємо його елементами списку. Компоненту ListView потрібні дані для заповнення. Джерелом заповнення може бути масив, база даних. Для прив'язки даних до списків використовуються так звані адаптери. Метод setAdapter(ListAdapter) пов'язує підготовлений перелік із адаптером.

РОЗДІЛ 2.

ПРОЕКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1. Функціональне призначення системи

В розробленому програмному забезпеченні основними вимогами було створення простого та зручного інтерфейсу завдяки чому спрощується користування . Користувач з легкістю може вводити свої данні та отримати візуалізацію рішення яке він отримав. Після цього користувач має змогу порівняти знайдене рішення з аналітичним розв'язком ,якщо таке буде існувати. Данні які він отримав можуть бути коефіцієнтами рівняння, граничними умовами, областю в якому шукається рішення та щільністю дискретної області дивлячись на те що він обрав.

Вся вхідна інформація повинна бути перевірена на правильність і очікуваний тип.

2.2.Опис застосованих математичних методів

2.2.1. Метод скінченних елементів

Історичними попередниками методу кінцевих елементів (МКЕ) були різні методи будівельної механіки та механіки твердого тіла, що деформується. У 19 столітті французький математик і фізик Пуассон пропонував розглядати суцільне середовище як систему кінцевих обсягів, а Д.Максвеллом та А.Кастільяно було закладено основи аналізу стрижневих конструкцій. Після цього було сформульовано метод сил і потім метод переміщень. Завдяки технічному прогресу в 20 столітті відбувся швидкий розвиток в галузі авіації та космонавтики, що сприяло виникненню різних матричних методів аналізу конструкцій. Значний внесок у розвиток будівельної механіки зробив Д.Аргіріс.

Перше повідомлення про розрахунок конструкцій методом кінцевих елементів з'явилося в 1956 р. у статті М.Тернера, Р.Клафа, Г.Мартіна та Л.Топпа. Р.Клаф 1960 р. ввів термін «кінцеві елементи». [10] У 1959 Д.Тернер запропонував метод прямої жорсткості як комп'ютерну реалізацію методу

кінцевих елементів. У 1963 році метод отримав широку популярність після того, як було доведено його зв'язок з методом Релея — Ритца, який зводить поставлене завдання до системи лінійних рівнянь рівноваги та шукає наближене рішення. Область застосування МКЕ значно розширилася 1968 року. Було встановлено, що рівняння, що визначають елементи в завданнях, можуть бути отримані методом Галеркіна або методу найменших квадратів. Ці факти вплинули обґрунтування МКЭ, який став застосовуватися для чисельного розв'язання диференціальних рівнянь і побудови дискретних апроксимацій. [9]

Подальші дослідження були націлені на отримання програмного забезпечення з відкритим кодом для реалізації цього методу. Наприкінці 1960-х NASA випустило ПЗ NASTRAN для аерокосмічного аналізу.[6] Суворе математичне обґрунтування для методу кінцевих елементів було представлено в 1973 Джорджем Фіксом і Гільбертом Стренгом. Найбільш відома їхня публікація – «Теорія методу кінцевих елементів». У ній розглядається вплив різних факторів на обчислювальну ефективність методу кінцевих елементів. [7]

В даний час існує велика кількість сучасних МКЕ комплексів, такі, як Ansys, MSC/Nastran або SAP-7, що включають теплові, рідинні та структурні компоненти. Поява таких продуктів дозволило прискорити процес проектування об'єктів аерокосмічної, авіаційної чи машинобудівної галузі десятки раз. [8]

2.2.2. Метод Галеркіна

Для визначення параметра u_i , що використовується в рівнянні, сформуємо алгебраїчну лінійну систему рівнянь методом Галеркіна, інтегруючи добуток пробних функцій кусково-лінійної неперервної сім'ї функцій по області існування розв'язку:

$$\int_{x_{min}}^{x_{max}} [\varphi_j(x) \sum_{i=1}^n u_i \varphi_i(x)] dx = \int_{x_{min}}^{x_{max}} [y(x)] \varphi_i(x) dx , \quad (2.1)$$

де

$$j = 1, 2, \dots, n;$$

$y(x)$ – функція початкового температурного профілю.

В отриманій системі алгебраїчних лінійних рівнянь деякі інтеграли лівої частини утворюють смугасту квадратну матрицю тридіагональної структури, яка складається з двох інтегралів на головній діагоналі матриці:

$$m_{i,i} = \int_{x_{min}}^{x_{max}} \varphi_i(x) \varphi_i(x) dx = \frac{2}{3} \frac{x_{max} - x_{min}}{n+1}. \quad (2.2)$$

І над і під головною діагоналлю:

$$m_{k,k+1} = m_{i,i-1} = \int_{x_{min}}^{x_{max}} \varphi_i(x) \varphi_{i-1}(x) dx = \frac{1}{6} \frac{x_{max} - x_{min}}{n+1}. \quad (2.3)$$

Спростимо завдання, вважаючи, що $y(x)=1$. У цьому випадку стовпець вільних членів p_i є величиною, що визначається інтегралом:

$$\int_{x_{min}}^{x_{max}} \varphi_i(x) dx = \frac{x_{max} - x_{min}}{n+1}.$$

Таким чином, рівняння приймає матричну форму (рис. 2.1).

$$\begin{array}{c} \left| \begin{array}{ccccc} \frac{2}{3} & \frac{1}{6} & 0 & 0 & 0 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & 0 \\ 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 \\ \hline 0 & 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ 0 & 0 & 0 & \frac{1}{6} & \frac{2}{3} \end{array} \right| \cdot \begin{array}{c} \left| \begin{array}{c} u_1 \\ u_2 \\ u_3 \\ \hline u_{n-1} \\ u_n \end{array} \right| = \begin{array}{c} \left| \begin{array}{c} 1 \\ 1 \\ 1 \\ \hline 1 \\ 1 \end{array} \right| \end{array}$$

Рис. 2.1. Матрична форма кусково-лінійних безперервних функцій

Матрицею тут є тридіагональною, яку можна розв'язати методом Гаусса, матричним методом (множення оберненої матриці) або прогонкою. Для систем низької розмірності метод рішення не є необхідним, але зі збільшенням точності рішення кількість скінченних елементів в області рішення потрібно

збільшувати, але це призведе до збільшення кількості невідомих у рівнянні. Для визначення впливу кількості скінченних елементів на інтервалі $x_{\min} \dots x_{\max}$ на точність початкового наближення використовується матричний метод розв'язування системи для різних чисел n в інженерному середовищі MathCAD, а середнє значення $u(x)$ визначається на цьому інтервалі. Відносні похибки для різної кількості скінченних елементів наведені нижче (рис. 2.2).

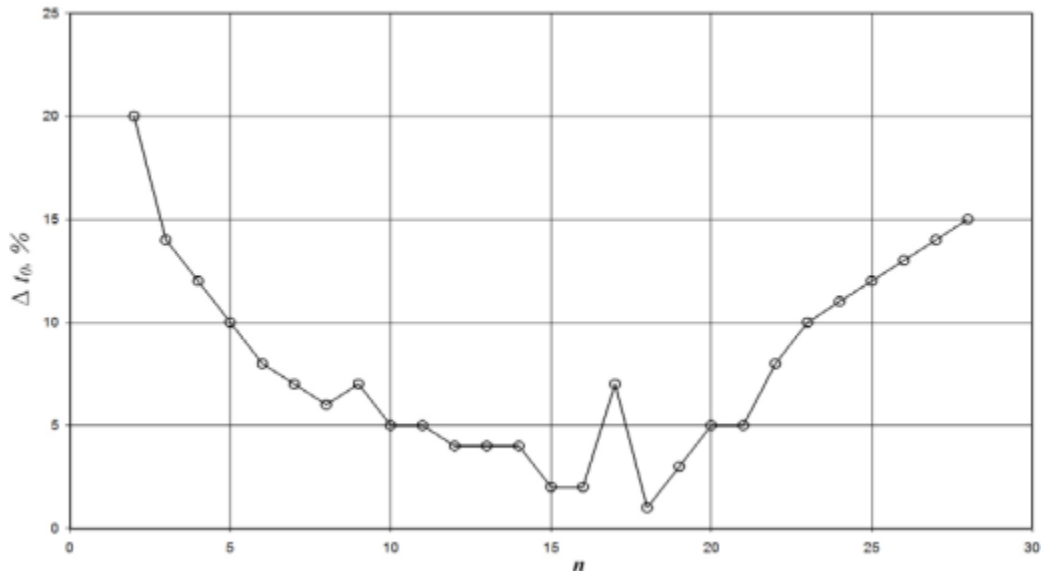


Рис. 2.2. Точність апроксимації початкового температурного профілю в залежності від кількості скінченних елементів матричним метод

З наведених даних видно, що найкраща точність апроксимації початкового профілю температури досягається з 18 кінцевими елементами в області розв'язку. Водночас апроксимація початкового профілю температури має відносну похибку 1,5%. Якщо ви бажаєте отримати рішення з високою точністю, вам слід використовувати поточний метод. Це дозволяє розв'язувати великорозмірні системи без значних похибок округлення. Цей метод має серйозні обмеження і використовується лише для систем лінійних рівнянь у смуговій алгебрі. При застосуванні методу скінченних елементів ширина. У деяких випадках вихідна проблема настільки серйозна, що навіть метод кінцевих елементів не може її вирішити. У цьому випадку потрібно змінити параметри завдання. У той же час існують системи алгебраїчних рівнянь, де невеликі зміни коефіцієнтів або вільних членів призводять до великих змін розв'язку. Таку

систему рівнянь називають необумовленою. Розглянемо застосування методу Хунді до представленої системи.

2.2.3. Метод Ейлера

Найбільш простим варіантом чисельного інтегрування системи диференціальних рівностей є метод Ейлера, який може бути представлений наступальною схемою розрахунку (рис. 2.4).

$$(-1) \cdot \begin{array}{c} \left| \begin{array}{ccccc} n+1 & -\binom{n+1}{2} & 0 & 0 & 0 \\ -\binom{n+1}{2} & n+1 & -\binom{n+1}{2} & 0 & 0 \\ 0 & -\binom{n+1}{2} & n+1 & -\binom{n+1}{2} & 0 \\ \hline 0 & 0 & -\binom{n+1}{2} & n+1 & -\binom{n+1}{2} \\ 0 & 0 & 0 & -\binom{n+1}{2} & n+1 \end{array} \right| \cdot \begin{array}{c} u_1 \\ u_2 \\ u_3 \\ \hline u_{n-1} \\ u_n \end{array} \end{array} = \frac{X_{\max} - X_{\min}}{n+1} \cdot \begin{array}{c} \left| \begin{array}{ccccc} \frac{2}{3} & \frac{1}{6} & 0 & 0 & 0 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & 0 \\ 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 \\ \hline 0 & 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ 0 & 0 & 0 & \frac{1}{6} & \frac{2}{3} \end{array} \right| \cdot \begin{array}{c} u'_1 \\ u'_2 \\ u'_3 \\ \hline u'_{n-1} \\ u'_n \end{array} \end{array}$$

Рис. 2.3. Варіант чисельного інтегрування системи диференціальних рівнянь

$$\begin{array}{c} \left| \begin{array}{c} u_1(\tau_k) \\ u_2(\tau_k) \\ u_3(\tau_k) \\ \hline u_{n-1}(\tau_k) \\ u_n(\tau_k) \end{array} \right| = \begin{array}{c} \left| \begin{array}{c} u_1(\tau_{k-1}) \\ u_2(\tau_{k-1}) \\ u_3(\tau_{k-1}) \\ \hline u_{n-1}(\tau_{k-1}) \\ u_n(\tau_{k-1}) \end{array} \right| \cdot \left\{ \begin{array}{c} \left| \begin{array}{ccccc} n+1 & -\binom{n+1}{2} & 0 & 0 & 0 \\ -\binom{n+1}{2} & n+1 & -\binom{n+1}{2} & 0 & 0 \\ 0 & -\binom{n+1}{2} & n+1 & -\binom{n+1}{2} & 0 \\ \hline 0 & 0 & -\binom{n+1}{2} & n+1 & -\binom{n+1}{2} \\ 0 & 0 & 0 & -\binom{n+1}{2} & n+1 \end{array} \right| \cdot \frac{X_{\max} - X_{\min}}{n+1} \cdot \left| \begin{array}{ccccc} \frac{2}{3} & \frac{1}{6} & 0 & 0 & 0 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & 0 \\ 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 \\ \hline 0 & 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ 0 & 0 & 0 & \frac{1}{6} & \frac{2}{3} \end{array} \right|^{-1} \cdot \left| \begin{array}{c} u_1(\tau_{k-1}) \\ u_2(\tau_{k-1}) \\ u_3(\tau_{k-1}) \\ \hline u_{n-1}(\tau_{k-1}) \\ u_n(\tau_{k-1}) \end{array} \right| \cdot \Delta\tau \end{array} \right. \end{array}$$

Рис. 2.4. Представлення метода Ейлера у вигляді розрахункової схеми

Таким чином, метод вказівки елементів дозволяє створити континуум частинами, аналогічній системі нерівномірно-переривчастих лінійних функцій з напірними коефіцієнтами, терміном придатності. За будь-якої системи варіації алгебра може створити лінійну структуру, що дозволяє створити широкий метод завдання великого розмаїття, зберігаючи при цьому достатню точність рішення при великій кількості елементів індукції, що охоплюють область рішення.

2.2.4. Рівняння теплопровідності

Рівняння теплопровідності [21] регулює погодинну реакцію та інші явища теплового потоку. Розв'язок рівняння теплопровідності складається з

незалежної від часу суми кінцевого розподілу температури та безперервної експоненціально спадної ортогональної функції, яка описує еволюцію розподілу температури від початкового стану $f(x)$ до кінцевого стану. Рівняння теплопровідності отримують записом балансу швидкості тепловіддачі контрольного об'єму [22]. Рівняння теплопровідності є загальним і його можна знайти в багатьох інших задачах, що описують дифузію маси [23]. Хоча теплообмін і температура тісно пов'язані, вони різні за своєю природою. На відміну від температури, теплопередача має напрямок і величину і тому є вектором. Рівняння теплопровідності в твердому тілі можна записати у вигляді

$$\frac{\partial T}{\partial t} = k \nabla^2 T,$$

де

T – значення температури;

k – коефіцієнт теплопровідності.

Диференціальні рівняння для теплопровідності виводяться із застосування закону теплопровідності Фур'є, і фундаментальні характеристики цих рівнянь залежать від форми та змінюються як функція системи координат, обраної для представлення твердого тіла [24]. Теплопровідність – це передача тепла за відсутності потоку маси, що регулюється законом Фур'є [25],

$$q = -kA \frac{\Delta T}{\Delta x},$$

де q – швидкість теплового потоку через площу A і товщину Δx , на якій зміна температури становить ΔT .

2.3. Опис використаних технологій та мов програмування

Однією з популярних мов є Java, яка є строго типізованою кросплатформною об'єктно-орієнтованою мовою програмування. Він був випущений компанією Sun Microsystems у 1995 році. Вихідний код Java

компілюється в байт-код, який виконується у віртуальній машині Java (JVM). Java — це добре продумана і повна реалізація об'єктно-орієнтованого програмування, яка має спільний C-подібний синтаксис з такими мовами, як C, C++, C# і JavaScript [4]. Java була створена на основі таких принципів [5]:

- повна підтримка об'єктно-орієнтованого програмування;
- кросплатформенність;
- вбудована підтримка мережевих протоколів;
- виконати захищений код з віддаленого пристрою;
- простота програмування.

Розробники Happydesk позиціонують її як інтуїтивно зрозумілу систему. Функції цієї системи: збирання, облік, обробка та аналіз запитів від клієнтів. Happydesk є гнучкою багатоканальною системою. Це означає, що запити від клієнтів можуть надходити через електронну пошту, програми обміну повідомленнями або спеціальний віджет на сайті.

Існують правила направлення заявок. Заявки розподіляються виконавцям, враховуючи завантаженість кожного виконавця та його досвід у виконанні подібних заявок [12].

Для виконавців Happydesk має такі можливості, як заготовлені відповіді, які виконавець може додавати та використовувати; перегляд повної історії заявок від клієнтів; складний фільтр, що налаштовується за заявками; керування пріоритетом повідомлень.

Клієнт може самостійно знайти вирішення своєї проблеми за допомогою зручного пошуку на базі знань. Завдяки цьому кількість листів до системи підтримки скорочується на 64% [2]. Крім цього, Happydesk надає можливість клієнтам оцінювати задоволеність роботою служби. Також можна переглядати статистику якості роботи окремих працівників чи відділів [13].

Happydesk інтегруємо у існуючі системи підтримки клієнтів. Ця система підтримує делегування запитів між співробітниками, розмежування прав адміністрування. Контроль ефективності роботи працівників здійснюється за допомогою графіків та зведених таблиць.

У безкоштовній версії Happydesk має обмежену кількість можливостей. Зокрема, безкоштовна версія може підтримувати трохи більше трьох співробітників; в ній неможливий автоматичний розподіл заявок, завдання правил та шаблонів для виконавців [13].

Freshdesk є багатоканальною системою. Він інтегрований із Twitter, Facebook. Дана система має спеціальний віджет для надсилання заявок.

Зібрані заявки надходять до спільної поштової скриньки. Це рішення дозволяє виконавцям обробляти заявки, не заважаючи один одному. Повідомлення, яке надходить на адресу електронної пошти служби, стає заявкою. Далі виконавцем на кожну заявку може бути пріоритет і категорія. Виконавець може делегувати заявку члену команди.

Завдяки базі знань та клієнтським форумам клієнт отримує можливість вирішувати проблеми, не надсилаючи заявок до системи. База знань допомагає клієнтам, а також скорочує обсяг заявок, що входять. Додатковою функцією форумів є збір відгуків щодо якості роботи системи. Зовнішній вигляд системи може змінюватись [4].

Freshdesk інтегруємо із CRM системами. Цей Help Desk має програми, сумісні з ОСiOSта Android, що дозволяє робити підтримку клієнтів мобільного.

Для відстеження продуктивності Freshdesk пропонує використовувати звіти. Це можуть бути базові звіти, зведені звіти та аналіз навантаження.

Відмінною рисою Freshdesk є гейміфікація. Вона служить мотивації виконавців. Кожному виконавцю нараховується певна кількість балів залежно від часу виконання заявки та оцінки якості виконаної роботи від клієнта. Нарховані бали можуть забиратися у разі недотримання термінів або надходження негативного відкликання від клієнта. Ці бали бачить керівник та може залежно від них регулювати розмір премії виконавця. Також є обов'язкові для всіх виконавців «квести». Наприклад, «квест» може бути таким: «закрити 10 заявок на тиждень».

Безкоштовна версія надає лише базу знань та підтримку на адресу електронної пошти [3].

Серверна частина програми реалізована за допомогою бібліотеки Spring Boot, клієнтська - Angular.

