

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Програмний комплекс для математичного моделювання енергій
активації у цеолітах на основі різних підходів з використанням мови
програмування Python та системи Wolfram Mathematica

Виконала: студентка IV курсу, групи СП-43
спеціальності 121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

		Хемій С.Б.
	(підпис)	(прізвище та ініціали)
Керівник		Бойко І.В.
	(підпис)	(прізвище та ініціали)
Нормоконтроль		Стоянов Ю.М.
	(підпис)	(прізвище та ініціали)
Завідувач кафедри		Петрик М.Р.
	(підпис)	(прізвище та ініціали)
Рецензент		
	(підпис)	(прізвище та ініціали)

АНОТАЦІЯ

Програмний комплекс для математичного моделювання енергій активації у цеолітах на основі різних підходів з використанням мови програмування Python та системи Wolfram Mathematica // Кваліфікаційна робота освітнього рівня «Бакалавр» // Хемій Софія Богданівна // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра прогограмної інженерії, група СП-43 // Тернопіль, 2025 // С. 86, рис. – 24, табл. – 0 , кресл. – 0, додат. – 3, бібліогр. – 28.

Ключові слова: спектральні характеристики, математична модель, ефективний потенціал.

У даній кваліфікаційній роботі на здобуття освітнього ступеня бакалавра виконано детальний огляд та аналіз математичних моделей, методів вирішення проблем із оптимізацією моделей, програмних рішень для застосування у предметній області електроніки низьковимірних структур. На основі цього розроблено архітектуру та компоненти комплексного програмного комплексу, призначеного для математичного моделювання спектральних характеристик електронних станів у квантових ямах із застосуванням різних ефективних потенціалів. Розглянуто широкий спектр ефективних потенціалів, зокрема: гармонійний осцилятор, ангармонійний осцилятор, потенціал Пешля-Теллера, модифікований потенціал Пешля-Теллера, а також потенціали Морзе та Леннарда-Джонса.

Кожен з компонентів програмного комплексу дозволяє користувачам змінювати вхідні фізичні та геометричні параметри відповідно до розроблених математичних моделей та типів функціональних матеріалів, що використовуються. Крім того, програмна система дозволяє зручно та ефективно візуалізувати ефективні потенціали, прикладені до потенціальних ям, проводити розрахунки залежностей електронних спектрів від вхідних параметрів та генерувати їх графічні представлення.

На основі створених програмних модулів у даній роботі було спроектовано, а потім сконструйовано програмний комплекс для безпосередньої роботи в галузі нано- та мікроелектроніки, як в аспекті інженерних, так і суто наукових застосувань.

ABSTRACT

Software package for mathematical modelling of activation energies in zeolites based on different approaches using Python programming language and Wolfram Mathematica // Qualification work of the educational level "Bachelor" // Khemii Sofiia Bohdanivna // Ivan Pulyu Ternopil National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Software Engineering , group SP-43 // Ternopil, 2025 // p.– 86 , fig. – 24 , tab. – 0, blueprints – 0, add. - 3, ref. – 28.

Keywords: spectral characteristics, mathematical model, effective potential.

This bachelor's qualification work provides a detailed review and analysis of mathematical models, methods for solving model optimization problems, and software solutions for application in the subject area of low-dimensional structure electronics. Based on this, an architecture and components of a comprehensive software complex designed for mathematical modeling of spectral characteristics of electronic states in quantum wells using various effective potentials have been developed. A wide range of effective potentials is considered, including: harmonic oscillator, anharmonic oscillator, Pöschl-Teller potential, modified Pöschl-Teller potential, as well as Morse and Lennard-Jones potentials.

Each component of the software complex allows users to change input physical and geometric parameters according to the developed mathematical models and types of functional materials used. In addition, the software system allows for convenient and efficient visualization of effective potentials applied to potential wells, calculations of electronic spectra dependencies on input parameters, and generation of their graphical representations.

Based on the created software modules, this work has designed and then constructed a software complex for direct work in the field of nano- and microelectronics, both in terms of engineering and purely scientific applications.

ЗМІСТ

ВСТУП.....	9
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ. ПРОГРАМНА РЕАЛІЗАЦІЯ МАТЕМАТИЧНИХ МОДЕЛЕЙ ВИНИКАЮЧИХ В ЕЛЕКТРОНІЦІ НИЗЬКОВИМІРНИХ СТРУКТУР	11
1.1 Сучасна технологічна основа, інженерні задачі, що потребують застосування методів інженерії програмного забезпечення в актуальних задачах електроніки	11
1.2 Принципи побудови математичних моделей, що виникають у задачах, реалізуючих роботу пристроїв нанoeлектроніки, їх числова та програмна реалізація.	14
1.3 Підходи до розробки програмного забезпечення та кінцевого математичного моделювання у задачах нанoeлектроніки. Найбільш часто використовувані програмні системи. Постановка проблеми та її актуальність.	17
РОЗДІЛ 2 РОЗРОБКА МАТЕМАТИЧНИХ МОДЕЛЕЙ ДЛЯ ЧИСЛОВОГО РОЗВ'ЯЗАННЯ СТАЦІОНАРНОГО РІВНЯННЯ ШРЕДІНГЕРА З ДОВІЛЬНИМ ЛОКАЛЬНИМ ПОТЕНЦІАЛОМ. ПРОЕКТУВАННЯ АРХІТЕКТУРИ ПРОГРАМНОЇ СИСТЕМИ, ЯКА РЕАЛІЗУЄ КОМПОНЕНТИ МАТЕМАТИЧНОЇ МОДЕЛІ ТА ЇХ ВІЗУАЛІЗАЦІЮ	24
2.1 Стаціонарне рівняння Шредінгера з довільним локальним потенціалом. Математична модель квантової ями, граничні умови та поняття про енергію активації.	24
2.2 Розробка різницевої схеми для математичної моделі із стаціонарним рівнянням Шредінгера та заміна її сітковою задачею. Стійкість різницевої схеми та її модифікації.	28
2.3 Типові локальні потенціали, що використовуються у якості математичних моделей локалізованих квантових ям, їх візуалізація.	31
2.3.1 Гармонічний осцилятор	31

