МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В. Н. Каразіна
Бахмутський навчально-науковий професійно-педагогічний інститут
Кафедра електромеханічних та комп’ютерних систем

До захисту допущено
Завідувач кафедри
__________ Інна НЕФЬОДОВА
(підпис) (ім’я, прізвище)

« 07 » грудня 2024 року

КВАЛІФІКАЦІЙНА РОБОТА (ПРОЄКТ)

рівень вищої освіти другий (магістерський)

спеціальність 015.39 Професійна освіта (Цифрові технології)

освітньо-професійна програма Професійна освіта. Комп’ютерні технології в управлінні та навчанні

тема «Професійна підготовка фахівців у галузі цифрових технологій до розробки навчальних ресурсів з програмування чисельних методів розв’язання задач диференціального та інтегрального числення мовою Python»

Виконав(ла)

здобувач(ка) групи БД-К23мг
(шифр групи)

Тимофій КУЧЕРБАЄВ __________
(ім’я, прізвище) (підпис)

Керівник роботи
к.ф.-м.н., доц. Інна НЕФЬОДОВА __________
(науковий ступінь, вчене звання, ім’я, прізвище) (підпис)

Рецензент роботи
д.ф.-м.н., проф. Олег ЛИТВИН __________
(науковий ступінь, вчене звання, ім’я, прізвище) (підпис)

Консультант
к.пед.н., доц. Юлія БОБРИКОВА __________
(науковий ступінь, вчене звання, ім’я, прізвище) (підпис)

Засвідчую, що у цій роботі немає цитат та вилучень з праць інших авторів без відповідних посилань
здобувач (ка)______ ______
(підпис)

Харків – 2024

Харківський національний університет імені В. Н. Каразіна

Факультет/ННІ Бахмутський навчально-науковий професійно-педагогічний інститут

Кафедра Електромеханічних та комп'ютерних систем

Рівень вищої освіти другий (магістерський)

Спеціальність 015.39 Професійна освіта (Цифрові технології)

Освітньо-професійна програма Професійна освіта. Комп'ютерні технології в управлінні та навчанні

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інна НЕФЬОДОВА

(підпис) (ім'я, прізвище)

« 08 » жовтня 2024 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ (ПРОЄКТ)

Кучербаєв Тимофій Маратович

(прізвище, ім'я, по батькові здобувача)

1. Тема роботи Професійна підготовка фахівців у галузі цифрових технологій до розробки навчальних ресурсів з програмування чисельних методів розв'язання задач диференціального та інтегрального числення мовою Python

керівник роботи Нефьодова Інна Віталіївна, к. ф.-м. н., доцент

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

затверджені наказом по університету від «08» жовтня 2024 року № 5101-5/3263

2. Строк подання здобувачем роботи «02» грудня 2024 р.

3. Перелік питань, які потрібно розробити: Актуальність професійної підготовки фахівців у галузі цифрових технологій до викладання освітнього модулю «Основи роботи в Scilab». Характеристика об'єктів галузі: стан і стратегії розвитку. Розробка лабораторних робіт для викладання освітнього модулю «Основи роботи в Scilab». Методика професійної підготовки фахівців у галузі цифрових технологій з розробки цифрових освітніх ресурсів для освітнього модуля «Основи програмування мовою Python»

4. План роботи

№ з/п	Назви етапів роботи
1	Огляд літературних джерел, нових розробок, опублікованих даних та іншої інформації, пов’язаної з темою роботи
2	Дослідження теоретичних підходів до актуальності професійної підготовки фахівців у галузі цифрових технологій до розробки навчальних ресурсів з програмування чисельних методів розв’язання задач диференціального та інтегрального числення мовою Python
3	Характеристика об'єктів галузі: стан і стратегії розвитку
4	Розробка методики професійної підготовки фахівців у галузі цифрових до розробки навчальних ресурсів з програмування чисельних методів розв’язання задач диференціального та інтегрального числення мовою Python
5	Розробка вимог до кадрового забезпечення об'єкту галузі
6	Оформлення першого варіанту тексту, подання його на ознайомлення науковому керівнику
7	Усунення недоліків, написання остаточного варіанту тексту, оформлення дипломної роботи
8	Подання роботи на кафедру, перевірка на плагіат та зовнішнє рецензування роботи
9	Захист дипломної роботи у ЕК

5. Дата видачі завдання «08» жовтня 2024 р.

Здобувач(ка) ____________ (підпис) Тимофій КУЧЕРБАЄВ (ім’я, прізвище)

Керівник роботи ____________ (підпис) Інна НЕФЬОДОВА (ім’я, прізвище)

РЕФЕРАТ

Об'єктом дослідження є процес професійної підготовки фахівців у галузі цифрових технологій до створення навчальних матеріалів з програмування чисельних методів для розв'язання задач диференційного та інтегрального числення за допомогою мови Python.

Предметом дослідження є методика професійної підготовки фахівців у галузі цифрових технологій до створення навчальних ресурсів з програмування чисельних методів для розв'язання задач диференційного та інтегрального чисельності з використанням мови Python.

Мета дослідження – теоретичне обґрунтування та часткова перевірка методики професійної підготовки фахівців у галузі цифрових технологій для створення навчальних ресурсів, що передбачають програмування чисельних методів розв’язання задач диференціального та інтегрального числення за допомогою мови Python.

В результаті виконання дослідження розроблено блок-схеми алгоритмів та програми мовою Python числових методів диференціювання та інтегрування.

За основними результатами дослідження виконана публікація тез доповіді на VIII Міжнародній науково-практичній конференції здобувачів вищої освіти та молодих учених «Студенти та молодь – для майбутнього країни» (м. Харків, 15 листопада 2024 р.).

Обсяг дипломної роботи становить: пояснювальна записка, презентація доповіді. Пояснювальна записка складається зі вступу, чотирьох розділів, висновків, списку використаних джерел, додатків. Загальний обсяг роботи 70 сторінок, з яких 56 сторінок основного тексту. Список використаних джерел становить 28 найменувань, 6 таблиць, 17 рисунків.

ПРОФЕСІЙНА ПІДГОТОВКА, ЦИФРОВІ ТЕХНОЛОГІЇ, ЧИСЕЛЬНІ МЕТОДИ, ДИФЕРЕНЦІЙНЕ ЧИСЛЕННЯ, ІНТЕГРАЛЬНЕ ЧИСЛЕННЯ, ПРОГРАМУВАННЯ PYTHON, АЛГОРИТМИ, БЛОК-СХЕМИ.

ABSTRACT

The object of the study is the process of professional training of specialists in the field of digital technologies to create educational materials on programming numerical methods for solving differential and integral calculus problems using the Python language.

The subject of the study is the methodology of professional training of specialists in the field of digital technologies to create educational resources on programming numerical methods for solving differential and integral calculus problems using the Python language.

The purpose of the study is to theoretically substantiate and partially verify the methodology for professional training of specialists in the field of digital technologies to create educational resources that involve programming numerical methods for solving problems of differential and integral calculus using the Python language.

As a result of the research, block diagrams of algorithms and programs in Python for numerical differentiation and integration methods were developed.

Based on the main results of the research, the abstract of the report was published at the VIII International Scientific and Practical Conference of Higher Education Students and Young Scientists «Students and Youth – for the Future of the Country» (Kharkiv, November 15, 2024).

The thesis consists of an explanatory note and a presentation of the report. The explanatory note consists of an introduction, four chapters, conclusions, a list of references, and appendices. The total volume of the thesis is 70 pages, of which 56 pages are the main text. The list of references consists of 28 titles, 6 tables, and 17 figures.

PROFESSIONAL TRAINING, DIGITAL TECHNOLOGIES, NUMERICAL METHODS, DIFFERENTIAL CALCULUS, INTEGRAL CALCULUS, PYTHON PROGRAMMING, ALGORITHMS, BLOCK DIAGRAMS.

ЗМІСТ

Вступ………………………………………………………………………. 7

Розділ 1 Актуальність професійної підготовки фахівців у галузі цифрових технологій до розробки навчальних ресурсів з програмування чисельних методів розв'язання задач диференціального та інтегрального числення мовою Python…………………………………… 11

Розділ 2 Характеристика об'єктів галузі: стан і стратегії розвитку……... 16

2.1 Особливості програмування числових методів мовою Python, переваги та недоліки………...………………………………………… 16

2.2 Числові методи розв'язання задач диференціального числення та їх реалізація мовою Python ………………………………………………. 19

2.3 Числові методи розв'язання задач інтегрального числення та їх реалізація мовою Python ………………………………………………. 23

Розділ 3 Вимоги до кадрового забезпечення об'єкту галузі……………… 40

Розділ 4 Методика професійної підготовки фахівців у галузі цифрових технологій до розробки навчальних ресурсів з програмування чисельних методів розв'язання задач диференціального та інтегрального числення мовою Python ………………………….……… 44

Висновки…………………………………………………………...…… 55

Список використаних джерел……………………………...…….. 57

Додаток А ……………………………………………………...……...….. 60

Додаток Б ……………………………………………………...……...….. 69

ВСТУП

Актуальність теми дослідження. Сучасна система вищої освіти в Україні перебуває на етапі трансформації, спрямованої на створення та вдосконалення нових стандартів якості. Однією з головних умов для задоволення потреб нашої держави у кваліфікованих фахівцях є пошук ефективних шляхів удосконалення професійної підготовки вищих навчальних закладів, а також їх впровадження та використання на практиці.

Професійна освіта набуває все більшої важливості не тільки як основний інструмент підготовки фахівців для різних секторів економіки, але й як ключовий етап розвитку особистості, доступний для все більшої кількості людей. Цей процес є важливим як для українського суспільства, так і на міжнародному рівні [3, с.6].

Нова освітня парадигма передбачає необхідність формування та розвитку професійної підготовки фахівців у галузі цифрових технологій, а також їх професійної компетентності, ерудиції та здатності до адаптації в умовах змінюваного світу. У сучасних умовах становлення професіонала неможливе без чіткого визначення змісту освіти, набору професійних компетенцій та ключових особистісних якостей, які мають бути притаманні спеціалісту. Це забезпечує не лише високий рівень знань, але й здатність ефективно діяти в різних професійних ситуаціях та середовищах [13].

Пріоритет у світовій педагогічній науці методології компетентнісного підходу, орієнтованого на потреби індустріального суспільства, підвищує важливість розробки та впровадження цього підходу в процес професійної підготовки фахівців у галузі цифрових технологій, зокрема при розробці навчальних ресурсів для програмування чисельних методів розв’язання задач диференціального та інтегрального числення мовою Python. Це дозволяє не лише задовольнити вимоги сучасного ринку праці, але й створити можливості для розвитку ключових професійних навичок, що є важливими для майбутніх спеціалістів у цій галузі.

Проблеми змісту та сутності професійної підготовки майбутніх ІТ-фахівців у системі вищої школи знайшли відображення у наукових дослідженнях провідних українських учених, таких як: В. Биков, А. Васильєв, А. Гуржій, М. Жалдак, М. Згуровський, Ю. Зубань, Ю. Коровайченко, В. Кухаренко, А. Манако, О. Меняйленко, Н. Морзе, В. Олійник, С. Шкарлет, Є. Полат, С. Раков, О. Співаковський, О. Спірін, С. Семеріков, Ю. Триус та ін.

Аналіз наукових джерел показав, що питання теоретико-методологічного обґрунтування професійної підготовки фахівців у галузі цифрових технологій до розробки навчальних ресурсів з програмування чисельних методів для розв'язання задач диференціального та інтегрального числення мовою Python не знайшли достатнього висвітлення в існуючих дослідженнях. Це виявило низку суперечностей, зокрема:

– зростаючими вимогами суспільства до професійної підготовки інженера-педагога, що не відповідають рівню розвитку теорії та практики його підготовки;

– необхідністю розробки науково-методичного забезпечення для підготовки майбутнього інженера-педагога до впровадження особистісно орієнтованих підходів у навчанні, при цьому відсутні відповідні організаційно-педагогічні умови для цього.

Необхідність подолання виявлених суперечностей й обумовила тему дослідження «Професійна підготовка фахівців у галузі цифрових технологій до розробки навчальних ресурсів з програмування чисельних методів розв’язання задач диференціального та інтегрального числення мовою Python.

Метою дослідження є теоретичне обґрунтування та часткова перевірка методики професійної підготовки фахівців у галузі цифрових технологій для створення навчальних ресурсів, що передбачають програмування чисельних методів розв’язання задач диференціального та інтегрального числення за допомогою мови Python.

З огляду на поставлену мету, у роботі вирішуються такі **завдання:**

1. Проаналізувати та розробити рівень актуальності проблем професійної підготовки фахівців у галузі цифрових технологій до створення навчальних ресурсів для програмування чисельних методів вирішення задач диференціального та інтегрального числення мовою Python.

2. Проаналізувати особливості програмування числових методів мовою Python.

3. Розробити програмне забезпечення мовою Python, що реалізує методи диференціального та інтегрального числення.

4. Розробити теоретичну та методичну основу концептуальної методичної системи професійної підготовки майбутніх фахівців у галузі цифрових технологій для створення навчальних ресурсів із програмування чисельних методів для розв'язання диференційного та інтегрального чисельності мовою Python.

Об'єкт дослідження: процес професійної підготовки фахівців у галузі цифрових технологій до створення навчальних матеріалів з програмування чисельних методів для розв'язання задач диференційного та інтегрального числення за допомогою мови Python.

