

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВОДНОГО ГОСПОДАРСТВА ТА
ПРИРОДОКОРИСТУВАННЯ

**Навчально-науковий інститут кібернетики, інформаційних технологій
та інженерії**

"До захисту допущена"

Зав. кафедри комп'ютерних наук та
прикладної математики

_____ 2024р.
«___» _____

КВАЛІФІКАЦІЙНА РОБОТА

**«Моделювання системи прогнозування напружено-деформованого
стану ґрунтових середовищ в нестационарному випадку»**

Виконав: Кравченко Роман Васильович

(прізвище, ім'я, по батькові)

група ПЗ-42 інт.

(підпис)

Керівник: доцент, к.т.н. Жуковська Н. А.

(науковий ступінь, вчене звання, посада, прізвище, ініціали)

(підпис)

Рівне – 2024

НН інститут **автоматики, кібернетики, інформаційних технологій та інженерії**

Кафедра **комп'ютерних наук та прикладної математики**

Освітньо-кваліфікаційний рівень **бакалавр**

Галузь знань 12 «Інформаційні технології»
(шифр і назва)

Спеціальність 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Турбал О. М.

“ ___ ” _____ 20__ року

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Кравчено Роман Васильович

(прізвище, ім'я, по батькові)

1. Тема роботи *"Моделювання системи прогнозування напружено-деформованого стану ґрунтових середовищ в нестационарному випадку"*

керівник роботи Жуковська Н. А., доцент, к.т.н.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від " 22 " квітня 2024 року
№ С - 525

2. Строк подання студентом роботи 01.05.2024

3. Вихідні дані до роботи: $L_x = 20.0, L_y = 10.0, n_x = 100, n_y = 50,$
 $n_t = 1000, \Delta t = 0.01, \alpha = 2 \times 10^{-3}, K = 1 \times 10^{-3}, D = 2 \times 10^{-4}, T_{left} = 30.0,$
 $T_{right} = 15.0, h_{left} = 10.0, h_{right} = 1.0, c_{left} = 1.0, c_{right} = 0.0.$

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): *Дослідити розподіл зміщень, напружень, деформацій, а також напорів, концентрації солей та температури по області дослідження в двовимірному випадку при наявності вільної поверхні та сил зв'язності.*

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): *Графіки розподілів зміщень, напружень, деформацій, напорів, концентрації солей, температури, напору; мультимедійна презентація.*

6. Консультанти розділів роботи

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
<i>Вступ,</i>	<i>Жуковська Н.А.</i>	<i>01.10.2023</i>	<i>01.10.2023</i>
<i>Розділ 1</i>	<i>Жуковська Н.А.</i>	<i>01.10.2023</i>	<i>01.10.2023</i>
<i>Розділ 2</i>	<i>Жуковська Н.А.</i>	<i>01.03.2024</i>	<i>01.03.2024</i>
<i>Розділ 3</i>	<i>Жуковська Н.А.</i>	<i>01.05.2024</i>	<i>01.05.2024</i>

7. Дата видачі завдання 01.10.23

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	<i>Вивчення літератури за обраною тематикою</i>	<i>01.10.2023-10.11.2023</i>	<i>виконав</i>
2.	<i>Формулювання постановки задачі</i>	<i>10.11.2023-21.10.2023</i>	<i>виконав</i>
3.	<i>Побудова математичної моделі</i>	<i>21.10.2023-27.10.2023</i>	<i>виконав</i>
4.	<i>Розробка обчислювального алгоритму розв'язку задачі</i>	<i>27.10.2023-10.11.2023</i>	<i>виконав</i>
5.	<i>Здійснення програмної реалізації</i>	<i>10.11.2023-25.11.2023</i>	<i>виконав</i>
6.	<i>Проведення чисельних експериментів</i>	<i>25.12.2023-27.02.2023</i>	<i>виконав</i>
7.	<i>Аналіз отриманих результатів</i>	<i>27.02.2023-30.04.2023</i>	<i>виконав</i>
8.	<i>Формулювання висновків роботи</i>	<i>30.04.2024-02.05.2024</i>	<i>виконав</i>
9.	<i>Вивчення питань з охорони праці</i>	<i>02.05.2024-10.05.2024</i>	<i>виконав</i>
10.	<i>Підготовка звіту кваліфікаційної роботи</i>	<i>10.05.2024-13.06.2024</i>	<i>виконав</i>
11.	<i>Підготовка мультимедійної презентації</i>	<i>13.06.2024-20.06.2024</i>	<i>виконав</i>
12.	<i>Підготовка до виступу</i>	<i>20.06.2024-23.06.2024</i>	<i>виконав</i>

Студент

_____ (підпис)

_____ (прізвище та ініціали)

Керівник роботи

_____ (підпис)

_____ (прізвище та ініціали)

ЗМІСТ

РЕФЕРАТ	5
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	6
ВСТУП	7
РОЗДІЛ I. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	10
1.1 Теоретичні основи напружено-деформованого стану ґрунтових середовищ	10
1.2 Огляд сучасних методів моделювання ґрунтових середовищ	14
1.3 Аналіз новітніх наукових матеріалів і розробок	18
1.4 Оцінка існуючих підходів до вирішення проблеми	22
РОЗДІЛ II. МЕТОДОЛОГІЯ ДОСЛІДЖЕННЯ	28
2.1 Опис програми GEO5	28
2.2 Опис програми RocScience	32
2.3 Аналіз та порівняння програм GEO5 та RocScience	36
2.4 Вибір мови та засобів програмування для реалізації задачі	40
2.5 Вибір архітектури програмного продукту	43
2.6 Вибір оптимальної моделі для дослідження	46
РОЗДІЛ III. РОЗРОБКА І ВПРОВАДЖЕННЯ СИСТЕМИ ПРОГНОЗУВАННЯ	48
3.1 Постановка задачі та вимоги до програмного продукту	48
3.2 Розробка основних компонентів програмного продукту	50
3.3 Реалізація математичної моделі та алгоритмів	51
3.4 Візуалізація результатів моделювання	54
ВИСНОВКИ	63
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	64
ДОДАТКИ	66

РЕФЕРАТ

Кваліфікаційна робота: 65 с., 21 малюнок, 5 таблиць, 1 додаток, 14 джерел.

Метою кваліфікаційної роботи є розробка та верифікація математичної моделі для прогнозування напружено-деформованого стану ґрунтових середовищ в умовах нестационарного навантаження.

Об'єкт дослідження – ґрунтові середовища, які знаходяться під впливом різних типів навантажень.

Предметом дослідження – моделі та методи прогнозування напружено-деформованого стану ґрунтових середовищ в нестационарному випадку.

Методи дослідження – використовується мова програмування Python та її бібліотеки, що забезпечують широкий спектр можливостей для чисельного моделювання, аналізу даних та візуалізації результатів.

Ключові слова: ГРУНТОВІ СЕРЕДОВИЩА, ПРОГНОЗУВАННЯ, ЧИСЛОВЕ МОДЕЛЮВАННЯ, АЛГОРИТМИ ЧИСЕЛЬНОГО МОДЕЛЮВАННЯ.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

MCE - Метод скінченних елементів

МДЕ - Метод дискретних елементів

MCP - Метод скінченних різниць

LEM - Лімітний рівноважний метод

МКЕ - Метод кінцевих елементів

Python - мова програмування Python

NumPy - Бібліотека Python для роботи з багатовимірними масивами

SciPy - Бібліотека Python для наукових і технічних обчислень

Matplotlib - Бібліотека Python для створення графіків

Pandas - Бібліотека Python для роботи з табличними даними

scikit-learn - Бібліотека Python для машинного навчання

GEO5 - Програмний комплекс для геотехнічного моделювання

RocScience - Програмний комплекс для геотехнічного моделювання та аналізу

RS2 - Двовимірний модуль RocScience для аналізу методом скінченних елементів

RS3 - Тривимірний модуль RocScience для аналізу методом скінченних елементів

Slide - Модуль RocScience для аналізу стійкості схилів

Dips - Модуль RocScience для статистичного аналізу структурних даних

RocSupport - Модуль RocScience для проектування тунелів

Settle3D - Модуль RocScience для тривимірного аналізу осідань та консолидації ґрунтів

PU - Графічний процесор

ВСТУП

Актуальність дослідження

Сучасні інженерні та геотехнічні проекти вимагають високої точності в прогнозуванні напружено-деформованого стану ґрунтових середовищ, оскільки це має критичне значення для безпечного та економічно ефективного будівництва. Нестационарний характер навантажень, таких як сейсмічні дії, підйом ґрунтових вод або динамічні навантаження від транспорту, викликає складні зміни у структурі ґрунтових середовищ, які важко передбачити за допомогою традиційних методів.

У зв'язку з цим, актуальність розробки нових моделей та методів прогнозування напружено-деформованого стану ґрунтів, що враховують нестационарні фактори, не викликає сумніву. Такий підхід дозволяє підвищити точність прогнозів і, як наслідок, знизити ризики аварій та надмірних витрат у процесі будівництва та експлуатації інженерних споруд.

Мета і завдання дослідження

Метою даної роботи є розробка та верифікація математичної моделі для прогнозування напружено-деформованого стану ґрунтових середовищ в умовах нестационарного навантаження. Для досягнення цієї мети необхідно вирішити наступні завдання:

1. Провести аналіз теоретичних основ та існуючих методів моделювання ґрунтових середовищ.
2. Оцінити новітні наукові розробки в області моделювання нестационарних процесів у ґрунтах.
3. Обґрунтувати вибір математичної моделі для опису напружено-деформованого стану ґрунтів.
4. Розробити алгоритми та програмне забезпечення для реалізації обраної моделі.
5. Провести верифікацію та валідацію моделі на основі експериментальних даних.

6. Створити систему прогнозування та оцінити її ефективність на прикладі реальних інженерних задач.

Наукова новизна роботи

Наукова новизна даного дослідження полягає у розробці нової математичної моделі, яка враховує нестационарні фактори впливу на напружено-деформований стан ґрунтових середовищ. Запропонована модель дозволяє більш точно описувати поведінку ґрунтів під дією різних типів навантажень, що змінюються з часом. Крім того, у роботі розроблені нові алгоритми та методи чисельного моделювання, що забезпечують високу точність і ефективність обчислень.

Практичне значення результатів

Практичне значення результатів роботи полягає в можливості їх застосування для вирішення реальних інженерних задач у галузі будівництва та геотехніки. Розроблена система прогнозування може бути використана для:

- Проектування та оцінки надійності фундаментів та інших інженерних споруд.

- Аналізу стабільності схилів і дамб.

- Прогнозування впливу динамічних навантажень, таких як сейсмічні дії, на ґрунтові масиви.

- Оцінки впливу підйому ґрунтових вод на будівельні об'єкти.

Використання результатів дослідження дозволить знизити ризики аварій та оптимізувати витрати на будівництво та експлуатацію інженерних споруд.

Структура роботи

Дипломна робота складається з п'яти розділів:

1. Вступ, де обґрунтована актуальність дослідження, визначені мета і завдання, наукова новизна та практичне значення результатів.

2. Аналіз предметної області, включаючи теоретичні основи, огляд сучасних методів моделювання та аналіз новітніх наукових розробок.

3. Методологія дослідження, де детально розглядаються вибір математичної моделі, алгоритми та методи розв'язання, а також розробка програмного забезпечення.

4. Розробка і впровадження системи прогнозування, з описом архітектури системи, використаних інформаційних технологій, результатів моделювання та їх аналізу.

5. Висновки, де узагальнюються результати дослідження, робляться висновки щодо наукової та практичної цінності роботи, а також надаються рекомендації для подальших досліджень.

Таким чином, дана робота спрямована на вирішення актуальної інженерної проблеми та має значний потенціал для практичного застосування у будівництві та геотехніці.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Теоретичні основи напружено-деформованого стану ґрунтових середовищ

Теорія напружено-деформованого стану ґрунтових середовищ є однією з ключових складових геотехніки та інженерної механіки ґрунтів. Вона виникла у відповідь на потребу забезпечення безпеки та надійності інженерних споруд, зокрема фундаментів, підпірних стінок та земляних дамб.

Історія розвитку теорії можна поділити на декілька ключових етапів:

- Початковий етап (до 19 століття):

Перші спроби розуміння поведінки ґрунтів під навантаженнями сягають часів античності. Давньоримські інженери використовували емпіричні підходи для будівництва доріг, акведуків та інших споруд. Проте теоретичні основи були відсутні.

- Класичний період (19 - початок 20 століття):

У цей час з'являються перші наукові роботи, що заклали основи геотехніки. Карл Кулон у 1776 році запропонував теорію руйнування ґрунтів, відому як теорія Кулона, яка описує умови стійкості ґрунтових масивів. У 1857 році Вільям Джон Макквей Карман запропонував теорію дренажу ґрунтів. Внесок Карла Терцагі, який у 1910 році розробив теорію ефективних напружень, є фундаментальним для сучасної геотехніки.

- Розвиток чисельних методів (середина 20 століття):

Після Другої світової війни розвиток обчислювальної техніки дозволив застосовувати чисельні методи для моделювання напружено-деформованого стану ґрунтів. Метод кінцевих елементів (МКЕ), розроблений у 1950-х роках, став революційним інструментом, який дозволив моделювати складні геотехнічні задачі з високою точністю.

- Сучасний період (кінець 20 - початок 21 століття):

Сучасні дослідження зосереджені на удосконаленні моделей, що враховують неоднорідність, анізотропність та нестационарність ґрунтів.

Використання потужних обчислювальних ресурсів, хмарних технологій та машинного навчання дозволяє створювати більш точні та надійні моделі.

Теоретичні основи

Напружено-деформований стан ґрунтових середовищ визначається взаємодією між зовнішніми навантаженнями та внутрішніми силами у ґрунті. Основними характеристиками, які описують цей стан, є напруження та деформації.

Напруження - це внутрішні сили, які діють на одиницю площі всередині ґрунтового масиву. Вони можуть бути нормальними (перпендикулярними до площини) та тангенціальними (дотичними до площини).

Деформація - це зміна форми та розмірів ґрунтового елемента під дією зовнішніх навантажень. Деформації можуть бути пружними (відновлюваними) та пластичними (незворотними).

Моделі ґрунтів

Для опису поведінки ґрунтів під навантаженнями використовуються різні математичні моделі. Найбільш поширеними є:

- Лінійно-пружна модель - ґрунт розглядається як лінійно-пружне середовище, в якому деформації пропорційні прикладеним напруженням. Ця модель застосовується для опису малих деформацій.
- Пружно-пластична модель - враховує пластичні деформації, що виникають після досягнення граничного стану напружень. Прикладом є модель Мора-Кулона, яка описує граничний стан ґрунту з урахуванням його когезії та внутрішнього кута тертя.
- Гіпопластична модель - розроблена для опису поведінки ґрунтів в умовах великих деформацій. Ця модель включає нелінійні залежності між напруженнями та деформаціями.

Основні рівняння, які описують напружено-деформований стан ґрунтів, включають рівняння рівноваги, рівняння сумісності та закони матеріалу.

1. Рівняння рівноваги забезпечують баланс внутрішніх та зовнішніх сил у ґрунтовому масиві. Вони записуються у вигляді диференціальних рівнянь для напружень.

2. Рівняння сумісності визначають умови, за яких деформації в ґрунті є сумісними, тобто не призводять до розривів чи перекриттів у матеріалі.

3. Закони матеріалу описують зв'язок між напруженнями та деформаціями у ґрунті. Для різних моделей ґрунтів ці закони мають різний вигляд.

Напружено-деформований стан у нестационарному випадку

У випадку нестационарного навантаження напружено-деформований стан ґрунтового середовища змінюється з часом. Це може бути спричинено різними факторами, такими як:

- Динамічні навантаження (наприклад, сейсмічні хвилі, вібрації від транспорту).
- Зміни у рівні ґрунтових вод, що призводять до зміни порового тиску.
- Повзучість ґрунтів, коли деформації наростають з часом під постійним навантаженням.

Для моделювання нестационарного стану використовуються динамічні рівняння руху та відповідні граничні умови. Важливу роль відіграють також початкові умови, які визначають стан ґрунту на початку процесу моделювання.

Сучасні підходи та методи

Аналітичні методи

Аналітичні методи базуються на спрощених моделях та дозволяють отримувати швидкі оцінки стану ґрунтів. До них відносяться різні теоретичні підходи, зокрема, методи Кулона та Ранкіна, які застосовуються для оцінки стійкості схилів і розрахунку тиску на підпірні стінки.