2.3.2 Ангармонічний осцилятор	34
2.3.3 Потенціал Пешля-Теллера	35
2.3.4 Модифікований потенціал Пешля-Теллера	36
2.3.5 Потенціал Морзе	38
2.3.6 Потенціал Леннарда-Джонса	39
РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ДЛЯ МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ЕНЕРГІЙ АКТИВАЦІЇ З РІЗНИМИ ПОТЕНЦІАЛАМИ.	42
3.1 Методологія розробки	42
3.2 Аналіз вимог для спеціалізованого програмного забезпечення з моделювання енергій активації.	44
3.3 Проектування архітектури програмної системи. Компоненти та функціонал програмної системи.	45
3.4 Розробка програмного комплексу для математичного моделювання енергій активації.	50
3.5 Тестування та верифікація	59
РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	61
4.1 Долікарська допомога при ураженні електричним струмом.	61
4.2 Інженерно-технічні рішення з охорони праці.	63
ВИСНОВКИ	65
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	67
ДОДАТКИ	71
ДОДАТОК А Лістинг програмного коду для візуалізації ефективних потенціалів та математичного моделювання енергій активації з різними потенціалами	72
ДОДАТОК Б Тези доповіді на конференції	83
ДОДАТОК В Диск з роботою	86

ВСТУП

Сучасні оптоелектронні та напівпровідникові пристрої відіграють значну роль в розвитку технологій. Розробка низьковимірних матеріалів в яких ключове значення відводиться квантовим ефектам привела до розробки транзисторів, інтегральних схем, що в свою чергу використовуються як складові частини лазерів, діодів та детекторів з дуже широкими частотними та температурними режимами їх роботи.

Сфера застосування цих приладів узагалі надзвичайно різноманітна: від лазерів із низькою інтенсивністю випромінювання, що застосовуються у медицині та біотехнологіях до детекторів високочастотних електромагнітних хвиль у каскадних напівпровідникових детекторах, що мають застосування у пристроях нічного бачення та особливо у військових технологіях в радарх, що працюють у вікнах прозорості атмосфери реалізуючи принцип “антистелз”.

Незважаючи на функціональну різноманітність пристроїв нано- та мікроелектроніки, їх основою, що відповідає за їхні частотні характеристики є квантові ями та їх узгоджені масиви. В результаті кінцевих електронний прилад має властивості, що по-перше визначені фізичними характеристиками використовуваних при його виробництві матеріалів, і по-друге, що дуже важливо, геометричними параметрами цих квантових ям.

Згадані питання в основному вирішуються в площині предметної області інформаційних технологій, зокрема інженерії програмного забезпечення. Для прикладу для вирощування наноструктур із наперед заданими властивостями слід виконувати їх математичне моделювання, що приводить до проблем із необхідністю розгляду таких проблем окремо, для кожного випадку конкретно взятих параметрів. В результаті виникає необхідність розробки програмних систем, які б задовольнили потреби в дослідників та інженерів на різних етапах їхньої роботи. Зокрема послідовна розробка математичних моделей, що характеризують різноманітні ефективні потенціали у квантових ямах дала б

можливість розробити складні програмні системи, які дозволяють користувачам візуалізувати та кількісно описувати різноманітні умови генерації чи детектування електромагнітних хвиль, режими роботи, маніпулювати вхідними фізичними та геометричними параметрами. Розробці архітектури такої програмної системи, її складових, що є програмною реалізацією математичних моделей електронних станів у квантових ямах із різноманітними потенціалами, а також розробці людино-машинного інтерфейсу присвячена дана кваліфікаційна робота бакалавра.

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ. ПРОГРАМНА РЕАЛІЗАЦІЯ МАТЕМАТИЧНИХ МОДЕЛЕЙ ВИНΙΚАЮЧИХ В ЕЛЕКТРОНІЦІ НИЗЬКОВИМІРНИХ СТРУКТУР

1.1 Сучасна технологічна основа, інженерні задачі, що потребують застосування методів інженерії програмного забезпечення в актуальних задачах електроніки

У сучасній електроніці останні два десятиліття набув значного поширення напрям застосування мікро- і наноструктур, що характеризуються своєю високотехнологічністю та активними зв'язками з іншими дисциплінами, особливо з засобами та методологіями інформаційних технологій. Згадані низьковимірні структури мають застосування у різноманітній інженерній та технологічній діяльності людини, наприклад у оптоелектронній техніці, напівпровідниковій електроніці, а також мають дуже важливе значення для технологій, що застосовуються у військовий час. Це пов'язано з тим, що нанодетектори та нанолазери, які створені на основі низьковимірних напівпровідникових структур [2-7], є функціональними елементами устаткування, призначеного для фіксування польоту ворожих військових літальних засобів (літаків та БПЛА) завдяки так званій «anti-stealth» технології [8-10], що є надзвичайно важливим для розвитку української науки та техніки у цей важкий час. Інше кардинально відмінне застосування такого типу структур має місце в медицині, де напівпровідникові лазери з успіхом застосовуються для здійснення високоточних операцій, особливо в онкології та як засіб ефективної доставки ліків у необхідне місце людського організму (так звані ефективні макромолекули) [11-13]. Приклади різноманітних за своєю формою та розмірами низькорозмірних структур показано на рисунку 1.1.