Предмет дослідження: методика професійної підготовки фахівців у галузі цифрових технологій до створення навчальних ресурсів з програмування чисельних методів для розв'язання задач диференційного та інтегрального чисельності з використанням мови Python.

В ході проведення дослідження використовувалися такі **методи:**

– теоретичні методи (аналіз, синтез, порівняння, моделювання, узагальнення), які були спрямовані на дослідження психолого-педагогічної літератури, визначення концептуальних основ дослідження та уточнення особливостей процесу професійної підготовки майбутніх фахівців у галузі цифрових технологій;

– емпіричні методи (анкетування, бесіди з учасниками експерименту, педагогічне спостереження, самооцінка, тестування), що використовувалися для оцінки рівня розвитку компетентності майбутніх фахівців у сфері цифрових технологій.

Наукова новизна результатів дослідження виявилася в розробці теоретичних основ структури професійної підготовки фахівців у галузі цифрових технологій для створення навчальних матеріалів, орієнтованих на програмування чисельних методів розв'язання задач диференційного та інтегрального числення з використанням мови Python. Крім цього, в роботі детально уточнено поняття «професійна підготовка» фахівців у сфері цифрових технологій.

Також значно розширено підхід до навчання майбутніх фахівців у цій галузі в контексті розробки навчальних ресурсів для чисельних методів, зокрема для вирішення завдань диференціального та інтегрального числення мовою Python. Особливо увага приділяється формуванню самоосвітньої компетентності, що є важливою складовою професійної підготовки.

Теоретичне та практичне значення отриманих результатів виникає в розробці методології професійної підготовки фахівців у галузі цифрових технологій для створення навчальних ресурсів з програмування чисельних методів вирішення диференційного та інтегрального числа завдань з використанням мови Python. Також у роботі обґрунтовано зміст, форми та методи організації процесу підготовки фахівців, що сприяють розвитку їхніх навичок і компетенцій у цій галузі.

Апробація результатів дослідження: за основними результатами дослідження виконана публікація тез доповіді на VIII Міжнародній науково-практичній конференції здобувачів вищої освіти та молодих учених «Студенти та молодь – для майбутнього країни» (м. Харків, 14-15 листопада 2024 р.).

Структура роботи. Робота складається із вступу, чотирьох розділів, висновків, списку використаних джерел та додатків.

РОЗДІЛ 1
АКТУАЛЬНІСТЬ ПРОФЕСІЙНОЇ ПІДГОТОВКИ ФАХІВЦІВ У ГАЛУЗІ ЦИФРОВИХ ТЕХНОЛОГІЙ ДО РОЗРОБКИ НАВЧАЛЬНИХ РЕСУРСІВ З ПРОГРАМУВАННЯ ЧИСЕЛЬНИХ МЕТОДІВ РОЗВ'ЯЗАННЯ ЗАДАЧ ДИФЕРЕНЦІАЛЬНОГО ТА ІНТЕГРАЛЬНОГО ЧИСЛЕННЯ МОВОЮ PYTHON

Соціально-економічні зміни, що відбулися в Україні за останні роки, а також трансформації в освітній сфері, зокрема в системі вищої освіти, поставили перед суспільством завдання підвищення якості професійної діяльності фахівців у галузі цифрових технологій. Це питання стало особливо актуальним у контексті потреби задоволення соціального замовлення на висококваліфіковані кадри, які володіють новими знаннями та навичками, здатними освоювати нові спеціальності замість тих, які вже не відповідають вимогам сучасного ринку праці [10]. У зв'язку з цими змінами постала проблема підготовки фахівців у галузі цифрових технологій, здатних швидко адаптуватися до нових умов соціального середовища та ринку праці, що постійно змінюється, а також до появи нових спеціальностей та спеціалізацій.

Майбутнє та сьогодення України вимагають від вітчизняних управлінців підвищеного рівня професійної компетентності, здатності приймати обґрунтовані та ефективні рішення, а також готовності брати на себе відповідальність за модернізацію країни. Це включає в себе успішну реалізацію реформ, необхідних для інтеграції України до європейського співтовариства, а також для забезпечення стабільного розвитку держави на міжнародній арені [9].

Сьогодні, як ніколи раніше, Україні необхідні висококваліфіковані фахівці на всіх рівнях державної влади. Від рівня їх компетентності та професіоналізму залежить не лише ефективне управління країною, а й її становлення як правової, соціальної та демократичної держави.

У зв'язку з цим питання реформування та вдосконалення існуючої системи підготовки, перепідготовки та підвищення кваліфікації фахівців набуває особливого значення. Трансформація цієї системи у сучасну, ефективну та злагоджену структуру, здатну випускати висококласних спеціалістів, є важливим кроком для досягнення стратегічних цілей країни [15].

Зважаючи на високі вимоги сучасного ринку праці до кваліфікованих фахівців у галузі цифрових технологій, існує нагальна потреба в підвищенні рівня професійної підготовки таких фахівців. Особливо важливим є удосконалення їхніх навичок у розробці навчальних ресурсів з програмування чисельних методів для розв'язання задач диференціального та інтегрального числення за допомогою мови Python. Це дозволить не лише відповідати сучасним вимогам ринку праці, але й забезпечити високий рівень технічної та методичної підготовки у галузі цифрових технологій [13].

Різні аспекти професійної підготовки майбутніх фахівців у галузі цифрових технологій у вищій освіті досліджувались багатьма науковцями, що розробили теоретико-методологічні засади цієї теми. Теоретичною основою дослідження стали загально-дидактичні принципи організації навчання, які були представлені працями таких учених, як Ю.К. Бабанський, А.С. Бєлкін, В.І. Качуровський, С.А. Новосьолов, Г.М. Романцев, Є.В. Ткаченко та інших. Акмеологічна теорія, яка вивчає розвиток дорослих, базується на роботах С.І. Змєєва, Г.Л. Ільїна, Е.М. Нікітіна та інших дослідників. Крім того, теорія індивідуалізації навчання була розвинута за участю М.А. Данилова, А.А. Кірсанова, М.Н. Скаткіна, І. Унта та інших. Основні положення діяльнісного підходу в контексті педагогіки та психології активно розроблялись Л.С. Виготським, А.М. Леонтьєвим, а також іншими видатними вченими. Зокрема, дослідження в галузі педагогіки та психології особистості, яка функціонує в умовах сучасних інформаційних технологій, значною мірою спираються на роботи таких учених, як Г.М. Александров, А.Г. Асмолов, А.В. Антонов, Є.В. Бондаревська, С. Венір, П.Я. Гальперін, Б.С. Гершунський та інші.

Аналіз педагогічної та психологічної літератури підтвердив актуальність підвищення рівня професійної підготовки фахівців у галузі цифрових технологій, враховуючи специфіку цієї сфери, яка залишається недостатньо розробленою. Це вказує на необхідність звернення до цієї проблеми та підтверджує актуальність даного дослідження. Виявлені суперечності вказують на такі ключові аспекти: по-перше, існує потреба в теоретичному обґрунтуванні поняття "підготовка", однак відповідне обґрунтування ще не розроблене; по-друге, спостерігається невідповідність між спеціальністю, обраною фахівцем, і рівнем знань, що він отримує у системі професійної освіти, які не завжди відповідають вимогам сучасного ринку праці; по-третє, необхідність постійного оновлення знань у зв'язку з інтенсивним розвитком нових інформаційних технологій та їх використанням у виробничому процесі, що часто ускладнюється відсутністю можливості оперативного набуття нових компетенцій. Розв'язання цих протиріч є основним завданням даного дослідження [4-6].

Одним з основних завдань професійної підготовки фахівців у галузі цифрових технологій є формування високої професійної компетентності. Важливо зазначити, що основи цієї компетентності закладаються ще під час навчання. Результатом освітнього процесу є здобуття компетенцій, які включають знання, розуміння, практичні вміння, ціннісні орієнтації та інші особистісні якості, що набуваються особою протягом проходження освітньої програми. Вони визначають здатність майбутнього фахівця ефективно виконувати професійні завдання та адаптуватися до вимог постійно змінюваного ринку праці, особливо в умовах розвитку цифрових технологій [10].

Компетентний фахівець – це професіонал, який має глибокі знання в області своєї діяльності і здатний у межах обмеженого часу приймати обґрунтовані рішення, вибираючи з різноманітних варіантів найбільш оптимальні. Професійна перепідготовка інженерів-енергетиків відображає їхню професійну компетентність як поєднання загальнолюдських ідеалів і цінностей, притаманних інженерній професії, а також розвиток важливих професійних якостей. Вона включає освоєння новітніх технологій, що забезпечують ефективне виконання завдань у їхній професійній діяльності та сприяють підвищенню рівня кваліфікації фахівців [2].

Компетентність у міркуваннях науковців постає однією із властивостей особистості, основою якої виступають не лише знання, вміння, навички фахівця, а й обізнаність останнього, досвід його діяльності у своїй професійній сфері [7].

Згідно з поглядами О. І. Бондарчука, В. О. Болотова та В. В. Сєрікова, компетентність є широким поняттям, яке відображає здатність здійснювати діяльність на високому рівні, володіючи необхідними знаннями та навичками. Вона характеризує ступінь відповідності знань і вмінь фахівця до вимог і складності завдань, що стоять перед ним. Часто термін «компетентність» використовують як синонім терміна «кваліфікація», проте між ними є відмінності. Кваліфікація є офіційним результатом оцінки і визнання здобутих знань і вмінь, що підтверджується документом про вищу освіту, виданим уповноваженою установою. Тобто кваліфікація є формальним свідченням того, що особа досягла необхідного рівня компетентності згідно з вимогами стандартів вищої освіти [14, 15].

Чимало науковців і практиків, зокрема В. Андрущенко, Л. Бірюк, Н. Бібік, В. Бобрицька, Л. Васильченко, Л. Ващенко, К. Віаніс-Трофименко, І. Галяміна, С. Гончаренко, О. Дубасенюк, І. Зимня, І. Зязюн, В. Кремень, О. Локшина, О. Овчарук, В. Огнев'юк, Л. Паращенко, О. Пометун, Дж. Равен, О. Савченко, С. Сисоєва, В. Ягупов та інші, присвятили свої дослідження філософському осмисленню сучасних аспектів компетентнісного підходу та формуванню особистості фахівця. Вивчення професійних компетентностей фахівців вказує на необхідність розвитку організаторських здібностей, здатності приймати відповідальні рішення, прагнення до вдосконалення комунікативних навичок, професійних знань і практичних умінь. Також важливо вміти оцінювати соціальні процеси та визначати свою роль у них через призму професійної діяльності. Проте в педагогічній теорії та практиці залишається недостатньо розробленою проблема, яка б комплексно враховувала сучасні тенденції розвитку професійних компетентностей фахівців у галузі цифрових технологій [6, 9].

Професійна компетентність є об'єктивною категорією, яка відображає сукупність знань, умінь і навичок, що відповідають певному рівню і є визнаними суспільством. Ці знання та вміння повинні бути практично застосовані в професійній діяльності, забезпечуючи ефективність виконання завдань і досягнення поставлених цілей у відповідній сфері. Компетентність є важливим чинником, який дозволяє фахівцю адаптуватися до вимог ринку праці та забезпечувати високий рівень професійної діяльності [13, с.41].

З урахуванням сучасних тенденцій і процесів, що стосуються розвитку професійної підготовки фахівців у галузі цифрових технологій, а також особистісних потреб у підготовці кваліфікованих фахівців, можна визначити важливу проблему: підвищення ефективності професійної підготовки в цій галузі, зокрема у контексті розробки навчальних ресурсів для програмування чисельних методів розв’язання задач диференціального та інтегрального числення з використанням мови Python.

Таким чином, професійна підготовка фахівців у галузі цифрових технологій, орієнтована на розробку навчальних ресурсів для програмування чисельних методів розв’язання задач диференціального та інтегрального числення мовою Python, вимагає вирішення важливої проблеми. Ми вбачаємо її в необхідності теоретичного обґрунтування методики професійної підготовки, яка б ефективно відповіла на потреби сучасного ринку праці та забезпечила високий рівень компетентності майбутніх фахівців.

РОЗДІЛ 2
ХАРАКТЕРИСТИКА ОБ'ЄКТІВ ГАЛУЗІ: СТАН І СТРАТЕГІЇ РОЗВИТКУ

2.1 Особливості програмування числових методів мовою Python, переваги та недоліки

Python широко застосовується для розв'язання задач лінійної алгебри, оптимізації, інтегрування, диференціювання та роботи з великими обсягами даних. Програмування числових методів мовою Python має свої особливості завдяки її потужним бібліотекам, простоті використання та ефективності у розв’язанні задач чисельного аналізу. Наведемо деякі особливості.