Spring Boot — бібліотека з відкритим кодом для створення мікросервісів, що базується на мові Java. Ця бібліотека використовується для написання повноцінних Spring-додатків. Spring Boot має такі цілі:

- уникнути складності конфігурації xml у Spring;
- спростити розробку готових до запуску Spring додатків;
- надати простий спосіб створення та налаштування додатків.

Вхідною точкою програми є клас з методом main, позначений інструкцією @SpringBootApplication. Основні пакети starter наступні:

- spring-boot-starter-actuator - використовується для моніторингу та управління додатком;
- spring-boot-starter-security - використовується для Spring Security, що відповідає за авторизацію та аутентифікацію користувачів та доступ до даних;
- spring-boot-starter-web – використовується для створення REST (Representational state transfer) архітектури;
- spring-boot-starter-test – використовується для написання тест-кейсів;
- spring-boot-starter-data-jpa — використовується для доступу до даних, наприклад, реляційних і нереляційних баз даних.

Spring Boot веб-додаток включає вбудований веб-сервер, наприклад Apache Tomcat, Jetty або JBoss Undertow [4].

Angular — бібліотека компанії Google, призначена в основному для розробки SPA (single page application) додатків.

Бібліотека адаптує та розширює HTML за допомогою директив, а також синхронізує модель та подання за рахунок різного виду прив'язок даних.

Плюси Angular:

1. Angular поділяє DOM-маніпуляції та логіку програми. Це допомагає зробити процес розробки та тестування простіше.
2. Angular добре працює з бібліотеками Karma та Jasmine, які призначені

для тестування.

3. Angular дотримується MVC (Model View Controller) шаблону проектування та заохочує слабкий зв'язок між уявленням, даними та логікою компонентів.

4. Використання типізованої мови Typescript запобігає множині помилок у процесі розробки.

5. Angular можна використовувати для створення мобільних додатків.

Селектор позначає ім'я, яким даний компонент буде додаватися в інші, шаблон — шлях до html-файлу, відповідний даної компоненті [5].

Будь-який Angular додаток має головний модуль, що називається AppModule. Модуль є класом, позначеним декоратором @NgModule, функція якого приймає об'єкт з полями declarations (сюди входять компоненти, директиви, канали), exports (класи, які будуть використовуватися в інших модулях), imports (модулі, класи яких необхідні для використання в компонентах поточного модуля), providers (сервіси), bootstrap (компонент, що викликається під час завантаження програми). Сервіси є класами, позначені декоратором @Service. Дані класи виконують завдання зберігання та отримання даних, представляють канал взаємодії між різними компонентами та інкапсулюють бізнес логіку.

SPA (single page application) — програма, яка є веб-сайтом або звичайною програмою, розміщеною на одній веб-сторінці.

Архітектура програми SPA побудована так, що під час переходу на нову сторінку оновлюється лише частина вмісту. Таким чином, немає необхідності повторно завантажувати одні й самі елементи [6-7].

Основні переваги SPA додатків:

1) Висока швидкість роботи програми.

Оскільки SPA оновлює не всю сторінку, а лише необхідні частини, це значно прискорює роботу.

2) Висока швидкість розробки.

Готові бібліотеки надають потужні інструменти для розробки веб-додатків. Back-end та front-end розробники можуть працювати над проектами паралельно. І вони не будуть заважати один одному завдяки чіткому поділу.

3) Мобільні додатки.

На основі готового коду ви легко зможете розробити свій мобільний додаток.

Недоліком SPA додатків є погана пошукова оптимізація.

SPA працює на основі JavaScript та завантажує інформацію на запит з боку клієнта. Пошукові системи важко імітувати цю поведінку, тому більшість сторінок недоступні для сканування пошуковими системами.

2.4. Опис структури ситеми та алгоритмів її функціонування

Як СУБД використовується PostgreSQL.

PostgreSQL – об'єктно-реляційна СУБД з відкритим вихідним кодом.

Як об'єктно-реляційна СУБД PostgreSQL підтримує об'єкти користувача та їх поведінки, в тому числі операції, індекси, функції, типи даних. Дана СУБД може працювати з числовим, текстовим, булевим типом, а також з грошовим, що перераховується, геометричним, xml, JSON типом. Для гарантії цілісності даних PostgreSQL надає такі механізми як первинні та зовнішні ключі, унікальні та перевірені обмеження, обмеження NOT NULL. Ця СУБД відповідає стандарту ANSI_SQL: 2008, відповідає вимогам ACID (Atomicity, Consistency, Isolation, Durability), тобто атомарності, узгодженості, ізольованості та надійності).

Spring Boot додаток створюється за допомогою Spring Initializer, який дозволяє вибрати компоненти, що використовуються в додатку, а потім зібрати проект.

Класи Spring Boot програми були розміщені за наступними пакетами:

- 1) Config - для класів конфігурацій.
- 2) Controllers - для класів-контролерів, що опрацьовують запити.
- 3) Exceptions — для винятків користувача.

- 4) Jwt - для класів, пов'язаних з автентифікацією.
- 5) Mappers - для класів, які служать для перетворення об'єктів.
- 6) Models - для jpa сутностей та репозиторіїв.
- 7) Services - для сервісів [9].

Для роботи з Angular встановлюється сервер NodeJS та пакетний менеджер npm. Після цього встановлюється бібліотека Angular за допомогою команди `npm install-g @angular/cli`.

Для створення проекту з командного рядка використовується команда `ng new ім'я проекту`.

Структура Angular програми має такий вигляд:

- 1) Components – містить прості компоненти.
- 2) Containers - містить складні компоненти, які є окремою сторінкою в додатку.
- 3) Services - містить класи сервісів.
- 4) Store містить класи, які пов'язані зі сховищем станів NgRx.
- 5) Guards-містить клас для обмеження доступу маршрутам користувачів, які не мають на це прав.
- 6) Interceptors — класи, які перехоплюють та обробляють вхідні та вихідні запити.

2.5. Обґрунтування та організація вхідних та вихідних даних програми

Support Library [5] – бібліотека, яка робить функції новіших версій доступними на старих версіях Android. Наприклад, фрагменти з'являються лише в третій версії (рівень API 11):

- Додайте перевірку версії системи в код і виконайте ту чи іншу залежно від результату. Тобто, якщо версія 11 і вище, ми використовуємо Fragment, інакше використовуємо Activity. Запускаючи програму на старішій версії, вам доведеться або відмовитися від інновації та використовувати те, що доступно, або заново винайти велосипед і впровадити інновацію самостійно.

– Ви можете ігнорувати стару версію та націлювати розроблену програму лише на нову версію. Тоді великий відсоток потенційних користувачів вашої програми буде втрачено. На даний час на версії Android сидить 80% користувачів [5]. Але на той час нова версія системи Android випустить додатки з новими функціями, і, відповідно, деякі користувачі знову перевірятимуться. Загалом, подальша дискримінація користувачів за версіями неминуча.

– Використовуйте Support Library бібліотеку. Містить класи - аналоги нововведень останньої версії, для старих версій.

На даний момент існує дві бібліотеки v4 і v13 [8]. Цифри тут вказують на найнижчий рівень API, на якому можна використовувати цю бібліотеку. Тобто програми, які використовують v4, можуть запускатися на рівнях API ≥ 4 і мати доступ до інновацій, що містяться в цій бібліотеці (наприклад, фрагменти). Ці бібліотеки регулярно оновлюються, до них додаються нові класи, відкриваючи нові можливості.

Отже, якщо правильного рішення немає, воно, швидше за все, з'явиться в майбутньому. Отже, завдяки цій бібліотеці той самий код працює як на старій, так і на новій версії та використовує можливості нової версії.

Room [4] — це новий спосіб зберігання даних програми в програмах Android. Це частина нової архітектури Android, набору бібліотек від Google, які підтримують пов'язані архітектури програм. Кімната надається як альтернатива Realm, ORMLite, GreenDao та багатьом іншим. Це високорівневий інтерфейс для низькорівневих прив'язок SQLite, вбудованих в Android, ви можете прочитати більше про нього в документації. Він виконує більшу частину роботи під час компіляції, створюючи API поверх вбудованого API SQLite, тому вам не доведеться мати справу з Cursor або ContentResolver. Кімната складається з трьох основних компонентів: Entity, Dao і Database [4]. Для позначення об'єктів, які потрібно зберігати в базі даних, необхідно використовувати анотації сутностей. Об'єкти Dao описують методи роботи з основними даними.

Основний клас, який використовує базу даних, буде позначено анотацією бази даних.

Цей клас має бути абстрактним і наслідувати RoomDatabase.

Яка версія Entity та бази даних буде використовуватися в параметрах анотації Database[2]. Для кожного класу сутностей у списку сутностей буде створено таблицю. У класі Database необхідно описати абстрактний метод отримання об'єктів Dao, який буде використовуватися при розробці програми. Іноді об'єкти Entity можуть містити непримітивні поля, які неможливо зберегти в базі даних. Ця категорія також включає списки або літери (List, Array тощо).

2.6. Опис розробленої системи

Найбільш глобальним рішенням є реєстрація конвертера для бази даних. Таким чином, Room можна використовувати в усіх Entity і Dao. Якщо у вас кілька конвертерів, розділяйте їх комами. JSON [3] — текстовий формат для обміну даними між комп'ютерами. JSON є текстовим і читабельним. Цей формат дозволяє описувати об'єкти та інші структури даних. Цей формат в основному використовується для передачі структурованої інформації через мережу (через процес, який називається серіалізацією). Його розробив та популяризував Д.Кроксфорд[3].

JSON знайшов своє основне призначення при написанні веб-програм, а саме при використанні технології AJAX [3]. JSON використовується в AJAX як альтернатива XML (використовується в AJAX) під час асинхронної передачі структурованої інформації між клієнтом і сервером. У той же час перевага JSON над XML полягає в тому, що він допускає складні структури в атрибутах, має невеликий розмір і безпосередньо інтерпретується як об'єкти за допомогою JavaScript. Співпрограми [3] спрощують асинхронне програмування, приховуючи всю складність усередині бібліотеки. Ідея виникла в мові програмування Simula в 1967 році [5]. Спочатку було лише кілька методів: від'єднати та відновити (таким чином, вони дозволяють зупиняти та починати виконання). Однак від цієї ідеї відмовилися, коли з'явилася трансляція. З

одного боку, є сенс запускати асинхронні завдання в окремих потоках. Хоча іноді таке рішення не завжди є ефективним. Наприклад [13]:

- обидва потоки і перемикання між ними є дорогими з точки зору витрат;
- значно навантажити систему може створення тисячі потоків;
- програмний код повинен використовувати лише один потік (наприклад, в JavaScript);
- якщо у вас є багато змінних стану ("mutable state"), то немає необхідності використовувати потоки;
- легко допустити помилку при зміні інтерфейсу, доступного лише з його потоку. На практиці корутини - це обчислення, які можна призупинити без блокування потоку.

Існують дві функції для запуску [14]:

- `launch {}`;
- `async {}`.

Різниця полягає в тому, що `launch {}` не повертає нічого, тоді як `async {}` повертає екземпляр. `Deferred` містить функцію `await()`, яка повертає результат, схожий на `Future` у Java. У Java, для отримання результату, ми використовуємо `future.get()`. Крім того, корутини не можна викликати в головному потоці подій, оскільки це призведе до помилки:

- Функцію переривання можна викликати лише з іншої функції переривання.
- Функцію затримки можна викликати лише з корутини або іншої функції переривання.

2.6.1. Використані технічні засоби

Архітектура програмного забезпечення (ПЗ) – це спосіб організації програмної або обчислювальної системи, у якому її елементи абстрагуються на окремих етапах роботи. Система може мати кілька рівнів абстракції і багато етапів роботи, кожен з яких може мати власну архітектуру [3].

Проектування архітектури ПЗ - це процес розробки, який відбувається після аналізу та формулювання вимог. Метою такого проектування є перетворення вимог системи в вимоги до ПЗ та побудова архітектури системи на їхній основі.

Model-View-Presenter (MVP) [3] - це шаблон проектування, що базується на MVC, і використовується переважно для створення користувацького інтерфейсу.

У даному шаблоні елемент Presenter виступає в ролі посередника, подібно до контролера у шаблоні MVC. Він відповідає за керування подіями, пов'язаними з інтерфейсом користувача (наприклад, використання миші) так само, як у інших шаблонах відповідає уявлення.

Якщо користувач в Activity натисне кнопку "Оновити", то Activity повідомляє про це презентер. При цьому Activity не зобов'язує презентер завантажувати дані - воно просто повідомляє про натискання кнопки "Оновити". Презентер вже самостійно вирішує, що робити після натискання цієї кнопки. Він запитує дані у моделі і передає їх у Activity для відображення на екрані. Якщо екран відображає дані з бази даних, то модель виступає в ролі самої бази даних. Презентер може підписатися на повідомлення від моделі про оновлення. Коли дані в базі даних змінюються, модель сповіщає презентер про це. Презентер отримує зміни і передає їх у Activity.

Можна сказати, що презентер - це логіка, винесена з Activity до окремого класу. А Activity залишається для відображення даних і взаємодії з користувачем. Якщо створити інше Activity для відображення даних, то не потрібно буде переносити логіку до нового Activity, оскільки можна використовувати наявний презентер. У разі зміни логіки необхідно буде тільки змінити код презентера, а не дублювати його у Activity.

Шаблон MVVM (Model-View-ViewModel) дозволяє відокремити логіку вашої програми від візуальної частини (презентації). Цей патерн є архітектурним, тобто визначає загальну архітектуру програми [4]. Цей шаблон був представлений Джоном Госсманом у 2005 році як модифікація шаблону

Presentation Model, спочатку призначеного для розробки програм у WPF. Хоча цей шаблон зараз перевершив WPF і використовується в різних технологіях, включаючи розробку Android та iOS.

Розглянемо цю архітектуру докладніше:

- View —цей рівень містить компоненти інтерфейсу користувача, які відповідають за керування кодом компонентів інтерфейсу користувача (видів), таких як ініціалізація підвидів, відображення індикаторів виконання, обробка даних, введених користувачами, і обробка анімації. Наприклад, такий код міститься в активуючих і фрагментах;
- ViewModel – цей тип об'єкта надає дані для компонентів інтерфейсу користувача. У нашому випадку View буде використовувати LiveData для відстеження змін даних у ViewModel;
- Model – модель описує дані, які використовуються в програмі. Моделі можуть містити логіку, безпосередньо пов'язану з цими даними, наприклад, логіку перевірки властивостей моделі. При цьому модель не повинна містити жодної логіки, пов'язаної з відображенням даних і взаємодією з елементами візуального керування.

2.6.2. Використані програмні засоби

Сьогодні цифрові технології все частіше використовуються людьми з усіх верств суспільства. Багато людей використовують доступні бездротові технології, щоб залишатися на зв'язку з іншими, де б вони не були. Зростаюча популярність смартфонів привернула увагу майже кожного. Метод у контексті засобів розробки програмного забезпечення – це набір технік та правил.

Під засобом розробки програмного забезпечення розуміється набір методів та методик, прийомів та інструментальних програм для створення програмного коду.

На різних етапах розробки програмного забезпечення використовуються різні засоби та методи. Для того, щоб систематизувати існуючі засоби розробки

програмного забезпечення, виділимо основні три стадії розробки: проектування, реалізація, тестування [1].

Стадію проектування можна розбити на такі стадії:

- аналіз вимог;
- розробка архітектури програми;
- розробка макетів інтерфейсу користувача.

На стадії проектування для опису моделей функціонування програми застосовуються блок-схеми, UML (Unified Modeling) діаграми, DFD (Data Flow Diagrams) діаграми та IDEF0 діаграми. Також можна використовувати BPMN (Business Process Model and Notation) діаграми [1-2].

IDEF0 діаграми описують систему, починаючи від виявленої точки зору до конкретної мети. Ці діаграми розташовуються різних рівнях. На верхньому рівні є найбільш загальна діаграма, а кожен наступний рівень може включати до 6 діаграм, які деталізують діаграму на верхньому рівні. Діаграма складається з блоків та дуг. Блоки діаграми пронумеровані від 1 до 6. Верхня діаграма має назву A0. Кожен процес на діаграмі A0 може бути розкладений на підпроцеси та змодельований на діаграмі нижнього рівня, яка матиме назву A1-A6 залежно від номера обраного блоку. Кожен процес із діаграм A1-A6 може бути розкладений на підпроцес і змодельований на діаграмах A16-A66, де перша цифра відповідає за номер вище стоящої діаграми, друга на номер процесу. Блоки є дією, а дуги визначають взаємозв'язок блоків. Ліва сторона блоку призначена для вхідних дуг, права - для вихідних, верхня - для дуг управління, нижня - для механізмів [3].

Другий етап стадії проектування здійснюють за допомогою діаграм класів та пакетів.

Для створення UML діаграм можна використовувати Enterprise Architect — програмне забезпечення від австралійської компанії Sparx Systems, розроблене з метою управління бізнес-проектами та побудови бізнес-планів.

Розробка макетів інтерфейсу користувача допомагає зрозуміти, як будуть виглядати вікна, сторінки в програмі, що розробляється. Ця стадія здійснюється із застосуванням таких засобів як Figma або Axure. Figma - крос-платформний сервіс для прототипування та розробки інтерфейсів. Figma дозволяє одночасно працювати над одним і тим самим проектом кільком людям [4].

На вибір засобів реалізації програмного продукту впливають підходи, обрані на етапі проектування. Проте, можна назвати основні види коштів. По-перше, це мови програмування. Розглянемо мови для створення веб-застосунків.

1. Java - сильно типізована об'єктно-орієнтована мова програмування. Ця мова була розроблена компанією Sun Microsystems, яку потім придбала компанія Oracle. Програми java є кросплатформовими, тому що вони транслюються в спеціальний байт-код і можуть виконуватися на будь-якій архітектурі за допомогою віртуальної java-машини. На даний момент Java одна з найпопулярніших мов програмування. Завдяки безлічі доповнень і бібліотек, що підключаються, він ідеально підходить для вирішення поставленого завдання.
2. PHP — мова, за допомогою якої написано більшість сайтів та веб-сервісів. Приклад сайтів написаних на PHP: facebook.com. Ця мова приваблює багатьох фахівців своєю простотою [4-5] .
3. Python — інтерпретована мова програмування. Простий у використанні та гнучкий. Містить невелику кількість ключових слів. Інтерпретатор Python реалізований на багатьох платформах та операційних системах.
4. NodeJS - програмна платформа, яка заснована на програмному засобі (движку) V8він служить для трансляції JavaScript в машинний код. NodeJS дозволяє JavaScript взаємодіяти з пристроями вводу-виводу, підключати зовнішні бібліотеки, які можуть бути написані різними мовами. NodeJS в основному використовується на сервері та виконує роль

веб-сервера. Однак він дозволяє розробляти і десктопні віконні програми та програмувати мікроконтролери. NodeJS базується на орієнтованому та асинхронному (або реактивному) програмуванні з неблокуючим введенням/виведенням [6-7].