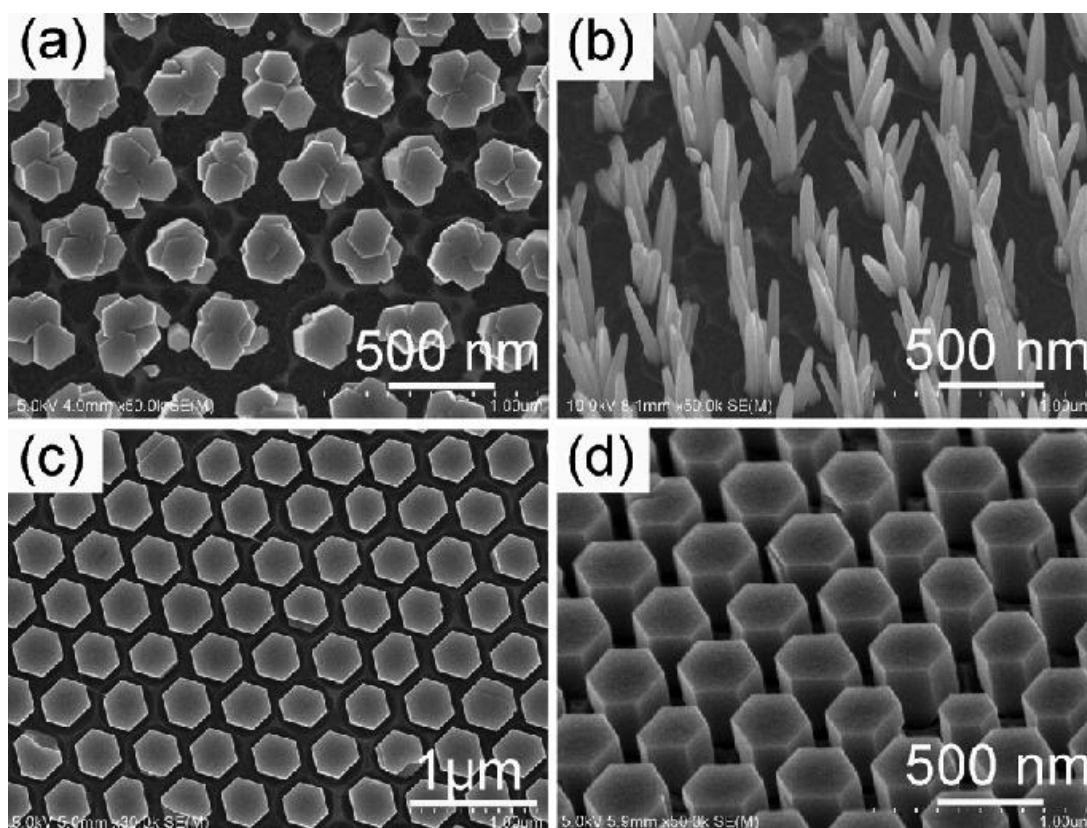


Рисунок 1.1 – Приклади вирощування низькорозмірних структур з різною симетрією, розмірами та формою

Щодо застосування методів інженерії програмного забезпечення, то її предметна область в плані застосування до задач нанoeлектроніки є однією з визначальних, адже як показує аналіз стану сучасних технологій, програмні системи та комплекси здійснюють супровід функціонування складових електроніки на всіх етапах їх життєвого процесу – починаючи від етапу їх створення та вирощування до моменту безпосередньої імплементації в конкретні прилади. Крім того, безпосереднє функціонування електронних приладів, основою яких є системи низької розмірності, абсолютно неможливе без застосування спеціалізованого програмного забезпечення, яке з одного боку забезпечує саму роботу, а з іншого боку максимальну ефективність цих приладів.

Програмне забезпечення, яке стосується засобів нанoeлектроніки, можна за його призначенням поділити на кілька груп. До першої групи слід віднести програмне забезпечення контрольного типу, яке повністю контролює процес та часові рамки під час створення низьковимірних систем [14-17] за наперед

заданими параметрами. Прикладом таких програмних систем є програмні комплекси, підтримувані компаніями Nextnano.de, Semisoft.org, Microandnanosoft [14-17]. Такі програмні комплекси є багатофункціональними і за своєю суттю є фактично набором прикладних програм.

Складність роботи з такими програмними комплексами полягає в тому, що над кожним пакетом програм можуть працювати десятки і сотні фахівців з інженерії програмного забезпечення та математичного моделювання, при чому їхні предметні області та галузі знань можуть не перетинатися. Це фактично змушує цілком по-іншому дивитись на такі розробки з точки зору менеджменту проектів програмного забезпечення, в основному з точки зору організації роботи. Приклад роботи із програмним забезпеченням, що розробляється корпорацією Nextnano, подано на рисунку 1.2. Слід відзначити, що таке програмне забезпечення відзначається досить великою швидкістю та хорошою візуалізацією результатів моделювання.

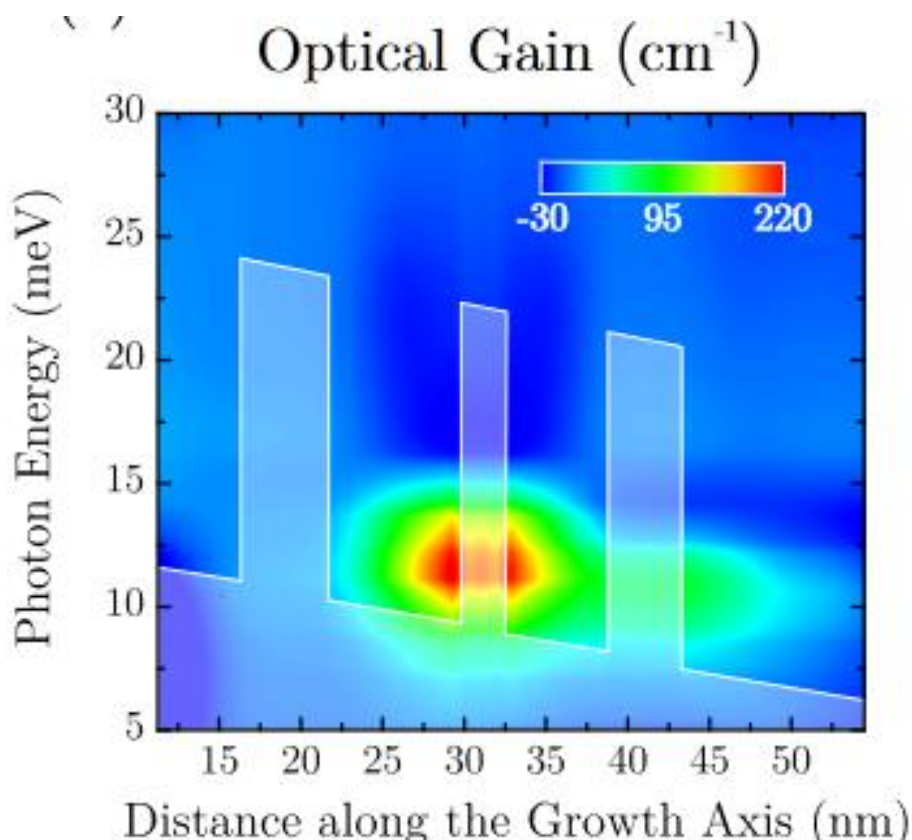


Рисунок 1.2 – Приклад розрахунку оптичного відгуку наноструктури за допомогою програмного забезпечення Nextnano

До недоліків цієї групи програмних систем слід віднести такі: їх велика дороговизна – навіть враховуючи багатомільйонні витрати на вирощування низьковимірних структур купівля програмного забезпечення або замовлення у згаданих компаній спеціалізованих програмних розробок потребує великих фінансових затрат. Крім того, користувач рідко коли може придбати для себе лише певний конкретний, потрібний лиш йому, пакет необхідних програм. У зв'язку з тим, що дана предметна область розвивається дуже швидко, доводиться кілька разів в рік оновлювати всю програмну систему, але фактично в ній залишається 90-95% програмного забезпечення, яке вже є неактуальним та не може бути ніде застосованим. Це становить серйозну проблему і потребує іншого підходу до контролю за процесами розробки такого програмного забезпечення.