1. Бібліотеки для чисельних розрахунків

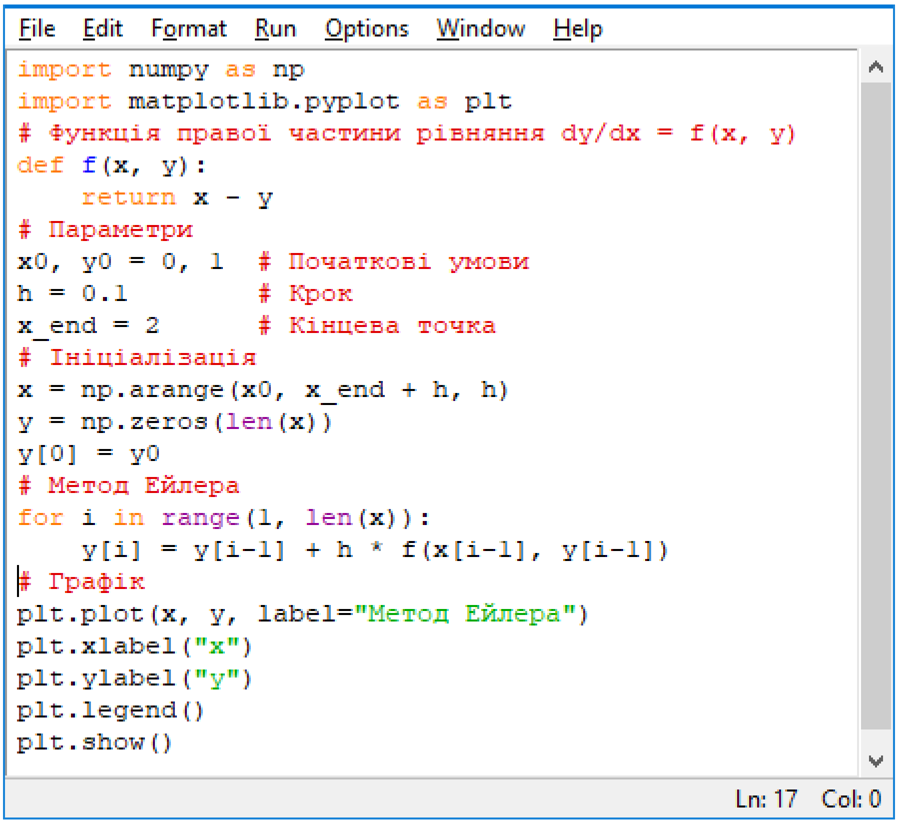
Python має великий набір спеціалізованих бібліотек, які значно полегшують реалізацію числових методів:

- NumPy: базова бібліотека для роботи з багатовимірними масивами, лінійною алгеброю, математичними функціями.
- SciPy: розширює можливості NumPy для оптимізації, інтеграції, роботи з диференціальними рівняннями та іншими задачами.
- SymPy: забезпечує символьні обчислення, що може бути корисним для аналітичної перевірки числових результатів.
- Matplotlib та Seaborn: дозволяють візуалізувати результати числових розрахунків.
- Pandas: зручний інструмент для роботи з таблицями даних та їх попередньої обробки перед числовим аналізом.

2. Простота реалізації

- Python дозволяє швидко реалізувати числові методи, завдяки своєму простому синтаксису.
- Велика кількість вбудованих функцій зменшує обсяг коду.

Наприклад, реалізація методу Ейлера для числового розв'язання звичайного диференціального рівняння. Код програми показано на рис. 2.1.



```
import numpy as np
import matplotlib.pyplot as plt
# функція правої частини рівняння dy/dx = f(x, y)
def f(x, y):
    return x - y
# Параметри
x0, y0 = 0, 1  # Початкові умови
h = 0.1        # Крок
x_end = 2      # Кінцева точка
# Ініціалізація
x = np.arange(x0, x_end + h, h)
y = np.zeros(len(x))
y[0] = y0
# Метод Ейлера
for i in range(1, len(x)):
    y[i] = y[i-1] + h * f(x[i-1], y[i-1])
# Графік
plt.plot(x, y, label="Метод Ейлера")
plt.xlabel("x")
plt.ylabel("y")
plt.legend()
plt.show()
```

Рис. 2.1 Код програми, що реалізує метод Ейлера для числового розв'язання звичайного диференціального рівняння

3. Ефективність і масштабованість

– Python може здаватися повільним через інтерпретований характер, але спеціалізовані бібліотеки (NumPy, SciPy) реалізовані на мові C, що забезпечує високу швидкість виконання обчислень.

– Підтримка багатопоточності та розподілених обчислень через бібліотеки (наприклад, Dask, Joblib).

4. Робота з великими даними

– Python зручно працює з великими наборами даних завдяки Pandas та NumPy.

– Для оптимізації обчислень використовуються інструменти, як-от Numba (компіляція Python-коду) або Cython.

5. Точність обчислень. Python дозволяє налаштовувати точність числових обчислень:

- Використання спеціалізованих типів даних, наприклад, decimal для точних обчислень.
- Контроль над кроком, ітераціями та умовами збіжності.

6. Реалізація складних методів. Python дозволяє легко реалізовувати числові методи будь-якого рівня складності.

Приклад. Розв’язання систем лінійних алгебраїчних рівнянь (код програми наведено на рис. 2.2)

```python
import numpy as np
A = np.array([[3, 2], [1, 2]])
b = np.array([18, 14])
x = np.linalg.solve(A, b)   # Розв'язок
print(x)
```

Рис. 2.2 Код програми, що розв’язує систему лінійних алгебраїчних рівнянь

Приклад. Числове інтегрування (код програми наведено на рис. 2.3)

```python
from scipy.integrate import quad
# Інтегрування функції f(x) = x^2 на [0, 1]
result, _ = quad(lambda x: x**2, 0, 1)
print(result)
```

Рис. 2.3 Код програми, що знаходить значення визначеного інтеграла

7. Візуалізація

Python підтримує візуалізацію розрахунків, що є важливим для аналізу результатів:

- Відображення графіків, векторних полів, розв'язків рівнянь.
- Інтерактивна побудова графіків через бібліотеки, як-от Plotly.

Переваги використання Python для числових методів

- Простота реалізації навіть для складних задач.
- Широкий вибір бібліотек для різних потреб.
- Ефективність завдяки використанню високопродуктивних бібліотек.
- Інтерактивність і можливість швидкого прототипування.

Недоліки

- Нижча швидкість порівняно з C++ чи Fortran у задачах з високою обчислювальною інтенсивністю.
- Залежність від зовнішніх бібліотек.

Python є чудовим вибором для числових методів, особливо якщо потрібна швидка розробка, інтерактивність і багатство інструментів.

2.2 Числові методи розв'язання задач диференціального числення та їх реалізація мовою Python

Чисельне диференціювання – це метод обчислення похідних функцій у точках за допомогою чисельних методів, особливо коли аналітична форма похідної недоступна або функція відома лише в дискретних точках.

Найбільш прості формули чисельного диференціювання отримуються тоді, коли функція $f(x)$ може бути задана таблицею своїх значень $y_k=f(x_k)$ у рівновіддалених вузлах $x_k = x_0 + kh$, $k = 0, 1, 2, \ldots, N$. Величину h при цьому називають кроком чисельного диференціювання. [10] Для обчислення значення похідної застосовують наближену рівність

$$y' \approx \frac{f(x+\Delta x)-f(x)}{\Delta x} = \frac{\Delta y}{\Delta x}.$$

Основні підходи до чисельного диференціювання базуються на використанні скінченних різниць. Нижче наведено найпоширеніші формули.

1. Формули скінченних різниць

Одностороння різниця (вперед): $f'(x) \approx \frac{f(x+h)-f(x)}{h}$, *де* $h - \text{крок}$.

Одностороння різниця (назад): $f'(x) \approx \dfrac{f(x) - f(x-h)}{h}$

Центральна різниця: $f'(x) \approx \dfrac{f(x+h) - f(x-h)}{2h}$

Центральна різниця є більш точною, оскільки похибка першого порядку O(h^2).

2. Формули для другої похідної

Центральна різниця: $f''(x) \approx \dfrac{f(x+h) - 2f(x) + f(x-h)}{h^2}$.

3. Точність

Точність чисельного диференціювання залежить від вибору кроку h/

Занадто великий h збільшує похибку апроксимації.

Занадто малий h підвищує чутливість до похибок округлення.

Оптимальний h зазвичай обирається емпірично або через аналіз похибок.

4. Похибка

Основні джерела похибок:

– похибка апроксимації пов’язана з нехтуванням вищими членами ряду Тейлора;

– похибка округлення виникає через обмежену точність чисел у комп'ютерних обчисленнях.

5. Практичне застосування

Чисельне диференціювання часто використовується при розв’язанні задач, де функція визначена експериментальними даними; для апроксимації похідних у чисельному моделюванні (наприклад, у методі скінченних елементів).

Приклад 1. Чисельне диференціювання функції $y = \sin(x)$ мовою Python. Код програми наведений на рис. 2.4, результат виконання програми на рис. 2.5.

```python
import numpy as np
import matplotlib.pyplot as plt

# Визначаємо функцію
def func(x):
    return np.sin(x)

# Параметри
x = np.linspace(0, 2 * np.pi, 1000)  # Точки для графіку функції
h = 0.01  # Крок для чисельного диференціювання

# Чисельне диференціювання
forward_diff = (func(x + h) - func(x)) / h  # Одностороння (вперед)
backward_diff = (func(x) - func(x - h)) / h  # Одностороння (назад)
central_diff = (func(x + h) - func(x - h)) / (2 * h)  # Центральна різниця

# Точна похідна
exact_derivative = np.cos(x)

# Графіки
plt.figure(figsize=(10, 6))
plt.plot(x, exact_derivative, label="Точна похідна (cos(x))", color="black", linewidth=2)
plt.plot(x, forward_diff, label="Одностороння (вперед)", linestyle="--")
plt.plot(x, backward_diff, label="Одностороння (назад)", linestyle="--")
plt.plot(x, central_diff, label="Центральна різниця", linestyle=":")

plt.xlabel("x")
plt.ylabel("Похідна")
plt.title("Чисельне диференціювання функції sin(x)")
plt.legend()
plt.grid()
plt.show()
```

Рис. 2.4 Код програми чисельного диференціювання функції $y = \sin(x)$

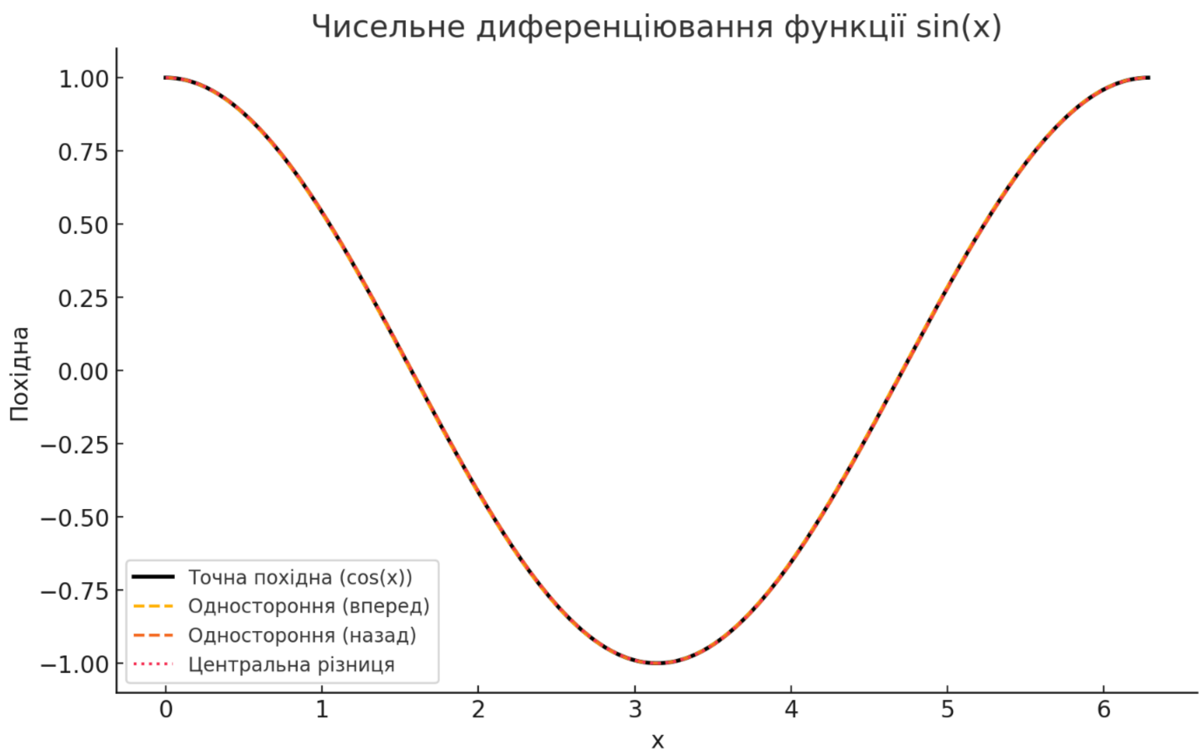


Рис. 2.5 Результат виконання програми з прикладу 1

Приклад 2. Чисельне диференціювання таблично заданої функції $y = \sin(x)$ мовою Python.