Другий вид засобів реалізації – це системи управління базами даних. Вибір системи управління базою даних є важливою частиною будь-якого проекту. Від цього вибору істотно залежить швидкість роботи програми. Відповідно це позначиться на рівні задоволеності користувача. Розглянемо найпоширеніші бази даних:

1. MySQL є популярною повноцінною системою управління базами даних. Вона відмінно підходить для роботи з різними програмами. Для цієї СУБД існує велика кількість плагінів, які дозволяють полегшити роботу із системою. Також MySQL добре документована. MySQL забезпечує безпечний доступ до даних. Недоліками цієї СУБД є повільна розробка проекту та слабка стійкість до відмови [8-9].
2. PostgreSQL – реляційна СУБД з відкритим вихідним кодом. Вона є професійнішою СУБД, ніж MySQL. У PostgreSQL велика продуктивність, і дана СУБД легко розширюється за допомогою процедур, що зберігаються. Як і MySQL, вона надає безпечний доступ до даних.
3. MongoDB є базою даних NoSQL. Вона не структурована та має хорошу пропускну здатність. У MongoDB відсутні класичні транзакції, які є у MySQL та PostgreSQL.

Третій засіб – це системи управління версіями: Git, SVN (Subversion), Mercurial.

На стадії тестування відбувається випробування програмного продукту на відповідність вимогам, виявляються помилки, покращується якість програмного продукту. Серед засобів тестування програмного забезпечення виділяють кошти для аналізу коду, тестування функціональності та продуктивності [9].

Останнім часом особливою популярністю користуються веб-програми. Веб-додаток є клієнт-серверним додатком. Клієнт взаємодіє із сервером за

допомогою браузера. Сервер приймає запити HTTP від клієнтів і видає їм HTTP відповіді. Логіка розподіляється між сервером та клієнтом. Зберігання інформації в базі даних.

Компонент LiveData [4] - призначений для зберігання об'єктів і дозволяє підписуватися на їх зміни. Ключовою особливістю є те, що компонент знає про життєвий цикл, тож вам не потрібно турбуватися про поточний етап передплатника, і якщо передплатника буде знищено, компонент відпишеться сам. Щоб дозволити LiveData враховувати життєвий цикл, використовується компонент Lifecycle, але його також можна використовувати без прив'язки життєвого циклу.

Клас LiveData — це абстрактний загальний клас, який інкапсулює всю логіку компонента. Відповідно, щоб створити наш власник LiveData, нам потрібно імітувати цей клас, як типовий приклад, вказати тип, який ми маємо намір зберігати в ньому, і описати логіку оновлення збереженого об'єкта.

В ідеалі ViewModel не повинна нічого знати про Android. Це покращує тестування та модульність, а також зменшує кількість витоків пам'яті. Основне правило полягає в тому, що ViewModels не повинні мати імпорт Android (крім android.arch.*).

Демонстрація модуля MVVM з LiveData [7]

2.6.3. Виклик та завантаження програми

Щоб провести розрахунок для тіл обергання оберемо середовище Matlab разом з пакетом pdetool, що входить до неї. Це дозволить вирішувати задачі для диференціальних рівнянь з частинними похідними.

Пакет PDE Toolbox можна викликати за допомогою командного вікна Matlab. Для цього у ньому слід записати команду pdetool (рис.2.5).

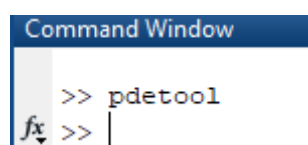


Рис. 2.5. Виклик PDE Toolbox у середі Matlab

Найближчим часом для моделювання даних відкриється окреме вікно під назвою PDE Modeler (рис. 2.6), де можна розв'язувати диференціальні рівняння з частинними похідними, зокрема розраховувати температурне поле в рухомих об'єктах.

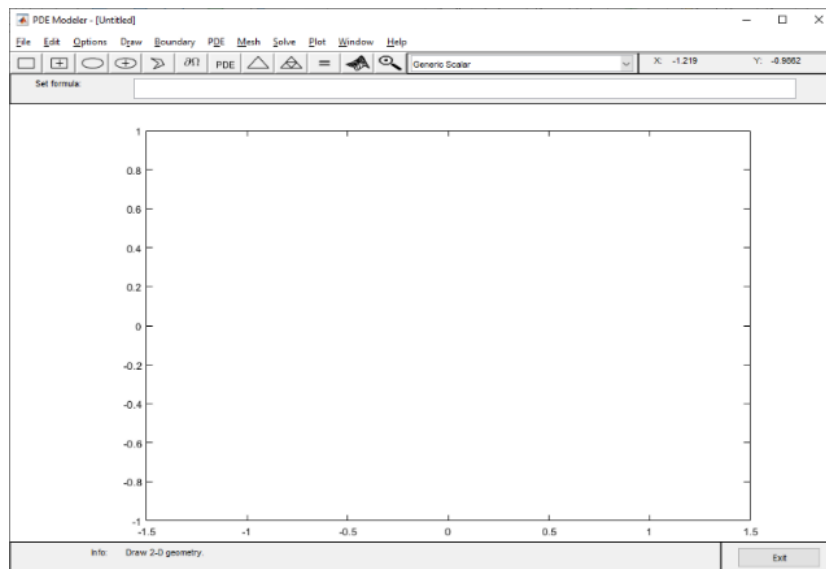


Рис. 2.6. Вікно PDE Mode

Переходимо у режим Draw Mode (рис. 2.7) та розраховуємо температурну область . Натискаємо меню малювання, і знаходимо в меню потрібний режим.

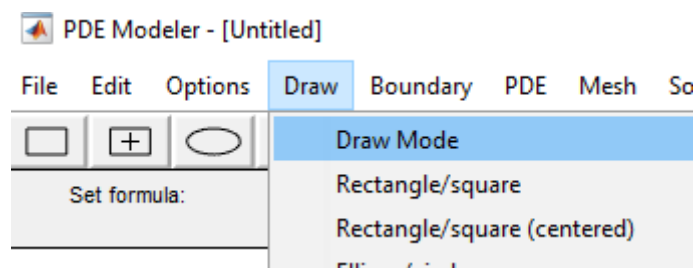


Рис. 2.7. Меню Draw

Після цього натиснемо на фігуру яка нам потрібна , щоб ми мали змогу намалювати її на полотні PDE Modeler. Потім створюємо фігуру яка нам потрібна за допомогою миші у будь-якому місці.

2.6.4. Опис інтерфейсу користувача

На початку для створення оберемо фігуру еліпсу. За допомогою попереднього пункту створюємо фігуру (рис.2.8). Налаштуємо еліпс так, щоб його центр був у центрі координат, перша піввісь дорівнювала 0.8, а друга – 0.5.

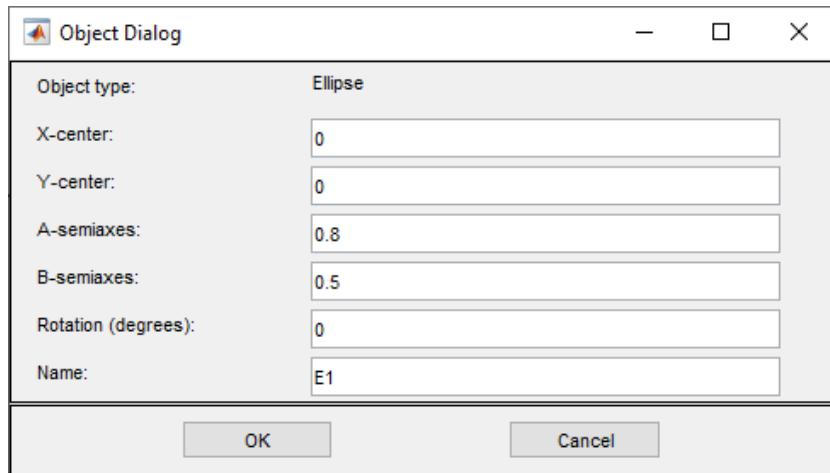


Рис. 2.8. Параметри розташування і розмірів еліпсу

В результаті на полотні з'явиться фігура еліпсу (рис. 2.9) з заданими параметрами.

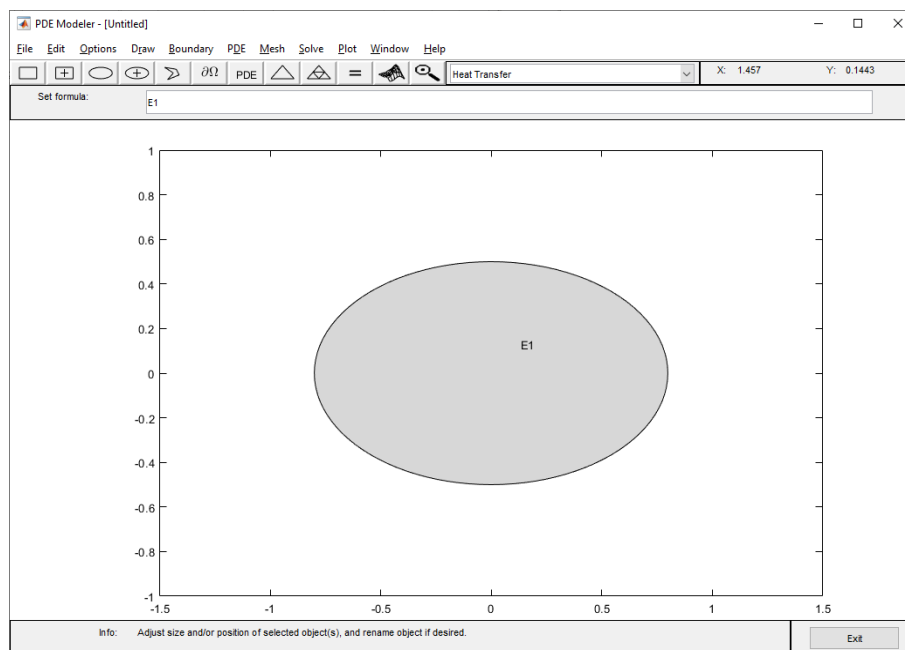


Рис. 2.9. Еліпс

Задаємо наступні граничні умови для меж еліпсу (рис. 2.10). Для верхньої лівої межі налаштуємо температуру рівною 1 (рис. 2.11). Для правої верхньої – 0.7 (рис. 2.12). Для нижніх – 0 (рис.2.13).

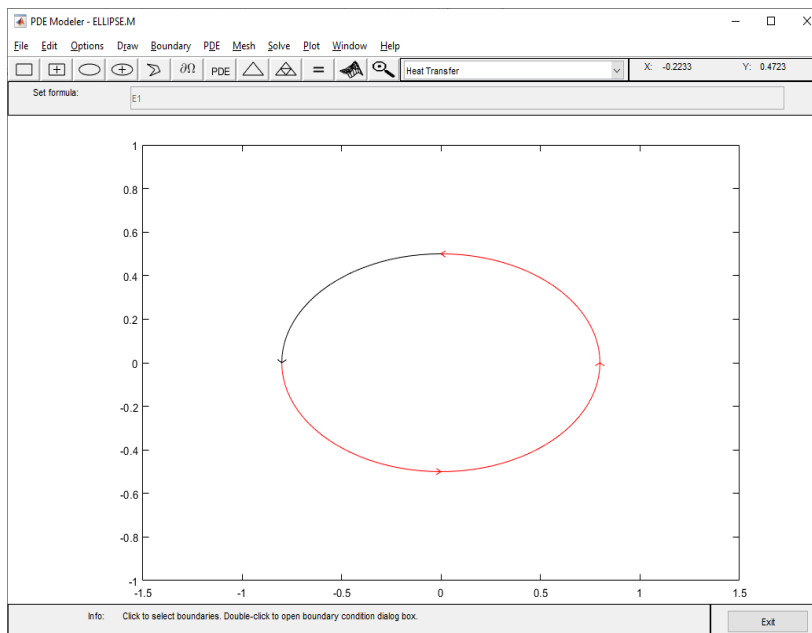


Рис. 2.10. Обрані межі еліпсу

Condition type:	Coefficient	Value	Description
<input type="radio"/> Neumann	g	0	Heat flux
<input checked="" type="radio"/> Dirichlet	q	0	Heat transfer coefficient
	h	1	Weight
	r	1	Temperature

Рис. 2.11. Параметри виразу для лівої верхньої межі еліпсу

Boundary Condition

Boundary condition equation: $h \cdot T = r$

Condition type:	Coefficient	Value	Description
<input type="radio"/> Neumann	g	0	Heat flux
<input checked="" type="radio"/> Dirichlet	q	0	Heat transfer coefficient
	h	1	Weight
	r	0.7	Temperature

OK Cancel

Рис. 2.12. Параметри виразу для правої верхньої межі еліпсу

Boundary Condition

Boundary condition equation: $h \cdot T = r$

Condition type:	Coefficient	Value	Description
<input type="radio"/> Neumann	g	0	Heat flux
<input checked="" type="radio"/> Dirichlet	q	0	Heat transfer coefficient
	h	1	Weight
	r	0	Temperature

OK Cancel

Рис. 2.13. Параметри виразу для нижніх меж еліпсу

Зробимо триангуляцію області та підвищимо її щільність (рис. 2.14).

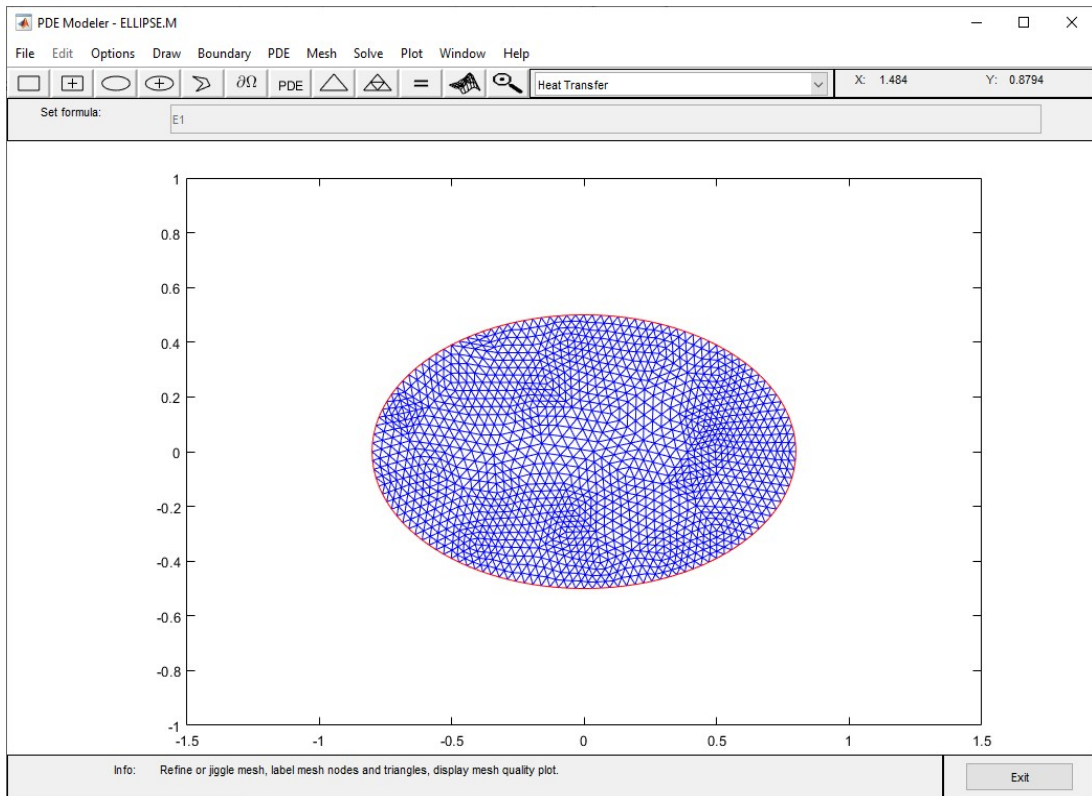


Рис. 2.14. Триангуляція еліпсу

Останнім кроком розрахуємо температурні поля та отримаємо результат для еліпса (рис. 2.15).

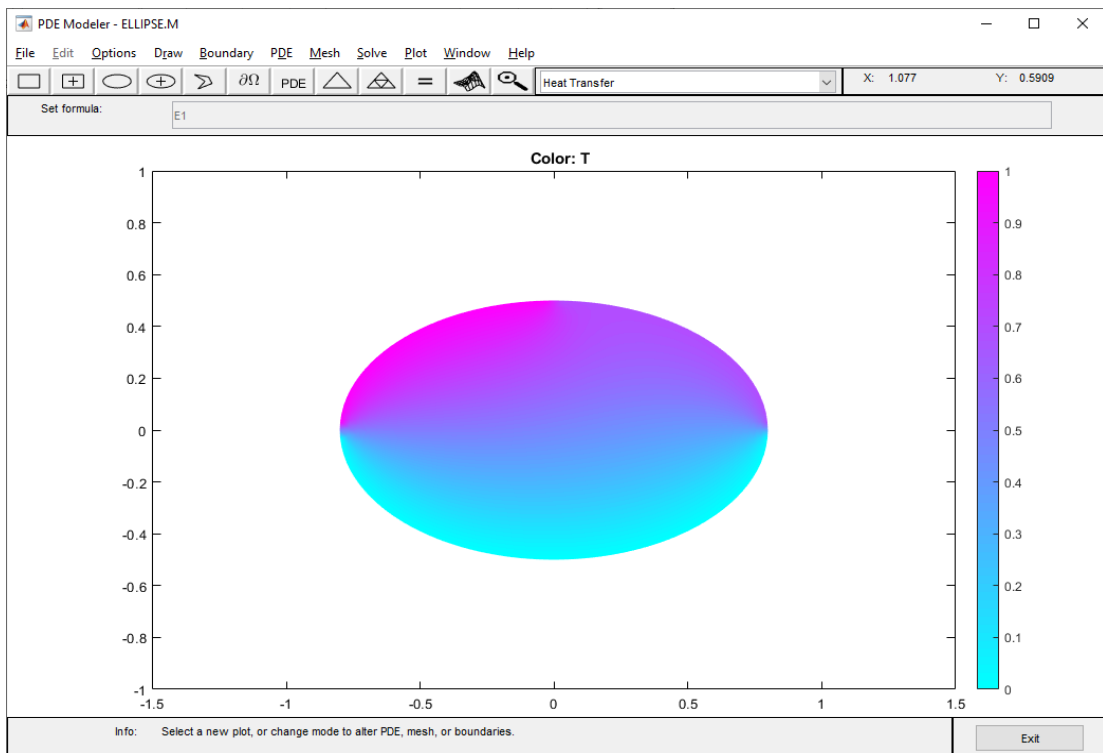


Рис. 2.15. Розподіл температурних полів в еліпсі

Тепер зробимо розрахунок для однопорожнинного гіперболоїда. Створимо прямокутник, натиснувши на кнопку Rectangle в меню Draw. Намалюємо його таким самим чином, яким були створені фігури раніше. Двічі натиснувши на прямокутник задамо йому наступні параметри: низ прямокутника – -0.5, лівий край – -0.8, ширина – 1.6, висота – 1 (рис. 2.16).