Чисельні методи

Чисельні методи, такі як метод кінцевих елементів (МКЕ), дозволяють моделювати складні системи та враховувати реальні умови експлуатації. Вони широко використовуються для аналізу напружено-деформованого стану

ґрунтів під дією різних навантажень, включаючи динамічні та нестационарні процеси. Основні переваги чисельних методів:

- Можливість моделювання складної геометрії та неоднорідних матеріалів.
- Висока точність розрахунків.
- Гнучкість у налаштуванні моделі під конкретні умови.
- Експериментальні методи

Експериментальні методи включають лабораторні та польові дослідження для визначення фізико-механічних властивостей ґрунтів. Лабораторні випробування, такі як тріщинні випробування та випробування на зсув, дозволяють отримати дані для калібрування математичних моделей. Польові дослідження, включаючи геофізичні методи та інструментальні вимірювання, забезпечують інформацію про реальні умови в природних ґрунтових масивах.

Незважаючи на значні досягнення в області моделювання напружено-деформованого стану ґрунтових середовищ, існує ряд викликів, які потребують подальших досліджень. Серед них:

- Складність математичних моделей, що враховують різноманітні фактори впливу, такі як неоднорідність, анізотропність та часові зміни властивостей ґрунтів.
- Неоднорідність та анізотропність ґрунтів, що ускладнює моделювання їхньої поведінки.
- Вплив екологічних та кліматичних змін на стан ґрунтових середовищ, включаючи підвищення рівня ґрунтових вод та зміну температурних режимів.

У даному розділі було розглянуто історію розвитку та теоретичні основи напружено-деформованого стану ґрунтових середовищ. Встановлено, що для моделювання поведінки ґрунтів під навантаженням використовуються різні математичні моделі, які враховують як пружні, так і пластичні деформації. Описано основні рівняння та принципи, що лежать в основі цих моделей.

Також розглянуто сучасні методи та підходи до моделювання нестационарних процесів у ґрунтах. Визначено основні виклики та перспективи подальших досліджень у цій області.

Таким чином, аналіз предметної області показав, що існує необхідність у подальшому розвитку теоретичних основ та методів моделювання напружено-деформованого стану ґрунтових середовищ, особливо в умовах нестационарного навантаження. Це дозволить підвищити точність прогнозів та забезпечити безпеку та надійність інженерних споруд, що зводяться на різних типах ґрунтів.

1.2 Огляд сучасних методів моделювання ґрунтових середовищ

Моделювання ґрунтових середовищ є однією з найважливіших задач в геотехнічній інженерії. Від точності моделювання залежить надійність та безпека інженерних споруд. Існує безліч методів, що дозволяють моделювати напружено-деформований стан ґрунтових середовищ. У цьому розділі розглянемо сучасні методи моделювання, поділяючи їх за призначенням та основними характеристиками.

Класифікація методів моделювання

Сучасні методи моделювання ґрунтових середовищ можна поділити на кілька основних груп:

1. Пружні та нелінійно-пружні моделі
2. Пластичні моделі
3. В'язкопластичні моделі
4. Консолідаційні моделі
5. Методи на основі чисельного моделювання

Пружні та нелінійно-пружні моделі

Лінійно-пружні моделі використовуються для опису поведінки ґрунтів при невеликих деформаціях. Вони базуються на законі Гука, що визначає лінійну залежність між напруженнями і деформаціями. Основні рівняння, що описують лінійно-пружну модель:

$$\sigma = E \cdot \varepsilon$$

де σ — напруження;

E — модуль пружності;

ε — деформація.

Нелінійно-пружні моделі враховують нелінійний характер деформацій ґрунтів при більших навантаженнях. Вони дозволяють більш точно описувати поведінку ґрунтів у діапазоні великих деформацій.

Пластичні моделі

Пластичні моделі використовуються для опису незворотних деформацій ґрунтів. Вони важливі для прогнозування поведінки ґрунтів після досягнення границі текучості. Основні моделі пластичності включають:

Модель Кулона-Мора

Ця модель є однією з найпоширеніших для опису міцності ґрунтів. Вона визначає умови, за яких ґрунт переходить у пластичний стан:

$$\tau = c + \sigma \tan \phi$$

τ — дотичне напруження;

c — коефіцієнт зчеплення;

σ — нормальне напруження;

ϕ — кут внутрішнього тертя.

Модель Друкера-Прагера

Ця модель є узагальненням критерію Кулона-Мора і використовується для опису поведінки ґрунтів у тривимірному напруженому стані:

$$\sqrt{J_2} = A + BI_2$$

де J_2 — другий інваріант тензора девіаторних напружень,

I_1 — перший інваріант тензора напружень,

A і B — матеріальні константи.

В'язкопластичні моделі

В'язкопластичні моделі описують поведінку ґрунтів, що мають як пластичні, так і в'язкі властивості. Вони важливі для моделювання поведінки ґрунтів під довготривалими навантаженнями.

Модель Бінгама

Ця модель використовується для опису матеріалів, що проявляють властивості як в'язкої рідини, так і пластичної твердої речовини. Вона визначає залежність між напруженням та швидкістю деформацій:

$$\tau = \tau_0 + \eta \frac{d\varepsilon}{dt}$$

де τ — дотичне напруження,

τ_0 — границя текучості,

η — коефіцієнт в'язкості,

$\frac{d\varepsilon}{dt}$ — швидкість деформації.

Консолідаційні моделі

Консолідаційні моделі описують процес витіснення порової води з ґрунту під дією навантаження, що призводить до його ущільнення. Основні рівняння консолідації базуються на рівнянні Терцагі:

$$\frac{\partial u}{\partial t} = C_v \frac{\partial^2 u}{\partial z^2}$$

де u — надлишковий поровий тиск;

t — час;

z — вертикальна координата;

C_v — коефіцієнт консолідації.

Методи на основі чисельного моделювання

Чисельні методи дозволяють розв'язувати складні задачі моделювання напружено-деформованого стану ґрунтів, які не можуть бути вирішені аналітичними методами.

Метод скінченних елементів (МСЕ)

МСЕ є одним з найбільш поширених чисельних методів для моделювання ґрунтових середовищ. Він дозволяє розбити складну геометрію на простіші елементи та розв'язувати рівняння механіки ґрунтів для кожного з них.

Основні етапи МСЕ включають:

Розбиття на скінченні елементи: розбиваємо досліджувану область на простіші елементи.

Побудова рівнянь: для кожного елемента складаються рівняння рівноваги та суміжності.

Агрегація та розв'язання: зводимо локальні рівняння в глобальну систему рівнянь та розв'язуємо її.

Метод скінченних різниць (МСР)

$$\frac{d^2u}{dx^2} \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{(\Delta x)^2}$$

МСР використовується для розв'язання диференціальних рівнянь, що описують поведінку ґрунтових середовищ. Основна ідея методу полягає у заміні похідних скінченними різницями:

Метод скінченних об'ємів (МСО)

МСО є альтернативним чисельним методом, що використовується для розв'язання рівнянь в часткових похідних. Метод полягає у розбитті області на контрольні об'єми та застосуванні закону збереження до кожного об'єму.

Метод граничних елементів (МГЕ)

МГЕ є чисельним методом, що дозволяє зменшити розмірність задачі. Він полягає у зведенні рівнянь, що описують процеси в об'ємі, до рівнянь, що описують процеси на граничній поверхні.

Спеціалізовані програмні засоби

Для моделювання напружено-деформованого стану ґрунтових середовищ використовуються спеціалізовані програмні засоби, такі як:

- PLAXIS: програмне забезпечення для геотехнічного аналізу, що базується на методі скінченних елементів.
- FLAC: програмне забезпечення для моделювання поведінки ґрунтів та порід за методом скінченних різниць.
- ABAQUS: загальне програмне забезпечення для чисельного моделювання, що також підтримує геотехнічні задачі.

Використання штучного інтелекту та машинного навчання

У сучасних дослідженнях все більшого значення набувають методи штучного інтелекту та машинного навчання для моделювання напружено-деформованого стану ґрунтів. Ці методи дозволяють аналізувати великі обсяги даних та виявляти складні закономірності, що не завжди вдається зробити за допомогою традиційних методів.

Нейронні мережі використовуються для прогнозування поведінки ґрунтів на основі навчання на експериментальних даних. Вони дозволяють будувати нелінійні моделі, що враховують взаємозв'язки між різними параметрами.

Метод опорних векторів

Метод опорних векторів використовується для класифікації та регресії. Він дозволяє будувати моделі, що оптимально розділяють дані на класи або прогнозують значення цільової змінної.

Сучасні методи моделювання напружено-деформованого стану ґрунтових середовищ включають широкий спектр підходів, що дозволяють враховувати різні фізичні властивості та умови навантаження ґрунтів. Вибір конкретного методу залежить від завдання моделювання, доступних даних та вимог до точності результатів. Використання чисельних методів, таких як метод скінченних елементів, метод скінченних різниць та метод граничних елементів, дозволяє вирішувати складні задачі, що не можуть бути розв'язані аналітичними методами. Крім того, застосування штучного інтелекту та машинного навчання відкриває нові можливості для прогнозування поведінки ґрунтів на основі великих обсягів даних.

1.3 Аналіз новітніх наукових матеріалів і розробок

Розвиток сучасних технологій та методів дослідження в галузі геотехніки призвів до появи нових підходів до моделювання напружено-деформованого стану ґрунтових середовищ. У цьому розділі проведено аналіз новітніх наукових матеріалів і розробок, які висвітлюють сучасні тенденції та інновації

в цій сфері. Огляд включає відомі наукові положення, опис робіт, що покладені в основу досліджень, та інформацію про наукові журнали з відкритим доступом. Висловлюється також авторське ставлення до існуючих поглядів і оцінка підходів до вирішення проблеми.

Одним з ключових напрямків сучасних досліджень є розробка моделей багатофазних середовищ, які враховують одночасну взаємодію твердих частинок ґрунту, води та повітря. Відомі моделі такого типу включають рівняння стану, які описують поведінку кожної з фаз і їхню взаємодію. Наприклад, модель Біота [Biot, M.A., 1941] є основою для багатьох сучасних підходів до моделювання пористих середовищ.

Ці моделі дозволяють більш точно описати процеси, що відбуваються в ґрунті під дією навантажень, включаючи консолідацію, капілярні ефекти та рух порової води. Використання таких моделей значно підвищує точність прогнозів і дозволяє врахувати складні фізичні явища, що впливають на поведінку ґрунтів.

Сучасні дослідження в області чисельних методів акцентують увагу на вдосконаленні алгоритмів, що забезпечують високу точність і ефективність обчислень. Особливо актуальними є роботи, присвячені вдосконаленню методу скінченних елементів (МСЕ) та методу дискретних елементів (МДЕ). Наприклад, в роботах Zienkiewicz et al. (2013) розглядаються новітні підходи до адаптивного МСЕ, що дозволяє автоматично підлаштовувати сітку під особливості розв'язуваної задачі.

Крім того, активно розвиваються методи мультифізичного моделювання, які дозволяють одночасно враховувати кілька фізичних процесів. Такий підхід використовується, наприклад, для моделювання взаємодії ґрунтів з водою і температурними полями (Huang et al., 2017).

Моделювання на основі машинного навчання

В останні роки машинне навчання та штучний інтелект набувають все більшого значення у геотехнічному моделюванні. Дослідники використовують нейронні мережі та інші методи машинного навчання для прогнозування

властивостей ґрунтів, аналізу стабільності схилів та інших задач. Наприклад, в роботах Zhang et al. (2018) розглядається застосування глибинних нейронних мереж для прогнозування поведінки ґрунтів під дією навантажень. Такий підхід дозволяє враховувати велику кількість параметрів і швидко адаптувати моделі до нових даних.

Однією з ключових класичних робіт є дослідження Біота (1941), яке заклало основи теорії поропружності. Це дослідження стало відправною точкою для подальшого розвитку моделей багатофазних середовищ, що враховують взаємодію твердих частинок, рідини та газу в ґрунті.

Роботи Кулона (1773) та Мора (1900) з визначенням критеріїв міцності ґрунтів також мають велике значення для сучасних досліджень. Ці критерії міцності, такі як критерій Кулона-Мора, залишаються основою для багатьох сучасних моделей напружено-деформованого стану ґрунтів.

Сучасні дослідження продовжують розвивати теоретичні основи, закладені класичними роботами. Наприклад, роботи Nicher and Shao (2008) розглядають новітні підходи до моделювання пластичності ґрунтів, включаючи теорію зміцнення та впливу мікроструктурних змін.

Інші дослідження, такі як роботи Schrefler (2012), акцентують увагу на мультифазному моделюванні і врахуванні взаємодії різних фізичних процесів. Це дозволяє точніше моделювати складні процеси, що відбуваються в ґрунтах, такі як консолидація, сейсмічні впливи та інші динамічні навантаження.

Серед наукових журналів, які публікують дослідження в галузі моделювання ґрунтових середовищ, варто виділити наступні:

Geosciences: журнал публікує дослідження з різних аспектів геонаук, включаючи моделювання ґрунтових середовищ.

Journal of Applied Geophysics: фокусується на застосуванні геофізичних методів, включаючи моделювання напружено-деформованого стану ґрунтів.

Geoscience Frontiers: журнал охоплює широкий спектр геонаукових досліджень, включаючи новітні методи моделювання ґрунтових середовищ.

Ці журнали надають доступ до новітніх досліджень та розробок, що дозволяє вченим і інженерам залишатися в курсі останніх тенденцій і інновацій у галузі.

Існуючі погляди на моделювання напружено-деформованого стану ґрунтових середовищ відображають високу складність і багатогранність цієї проблеми. Класичні теорії, такі як поропружність Біота, залишаються основою для багатьох сучасних моделей. Проте, існують значні відмінності у підходах до моделювання залежно від специфіки задачі та доступних даних.

Сучасні підходи, що використовують чисельні методи, машинне навчання та мультифізичне моделювання, дозволяють значно підвищити точність і ефективність прогнозів. Проте, вони також потребують значних обчислювальних ресурсів і глибоких знань у відповідних галузях.

На мою думку, подальший розвиток моделювання напружено-деформованого стану ґрунтових середовищ повинен включати інтеграцію класичних теорій з новітніми методами чисельного моделювання та машинного навчання. Це дозволить створити більш точні і універсальні моделі, які враховують всі важливі аспекти поведінки ґрунтів.

Існуючі підходи до моделювання напружено-деформованого стану ґрунтових середовищ мають свої переваги і недоліки. Метод скінченних елементів (МСЕ) та метод дискретних елементів (МДЕ) є найбільш поширеними і забезпечують високу точність розрахунків для різних типів задач. Проте, вони потребують значних обчислювальних ресурсів і складних налаштувань.

Машинне навчання та нейронні мережі відкривають нові можливості для моделювання складних процесів у ґрунтах. Ці методи дозволяють швидко аналізувати великі обсяги даних і адаптувати моделі до нових умов. Проте, вони також вимагають наявності великих наборів даних для навчання і належної кваліфікації для налаштування моделей.

Комбіновані підходи, які інтегрують кілька різних методів, забезпечують найбільшу гнучкість і точність. Вони дозволяють враховувати взаємодію

різних фізичних процесів і адаптувати моделі до специфічних умов задачі. Проте, такі підходи є найскладнішими у реалізації і потребують високої кваліфікації і значних обчислювальних ресурсів.

Аналіз новітніх наукових матеріалів і розробок показує, що моделювання напружено-деформованого стану ґрунтових середовищ є складною і багатогранною задачею. Існуючі підходи, включаючи чисельні методи, машинне навчання та мультифізичне моделювання, забезпечують високу точність і ефективність розрахунків. Проте, вони також потребують значних обчислювальних ресурсів і спеціальних знань.

Подальший розвиток моделювання повинен включати інтеграцію класичних теорій з новітніми методами, що дозволить створити більш точні і універсальні моделі. Важливо також забезпечити доступ до новітніх досліджень через наукові журнали з відкритим доступом, що дозволить інженерам і науковцям залишатися в курсі останніх тенденцій і інновацій у галузі.

Сучасні дослідження і розробки відкривають нові можливості для моделювання напружено-деформованого стану ґрунтових середовищ, що сприятиме підвищенню безпеки та ефективності будівельних і гірничих робіт. Інтеграція різних підходів і використання новітніх технологій дозволить створити більш точні і надійні моделі, що враховують всі важливі аспекти поведінки ґрунтів.

1.4 Оцінка існуючих підходів до вирішення проблеми

Моделювання напружено-деформованого стану ґрунтових середовищ є складним і багатоаспектним завданням. Існують різні методи, які використовуються для вирішення цієї проблеми, кожен з яких має свої переваги та недоліки. У цьому розділі проведемо огляд літератури, аналіз ефективності методів, порівняємо різні підходи та ідентифікуємо прогалини і можливості для вдосконалення.

Метод числового моделювання є одним з найбільш поширених підходів до вирішення задач моделювання напружено-деформованого стану ґрунтових середовищ. В основному використовуються метод скінченних елементів (МСЕ), метод скінченних різниць (МСР) та метод дискретних елементів (МДЕ).

Метод скінченних елементів (МСЕ): Відзначається високою точністю і гнучкістю, дозволяє моделювати складні геометрії та неоднорідності ґрунтів (Zienkiewicz et al., 2013). Проте, потребує значних обчислювальних ресурсів.