1.2 Принципи побудови математичних моделей, що виникають у задачах, реалізуючих роботу пристроїв нанoeлектроніки, їх числова та програмна реалізація.

Математичні моделі, що реалізують задачі нано- та мікроелектроніки своєю основою зазвичай мають реалізацію можливостей розрахунків електронного спектру. В основному такі моделі базуються на відшуванні аналітичних або ж числових розв'язків стаціонарного чи залежного від часу рівняння Шредінгера. При цьому адекватність таких математичних моделей буде визначатись фундаментальними принципами квантової механіки [2].

Ми звикли до класичної фізики принаймні до появи квантової механіки. Класична фізика ґрунтувалася на законах руху Ньютона та математично виражалася диференціальними рівняннями, які описують, наприклад, рух електрона. Електрон приймається нескінченно малою частинкою в цьому випадку. Оскільки він є частинкою, його рух визначається не лише початковими умовами, а також і граничними умовами. У просторовій області він може не торкатися меж

локалізованого простору, навіть якщо він може із скінченною ймовірністю існувати поблизу однієї з меж згідно із постулатами квантової механіки, що визначають обмеження для хвильової функції.

Непараболічне рівняння Шредінгера, що не залежить від часу, використовується майже у всіх розрахунках зонної структури нанопристроїв з квантовими ямами. Для напівпровідникових лазерів розгляд випадків відхилення профілю квантової ями від прямокутної має важливе значення, оскільки великий розрив зони в квантових ямах призводить до станів, що знаходяться високо над краєм зони провідності, де згаданий ефект стає помітним і визначальним. Таке відхилення приводить до необхідності побудови математичних моделей для кожного випадку окремо. Така модифікація математичної моделі може бути уведена шляхом введення для електрона локальної ефективної маси згідно співвідношення [18]:

$$\frac{1}{m(E, z)} = \frac{1}{m(z)} \frac{1}{\{1 + \alpha(z)(E - U(z))\}}, \quad (1.1)$$

де $m(z)$ - ефективна маса електрона у локальній області, $\alpha(z)$ - підгоночний параметр моделі, E - енергія активації. Особливу увагу слід приділити залежності $U(z)$, яка становить собою ключову проблему із оптимізації математичної моделі та розрахунку енергій активації. Чим краще у даних умовах вибрана ця функція, тим точніше будуть розрахованими енергії активації, а також хвильові функції електрона, іншої квазічастинки чи взагалі молекули. В загальному випадку функцію $U(z)$ ще прийнято називати модельним потенціалом. Існує два підходи до описання модельного потенціалу за межами локальної області шириною w . В першому, спрощеному підході потенціал за межами області нескінченний, тобто:

$$U(z) = U(z) \rightarrow \infty, \quad \begin{matrix} z \rightarrow 0 \\ z \rightarrow w \end{matrix} \quad (1.2)$$

в такому разі рівняння Шредінгера суттєво спрощується і часто навіть вдається отримати його аналітичні розв'язки, проте такі математичні моделі дають лише якісні кількісні результати.

В другому підході потенціал за межами локальної області є скінченним, тобто:

$$U(z) = U(z) \rightarrow U_0, \quad (1.3)$$

$z \rightarrow 0$
 $z \rightarrow w$

або

$$U(z) \rightarrow U_1; U(z) \rightarrow U_2. \quad (1.4)$$

$z \rightarrow 0$
 $z \rightarrow w$

такі математичні моделі є більш реалістичним, проте вони здебільшого можуть бути реалізованими чисельно, а сама така реалізація вимагає застосування спеціалізованого програмного забезпечення або ж його розробки під конкретні такі завдання.

Метод стрільби, скоріш всього, є найпростішим методом вирішення проблеми програмної реалізації згаданих математичних моделей і використовується у значній частині робіт з проектування приладів електроніки через його низькі обчислювальні витрати. Також були продемонстровані інші підходи ітераційного пошуку, такі як пошук власних значень з аналогічними перевагами методу стрільби. Однак для високої точності кількість ітерацій у таких методах стає величезною, що робить їх непрактичними. Більш привабливим підходом є вирішення проблеми безпосередньо шляхом діагоналізації матриці Гамільтона з використанням програмних процедур лінійної алгебри, хоча цей підхід потребує лінійного чи потокового розпаралелення. Існує кілька методів, які використовують наближення Тейлора першого порядку, що залежить від енергії ефективної маси [19], але через природу цього наближення воно справедливе тільки для станів поблизу краю зони провідності. Хоча ці наближені методи добре

працюють для квантових ямних систем з малими зсувами зон, таких як GaAs, було показано, що вони не працюють для матеріалів з великими зсувами зон, таких як ті, які використовуються в нанoeлектроніці, де стани можуть існувати набагато вищій за край зони провідності. Альтернативний підхід полягає в лінеаризації кубічного дисперсійного рівняння, отриманого з непараболічного рівняння Шредінгера шляхом розширення порядку матриці, таким чином що може бути отриманий прямий і точний розв'язок. Загалом враховуючи варіативність та різноманітність просторового конфайнменту у таких математичних моделях слід однозначно мати на увазі можливість розгляду багатоточкових різницевих схем для рівняння Шредінгера з різними типами початкових і крайових умов для нього [18, 20].

1.3 Підходи до розробки програмного забезпечення та кінцевого математичного моделювання у задачах нанoeлектроніки. Найбільш часто використовувані програмні системи. Постановка проблеми та її актуальність.