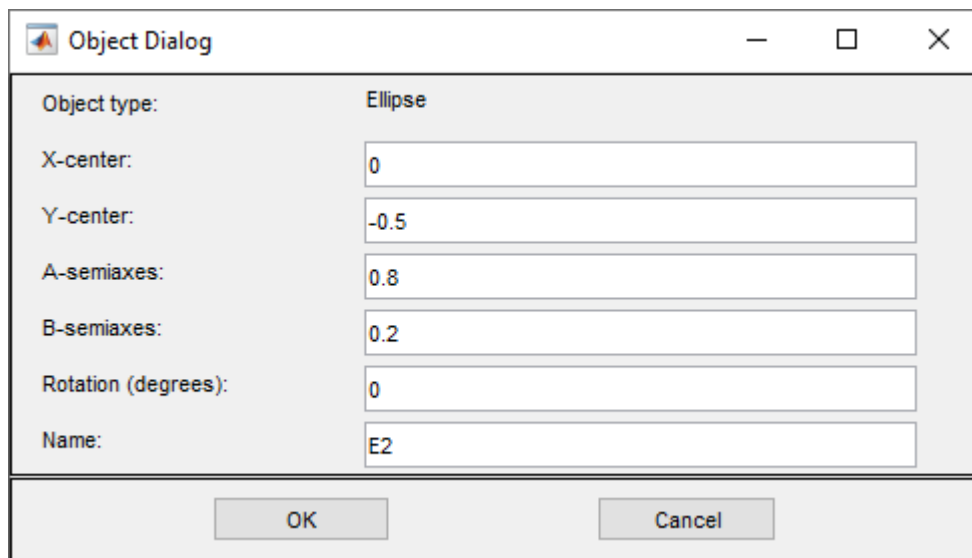


Рис. 2.16. Параметри розмірів і розташування прямокутника

Також створимо два еліпса, які будуть симетрично знаходитись щодо осі абсцис. Параметри для верхнього еліпса можна побачити на рис. 2.17, а для нижнього – на рис. 2.18 .

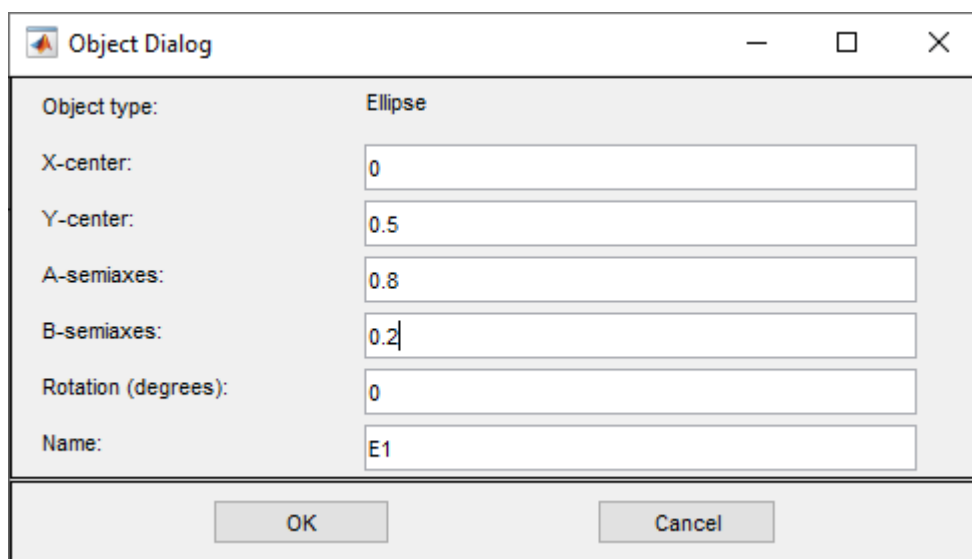


Рис. 2.17. Параметри розмірів і розташування верхнього еліпсу



Рис. 2.18. Параметри розмірів і розташування нижнього еліпсу

В результаті отримаємо наступні фігури (рис. 2.19).

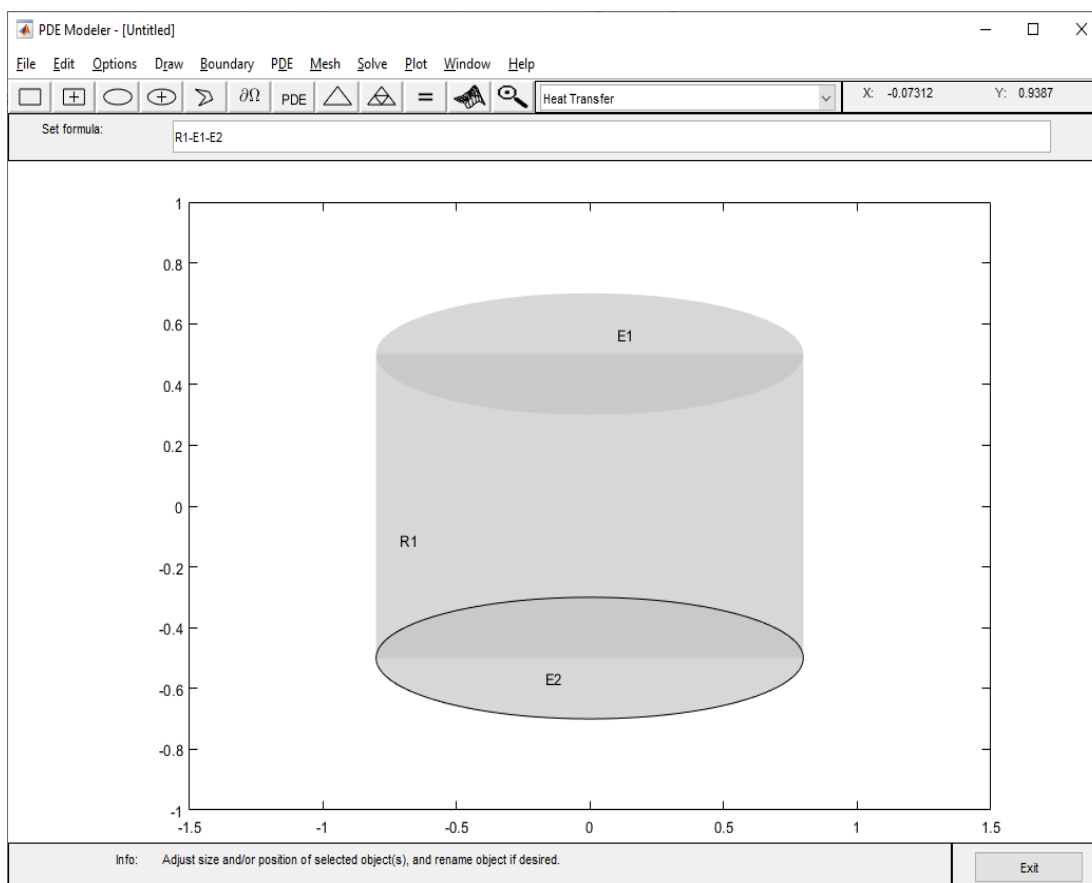


Рис. 2.19. Фігури для моделювання однопорожнинного гіперboloїда

Щоб отримати однопорожнинний гіперboloїд потрібно від прямокутника відняти створені еліпси. Для цього в полі Set formula задамо формулу R1-E1-E2 (рис. 2.20).

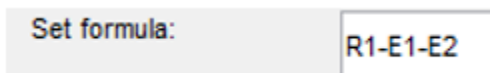


Рис.2.20.Формула моделювання однопорожнинного гіперboloїда

Змінимо режим на Boundary Mode. Тут можна побачити, що межі, які ми отримали (рис. 2.21) візуально вже більш нагадують однопорожнинний гіперboloїд.

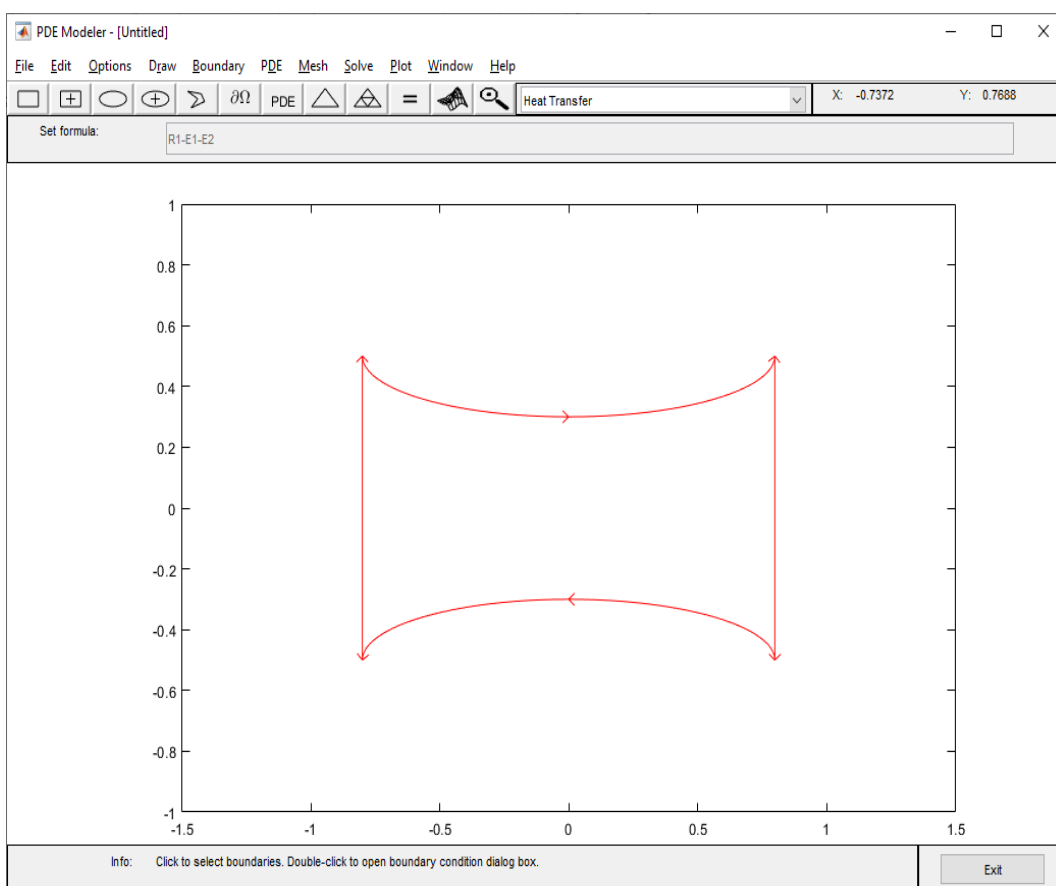


Рис. 2.21. Межі однопорожнинного гіперboloїда

Задамо наступні температури для меж: температура для верхніх меж дорівнює 1 (рис. 2.22), для бокових – 0.6 (рис. 2.23), для нижніх – 0 (рис. 2.24).

Boundary Condition dialog box showing parameters for the top boundary. The boundary condition equation is $h \cdot T = r$. The condition type is Dirichlet. The coefficients and their values are:

Coefficient	Value	Description
g	0	Heat flux
q	0	Heat transfer coefficient
h	1	Weight
r	1	Temperature

Рис. 2.22. Параметри виразу для верхніх меж

Boundary Condition dialog box showing parameters for the side boundary. The boundary condition equation is $h \cdot T = r$. The condition type is Dirichlet. The coefficients and their values are:

Coefficient	Value	Description
g	0	Heat flux
q	0	Heat transfer coefficient
h	1	Weight
r	0.6	Temperature

Рис. 2.23. Параметри виразу для бокових меж

Boundary Condition dialog box showing parameters for the bottom boundary. The boundary condition equation is $h \cdot T = r$. The condition type is Dirichlet. The coefficients and their values are:

Coefficient	Value	Description
g	0	Heat flux
q	0	Heat transfer coefficient
h	1	Weight
r	0	Temperature

Рис. 2.24. Параметри виразу для нижніх меж

Проведемо триангуляцію та підвищимо щільність сітки. Розрахуємо результат (рис. 2.25).

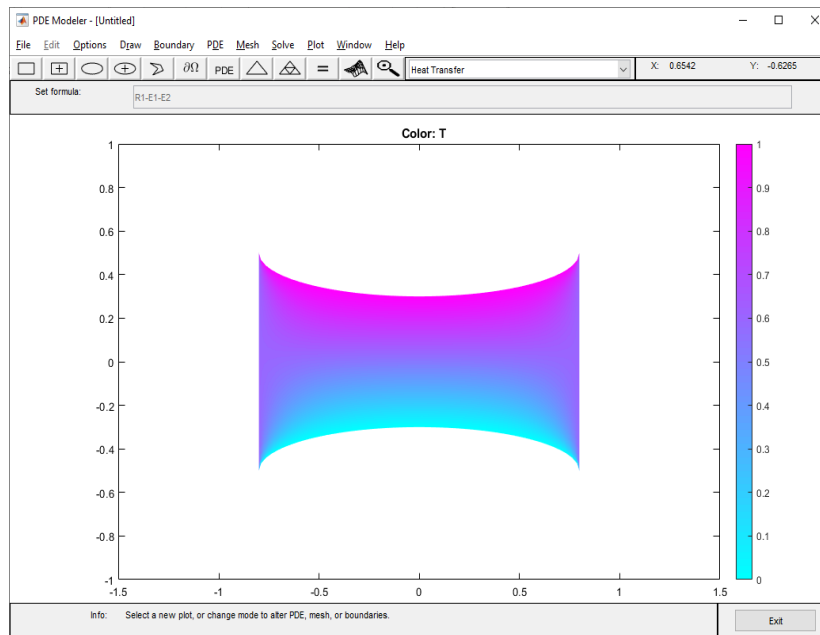


Рис. 2.25. Розподіл температури однопорожнинного гіперboloїда

Проведемо розрахунок температурних полів для конічної поверхні. Для цього створимо два полігона обравши Polygon у меню Draw.

Намалюємо три лінії так, щоб кінець останньої знаходився приблизно на початку першої лінії. Після цього двічі натиснувши на отриману фігуру відкриємо вікно Object Dialog (рис. 2.26). Навпроти поля Coordinates є випадальний список з кількістю елементів яка дорівнює кількості вершин створеного полігону. Обираючи кожну з вершин ми можемо налаштувати точніше їх координати (рис. 2.27 і рис. 2.28).

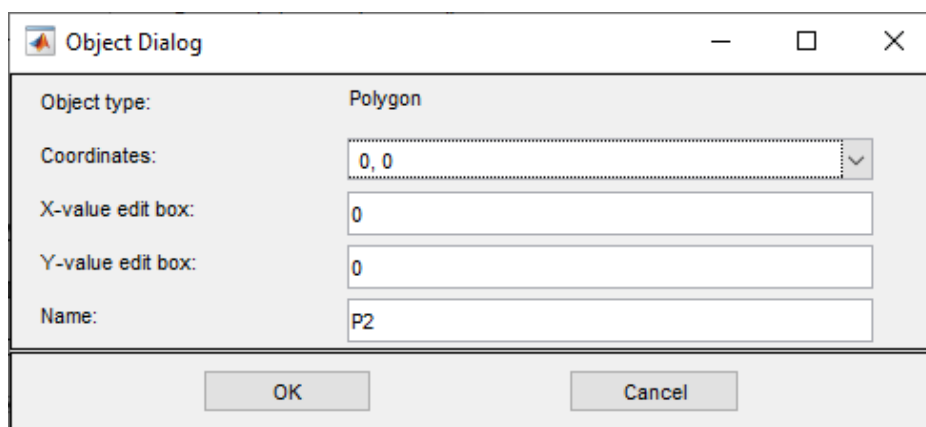


Рис. 2.26. Параметри розташування першої координати першого полігону

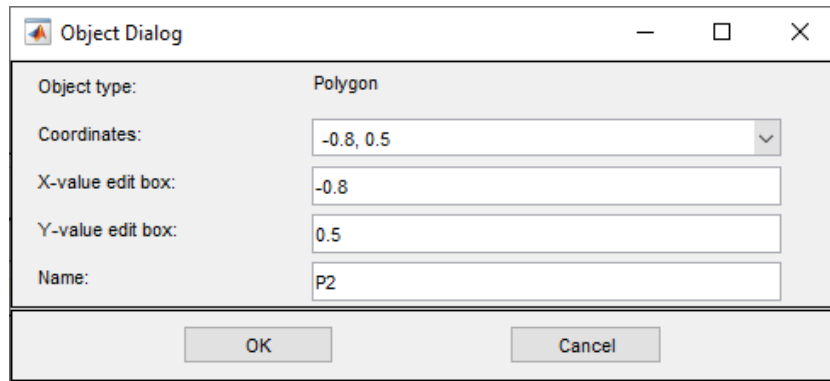


Рис. 2.27. Параметри розташування другої координати першого полігону

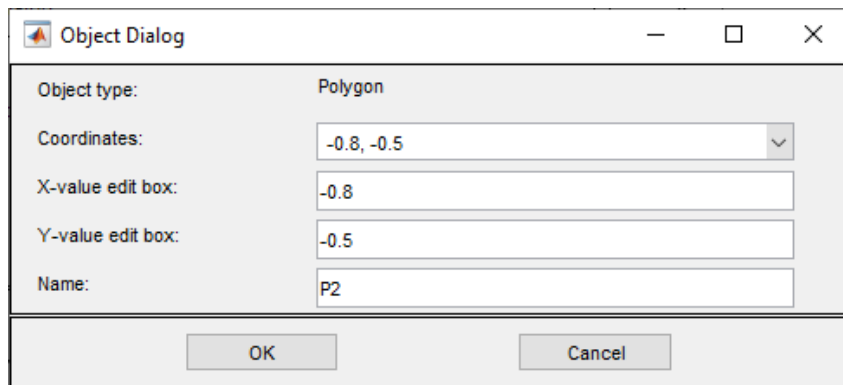


Рис. 2.28. Параметри розташування третьої координати першого полігону

Створимо другий полігон таким самим чином, для його другої і третьої вершини змінимо x-координату на 0.8. Все інше налаштуємо так само, як і для першого полігону. Отримаємо кінчну поверхню (рис. 2.29).

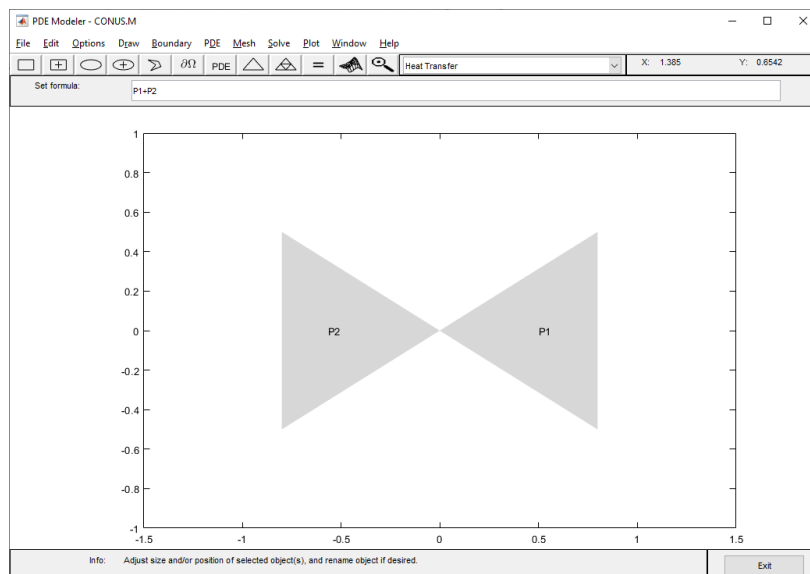


Рис. 2.29. Фігури для кінчної поверхні

У режимі Boundary Mode налаштуємо температуру для двох верхніх меж, які показані чорним кольором (рис. 2.30), дорівнюватись 1, а для всіх інших залишимо температуру дорівнюватись 0.