Код програми наведений на рис. 2.6.

```
import numpy as np
import matplotlib.pyplot as plt

# Таблично задані точки
x_table = np.linspace(0, 2 * np.pi, 10)  # Значення x
y_table = np.sin(x_table)  # Відповідні значення y = sin(x)

# Інтерполяція для більш щільної сітки
x = np.linspace(0, 2 * np.pi, 1000)  # Точки для графіку
h = 0.01  # Крок для чисельного диференціювання

# Інтерполяція значень функції
y_interpolated = np.interp(x, x_table, y_table)

# Чисельне диференціювання
# Одностороння (вперед)
forward_diff = (np.interp(x + h, x_table, y_table, left=0, right=0) - y_interpolated) / h
# Одностороння (назад)
backward_diff = (y_interpolated - np.interp(x - h, x_table, y_table, left=0, right=0)) / h
# Центральна різниця
central_diff = (np.interp(x + h, x_table, y_table, left=0, right=0) - \
                np.interp(x - h, x_table, y_table, left=0, right=0)) / (2 * h)

# Точна похідна для порівняння
exact_derivative = np.cos(x)

# Графіки
plt.figure(figsize=(10, 6))
plt.plot(x, exact_derivative, label="Точна похідна (cos(x))", color="black", linewidth=2)
plt.plot(x, forward_diff, label="Одностороння (вперед)", linestyle="--")
plt.plot(x, backward_diff, label="Одностороння (назад)", linestyle="--")
plt.plot(x, central_diff, label="Центральна різниця", linestyle=":")

# Відобразити таблично задані точки
plt.scatter(x_table, y_table, color="red", label="Табличні точки", zorder=5)

plt.xlabel("x")
plt.ylabel("Похідна")
plt.title("Чисельне диференціювання для таблично заданої функції")
plt.legend()
plt.grid()
plt.show()
```

Рис. 2.6 Код програми чисельного диференціювання таблично заданої функції

Результат виконання програми показано на рис 2.7. Графік показує

Точну похідну ($\cos(x)$) – чорна лінія.

Чисельні похідні (вперед, назад і центральна різниця) – штрихові та пунктирні лінії.

Таблично задані точки (x, y) – червоні точки.

Це ілюструє обчислення похідних для таблично заданої функції з використанням інтерполяції.

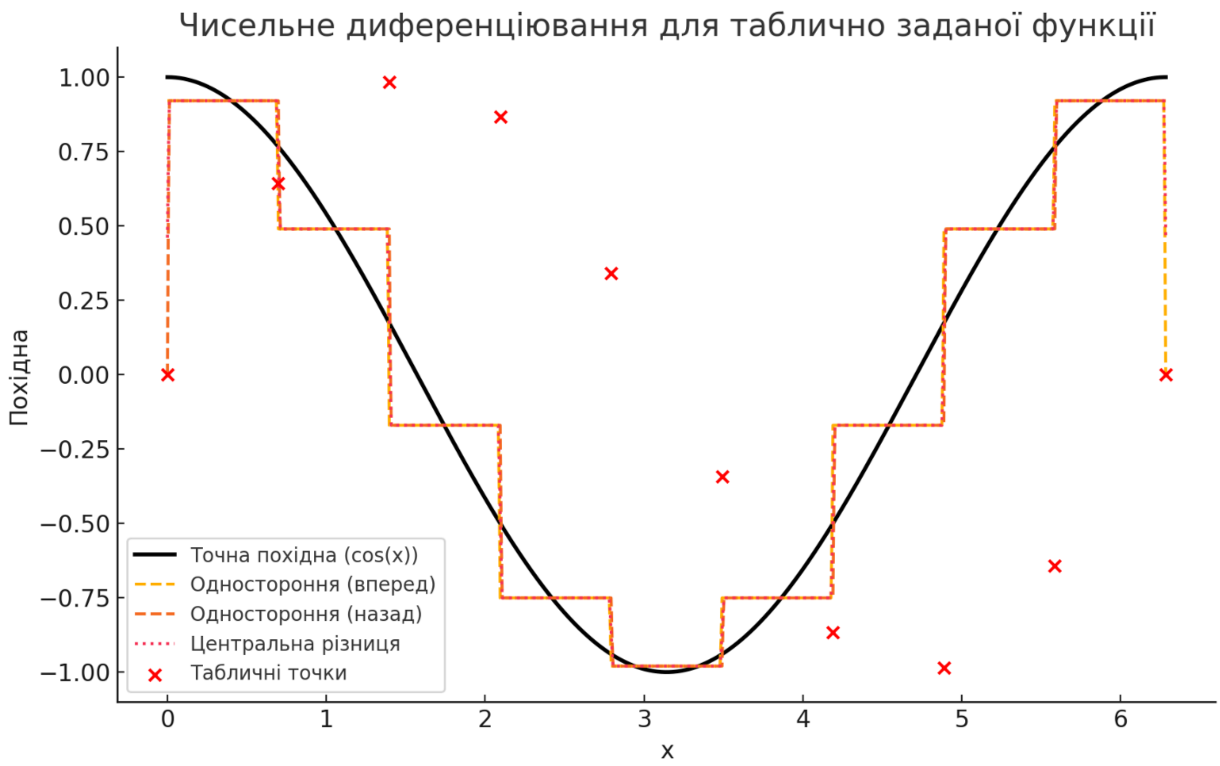


Рис. 2.7 Результат виконання програми з прикладу 2

2.3 Числові методи розв’язання задач інтегрального числення та їх реалізація мовою Python

Задачі, які зводяться до обчислення визначених інтегралів, охоплюють широкий спектр прикладних і теоретичних проблем. Наприклад, у фізиці обчислення визначеного інтеграла часто використовується для знаходження площі під кривою, яка може бути інтерпретована як виконана робота при змінному силовому впливі. У механіці, визначений інтеграл допомагає розраховувати масу тонкого дроту або пластини з нерівномірною густиною, коли густина описується функцією.

У геометрії, інтеграли дозволяють визначити площу криволінійних фігур, об'єм тіл обертання, а також довжину дуги кривої. В екології чи економіці вони використовуються для визначення загального накопичення ресурсу або витрат, якщо швидкість накопичення чи витрат задається функцією. У статистиці визначені інтеграли потрібні для роботи з неперервними випадковими величинами, наприклад, для розрахунку ймовірностей подій на основі щільності розподілу.

В теорії диференціальних рівнянь, визначені інтеграли застосовуються для знаходження розв’язків, зокрема в задачах про рух рідин чи теплообмін. Всі ці задачі об’єднує те, що інтегрування дає змогу отримати кількісну характеристику на основі функціонального опису явищ чи процесів.

Якщо функція $f(x)$ неперервна на відрізку $[a, b]$ й відома її первісна $F(x)$, то визначений інтеграл від a до b може бути обчислений за формулою Ньютона- Лейбніца

$$I = \int_a^b f(x)dx = F(b) - F(a),$$

Де $F'(x) = f(x)$.

Але не для кожної елементарної функції $f(x)$ первісна є теж елементарною. Поширеними також є ситуації, коли підінтегральна функція подається графіком або таблицею експериментально одержаних значень. У всіх цих випадках не можна скористатися формулою Ньютона-Лейбніца і тому вдаються до чисельних методів інтегрування. Вони дозволяють знайти значення інтегралу безпосередньо по значеннях підінтегральної функції $f(x)$ і не залежать від способу її подання. Суть чисельних методів полягає в заміні підінтегральної функції допоміжною, інтеграл від якої легко обчислюється в елементарних функціях. Найчастіше $f(x)$ замінюється деяким інтерполяційним многочленом, що призводить до так званих квадратурних формул:

$$\int_a^b f(x)dx = \sum_{i=0}^{n} A_i f(x_i) + R_n f(x),$$

де x_i – вузли інтерполяції, A_i – коефіцієнти, $R_n(f)$ – залишковий член або похибка методу. Відкидання $R_n(f)$ призводить до похибки відсікання. Під час обчислень до неї додаються похибки округлення.

Числові методи розв’язання задач інтегрального числення використовуються, коли аналітичне знаходження інтегралів неможливе або надто складне. Основні задачі, які вирішуються числовими методами, це:

– обчислення визначених інтегралів;

– наближення розв'язку інтегральних рівнянь.

Відомо, що $\int_a^b f(x)dx$ при $f(x) \geq 0$ дорівнює площі криволінійної трапеції, обмеженної графіком функції $y = f(x)$, віссю абсцис і прямими $x = a,\ x = b$.

Використання для обчислення значення інтегралу квадратурної формули полягає в наступному. Відрізок інтегрування $[a,b]$ розбивається на n частин $[x_i, x_{i+1}]$, $i = \overline{0, n-1}$, і наближено обчислюються значення площ, що відповідають кожному відрізку. Сума цих площ і дає наближене значення інтегралу.

В залежності від способу розбиття відрізку інтегрування $[a,b]$ системою точок (вузлів інтерполяції) x_i, $i = \overline{0,n}$, розрізняють два підходи до побудови квадратурних формул.

При першому підході місцеположення і довжина інтервалів розбиття обираються заздалегідь, на початку обчислень. Для рівновіддалених точок

$$x_i = x_0 + ih, \quad h = \frac{b-a}{n}, \quad x_0 = a, \quad x_n = b, \quad i = \overline{0,n}$$

квадратурні формули називаються формулами Ньютона-Котеса. Вони розрізняються степенями використаних інтерполяційних многочленів. Широко застосовуються формули прямокутників, трапецій і Сімпсона.

При другому підході місцеположення і довжина інтервалів обираються таким чином, щоб досягти найвищої точності при заданому числі інтервалів n (формула Гаусса).

Розглянемо основні числові методи для обчислення визначених інтегралів.

2.2.1 Метод прямокутників.

Квадратурна формула прямокутників випливає з геометричного змісту визначеного інтеграла, як площі криволінійної трапеції. Для знаходження площі цієї трапеції розіб'ємо її основу (відрізок інтегрування $[a,b]$) точками $a = x_0 < x_1 < \ldots < x_n = b$ на n рівних частин довжиною $h = \dfrac{b-a}{n}$. Через ці точки проведемо прямі $x = x_i,\ i = \overline{0,n}$, паралельно осі OY. Як результат, криволінійна трапеція розіб'ється на n смужок шириною h. Кожну смужку наближено замінемо прямокутниками висотою $f(\xi_i)$ де ξ_i – довільна точка на відрізку $[x_{i-1}, x_i],\ i = \overline{1,n}$ Площі прямокутників рівні $S_i = f(\xi_i)h,\ i = \overline{1,n}$ Підсумовуючи площі всіх прямокутників, одержимо наближене значення інтеграла:

$$\int_a^b f(x)dx \approx \frac{b-a}{n}(f(\xi_1) + f(\xi_2) + \ldots + f(\xi_n)).$$

Поклавши $\xi_i = x_{i-1}$, або $\xi_i = x_i$, або $\xi_i = \dfrac{x_{i-1} + x_i}{2}$, отримуємо три формули прямокутників (лівих, правих і середніх):

$$\int_a^b f(x)dx \approx I_l = \frac{b-a}{n}(f(x_0) + f(x_1) + \ldots + f(x_{n-1})) = \frac{b-a}{n}\sum_{k=0}^{n-1} f(x_k);$$

$$\int_a^b f(x)dx \approx I_r = \frac{b-a}{n}(f(x_1)+f(x_2)+...+f(x_n)) = \frac{b-a}{n}\sum_{k=1}^{n} f(x_k);$$

$$\int_a^b f(x)dx \approx I_m = \frac{b-a}{n}\left(f\left(x_0+\frac{h}{2}\right)+f\left(x_1+\frac{h}{2}\right)+...+f\left(x_{n-1}+\frac{h}{2}\right)\right)=$$

$$=\frac{b-a}{n}\sum_{k=0}^{n-1} f\left(x_k+\frac{h}{2}\right)=\frac{b-a}{n}\sum_{k=0}^{n-1} f\left(a+\frac{2k-1}{2}h\right).$$

Всі формули прямокутників дають точний результат, якщо функція $f(x)$ стала на $[a,b]$, а формула середніх прямокутників дає точний результат і у випадку, коли підінтегральна функція є лінійною на $[a,b]$, або довільна, але симетрична відносно середини відрізка $[a,b]$.

Алгоритм цього методу представлений на рисунку 2.8

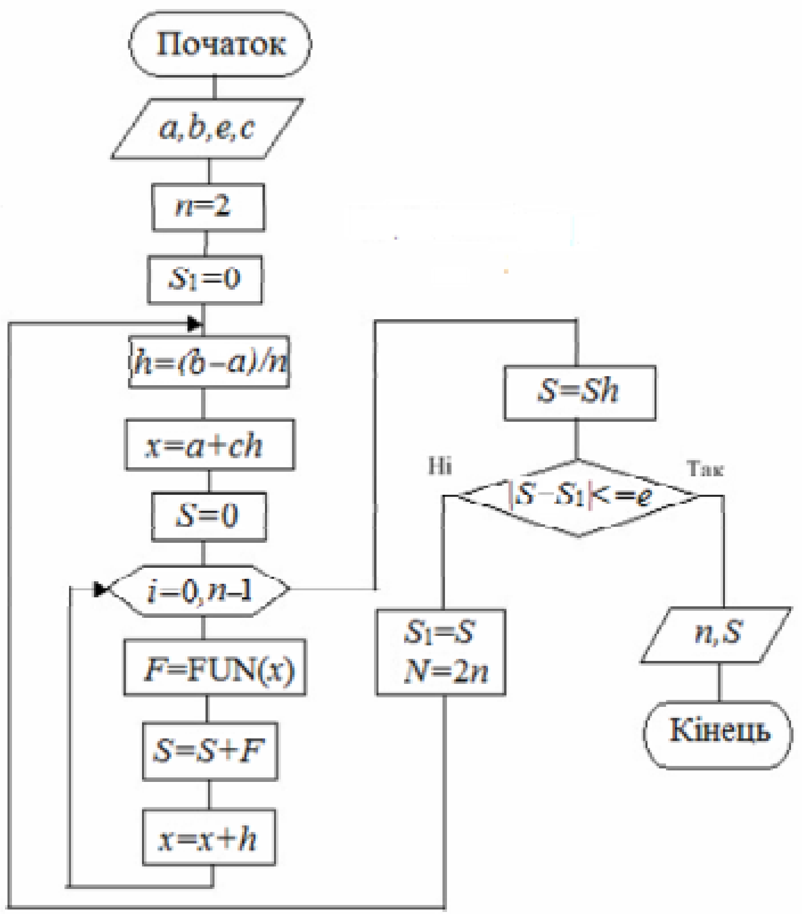


Рис. 2.8 Блок-схема алгоритму методу прямокутників

В алгоритмі введено такі позначення (рис. 2.8):

a, b – кінці інтервалу;

e – задана точність;

c = 0 – метод лівих прямокутників;

c = 1 – метод правих прямокутників;

S_1 – значення інтеграла на попередньому кроці;

S – значення інтеграла на поточному кроці.