Метод скінченних різниць (МСР): Простіший у реалізації, менш ресурсомісткий, проте менш точний для складних геометрій (Smith, 2014).

Метод дискретних елементів (МДЕ): Застосовується для моделювання гранульованих матеріалів, дозволяє враховувати індивідуальні властивості частинок (Cundall & Strack, 1979). Потребує великих обчислювальних ресурсів для великих систем.

Машинне навчання стає все більш популярним у моделюванні ґрунтових середовищ завдяки своїй здатності швидко аналізувати великі обсяги даних та адаптуватися до нових умов.

Нейронні мережі використовуються для прогнозування властивостей ґрунтів і поведінки під дією навантажень (Zhang et al., 2018). Потребують великого обсягу навчальних даних.

Глибинне навчання: Забезпечує високу точність, проте є складним у налаштуванні та вимагає значних обчислювальних ресурсів (LeCun et al., 2015).

Емпіричні моделі базуються на статистичних аналізах і спостереженнях, що дозволяє швидко отримувати результати без великих обчислювальних витрат. Проте, їх точність залежить від якості вихідних даних та обмежується умовами, в яких ці дані були отримані.

Аналіз ефективності методів

Метод скінченних елементів (МСЕ) забезпечує високу точність та гнучкість у моделюванні. Він дозволяє детально враховувати геометричні та

матеріальні особливості ґрунтових середовищ. Однак, обчислювальні витрати є значним обмеженням, особливо для великих систем або задач з високою динамікою процесів (Zienkiewicz et al., 2013).

Метод скінченних різниць (МСР) є менш ресурсомістким і простішим у реалізації. Він добре підходить для задач зі стабільними і простими геометріями. Проте, його точність значно поступається МСЕ, особливо для складних геометрій та умов (Smith, 2014).

Метод дискретних елементів (МДЕ) дозволяє моделювати поведінку гранульованих матеріалів з високою точністю. Цей метод особливо ефективний для задач, де важливі локальні взаємодії частинок. Недоліком є висока обчислювальна вартість, яка збільшується з кількістю частинок у системі (Cundall & Strack, 1979).

Машинне навчання та глибинні нейронні мережі забезпечують можливість швидкої адаптації до нових даних та високої точності прогнозів. Проте, ці методи потребують великого обсягу навчальних даних та обчислювальних ресурсів для навчання моделей. Вони також можуть бути складними у налаштуванні та валідації (Zhang et al., 2018; LeCun et al., 2015).

Емпіричні моделі є швидкими у використанні та не потребують значних обчислювальних ресурсів. Проте, їх точність обмежена якістю вихідних даних і вони можуть бути непридатні для умов, відмінних від тих, де були отримані дані для їх побудови (Sivakugan et al., 2011).

Порівняння методів

Метод	Точність	Обчислювальні витрати	Гнучкість	Складність реалізації
Метод скінченних елементів	Висока	Високі	Висока	Висока
Метод скінченних різниць	Середня	Низькі	Середня	Низька
Метод дискретних елементів	Висока	Високі	Висока	Висока
Машинне навчання	Висока	Високі	Висока	Висока
Емпіричні моделі	Низька	Низькі	Низька	Низька

Ідентифікація проблем та можливостей для вдосконалення

Існуючі методи моделювання мають значні досягнення, проте кожен з них має свої обмеження. Жоден з методів не забезпечує універсальності і достатньої точності у всіх можливих умовах.

Проблеми в числових методах

- Обчислювальні витрати: Багато числових методів, особливо МСЕ і МДЕ, потребують значних обчислювальних ресурсів, що може бути обмеженням для великих або дуже детальних моделей.
- Складність реалізації: Ці методи вимагають високих знань та досвіду для правильного налаштування моделей і інтерпретації результатів.

Проблеми в методах машинного навчання

- Необхідність великих даних: Методи машинного навчання потребують великого обсягу навчальних даних, що може бути проблемою для специфічних задач або нових умов.
- Чорний ящик: Результати машинного навчання можуть бути складними для інтерпретації, оскільки внутрішні механізми моделей часто залишаються непрозорими.

Проблеми в емпіричних моделях

- **Обмежена застосовність:** Емпіричні моделі можуть бути точними лише в умовах, схожих на ті, де були отримані вихідні дані. Вони не завжди добре переносяться на нові або відмінні умови.

Можливості для вдосконалення

Інтеграція числових методів і машинного навчання

Поєднання числових методів і машинного навчання може значно підвищити точність і ефективність моделювання. Наприклад, числові моделі можуть використовуватися для генерації навчальних даних для моделей машинного навчання, що дозволить зменшити необхідність у великому обсязі емпіричних даних.

Гібридні підходи

Розробка гібридних підходів, які поєднують різні методи, може забезпечити більшу гнучкість і точність. Наприклад, використання MSE для основного моделювання і методів машинного навчання для корекції і адаптації моделей під нові умови.

Розвиток обчислювальних технологій

Подальший розвиток обчислювальних технологій, таких як паралельні обчислення та використання графічних процесорів (GPU), може значно зменшити обчислювальні витрати і зробити числові методи більш доступними для широкого використання.

Підвищення інтерпретації результатів

Розробка методів для покращення інтерпретації результатів моделей машинного навчання, таких як пояснювані нейронні мережі (explainable neural networks), може підвищити їх прийнятність і довіру до них.

Оцінка існуючих підходів показує, що кожен метод має свої сильні та слабкі сторони. Числові методи забезпечують високу точність, але вимагають значних обчислювальних ресурсів і складні у реалізації. Методи машинного навчання є перспективними завдяки своїй здатності швидко адаптуватися до нових даних, проте потребують великого обсягу навчальної інформації та

складні у налаштуванні. Емпіричні моделі є швидкими і простими у використанні, але їх точність обмежується якістю вихідних даних.

Подальший розвиток методів моделювання має включати інтеграцію різних підходів для досягнення більшої точності і гнучкості. Використання гібридних методів, що поєднують числове моделювання і машинне навчання, а також розвиток обчислювальних технологій і методів пояснення моделей машинного навчання, може значно покращити результати і розширити можливості застосування моделей.

Це забезпечить більш надійні прогнози і підвищить безпеку та ефективність будівельних та гірничих робіт, що є критично важливим для сучасного інженерного проектування і управління ризиками.

РОЗДІЛ 2

МЕТОДОЛОГІЯ ДОСЛІДЖЕННЯ

2.1 Опис програми GEO5

GEO5 є потужним програмним комплексом, спеціалізованим на геотехнічному моделюванні та аналізі. Цей інструмент широко використовується для розрахунків та проектування різних типів геотехнічних конструкцій. У цьому підрозділі розглянемо функціональні можливості GEO5, основні модулі програми, інтерфейс користувача та методи обчислень, що використовуються в програмі.

Функціональні можливості GEO5

GEO5 надає широкий спектр функціональних можливостей для геотехнічного аналізу:

1. Розрахунок стабільності схилів: Програма дозволяє оцінювати стабільність природних та штучних схилів з урахуванням різних типів навантажень.
2. Аналіз фундаментів: Включає розрахунки несучої здатності та осідань різних типів фундаментів, таких як стрічкові, плитні та стовпчасті фундаменти.
3. Проектування підпірних стін: GEO5 дозволяє проектувати підпірні стіни різних типів та проводити аналіз їх стійкості.
4. Моделювання тунелів та котлованів: Програма включає інструменти для аналізу тунелів, котлованів та інших підземних споруд.
5. Геоекологічні розрахунки: Включає модулі для оцінки впливу будівельних робіт на довкілля.

Основні модулі програми

GEO5 складається з кількох модулів, кожен з яких спеціалізується на конкретному типі аналізу:

1. Slope Stability: Модуль для розрахунку стабільності схилів.
2. Spread Footing: Модуль для розрахунку несучої здатності та осідань стрічкових фундаментів.

3. Retaining Wall: Модуль для проектування та аналізу підпірних стін.
4. Pile Group: Модуль для розрахунку груп пальових фундаментів.
5. Tunnel: Модуль для аналізу тунелів та підземних споруд.

Кожен модуль інтегрується з іншими, що дозволяє проводити комплексний аналіз геотехнічних задач.

Інтерфейс користувача

Інтерфейс GEO5 є інтуїтивно зрозумілим та зручним для користувача.

Він складається з наступних основних компонентів:

Головне меню: Містить усі необхідні інструменти та опції для налаштування розрахунків.

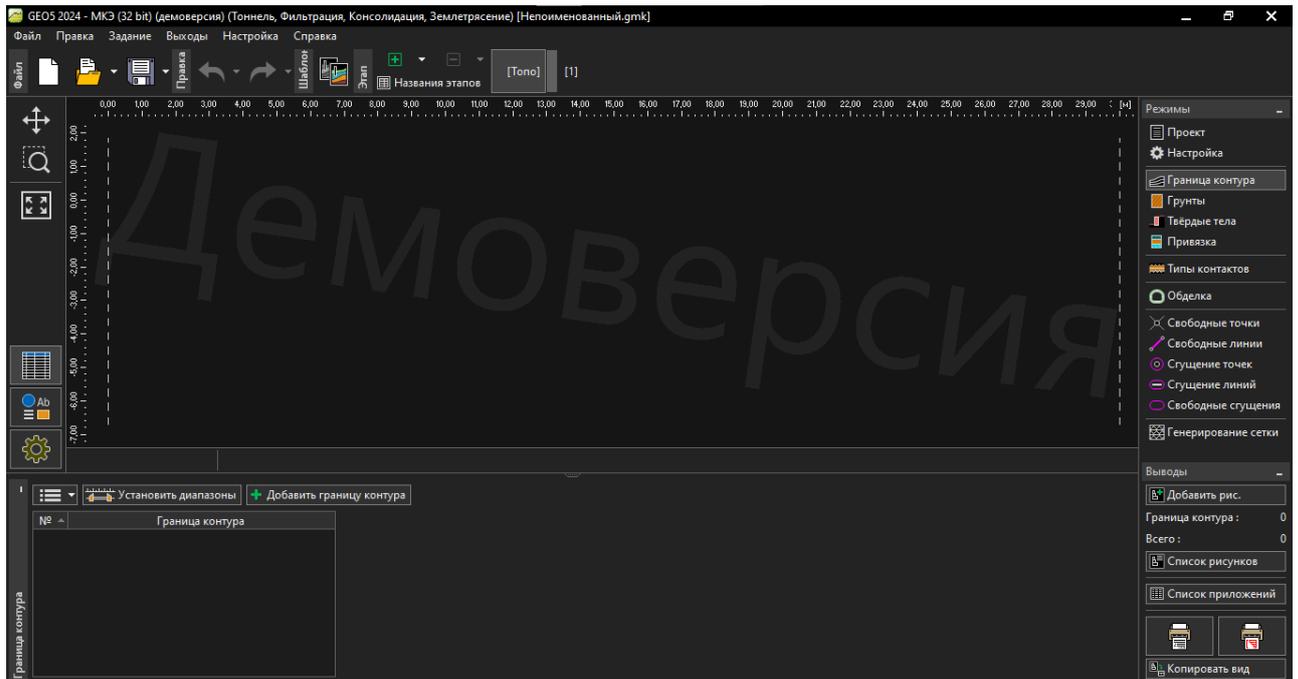


Рис. 2.1.1 Головне меню GEO5

Робоче поле: Відображає моделі, креслення та результати розрахунків.

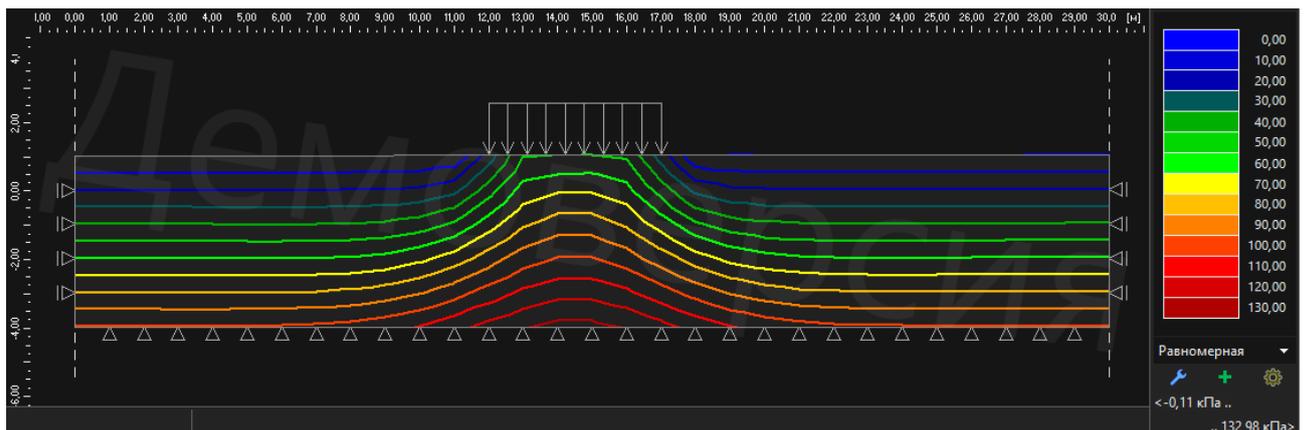


Рис. 2.1.2 Робоче поле GEO5

Панель інструментів: Містить часто використовувані команди для швидкого доступу.

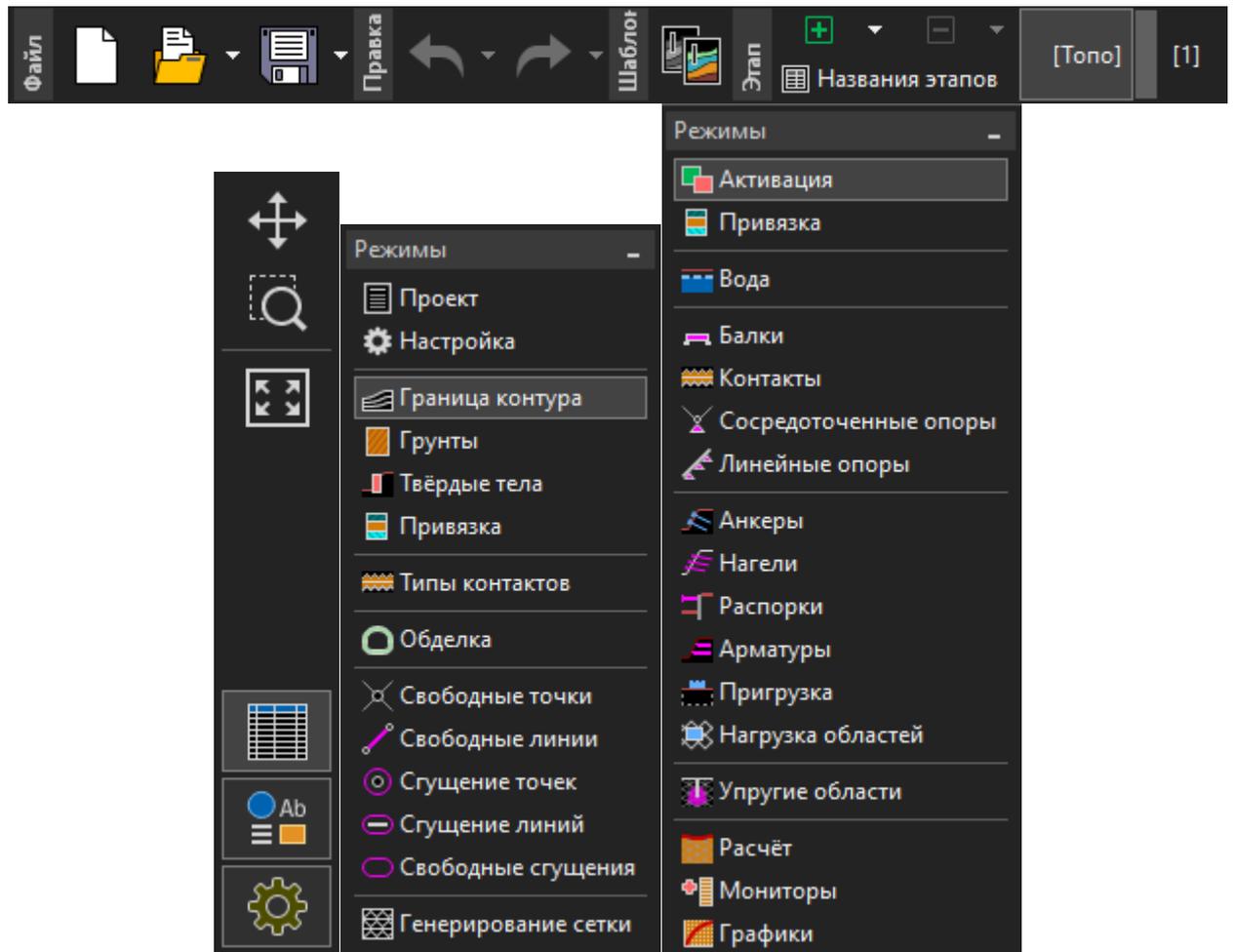


Рис. 2.1.3 Панель інструментів GEO5

Вікно властивостей: Дозволяє налаштувати параметри моделей та обчислень.