Розробка та верифікація програмного забезпечення, що реалізує математичні моделі в нанoeлектроніці можна поділити на три найважливіші рівні, які реалізують найбільш базові підходи до його розробки.

Перший рівень стосується безпосередньої реалізації тієї чи іншої математичної моделі у вигляді коду, виконання якого дає розв'язання спектральної задачі для електрона та його розподілу ймовірності у локальній області. Алгоритми, які виникають на цьому етапі в більшості випадків реалізують відшукування коренів дисперсійних рівнянь методом стрільби, методом золотого перерізу чи методом дотичних. Для простих задач допустимим може бути застосування мов програмування доволі низького рівня, таких як C, C++, Fortran чи Python. Для прикладу, якщо маємо дисперсійне рівняння $\text{dispeq}(E)=0$ на проміжку $0 \leq E \leq U$, де U – скінчення висота потенціального бар'єра, то

псевдокод реалізуючий його розв'язання методом стрільби буде такий (див. рис. 1.3):

```

Input: dispeq(E), U, tolerance, max_iter
Output: E_solution

1. Define function dispeq(E):           # The dispersion equation
2. Set E_low = 0                        # Lower bound for E
3. Set E_high = U                       # Upper bound for E
4. Set iter = 0                         # Initialize iteration counter

5. While iter < max_iter:
    6. E_mid = (E_low + E_high) / 2     # Midpoint between E_low and E_high
    7. Calculate dispeq_low = dispeq(E_low) # Evaluate dispeq at E_low
    8. Calculate dispeq_mid = dispeq(E_mid) # Evaluate dispeq at E_mid

    9. If abs(dispeq_mid) < tolerance:   # Check if solution is close enough
        10. Return E_mid as E_solution # Solution found

    11. If dispeq_low * dispeq_mid < 0: # Check if the root lies between E_low and E_mid
        12. E_high = E_mid             # Update the upper bound
    13. Else:
        14. E_low = E_mid              # Update the lower bound

    15. iter = iter + 1                 # Increment iteration counter

16. Return "No solution found within tolerance"

```

Рисунок 1.3 – Реалізація розв'язання дисперсійного рівняння методом стрільби у псевдокодi

Аналогічним чином, практична реалізація цього коду мовою C++ є такою:

```

#include <iostream>
#include <cmath>

using namespace std;

double dispeq(double E) {
    return E * E - 2.0 * E + 1.0;
}

double solve_dispeq(double U, double tolerance, int max_iter) {
    double E_low = 0.0;
    double E_high = U;
    int iter = 0;

    while (iter < max_iter) {
        double E_mid = (E_low + E_high) / 2.0;
        double dispeq_low = dispeq(E_low);
        double dispeq_mid = dispeq(E_mid);
        if (std::abs(dispeq_mid) < tolerance) {
            return E_mid;
        }

        if (dispeq_low * dispeq_mid < 0) {
            E_high = E_mid;
        } else {
            E_low = E_mid;
        }
        iter++;
    }

    cerr << "Розв'язання не знайдено в межах точності за " << max_iter << " ітерацій."
    return NAN;
}

```

Рисунок 1.4 – Реалізація розв'язання дисперсійного рівняння методом стрільби мовою C++

Результуючий код, та отриманий результат можна вже безпосередньо використовувати для забезпечення наочності та якісної роботи з результатом. Проте, як можна бачити, застосування мов програмування нижнього рівня має

суттєвий недолік, який полягає в тому, що для кінцевої візуалізації треба застосовувати додаткові програмні засоби. В результаті значно погіршується портабельність коду, його варіативність та головне, безпосередньому користувачу дуже важко працювати з такою програмною системою.

У результаті такий підхід до розробки програмного забезпечення для наноелектроніки вижив себе і морально застарів, його застосування зараз є поодиноким. Як наслідок має місце застосування спеціальних пакетів комп'ютерної математики, таких як Wolfram Mathematica, Maple, MATLAB, що за своїм змістом та пакетами розширеннями є системами об'єктно-орієнтованого програмування тільки з тими особливостями, що звичні для програміста об'єкти за своїми властивостями є розширеними графікою, аналітичними виразами, модулями анімації та симуляції. Згадані математичні пакети становлять собою другий рівень реалізації математичних моделей, адже вони можуть реалізовувати графічно тримані дані від першого етапу програмної реалізації моделей так можуть виконувати це етап самостійно, адже ці системи також є потужними засобами програмування, в яких реалізація програмного коду здійснюється такими мовами програмування як C і C++.

Також слід зауважити, що математичні пакети, володіючи значними за варіативністю елементами для роботи з графікою дозволяють ефективно візуалізувати кожен проміжний та кінцевий етап розрахунків, таким чином мінімізуючи можливість виникнення помилок та забезпечують інтерактивний доступ користувача до програмної системи надаючи можливість широкої зміни параметрів реалізованої моделі. Приклад програмної реалізації математичної моделі, що виражена у формі дисперсійного рівняння з його подальшою візуалізацією у системі Wolfram Mathematica та програмним кодом для методу стрільби подано на рисунку 1.5.

```

dispeq[E_] := Sin[E] - 0.5
[си... експонента]

(*Метод бісекції для пошуку кореня дисперсійного рівняння*)
solveDispeq[U_, tolerance_, maxIter_] :=
Module[{E_low = 0, E_high, E_mid, iter = 0}, (*Ініціалізація верхньої межі*) E_high = U;
[програмний модуль]
(*Пошук кореня методом бісекції*) While[iter < maxIter, E_mid = (E_low + E_high) / 2;
[цикл-поки]
(*Перевірка, чи знайдено корінь*) If[Abs[dispeq[E_mid]] < tolerance, Return[E_mid] ×
[... абсолютне значення] [повернути управління]
If[dispeq[E_low] × dispeq[E_mid] < 0, E_high = E_mid, (*Якщо корінь між E_low та E_mid*) E_low = E_m
[умовний оператор]
(*Якщо корінь між E_mid та E_high*)];
iter++;];
Return["Розв'язання не знайдено"]];
[повернути управління]

(*Задаємо параметри*) ×
U = 10;
(*Верхня межа для пошуку*) tolerance = 10^-6;
(*Допуск*) maxIter = 100;
(*Максимальна кількість ітерацій*) (*Знаходимо розв'язання*)
solution = solveDispeq[U, tolerance, maxIter]

(*Графік дисперсійного рівняння на інтервалі [0,U]*) ×
Plot[dispeq[E], {E, 0, U}, PlotLabel → "Графік дисперсійного рівняння", AxesLabel → {"E", "dispeq(E)"},

