

Жиляк О.Р., студент 2 курсу спеціальності 122 «Комп'ютерні науки»

Науковий керівник:

Потапова Н. А., к.е.н., доцент, доцент кафедри інформаційних технологій

ВИКОРИСТАННЯ МЕТОДІВ ОБЧИСЛЕННЯ ДЛЯ РОЗВ'ЯЗКУ СИСТЕМ НЕЛІНІЙНИХ РІВНЯНЬ

Донецький національний університет імені Василя Стуса, м. Вінниця

Використання методів обчислення для розв'язку систем нелінійних рівнянь є важливим аспектом в багатьох наукових і прикладних дослідженнях. Системи нелінійних рівнянь зустрічаються в різних галузях, включаючи фізику, інженерію, економіку та багато інших.

Методи обчислення для розв'язку систем нелінійних рівнянь використовуються для знаходження числових розв'язків цих систем, коли аналітичні методи стають непрактичними або неможливими. Одним з найпоширеніших методів є метод Ньютона, який базується на ітеративному процесі апроксимації розв'язку шляхом послідовного покращення.

Процес розв'язання системи нелінійних рівнянь зазвичай складається з наступних етапів:

Формулювання задачі: Визначення системи нелінійних рівнянь, яку потрібно розв'язати, та постановка початкових умов або обмежень.

Вибір методу: Вибір підходящого методу обчислення для розв'язку системи нелінійних рівнянь залежно від її властивостей та поставленої задачі. Метод Ньютона є одним з загальнозживаних методів, але також існують інші методи, такі як методи ітерацій, методи послідовних наближень тощо.

Початкова апроксимація: Вибір початкового наближення для розв'язку системи нелінійних рівнянь, що може вплинути на швидкість збіжності методу.

Ітераційний процес: Проведення ітераційного процесу для покращення апроксимації розв'язку. Кожна ітерація включає оновлення значень змінних системи згідно з обраною методикою.

Збіжність: Перевірка умов збіжності методу та оцінка точності отриманого розв'язку. Якщо метод не збігається або збігається повільно, можуть застосовуватися додаткові покращені методи, такі як методи релаксації, методи розбиття на підінтервали, методи наближення тощо.

Аналіз результатів: Оцінка і аналіз отриманого розв'язку системи нелінійних рівнянь з точки зору відповідності поставленій задачі та властивостям системи.

Використання методів обчислення для розв'язку систем нелінійних рівнянь вимагає вміння програмування та розуміння математичних основ методів. Деякі інструменти та мови програмування, такі як MATLAB, Python з бібліотеками NumPy та SciPy, мають вбудовані функції для розв'язку систем нелінійних рівнянь, що полегшує їх застосування у практичних завданнях.

Список літератури.

1. Використання методів hneu 2020р [Електроний ресурс] URL: https://pns.hneu.edu.ua/pluginfile.php/293300/mod_resource/content/1/Тема%204.pdf.
2. Нелінійні розрахунки nubip 2021р [Електроний ресурс] URL: <https://elearn.nubip.edu.ua/mod/book/view.php?id=355969&chapterid=123767>

УДК 004.652.4-021.272(043.2)

*Журовський Я.О., здобувач 2 курсу спеціальності 122 «Комп'ютерні науки»,
Зелінська О.В., к.т.н., доцент
кафедри інформаційних технологій*

ПОРІВНЯЛЬНИЙ АНАЛІЗ АРХІТЕКТУР НЕРЕЛЯЦІЙНИХ БАЗ ДАНИХ

Донецький національний університет імені Василя Стуса, м. Вінниця

Нереляційні бази даних дедалі більше використовуються у сучасних ІТ-проектах. Вони дозволяють ефективно зберігати та обробляти великі обсяги даних. Але на відміну від реляційних баз даних, вони не використовують стандартні таблиці зі зв'язками між ними. Замість цього, нереляційні бази даних характеризуються різними типами архітектур.

Основні типи архітектур нереляційних баз даних можна розділити на декілька категорій: документ-орієнтовані, ключ-значення, колоночні та графові.

Документ-орієнтовані бази даних використовують документи, наприклад, JSON або BSON, які містять різні дані відносно одного об'єкту. Найбільш відомими прикладами документ-орієнтованих баз даних є MongoDB, Couchbase та CouchDB.

У ключ-значення базах даних, дані зберігаються у вигляді пар "ключ-значення". Це дозволяє дуже швидко знаходити дані за ключем. Redis, Riak і Aerospike є прикладами ключ-значення баз даних.

Колоночні бази даних зберігають дані у вигляді колонок, а не рядків. Це дозволяє ефективно зберігати та обробляти великі обсяги даних, зокрема, аналізувати та здійснювати запити до великих даних. Cassandra, HBase та Amazon DynamoDB - це декілька з прикладів колоночних баз даних.