

С. В. ПОГРЕБНЯК, О. О. ВОДКА

ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ ВИЗНАЧЕННЯ НАПРУЖЕНО-ДЕФОРМОВАНОГО СТАНУ КОНСТРУКЦІЇ ІЗ ЕЛАСТОМЕРІВ НА ОСНОВІ НЕЙРОМЕРЕЖЕВОЇ МОДЕЛІ

У роботі представлено інформаційну систему для визначення механічних характеристик еластомерних матеріалів на основі інтеграції штучних нейронних мереж із класичними методами чисельного аналізу. Запропоновано гібридний підхід, у межах якого нейромережевий модуль виконує апроксимацію експериментальної кривої навантаження-розвантаження, включно з ефектом Маллінса. Отриманий прогноз напружено-деформованої відповіді використовується у поєднанні з ітераційним алгоритмом Ньютона-Рафсона для обчислення дотичного модуля пружності за умови відсутності аналітичних похідних. Для забезпечення чисельної стійкості реалізовано механізм контролю ітераційного кроку, обмеження корекції деформації та систему локального відновлення значень у разі розбіжності, що унеможливає поширення помилок на весь розрахунок. Архітектура застосунку побудована як кросплатформне настільне середовище на основі PySide6, що поєднує інструмент інтерактивного редагування структури скінченно-елементної моделі, динамічну валідацію даних та підтримку топологічної цілісності завдяки використанню патерну делегування. Передбачені серіалізація й десеріалізація у форматі CSV, включно з перевіркою коректності полів, автоматичним приведенням типів і відновленням порожніх комірок, що забезпечує відтворюваність результатів та обмін моделями між користувачами без втрати інформації. Результат роботи представлений у вигляді графічної візуалізації деформованого стану конструкції та детального аналітичного звіту, який містить переміщення і реакції вузлів, внутрішні зусилля, деформаційні характеристики елементів і отримані ефективні значення модулю пружності. Запропонована система забезпечує високу стабільність і демонструє практичну придатність під час моделювання еластомерів із вираженою нелінійною поведінкою. Отримані результати підтверджують перспективність поєднання машинного навчання та класичних методів механіки для подальшого розвитку сучасних інструментів інженерного аналізу.

Ключові слова: інформаційна система, штучні нейронні мережі, тангенціальний модуль, еластомерні матеріали, інженерний аналіз.

S. POHREBNIAK, O. VODKA

INFORMATION SYSTEM FOR DETERMINING THE STRESS-STRAIN STATE OF ELASTOMERS STRUCTURES BASED ON A NEURAL-NETWORK

This work presents an information system designed to determine the mechanical characteristics of elastomeric materials through the integration of artificial neural networks with classical numerical analysis methods. A hybrid framework is proposed in which a neural network module approximate the experimental loading-unloading curve, including the Mullins effect as a key manifestation of material hysteresis. The predicted stress-strain response is incorporated into an iterative Newton-Raphson scheme to compute the tangent modulus of elasticity in the absence of analytical derivatives. Numerical stability is ensured by implementing an adaptive step-control mechanism, constraints on deformation corrections, and a local recovery procedure that prevents the propagation of divergence throughout the global solution. The system architecture is implemented as a cross-platform desktop application based on PySide6, combining interactive editing of the finite-element model structure, dynamic data validation, and maintenance of topological consistency through the delegation pattern. The model supports serialization and deserialization in CSV format, including field validation, automatic type conversion, and restoration of empty entries, thereby enabling reproducible calculations and reliable model exchange between users. The outputs of the system include both a graphical representation of the deformed structural configuration and a detailed analytical report containing nodal displacements and reactions, internal forces, strain measures, and resulting effective elastic moduli. The proposed system provides high computational stability and demonstrates practical applicability for modelling elastomers with pronounced nonlinear behaviour. The results highlight the potential of combining machine learning with classical mechanical techniques to advance modern tools for the engineering analysis.

Keywords: information system, artificial neural networks, tangential modulus, elastomeric materials, engineering analysis.

Вступ. Зі зростанням складності інженерних систем та обсягів обчислювальних даних збільшується потреба у високоефективних інформаційних системах, що інтегрують моделювання, розрахунки та візуалізацію. Сучасні інженерні дослідження вимагають швидкого та точного оцінювання механічних характеристик матеріалів і конструкцій, зокрема у випадках нелінійних ефектів, таких як ефект Маллінса при навантаженні-розвантаженні гумових матеріалів.

У цьому контексті особливо актуальним є створення користувацьких інтерфейсів, що забезпечують ефективну взаємодію дослідника або інженера з обчислювальним ядром. Вибір

архітектурного патерну системи визначає ефективність взаємодії. Хмарні архітектури забезпечують масштабованість та віддалене виконання складних розрахунків, проте для задач з інтерактивною візуалізацією та багаторазовими ітераціями математичного ядра затримки через передачу даних мережею можуть бути критичними. Монолітні настільні застосунки, натомість, дозволяють працювати безпосередньо з апаратними ресурсами робочої станції, забезпечуючи швидкий відгук системи, цілісність даних у межах одного процесу та ефективну обробку графічної інформації.

Використання штучних нейронних мереж (ШНМ) для моделювання механічної поведінки матеріалів є

© С. В. Погребняк, О. О. Водка 2025



Дослідницька стаття: Цю статтю опубліковано видавництвом НТУ «ХП» у збірнику «Вісник Національного технічного університету «ХП» Серія: Динаміка та міцність машин». Ця стаття поширюється за міжнародною ліцензією [Creative Commons Attribution \(CC BY 4.0\)](https://creativecommons.org/licenses/by/4.0/). **Конфлікт інтересів:** Автор/и заявив/или про відсутність конфлікту.



однією з найбільш стрімко розвиваних галузей сучасної комп'ютерної механіки. Значна кількість досліджень демонструє потенціал ШНМ у відтворенні складної нелінійної поведінки та ефектів, залежних від історії деформації [1-3]. Однак існуючі підходи часто не враховують вимоги до інтеграції локальної конститутивної моделі у зовнішній інтеграційний метод розв'язання, зокрема метод Ньютона-Рафсона, що є критично важливим для побудови повноцінної інформаційної системи прогнозування механічної відповіді матеріалів.