На практиці для оцінки похибки часто обчислюють інтеграл з кроком h і з кроком $\frac{h}{2}$ і вважають, що співпадаючі десяткові знаки результату є точними. При необхідності n подвоюють. Можна також скористатись правилом Рунге для оцінки цієї похибки:

$$\Delta \approx \frac{|I_n - I_{2n}|}{3},$$

де I_n – значення інтеграла, обчисленого з кроком h, а I_{2n} – з кроком $\frac{h}{2}$.

Приклад 3. Знайти значення визначеного інтегралу $\int_0^1 \sqrt{2x^2+1}\,dx$ за допомогою методів лівих, правих і середніх прямокутників. Реалізувати розв’язання мовою програмування Python.

Код програми наведений на рис. 2.9, результат виконання програми – на рис. 2.10.

```python
import math
def integral_pryam (f, a, b, n, c=1/2):
    'metod pryamokutnykiv livi: c=0, pravi: c=1, seredni: c=1/2'
    m = n//2
    h = (b - a)/m
    s1 = 0
    x = a + c*h
    for k in range(m):
        s1 = s1 + h * f(x)
        x += h
    h = (b - a)/n
    s2 = 0
    x = a + c*h
    for k in range(n):
        s2 = s2 + h * f(x)
        x += h
    d = abs(s2-s1)/3
    return s2, d
def f(x):
    return math.sqrt(2 * x**2 + 1)
integral, pohybka = integral_pryam(f, 0, 1, 20)
print('s_integral =', '%f' % integral, '\tpohybka =', '%f' % pohybka)
integral, pohybka = integral_pryam (f, 0, 1, 20, 0)
print('l_integral =', '%f' % integral, '\tpohybka =', '%f' % pohybka)
integral, pohybka = integral_pryam(f, 0, 1, 20, 1)
print('p_integral =', '%f' % integral, '\tpohybka =', '%f' % pohybka)
```

Ln: 26   Col: 69

Рис. 2.9 Код програми, що реалізує метод прямокутників для прикладу 3

```
s_integral = 1.271154   pohybka = 0.000120
l_integral = 1.253213   pohybka = 0.005860
p_integral = 1.289816   pohybka = 0.006341
```

Рис. 2.10 Результат виконання коду

2.2.2 Метод трапецій.

Цей метод використовує трапецієподібну апроксимацію під графіком функції. Квадратурна формула трапецій також випливає з геометричного змісту визначеного інтеграла, як площі криволінійної трапеції. Для знаходження площі цієї трапеції розіб'ємо її основу (відрізок інтегрування $[a,b]$) точками $a = x_0 < x_1 < \ldots < x_n = b$ на n рівних частих довжиною $h = \dfrac{b-a}{n}$. Через ці точки проведемо прямі $x = x_i,\ i = \overline{0,n}$ паралельно осі Oy. Як результат, криволінійна трапеція розіб'ється на n смужок шириною h. Кожну смужку наближено замінемо прямолінійною трапецією з основами $y_i = f(x_i)$, $y_{i+1} = f(x_{i+1})$ і висотою h $(i = \overline{0,n-1})$.

Площі прямолінійних трапецій дорівнюють $S_i = \frac{y_i + y_{i+1}}{2} h, \; i = \overline{0, n-1}$.

Підсумовуючи площі всіх прямолінійних трапецій, отримаємо наближене значення визначеного інтеграла:

$$\int_a^b f(x)dx \approx \sum_{i=0}^{n-1} S_i = \frac{y_0 + y_1}{2} h + \frac{y_1 + y_2}{2} h + \ldots + \frac{y_{n-1} + y_n}{2} h =$$

$$= h \cdot \left(\frac{y_0}{2} + y_1 + \ldots + y_{n-1} + \frac{y_n}{2} \right) = h \cdot \left(\frac{f(a) + f(b)}{2} + \sum_{k=1}^{n-1} f\left(a + k \cdot h\right) \right). \qquad (8.3)$$

Для залишкового члена квадратурної формули трапецій справедливі такі оцінки.

1. Якщо $f(x)$ неперервна і має кусково-неперервну $f'(x)$ на $[a,b]$ і $M_1 = \max_{x \in [a,b]} \left| f'(x) \right|$, то

$$\left| R_n(f) \right| \le \frac{(b-a)^2}{4n} M_1.$$

2. Якщо $f(x)$ має кусково-неперервну $f''(x)$ на $[a,b]$ і $M_2 = \max_{x \in [a,b]} \left| f''(x) \right|$, то

$$\left| R_n(f) \right| \le \frac{(b-a)^3}{12n^2} M_2.$$

Блок-схему алгоритму методу трапецій наведено на рис. 2.11.

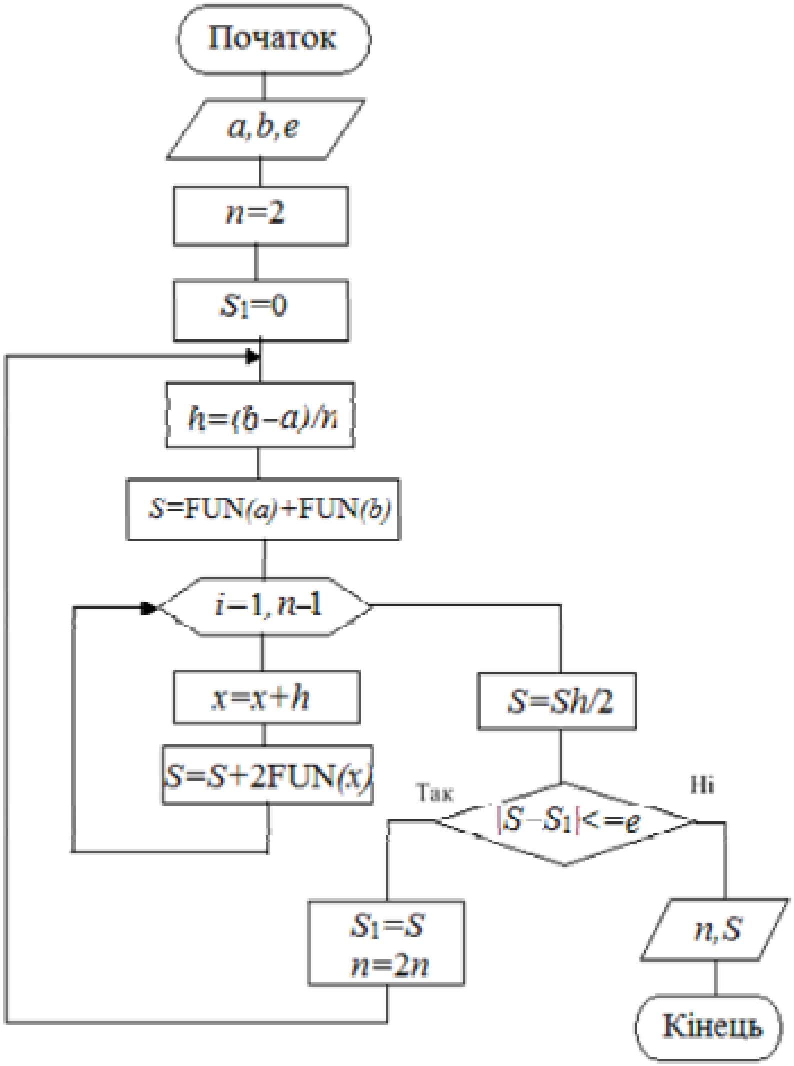


Рис. 2.11 Блок-схема алгоритму методу трапецій

Якщо треба обчислити інтеграл з похибкою не більшою ε, то необхідна кількість частин n розбиття інтервалу інтегрування знаходиться із нерівності

$$n \geq \frac{(b-a)^2 M_1}{4\varepsilon}$$

в першому випадку та із нерівності

$$n \geq \sqrt{\frac{(b-a)^3 M_2}{12\varepsilon}}$$

в другому випадку.

Звичайно n вибирається не меншим того, яке дає ця формула, і таким, щоб було зручно обчислювати величини $y_0, y_1, \ldots, y_n$.

На практиці для оцінки похибки часто користуються правилом Рунге.

Приклад 4. Знайти значення визначеного інтегралу $\int_0^1 \sqrt{2x^2+1}\,dx$ за допомогою методу трапецій. Реалізувати розв’язання мовою програмування Python.

Код програми, що реалізує метод трапецій для обчислення визначеного інтегралу мовою Python наведений на рис. 2.12, результат виконання програми на рис. 2.13.

```
File  Edit  Format  Run  Options  Window  Help
import math
def integral_pryam (f, a, b, n):
    'metod trapetsiy'
    m = n//2
    h = (b - a)/m
    s1 = (f(a) + f(b))*h/2
    x = a + h
    for k in range(m-1):
        s1 = s1 + h * f(x)
        x += h
    h = (b - a)/n
    s2 =(f(a) + f(b))*h/2
    x = a + h
    for k in range(n-1):
        s2 = s2 + h * f(x)
        x += h
    d = abs(s2 - s1)/3
    return s2, d
def f(x):
    return math.sqrt(2 * x**2 + 1)
integral, pohybka = integral_pryam (f, 0, 1, 20)
print ('integral_trp␣=', '%f' % integral, ' pohybka␣=', '%f' % pohybka)
                                                        Ln: 22  Col: 70
```

Рис. 2.12 Код програми, що реалізує метод трапецій для прикладу 4

```
integral_trp = 1.271514 pohybka = 0.000241
```

Рис. 2.13 Результат виконання програми

2.2.3 Метод Сімпсона (параболічне наближення).

Метод заснований на апроксимації функції на кожному підінтервалі параболами.

При парній кількості інтервалів розбиття $n = 2m$ і заміні підінтегральної функції на кожних двох суміжних частинах параболою, що проходить через точки $(x_{2i}, y_{2i}),(x_{2i+1}, y_{2i+1}),(x_{2i+2}, y_{2i+2}),\ i = \overline{0, m-1}$ квадратурна формула набуває вигляду

$$\int_a^b f(x)dx \approx \frac{h}{3}\left(f(a) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \ldots + 2f(x_{n-2}) + 4f(x_{n-1}) + f(b)\right) =$$

$$= \frac{h}{3}\cdot\left(f\left(a\right) + f\left(b\right) + 4\sum_{k=1}^{\frac{n}{2}} f\left(a + \left(2k-1\right)\cdot h\right) + 2\sum_{k=1}^{\frac{n}{2}-1} f\left(a + 2k\cdot h\right)\right)$$

Цей вираз отримав назву формули Сімпсона або формули парабол. Значення функції $f(x)$ у вузлах з непарними номерами $x_1, x_3, \ldots, x_{n-1}$ містяться у формулі з коефіцієнтом 4, у вузлах з парними номерами $x_2, x_4, \ldots, x_{n-2}$ – з коефіцієнтом 2, а в точках $x_0 = a$, $x_n = b$ – з коефіцієнтом 1.

Геометричний зміст формули Сімпсона полягає в наступному. Через три послідовні точки з ординатами $y_{i-1}, y_i, y_{i+1},\ i = \overline{1, n-1}$ розбиття проводиться парабола, і визначається площа одержаної фігури. Сума площ всіх побудованих таким чином фігур дає наближене значення інтегралу.

Якщо $f(x)$ має кусково-неперервну $f''(x)$ на $[a,b]$ і $M_2 = \max\limits_{x\in[a,b]}\left| f''(x)\right|$, то для залишкового члена квадратурної формули Сімпсона справедлива оцінка

$$\left|R_n(f)\right| \le \frac{(b-a)^3}{81n^2} M_2.$$

При наявності на відрізку $[a,b]$ неперервної похідної четвертого порядку підінтегральної функції оцінка похибки відсікання визначається формулою

$$\left|R_n(f)\right| \le \frac{(b-a)^5}{180n^4} M_4, \; M_4 = \max_{x\in[a,b]} \left| f^{(4)}(x) \right|.$$

Підкреслимо, що формула Сімпсона є точною для поліномів до третього степеня включно, оскільки для них $f^{(4)}(x) = 0$.

Блок-схему алгоритму методу Сімпсона наведено на рис. 2.14.