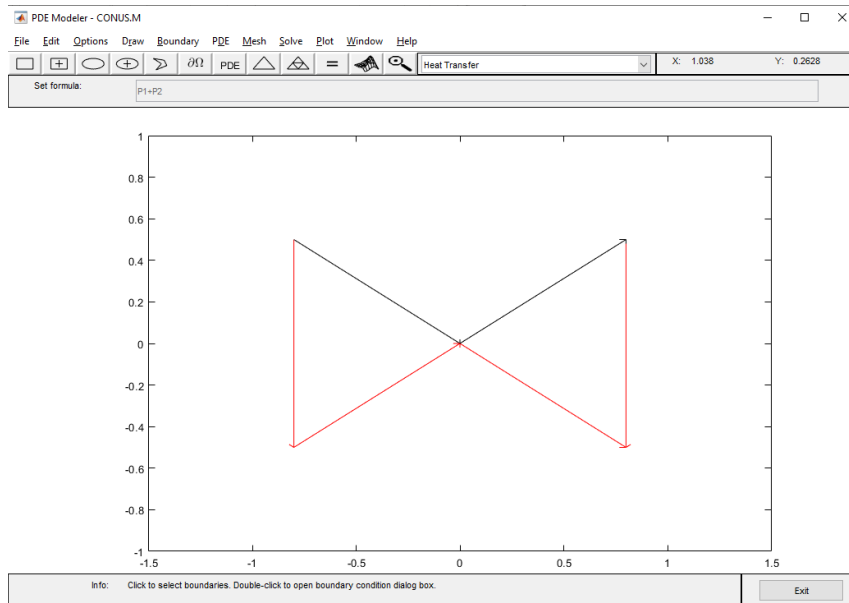


Рис. 2.30. Межі конічної поверхні

Проведемо триангуляцію, збільшимо щільність сітки. В результаті отримаємо наступний розрахунок температурних полів (рис. 2.31).

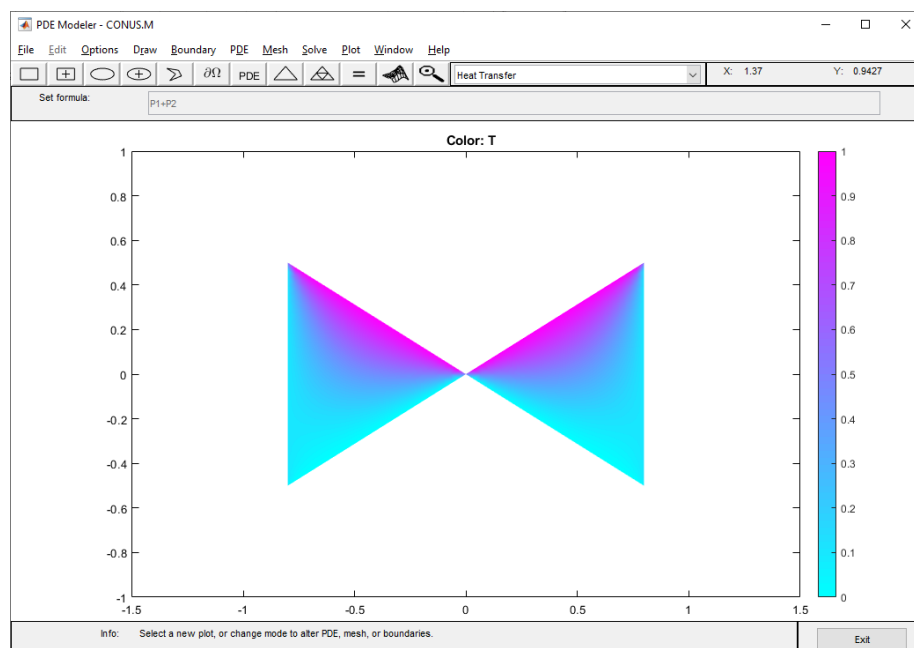


Рис. 2.31. Розподіл температури конічної поверхні

Тепер змодуємо поверхню, для розрахунку розподілу температурних полів для прокатного валу. Створимо наступну фігуру (рис. 2.32).

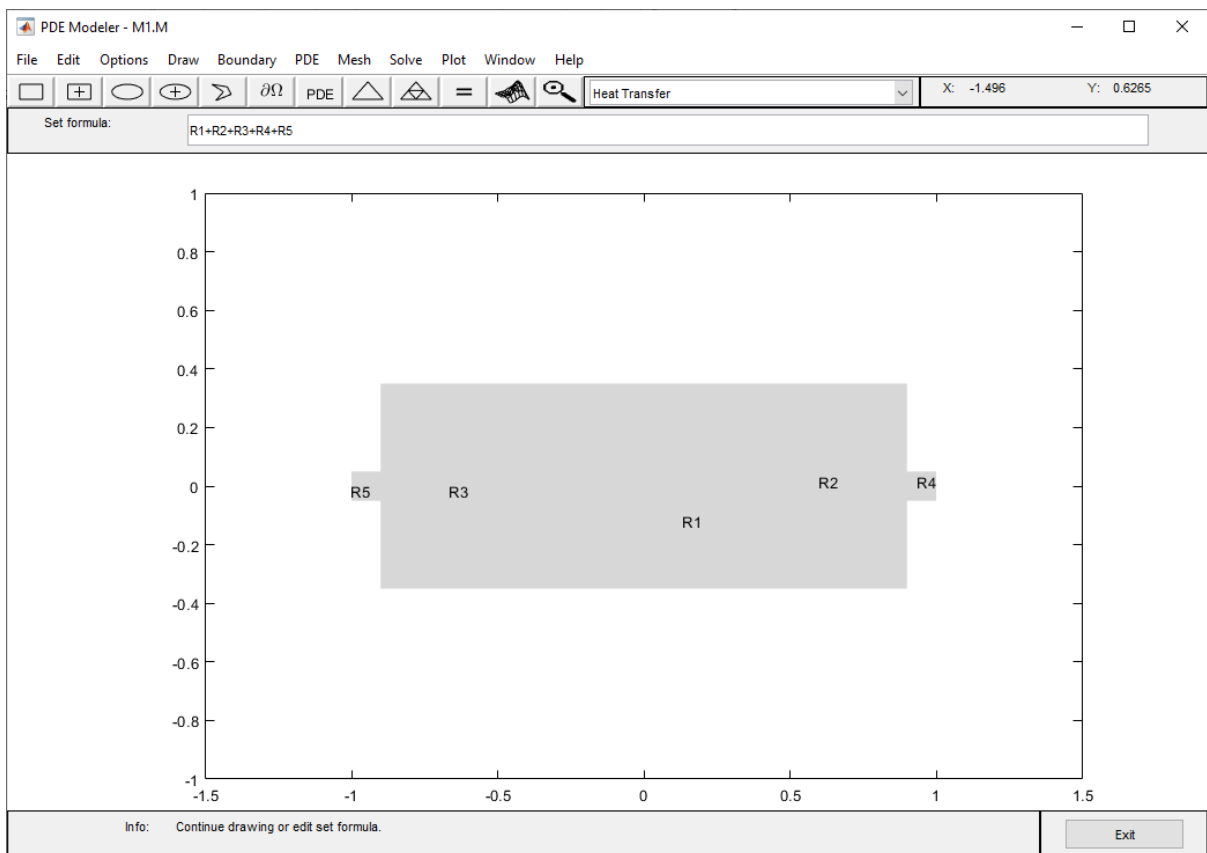


Рис. 2.32. Фігури для моделювання прокатного валу

Для верхньої центральної межі (межа 5) встановимо температуру рівною одиниці. Для меж 4 і 6 – 0.5. Для меж 8, 9 і 10 встановимо 0.1. Всі інші межі залишимо з температурою, яка дорівнює 0. Нумерацію меж можна побачити на рис. 2.33. Нумерацію меж можна підключити в меню Boundary.

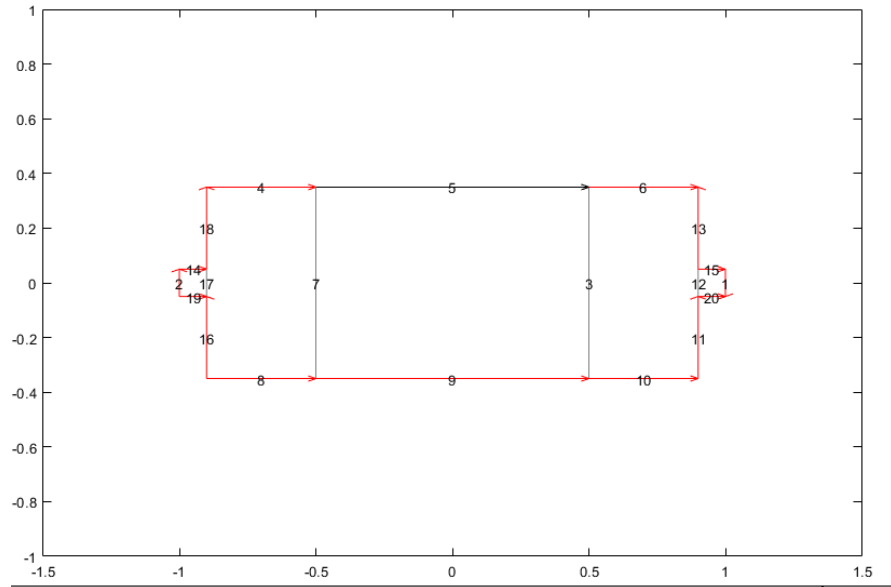


Рис. 2.33. Межі прокатного валу

В результаті отримаємо наступний розподіл температурних полів для прокатного валу (рис. 2.34).

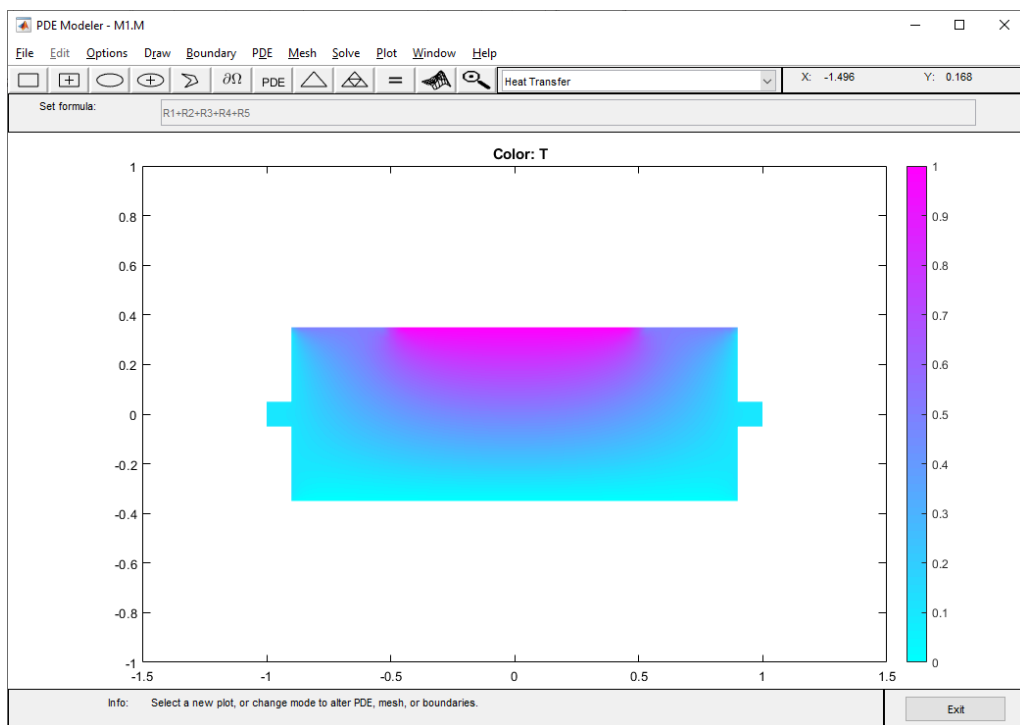


Рис. 2.34. Розподіл температури прокатного валу

У ході виконання кваліфікаційної роботи була знайдена програма, яка дозволяє розраховувати розподіл температурних полів за допомогою пакету PDE Toolbox. Для її роботи треба виконати файл зі скриптом у середі Matlab. Програма, використовуючи Command Window, поступово задає питання, для

проектування фігур, налаштування їх температур, а потім виводить результат у окреме вікно.

Створимо поверхню еліпсу за допомогою цієї програми. Запустимо програму і вкажемо, що нам потрібна одна фігура типу еліпс. Налаштуємо центр координати, півосі та кут нахилу у радіанах (рис.2.35).

```
Command Window
>> solve
Count of figures
1
Type of figure: 1 - circle, 2 - polygon, 3 - rectangle, 4 - ellipse
4
ellipse
x coordinate
0
y coordinate
0
semiaxis first
0.8
semiaxis second
0.5
rotational angle
0
Set formula
E1
Line temperature
0
Line temperature
0
Line temperature
0.7
Line temperature
1
```

Рис. 2.25. Параметри еліпсу

Matlab виведе розподіл температури для еліпсу у окремому вікні (рис.2.36).

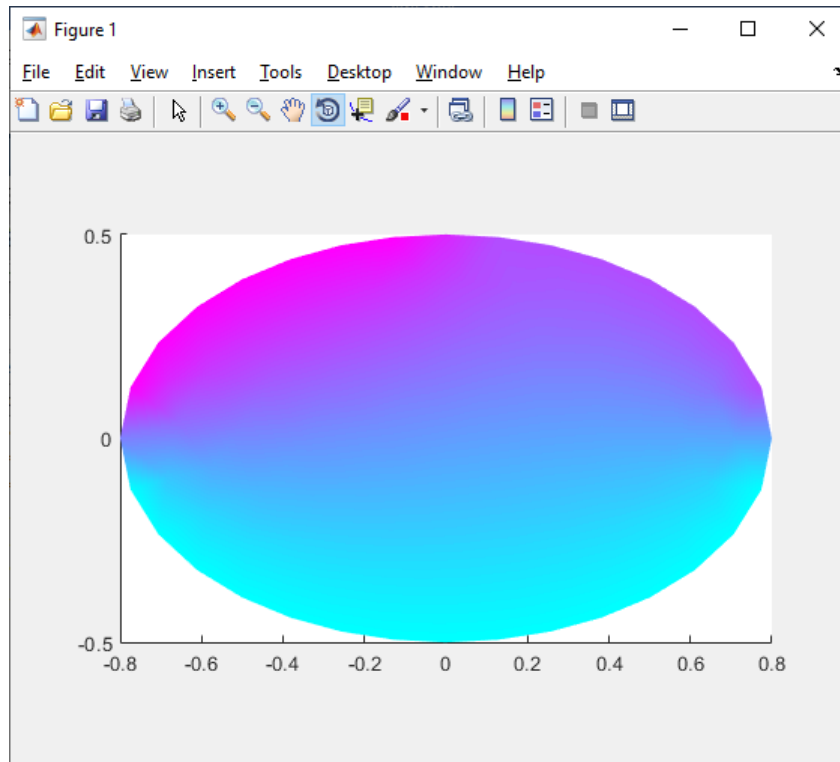


Рис. 2.36. Розподіл температури для еліпсу

Повторимо моделювання однопорожнинного гіперболоїда за допомогою програми. Тут доведеться зробити три фігури: прямокутник (рис. 2.37) і два еліпса (рис. 2.38). Розмістимо їх так само, як це було зроблено за допомогою інтерфейсу. У формулі напишемо, що еліпси потрібно відняти від прямокутника.

```

Command Window

>> solve
Count of figures
3
Type of figure: 1 - circle, 2 - polygon, 3 - rectangle, 4 - ellipse
3
rectangle
x coordinate of 1 line
-0.8
y coordinate of 1 line
-0.5
x coordinate of 2 line
0.8
y coordinate of 2 line
-0.5
x coordinate of 3 line
0.8
y coordinate of 3 line
0.5
x coordinate of 4 line
-0.8
y coordinate of 4 line
0.5
Type of figure: 1 - circle, 2 - polygon, 3 - rectangle, 4 - ellipse
4
ellipse

```

Рис. 2.37. Параметри прямокутника однопорожнинного гіперболоїда

```
Command Window
x coordinate
0
y coordinate
0.5
semiaxis first
0.8
semiaxis second
0.2
rotational angle
0
Type of figure: 1 - circle, 2 - polygon, 3 - rectangle, 4 - ellipse
4
ellipse
x coordinate
0
y coordinate
-0.5
semiaxis first
0.8
semiaxis second
0.2
rotational angle
0
Set formula
R1-E1-E2
```

Рис. 2.38. Параметри еліпсів і формула однопорожнинного гіперboloїда

Задаємо наступні значення температур (рис. 2.39).

```
Command Window
Line temperature
0.6
Line temperature
0.6
Line temperature
1
Line temperature
1
Line temperature
0
Line temperature
0
```

Рис. 2.39. Параметри температур меж

Однопорожнинний гіперboloїд повторює результат (рис. 2.40), який ми отримали, вирішуючи цю задачу за допомогою вікна PDE Modeler.