Значна частина робіт фокусується на фізично-інформованих нейронних мережах (Physics-Informed Neural Networks або PINN), які застосовуються як сурогатні розв'язувачі диференціальних рівнянь у частинах похідних (ДРЧП) і замінюють собою весь чисельний метод [4-6]. Наприклад, у [4-5] PINN апроксимують повні поля переміщень та напружень, тоді як SciANN [6] пропонує програмну інфраструктуру для розв'язання широкого класу ДРЧП. Хоча такі методи довели свою ефективність, вони не надають локальної конститутивної моделі, яку можна було б інтегрувати у зовнішній ітераційний алгоритм. Крім того, вони можуть мати обмеження щодо точності прогнозування вторинних полів, зокрема напружень [4]. Інші сурогатні підходи, такі як "U-Mesh" [7], спрямовані на апроксимацію прямого відображення між прикладеними навантаженнями та полями переміщень, тобто не формують явної матеріальної моделі на рівні точки інтегрування.

Окремі дослідження пропонують спеціалізовані ШНМ-моделі для конкретних класів матеріалів. Наприклад, у [8] представлено нейронну конститутивну модель для кристалічних структур, що використовує окремі мережі для прогнозування напружень та дотичної жорсткості, але цей підхід орієнтований на анізотропічні кристалічні матеріали та непридатний для ізотропічної гіпереластичності. Моделі зниженого порядку (ROM), такі як у [9], застосовують ШНМ для оптимізації проєкційних операторів у задачах гіпереластичності, що також відрізняється від задачі створення локальної матеріальної моделі. Підхід використаний в публікації [10], використовує ШНМ для заміни процедури return-mapping, але він передбачає наявність явних внутрішніх змінних, тоді як у задачі моделювання гуми таких змінних немає.

Інша група методів спрямована на безпосереднє прогнозування дотичної жорсткості чи історії навантаження. У [11] запропоновано симетрично-позитивно визначену нейронну мережу (SPD-NN), яка передбачає дотичну матрицю через розклад Холєцького, що забезпечує чисельну стабільність. Проте цей підхід передбачає прямий вивід модуля, тоді як у даній роботі він має обчислюватися ітеративно з кривої навантаження-розвантаження. Зокрема, у [12] прогноуються похідні енергії деформації для біополімерних гелів, а в [13] застосовано чисельне диференціювання на основі комплексних змінних (CVDM) для отримання модулів з виходу ШНМ.

Підходи на основі рекурентних мереж (RNN) [1, 3] моделюють матеріали з пам'яттю або процес руйнування, оновлюючи внутрішні змінні стану, проте вони не надають явного методу для обчислення дотичного модуля, необхідного для подальших використання в розрахунках на міцність.

Таким чином, аналіз літератури свідчить про відсутність універсального підходу, який би поєднував:

- I. Здатність нейронної мережі відтворювати повну нелінійну криву навантаження-розвантаження з ефектом Маллінса;
- II. Можливість інтеграції цієї кривої у зовнішній чисельний алгоритм для ітеративного обчислення дотичного модуля пружності;
- III. Придатність та зручність до використання в інформаційних системах для інженерних застосувань.

У даній роботі пропонується підхід, спрямований на подолання зазначених недоліків, шляхом розробки інформаційної системи, яка поєднує нейромережевий модуль прогнозування кривої механічної відповіді гуми та ітеративний алгоритм Ньютона-Рафсона для обчислення дотичного модуля пружності. Для реалізації інтерактивного інтерфейсу цієї системи, що забезпечує введення структури конструкції, попередній перегляд 2D-моделі та розрахунок напружено-деформованого стану за допомогою бібліотеки Mechanics [14] для Python, доцільним є використання настільної монолітної архітектури, яка оптимізує продуктивність та зручність користувача.

1. Огляд сучасних підходів до побудови інформаційних систем та існуючих математичних моделей. Сучасні інженерні інформаційні системи формуються як комплексні інтелектуальні середовища, призначені для автоматизованого моделювання, аналізу та підтримки прийняття рішень у задачах механіки деформованого твердого тіла. Розвиток таких систем нерозривно пов'язаний із впровадженням адаптивних алгоритмів, нейромережевих моделей, а також числових методів, здатних забезпечити точне визначення параметрів матеріалу за умови нелінійності та гістерезису. У сучасних прикладних інформаційних системах для інженерного аналізу дедалі ширше інтегруються модулі числових розрахунків та алгоритми машинного навчання, що дає змогу підвищити точність апроксимації експериментальних деформаційних залежностей і автоматизувати процес оцінювання параметрів матеріалу.

Традиційні підходи до числового опису поведінки гумових і полімерних матеріалів базуються на розв'язанні рівняння нев'язки:

$$F_{ext}(x) - F_{int}(x) = f(x) = 0, \quad (1)$$

де, $F_{ext}(x)$ – вектор зовнішніх сил прикладених до системи, $F_{int}(x)$ – вектор внутрішніх сил, що виникають у матеріалі конструкції внаслідок деформації, $f(x)$ – нев'язка, визначає різницю між зовнішнім навантаженням і внутрішньою відповіддю

моделі матеріалу або між експериментальними та модельними деформаційними даними (1). У таких постановках одним з основних інструментів залишається метод Ньютона-Рафсона [13], який у скалярному випадку задається формулою:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}, \quad (2)$$

що демонструє квадратичну збіжність за умови коректного визначення похідної f' . Значна частина числових розрахунків у механіці ґрунтується на цій ітерації (2), оскільки вона забезпечує високу точність у задачах з помірною нелінійністю.

Альтернативні підходи, відомі як квазиньютонівські або січні методи, дозволяють уникнути прямого обчислення похідних. Найпоширенішим прикладом є січне наближення (3):

$$f'(x_i) \approx \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}, \quad (3)$$

яке приводить до ітерації:

$$x_{i+1} = x_i - f(x_i) \frac{x_i - x_{i-1}}{f(x_i) - f(x_{i-1})}, \quad (4)$$

Ітераційна схема (3) має суперлінійну збіжність і забезпечує обчислювальну ефективність у випадках, коли функція $f(x)$ формується емпірично або через складні числові моделі, зокрема за допомогою нейронних мереж.