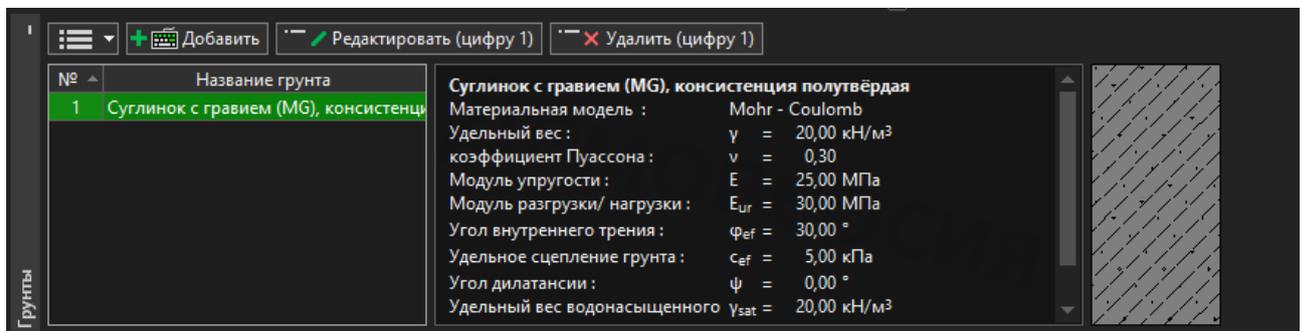


Рис. 2.1.4 Вікно властивостей GEO5

Програма підтримує багатомовний інтерфейс, що робить її доступною для користувачів з різних країн.

Методи обчислень в GEO5

GEO5 використовує сучасні методи чисельного моделювання та аналізу для забезпечення високої точності та надійності результатів. Серед основних методів:

- Метод кінцевих елементів (МКЕ): Використовується для детального моделювання напружено-деформованого стану ґрунтових середовищ та конструкцій.
- Лімітний рівноважний метод (LEM): Застосовується для оцінки стабільності схилів та підпірних стін.
- Аналітичні методи: Включають класичні аналітичні підходи для розрахунків фундаментів та інших конструкцій.
- Емпіричні методи: Використовуються для швидкої оцінки та проектування на основі накопиченого досвіду та даних.

GEO5 також підтримує можливість імпорту та експорту даних у різних форматах, що дозволяє інтегрувати програму з іншими інженерними та проектними інструментами.

Доступність програми GEO5

GEO5 – це потужний програмний комплекс для геотехнічного моделювання, який розробила компанія Fine Software. Програма доступна для придбання через різні типи ліцензій, що дозволяє користувачам обрати найбільш підходящий варіант відповідно до їх потреб.

Ліцензії:

GEO5 пропонує як постійні, так і тимчасові ліцензії. Існує можливість придбання окремих модулів або повного комплексу, що дозволяє гнучко налаштовувати програму під конкретні задачі.

Вартість залежить від обраного типу ліцензії та кількості модулів. Є знижки для освітніх установ та великих організацій.

Пробні версії та демонстраційні можливості

GEO5 пропонує демонстраційні версії, що дозволяють користувачам ознайомитися з функціональністю програми. Ці демо-версії мають обмежений

функціонал, але достатні для початкової оцінки можливостей програмного комплексу. Завантажити пробну версію можна безпосередньо з офіційного сайту Fine Software.

GEO5 є потужним та гнучким інструментом для геотехнічного моделювання та аналізу. Він надає широкий спектр функцій для вирішення різноманітних інженерних задач, що робить його незамінним у практиці геотехнічних досліджень та проектування. Інтуїтивно зрозумілий інтерфейс, сучасні методи обчислень та модульна структура забезпечують високу продуктивність та зручність використання програми.

2.2 Опис програми RocScience

RocScience є однією з провідних програмних платформ для геотехнічного моделювання та аналізу. Вона використовується інженерами для вирішення складних задач в галузі геотехніки, таких як аналіз стійкості схилів, проектування підпірних стін, моделювання тунелів та інших підземних споруд.

Функціональні можливості RocScience

RocScience пропонує широкий спектр інструментів для геотехнічного аналізу, що охоплюють різні аспекти проектування та моделювання:

1. Аналіз стійкості схилів: Програма дозволяє проводити детальний аналіз стійкості схилів за допомогою різних методів, включаючи метод скінченних елементів та лімітні рівноважні методи.
2. Проектування фундаментів: RocScience забезпечує інструменти для розрахунку несучої здатності та осідань різних типів фундаментів, включаючи стрічкові та плитні фундаменти.
3. Моделювання підпірних стін: Інструменти для аналізу та проектування підпірних стін різних типів, включаючи гравітаційні, анкерові та комбіновані стіни.
4. Аналіз тунелів та підземних споруд: Модулі для моделювання та аналізу тунелів, включаючи розрахунок напружень та деформацій у навколишньому ґрунті.

5. Динамічний аналіз: RocScience підтримує аналіз впливу сейсмічних навантажень та інших динамічних впливів на геотехнічні конструкції.

Основні модулі програми

RocScience складається з кількох спеціалізованих модулів, кожен з яких призначений для вирішення конкретних типів геотехнічних задач:

1. Slide: Модуль для аналізу стійкості схилів з використанням різних методів, включаючи метод скінченних елементів та лімітні рівноважні методи.
2. RS2 (Phase2): Потужний модуль для двовимірного аналізу методом скінченних елементів, що дозволяє моделювати напружено-деформований стан ґрунтів та конструкцій.
3. RS3: Тривимірний аналог RS2 для детального тривимірного моделювання та аналізу геотехнічних задач.
4. Dips: Модуль для статистичного аналізу структурних даних, включаючи аналіз тріщинуватості та схилових масивів.
5. RocSupport: Інструмент для проектування тунелів, що дозволяє моделювати взаємодію тунельної обробки з навколишнім ґрунтом.
6. Settle3D: Модуль для тривимірного аналізу осідань та консолидації ґрунтів під дією навантажень.,

Інтерфейс користувача

Інтерфейс RocScience є зручним та інтуїтивно зрозумілим, що забезпечує легкість використання та високу продуктивність роботи:

Головне меню: Надає доступ до всіх основних функцій та інструментів програми.

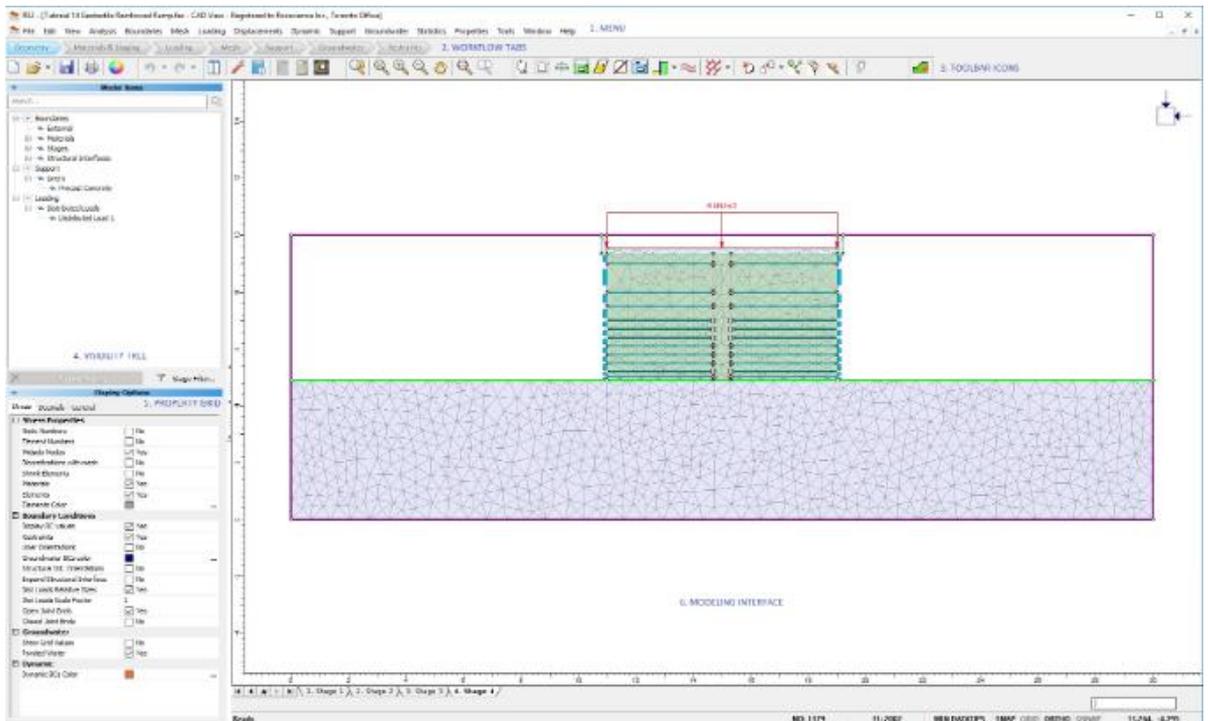


Рис. 2.2.1 Головне меню RocScience

Робоче поле: Відображає моделі, графіки та результати розрахунків.

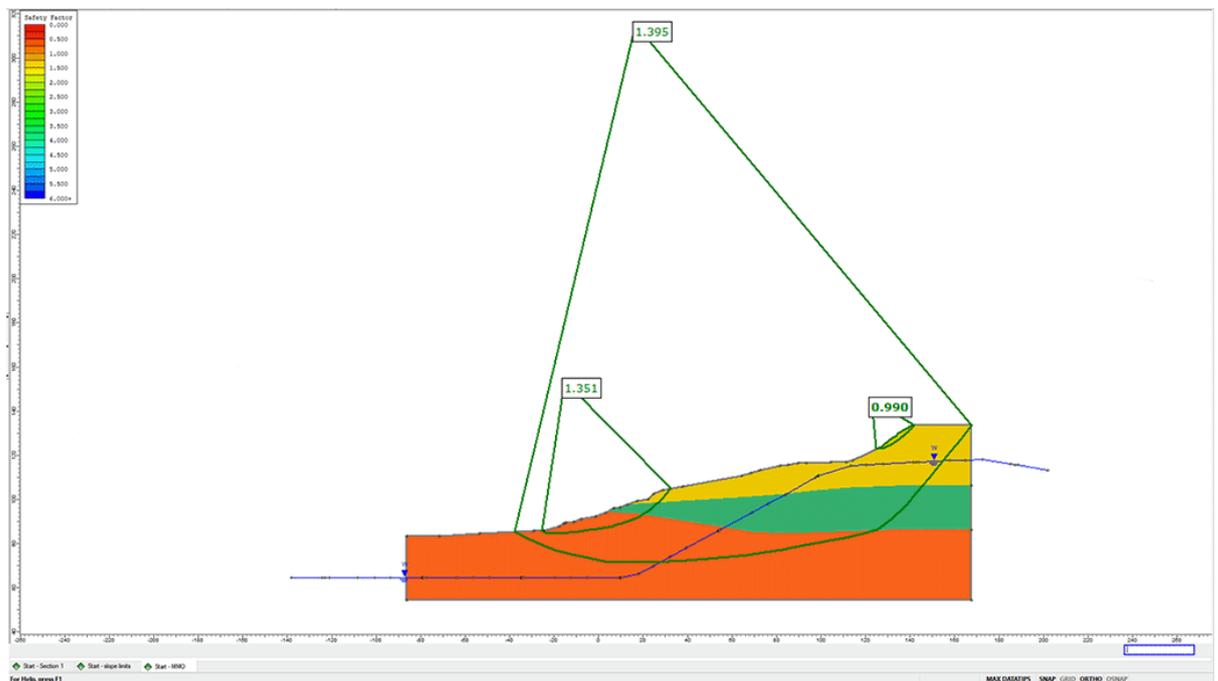


Рис. 2.2.2 Робоче поле RocScience

Панель інструментів: Включає часто використовувані команди для швидкого доступу.



Рис. 2.2.3 Робоче поле RocScience

Вікно властивостей: Дозволяє налаштовувати параметри моделей, методів розрахунку та візуалізації.

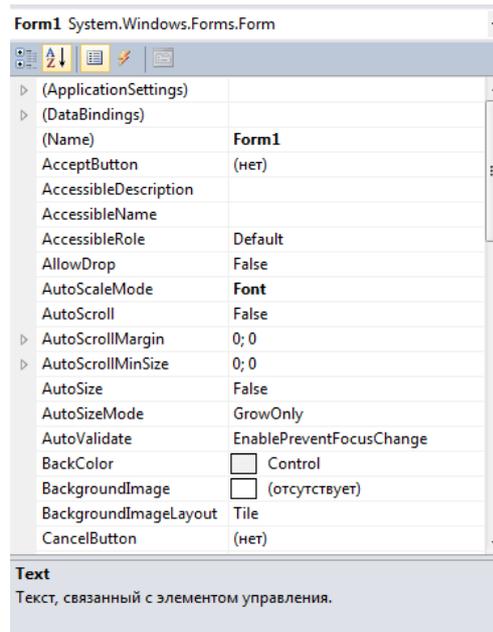


Рис. 2.2.4 Вікно властивостей RocScience

Програма підтримує багатомовний інтерфейс та має добре організовану документацію, що полегшує навчання та використання.

Методи обчислень в RocScience

RocScience використовує сучасні чисельні методи та алгоритми для забезпечення високої точності та надійності результатів:

- Метод скінченних елементів (МСЕ): Використовується для детального двовимірного та тривимірного моделювання напружено-деформованого стану ґрунтових середовищ та конструкцій.
- Лімітні рівноважні методи (LEM): Застосовуються для оцінки стійкості схилів та підпірних стін, дозволяючи проводити швидкий та надійний аналіз.
- Аналітичні методи: Використовуються для швидких оцінок та розрахунків, особливо на початкових етапах проектування.
- Динамічний аналіз: Включає методи моделювання впливу динамічних навантажень, таких як сейсмічні дії, на геотехнічні конструкції.

RocScience також підтримує інтеграцію з іншими інженерними та геотехнічними інструментами через імпорт та експорт даних у різних форматах.

Доступність програми RocScience

RocScience – інший провідний програмний комплекс для геотехнічного аналізу, розроблений компанією RocScience Inc. Програма доступна через різні типи ліцензій, включаючи індивідуальні та мережеві.

Ліцензії:

RocScience пропонує постійні та тимчасові ліцензії для комерційного та академічного використання. Доступні як індивідуальні, так і мережеві ліцензії, що дозволяє використовувати програму в командній роботі.

Вартість залежить від обраного пакету та модулів. Існує система знижок для освітніх установ та великих організацій.

Пробні версії та демонстраційні можливості

RocScience пропонує пробні версії, які дозволяють користувачам оцінити функціональність програми перед придбанням. Пробні версії зазвичай мають обмежений термін використання (наприклад, 30 днів), але надають повний доступ до всіх функцій. Це дає можливість повноцінно випробувати програмний комплекс у реальних умовах.

RocScience є потужним інструментом для геотехнічного аналізу та моделювання, що надає широкий спектр функцій та можливостей для інженерів. Зручний інтерфейс, сучасні методи обчислень та модульна структура забезпечують високу точність, надійність та продуктивність роботи. Завдяки своїм можливостям RocScience є незамінним інструментом для вирішення складних геотехнічних задач у проектуванні та аналізі інженерних споруд.

2.3 Аналіз та порівняння програм GEO5 та RocScience

GEO5 та RocScience є провідними програмними комплексами, що використовуються в геотехнічному моделюванні та аналізі. Кожна з цих програм має свої унікальні особливості, які роблять їх корисними для певних

задач та умов. Проведемо детальний аналіз та порівняння GEO5 та RocScience, оцінюючи їхні можливості, ефективність, зручність використання та інші важливі аспекти.

Функціональні можливості

GEO5

GEO5 розроблено для широкого спектру геотехнічних задач, таких як аналіз стабільності схилів, розрахунки фундаментів, проектування підпірних стін, моделювання тунелів та геоекологічні розрахунки. Програма надає інтуїтивно зрозумілий інтерфейс та модульну структуру, що дозволяє користувачам вибирати тільки ті інструменти, які потрібні для конкретного проекту.

- Розрахунок стабільності схилів: Включає кілька методів аналізу, таких як метод скінченних елементів (МКЕ) та лімітний рівноважний метод (LEM).
- Аналіз фундаментів: Модулі для стрічкових, плитних та стовпчастих фундаментів забезпечують детальні розрахунки несучої здатності та осідань.
- Проектування підпірних стін: Інструменти для аналізу гравітаційних та анкерових стін з урахуванням різних типів навантажень.
- Моделювання тунелів: Спеціалізовані модулі для аналізу тунелів та котлованів.
- Геоекологічні розрахунки: Інструменти для оцінки впливу будівельних робіт на довкілля.