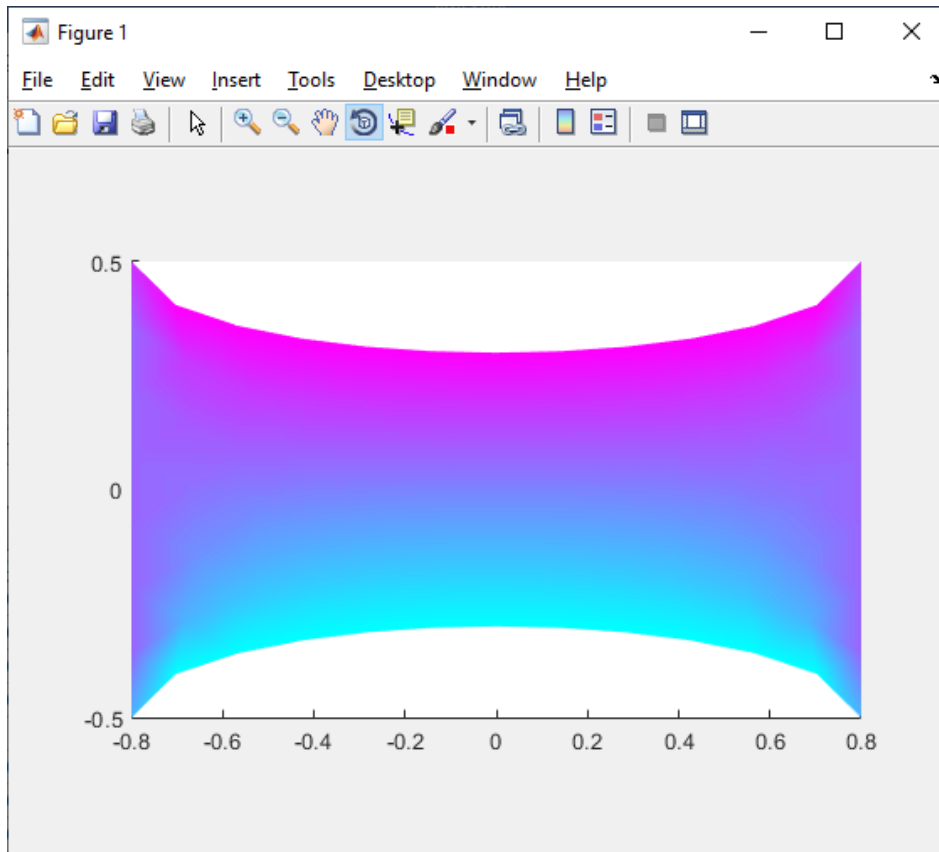


Рис. 2.40. Розподіл температури однопорожнинного гіперboloїда

Тепер зробимо розподіл температури для конічної поверхні. Для цього потрібно зробити два полігона (рис. 2.41 і рис. 2.42). Запустимо програму і введемо потрібні дані, напишемо температури меж (рис. 2.43).

```

Command Window
>> solve
Count of figures
2
Type of figure: 1 - circle, 2 - polygon, 3 - rectangle, 4 - ellipse
2
polygon
number of lines
3
x coordinate of 1 line
0
y coordinate of 1 line
0
x coordinate of 2 line
-0.8
y coordinate of 2 line
-0.5
x coordinate of 3 line
-0.8
y coordinate of 3 line
0.5
Type of figure: 1 - circle, 2 - polygon, 3 - rectangle, 4 - ellipse
2
polygon
number of lines

```

Рис. 2.41. Параметри першого полігону конічної поверхні

```

Command Window
y coordinate of 3 line
0.5
Type of figure: 1 - circle, 2 - polygon, 3 - rectangle, 4 - ellipse
2
polygon
number of lines
3
x coordinate of 1 line
0
y coordinate of 1 line
0
x coordinate of 2 line
0.8
y coordinate of 2 line
-0.5
x coordinate of 3 line
0.8
y coordinate of 3 line
0.5
Set formula
P1+P2

```

Рис. 2.42. Параметри другого полігону кінчної поверхні

```

Command Window
Line temperature
0
Line temperature
0
Line temperature
1
Line temperature
0
Line temperature
0
Line temperature
1

```

Рис. 2.43. Параметри температури меж кінчної поверхні

Як результат отримаємо кінчну поверхню подібну до попередньої, зробленої у PDE Modeler (рис. 2.44).

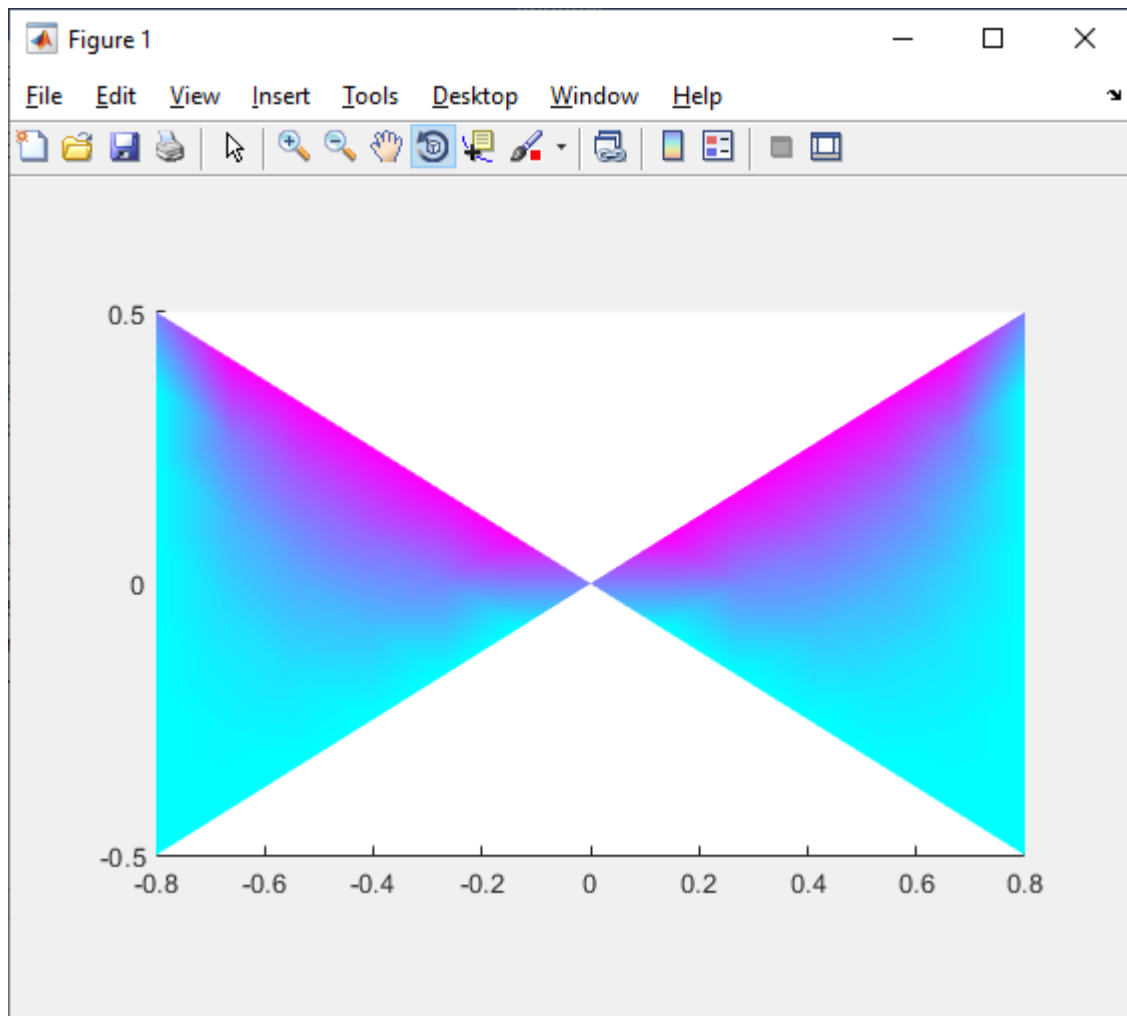


Рис. 2.44. Розподіл температури конічної поверхні

Тепер створимо поверхню прокатного валу. Для цього потрібно створити 5 прямокутників (рис.2.45–2.49). Додаємо інформацію про температури їх меж (рис. 2.50-2.51).


```
Command Window
Count of figures
5
Type of figure: 1 - circle, 2 - polygon, 3 - rectangle, 4 - ellipse
3
rectangle
x coordinate of 1 line
-0.5
y coordinate of 1 line
-0.35
x coordinate of 2 line
0.5
y coordinate of 2 line
-0.35
x coordinate of 3 line
0.5
y coordinate of 3 line
0.35
x coordinate of 4 line
-0.5
y coordinate of 4 line
0.35
```

Рис. 2.45. Параметры первого прямоугольника прокатного валу

```
Command Window
0.35
Type of figure: 1 - circle, 2 - polygon, 3 - rectangle, 4 - ellipse
3
rectangle
x coordinate of 1 line
0.5
y coordinate of 1 line
-0.35
x coordinate of 2 line
0.9
y coordinate of 2 line
-0.35
x coordinate of 3 line
0.9
y coordinate of 3 line
0.35
x coordinate of 4 line
0.5
y coordinate of 4 line
0.35
```

Рис. 2.46. Параметры второго прямоугольника прокатного валу

```
Command Window
Type of figure: 1 - circle, 2 - polygon, 3 - rectangle, 4 - ellipse
3
rectangle
x coordinate of 1 line
-0.9
y coordinate of 1 line
-0.35
x coordinate of 2 line
-0.5
y coordinate of 2 line
-0.35
x coordinate of 3 line
-0.5
y coordinate of 3 line
0.35
x coordinate of 4 line
-0.9
y coordinate of 4 line
0.35
```

Рис. 2.47. Параметры третьего прямоугольника прокатного валу

```
Command Window
Type of figure: 1 - circle, 2 - polygon, 3 - rectangle, 4 - ellipse
3
rectangle
x coordinate of 1 line
0.9
y coordinate of 1 line
-0.05
x coordinate of 2 line
1
y coordinate of 2 line
-0.05
x coordinate of 3 line
1
y coordinate of 3 line
0.05
x coordinate of 4 line
0.9
y coordinate of 4 line
0.05
```

Рис. 2.48. Параметры четвертого прямоугольника прокатного валу

```
Command Window
Type of figure: 1 - circle, 2 - polygon, 3 - rectangle, 4 - ellipse
3
rectangle
x coordinate of 1 line
-1
y coordinate of 1 line
-0.05
x coordinate of 2 line
-0.9
y coordinate of 2 line
-0.05
x coordinate of 3 line
-0.9
y coordinate of 3 line
0.05
x coordinate of 4 line
-1
y coordinate of 4 line
0.05
Set formula
R1+R2+R3+R4+R5
```

Рис. 2.49. Параметри п'ятого прямокутника і формула прокатного валу

```
Command Window
Line temperature
0
Line temperature
0
Line temperature
0.1
Line temperature
0.1
Line temperature
0.1
Line temperature
0.1
Line temperature
0.1
Line temperature
0.1
Line temperature
0.1
Line temperature
0.5
Line temperature
1
Line temperature
0.5
Line temperature
0.1
Line temperature
0.1
Line temperature
0.1
```

Рис. 2.50. Параметри температури меж прокатного валу

```
Command Window
Line temperature
0.1
Line temperature
0.1
Line temperature
0
Line temperature
0
Line temperature
0
Line temperature
0
Line temperature
0
Line temperature
0
```

Рис. 2.51. Параметри температури меж прокатного валу

У окремому вікні отримуємо розподіл температури для прокатного валу (рис. 2.52).

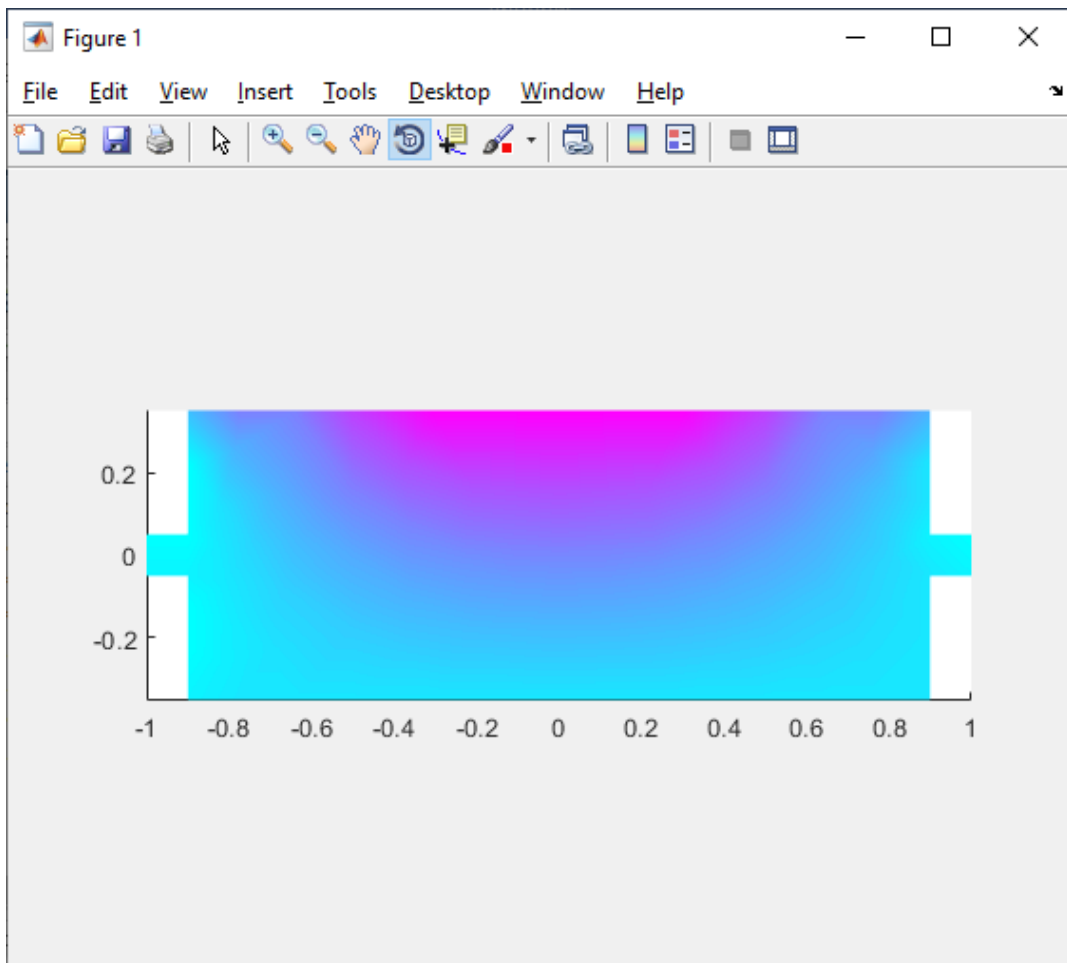


Рис. 2.52. Розподіл температури прокатного валу

Порівняємо результат роботи програми з аналітичним рішенням. Для цього оберемо прямокутник із температурою 1 на одній стороні, а на інших – 0. Для отримання рішення використаємо формулу (2.4).

$$u(x, y) = \left(\frac{\pi^2 + b^2 a}{\pi^2 \operatorname{sh} \frac{\pi a}{b}} \operatorname{sh} \frac{\pi x}{b} - \frac{b^2 x}{\pi^2} \right) \sin \frac{\pi y}{b} \quad (2.4)$$

де

a і b – сторони прямокутника.

За допомогою формули наступний результат рішення це (рис. 2.53).

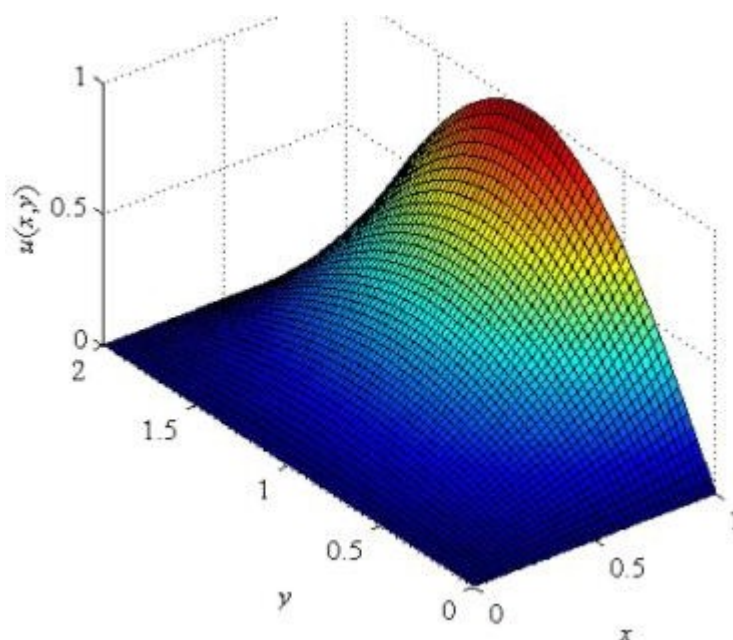


Рис. 2.53. Результат аналітичного розв'язання

Порівняємо цей результат з рішенням за допомогою програми (рис. 2.54).

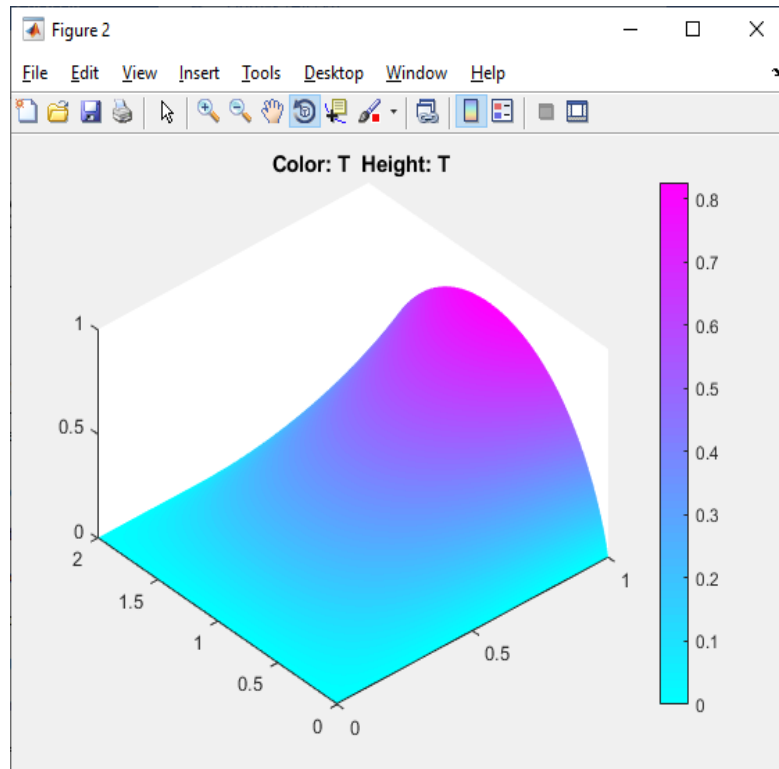


Рис. 2.54. Розподіл температури за допомогою програми

Як можна побачити, фігури мають практично однаковий розподіл. Тому можна вважати, що програма працює з достатньо точним результатом.

РОЗДІЛ 3

ЕКОНОМІЧНА ЧАСТИНА

3.1. Визначення трудомісткості та вартості розробки програмного продукту

Початкові дані:

1. передбачуване число операторів програми –653.
2. коефіцієнт складності програми – 1,5.
3. коефіцієнт корекції програми в ході її розробки – 0,4.
4. годинна заробітна плата програміста– 145 грн/год.
5. коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі – 1,3.
6. коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності –1,1.
7. вартість машино-години ЕОМ –11,25 грн/год.