У контексті одновісне деформування конструктивних елементів, зокрема балок, використовується класичне співвідношення між відносною деформацією та подовженням елемента. Для балки довжиною l (рис. 1), при подовженні Δl відносна деформація визначається як:

$$u = \frac{\Delta l}{l}, \quad (5)$$

де u є градієнтом переміщення уздовж осі елемента. Внутрішня сила, що виникає у перерізі балки, подається через площу перерізу A і напруження σ :

$$f = \sigma A, \quad (6)$$

а саме напруження пов'язане з деформацією через матеріальну модель:

$$\sigma = \varepsilon E(\varepsilon), \quad (7)$$

де $E(\varepsilon)$ – є тангенціальним модулем пружності. У межах запропонованого підходу $E(\varepsilon)$ визначається за допомогою алгоритму Ньютона-Рафсона та нейромережевого алгоритму, що апроксимує нелінійну матеріальну відповідь на основі експериментальних даних, а заданий аналітично модуль пружності використовується лише у разі помилки або відсутності збіжності алгоритму.

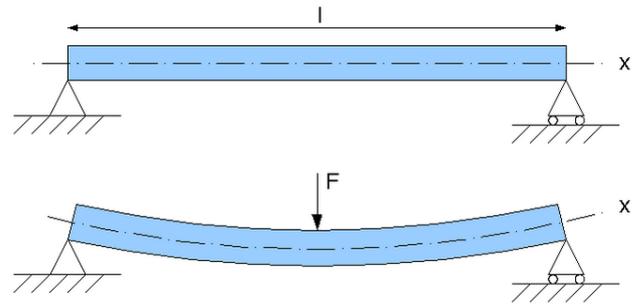


Рис. 1 – Приклад одновісного деформування балки

Окрему увагу в сучасних дослідженнях приділено застосуванню штучних нейронних мереж для апроксимації кривих навантаження-розвантаження, оцінювання тангенціального модуля пружності та побудови нелінійних конститутивних залежностей. Інтеграція нейромережевих алгоритмів у структуру інформаційних систем дає змогу замінити аналітично складні матеріальні моделі їхніми числовими аналогами, що підвищує адаптивність і дозволяє ефективно працювати з матеріалами, поведінка яких визначається експериментальними даними.

Таким чином, сучасні підходи до побудови інформаційних систем для моделювання нелінійної деформації ґрунтуються на поєднанні класичних числових методів, інтелектуальних моделей матеріалів та механізмів автоматизованої обробки даних, що забезпечує зростання точності та функціональної гнучкості інженерного аналізу.

У цьому контексті метою дослідження є розробка інформаційної системи, що інтегрує обчислювальний модуль на основі штучних нейронних мереж для апроксимації петлі гістерезису гумового матеріалу та її подальшого використання у розрахунку міцності конструктивних елементів.

2. Архітектура інформаційної системи.

Розроблена інформаційна система реалізована як кросплатформовий настільний застосунок (Desktop Application), що побудований на принципах модульності та об'єктно-орієнтованого програмування (ООП). Така архітектура забезпечує чітке розмежування трьох ключових рівнів: представлення даних (Presentation Layer), обробки та валідації даних (Business Logic Layer) та обчислювального ядра (Calculation Core).

Структура системи складається з трьох основних функціональних блоків (рис. 2):

I. *Модуль графічного інтерфейсу користувача (GUI)* – відповідає за інтерактивну взаємодію з дослідником або інженером, візуалізацію топології конструкції та відображення результатів розрахунку.

II. *Модуль управління даними та валідації* – забезпечує цілісність вхідних даних, їх серіалізацію у формат CSV та підготовку структур для обчислювального ядра.

III. *Обчислювальний модуль (Solver Interface)* – виступає проміжним шаром між інтерфейсом та бібліотекою скінченно елементного аналізу,

інтегруючи нейромережевий апроксиматор у ітераційний процес розв'язання.

Програмна реалізація виконана на мові Python із застосування фреймворку PySide6 (Qt for Python) [15], що забезпечує нативну швидкодію та гнучкі можливості налаштування віджетів інтерфейсу.

2.1. Організація компонентів інтерфейсу та валідація даних. Критично важливим для забезпечення коректності інженерних розрахунків є етап попередньої обробки даних. Для цього в архітектурі системи застосовано патерн делегування (Delegation Pattern) [16], що дозволяє динамічно контролювати введення даних. Головне вікно програми (Main Window) агрегує три основні таблиці: вузли (Nodes), навантаження (Loads) та стержневі елементи (Bars).