```



Рисунок 1.5 – Реалізація розв'язання дисперсійного рівняння методом стрільби мовою C++ з візуалізацією у Wolfram Mathematica

Заключним, третім рівнем реалізації математичних моделей, що створені за допомогою систем комп'ютерної математики є програмні системи із розробленою користувачем архітектурою. Особливістю таких програмних систем, це проста і зручна взаємодія із зовнішнім користувачем, який може досить широко варіювати множиною вхідних параметрів. Такі програмні системи є повністю самостійними, їх компіляція не потребує наявності програмного середовища в якому вони були створені. Приклад такої програмної системи, що реалізує математичну модель

залежної від часу динаміки хвильового пакета у області із довільними розмірними параметрами подано на рисунку 1.6.

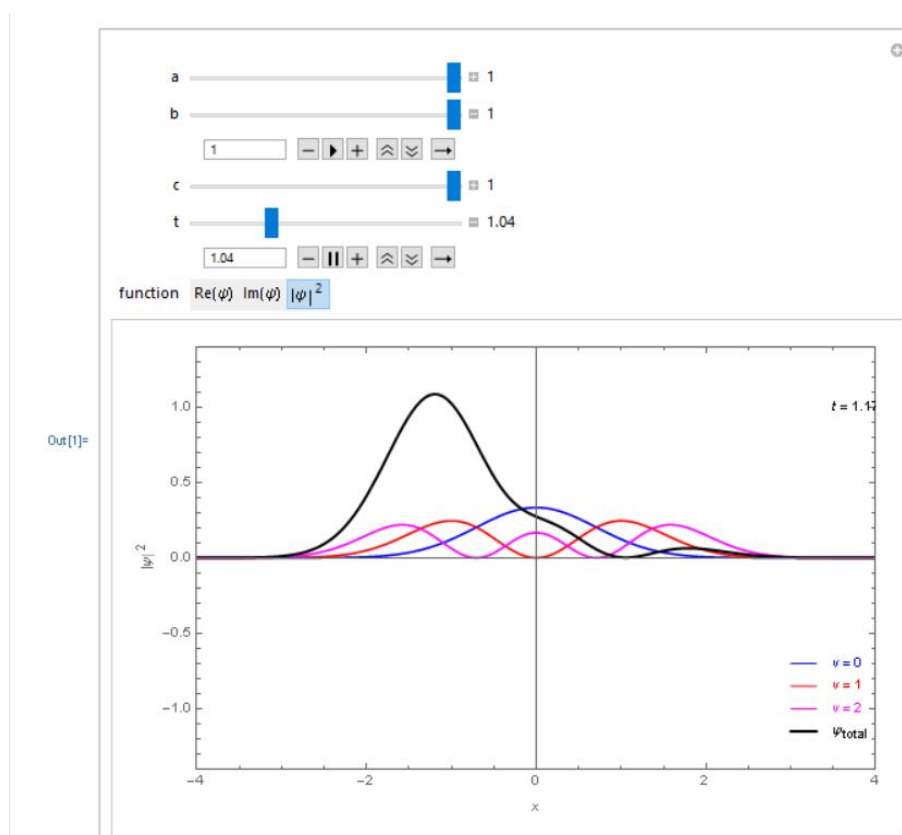


Рисунок 1.6 – Реалізація еволюції хвильового пакета у форму Гауссівської кривої

Таким чином, слід зробити висновок, що на даний момент в розробці програмного забезпечення в предметній області наноелектроніки є чітко виражена проблема, яка полягає в таких основних складових:

- відсутній чіткий, в тому числі числовий підхід до побудови математичних моделей електронних станів, енергій активації для структур із довільним просторовим конфайнментом та потенціальною енергією з довільною просторовою геометрією;
- відсутні послідовні алгоритми, які застосовні до таких математичних моделей у загальному випадку;
- відсутні архітектури програмних систем, що забезпечують числову реалізацію згаданих математичних моделей, їх візуалізацію, динамічну специфікацію параметрів моделей;

- відсутні програмні системи, які реалізують основні модельні просторові потенціали математичних моделей енергій активації, їх порівняння не здійснене, в тому числі й в плані базової верифікації.

Згадані складові проблеми послідовно вирішуються у даній дипломній роботі й становлять собою основу для здійснення наукової діяльності у даній важливій предметній області.

РОЗДІЛ 2 РОЗРОБКА МАТЕМАТИЧНИХ МОДЕЛЕЙ ДЛЯ ЧИСЛОВОГО РОЗВ'ЯЗАННЯ СТАЦІОНАРНОГО РІВНЯННЯ ШРЕДІНГЕРА З ДОВІЛЬНИМ ЛОКАЛЬНИМ ПОТЕНЦІАЛОМ. ПРОЕКТУВАННЯ АРХІТЕКТУРИ ПРОГРАМНОЇ СИСТЕМИ, ЯКА РЕАЛІЗУЄ КОМПОНЕНТИ МАТЕМАТИЧНОЇ МОДЕЛІ ТА ЇХ ВІЗУАЛІЗАЦІЮ

2.1 Стаціонарне рівняння Шредінгера з довільним локальним потенціалом. Математична модель квантової ями, граничні умови та поняття про енергію активації.

Розглядається одинична потенціальна яма, з довільним геометричним конфайнментом її дна. Приймається, що потенціальна енергія електрона у квантовій ямі визначається її конфайнментом, що виражається координатною залежністю $U = U(z)$ так як це подано на рисунку 2.1.