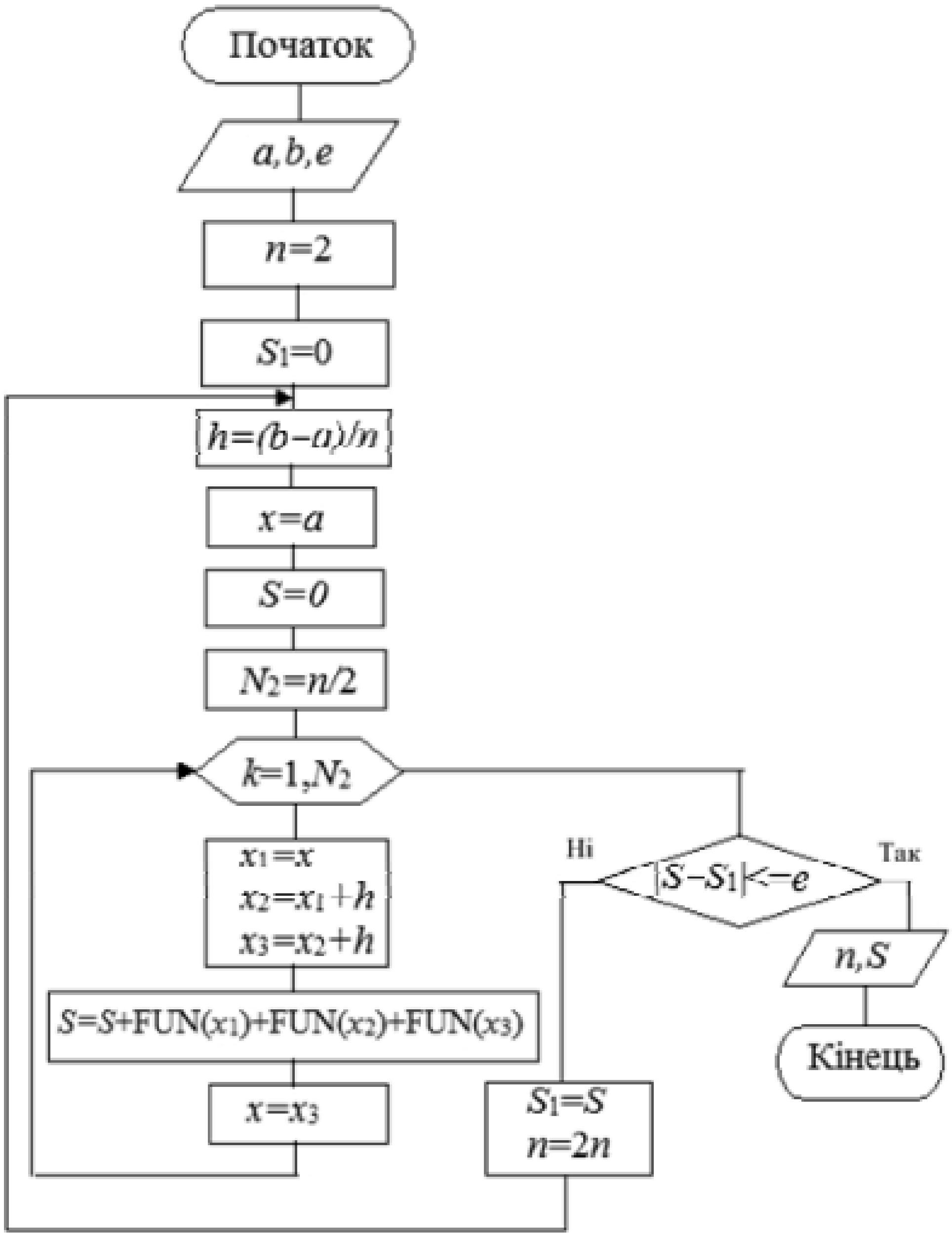


Рис. 2.14 Блок-схема алгоритму методу Сімпсона

На практиці для оцінки похибки часто обчислюють інтеграл з кроком h і з кроком $\frac{h}{2}$ і вважають, що співпадаючі десяткові знаки результату є точними. При необхідності n подвоюють. Можна також скористатисть правилом Рунге для оцінки цієї похибки:

$$\Delta \approx \frac{|I_n - I_{2n}|}{15},$$

де I_n – значення інтеграла, обчисленого з кроком h, а I_{2n} – з кроком $\frac{h}{2}$.

Приклад 5. Знайти значення визначеного інтегралу $\int_0^1 \sqrt{2x^2+1}\,dx$ за допомогою метода Сімпсона. Реалізувати розв'язання мовою програмування Python.

Код програми, що реалізує метод Сімпсона для обчислення визначеного інтегралу мовою Python наведений на рис. 2.15, результат виконання програми на рис. 2.16.

```
File  Edit  Format  Run  Options  Window  Help
import math
def simpson (f, a, b, delta=0.000001):
    def simp(f, a, b, n):
        h = (b - a)/n
        m = n//2
        s = f(a) + f(b)+ 4*f(b-h)
        x = a
        for i in range(m-1):
            s = s + 4*f(x+h) + 2*f(x+2*h)
            x = x + 2*h
        s = s*h/3
        return s
    d, n = 1, 1
    while abs(d) > delta:
        d = (simp(f, a, b, n*2) - simp(f, a, b, n))/15
        n *= 2
    integral = abs(simp(f, a, b, 2*n))
    print('Simpson: n      integral        pohybka')
    print('\t%s\t%f\t%f' % (2*n, integral, abs(d)))
def f(x):
    return math.sqrt(2 * x**2 + 1)
simpson(f, 0, 1, 0.0001)
                                              Ln: 23   Col: 0
```

Рис. 2.15 Код програми, що реалізує метод Сімпсона для прикладу 5

```
Simpson: n         integral          pohybka
         8         1.271273          0.000038
```

Рис. 2.16 Результат виконання програми

2.2.4 Метод Гаусса (квадратурні формули).

Цей метод обчислює інтеграл через лінійні комбінації значень функції в спеціально обраних точках (вузлах Гаусса).

Суть формул Гаусса полягає в наступному: при заданій кількості інтервалів розбиття треба розмістити їх кінці так, щоб одержати найвищу точність інтегрування.

В математичному плані це означає вибір коефіцієнтів A_i вузлів t_i, $(i=\overline{1,n})$ квадратурних формул Гаусса

$$\int_{-1}^{1} f(x)dx = \sum_{i=1}^{n} A_i f(t_i) + R_n(f)$$

такими, щоб ці формули були точними для многочленів найвищого можливого степеня N. Можна довести, що при n вузлах точно інтегруються всі многочлени степеня $N \le 2n-1$.

Вузли t_i є коренями многочлена Лежандра

$$P_n(t) = \frac{1}{2^n n!} \frac{d^n (t^2-1)^n}{dt^n}.$$

Коефіцієнти A_i, що називаються ваговими коефіцієнтами, обчислюються за формулою

$$A_i = \frac{2}{(1-t_i^2)(P_n'(t_i))^2}, \; i=\overline{1,n}.$$

Похибку формули, яка визначається залишковим членом $R_n(f)$, можна оцінити по виразу

$$R_n(f) = \frac{2^{2n+1}}{(2n+1)(2n)!} \left(\frac{(n!)^2}{(2n)!} \right)^2 f^{(2n)}(\xi), \; \xi \in [-1,1].$$

При обчисленні $\int_a^b f(x)dx$ відрізок $[a,b]$ перетворюється у відрізок $[-1,1]$ заміною змінної

$$x = \frac{a+b}{2} + \frac{b-a}{2} t \, .$$

В результаті формула Гаусса набуває вигляду

$$\int_a^b f(x)dx = \frac{b-a}{2} \sum_{i=1}^{n} A_i f(x_i) + \overline{R}_n(f),$$

де $x_i = \frac{a+b}{2} + \frac{b-a}{2} t_i$, $\overline{R_n}(f) = \left(\frac{b-a}{2} \right)^{2n+1} R_n(f)$, або

$$\int_a^b f(x)dx \approx \frac{b-a}{2} \sum_{i=1}^{n} A_i f(x_i) = \frac{b-a}{2} \sum_{i=1}^{n} A_i f\left(\frac{a+b}{2} + \frac{b-a}{2} \cdot t_i \right)$$

Квадратурна формула Гаусса забезпечує високу точність інтегрування при невеликій кількості вузлів.

Для вузлів t_i і коефіцієнтів A_i складені таблиці, які можна знайти у довідниках.

Приклад 6. Знайти значення визначеного інтегралу $\int_0^1 \sqrt{2x^2+1}\, dx$ за допомогою метода Гаусса. Реалізувати розв'язання мовою програмування Python.

Код програми на мові Python представлено на рис. 2.17.

Значення визначеного інтегралу $\int_0^1 \sqrt{2x^2+1}\, dx$ обчислене за допомогою методу Гаусса з трьома вузлами, дорівнює приблизно 1.2713.

File Edit Format Run Options Window Help

```
import numpy as np
# Function to integrate
def f(x):
    return np.sqrt(2 * x**2 + 1)
# Gauss-Legendre nodes and weights for n = 3
nodes = np.array([-np.sqrt(3/5), 0, np.sqrt(3/5)])
weights = np.array([5/9, 8/9, 5/9])
# Limits of integration
a, b = 0, 1
# Transforming the nodes to the interval [a, b]
transformed_nodes = 0.5 * (b - a) * nodes + 0.5 * (b + a)
transformed_weights = 0.5 * (b - a) * weights
# Computing the integral
integral = sum(w * f(x) for x, w in zip(transformed_nodes, transformed_weights))
integral
```

Ln: 13 Col: 0

Рис. 2.17 Код програми на мові Python для прикладу 6.

Щоб оцінити похибку інтегрування, можна порівняти числовий результат з аналітичним значенням інтегралу (якщо воно відоме) або скористатися більш точним числовим методом (наприклад, збільшити кількість вузлів в методі Гаусса або застосувати інший метод з відомою точністю).

Однак, для методу Гаусса з трьома вузлами (тобто, використовуючи поліном Лежандра 3-го ступеня), похибка залежить від четвертої похідної функції, яка інтегрується. Формула для похибки має вигляд:

$$\frac{(b-a)^5}{2880} f^{(4)}(\xi),$$

де ξ – деяка точка з інтервалу [a,b], $f^{(4)}(x)$ – четверта похідна функції $f(x)$.

Оцінка похибки для обчислення інтегралу методом Гаусса з трьома вузлами становить приблизно 0,00000629.

2.2.5 Порівняння методів чисельного інтегрування.

Аналіз методів наближеного обчислення інтегралів дозволяє зробити висновки, що представлені в таблиці 2.1. Зроблені висновки підтверджуються розглянутими прикладами.

Таблиця 2.1

Порівняння методів чисельного інтегрування

Метод	Точність	Складність	Переваги	Недоліки
Прямокутників	Низька	Низька	Простота реалізації	Низька точність
Трапецій	Середня	Низька	Простота реалізації	Погано працює для функцій зі зміною кривини
Сімпсона	Висока	Середня	Висока точність для гладких функцій	Не працює при непарному n
Гаусса	Дуже висока	Висока	Ефективність	Вимагає попереднього обчислення вузлів

Теоретичний і експериментальний аналіз формул чисельного інтегрування показав, що:

1) для функцій, що мають неперервні похідні досить високого порядку, при однаковій кількості вузлів формули Гаусса дають значно точніший результат, ніж формула Сімпсона, а остання – більш точний результат, ніж формула прямокутників. При цьому для одержання однакової точності за формулою Гаусса необхідно виконати менше операцій, ніж за формулою Сімпсона, а за останньою – менше, ніж за формулами прямокутників або трапецій. Але для формули Гаусса треба зберігати в пам'яті ЕОМ всі значення вузлів t_i і коефіцієнтів A_i;

2) формула Гаусса забезпечує високу точність при невеликій кількості вузлів, тому її особливо вигідно застосовувати при інтегруванні складних функцій, на обчислення значень яких у вузлах витрачається багато часу. Але для функцій малої гладкості, а також для функцій, що мають розриви похідних, всі вказані формули дають приблизно однакову точність;

3) при інтегруванні функцій, що задані таблицею експериментальних значень, використання формули Гаусса майже неможливе. Формула

Сімпсона є достатньо точною, зручно програмується для ЕОМ і тому широко використовується у практичних розрахунках.

РОЗДІЛ 3
ВИМОГИ ДО КАДРОВОГО ЗАБЕЗПЕЧЕННЯ ОБ'ЄКТУ ГАЛУЗІ

Фахівець, який володіє мовою програмування Python та обчислювальними методами, має широкі перспективи розвитку кар'єри завдяки універсальності цих навичок і попиту на ринку. Ось кілька ключових напрямків, які можуть бути цікавими:

Розробка в галузі Data Science та Machine Learning. Python є основною мовою для роботи з даними, аналізу, машинного навчання та штучного інтелекту. Знання обчислювальних методів дозволяє розробляти складні моделі, прогнозувати події та працювати над оптимізацією процесів у різних галузях, таких як фінанси, медицина чи маркетинг.

Інженерія обчислювальної фізики та числового моделювання. У цій галузі Python разом із такими бібліотеками, як NumPy, SciPy, SymPy та Matplotlib, використовується для моделювання природних процесів і складних систем. Це може включати симуляцію руху рідин, розрахунки в аеродинаміці чи матеріалознавстві.

Автоматизація процесів і DevOps. Python широко використовується для створення скриптів автоматизації, управління інфраструктурою, розробки CI/CD-процесів і налаштування серверних систем.

Науково-дослідна діяльність. Поєднання Python із математичними та фізичними обчисленнями дозволяє працювати в наукових установах, лабораторіях, компаніях, які займаються R&D.

Економіка та фінансові технології (FinTech). Python часто використовується для розрахунків фінансових ризиків, моделювання економічних процесів і побудови алгоритмів для автоматичної торгівлі.

Розробка програмного забезпечення. Багато компаній шукають програмістів із глибоким розумінням числових методів для створення прикладного ПЗ, яке вирішує складні математичні чи інженерні задачі.