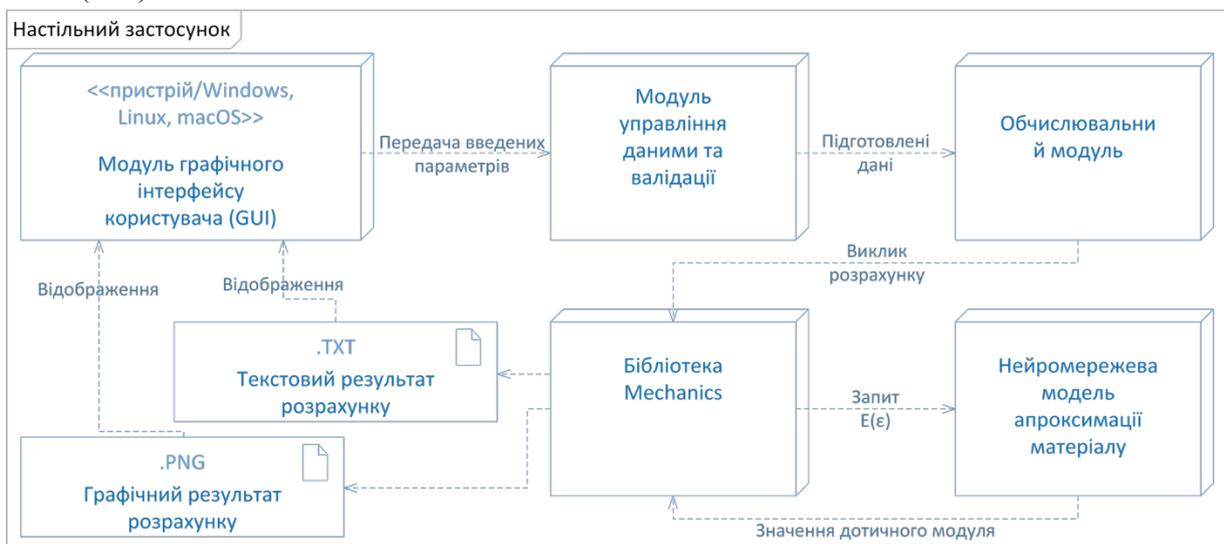


Рис. 2 – UML-діаграма розгортання настільного застосунку

2.2. Інтеграція з нейромережевим розв'язувачем. Взаємодія між інтерфейсом та математичним ядром реалізована через дворівневу систему обробки даних. Система формує проміжний текстовий опис задачі у спеціалізованому форматі, який підлягає суворому контролю перед передачею до основного парсера бібліотеки Mechanics. Це дозволяє виявити помилки на ранніх етапах, що можуть виникнути внаслідок некоректної валідації на попередніх етапах обробки даних, та надати користувачу узагальнені, інформативні повідомлення про некоректні вхідні дані до ініціалізації обчислювального ядра. Процес розрахунку ініціюється викликом методу calculate() та включає наступні етапи (рис. 3):

I. *Агрегація та серіалізація даних* – збір параметрів із таблиць інтерфейсу та генерація структурованого вхідного файлу опису задачі.

II. *Синтаксична верифікація* – застосування алгоритму попереднього парсингу, який базується на регулярних виразах. Модуль перевіряє формат запису вузлів, навантажень та стержнів, а також контролює наявність та послідовність блоків даних. Саме цей етап

Для кожної таблиці реалізовано спеціалізовані класи-делегати, що наслідують QStyledItemDelegate:

- *DoubleValidatorDelegate* – забезпечує введення лише числових значень з плаваючою комою та автоматичну обробку різних роздільників дробової частини, мінімізуючи помилку парсингу.

- *NodeIDComboBoxDelegate* – динамічно формує списки доступних ідентифікаторів вузлів, гарантуючи топологічну зв'язність схеми та коректність прикладення навантажень.

Такий підхід дозволяє виявляти помилки ще на етапі вводу, розвантажуючи обчислювальне ядро від обробки виключних ситуацій, пов'язаних із некоректним форматом даних.

дозволяє виявляти помилки форматування та надавати користувачеві деталізовані, локалізовані повідомлення про некоректні вхідні дані та завершити виконання з помилкою.

III. *Ініціалізація нейромережевого середовища* – створення екземпляру класу NNSolution, що завантажує попередньо навчені ваги нейронної мережі для апроксимації матеріальної моделі.

IV. *Побудова скінченно-елементної моделі* – парсинг вже валідного вхідного файлу внутрішніми засобами бібліотеки та формування об'єктного графу конструкції.

V. *Інтеграція розв'язання* – виконання методу solve бібліотеки Mechanics де на кожному кроці для оновлення матриці жорсткості викликається метод Ньютона-Рафсона для розрахунку дотичного модулю з використанням нейромережевого модулю.

Результат моделювання, включаючи поля переміщень та напружень, експортуються у графічні та текстові звіти, які асинхронно завантажуються у відповідні панелі інтерфейсу для подальшого аналізу користувачем.