Заробітна плата представників ІТ-департаменту залежить від багатьох факторів. Це також попит на мови програмування, попит на спеціалізацію. У деяких випадках початківець DevOps-фахівець може заробити більше, ніж розробник JavaScript середнього рівня, маючи досвід і знання. Такі мови, як Java, PHP, JavaScript і Python, існують уже кілька років. Велику роль відіграє і спеціалізація. «Існують категорії співробітників, які тільки приходять у цю сферу, освоюють її ази. Отримують вони в рази менше. Працівник рівня Junior на початку старту кар'єри зазвичай заробляє 15-20 тис. грн.» [26]

Весь процес розробки продукту можна розділити на дві частини: одноразова розробка white label-основи та процес брендування під конкретного замовника якщо він проявить таке бажання(в моїй роботі це не використовується).

Замість платного Github використовується безкоштовний GitLab. Як хостинг для сервера використовується Хмара.

Мінімальні місячні витрати на Інтернет, електроенергію та обладнання для розробки сайту залежать від параметрів, які ви виберете для цих ресурсів.
приклад:

- Обладнання ви можете мати будь-яке на ваш вибір та змогу. Це може бути як і новий так і б/у комп'ютер , з гарантією чи без. За знайденою інформацією[27] якщо припустити , що комп'ютер ви використовували протягом декількох років, а комплектуючі не оновлювали, то щомісячна амортизація обладнання становить приблизно 250 грн;
- Інтернет: Ви можете вибрати тарифний план з необмеженими даними, різною швидкістю, або додатковими послугами . За даними середня вартість інтернету в Україні становить близько 100 грн на місяць;[28]
- Електроенергія: ви можете економити електроенергію у різний спосіб, регулюючи яркість екрану монітору та виключаючи пристрій коли не працюєте. За отриманими даними при пошуку отримуємо інформацію, що якщо ваш комп'ютер споживає приблизно 200 Вт/год. , а ви працюєте 8 годин, то ваше споживання за місяць приблизно становить 96 грн[29];

Таким чином , мінімальні щомісячні витрати на обладнання , електроенергію та інтернет для створення програмного забезпечення становить 450 грн.

Нормування праці в процесі створення ПЗ істотно ускладнено в силу творчого характеру праці програміста. Тому трудомісткість розробки ПЗ може бути розрахована на основі системи моделей з різною точністю оцінки.

Проведемо підрахунки і отримуємо що розробник за годину витрачає приблизно 11,25 грн

Трудомісткість розробки ПЗ можна розрахувати за формулою:

$$t = t_o + t_u + t_a + t_n + t_{отл} + t_{\partial}, \text{ людино-годин,} \quad (3.1)$$

де t_o - витрати праці на підготовку й опис поставленої задачі (приймається 50 людино-годин);

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;

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

t_d - витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у програмному забезпеченні, яке розробляється.

Умовне число операторів (підпрограм):

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

де q - передбачуване число операторів;

C - коефіцієнт складності програми;

p - коефіцієнт кореляції програми в ході її розробки.

Отримаємо:

$$Q = 653 \cdot 1.5 \cdot (1 + 0.4) = 1371.3$$

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

$$t_u = \frac{Q \cdot B}{(75..85) \cdot k}, \text{ ЛЮДИНО-ГОДИН,}$$

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

k - коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності.

Отримаємо:

$$t_u = \frac{1371.3 \cdot 1.3}{81 \cdot 1.1} = 20, \text{ люДИНО-ГОДИН,}$$

Витрати праці на розробку алгоритму рішення задачі:

$$t_a = \frac{Q}{(20 \dots 25) \cdot k}, \text{ люДИНО-ГОДИН.}$$

Отримаємо:

$$t_a = \frac{1371.3}{23 \cdot 1.1} = 54.2, \text{ люДИНО-ГОДИН}$$

Витрати на складання програми по готовій блок-схемі:

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

Отримаємо:

$$t_n = \frac{1371.3}{23 \cdot 1.1} = 54.2, \text{ люДИНО-ГОДИН}$$

Витрати праці на налагодження програми на ЕОМ:

- за умови автономного налагодження одного завдання:

$$t_{отл} = \frac{Q}{(4.5) \cdot k}$$

Отримаємо:

$$t_{отл} = \frac{1371.3}{4.1 \cdot 1.1} = 304 \text{ люДИНО-ГОДИН}$$

- за умови комплексного налагодження завдання:

$$t_{отл}^k = 1.5 \cdot t_{отл}$$

Отримаємо:

$$t_{\text{отл}}^k = 1.5 * 304 = 456$$

Витрати праці на підготовку документації:

$$t_{\partial} = t_{\partial p} + t_{\partial o},$$

де $t_{\partial p}$ - трудомісткість підготовки матеріалів і рукопису

$$t_{\partial p} = \frac{Q}{(15 \dots 20) \cdot k}, \text{ ЛЮДИНО-ГОДИН.}$$

Підрахуємо:

$$t_{\text{др}} = \frac{1371.3}{16 * 1.1} = 77.9$$

$t_{\partial o}$ - трудомісткість редагування, печатки й оформлення документації

$$t_{\partial o} = 0,75 \cdot t_{\partial p}, \text{ ЛЮДИНО-ГОДИН.}$$

Підрахуємо:

$$t_{\text{до}} = 0,75 * 77,9 = 58,4$$

Отже витрати праці на підготовку документації:

$$t_{\text{д}} = 77.9 + 58.4 = 136.3$$

Повертаючись до формули (3.1) ми отримуємо:

$$t = 50.3 + 20 + 54.2 + 54.2 + 304 + 136.3 = 619 \text{ ЛЮДИНО-ГОДИН}$$

3.2. Рахунок витрат на створення програми

Витрати на створення ПЗ $K_{по}$ включають витрати на заробітну плату виконавця програми $З_{зп}$ і витрат машинного часу, необхідного на налагодження програми на ЕОМ

$$K_{по} = З_{зп} + З_{мв}, \text{ грн.}$$

Заробітна плата виконавця визначається за формулою:

$$З_{зп} = t * C_{пр}, \text{ грн,}$$

де : t -загальна трудомісткість, людино-годин;

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

Підраховуємо:

$$З_{зп} = 619 * 145 = 89755 \text{ грн}$$

Вартість машинного часу , необхідного для налагодження програми на ЕОМ:

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

де $t_{омл}$ - трудомісткість налагодження програми на ЕОМ, год.

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

Підраховуємо:

$$З_{мв} = 304 * 11.25 = 3420 \text{ грн}$$

Отже повертаючись до формули (3.2.1) ми отримуємо :

$$K_{по} = 89755 + 3420 = 93175 \text{ грн}$$

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

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

де B_k - число виконавців (1);

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

Отримаємо:

$$T = \frac{527}{1 * 176} = 2.9 \text{ місяців}$$

Таким чином було розраховано наступні показники:

- Трудомісткість розробки ПЗ – 527 людино-годин;
- Витрати на створення – 93175 грн;
- Очікуваний період створення ПЗ – 2,9 міс

ВИСНОВОК

Метою кваліфікаційної роботи є розробка програмного забезпечення для розрахунку температурних полів при умовах Діріхле.. Робота є актуальною через великий попит на подібні розробки, вони оптимізують та спрощують дії щодо підрахунку..

Такі програми на даний момент часу є досить корисними, вони допомагають проводити підрахунки . Такі проекти є досить актуальними і мають широкий попит. Основна задача розроблювальної програми полягає в зменшенні часу та кількості помилок при підрахунку.

Отже, головна мета створення даної програми – допомога людям при вирішенні задач при підрахунку температурних полів та впевненості про зменшення кількості помилок , що полегшить їх розробку, оскільки аналіз розподілу температури може створювати більш ефективні та безпечні форми

Програма написана за допомогою такої мови web-програмування як Java.

Вона є однією із популярних крос-платформних об'єктно-орієнтованих мов програмування з суворою типізацією. Java має продуману та повну реалізацію об'єктно-орієнтованого програмування.

Для досягнення поставлених цілей виконано наступні завдання:

- Вивчення фізичний законів розподілу температури в температурному полі;
- вивчено методику розрахунку розподілу температурного поля;
- Проаналізовано метод скінченних елементів і рівняння.

Під час виконання даної кваліфікаційної роботи були виконані наступні задачі:

- створення зручного інтерфейсу для користувача;
- зручний спосіб введення даних;
- візуалізація отриманого рішення;
- порівняння знайденого рішення з його аналітичним розв'язком, якщо такий існує;

– перевірку вхідних даних на її коректність.

В «Економічному розділі» визначено трудомісткість розробки програмного забезпечення (527 чол-год), підраховані витрати на створення програмного забезпечення (93175 грн) і оікуваний період розробки (2,9 міс.)

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Приходько М.А. Термодинаміка та теплопередача. Навчальний посібник./ М.А. Приходько, Г.Г. Герасимов - Рівне: НУВГП, 2008.- 250 с.
2. J. Gembarovic J., Loffler M., Gembarovic Jr. J. Simple algorithm for temperature distribution calculations / J. Gembarovic, M. Loffler, J. Gembarovic Jr. // Applied Mathematical Modelling. – 2004. – Vol. 28, № 2. – P. 173-182.
3. Zhang Z. Rock Fracture and Blasting / Z. Zhang. – Oxford : Elsevier, 2016. – 1430 p.
4. Speight J.G. Natural Gas / J. G. Speight. – Oxford : GPP, 2018. – 462 p.
5. <https://dou.ua/lenta/articles/how-to-learn-java>
6. MacNeal, Richard H., page i., "The NASTRAN Theoretical Manual", December 1972
7. https://stud.com.ua/180965/prirodoznavstvo/metod_kintsevih_elementi
[v](#)
8. Васильєв В. А. Аналіз та вибір програмних продуктів для вирішення інженерних завдань приладобудування [Електронний ресурс] / В. А. Васильєв, М. А. Калмикова // Електронний науково-практичний журнал «Сучасна техніка та технології». -Режим доступу: technology.snauka.ru/2013/03/1702
9. А. В. Гільчук, А. А. Халатов, Т. В. Доник ТЕОРІЯ ТЕПЛОПРОВІДНОСТІ НАВЧАЛЬНИЙ ПОСІБНИК [Електронний ресурс]. Режим доступу <https://ela.kpi.ua/bitstream/123456789/39046/1/Teploprovidnist.pdf>
10. <https://budtehnika.pp.ua/7972-osnovi-teploperedach.html>
11. Чейлитко А.О. Математичне моделювання та оптимізація теплотехнічних процесів. Навчально-методичний посібник для студентів ЗДІА спеціальності 144 “Теплоенергетика” денної та заочної форми навчання / Укл.: А.О.Чейлитко – Запоріжжя. Видавництво ЗДІА, 2018.
12. Задачин В. М. 3-15 Чисельні методи : навчальний посібник / В. М. Задачин, І. Г. Конюшенко. – Х. : Вид. ХНЕУ ім. С. Кузнеця, 2014. – 180 с. (Укр.

мов.)

13. Задачин В. М. Робоча програма навчальної дисципліни "Чисельні методи" для студентів напряму підготовки "Комп'ютерні науки" всіх форм навчання / В. М. Задачин, І. Г. Конюшенко. – Х. : Вид. ХНЕУ, 2012.

14. Обчислювальна математика. Конспект лекцій: [Електронний ресурс]: навч. посіб. для студ. спеціальності 171 «Електроніка», спеціалізації «Електронні та інформаційні технології кінематографії та аудіовізуальних систем»/М.І. Романюк; О. А. Батіна; КПІ ім. Ігоря Сікорського.

15. Погорелов А.І. Тепломасообмін (основи теорії і розрахунку): Навчальний посібник для вузів, 2-ге видання. – Львів: «Новий світ - 2000», 2004.

16. Kosachev, V. S. Using rounding function in the problems of finite-element analysis / V. S. Kosachev, E. P. Koshevoy, S. A. Podgorny // Studies in mathematical science. 2012. – V. 4. No. 2. – P. 17-24.

17. Akin, J. E. Finite element analysis with error estimators / J. E. Akin. - Butterworth-Heinemann, 2005. – 512 p.

18. Chen, Z. X. Finite element methods and their applications / Z. X. Chen. –Berlin : Springer, 2005. – 424 p.

19. Solin, P. Partial differential equations and the finite element method / P. Solin. – Hoboken : Wiley-Interscience, 2006. – 504 p.

20. Thomee, V. Galerkin finite element methods for parabolic problems / V. Thomee. – Berlin : Springer, 2006. – 459 p.

21. Wilson, Jon S. Sensor Technology Handbook / Jon S. Wilson. – Oxford :Elsevier, 2004. – 704 p.

22. Marghitu, Dan B. Mechanical Engineer's Handbook. A volume in Academic Press Series in Engineering / Dan B. Marghitu. – San Diego : Academic Press, 2001. – 864 p.

23. Gumerov, Nail A. Fast Multipole methods for the Helmholtz Equation in Three Dimensions. A volume in Elsevier Series in Electromagnetism / Nail A. Gumerov, Ramani Duraiswami – Oxford : Elsevier, 2005. – 520 p.

24. Meyers, Robert A. Encyclopedia of Physical Science and Technology,

Third Edition / Robert A. Meyers – San Diego: Academic Press, 2001. – 15453 p.

25. Arfken, George B. University Physics / George B. Arfken, David F. Griffing, Donald C. Kelly, Joseph Priest – San Diego : Academic Press, 1984. – 905 p.

26. <https://lemon.school/uk/blog/skolko-zarabatyvayut-it-speczialysty-v-ukrayne-v-2023-godu>

27. <https://hotline.ua/computer/nastolnye-kompyutery/>

28. <https://www.ipay.ua/ua/internet>

29. <https://www.ukr.net/news/economics/finance/2021/10/01/101.html>

Код програми

```
Main.m;  
close all;  
  
%-----Вхідні параметри  
%sectNum - кількість секцій у тривимірній системі  
%k - кількість вузлів у тривимірній системі  
%H - висота  
%W - ширина  
%scale -  
%ro -  
%modY -  
%P -  
%sigmaT -  
%K -  
%Krit -  
  
g = 9.81;  
ro = 7800;  
modY = 210*10^9;  
sigmaT = 600*10^6;  
K = 2;  
Krit = sigmaT/K;  
  
h = msgbox('Завдання');  
[Selection,ok] = listdlg('ListString', {'1'; '2'; '3'}, 'Name', 'Виберіть номер  
завдання', 'ListSize', [300 90], 'SelectionMode', 'single ');  
  
switch Selection  
  
case 1  
    outPar=inputdlg({'Кількість секцій(6,10,14...)' 'Висота(м)' 'Ширина(м)' 'Довжина(м)'  
'Input Data',1,{'14','2','1','3','0.01','1'});
```

```

sectNum = str2double(outPar{1});
H = str2double(outPar{2});
W = str2double(outPar{3});
L = str2double(outPar{4});
A = str2double(outPar{5});
scale = str2double(outPar{6});
k = 2 * sectNum +4;
trussGen(k, H, L, W, modY, ro);
forceGen(k, H, L, W, A, ro, g);
forces = load('Forces.txt', '-ascii');
P = fminbnd (@ (P) minS (Selection, P, k, H, L, W, A, ro, g, Krit, modY), 0, 5000000);
msg = {'num2str(P)'};
outVar = msgbox(msg, 'Відповідь');

case 2
outPar=inputdlg({'Кількість секцій(6,10,14...) 'Висота(м)' 'Ширина(м)' 'Довжина(м)'}),
'Input Data',1,{'14','2','1','3','25000','1'});
sectNum = str2double(outPar{1});
H = str2double(outPar{2});
W = str2double(outPar{3});
L = str2double(outPar{4});
P = str2double(outPar{5});
scale = str2double(outPar{6});
k = 2 * sectNum +4;
trussGen(k, H, L, W, modY, ro);
A = fminbnd (@ (A) minS (Selection, P, k, H, L, W, A, ro, g, Krit, modY), 0, 5000);
msg = {'Мінімальна площа поперечного перерізу(м^2):',num2str(A)};
outVar = msgbox(msg, 'Відповідь');
forceGen(k, H, L, W, A, ro, g);
forces = load('Forces.txt', '-ascii');

case 3
outPar=inputdlg({'Кількість секцій(6,10,14...) 'Висота(м)' 'Ширина(м)' 'Площа
поперечного перерізу(м^2)' }, 'Input Data',1, {'14','2','1','25000','0.01','1'});
sectNum = str2double(outPar{1});

```

```

H = str2double(outPar{2});
W = str2double(outPar{3});
P = str2double(outPar{4});
A = str2double(outPar{5});
scale = str2double(outPar{6});
k = 2 * sectNum +4;
L = fminbnd (@ (L) minS (Selection, P, k, H, L, W, A, ro, g, Krit, modY), 1, 100);
msg = {'num2str(L)};
outVar = msgbox(msg, 'Відповідь');
trussGen(k, H, L, W, modY, ro);
forceGen(k, H, L, W, A, ro, g);
forces = load('Forces.txt', '-ascii');
otherwise
break;
end

Nodesload = load('nodes.txt', '-ascii');
nodes = reshape(Nodesload,1,k*3);

elements= load('elements.txt', '-ascii');

s = size (elements);
N1 = s(1);

s = size(forces);
f = reshape(forces', 1, s(1)*s(2));
s = size (f);

for i = 1: 1: s (2)
if mod(i, 6) == 0
f(i) = f(i) - P * g;
end;
end;

[u] = trussDisplace(elements, f, nodes, A, modY, k, N1);

```

```

u = scale*u;
nodesnew=nodes+u';

firstFig(elements, Nodesload, k, L);
secondFig(nodes,nodesnew,elements, L, k, N1, P,modY);

minS.m
function [a] = minS (Var, P, k, h, len, wid, AA, ro, g, Krit, modY)

if Var == 2
forceGen(k , h, len,wid, AA, ro, g);
end;

if Var == 3
trussGen(k, h, len, wid,modY,ro);
forceGen(k , h, len,wid, AA, ro, g);
end;
display(len)
forces = load('Forces.txt', '-ascii');

s = size (forces);
f=reshape(forces', 1, s(1)*s(2));

s = size (f);

for i = 1: 1: s (2)
if mod(i, 6) == 0
f(i) = f(i) - P * g;
end;
end;
Nodes = load('nodes.txt','-ascii');
Elementsload = load('elements.txt', '-ascii');
s = size (Nodes);
N=s(1);

```

```

nodes = reshape(Nodes',1,s(1)*3);

elements=Elementsload(:,[1 2]);

s = size (elements);
N1=s(1);

[u] = trussDisplace(elements, f, nodes, AA, modY, N, N1);

nodesnew=nodes+u';

epsilon = zeros(1,N1);
sigma = zeros (1, N1);
for i = 1:1:N1
xi1=nodes(elements(i,1)*3-2);
yi1=nodes(elements(i,1)*3-1);
zi1=nodes(elements(i,1)*3);
xj1=nodes(elements(i,2)*3-2);
yj1=nodes(elements(i,2)*3-1);
zj1=nodes(elements(i,2)*3);

xi2=nodesnew(elements(i,1)*3-2);
yi2=nodesnew(elements(i,1)*3-1);
zi2=nodesnew(elements(i,1)*3);
xj2=nodesnew(elements(i,2)*3-2);
yj2=nodesnew(elements(i,2)*3-1);
zj2=nodesnew(elements(i,2)*3);

L1 = sqrt((xi1 - xj1)^2 + (yi1 - yj1)^2 + (zi1 - zj1)^2);
L2 = sqrt((xi2 - xj2)^2 + (yi2 - yj2)^2+ (zi2 - zj2)^2);
epsilon(i) = (L2 - L1)/L1;
sigma(i) = modY * epsilon(i);

end;