Викладання та консалтинг. Спеціалісти з Python та обчислювальних методів можуть працювати в освітніх закладах, навчати інших або консультувати компанії з оптимізації їхніх технічних процесів.

Робота у стартапах чи створення власного продукту. Гнучкість Python дозволяє швидко реалізовувати ідеї, від прототипів до готових продуктів, що робить його ідеальним інструментом для інноваційного бізнесу.

Перспективи відкриваються у різних секторах, включаючи IT, енергетику, охорону здоров'я, транспорт і космічну індустрію. Вибір конкретного напрямку залежить від особистих інтересів і подальшого навчання, зокрема у суміжних галузях, таких як машинне навчання, фізика чи бізнес-аналіз.

До фахівців, які володіють Python та обчислювальними методами, висувається низка вимог, залежно від сфери їх застосування. Основні вимоги поділяються на технічні навички, математичну підготовку та "м'які" компетенції.

Технічні навички

1. Глибоке знання Python: включаючи стандартну бібліотеку та популярні бібліотеки, як-от NumPy, SciPy, Pandas, Matplotlib, TensorFlow, PyTorch тощо, залежно від завдань.
2. Робота з алгоритмами та структурами даних: необхідно розуміти основи алгоритмів, пошуку, сортування, графів тощо.
3. Навички числового програмування: уміння реалізовувати обчислювальні методи, як-от чисельне інтегрування, розв'язання диференціальних рівнянь, інтерполяція та апроксимація функцій.
4. Знання баз даних: робота з SQL та NoSQL для обробки великих обсягів даних.
5. Досвід у наукових обчисленнях: використання Python для симуляцій, моделювання або аналізу даних.

6. Знання основ роботи з великими даними та машинним навчанням: базові алгоритми, робота з бібліотеками машинного навчання, як-от scikit-learn, TensorFlow або PyTorch.
7. Досвід роботи з інструментами візуалізації даних: створення графіків, інфографіки, інтерактивних дашбордів.
8. Робота з інструментами контролю версій (Git): вміння працювати в команді над великими проєктами.

Математична підготовка

1. Математичний аналіз: глибоке розуміння числових методів, включаючи інтегрування, диференціювання, розв'язання рівнянь та систем.
2. Лінійна алгебра: знання матриць, векторів, спектрального аналізу, які є ключовими у багатьох методах, особливо в машинному навчанні.
3. Теорія ймовірностей та статистика: знання основ ймовірностей, регресійного аналізу, тестування гіпотез і розподілів даних.
4. Диференціальні рівняння: важливо для роботи у фізичних симуляціях чи фінансовому моделюванні.
5. Оптимізація: розуміння методів мінімізації та максимізації, які широко використовуються в аналізі даних і машинному навчанні.

"М'які" навички

1. Аналітичне мислення: здатність аналізувати складні системи, формулювати задачі та знаходити оптимальні підходи для їх вирішення.
2. Робота в команді: ефективна комунікація з колегами, вміння інтегрувати свої рішення у більші проєкти.
3. Самоорганізація: здатність працювати над великими задачами з мінімальним контролем.
4. Навички презентації та звітності: вміння пояснювати результати роботи, створювати презентації або надавати звіти для замовників чи колег.

5. Готовність до навчання: швидке засвоєння нових технологій, бібліотек, інструментів та алгоритмів.

Додаткові вимоги

- Знання іноземної мови, зазвичай англійської, на рівні достатньому для роботи з документацією, спілкування в міжнародних командах та участі у конференціях.
- Досвід роботи у конкретній галузі (фінанси, енергетика, біотехнології) може бути перевагою.
- Навички роботи з інструментами DevOps, контейнерами (Docker) та хмарними платформами (AWS, Azure, GCP) є додатковим плюсом для спеціалістів, які працюють із великими проєктами чи даними.

Ці вимоги можуть варіюватися залежно від конкретного напряму, але загалом вони формують основу для успішної кар’єри в обраній сфері.

Фахівець повинен мати аналітичне мислення і здатність ефективно працювати з великими масивами даних, інтерпретуючи результати обчислень і будуючи логічні висновки. Високо цінуються комунікативні навички, адже складні технічні рішення часто потрібно пояснювати нетехнічній аудиторії. Також важлива здатність до самостійного навчання, оскільки технології та підходи швидко змінюються, і фахівець має залишатися актуальним у своїй сфері.

РОЗДІЛ 4
МЕТОДИКА ПРОФЕСІЙНОЇ ПІДГОТОВКИ ФАХІВЦІВ У ГАЛУЗІ ЦИФРОВИХ ТЕХНОЛОГІЙ ДО РОЗРОБКИ НАВЧАЛЬНИХ РЕСУРСІВ З ПРОГРАМУВАННЯ ЧИСЕЛЬНИХ МЕТОДІВ РОЗВ'ЯЗАННЯ ЗАДАЧ ДИФЕРЕНЦІАЛЬНОГО ТА ІНТЕГРАЛЬНОГО ЧИСЛЕННЯ МОВОЮ PYTHON
ДИДАКТИЧНИЙ ПРОЄКТ КОНСУЛЬТАТИВНОГО ЗАНЯТТЯ З ТЕМИ «НАБЛИЖЕНЕ ОБЧИСЛЕННЯ ІНТЕГРАЛІВ» ДИСЦИПЛІНИ «МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ СИСТЕМ І ПРОЦЕСІВ» ДЛЯ ЗДОБУВАЧІВ ВИЩОЇ ОСВІТИ СПЕЦІАЛЬНОСТІ 015 ПРОФЕСІЙНА ОСВІТА (ЦИФРОВІ ТЕХНОЛОГІЇ)

4.1 Вихідні дані:

навчальний заклад: Бахмутський навчально-науковий професійно-педагогічний інститут Харківського національного університету імені В. Н. Каразіна;

галузь знань: 01 Освіта / Педагогіка;

спеціальність: 015 Професійна освіта (Цифрові технології);

рівень вищої освіти: перший (бакалаврський);

дисципліна: «Математичне моделювання систем і процесів»;

тема: «Наближене обчислення інтегралів».

Отже, дисципліна містить такі характеристики як:

кількість кредитів – 6;

загальна кількість годин для вивчення дисципліни – для денної форми навчання 180 навчальних годин, з яких: 90 годин самостійної роботи та 90 годин аудиторних занять (38 годин лекційних занять, 24 години практичних занять та 28 годин лабораторних занять).

Форма контролю: іспит.

Значний обсяг навчального матеріалу, складність поставлених навчальних цілей і велика частка часу, відведеного на самостійну роботу, зумовлюють важливість організації консультативних занять. Такі заняття є необхідними для глибшого розуміння та уточнення ключових аспектів дисципліни «Математичне моделювання систем і процесів».

4.2 Проектування цілей консультативного заняття

Формулювання навчальних цілей заняття є ключовим аспектом, від якого залежить побудова його методичної структури, вибір форм організації, підходів, методів викладання, засобів навчання та механізмів контролю. Навчальні цілі виступають системоутворюючим компонентом, адже вони визначають кінцеві результати підготовки майбутніх фахівців, слугуючи основою для формування всієї системи навчання.

Зважаючи на це, методична підготовка спеціалістів передбачає опанування вмінням чітко та диференційовано визначати навчальні цілі, орієнтуючись на різні рівні професійної компетенції та глибину засвоєння матеріалу.

Проектування цілей консультативного заняття представлені у табл. 4.1.

Таблиця 4.1

Цілі консультативного заняття

Цілі консультативного заняття	Цілі формування різних рівнів засвоєння навчального матеріалу	Умови досягнення	Результат у вигляді дій здобувачів вищої освіти
1	2	3	4
1	З переліку визначень впізнавати основні поняття теми «Наближене обчислення інтегралів» такі, як інтегрування, коефіцієнти, квадратурна формула прямокутників.	Знати визначення понять: Інтеграл, Пряма, Вектор.	Правильно названо з переліку основні поняття теми «Наближене обчислення інтегралів» такі, як інтегрування, коефіцієнти, квадратурна формула прямокутників.
2	Уміти самостійно підбирати, використовувати та створювати порівняльну характеристику формул	Виконання дій першого рівня.	Правильно самостійно підібрано та використано растрова порівняльну характеристику формул прямокутників, трапецій,

Продовження таблиці 4.1

	прямокутників, трапецій, Сімпсона і Гаусса чисельного інтегрування, уміти використовувати обчислення інтегралів.		Сімпсона і Гаусса чисельного інтегрування, уміти використовувати обчислення інтегралів
3	Уміти наводити методику наближеного обчислення подвійних інтегралів за формулами середніх прямокутників, Сімпсона і Гаусса.	Виконання дій першого і другого рівнів.	Правильно наводено методику наближеного обчислення подвійних інтегралів за формулами середніх прямокутників, Сімпсона і Гаусса.
4	Уміти правильно використовувати чисельні методи обчислення визначених інтегралів в педагогічній діяльності.	Виконання дій першого, другого і третього рівнів.	Правильно використані чисельні методи обчислення визначених інтегралів в педагогічній діяльності.

У результаті підготовки було сформульовано цілі консультативного заняття з теми «Наближене обчислення інтегралів», що входить до дисципліни «Математичне моделювання систем і процесів». Це заняття розроблено спеціально для здобувачів вищої освіти за спеціальністю 015 Професійна освіта (Цифрові технології).

4.4 Визначення найбільш складних для розуміння та засвоєння питань.

На цьому етапі необхідно виокремити питання, які можуть становити найбільшу складність для студентів у процесі розуміння та опанування. Це дозволить акцентувати увагу на тих аспектах теми, які потребують більш глибокого аналізу, додаткових пояснень або використання наочних матеріалів. Визначення таких складних моментів також допоможе викладачу адаптувати методику викладання, зробивши її більш ефективною для досягнення поставлених навчальних цілей (табл. 4.2).

Таблиця 4.2

Обрання питань для консультування та формулювання відповідей на можливі питання

Теми (або тема) дисципліни	Зміст програми за кожною темою	Найбільш складні питання за темами (темою)	Відповіді на питання
1	2	3	4
«Наближене обчислення інтегралів»	1.Чисельне інтегрування функцій однієї змінної. 2. Чисельне інтегрування функцій двох змінних.	1. В зв'язку з чим виникає необхідність використання чисельних методів обчислення визначених інтегралів?	1. Необхідність використання чисельних методів обчислення визначених інтегралів виникає у зв'язку з низкою факторів: складність аналітичного розв'язання, практичні задачі, висока розмірність задачі, робота з великими обсягами даних, похибки моделювання. Таким чином, чисельні методи є потужним інструментом для розв'язання інтегралів, особливо у випадках, коли аналітичне обчислення є недоцільним або неможливим.
		2. В чому полягає суть методів прямокутників, трапецій і Сімпсона наближеного обчислення визначених інтегралів?	2. Методи прямокутників, трапецій і Сімпсона є чисельними способами наближеного обчислення визначених інтегралів. Їх суть полягає у заміні криволінійної області під графіком функції геометричними фігурами (прямокутниками, трапеціями або параболами) і обчисленні площі цих фігур як наближеного значення інтегралу.
		3. Наведіть квадратурні формули середніх прямокутників, трапецій, Сімпсона для функцій однієї змінної.	3. Квадратурна формула прямокутників випливає з геометричного змісту визначеного інтеграла, як площі криволінійної трапеції. Для знаходження площі цієї трапеції розіб'ємо її основу (відрізок інтегрування $[a,b]$) точками $a = x_0 < x_1 < ... < x_n = b$ на n рівних частин довжиною $h = \frac{b-a}{n}$. Через ці точки проведемо прямі $x = x_i,\ i = \overline{0,n}$, паралельно осі OY. Як результат, криволінійна трапеція розіб'ється на n смужок шириною h. Кожну смужку наближено замінемо прямокутниками висотою $f(\xi_i)$ де ξ_i – довільна точка на відрізку $[x_{i-1}, x_i],\ i = \overline{1,n}$ Площі

Продовження таблиці 4.2

			прямокутників рівні $S_i = f(\xi_i)h,\ i=\overline{1,n}$ Підсумовуючи площі всіх прямокутників, одержимо наближене значення інтеграла: $\int_a^b f(x)dx \approx \frac{b-a}{n}(f(\xi_1)+f(\xi_2)+...+f(\xi_n))$
		4. Наведіть оцінки залишкових членів квадратурних формул середніх прямокутників, трапецій, Сімпсона.	4. Оцінки залишкових членів квадратурних формул показують похибку чисельного методу наближеного обчислення визначених інтегралів. Метод середніх прямокутників. Залишковий член (похибка) методу середніх прямокутників оцінюється за формулою: R=−(b−a)324n2f′′(ξ),R = -\frac{(b - a)^3}{24 n^2} f''(\xi),R=−24n2(b−a)3f′′(ξ), де: f′′(ξ)f''(\xi)f′′(ξ) — друга похідна функції f(x)f(x)f(x) у деякій точці ξ∈[a,b]\xi \in [a, b]ξ∈[a,b], nnn — кількість підінтервалів. Похибка залежить від другої похідної Метод Сімпсона. Залишковий член для методу Сімпсона оцінюється за формулою: R=−(b−a)5180n4f(4)(ξ),R = -\frac{(b - a)^5}{180 n^4} f^{(4)}(\xi),R=−180n4(b−a)5f(4)(ξ), де: f(4)(ξ)f^{(4)}(\xi)f(4)(ξ) — четверта похідна функції f(x)f(x)f(x) у деякій точці ξ∈[a,b]\xi \in [a, b]ξ∈[a,b], nnn — кількість підінтервалів (має бути парним). Похибка методу Сімпсона зменшується набагато швидше, ніж у методів прямокутників і трапецій, оскільки вона обернено пропорційна четвертому степеню кількості підінтервалів nnn.