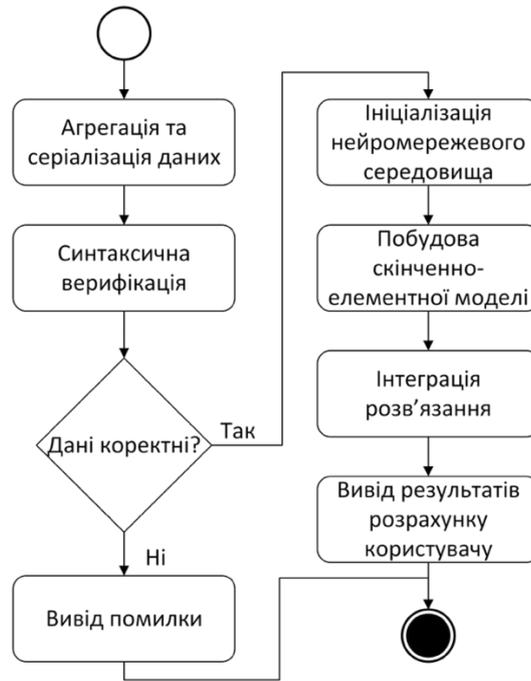


Рис. 3 – UML-діаграма активності інтеграції з ШНМ

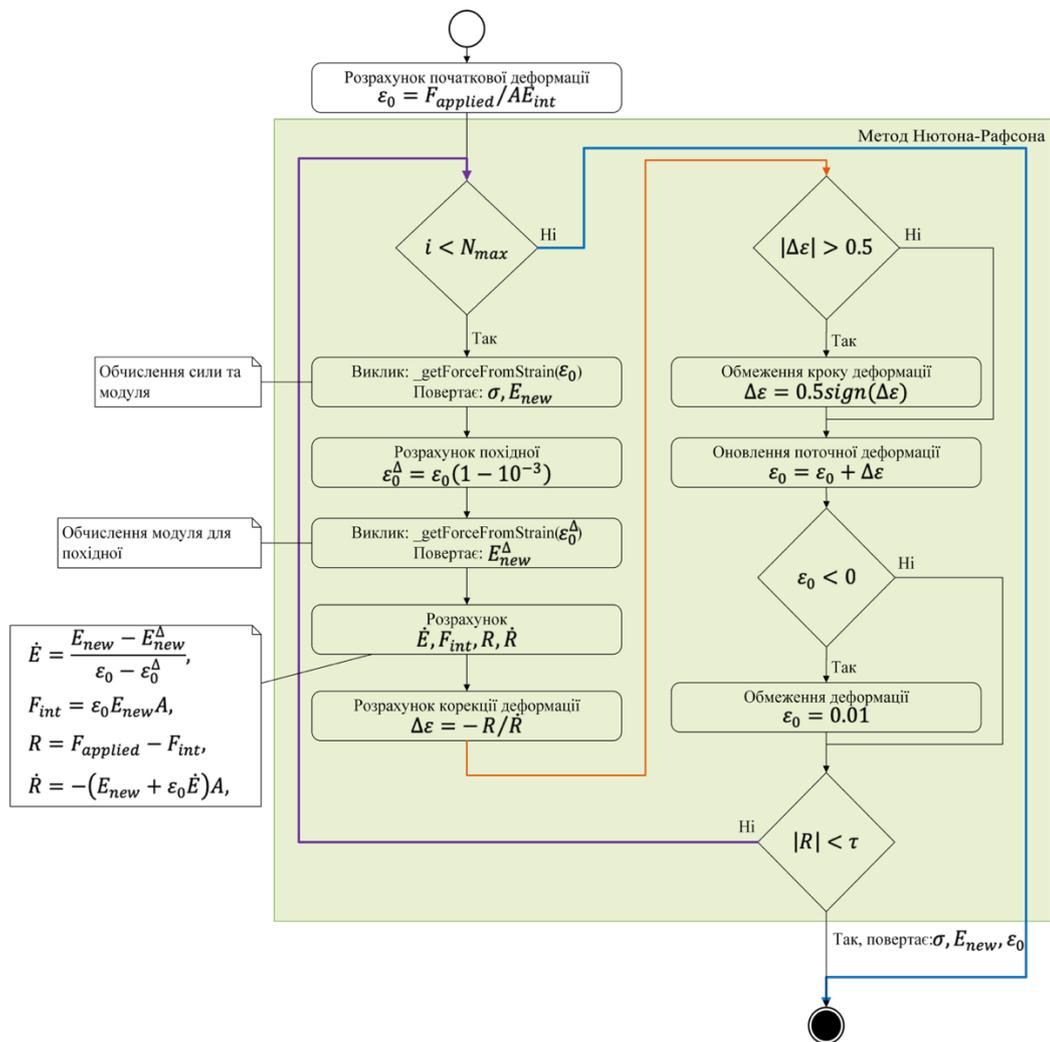


Рис. 4 – UML-діаграма активності алгоритму Ньютона-Рафсона

3. Алгоритмічне забезпечення та програмна реалізація гібридного розв'язувача. Основним обчислювальним компонентом розробленої інформаційної системи є гібридний модуль, який реалізує параметричну змінність матеріальних характеристик під час формування матриць жорсткості. На відміну від традиційних підходів, де модуль пружності є фіксованою величиною, запропонована архітектура передбачає використання адаптивного алгоритму, що поєднує об'єктну модель конструкції з нейромережовим прогнозним модулем.

3.1. Інтеграція нейромережевого компонента в обчислювальний процес. Програмна реалізація базується на перевизначенні методу побудови глобальної матриці жорсткості у бібліотеці скінченних елементів. Під час формування локальних матриць система автоматично ініціює процедуру отримання актуального модуля пружності шляхом звернення до нейромережевого компонента.

I. *Оцінювання параметрів навантаження.* Система аналізує топологію елемента та агрегує дані про зовнішні дії, що визначають значення прикладеної сили.

II. *Запит до нейромережевого модуля.* Здійснюється виклик функції `compute_modulus_newton`, яка повертає прогнозоване значення модуля пружності на основі поточних станів деформації.

III. *Оновлення параметрів елемента.* Отриманий модуль пружності передається у модель стержня, після чого формується локальна матриця жорсткості, що надалі передається до глобального збирача.

3.2. Реалізація інтеграційного керування та забезпечення стійкості обчислень. Оскільки нейромережева модель повертає прогноз напружень, система реалізує ітераційний контур на основі методу Ньютона-Рафсона для визначення рівноважної деформації. З алгоритмічної точки зору це відповідає процедурі пошуку кореня функції нев'язки з адаптивним регулюванням величини коригування.

Програмний алгоритм включає такі кроки (рис. 4):

I. *Ініціалізація.* Формується початкове наближення деформації шляхом лінійної екстраполяції значень.

II. *Ітераційний цикл взаємодії з нейромережею.* На кожній ітерації виконуються два послідовні обчислення:

a. прогноз напружень для поточної деформації;

b. прогноз напружень для збуреного стану, що забезпечує можливість чисельного наближення похідної відгуку.

Така схема компенсує відсутність аналітичних градієнтів у нейромережовій моделі.

III. *Стабілізація переходів між ітераціями.* Для забезпечення стійкості обчислень застосовується обмеження максимально допустимого кроку корекції. Якщо величина зміни деформації перевищує встановлений поріг (0.5), система автоматично зменшує цей крок до допустимого рівня (± 0.5).

IV. *Обробка виняткових ситуацій.* У разі відсутності збіжності протягом граничної кількості ітерацій система фіксує параметри помилки та припиняє обчислення для відповідного елемента, що унеможливує поширення нестабільності на весь розв'язувач. Після фіксації відмови елемент автоматично переводиться у режим використання наперед визначених користувачем параметрів матеріалу, що були встановлені у графічному інтерфейсі. Це забезпечує можливість завершення глобального розрахунку без втрати цілісності моделі.

3.3. Потік даних під час формування матриці жорсткості. Після завершення ітераційного процесу прогноз нейромережі трансформується у форму, необхідну для обчислювального ядра методу скінченних елементів. Значення модуля пружності перетворюється у тензорну структуру з використанням непрямих косинусів елемента, після чого формується матриця жорсткості розміру 4×4 . Таким чином, обчислювальна схема зберігає сумісність з традиційними інтерфейсами FEM-ядра, тоді як адаптивне оновлення матеріальних параметрів відбувається приховано для базових компонентів розв'язувача.