RocScience

RocScience надає потужні інструменти для складного геотехнічного аналізу, зокрема для аналізу стійкості схилів, проектування фундаментів, підпірних стін та тунелів, а також для динамічного аналізу. Програма включає модулі, які підтримують як двовимірне, так і тривимірне моделювання.

- Slide: Аналіз стійкості схилів за допомогою різних методів, включаючи метод скінченних елементів.

- RS2 (Phase2): Двовимірний аналіз методом скінченних елементів для моделювання напружено-деформованого стану ґрунтів та конструкцій.
- RS3: Тривимірний аналог RS2 для детального тривимірного моделювання.
- Dips: Статистичний аналіз структурних даних, включаючи аналіз тріщинуватості та схилових масивів.
- RocSupport: Проектування тунелів з урахуванням взаємодії тунельної обробки з навколишнім ґрунтом.
- Settle3D: Тривимірний аналіз осідань та консолидації ґрунтів під дією навантажень.

Точність розрахунків

Обидві програми забезпечують високу точність розрахунків завдяки використанню сучасних чисельних методів, таких як метод скінченних елементів (MCE) та лімітні рівноважні методи (LEM). Проте, RocScience, особливо з модулями RS2 та RS3, пропонує більш детальні можливості для тривимірного моделювання, що може бути критично важливим для складних геотехнічних задач.

Обчислювальні витрати

RocScience, завдяки своїм потужним модулям для тривимірного моделювання, може вимагати значних обчислювальних ресурсів. Це може бути обмеженням для користувачів з менш потужними комп'ютерами або обмеженими ресурсами. У той час як GEO5 є більш оптимізованим для двовимірних задач і може бути більш ефективним у використанні обчислювальних ресурсів для менш складних аналізів.

Зручність використання

GEO5 має інтуїтивно зрозумілий інтерфейс, який дозволяє швидко навчатися та ефективно використовувати програму навіть новачкам. Модульна структура дозволяє користувачам легко обирати та використовувати лише

необхідні інструменти, що робить програму дуже гнучкою та зручною у використанні.

RocScience також пропонує зручний інтерфейс, але через більшу кількість функцій і можливостей, програма може здатися більш складною для новачків. Проте, для досвідчених користувачів вона пропонує багатий набір інструментів для виконання складних задач.

Підтримка користувачів та документація

Обидві програми пропонують хорошу підтримку користувачів та мають обширну документацію. GEO5 надає детальні керівництва та відеоуроки, що допомагають швидко освоїти програму. RocScience також забезпечує користувачів докладною документацією, вебінарами та технічною підтримкою, що є дуже корисним для вирішення складних технічних питань.

Інтеграція з іншими системами

RocScience забезпечує більш широкі можливості для інтеграції з іншими інженерними та проектними інструментами через підтримку різних форматів імпорту та експорту даних. Це може бути важливо для користувачів, які працюють з великими проектами, що вимагають інтеграції різних видів даних.

GEO5 також підтримує інтеграцію з іншими системами, але її можливості в цьому відношенні можуть бути дещо обмеженими порівняно з RocScience.

Отже, обидві програми, GEO5 та RocScience, є потужними інструментами для геотехнічного моделювання та аналізу. GEO5 є більш зручним для новачків і оптимізованим для двовимірних задач, що робить його ідеальним вибором для менш складних проектів. RocScience, з іншого боку, пропонує потужніші можливості для тривимірного моделювання та більш детальний аналіз, що робить його незамінним для складних геотехнічних задач.

Вибір між GEO5 та RocScience залежить від конкретних потреб проекту та доступних ресурсів. Для більшості стандартних задач GEO5 буде достатнім і зручним вибором. Для складніших задач, які потребують детального

тривимірного моделювання та аналізу, RocScience є більш підходящим варіантом.

2.4 Вибір мови та засобів програмування для реалізації задачі

Вибір мови програмування та інструментів є критично важливим етапом у розробці програмного продукту. У цьому підрозділі розглядаються різні мови програмування, їхні переваги та недоліки, а також вибір найбільш підходящих інструментів для реалізації задачі.

Для вибору мови програмування розглянемо кілька найбільш популярних мов, які часто використовуються в наукових обчисленнях та розробці інженерного програмного забезпечення.

Python

Python є однією з найпопулярніших мов програмування в наукових дослідженнях та інженерії завдяки своїй простоті, гнучкості та великій кількості доступних бібліотек.

Переваги:

Легкість у навчанні та використанні.

- Велика кількість бібліотек для чисельного моделювання (NumPy, SciPy), візуалізації (Matplotlib), аналізу даних (Pandas) та машинного навчання (scikit-learn).
- Широка підтримка та активна спільнота.

Недоліки:

- Повільніше виконання порівняно з компільованими мовами (наприклад, C++).

C++

C++ є високопродуктивною мовою програмування, що забезпечує швидке виконання та ефективне управління пам'яттю, що важливо для обчислювальних задач.

Переваги:

- Висока продуктивність.

- Ефективне управління ресурсами.
- Підтримка об'єктно-орієнтованого програмування.

Недоліки:

- Складність у навчанні та використанні.
- Більш тривалий процес розробки.

MATLAB

MATLAB є спеціалізованою платформою для чисельних обчислень, яка широко використовується в наукових дослідженнях та інженерії.

Переваги:

- Потужні інструменти для чисельного аналізу та візуалізації даних.
- Велика кількість вбудованих функцій та бібліотек.
- Зручність використання для прототипування.

Недоліки:

- Висока вартість ліцензії.
- Обмежена продуктивність у порівнянні з компільованими мовами.

Враховуючи переваги та недоліки розглянутих мов програмування, для реалізації задачі було обрано Python. Цей вибір обґрунтований наступними причинами:

Python має простий і зрозумілий синтаксис, що дозволяє швидко розробляти та тестувати моделі.

Велика кількість бібліотек забезпечує підтримку всіх необхідних функцій для чисельного моделювання, аналізу даних та візуалізації.

Широка підтримка з боку спільноти та велика кількість навчальних ресурсів дозволяють швидко знаходити вирішення проблем і отримувати допомогу.

Для реалізації задачі чисельного моделювання будуть використані наступні бібліотеки Python:

NumPy

NumPy є основною бібліотекою для роботи з багатовимірними масивами та матрицями. Вона забезпечує високу продуктивність завдяки використанню оптимізованих алгоритмів.

Основні можливості:

- Операції з масивами та матрицями.
- Математичні функції високого рівня.
- Інтеграція з іншими бібліотеками.

SciPy

SciPy розширює можливості NumPy і включає модулі для чисельного інтегрування, оптимізації, розв'язання диференціальних рівнянь та інших наукових обчислень.

Основні можливості:

- Оптимізація та розв'язання рівнянь.
- Чисельне інтегрування.
- Робота з сигналами та обробка зображень.

Matplotlib

Matplotlib є потужною бібліотекою для створення графіків та візуалізації даних. Вона дозволяє створювати як прості графіки, так і складні візуалізації.

Основні можливості:

- Створення лінійних та нелінійних графіків.
- Візуалізація даних у вигляді 2D графіків.
- Налаштування та кастомізація графіків.

Pandas

Pandas забезпечує зручні засоби для роботи з табличними даними, що робить його незамінним для аналізу та маніпуляції даними.

Основні можливості:

- Робота з даними у форматі таблиць.
- Аналіз та маніпуляція даними.
- Інтеграція з іншими бібліотеками.

scikit-learn

scikit-learn є бібліотекою для машинного навчання, яка надає інструменти для класифікації, регресії, кластеризації та інших задач машинного навчання.

Основні можливості:

- Алгоритми класифікації та регресії.
- Алгоритми кластеризації.
- Підтримка попередньої обробки даних.

Крім бібліотек для чисельного моделювання, будуть використані додаткові інструменти для візуалізації та аналізу даних:

Jupyter Notebook

Jupyter Notebook є інтерактивним середовищем для виконання коду, що дозволяє поєднувати код, текст, графіку та інші елементи в одному документі. Це зручний інструмент для прототипування та аналізу даних.

Основні можливості:

- Виконання коду в інтерактивному режимі.
- Візуалізація результатів у вигляді графіків та таблиць.
- Інтерактивне документування процесу розробки.

Для розробки програмного забезпечення було обрано мову програмування Python та відповідні бібліотеки. Python забезпечує високу продуктивність, гнучкість та багатофункціональність, що робить його найкращим вибором для даного проекту. Використання бібліотек NumPy, SciPy, Matplotlib, Pandas та scikit-learn дозволить ефективно реалізувати математичну модель, виконати чисельні обчислення та візуалізувати результати у вигляді 2D графіків.

2.5 Вибір архітектури програмного продукту

Вибір архітектури програмного продукту є критично важливим для забезпечення ефективної реалізації задачі моделювання напружено-деформованого стану ґрунтових середовищ. Розглянемо основні архітектурні

підходи та обґрунтуємо вибір найбільш підходящої архітектури для даного проекту.

Основні архітектурні підходи

Монолітна архітектура

Монолітна архітектура передбачає створення єдиного цілісного програмного продукту, де всі компоненти тісно пов'язані між собою. Цей підхід простий у реалізації та забезпечує високу продуктивність, проте має недоліки, пов'язані з масштабованістю та складністю підтримки.

Переваги:

- Простота розробки.
- Висока продуктивність.

Недоліки:

- Складність підтримки та оновлення.
- Важкість масштабування.

Мікросервісна архітектура

Мікросервісна архітектура передбачає розбиття програми на незалежні сервіси, кожен з яких виконує окрему функцію. Це забезпечує гнучкість, масштабованість і легкість підтримки, але вимагає складнішого налаштування та управління.

Переваги:

- Легкість масштабування.
- Гнучкість та модульність.
- Простота підтримки та оновлення.

Недоліки:

- Складність налаштування.
- Вищі вимоги до управління інфраструктурою.

Орієнтована на події архітектура

Орієнтована на події архітектура базується на обробці подій, що генеруються різними компонентами системи. Цей підхід дозволяє ефективно

реагувати на зміни та забезпечує високу гнучкість, проте потребує ретельного планування та обробки подій.

Переваги:

- Висока гнучкість.
- Можливість швидкої обробки подій.

Недоліки:

- Складність реалізації та тестування.
- Вимоги до точності обробки подій.

Орієнтована на сервіси архітектура (SOA)

SOA передбачає створення системи, де сервіси взаємодіють через визначені інтерфейси. Це дозволяє забезпечити гнучкість та повторне використання компонентів, проте може бути складним у реалізації та управлінні.

Переваги:

- Гнучкість та повторне використання компонентів.
- Легкість інтеграції з іншими системами.

Недоліки:

- Складність реалізації.
- Високі вимоги до управління сервісами.

Для задачі моделювання та обчислення найбільш підходящою є мікросервісна архітектура. Цей вибір обґрунтований наступними причинами:

- Масштабованість: Мікросервісна архітектура дозволяє легко масштабувати окремі сервіси відповідно до навантаження, що особливо важливо для обчислювальних задач.

- Гнучкість: Кожен мікросервіс може бути розроблений, розгорнутий та оновлений незалежно від інших, що забезпечує гнучкість у розробці та підтримці.

- Модульність: Легкість інтеграції нових компонентів і можливість повторного використання існуючих модулів.

- Надійність: Збої в одному сервісі не впливають на роботу інших, що підвищує загальну надійність системи.

Вибір мікросервісної архітектури забезпечує найбільшу відповідність вимогам задачі. Вона дозволяє досягти високої гнучкості, масштабованості та надійності, що є критичними для успішної реалізації програмного продукту.

2.6 Вибір оптимальної моделі для дослідження

Для моделювання напружено-деформованого стану ґрунту необхідно вибрати оптимальну модель, яка забезпечить високу точність результатів та ефективність обчислень. Враховуючи специфіку задачі, а також доступні інструменти та ресурси, було обрано модель, яка найкраще відповідає цим критеріям.

Критерії вибору моделі

При виборі моделі для дослідження враховувались наступні критерії:

- Точність: Модель повинна забезпечувати високу точність у прогнозуванні напружено-деформованого стану ґрунтів.
- Обчислювальна ефективність: Модель повинна бути ефективною з точки зору обчислювальних ресурсів та часу виконання.
- Сумісність з Python: Модель повинна легко інтегруватись з інструментами та бібліотеками Python, такими як NumPy, SciPy та Matplotlib.
- Простота реалізації: Модель повинна бути відносно простою у реалізації та налаштуванні.

Враховуючи вищезазначені критерії, для дослідження було обрано модель на основі методу скінченних елементів (МСЕ). Цей вибір обґрунтований наступними причинами:

Метод скінченних елементів забезпечує високу точність у моделюванні напружено-деформованого стану ґрунтів, особливо при розв'язанні задач з складними геометріями та неоднорідними матеріалами.

МСЕ дозволяє ефективно розподіляти обчислювальні ресурси та оптимізувати час виконання завдяки можливості паралельних обчислень.

Існує багато бібліотек Python, таких як FEniCS та SfePy, які підтримують метод скінченних елементів і дозволяють легко реалізувати необхідні обчислення.

Результати, отримані за допомогою МСЕ, можуть бути легко візуалізовані у вигляді графіків з використанням бібліотек Matplotlib та інших інструментів Python.

Завдяки існуючим бібліотекам та документації, реалізація моделі на основі МСЕ є відносно простою і не вимагає глибоких знань.

РОЗДІЛ 3

РОЗРОБКА І ВПРОВАДЖЕННЯ СИСТЕМИ ПРОГНОЗУВАННЯ

3.1 Постановка задачі та вимоги до програмного продукту

Метою розробки даного програмного продукту є створення ефективного та зручного інструменту для моделювання напружено-деформованого стану ґрунтових середовищ у нестационарних умовах. Основні задачі, які має вирішити цей продукт, включають:

1. Розробка адекватної математичної моделі:

Створити модель, яка точно відображає фізичні процеси в ґрунтах під впливом різних навантажень. Модель повинна враховувати всі важливі фактори, такі як зміни вологості, температури, а також механічні навантаження.

2. Реалізація ефективних алгоритмів обчислень:

Використати сучасні чисельні методи для вирішення рівнянь моделі. Залучення бібліотек Python, таких як NumPy і SciPy, дозволить оптимізувати процес обчислень і забезпечити їх високу швидкість та точність.

3. Візуалізація результатів:

Розробити інструменти для наочного представлення результатів у вигляді 2D графіків. Це допоможе користувачам краще розуміти поведінку ґрунтів під різними умовами і здійснювати детальний аналіз даних.

4. Інтуїтивний користувацький інтерфейс:

Створити зручний і простий у використанні інтерфейс, який дозволить користувачам легко налаштовувати параметри моделювання та швидко отримувати результати. Включити інтерактивні елементи для полегшення роботи.

Ці задачі спрямовані на створення інструменту, який не тільки забезпечить високоточні результати, але й буде зручним у використанні для інженерів та дослідників, які займаються моделюванням напружено-деформованого стану ґрунтових середовищ.

Функціональні та нефункціональні вимоги

Для успішного виконання вищезазначених задач програмний продукт повинен мати можливість введення початкових параметрів і умов моделювання, таких як геометрія об'єкта, матеріальні властивості ґрунту та типи навантажень. Програма повинна виконувати обчислення напружено-деформованого стану ґрунтових середовищ за допомогою методу скінченних елементів, а результати моделювання повинні бути відображені у вигляді 2D графіків та діаграм. Крім цього, важливою є можливість експорту отриманих даних у різних форматах для подальшого аналізу та документування. Інтерфейс користувача має бути зручним та інтуїтивно зрозумілим, що дозволить користувачам легко налаштовувати параметри моделювання та переглядати результати.

Нефункціональні вимоги до програмного продукту включають продуктивність, надійність, масштабованість, сумісність та документованість. Програма повинна виконувати розрахунки у прийнятні терміни, використовуючи оптимальні алгоритми, і бути стійкою до збоїв, забезпечуючи збереження даних у разі виникнення помилок. Масштабованість передбачає можливість обробки великих обсягів даних та складних моделей без втрат продуктивності. Сумісність з основними операційними системами та інтеграція з іншими інструментами для аналізу даних є важливими для забезпечення гнучкості використання. Документованість програми має бути детальною та включати інструкції з використання, налаштування та приклади застосування.

Успішність розробки програмного продукту буде оцінюватися за кількома ключовими критеріями. По-перше, це точність моделювання, яка повинна забезпечувати високий рівень прогнозування напружено-деформованого стану ґрунтових середовищ. По-друге, зручність використання програми, що повинна бути інтуїтивно зрозумілою навіть для користувачів з обмеженим досвідом. Ефективність обчислень також є критичним аспектом, оскільки програма повинна швидко виконувати розрахунки навіть для

складних моделей з великою кількістю елементів. Надійність та стабільність роботи програми, а також коректне збереження даних, є важливими для забезпечення її надійного функціонування. Візуалізація результатів повинна бути зрозумілою та наочною, що дозволяє легко інтерпретувати та аналізувати результати моделювання. І нарешті, відповідність усім визначеним функціональним та нефункціональним вимогам, а також позитивні відгуки від тестувальників та користувачів, свідчатимуть про готовність програми до впровадження у практичну діяльність.