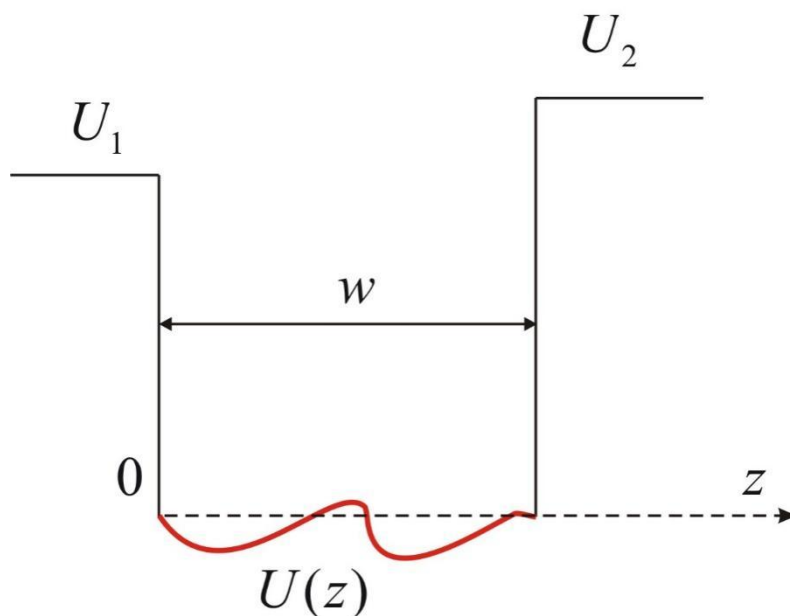


Рисунок 2.1 – Геометричний конфайнмент та енергетична схема потенціальної ями із стінками скінченної висоти

Відповідно спрощена модель локальної області потенціальної ями може бути отриманою із моделі на рисунку 2.1 при $U(z)_{z \rightarrow 0} \rightarrow \infty; U(z)_{z \rightarrow w} \rightarrow \infty$, що є

моделлю потенціальної ями, яка має безкінечно високі стінки. Геометричний схема такої локальної області подано на рисунку 2.2. В обох можливих моделях приймається, що потенціальна яма має однакову ширину рівну w .

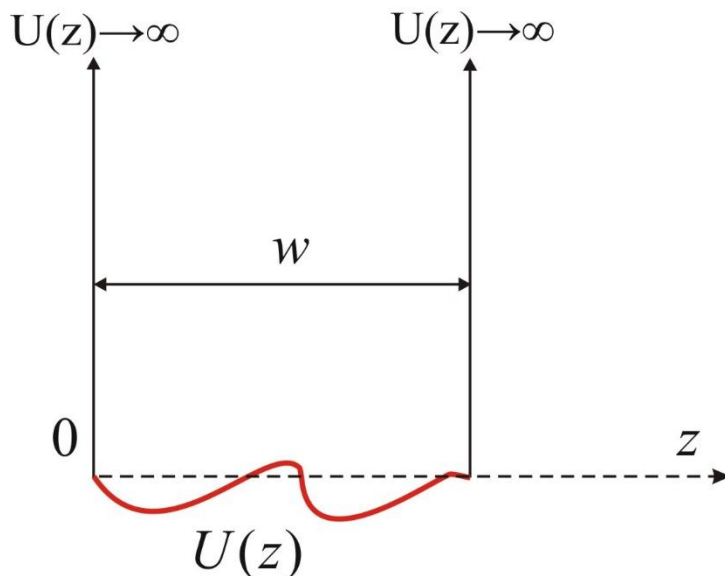


Рисунок 2.2 – Геометричний конфайнмент та енергетична схема потенціальної ями із стінками нескінченної висоти

Для даної локальної області визначення електронного спектру, енергії активації та хвильових функцій електрона зводиться до відшукування розв'язків стаціонарного рівняння Шредінгера:

$$\hat{H}(z)\Psi(z) = E\Psi(z), \quad (2.2)$$

де Гамільтоніан електрона для даної математичної моделі у модифікації Латгінжера є таким:

$$\hat{H}(z) = -\frac{\hbar^2}{2} \frac{d}{dz} \frac{1}{m(z)} \frac{d}{dz} + \tilde{U}(z). \quad (2.3)$$

У нашій математичній моделі приймається, що електрон має різні ефективні маси у досліджуваній локальній області та суміжному з нею середовищі. Це

приводить до координатної залежності величин ефективної маси $m(z)$ та потенціальної енергії електрона $\tilde{U}(z)$, які в нашому випадку з урахування тих позначень, що були зроблені на рисунку 2.1, є такими:

$$m(z) = \begin{cases} m_0, & 0 \leq z \leq w, \\ m_1, & z < 0, z > w, \end{cases}; \quad \tilde{U}(z) = \begin{cases} U(z), & 0 \leq z \leq w, \\ U_1, & z < 0, \\ U_2, & z > w. \end{cases} \quad (2.4)$$

При урахуванні виразів (2.4), рівняння Шредінгера дещо спрощується, а його запис для всієї локальної області буде таким:

$$\begin{cases} -\frac{\hbar^2}{2m_1} \frac{d^2\Psi(z)}{dz^2} + U_1\Psi(z) = E\Psi(z), & z < 0, \\ -\frac{\hbar^2}{2m_0} \frac{d^2\Psi(z)}{dz^2} + U(z)\Psi(z) = E\Psi(z), & 0 \leq z \leq w, \\ -\frac{\hbar^2}{2m_1} \frac{d^2\Psi(z)}{dz^2} + U_2\Psi(z) = E\Psi(z), & z > w. \end{cases} \quad (2.5)$$

Умова скінченності хвильової функції виражає необхідність того, що електрон має бути локалізованим у досліджуваній області із найбільшою ймовірністю. Безпосередньо для математичної моделі це виражається у умові нормування хвильової функції:

$$\int_{-\infty}^{+\infty} |\Psi(E, z)|^2 dz = 1, \quad (2.6)$$

а також її асимптотикою при значному віддаленні від цієї локальної області:

$$\lim_{z \rightarrow \pm\infty} \Psi(z) = 0. \quad (2.7)$$

Розв'язки рівняння Шредінгера у зовнішньому середовищі по відношенню до локальної області можуть бути отримані у точній аналітичній формі. Вони мають такий вигляд:

$$\begin{aligned}\Psi(z) &= A_1 e^{-\chi_1 z} + B_2 e^{\chi_1 z}, \chi_1 = \frac{\sqrt{2m_1(U_1 - E)}}{\hbar}, z < 0 \\ \Psi(z) &= A_3 e^{-\chi_2 z} + B_3 e^{\chi_2 z}, \chi_2 = \frac{\sqrt{2m_1(U_2 - E)}}{\hbar}, z > w\end{aligned}\quad (2.8)$$