4. Проектування інтерфейсу користувача та особливості програмної реалізації. Розробка клієнтської частини інформаційної системи спрямована на формування інженерно орієнтованого графічного середовища (рис. 5), яке забезпечує коректне введення структурних даних, контроль консистентності моделі та відображення результатів чисельного аналізу. Програмну реалізацію виконано мовою Python із використанням бібліотеки PySide6 (Qt for Python) [15], що гарантує кросплатформність, підтримку нативних графічних компонентів та сувору подієву модель обробки користувацьких дій.

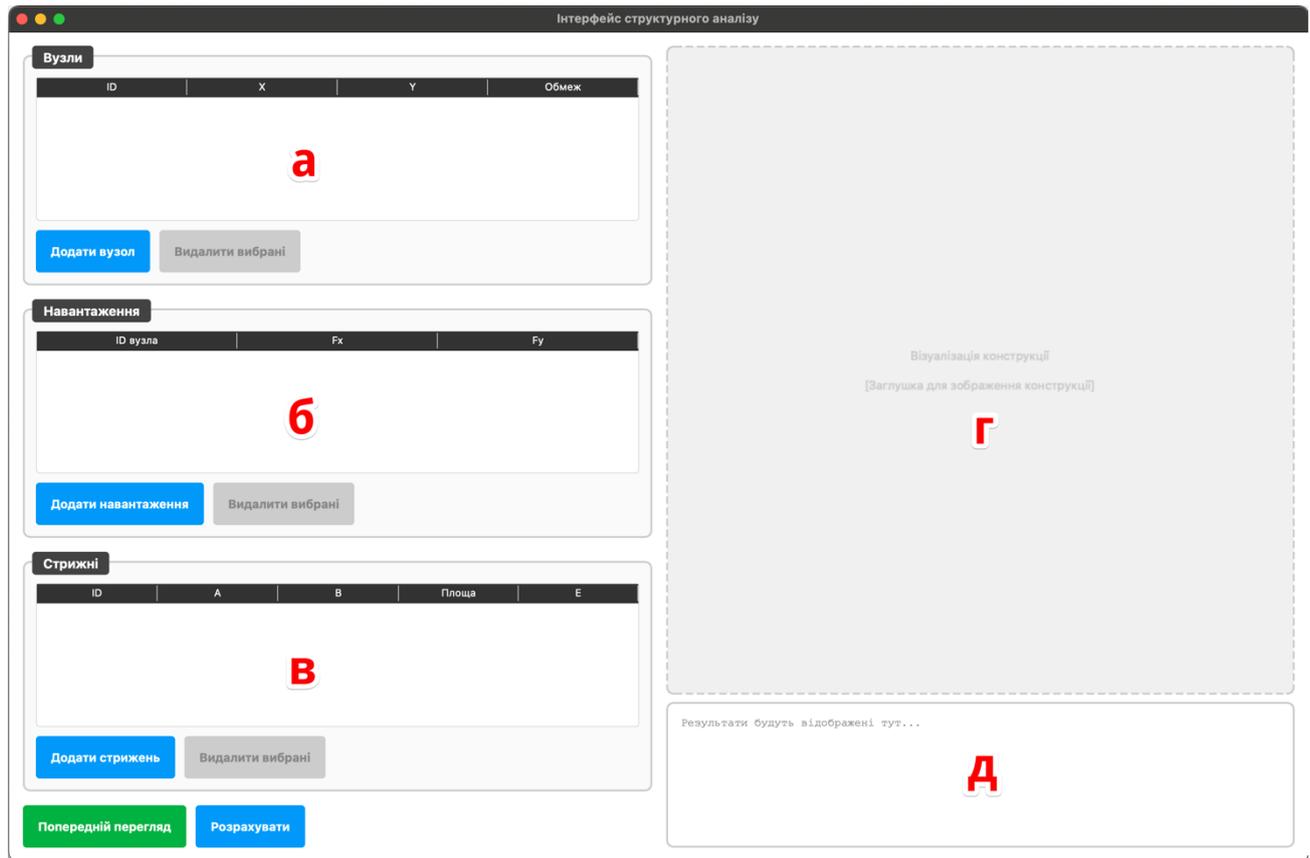


Рис. 5 – Інтерфейс застосунку, в якому: а – таблиця вузлів, б – таблиця навантажень, в – таблиця стрижнів, г – графічний модуль який відображає попередній перегляд моделі або результат розрахунку, д – текстовий опис результатів розрахунку.

Побудова конфігурації та розподілу напружено-деформованого стану здійснюється у вигляді векторних (SVG) або растрових (PNG) зображень. Згенеровані зображення інтегруються у віджет перегляду, який підтримує масштабування, що забезпечує можливість детальної оцінки геометричних змін та поведінки окремих елементів конструкції. Такий підхід дозволяє інженеру отримати інтуїтивно зрозумілу візуальну інтерпретацію результатів моделювання.

Паралельно із графічною візуалізацією система генерує детальний текстовий аналітичний звіт, що містить повний набір числових результатів розрахунку. Формат звіту організовано за двома логічними блоками – вузли (Node) та стрижневі елементи (Bar).