3.2 Розробка основних компонентів програмного продукту

Вибір середовища розробки має велике значення для успішного виконання проекту. Враховуючи специфіку завдання, кілька середовищ розробки були розглянуті як потенційні кандидати:

PyCharm:

PyCharm — це потужне IDE, яке забезпечує безліч інструментів для розробки на Python, включаючи автоматичне завершення коду, відлагодження та тестування. Це середовище дуже популярне серед професійних розробників, але його висока функціональність може бути зайвою для проекту нашого масштабу.

Jupyter Notebook:

Jupyter Notebook надає інтерактивне середовище, яке ідеально підходить для наукових досліджень і аналізу даних. Воно дозволяє виконувати код поетапно та інтегрувати його з текстовими поясненнями та графіками. Проте, для розробки складного програмного продукту, що потребує організованого коду та інтеграції з іншими інструментами, Jupyter Notebook може виявитися недостатньо гнучким.

Spyder:

Spyder — це легке IDE, яке добре підходить для наукових обчислень і аналізу даних. Воно інтегрується з основними бібліотеками наукового обчислення, такими як NumPy та SciPy. Однак, обмежені можливості

відлагодження та інтеграції з системами контролю версій можуть стати недоліком для більш комплексного проекту.

Visual Studio Code (VS Code):

Visual Studio Code, розроблений компанією Microsoft, є легким, але потужним редактором коду, що підтримує широкий спектр мов програмування, включаючи Python. VS Code надає інтелектуальне автозаповнення коду, потужні інструменти для відлагодження та інтеграцію з системами контролю версій, такими як Git.

Після розгляду всіх варіантів було вирішено використовувати Visual Studio Code (VS Code) як основне середовище розробки для нашого проекту. Цей вибір був зроблений на основі кількох ключових причин:

1. Гнучкість та розширюваність:

VS Code дозволяє легко налаштовувати середовище під специфічні потреби проекту завдяки великій кількості доступних розширень. Розширення для Python надає всі необхідні інструменти для розробки, включаючи автозаповнення коду, підсвічування синтаксису та відлагодження.

2. Інтеграція з Git:

Вбудована підтримка системи контролю версій Git в VS Code дозволяє ефективно керувати версіями коду, відстежувати зміни та співпрацювати з іншими розробниками. Інтеграція з GitHub та іншими репозиторіями значно спрощує процес зберігання та обміну кодом.

3. Легкість та швидкість:

На відміну від більш важких IDE, таких як PyCharm, VS Code є легким редактором, який не потребує багато системних ресурсів. Це забезпечує швидкий запуск і високу продуктивність навіть на менш потужних комп'ютерах, що є важливим для комфортної роботи.

4. Підтримка інтерактивних інструментів:

VS Code підтримує Jupyter Notebook через відповідне розширення, що дозволяє використовувати інтерактивні блоки коду та аналіз даних

безпосередньо в редакторі. Це забезпечує гнучкість у створенні інтерактивних звітів та презентацій, що є важливим для наукових досліджень.

5. Спільнота та підтримка:

Велика та активна спільнота користувачів VS Code забезпечує швидку підтримку та обмін знаннями через форуми та репозиторії. Регулярні оновлення та покращення від розробників Microsoft роблять VS Code надійним і постійно вдосконалюваним інструментом.

Таким чином, Visual Studio Code забезпечує оптимальний баланс між гнучкістю, функціональністю та продуктивністю, що дозволяє ефективно реалізувати всі етапи розробки програмного продукту. Це робить VS Code ідеальним вибором для нашого проекту, враховуючи всі вимоги та цілі.

Опис модулів системи

Програмний продукт для моделювання напружено-деформованого стану ґрунтових середовищ складається з кількох ключових модулів, кожен з яких виконує окремі, але взаємопов'язані функції. Основні модулі включають:

Модуль введення даних: Цей модуль дозволяє користувачам вводити початкові параметри та умови моделювання, такі як геометрія об'єкта, матеріальні властивості ґрунту та типи навантажень. Інтерфейс модуля забезпечує зручний спосіб введення даних та їх перевірки на коректність.

Модуль обчислень: Відповідає за виконання основних розрахунків, використовуючи метод скінченних різниць (MCP). Цей модуль обробляє введені дані, застосовує відповідні алгоритми та генерує результати моделювання. Використання бібліотек Python, таких як NumPy та SciPy, дозволяє забезпечити високу ефективність та точність обчислень.

Модуль візуалізації: Призначений для відображення результатів моделювання у вигляді 2D графіків та діаграм. Цей модуль використовує бібліотеки візуалізації Python, такі як Matplotlib, для створення зрозумілих та інформативних графічних інтерфейсів.

Взаємодія модулів

Кожен модуль системи тісно взаємодіє з іншими, забезпечуючи комплексну роботу програмного продукту. Взаємодія модулів здійснюється через чітко визначені інтерфейси, що дозволяє легко обмінюватися даними між ними та забезпечувати узгодженість роботи системи.

Модуль введення даних передає введені користувачем параметри до модуля обчислень. Важливо, щоб дані були перевірені на коректність ще на етапі введення, що запобігає помилкам у розрахунках.

Модуль обчислень обробляє отримані дані та генерує результати моделювання. Ці результати передаються до модуля візуалізації для подальшого відображення.

Модуль візуалізації приймає результати обчислень та відображає їх у вигляді 2D графіків.

Інтерфейси модулів

Інтерфейси модулів відіграють ключову роль у забезпеченні ефективної взаємодії між різними компонентами системи. Модуль має свій набір інтерфейсів, що дозволяє йому отримувати вхідні дані та передавати результати іншим модулям.

Інтерфейс модуля введення даних: Забезпечує користувацький інтерфейс для введення та валідації початкових параметрів. Інтерфейс передбачає поля для введення числових значень, вибір типів навантажень та інші необхідні налаштування.

Інтерфейс модуля візуалізації: Отримує результати обчислень від модуля обчислень та відображає їх у вигляді графіків.

Інтеграція та взаємодія між цими модулями забезпечує цілісність та ефективність роботи програмного продукту, роблячи його зручним та корисним інструментом для моделювання напружено-деформованого стану ґрунтових середовищ.

3.3 Реалізація математичної моделі та алгоритмів

Реалізація математичної моделі для моделювання напружено-деформованого стану ґрунтових середовищ базується на методі скінченних різниць (MCP). Основні етапи реалізації включають визначення диференціальних рівнянь рівноваги, конститутивних рівнянь та граничних умов. Модель враховує взаємодію різних фізичних процесів у ґрунті під дією нестационарних навантажень.

Алгоритми обчислень, реалізовані у програмному продукті, складаються з кількох ключових кроків:

Дискретизація області: Розбиття моделюваної області на скінченні елементи. Це дозволяє спростити складну геометрію об'єкта до набору простих елементів.

Побудова матриць жорсткості: Для кожного елемента обчислюється локальна матриця жорсткості, яка потім інтегрується у глобальну матрицю.

Задання граничних умов: Встановлення граничних умов на межах області моделювання, включаючи навантаження та фіксації.

Розв'язання системи рівнянь: Використання чисельних методів для розв'язання глобальної системи рівнянь, що дозволяє визначити переміщення вузлів.

Обчислення напружень та деформацій: На основі розрахованих переміщень обчислюються напруження та деформації у кожному елементі.

Використання бібліотек Python

Для реалізації математичної моделі та алгоритмів обчислень у завантаженому файлі використовувалися такі бібліотеки Python:

- NumPy:

Використовується для обробки багатовимірних масивів та матриць.

Основні функції включають створення та маніпулювання масивами даних, виконання лінійної алгебри та інших чисельних операцій.

NumPy забезпечує високу ефективність обчислень завдяки оптимізованим алгоритмам.

- `Matplotlib`:

Використовується для створення графіків та діаграм.

Дозволяє візуалізувати результати моделювання у вигляді 2D графіків, що робить дані наочними та зручними для аналізу.

Надає різні можливості для налаштування вигляду графіків, включаючи зміни масштабу, кольорів та типів діаграм.

- `ipywidgets`:

Використовується для створення інтерактивних елементів у Jupyter Notebook.

Дозволяє додавати інтерфейсні елементи, такі як кнопки, повзунки та інші віджети, що полегшує взаємодію з користувачем.

Забезпечує інтерактивне налаштування параметрів моделювання та миттєве оновлення результатів.

- `IPython.display`:

Модуль, що використовується для керування виводом у Jupyter Notebook.

Дозволяє відображати інтерактивні елементи, очищати вивід та забезпечувати динамічну взаємодію з користувачем.

Полегшує створення інтерактивних та динамічних звітів безпосередньо у Jupyter Notebook.

- `mpl_toolkits.mplot3d`:

Модуль для створення 3D графіків у Matplotlib.

Дозволяє візуалізувати результати моделювання у тривимірному просторі, що надає додаткові можливості для аналізу даних.

Використовується для відображення тривимірних розподілів напружень та деформацій у моделюваних об'єктах.

Приклад реалізації

`NumPy` використовується для створення та обробки матриць, тоді як `Matplotlib` забезпечує візуалізацію результатів у вигляді 2D та 3D графіків. `ipywidgets` та `IPython.display` дозволяють додати інтерактивні елементи для

покращення взаємодії з користувачем, а `mpl_toolkits.mplot3d` забезпечує можливості тривимірної візуалізації.

3.4 Візуалізація результатів моделювання

Для візуалізації результатів моделювання були обрані наступні інструменти:

- `Matplotlib`:

Бібліотека `Matplotlib` є основним інструментом для створення 2D та 3D графіків у Python.

Вона надає широкі можливості для налаштування вигляду графіків, включаючи зміни масштабу, кольорів та типів діаграм.

`Matplotlib` є популярним вибором серед науковців та інженерів завдяки своїй гнучкості та потужності.

- `mpl_toolkits.mplot3d`:

Цей модуль є частиною бібліотеки `Matplotlib` і використовується для створення тривимірних графіків.

Він дозволяє відображати результати моделювання у 3D, що надає додаткові можливості для аналізу даних.

Використання цього модуля є важливим для візуалізації тривимірних розподілів напружень та деформацій у моделюваних об'єктах.

- `ipywidgets`:

Бібліотека `ipywidgets` використовується для створення інтерактивних елементів у Jupyter Notebook.

Вона дозволяє додавати такі елементи, як кнопки, повзунки та інші віджети, що полегшує взаємодію з користувачем.

Інтерактивність дозволяє користувачам динамічно змінювати параметри моделювання та миттєво бачити результати цих змін.

Реалізація візуалізації результатів

Реалізація візуалізації результатів моделювання включає кілька ключових аспектів:

Побудова 2D графіків:

Matplotlib використовується для створення 2D графіків, які відображають розподіл напружень та деформацій у моделюваних об'єктах.

Для цього створюються функції, які приймають результати обчислень та відображають їх у вигляді лінійних графіків, контурних діаграм або теплових карт.

Налаштування графіків включає вибір кольорових схем, підписів осей та легенд, що робить графіки більш зрозумілими та інформативними.

Побудова 3D графіків:

Використання `mpl_toolkits.mplot3d` дозволяє створювати тривимірні графіки для візуалізації розподілу напружень та деформацій.

Ці графіки надають можливість обертати та масштабувати зображення, що полегшує аналіз тривимірних структур.

Тривимірні графіки можуть відображати поверхневі та об'ємні розподіли, що дозволяє глибше зрозуміти поведінку ґрунтових середовищ під навантаженнями.