Таким чином, ми виокремили ключові аспекти теми «Наближене обчислення інтегралів», які виявляються найбільш складними для розуміння та засвоєння. Ця тема є частиною дисципліни «Математичне моделювання систем і процесів» і спрямована на підготовку здобувачів вищої освіти спеціальності 015 Професійна освіта (Цифрові технології). Особливу увагу було приділено виявленню проблемних моментів, які потребують більш глибокого пояснення та додаткового опрацювання для ефективного засвоєння матеріалу.

4.5 Вибір дидактичних методів активізації

На наступному етапі, оберемо методи активізації навчальної діяльності здобувачів освіти на консультації (табл. 4.3).

Таблиця 4.3

Методи активізації навчальної діяльності здобувачів вищої освіти на консультації

Дидактичні методи	Реалізація методів при проведенні консультаційного заняття
1	2
Методи підвищення наочності	Застосування інтерактивної дошки та мультимедійного проектора для демонстрації презентаційних матеріалів на тему «Наближене обчислення інтегралів», а також використання наочних плакатів, що ілюструють основні поняття та методи в межах дисципліни «Математичне моделювання систем і процесів».
Мотиваційні методи	Для реалізації мотивації використаємо: тип: внутрішня мотивація; вид: вступна мотивація; метод: мотивуючий вступ; прийом: віднесення до особистості. Вивчення теми «Наближене обчислення інтегралів» є надзвичайно важливим для вашого професійного розвитку, оскільки ця тема має практичне значення в контексті вашої майбутньої діяльності. Знання з цієї дисципліни дозволить вам ефективно використовувати відповідні математичні методи та технології у сфері освіти, зокрема при розробці та застосуванні програмного забезпечення для обробки даних. Це є важливою частиною вашої роботи в освітньому процесі, а також у вашій професійній діяльності на підприємстві. Ваша здатність успішно виконувати ці завдання безпосередньо впливає на результативність роботи організації, а також допоможе вам стати кваліфікованим фахівцем у галузі програмування та цифрових технологій.
Комунікативні методи	Основною характеристикою комунікативного методу є його комунікативна складова, яка охоплює кілька важливих аспектів: 1. Вступ, що зацікавлює — це ключовий елемент виступу, який має залучити увагу аудиторії та налаштувати її на подальше сприйняття інформації. 2. Чітка дикція — важлива для того, щоб слухачі легко й швидко розуміли подану інформацію. 3. Уміння контролювати власні емоції — дозволяє слухачам зосередитися на темі, а не на особистості доповідача. 4. Зоровий контакт — свідчить про впевненість доповідача та його зацікавленість у слухачах. 5. Сила голосу — варіюється залежно від кількості слухачів, зовнішніх шумів, змісту та мети виступу. 6. Розкриття теми — надає доповіді логічної цілісності та допомагає слухачам краще запам'ятовувати основні ідеї. 7. Переконлива мова — підвищує увагу аудиторії до матеріалу та створює відчуття важливості сказаного.

Ми вибрали методи, які сприяють активізації навчального процесу під час консультацій. Це дозволить залучити студентів до глибшого осмислення навчального матеріалу, мотивуючи їх до самостійної діяльності та розвитку аналітичного мислення. Застосування інтерактивних підходів, таких як групові обговорення, вирішення практичних завдань і мозкові штурми, підвищить ефективність навчання, активуючи процес засвоєння знань і формуючи в студентів здатність самостійно вирішувати професійні задачі.

4.6 Вибір способів організації консультативного заняття

Необхідно обрати відповідні способи організації консультативного заняття, враховуючи інформацію, наведене в таблиці 4.4. Цей вибір має базуватися на аналізі потреб студентів, складності навчального матеріалу та цілей заняття, щоб забезпечити найбільш ефективну взаємодію між викладачем і слухачами.

Таблиця 4.4

Варіанти організації консультативного заняття

№ варіанта	Етапи організації заняття	Характеристика варіанта
1	2	3
1	- вступне слово лектора, - відповіді на питання здобувачів вищої освіти і обговорення їх, - заключне слово викладача.	Недоліком цього варіанту проведення лекції-консультації є відсутність послідовності, системи в питаннях, на які доводиться викладачу давати відповіді. Питання поступають хаотично, що знижує якість консультації.
2	- збір питань в письмовій формі до лекції, їх систематизація, - відповіді на питання, що поступили, - відповіді на додаткові питання, - обмін думками, - висновки.	Цей варіант, на відміну від попереднього, дозволяє викладачу групувати відповіді, що сприяє кращому засвоєнню навчального матеріалу здобувачами вищої освіти.
3	- видача завдань на самостійне вивчення матеріалу теми. - підготовка питань лектору. - відповіді і їх обговорення	В цьому випадку консультування грає функцію додаткового інформування зі складних питань і пояснення незрозумілого навчального матеріалу.
4	- повідомлення теми, - консультування декількома фахівцями в певній області науки	Цей варіант лекції-консультації проводиться, як правило, зі спеціальних дисциплін, іноді для цієї мети

Продовження таблиці 4.4

	і техніка з актуальних питань науки і нової техніки.	використовуються наукові семінари. Такі заняття дають можливість зіставити думки різних учених на одну і ту ж проблему і є чудовою школою ведення дискусії.

Згідно представленої таблиці обираємо 1 варіант організації консультативного заняття, на якому викладач пояснює питання, які здалися незрозумілими здобувачам вищої освіти.

4.7 Розробка сценарію проведення консультативного заняття

Представляємо розробку сценарію проведення консультативного заняття, що ґрунтується на вибраному варіанті його організації, як це зазначено в таблиці 4.5. У цьому сценарії враховано всі аспекти, що сприятимуть ефективному засвоєнню матеріалу та залученню студентів до активної навчальної діяльності.

Таблиця 4.5

Сценарій консультативного заняття

Етапи проведення консультативного заняття	Дії викладача	Дії здобувачів вищої освіти
1	2	3
Організаційний момент	Привітання викладача, перевірка присутності студентів під час переклички, а також оцінка зовнішніх умов у аудиторії, що створюють комфортну атмосферу для навчання.	Вітання викладача. Створення позитивного настрою для студентів, налаштовуючи їх на активну участь у навчальному процесі
Повідомлення теми і мети уроку	Оголошення теми заняття «Наближене обчислення інтегралів», визначення основних цілей навчання. У цьому контексті фіксується тема, пояснюються її значення та практична важливість.	Студентам представлено очікувані результати засвоєння матеріалу, які включають розвиток навичок і вмінь, необхідних для успішного виконання завдань з наближеного обчислення інтегралів.
Мотивація мети	Повідомлення важливості вивчення даної теми: «Наближене обчислення інтегралів» для вас, як майбутніх фахівців, є достатньо актуальним. Засвоєння цієї	Освідомлення значущості та актуальності вивчення цієї теми, а також зацікавленість до неї з боку студентів. Це допомагає студентам

	дисципліни дозволить вам ефективно застосовувати математичні методи та технології у сфері освіти, а також під час розробки та використання програмного забезпечення для обробки даних. Це стане важливою частиною вашої діяльності як в освітньому процесі, так і в професійній роботі на підприємствах. Ваші вміння та здатність успішно виконувати ці завдання безпосередньо впливають на ефективність роботи організації і сприяють вашому розвитку як висококваліфікованого фахівця в області програмування та цифрових технологій.	зрозуміти, чому тема є важливою для їхнього майбутнього професійного розвитку, а також мотивує їх активно брати участь у навчальному процесі та вивченні матеріалу.
Актуалізація знань	Викладач проводить фронтальне усне опитування з метою перевірки базових знань: 1. В чому полягає суть методу Гаусса наближеного обчислення визначених інтегралів? 2. Дайте порівняльну характеристику формул прямокутників, трапецій, Сімпсона і Гаусса чисельного інтегрування. 3. Наведіть методику наближеного обчислення подвійних інтегралів за формулами середніх прямокутників, Сімпсона і Гаусса.	Здобувачі вищої освіти беруть участь у анкетуванні, надаючи відповіді на поставлені запитання. Очікувані відповіді: 1. Метод Гаусса для наближеного обчислення визначених інтегралів є чисельним методом інтеграції, який базується на використанні спеціальних точок і ваг для досягнення високої точності при відносно малому числі точок. 2.Метод Гаусса забезпечує найвищу точність серед усіх методів при мінімальній кількості точок, але вимагає спеціальних знань і обчислень. Метод Симпсона дає високу точність для гладких функцій і є простішим у використанні, ніж метод Гаусса. Методи прямокутників і трапецій мають нижчу точність, але простіші для реалізації і можуть бути достатніми для простих функцій або великих відрізків.

Продовження таблиці 4.5

		3. У разі, коли потрібно досягти високої точності для подвійних інтегралів, методи Симпсона і Гаусса є найбільш ефективними. Проте метод середніх прямокутників може бути корисним у випадку, коли простота реалізації є пріоритетом, а точність не є критично важливою.
Формування ООД	Викладач проводить консультацію згідно плану, за допомогою методу - пояснення: План 1.Чисельне інтегрування функцій однієї змінної. 2. Чисельне інтегрування функцій двох змінних.	Слухають виклад матеріалу, записують ключові моменти та роблять конспекти для подальшого вивчення.
Визначення проблемних моментів під час вивчення питань теми та формування ВД	Викладач запитує здобувачів вищої освіти про недоречності, які виникли у них під час самостійного вивчення теми. Викладач відповідає на поставлені запитання: 1. Обчислення інтегралів є важливим інструментом у багатьох галузях математики, фізики, інженерії та інших науках. Однак, як і будь-який метод, воно має свої переваги та недоліки. Переваги: Універсальність застосування, Математичні моделі, Інструмент для чисельних методів, Застосування в реальних умовах. Недоліки: Складність аналітичних методів, Високі вимоги до обчислювальних ресурсів, Помилки числових методів, Необхідність в глибоких знаннях методів. Отже, хоча обчислення інтегралів є важливим і необхідним інструментом для розв'язання різноманітних задач, воно може бути складним і вимагати значних ресурсів, особливо в контексті числових методів.	Здобувачі вищої освіти запитують: 1. Які переваги і недоліки надає обчислення інтегралів?

Продовження таблиці 4.5

Підведення підсумків	Викладач підводить підсумки проведення консультації: «Сьогодні ми розглянули незрозумілі вам питання теми для самостійного вивчення. Зараз перевіримо як ви засвоїли незрозумілий вам матеріал. Скажіть, як ви розумієте що ж таке «Наближене обчислення інтегралів»?	Здобувачі вищої освіти слухають, відповідають: «Наближене обчислення інтегралів — це методи та техніки, що використовуються для знаходження значень інтегралів, коли їх точне обчислення є складним або неможливим через відсутність аналітичного розв'язку. Наближені методи застосовуються для чисельного обчислення визначених і невизначених інтегралів, коли функції, які потрібно інтегрувати, є складними або не мають елементарних первісних». Здобувачі вищої освіти прощаються.

Отже, на цьому етапі ми підготували детальний сценарій для проведення консультативного заняття, враховуючи обраний формат його організації.

На завершальному етапі розроблено контурний конспект з теми «Наближене обчислення інтегралів» дисципліни «Математичне моделювання систем і процесів» для здобувачів вищої освіти за спеціальністю 015 Професійна освіта (Цифрові технології). Детальний контурний конспект заняття з цієї теми представлено у Додатку А.

ВИСНОВКИ

У ході дослідження здійснено уточнення і систематизацію основних понять, що розкривають сутність та структуру професійної підготовки фахівців у галузі цифрових технологій для створення навчальних ресурсів, що використовують чисельні методи для розв’язання задач диференціального та інтегрального розрахунку мовою Python. Також було визначено основні напрямки для вдосконалення їх професійних компетентностей.

З’ясовано, що професійна підготовка фахівців у галузі цифрових технологій, яка спрямована на розробку навчальних ресурсів для програмування чисельних методів розв’язання математичних задач, є місцем і нагальним питанням у контексті сучасної педагогіки.

Проаналізовано важливість і актуальність підготовки фахівців для розробки навчальних ресурсів, які охоплюють чисельні методи для розв'язання завдань диференціального та інтегрального підрахунку.

Наведено детальний опис структури та організації системи професійної підготовки фахівців у галузі цифрових технологій для створення навчальних ресурсів, що орієнтовані на програмування чисельних методів розв'язання задач диференціального та інтегрального числення за допомогою мови Python.

Проаналізовано особливості програмування числових методів мовою Python. Наведено переваги та недоліки використання мови Python для програмування методів обчислювальної математики

Розроблено блок-схеми алгоритмів та програми мовою Python числових методів диференціювання та інтегрування.

Наведено вимоги до кадрового забезпечення об'єкту галузі.

Розроблено дидактичний проект консультативного заняття з теми «Наближене обчислення інтегралів» дисципліни «Математичне моделювання систем і процесів».

За основними результатами дослідження виконана публікація тез доповіді на VIII Міжнародній науково-практичній конференції здобувачів вищої освіти та молодих учених «Студенти та молодь – для майбутнього країни» (м. Харків, 14-15 листопада 2024 р.). Текст тез доповіді наведено у додатку Б.