У блоці Node для кожного вузла подаються такі характеристики:

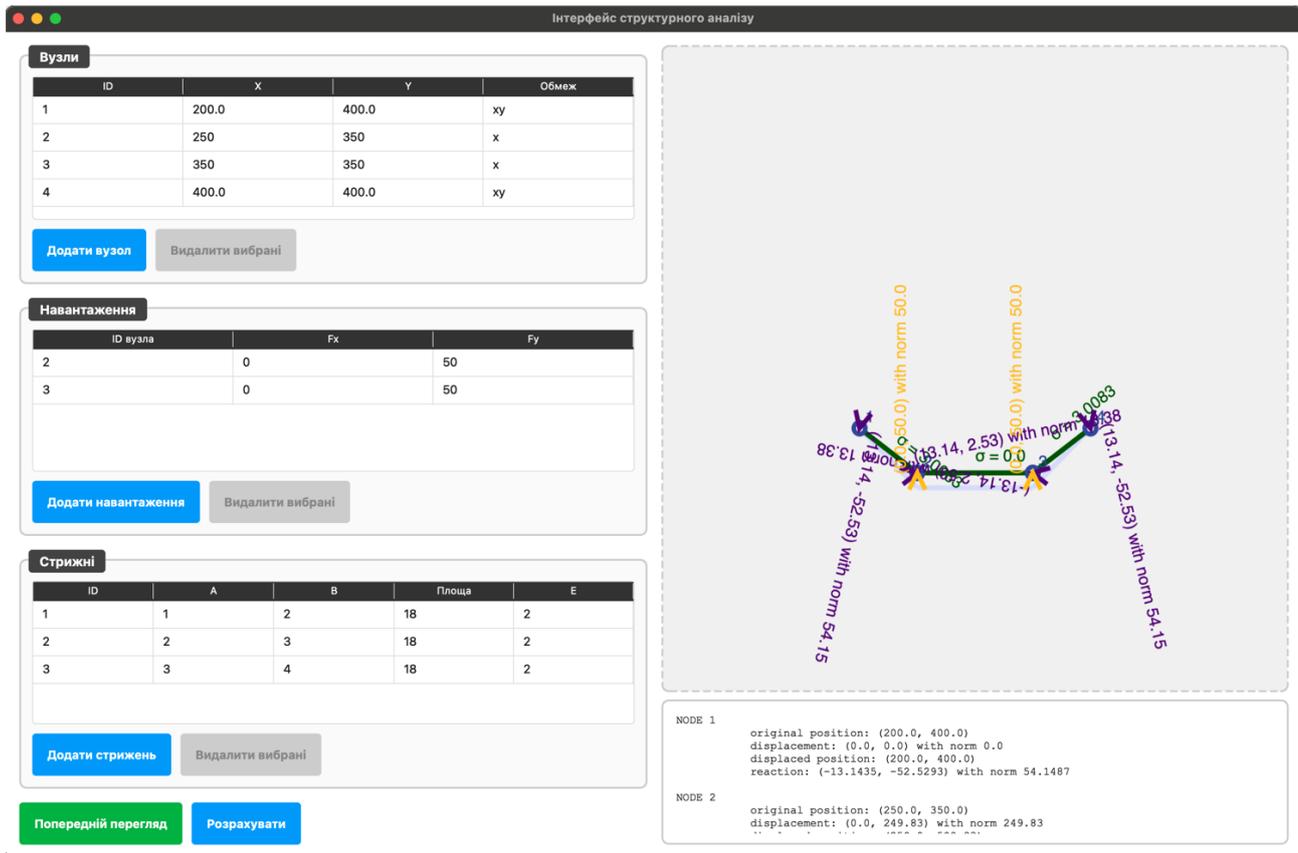
- original position – початкові координати вузла у глобальній системі;
- displacement – обчислений вектор переміщень та його евклідова норма;
- displaced position – положення вузла після деформації;
- reaction – реакційні зусилля, що виникають у закріплених вузлах, разом із нормою відповідного вектора.

У блоці Bar щодо кожного стрижневого елемента наведено:

- Δl (elongation) – абсолютна зміна довжини елемента;
- ε (strain) – відносна деформація;
- σ (stress) – напруження, визначене відповідно до прогнозованих або уточнених значень модуля пружності;
- позначення режиму роботи елемента (розтяг або стиск) за допомогою графічного маркера.

Така форма подання забезпечує можливість повного відтворення результатів розрахунку, їхньої подальшої технічної експертизи та архівування. Структурований вигляд звіту (рис. 6) сприяє швидкому аналізу поведінки конструкції, дозволяючи інженеру оцінити як глобальні переміщення, так і локальні силові фактори в елементах.

Для підвищення зручності взаємодії з користувачем під час виконання обчислювальних процедур значної тривалості застосовується асинхронне сповіщення про стан процесу. Зокрема, використовується діалоговий компонент `QProgressDialog`, який інформує про перебіг ітерацій, блокує доступ до елементів інтерфейсу, що можуть змінити структуру моделі, та запобігає виникненню помилок, пов'язаних із конкретними подіями. Реалізація такого механізму забезпечує передбачувану та керовану взаємодію користувача з системою під час виконання розрахунків.