```

```

t = max (abs (sigma)); display (t);
a = abs(t - Krit);
end

function [a] = minS (Var, P, k, h, len, wid, AA, ro, g, Krit, modY)

if Var == 2
forceGen(k , h, len,wid, AA, ro, g);
end;

if Var == 3
trussGen(k, h, len, wid,modY,ro);
forceGen(k , h, len,wid, AA, ro, g);
end;
display(len)
forces = load('Forces.txt', '-ascii');

s = size (forces);
f=reshape(forces', 1, s(1)*s(2));

s = size (f);

for i = 1: 1: s (2)
if mod(i, 6) == 0
f(i) = f(i) - P * g;
end;

end;
Nodes = load('nodes.txt','-ascii');
Elementsload = load('elements.txt', '-ascii');
s = size (Nodes);
N=s(1);
nodes = reshape(Nodes',1,s(1)*3);

elements=Elementsload(:,[1 2]);

```



```

s = size (elements);
N1=s(1);

[u] = trussDisplace(elements, f, nodes, AA, modY, N, N1);

nodesnew=nodes+u';

epsilon = zeros(1,N1);
sigma = zeros (1, N1);
for i = 1:1:N1
xi1=nodes(elements(i,1)*3-2);
yi1=nodes(elements(i,1)*3-1);
zi1=nodes(elements(i,1)*3);
xj1=nodes(elements(i,2)*3-2);
yj1=nodes(elements(i,2)*3-1);
zj1=nodes(elements(i,2)*3);

xi2=nodesnew(elements(i,1)*3-2);
yi2=nodesnew(elements(i,1)*3-1);
zi2=nodesnew(elements(i,1)*3);
xj2=nodesnew(elements(i,2)*3-2);
yj2=nodesnew(elements(i,2)*3-1);
zj2=nodesnew(elements(i,2)*3);

L1 = sqrt((xi1 - xj1)^2 + (yi1 - yj1)^2 + (zi1 - zj1)^2);
L2 = sqrt((xi2 - xj2)^2 + (yi2 - yj2)^2+ (zi2 - zj2)^2);
epsilon(i) = (L2 - L1)/L1;
sigma(i) = modY * epsilon(i);

end;

t = max (abs (sigma)); display (t);
a = abs(t - Krit);

```

end

trussGen.m

function [] = trussGen(k, H, L, W, modY, ro)

X = zeros (k, 3);

X(1,:) = 0;

X(2:2:k/2,1) = L:L:k/4*L;

X(3:2:k/2-1,1) = L:L:(k/4-1)*L;

X(2:1:k/2,2) = 0;

X(2:2:k/2,3) = 0;

X(3:2:k/2,3) = H;

X(k/2 + 1,1:3) = [0 W 0];

X(k/2+1:k,1) = X(1:k/2,1);

X(k/2+1:k,2) = W;

X(k/2+2:k,3) = X(2:1:k/2,3);

dlmwrite('nodes.txt',X)

%-----Генерація елементів

for i = 1:1:k/4-1

Y(2*i -1,1) = i;

Y(2*i -1,2) = i + 1;

Y(2*i,1) = i;

Y(2*i,2) = i + 2;

end;

for i = k/4:2:k/2-4

Y(2*i -1,1) = i;

Y(2*i -1,2) = i + 1;

Y(2*i +1,1) = i;

Y(2*i +1,2) = i + 3;

```
Y(2*i,1) = i;  
Y(2*i,2) = i + 2;  
end;
```

```
for i = k/4+1:2:k/2-3  
Y(2*i,1) = i;  
Y(2*i,2) = i + 2;  
end;
```

```
for i = k/2-2  
Y(2*i,1) = i;  
Y(2*i,2) = i + 2;  
Y(2*i -1,1) = i;  
Y(2*i -1,2) = i + 1;  
end;
```

```
for i=k/2-1  
Y(2*i -1,1) = i;  
Y(2*i -1,2) = i + 1;  
end;
```

```
for i = k/2+1:1:3*k/4-1  
Y(2*i -4,1) = i;  
Y(2*i -4,2) = i + 1;  
Y(2*i-3,1) = i;  
Y(2*i-3,2) = i + 2;  
end;
```

```
for i = 3*k/4:2:k-4  
Y(2*i -4,1) = i;  
Y(2*i -4,2) = i + 1;  
Y(2*i -2,1) = i;  
Y(2*i -2,2) = i + 3;  
Y(2*i-3,1) = i;
```

$Y(2*i-3,2) = i + 2;$

end;

for $i = 3*k/4+1:2:k-3$

$Y(2*i-3,1) = i;$

$Y(2*i-3,2) = i + 2;$

end;

for $i = k - 2$

$Y(2*i-3,1) = i;$

$Y(2*i-3,2) = i + 2;$

$Y(2*i -4,1) = i;$

$Y(2*i -4,2) = i + 1;$

end;

for $i = k-1$ $Y(2 * i -4,1) = i;$

$Y(2*i -4,2) = i + 1;$

end;

for $i = 1:1:k/2$

$Y(2*k-6+i,1) = i;$

$Y(2*k-6+i,2) = i + k/2;$

end;

for $i = 1:1:k/2-2$

if $i == 1$

$Y(2*k+i+(k/2-6),1) = i;$

$Y(2*k+i+(k/2-6),2) = i + 1+ k/2;$

else

$Y(2*k+i+(k/2-6),1) = i;$

$Y(2*k+i+(k/2-6),2) = i + 2+ k/2;$

end;

end;

```
Y(2*k+k-7,1) = 2;  
Y(2*k+k-7,2) = 1 + k/2;
```

```
for i = 4:1:k/2  
Y(3*k+i-10,1) = i;  
Y(3*k+i-10,2) = i-2 + k/2;  
end;
```

```
Y(7*k/2-9,1) = 1;  
Y(7*k/2-9,2) = k/2+3;  
Y(7*k/2-8,1) = k/2;  
Y(7*k/2-8,2) = k-1;
```

```
j=1;  
for i = 3:2:k/2-1  
Y(7*k/2-8+j,1) = i;  
Y(7*k/2-8+j,2) = k/2-1+i;  
j=j+1;  
end;
```

```
dlmwrite('elements.txt',Y)
```

```
trussDispalce.m
```

```
function [u] = trussDisplace( elements, f, nodes,A, E, k, N1)
```

```
K = zeros (k * 3, k * 3);
```

```
for i = 1: 1: N1;  
xi = nodes (elements (i, 1) * 3-2);  
yi = nodes(elements(i,1)*3-1);  
zi = nodes (elements (i, 1) * 3);  
xj = nodes (elements (i, 2) * 3-2);  
yj = nodes(elements(i,2)*3-1);  
zj = nodes (elements (i, 2) * 3);
```

```

dx = xj-xi;
dy = yj-yi;
dz = zj-zi;
l = sqrt (dx * dx + dy * dy + dz * dz);

cx = dx/l;
cy = dy/l;
cz = dz/l;
T = [cx cy cz 0 0 0; 0 0 0 cx cy cz];
Ke = (E * A / l). * (T' * [1 -1; -1 1] * T);

```

```

K([elements(i,1)*3-2 elements(i,1)*3-1 elements(i,1)*3 elements(i,2)*3-2 elements(i,2)*3- 1
elements(i,2)*3],...
[elements(i,1)*3-2 elements(i,1)*3-1 elements(i,1)*3 elements(i,2)*3-2 elements(i,2)*3-1
elements (i,2)*3])=...
K([elements(i,1)*3-2 elements(i,1)*3-1 elements(i,1)*3 elements(i,2)*3-2 elements(i,2)*3- 1
elements(i,2)*3],...
[elements(i,1)*3-2 elements(i,1)*3-1 elements(i,1)*3 elements(i,2)*3-2 elements(i,2)*3-1
elements (i,2)*3])+Ke;

```

```

end;

```

```

ux(1,:) = [1 0];
uy(1,:) = [1,0];
uz(1,:) = [1,0];

```

```

uy(2,:) =[k/2, 0];
uz(2,:) = [k/2,0];

```

```

ux(2,:) = [k/2+1 0];
uy(3,:) = [k/2+1,0];
uz(3,:) = [k/2+1,0];

```

```

uy(4,:)=[k,0];
uz(4,:)=[k,0];

f(ux(:,1)*3-2)= ux(:,2);
f(uy(:,1)*3-1)= uy(:,2);
f(uz(:,1)*3) = uz(:,2);
m = vertcat(ux(:,1)*3-2,uy(:,1)*3-1,uz(:,1)*3);

s = size(m);
K(m,:) = zeros(3*k,s(1))';
K(:,m) = zeros(3*k,s(1));
e(1:s(1),1) = 1;
K((m(:)-1)*k*3+m(:)) = e;
u = zeros (3 * k, 1);
u = Kf;

end

```

SecondFig.m

```
function [ ] = secondFig( nodes,nodesnew, elements, L,N, N1, P,modY )
```

```

figure;
hold on;
grid on;
xlabel('x');
ylabel('y');
zlabel('z');
colormap jet;
colorbar;

```

```

epsilon = zeros(1,N1);
for i = 1:1:N1
xi1(i)=nodes(elements(i,1)*3-2);
yi1(i)=nodes(elements(i,1)*3-1);

```

```

zi1(i)=nodes(elements(i,1)*3);
xj1(i)=nodes(elements(i,2)*3-2);
yj1(i)=nodes(elements(i,2)*3-1);
zj1(i)=nodes(elements(i,2)*3);

plot3([xi1(i) xj1(i)], [yi1(i) yj1(i)], [zi1(i) zj1(i)], 'Color', [0.5 0.9 0.9]);

xi2(i)=nodesnew(elements(i,1)*3-2);
yi2(i)=nodesnew(elements(i,1)*3-1);
zi2(i)=nodesnew(elements(i,1)*3);
xj2(i)=nodesnew(elements(i,2)*3-2);
yj2(i)=nodesnew(elements(i,2)*3-1);
zj2(i)=nodesnew(elements(i,2)*3);

L1 = sqrt((xi1(i) - xj1(i))^2 + (yi1(i) - yj1(i))^2 + (zi1(i) - zj1(i))^2);
L2 = sqrt((xi2(i) - xj2(i))^2 + (yi2(i) - yj2(i))^2 + (zi2(i) - zj2(i))^2);
epsilon(i) = (L2 - L1)/L1;
stress(i)= modY*epsilon(i);
patch([xi1(i) xj1(i)], [yi1(i) yj1(i)], [zi1(i) zj1(i)], [stress(i) stress(i)], 'EdgeColor', 'flat',
'LineWidth', 2);
end;

minX = min(nodes(1));
maxX = max(nodes(3*N-2));
minY = min(yi1(i), yj1(i));
maxY = max(yi1(i), yj1(i));
minZ = min(zi1(i), zj1(i));
maxZ = max(zi1(i), zj1(i));
axis([minX - 1 maxX + 1 minY - 1 maxY + 1 minZ - 1 maxZ + 1])

[~, mini] = min(epsilon);
[~, maxi] = max(epsilon); display(xi1(i))

for i=1:1:N1

```



```

plot3([xi2(i) xj2(i)], [yi2(i) yj2(i)], [zi2(i) zj2(i)], 'r');
end;

text((xi2(maxi)+xj2(maxi))/2, (yi2(maxi)+yj2(maxi))/2, (zi2(maxi)+zj2(maxi))/2,
num2str(epsilon(maxi)), 'FontSize', 12);
text((xi2(mini)+xj2(mini))/2, (yi2(mini)+yj2(mini))/2, (zi2(mini)+zj2(mini))/2,
num2str(epsilon(mini)), 'FontSize', 12);

```

```

plot3(nodesnew(46), nodesnew(47), nodesnew(48), '<');
plot3(nodesnew(25), nodesnew(26), nodesnew(27), '^');
plot3(nodesnew(1), nodesnew(2), nodesnew(3), '^');
plot3(nodesnew(22), nodesnew(23), nodesnew(24), '<');

```

```

for i=4:6:3*N/2-4
quiver3(nodesnew(i), nodesnew(i+1), nodesnew(i+2), 0, 0, -P/4, 5e-5, 'b');
end;

```

```

for i=3*N/2+4:6:3*N-5
quiver3(nodesnew(i), nodesnew(i+1), nodesnew(i+2), 0, 0, -P/4, 5e-5, 'b');
end;
daspect([1 1 1]);
view([-1 -1 1]);
end

```

forceGen.m

```

function [] = forceGen(k, H, L, W, A, ro, g)

```

```

HL = sqrt(H^2 + L^2);
WL = sqrt(W ^ 2 + L ^ 2);
HWL = sqrt(H^2 + WL^2);
WH = sqrt (H ^ 2 + W ^ 2);

```

```

V1 = H * A;

```

```

V2 = L * A;

```

$$\begin{aligned}V3 &= HL * A; \\V4 &= W * A; \\V5 &= WL * A; \\V6 &= HWL * A; \\V7 &= WH * A;\end{aligned}$$

$$\begin{aligned}m1 &= V1 * ro; \\m2 &= V2 * ro; \\m3 &= V3 * ro; \\m4 &= V4 * ro; \\m5 &= V5 * ro; \\m6 &= V6 * ro; \\m7 &= V7 * ro;\end{aligned}$$

$$\begin{aligned}F1 &= m1 * g / 2; \\F2 &= m2 * g / 2; \\F3 &= m3 * g / 2; \\F4 &= m4 * g / 2; \\F5 &= m5 * g / 2; \\F6 &= m6 * g / 2; \\F7 &= m7 * g / 2;\end{aligned}$$

$$\begin{aligned}\text{forces}(1, :) &= [0 \ 0 \ -(F2+F3+F4+F5+F6)]; \\ \text{forces}(2, :) &= [0 \ 0 \ -(2*F2+F1+F4+2*F5)]; \\ \text{forces}(3, :) &= [0 \ 0 \ -(F1+F2+2*F3+F4+F5+F7)];\end{aligned}$$

```
if k ~= 16
for i = 4:2:(k/4) - 2
forces(i, :) = [0 0 -(F1+2*F2+F3+F4+2*F5)];
end;
end;
```

```
if k ~= 16
for i = 5:2:(k/4) - 1
forces(i, :) = [0 0 -(F1+2*F2+F3+F4+2*F5+F7)];
```

```

end;
end;

forces(k/4, :) = [0 0 -(F1+2*F2+2*F3+F4+2*F5)];
forces(k/4 + 1, :) = [0 0 -(F1+2*F2+F4+2*F5+F7)];
if k ~= 16
for i = k/4 + 2:2:k/2 -4
forces(i, :) = [0 0 -(F1+2*F2+F3+F4+2*F5)];
end;
end;

if k ~= 16
for i = k/4 + 3:2:k/2 -3
forces(i, :) = [0 0 -(F1+2*F2+F3+F4+2*F5+F7)];
end;
end;

forces(k/2-2, :) = [0 0 -(F1+2*F2+F4+2*F5)];
forces(k/2-1, :) = [0 0 -(F1+F2+2*F3+F4+F5+F7)];
forces(k/2, :) = [0 0 -(F2+F3+F4+F5+F6)];
forces(k/2+1, :) = [0 0 -(F2+F3+F4+F5)];
forces(k/2+2, :) = [0 0 -(F1+2*F2+F4+2*F5+F7)];
forces(k/2+3, :) = [0 0 -(F1+F2+2*F3+F4+F5+F6)];

if k ~= 16
for i = k/2+4:2:(3*k/4) -2
forces(i, :) = [0 0 -(F1+2*F2+F3+F4+2*F5+F7)];
end;
end;

if k ~= 16
for i = k/2+5:2:(3*k/4) -1
forces(i, :) = [0 0 -(F1+2*F2+F3+F4+2*F5)];
end;
end;

```

```

forces(3*k/4, :) = [0 0 -(F1+2*F2+2*F3+F4+2*F5+F7)];
forces(3*k/4 + 1, :) = [0 0 -(F1+2*F2+F4+2*F5)];

```

```

if k ~= 16
for i = 3 * k / 4 + 2: 2: k -4
forces(i, :) = [0 0 -(F1+2*F2+F3+F4+2*F5+F7)];
end;
end;

```

```

if k ~= 16
for i = 3 * k / 4 + 3: 2: k -3
forces(i, :) = [0 0 -(F1+2*F2+F3+F4+2*F5)];
end;
end;

```

```

forces(k-2, :) = [0 0 -(F1+2*F2+F4+2*F5+F7)];
forces(k-1, :) = [0 0 -(F1+F2+2*F3+F4+F5+F6)];
forces(k, :) = [0 0 -(F2+F3+F4+F5)];

```

```

dlmwrite('Forces.txt',forces);

```

```

end

```

```

firstFig.m

```

```

function [ ] = firstFig( Elementsload, Nodes, k , L)

```

```

figure;
hold on;
grid on;
xlabel('x');

```

```

ylabel('y');
zlabel('z');
minX = min(Nodes(:, 1));
maxX = max(Nodes(:, 1));
minY = min(Nodes(:, 2));
maxY = max(Nodes(:, 2));
minZ = min(Nodes(:, 3));
maxZ = max(Nodes(:, 3));

axis([minX - 1 maxX + 1 minY - 1 maxY + 1 minZ - 1 maxZ + 1])
sE = size (Elementsload);
sN = size (Nodes);

for i = 1: 1: sN
x = Nodes (i, 1);
y = Nodes (i, 2);
z = Nodes (i, 3);
plot3(x,y,z, 'r', 'MarkerSize', 14);
text(x, y,z,strcat("", int2str(i)), 'color', [0 0 1]);
end;

для i = 1:1:sE
ni = Elementsload(i, 1);
nj = Elementsload(i, 2);
xi = Nodes (ni, 1);
xj = Nodes (nj, 1);
yi = Nodes (ni, 2);
yj = Nodes (nj, 2);
zi = Nodes (ni, 3);
zj = Nodes (nj, 3);
plot3([xi xj], [yi yj], [zi zj]);
text((xi + xj)/2, (yi + yj)/2, (zi + zj)/2, strcat("", int2str(i)));
end;

daspect([1 1 1]);

```

```
view([-1 -1 1]);  
end
```

ВІДГУК КЕРІВНИКА ЕКОНОМІЧНОГО РОЗДІЛУ

ПЕРЕЛІК ДОКУМЕНТІВ НА ОПТИЧНОМУ НОСІЇ

Ім'я файлу	Опис
Пояснювальні документи	
ПЗ_Козоріз.doc	Пояснювальна записка до кваліфікаційної роботи Документ Word.
ПЗ_Козоріз.pdf	Пояснювальна записка кваліфікаційної роботи в форматі PDF
Програма	
Program.rar	Архів. Містить коди програми
Презентація	
Презентація Козоріз.ppt	Презентація кваліфікаційної роботи