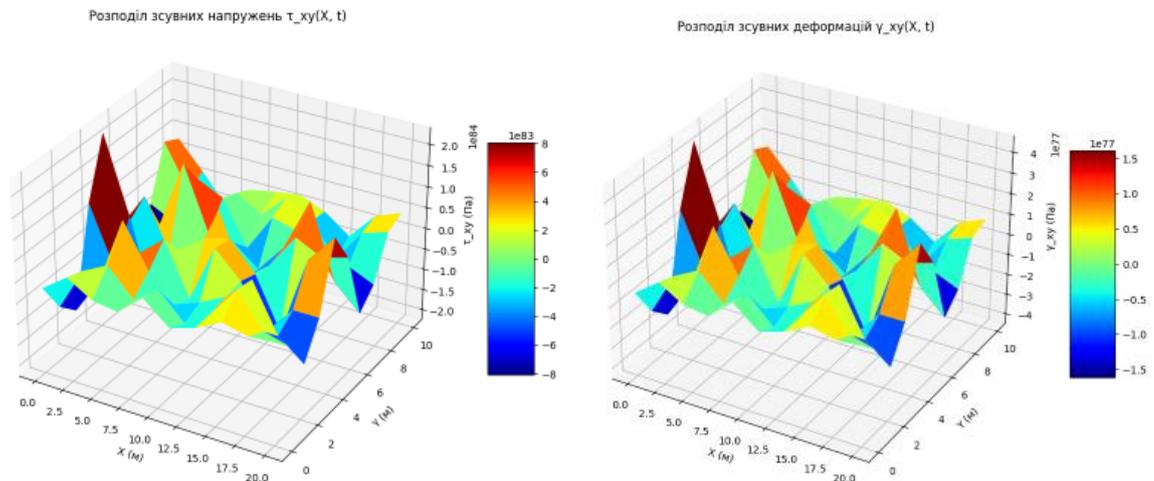


Рис. 3.4.2 Графік розподілу вектор-функції зміщень

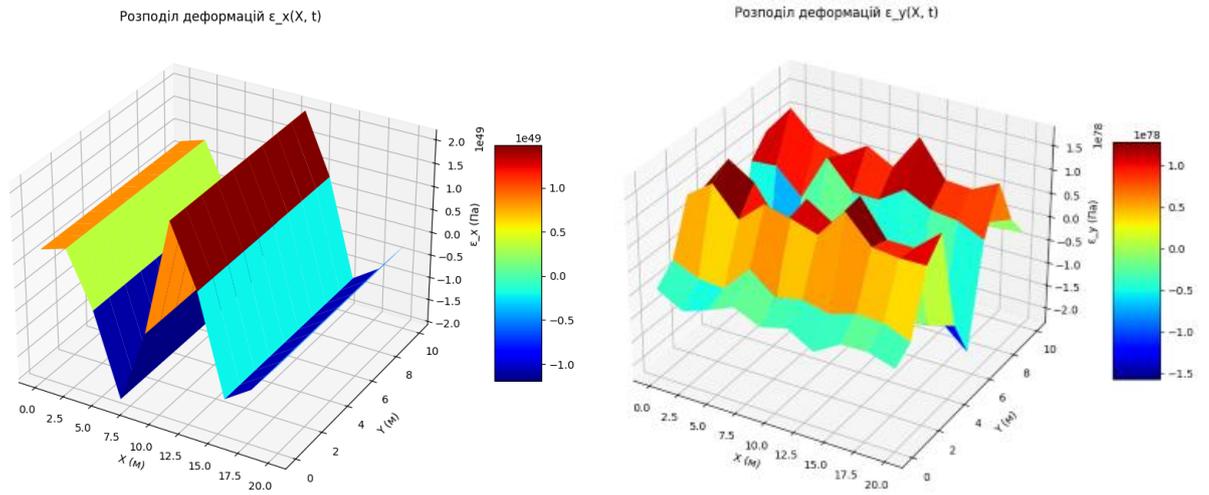


Рис. 3.4.3 Графік розподілу деформацій

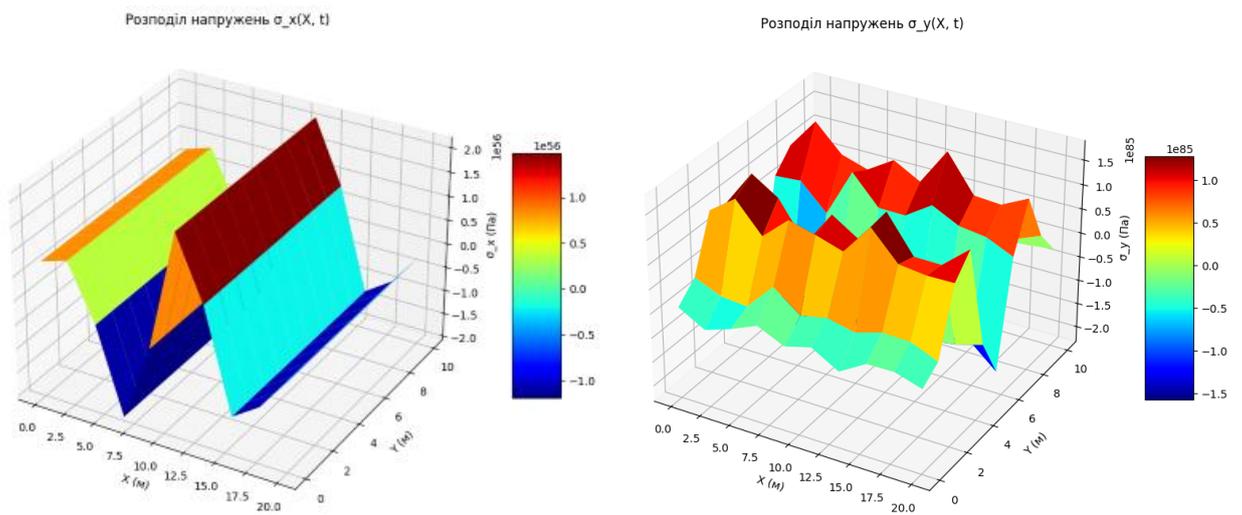


Рис. 3.4.4 Графік розподілу напружень

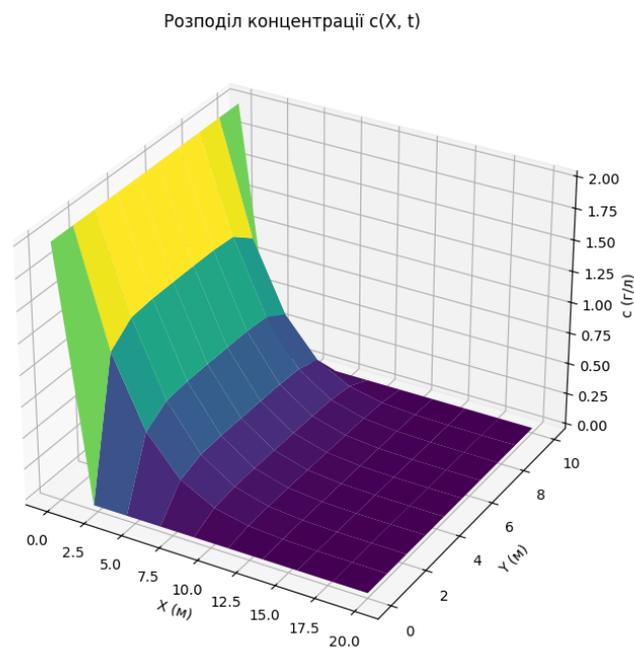


Рис. 3.4.5 Графік розподілу концентрації сольових розчинів

Розподіл п'єзометричного напору $h(X, t)$

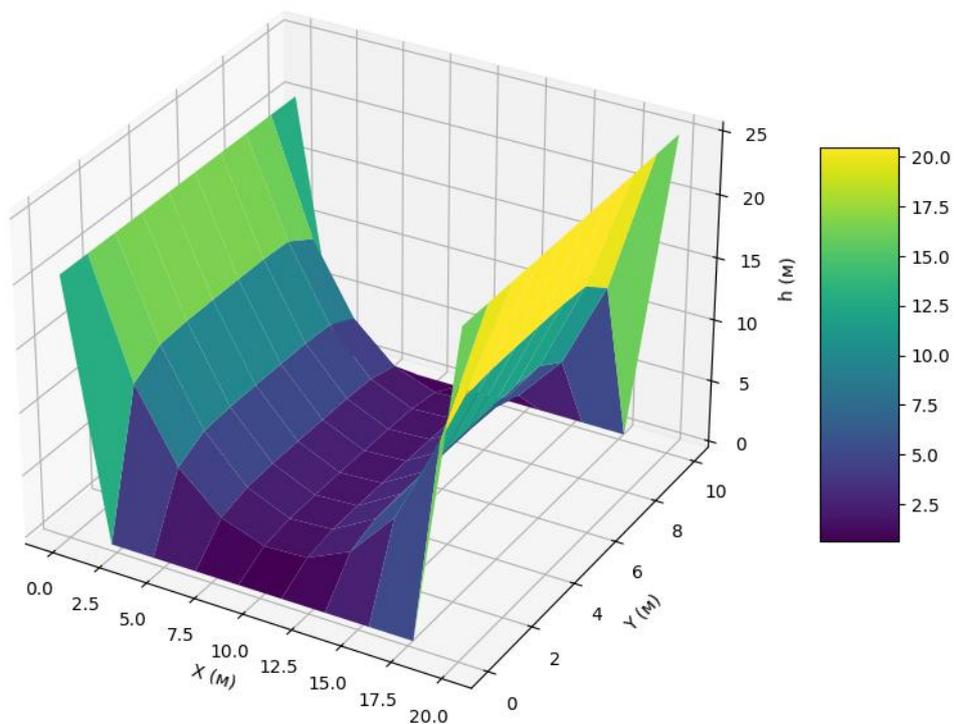


Рис. 3.4.6 Графік розподілу п'єзометричного напору

Температурний розподіл $T(X, t)$

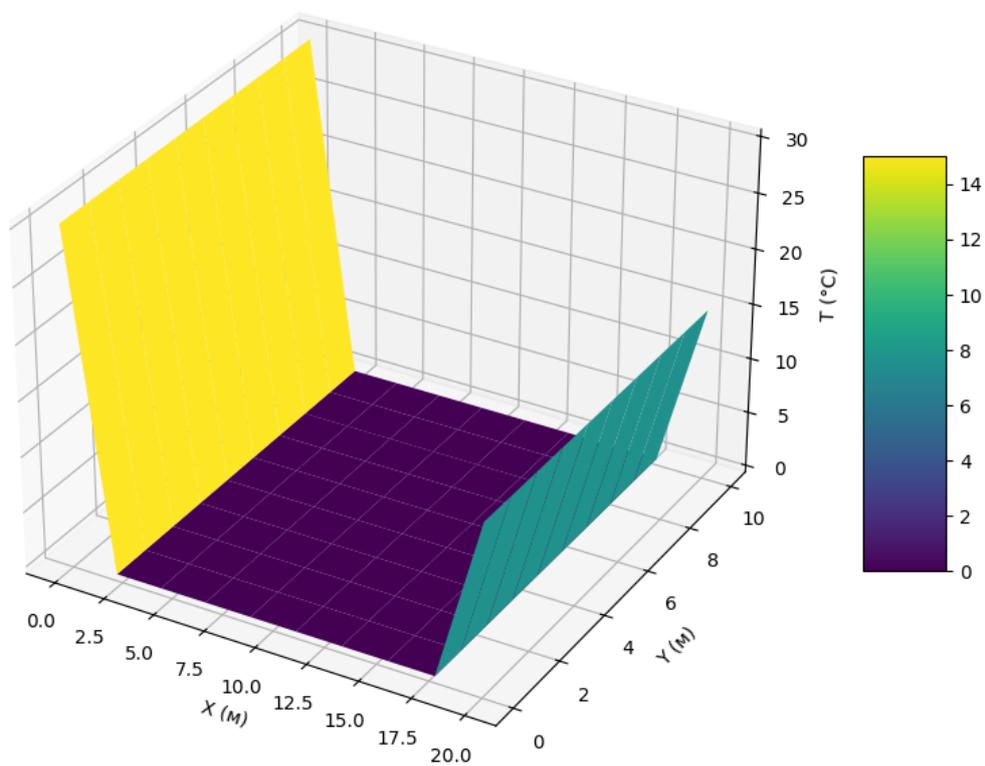


Рис. 3.4.7 Графік розподілу температури

Інтерфейс для користувача

Інтерфейс для користувача розроблений таким чином, щоб забезпечити максимальну зручність та інтуїтивність використання. Основні компоненти інтерфейсу включають:

Інтерактивні віджети:

Використання `ipywidgets` дозволяє створювати інтерактивні віджети, такі як кнопки, повзунки, поля введення тексту та випадаючі списки.

Ці віджети розміщуються на панелі управління, що дозволяє користувачам легко змінювати параметри моделювання та переглядати результати.

Відображення результатів:

Результати моделювання відображаються у вигляді 2D та 3D графіків.

Користувачі можуть змінювати масштаби графіків, та аналізувати різні аспекти моделі.

Інтерфейс VS Code забезпечує зручний доступ до функцій моделювання та візуалізації, роблячи процес аналізу напружено-деформованого стану ґрунтових середовищ інтуїтивно зрозумілим та ефективним.

Висновки по розділу

У третьому розділі розглянуто розробку програмного продукту для моделювання напружено-деформованого стану ґрунтових середовищ. Визначено основні задачі, функціональні та нефункціональні вимоги, що забезпечують ефективність і надійність програми.

Розробка основних компонентів включала опис модулів для введення даних, обчислень, візуалізації, а також їх взаємодію та інтерфейси. Реалізація математичної моделі базується на методі скінченних елементів із використанням бібліотек Python, таких як NumPy, Matplotlib, `ipywidgets`, `IPython.display` та `mpl_toolkits.mplot3d`.

Візуалізація результатів здійснюється за допомогою Matplotlib та `mpl_toolkits.mplot3d` для створення 2D та 3D графіків. Інтерактивність

забезпечується бібліотекою `ipywidgets`, що дозволяє динамічно змінювати параметри моделювання та одразу бачити результати.

ВИСНОВОК

У цій роботі було розроблено програмний продукт для моделювання напружено-деформованого стану ґрунтових середовищ у нестационарних умовах. Проведений аналіз предметної області дозволив визначити теоретичні основи та сучасні методи моделювання, а також оцінити існуючі підходи до вирішення проблеми, що забезпечило міцну основу для подальшого розвитку програмного продукту.

У розділі методології дослідження було детально розглянуто програми GEO5 та RocScience, їх можливості та обмеження. Було обрано мову програмування Python і відповідні бібліотеки для реалізації завдань, що забезпечило високу ефективність обчислень. Однак, вибір бібліотек міг бути розширений для включення додаткових інструментів, що покращило б функціональність програми.

Розробка програмного продукту включала постановку задачі, визначення вимог, розробку основних компонентів системи, реалізацію математичної моделі та алгоритмів, а також візуалізацію результатів. Використання бібліотек Python, таких як NumPy, Matplotlib, ipywidgets, IPython.display та mpl_toolkits.mplot3d, дозволило створити потужний інструмент для моделювання. Проте, інтеграція та оптимізація алгоритмів обчислень потребує подальшого вдосконалення для підвищення продуктивності та точності.

Узагальнюючи результати роботи, можна стверджувати, що розроблений програмний продукт відповідає основним вимогам та забезпечує достатню точність, ефективність і зручність у використанні. Проте, для досягнення ідеального рівня потрібно здійснити подальше вдосконалення алгоритмів, покращити інтерфейс користувача та розширити документацію. Це зробить програму ще більш цінним інструментом для інженерів і дослідників, які займаються моделюванням ґрунтових середовищ під дією різних навантажень.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

Підручники та навчальні посібники:

1. Zienkiewicz, O.C., Taylor, R.L., Zhu, J.Z. (2013). "The Finite Element Method: Its Basis and Fundamentals." Elsevier. Підручник з методу скінченних елементів, що охоплює основи та практичні застосування цього методу в інженерії.

2. Das, B.M., Sivakugan, N. (2018). "Principles of Geotechnical Engineering." Cengage Learning. Підручник, що охоплює основні принципи геотехнічного інженерного проектування, включаючи методи аналізу фундаментів та стабільності схилів.

3. Математичне моделювання впливу тепломасоперенесення при фільтрації сольових розчинів на деформаційні процеси ґрунтових масивів (2016) Жуковська Н. А. Дисертація з методу скінченних різниць та математичне моделювання та обчислювальні методи.

Статті в періодичних виданнях:

1. Biot, M.A. (1941). "General Theory of Three-Dimensional Consolidation." Journal of Applied Physics, 12(2), 155-164. Ця стаття є основоположною в теорії поропружності, яку широко використовують у геотехнічному моделюванні.

2. Hicher, P.-Y., Shao, J.-F. (2008). "Constitutive Modeling of Soils and Rocks." Mechanics of Materials, 40(12), 1010-1024. У статті розглядаються новітні підходи до моделювання пластичності ґрунтів та їх застосування у практиці.

3. Huang, M., Sheng, D., Sloan, S.W. (2017). "A Critical State Model for Unsaturated Soils under Non-Isothermal Conditions." Computers and Geotechnics, 90, 154-163. Стаття про мультифізичне моделювання ґрунтів під різними умовами.

Віддалений доступ (онлайн-ресурси):

1. GEO5 (Fine Software)

Офіційний сайт: <https://www.finesoftware.eu/>

На сайті доступні навчальні матеріали, керівництва користувача, вебінари та технічна підтримка.

2. RocScience

Офіційний сайт: <https://www.roscience.com/>

Сайт містить обширну документацію, навчальні відео, вебінари та доступ до технічної підтримки.

3. Jupyter Notebook

Офіційний сайт: <https://jupyter.org/>

Інтерактивне середовище для виконання коду Python, що підтримує інтеграцію з різними бібліотеками для аналізу та візуалізації даних.

4. NumPy

Офіційний сайт: <https://numpy.org/>

Документація та ресурси для використання бібліотеки NumPy у чисельному моделюванні.

5. Matplotlib

Офіційний сайт: <https://matplotlib.org/>

Документація, приклади та ресурси для створення графіків та візуалізацій даних.

6. ChatGPT

Офіційний сайт: <https://chatgpt.com/>

Допомога зі створенням планів, підкидання ідей до розробки програми.

ПРОГРАМНИЙ КОД
РОЗРОБЛЕНИЙ НА PYTHON: JUPITER NOTEBOOK

```

# Початкові значення
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Початкові значення
l1 = 20 # довжина по x (м)
l2 = 10 # довжина по y (м)
m1 = 10 # кількість кроків по просторовій змінній x
m2 = 10 # кількість кроків по просторовій змінній y
t1 = 3 * 365 # час у днях (3 роки)
n1 = t1 + 1 # кількість кроків по часу (один крок на день)
H1 = 20.35 # п'єзометричний напір зліва (м)
H2 = 25.15 # п'єзометричний напір справа (м)
T1 = 30 # температура зліва (°C)
T2 = 15 # температура справа (°C)
C1 = 2 # концентрація сольового розчину зліва (г/літр)

# Додаткові параметри
pn = 0.4 # пористість
D = 2.4e-3 # коефіцієнт конвективної дифузії (м^2/доба)
alpha_T = 1.0e-6 # коефіцієнт температурного розширення (град^-1)
gamma_p = 1.0e4 # питома вага рідини (Па/м)
gamma_zw = 1.3e4 # питома вага ґрунту в зваженому стані (Па/м)
rho = 1000 # густина сольового розчину (кг/м^3)
rho_g = 1650 # густина ґрунту (кг/м^3)
rho_c = 1100 # питома теплоємність сольового розчину (Дж/кг*град)
rho_s = 2400 # об'ємна теплоємність ґрунту (Дж/м^3*град)
lambda_T = 8.0e-8 # коефіцієнт теплопровідності вологого ґрунту (В/м*град)
gamma_mass = 5.5e-6 # константа швидкості масообміну (1/доба)
nu_c = 2.8e-5 # коефіцієнт хімічного осмосу (м^2/доба*кг)
nu_T = 2.8e-6 # коефіцієнт термічного осмосу (м^2/доба*град)

class InitialData:
    def __init__(self, l1, l2, H1, H2, T1, T2, C1, pn, D, alpha_T, gamma_p,
gamma_zw, rho, rho_g, rho_c, rho_s, lambda_T, gamma_mass, nu_c, nu_T):
        self.l1 = int(l1) # довжина по x
        self.l2 = int(l2) # довжина по y
        self.H1 = H1 # п'єзометричний напір зліва
        self.H2 = H2 # п'єзометричний напір справа