- Myocardium." *International Journal for Numerical Methods in Engineering* 126.14 (2025): e70067. <https://doi.org/10.1002/nme.70067>
5. Haghghat, Ehsan, and Ruben Juanes. "SciANN: A Keras/TensorFlow wrapper for scientific computations and physics-informed deep learning using artificial neural networks." *Computer Methods in Applied Mechanics and Engineering* 373 (2021): 113552. <https://doi.org/10.1016/j.cma.2020.113552>
 6. Mendizabal, Andrea, Pablo Márquez-Neila, and Stéphane Cotin. "Simulation of hyperelastic materials in real-time using deep learning." *Medical image analysis* 59 (2020): 101569. <https://doi.org/10.1016/j.media.2019.101569>
 7. Im, Sunyoung, et al. "Neural network constitutive model for crystal structures." *Computational Mechanics* 67.1 (2021): 185-206. <https://doi.org/10.1007/s00466-020-01927-w>
 8. Sibuet, Nicolas, et al. "Discrete Physics-Informed Training for Projection-Based Reduced-Order Models with Neural Networks." *Axioms* 14.5 (2025): 385. <https://doi.org/10.3390/axioms14050385>
 9. Jang, Dong Phill, Piemaan Fazily, and Jeong Whan Yoon. "Machine learning-based constitutive model for J2-plasticity." *International Journal of Plasticity* 138 (2021): 02919. <https://doi.org/10.1016/j.ijplas.2020.102919>
 10. Xu, Kailai, Daniel Z. Huang, and Eric Darve. "Learning constitutive relations using symmetric positive definite neural networks." *Journal of Computational Physics* 428 (2021): 110072. <https://doi.org/10.1016/j.jcp.2020.110072>
 11. Leng, Yue, et al. "Predicting the mechanical properties of biopolymer gels using neural networks trained on discrete fiber network data." *Computer Methods in Applied Mechanics and Engineering* 387 (2021): 114160. <https://doi.org/10.1016/j.cma.2021.114160>
 12. Chung, Ingyun, Sunyoung Im, and Maenghyo Cho. "A neural network constitutive model for hyperelasticity based on molecular dynamics simulations." *International Journal for Numerical Methods in Engineering* 122.1 (2021): 5-24. <https://doi.org/10.1002/nme.6459>
 13. Ypma, Tjalling J. "Historical development of the Newton–Raphson method." *SIAM review* 37.4 (1995): 531-551. <https://doi.org/10.1137/1037125>
 14. Orbaiceta, Ángel Sola, and ProQuest. *Hardcore Programming for Mechanical Engineers: Build Engineering Applications from Scratch*. No Starch Press, 2021.
 15. Qt for Python. PySide6 - Official Python module for Qt 6. URL: <https://doc.qt.io/qtforpython-6> (accessed: 29.11.2025)
 16. Gamma, Erich. *Design patterns: elements of reusable object-oriented software*. Vol. 431. Addison-Wesley, 1995.
 3. Haghghat, Ehsan, et al. "A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics." *Computer Methods in Applied Mechanics and Engineering* 379 (2021): 113741. <https://doi.org/10.1016/j.cma.2021.113741>
 4. Gültekin, Osman, et al. "A Physics-Informed Neural Network Model for the Anisotropic Hyperelasticity of the Human Passive Myocardium." *International Journal for Numerical Methods in Engineering* 126.14 (2025): e70067. <https://doi.org/10.1002/nme.70067>
 5. Haghghat, Ehsan, and Ruben Juanes. "SciANN: A Keras/TensorFlow wrapper for scientific computations and physics-informed deep learning using artificial neural networks." *Computer Methods in Applied Mechanics and Engineering* 373 (2021): 113552. <https://doi.org/10.1016/j.cma.2020.113552>
 6. Mendizabal, Andrea, Pablo Márquez-Neila, and Stéphane Cotin. "Simulation of hyperelastic materials in real-time using deep learning." *Medical image analysis* 59 (2020): 101569. <https://doi.org/10.1016/j.media.2019.101569>
 7. Im, Sunyoung, et al. "Neural network constitutive model for crystal structures." *Computational Mechanics* 67.1 (2021): 185-206. <https://doi.org/10.1007/s00466-020-01927-w>
 8. Sibuet, Nicolas, et al. "Discrete Physics-Informed Training for Projection-Based Reduced-Order Models with Neural Networks." *Axioms* 14.5 (2025): 385. <https://doi.org/10.3390/axioms14050385>
 9. Jang, Dong Phill, Piemaan Fazily, and Jeong Whan Yoon. "Machine learning-based constitutive model for J2-plasticity." *International Journal of Plasticity* 138 (2021): 102919. <https://doi.org/10.1016/j.ijplas.2020.102919>
 10. Xu, Kailai, Daniel Z. Huang, and Eric Darve. "Learning constitutive relations using symmetric positive definite neural networks." *Journal of Computational Physics* 428 (2021): 110072. <https://doi.org/10.1016/j.jcp.2020.110072>
 11. Leng, Yue, et al. "Predicting the mechanical properties of biopolymer gels using neural networks trained on discrete fiber network data." *Computer Methods in Applied Mechanics and Engineering* 387 (2021): 114160. <https://doi.org/10.1016/j.cma.2021.114160>
 12. Chung, Ingyun, Sunyoung Im, and Maenghyo Cho. "A neural network constitutive model for hyperelasticity based on molecular dynamics simulations." *International Journal for Numerical Methods in Engineering* 122.1 (2021): 5-24. <https://doi.org/10.1002/nme.6459>
 13. Ypma, Tjalling J. "Historical development of the Newton–Raphson method." *SIAM review* 37.4 (1995): 531-551. <https://doi.org/10.1137/1037125>
 14. Orbaiceta, Ángel Sola, and ProQuest. *Hardcore Programming for Mechanical Engineers: Build Engineering Applications from Scratch*. No Starch Press, 2021.
 15. Qt for Python. PySide6 - Official Python module for Qt 6. URL: <https://doc.qt.io/qtforpython-6> (accessed: 29.11.2025)
 16. Gamma, Erich. *Design patterns: elements of reusable object-oriented software*. Vol. 431. Addison-Wesley, 1995.

References (transliterated)

1. Dettmer, Wulf G., et al. "A framework for neural network based constitutive modelling of inelastic materials." *Computer Methods in Applied Mechanics and Engineering* 420 (2024): 116672. <https://doi.org/10.1016/j.cma.2023.116672>
 2. Stöcker, Julien Philipp, Jakob Platen, and Michael Kaliske. "Introduction of a recurrent neural network constitutive description within an implicit gradient enhanced damage framework." *Computers & Structures* 289 (2023): 107162. <https://doi.org/10.1016/j.compstruc.2023.107162>
- Надійшла (received) 07.12.2025
Прийнята до друку (accepted) 15.12.2025
Опублікована (published) 29.12.2025

Відомості про авторів / About the Authors

Погребняк Сергій Віталійович (Serhii Pohrebniak) – Національний технічний університет «Харківський політехнічний інститут», аспірант кафедри «Математичне моделювання та інтелектуальні обчислення в інженерії»; м. Харків, Україна; тел.: +380 (057) 707-68-79 ORCID: <https://orcid.org/0000-0003-1612-3075>; e-mail: serhii.pohrebniak@khp.edu.ua.

Водка Олексій Олександрович (Vodka Oleksii) – кандидат технічних наук, доцент, Національний технічний університет «Харківський політехнічний інститут», завідувач кафедри Математичного моделювання та інтелектуальних обчислень в інженерії; м. Харків, Україна; тел.: (057) 707-68-79; ORCID: <https://orcid.org/0000-0002-4462-9869>; e-mail: oleksii.vodka@gmail.com