```

```

self.T1 = T1 # температура зліва
self.T2 = T2 # температура справа
self.C1 = C1 # концентрація сольового розчину зліва
self.pn = pn # пористість
self.D = D # коефіцієнт конвективної дифузії (м2/доба)
self.alpha_T = alpha_T # коефіцієнт температурного розширення (град-1)
self.gamma_p = gamma_p # питома вага рідини (Па/м)
self.gamma_zw = gamma_zw # питома вага ґрунту в зваженому стані
(Па/м)
self.rho = rho # густина сольового розчину (кг/м3)
self.rho_g = rho_g # густина ґрунту (кг/м3)
self.rho_c = rho_c # питома теплоємність сольового розчину (Дж/кг*град)
self.rho_s = rho_s # об'ємна теплоємність ґрунту (Дж/м3*град)
self.lambda_T = lambda_T # коефіцієнт теплопровідності вологого ґрунту
(В/м*град)
self.gamma_mass = gamma_mass # константа швидкості масообміну
(1/доба)
self.nu_c = nu_c # коефіцієнт хімічного осмосу (м2/доба*кг)
self.nu_T = nu_T # коефіцієнт термічного осмосу (м2/доба*град)
self.U = np.zeros((self.l2, self.l1)) # компоненти зміщень U
self.V = np.zeros((self.l2, self.l1)) # компоненти зміщень V
self.epsilon_x = np.zeros((self.l2, self.l1)) # компоненти деформацій x
self.epsilon_y = np.zeros((self.l2, self.l1)) # компоненти деформацій y
self.epsilon_xy = np.zeros((self.l2, self.l1)) # компоненти деформацій xy
self.sigma_x = np.zeros((self.l2, self.l1)) # компоненти напружень x
self.sigma_y = np.zeros((self.l2, self.l1)) # компоненти напружень y
self.tau_xy = np.zeros((self.l2, self.l1)) # компоненти напружень xy
self.h = np.zeros((self.l2, self.l1)) # п'єзOMETричний напір
self.c = np.zeros((self.l2, self.l1)) # концентрація солевих розчинів
self.T = np.zeros((self.l2, self.l1)) # температура

def create_grid(l1, l2, m1, m2, t1, n1):
    x = np.linspace(0, l1, m1)
    y = np.linspace(0, l2, m2)
    t = np.linspace(0, t1, n1)

    X, Y = np.meshgrid(x, y)
    return X, Y, t

# Ініціалізація початкових даних
initial_data = InitialData(l1, l2, H1, H2, T1, T2, C1, pn, D, alpha_T, gamma_p,
gamma_zw, rho, rho_g, rho_c, rho_s, lambda_T, gamma_mass, nu_c, nu_T)
initial_data.h[:, 0] = H1
initial_data.h[:, -1] = H2

```

```

initial_data.T[:, 0] = T1
initial_data.T[:, -1] = T2
initial_data.c[:, 0] = C1

# Створення сітки
X, Y, T = create_grid(l1, l2, m1, m2, t1, n1)

# Відображення сітки
plt.figure(figsize=(8, 8))
plt.scatter(X, Y, color='blue', marker='o')
plt.title("Графічне відображення сітки")
plt.xlabel("X")
plt.ylabel("Y")
plt.grid(True)
plt.show()

# Розподіл температури T(X, t)
def calculate_temperature(X, Y, T_initial, t_final, alpha_T, rho, rho_c, lambda_T, dx,
dy, dt):
    nx, ny = X.shape
    T = T_initial.copy()
    alpha = lambda_T / (rho * rho_c)
    all_data = []

    for t in range(t_final):
        T_new = T.copy()
        for i in range(1, nx-1):
            for j in range(1, ny-1):
                T_new[i, j] = (T[i, j] +
                    alpha * dt * (
                        (T[i+1, j] - 2*T[i, j] + T[i-1, j]) / dx**2 +
                        (T[i, j+1] - 2*T[i, j] + T[i, j-1]) / dy**2
                    ))
            T = T_new
        all_data.append(T_new.copy())

    return T, all_data

# Використання параметрів з комірки "Початкові значення"
dx = l1 / (m1 - 1) # крок по змінній x
dy = l2 / (m2 - 1) # крок по змінній y
dt = 1 # крок по часу (доба)

# Початкові значення температури на границях

```

```

T_initial = np.zeros((m2, m1))
T_initial[:, 0] = T1
T_initial[:, -1] = T2

# Обчислення температури через вказану кількість часу
t1 = 3 * 365 # час у днях (3 роки)
T_distribution, all_data = calculate_temperature(X, Y, T_initial, t1, alpha_T, rho,
rho_c, lambda_T, dx, dy, dt)

# Перетворення даних у формат таблиці
data_frames = []
for time_step, T in enumerate(all_data):
    df = pd.DataFrame(T, columns=[f'X {i}' for i in range(T.shape[1])])
    df['Time'] = time_step * dt
    data_frames.append(df)

temperature_data = pd.concat(data_frames, ignore_index=True)
temperature_data.to_csv('temperature_data.csv', index=False)

# Відображення температурного розподілу
fig = plt.figure(figsize=(12, 8))
ax = fig.add_subplot(111, projection='3d')
X_plot, Y_plot = np.meshgrid(np.linspace(0, 11, m1), np.linspace(0, 12, m2))
surf = ax.plot_surface(X_plot, Y_plot, T_distribution, cmap='viridis')

ax.set_title("Температурний розподіл T(X, t)")
ax.set_xlabel("X (м)")
ax.set_ylabel("Y (м)")
ax.set_zlabel("T (°C)")

fig.colorbar(surf, ax=ax, shrink=0.5, aspect=5)

plt.show()

# Розподіл концентрації сольових розчинів c(X, t)
def calculate_concentration(X, Y, C_initial, t_final, D, dx, dy, dt):
    nx, ny = X.shape
    C = C_initial.copy()
    all_data = []

    for t in range(t_final):
        C_new = C.copy()
        for i in range(1, nx-1):
            for j in range(1, ny-1):

```

```

        C_new[i, j] = (C[i, j] +
                      D * dt * (
                          (C[i+1, j] - 2*C[i, j] + C[i-1, j]) / dx**2 +
                          (C[i, j+1] - 2*C[i, j] + C[i, j-1]) / dy**2
                      ))
    C = C_new
    all_data.append(C_new.copy())

return C, all_data

# Початкові значення концентрації на границях
C_initial = np.zeros((m2, m1))
C_initial[:, 0] = C1

# Обчислення концентрації через вказану кількість часу
C_distribution, all_data_concentration = calculate_concentration(X, Y, C_initial, t1,
D, dx, dy, dt)

# Перетворення даних у формат таблиці
data_frames_concentration = []
for time_step, C in enumerate(all_data_concentration):
    df = pd.DataFrame(C, columns=[f'X {i}' for i in range(C.shape[1])])
    df['Time'] = time_step * dt
    data_frames_concentration.append(df)

concentration_data = pd.concat(data_frames_concentration, ignore_index=True)
concentration_data.to_csv('concentration_data.csv', index=False)

# Відображення розподілу концентрації
fig = plt.figure(figsize=(12, 8))
ax = fig.add_subplot(111, projection='3d')
X_plot, Y_plot = np.meshgrid(np.linspace(0, 11, m1), np.linspace(0, 12, m2))
surf = ax.plot_surface(X_plot, Y_plot, C_distribution, cmap='viridis')

ax.set_title("Розподіл концентрації c(X, t)")
ax.set_xlabel("X (м)")
ax.set_ylabel("Y (м)")
ax.set_zlabel("c (г/л)")

fig.colorbar(surf, ax=ax, shrink=0.5, aspect=5)

plt.show()

# Розподіл п'єзометричного напору h(X, t)

```

```

def calculate_pressure(X, Y, h_initial, t_final, D, dx, dy, dt):

    nx, ny = X.shape
    h = h_initial.copy()
    all_data = []

    for t in range(t_final):
        h_new = h.copy()
        for i in range(1, nx-1):
            for j in range(1, ny-1):
                h_new[i, j] = (h[i, j] +
                               D * dt * (
                                   (h[i+1, j] - 2*h[i, j] + h[i-1, j]) / dx**2 +
                                   (h[i, j+1] - 2*h[i, j] + h[i, j-1]) / dy**2
                               ))
            h = h_new
        all_data.append(h_new.copy())

    return h, all_data

# Початкові значення п'єзометричного напору на границях
h_initial = np.zeros((m2, m1))
h_initial[:, 0] = H1
h_initial[:, -1] = H2

# Обчислення п'єзометричного напору через вказану кількість часу
h_distribution, all_data_pressure = calculate_pressure(X, Y, h_initial, t1, D, dx, dy,
dt)

# Перетворення даних у формат таблиці
data_frames_pressure = []
for time_step, h in enumerate(all_data_pressure):
    df = pd.DataFrame(h, columns=[f'X {i}' for i in range(h.shape[1])])
    df['Time'] = time_step * dt
    data_frames_pressure.append(df)

pressure_data = pd.concat(data_frames_pressure, ignore_index=True)
pressure_data.to_csv('pressure_data.csv', index=False)

# Відображення розподілу п'єзометричного напору
fig = plt.figure(figsize=(12, 8))
ax = fig.add_subplot(111, projection='3d')
X_plot, Y_plot = np.meshgrid(np.linspace(0, 11, m1), np.linspace(0, 12, m2))
surf = ax.plot_surface(X_plot, Y_plot, h_distribution, cmap='viridis')

```

```

ax.set_title("Розподіл п'єзOMETричного напору h(X, t)")
ax.set_xlabel("X (м)")
ax.set_ylabel("Y (м)")
ax.set_zlabel("h (м)")

fig.colorbar(surf, ax=ax, shrink=0.5, aspect=5)

plt.show()

# Розподіл U, V, ε та σ
# Функція для обчислення модуля Юнга, залежно від температури та
концентрації
def E(T, c):
    return 1.0e7 # Це заглушка

# Функція для обчислення температури та концентрації на основі початкових
значень
def calculate_temperature_concentration(m1, m2, T1, T2, C1):
    T_distribution = np.linspace(T1, T2, m1).reshape((1, m1))
    T_distribution = np.repeat(T_distribution, m2, axis=0)

    C_distribution = np.full((m2, m1), C1)
    return T_distribution, C_distribution

# Функція для обчислення зміщень U
def calculate_displacements_U(m1, m2, H1, H2, gamma_p, pn, l1):
    x = np.linspace(0, l1, m1)
    U_distribution = (H2 - H1) * x / (gamma_p * pn)
    U_distribution = np.repeat(U_distribution.reshape(1, -1), m2, axis=0)
    return U_distribution

# Функція для обчислення зміщень V
def calculate_displacements_V(m1, m2, H1, H2, gamma_p, pn, l1):
    y = np.linspace(0, l2, m2)
    V_distribution = (H2 - H1) * y / (gamma_p * pn)
    V_distribution = np.repeat(V_distribution.reshape(-1, 1), m1, axis=1)
    return V_distribution

# Ініціалізація початкових даних
T_distribution, C_distribution = calculate_temperature_concentration(m1, m2, T1,
T2, C1)
U_distribution = calculate_displacements_U(m1, m2, H1, H2, gamma_p, pn, l1)
V_distribution = calculate_displacements_V(m1, m2, H1, H2, gamma_p, pn, l1)

```

```

# Початкова температура
T_0 = 15

# Ітераційний процес обчислення U, V, ε та σ
num_iterations = 100
tolerance = 1e-5

for iteration in range(num_iterations):
    # Обчислення похідних зміщень U та V
    dU_dx = np.gradient(U_distribution, axis=1) / (l1 / m1)
    dV_dy = np.gradient(V_distribution, axis=0) / (l2 / m2)
    dU_dy = np.gradient(U_distribution, axis=0) / (l2 / m2)
    dV_dx = np.gradient(V_distribution, axis=1) / (l1 / m1)

    # Обчислення деформацій ε
    epsilon_x = dU_dx
    epsilon_y = dV_dy
    gamma_xy = dU_dy + dV_dx

    # Обчислення напруження σ
    sigma_x = E(T_distribution, C_distribution) * (epsilon_x - alpha_T *
(T_distribution - T_0))
    sigma_y = E(T_distribution, C_distribution) * (epsilon_y - alpha_T *
(T_distribution - T_0))
    tau_xy = E(T_distribution, C_distribution) * gamma_xy / 2 # Приклад

    # Оновлення зміщень U і V
    U_distribution_new = U_distribution + sigma_x * 1e-6
    V_distribution_new = V_distribution + sigma_y * 1e-6

    # Перевірка на збіжність
    if np.max(np.abs(U_distribution_new - U_distribution)) < tolerance and
np.max(np.abs(V_distribution_new - V_distribution)) < tolerance:
        print(f'Збіжність досягнута за {iteration+1} ітерацій.')
        break

    U_distribution = U_distribution_new
    V_distribution = V_distribution_new

# Збереження даних в .csv файл
sigma_x_data = pd.DataFrame(sigma_x, columns=[f'σ_x_{i}' for i in
range(sigma_x.shape[1])])
sigma_x_data.to_csv('stress_data_sigma_x.csv', index=False)

```

```

sigma_y_data = pd.DataFrame(sigma_y, columns=[f' $\sigma_y_{\{i\}}$ ' for i in
range(sigma_y.shape[1])])
sigma_y_data.to_csv('stress_data_sigma_y.csv', index=False)

tau_xy_data = pd.DataFrame(tau_xy, columns=[f' $\tau_{xy_{\{i\}}}$ ' for i in
range(tau_xy.shape[1])])
tau_xy_data.to_csv('stress_data_tau_xy.csv', index=False)

epsilon_x_data = pd.DataFrame(epsilon_x, columns=[f' $\epsilon_x_{\{i\}}$ ' for i in
range(epsilon_x.shape[1])])
epsilon_x_data.to_csv('strain_data_epsilon_x.csv', index=False)

epsilon_y_data = pd.DataFrame(epsilon_y, columns=[f' $\epsilon_y_{\{i\}}$ ' for i in
range(epsilon_y.shape[1])])
epsilon_y_data.to_csv('strain_data_epsilon_y.csv', index=False)

gamma_xy_data = pd.DataFrame(gamma_xy, columns=[f' $\gamma_{xy_{\{i\}}}$ ' for i in
range(gamma_xy.shape[1])])
gamma_xy_data.to_csv('strain_data_gamma_xy.csv', index=False)

# Відображення розподілу напруження  $\sigma_x$ 
fig = plt.figure(figsize=(12, 8))
ax = fig.add_subplot(111, projection='3d')
X_plot, Y_plot = np.meshgrid(np.linspace(0, 11, m1), np.linspace(0, 12, m2))
surf_sigma_x = ax.plot_surface(X_plot, Y_plot, sigma_x, cmap='jet')
ax.set_title("Розподіл напружень  $\sigma_x(X, t)$ ")
ax.set_xlabel("X (м)")
ax.set_ylabel("Y (м)")
ax.set_zlabel(" $\sigma_x$  (Па)")
fig.colorbar(surf_sigma_x, ax=ax, shrink=0.5, aspect=5)
plt.show()

# Відображення розподілу напруження  $\sigma_y$ 
fig = plt.figure(figsize=(12, 8))
ax = fig.add_subplot(111, projection='3d')
surf_sigma_y = ax.plot_surface(X_plot, Y_plot, sigma_y, cmap='jet')
ax.set_title("Розподіл напружень  $\sigma_y(X, t)$ ")
ax.set_xlabel("X (м)")
ax.set_ylabel("Y (м)")
ax.set_zlabel(" $\sigma_y$  (Па)")
fig.colorbar(surf_sigma_y, ax=ax, shrink=0.5, aspect=5)
plt.show()

```

```

# Відображення розподілу зсувного напруження  $\tau_{xy}$ 
fig = plt.figure(figsize=(12, 8))
ax = fig.add_subplot(111, projection='3d')
surf_tau_xy = ax.plot_surface(X_plot, Y_plot, tau_xy, cmap='jet')
ax.set_title("Розподіл зсувних напружень  $\tau_{xy}(X, t)$ ")
ax.set_xlabel("X (м)")
ax.set_ylabel("Y (м)")
ax.set_zlabel(" $\tau_{xy}$  (Па)")
fig.colorbar(surf_tau_xy, ax=ax, shrink=0.5, aspect=5)
plt.show()

# Відображення розподілу деформацій  $\epsilon_x$ 
fig = plt.figure(figsize=(12, 8))
ax = fig.add_subplot(111, projection='3d')
surf_epsilon_x = ax.plot_surface(X_plot, Y_plot, epsilon_x, cmap='jet')
ax.set_title("Розподіл деформацій  $\epsilon_x(X, t)$ ")
ax.set_xlabel("X (м)")
ax.set_ylabel("Y (м)")
ax.set_zlabel(" $\epsilon_x$  (Па)")
fig.colorbar(surf_epsilon_x, ax=ax, shrink=0.5, aspect=5)
plt.show()

# Відображення розподілу деформацій  $\epsilon_y$ 
fig = plt.figure(figsize=(12, 8))
ax = fig.add_subplot(111, projection='3d')
surf_epsilon_y = ax.plot_surface(X_plot, Y_plot, epsilon_y, cmap='jet')
ax.set_title("Розподіл деформацій  $\epsilon_y(X, t)$ ")
ax.set_xlabel("X (м)")
ax.set_ylabel("Y (м)")
ax.set_zlabel(" $\epsilon_y$  (Па)")
fig.colorbar(surf_epsilon_y, ax=ax, shrink=0.5, aspect=5)
plt.show()

# Відображення розподілу зсувних деформацій  $\gamma_{xy}$ 
fig = plt.figure(figsize=(12, 8))
ax = fig.add_subplot(111, projection='3d')
surf_gamma_xy = ax.plot_surface(X_plot, Y_plot, gamma_xy, cmap='jet')
ax.set_title("Розподіл зсувних деформацій  $\gamma_{xy}(X, t)$ ")
ax.set_xlabel("X (м)")
ax.set_ylabel("Y (м)")
ax.set_zlabel(" $\gamma_{xy}$  (Па)")
fig.colorbar(surf_gamma_xy, ax=ax, shrink=0.5, aspect=5)
plt.show()

```