Враховуючи асимптотику хвильової функції, що задається співвідношенням (2.7), ми повинні врахувати те, що хвильова функція має бути скінченною при $z < 0$ і $z > w$. Це приводить до того, що у виразах для хвильової функції (2.8) ми повинні прийняти $A_1 = 0$ і $B_3 = 0$. У результаті цього остаточно для хвильових функцій у зовнішній області маємо:

$$\begin{aligned}\Psi(z) &= B_2 e^{\chi_1 z}, z < 0 \\ \Psi(z) &= A_3 e^{-\chi_2 z}, z > w\end{aligned}\quad (2.9)$$

Для хвильової функції в досліджуваній локальній області у обох математичних моделях, яким відповідають рисунки 2.1-2.2, мають виконуватись граничні умови, які відповідають за скінченність хвильової функції та неперервність потоку її ймовірності. Ці граничні умови такі:

$$\left\{ \begin{aligned} \Psi(z)_{z \rightarrow -0} &= \Psi(z)_{z \rightarrow +0}; \\ \left. \frac{d\Psi(z)}{dz} \right|_{z \rightarrow -0} &= \left. \frac{d\Psi(z)}{dz} \right|_{z \rightarrow +0} \end{aligned} \right\}; \left\{ \begin{aligned} \Psi(z)_{z \rightarrow w-0} &= \Psi(z)_{z \rightarrow w+0}; \\ \left. \frac{d\Psi(z)}{dz} \right|_{z \rightarrow w-0} &= \left. \frac{d\Psi(z)}{dz} \right|_{z \rightarrow w+0} \end{aligned} \right\}. \quad (2.10)$$

Умови (2.10) приводять до дисперсійного рівняння, яке має трансцендентну форму. Його розв'язки визначають спектр електрона, який знаходиться у локальній області на набуває дискретних значень: $E_n, n=1,2,3,\dots$. В результаті величина

енергії активації визначається як різниця між рівнями електронної енергії:

$$\Delta E_{nm} = E_m - E_n, m > n.$$

2.2 Розробка різницевої схеми для математичної моделі із стаціонарним рівнянням Шредінгера та заміна її сітковою задачею. Стійкість різницевої схеми та її модифікації.

У квантовій механіці будь-яке рівняння має бути змінено для відповідності вимогам хвильової теорії, тобто електрон стає хвилею (частинкою-хвилею), тому стан частинки має бути заданий відповідними граничними умовами. Таким чином, нове представлення рівняння Шредінгера всерівно описує своєрідні хвилі, відповідаюче частинці. Рівняння Шредінгера, в одновимірному випадку, разом із довільним потенціалом $V(z)$ має такий вигляд:

$$H(z)\Psi(z) = \left[-\frac{\hbar^2}{2m} \frac{d^2}{dz^2} + V(z) \right] \Psi(z) = E\Psi(z), \quad (2.11)$$

в такому його переставленні нам зручно підійти до побудови різницевої схеми цього рівняння. У співвідношенні (2.11) $V(z)$ - потенціал, в якому виникає хвиля; H - гамільтоніаном задачі, аналогічно (2.3), а $\Psi(z)$ - хвильова функція або ж стан частинки-хвилі. У рівнянні (2.11) енергія E була визначена довільним чином, тобто може бути будь-якою величиною. Як було описано вище, хвиля повинна задовольняти граничним умовам, які визначають спектр значень E . Таке рівняння називається будемо називати власним рівнянням. Таким чином, отримана як розв'язок (2.11) хвильова функція є фактичним власним станом частинки, відповідно тримана енергія є власною енергією. У матричній термінології власний вектор та власне значення часто використовуються як відповідні власному стану та власній енергії відповідно.

З'ясуємо, як граничні умови визначають власні енергії для одновимірного рівняння Шредінгера. Будь-який диференціальний оператор може бути апроксимований скінченною різницевою формою, тому такий метод називається методом скінченних різниць. Наприклад, якщо неперервна змінна x дискретизується в ряд рівномірно розподілених точок x_i , таких що $x_i = x_0 + i\Delta$ ($i = 0, 1, 2, \dots, N$; Δ – константа – крок сітки), тоді друга похідна апроксимується таким чином:

$$\frac{d^2}{dx^2} f(x_i) = f''(x) \approx \frac{f(x_{i-1}) - 2f(x_i) + f(x_{i+1}))}{\Delta^2}. \quad (2.12)$$

Тоді із рівняння (2.11) при урахуванні (2.12) легко отримується набір одновимірних рівнянь, що визначаються вузлами сітки x_i :

$$\begin{aligned} a(\Psi_2 - 2\Psi_1 + \Psi_0) + V_1\Psi_1 &= E\Psi_1; \\ a(\Psi_3 - 2\Psi_2 + \Psi_1) + V_2\Psi_2 &= E\Psi_2; \\ \dots \\ a(\Psi_N - 2\Psi_{N-1} + \Psi_{N-2}) + V_{N-1}\Psi_{N-1} &= E\Psi_{N-1}, \end{aligned} \quad (2.13)$$

де $V_i = V(x_i)$, $\Psi_i = \Psi(x_i)$, а $a = -\frac{\hbar^2}{2m\Delta^2}$.

Для кожної математичної моделі локальної зони дуже важливими є граничні умови. У найпростішій математичній моделі ми припустимо, що хвильова функція набуває значення нуль як у самій лівій, так і в правій координаті локальної області, тобто $\Psi_0 = \Psi_N$. Тоді усі ці рівняння (2.13) безпосередньо можуть бути подані в матричній формі як далі:

$$\begin{pmatrix} -2a+V_1 & a & 0 & 0 & 0 & \dots & 0 \\ a & -2a+V_2 & a & 0 & 0 & \dots & 0 \\ 0 & a & -2a+V_3 & a & 0 & \dots & 0 \\ 0 & 0 & a & -2a+V_4 & a & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & -2a+V_{N-1} \end{pmatrix} \times \\
\times \begin{pmatrix} \Psi_1 \\ \Psi_2 \\ \Psi_3 \\ \Psi_4 \\ \vdots \\ \Psi_{N-1} \end{pmatrix} = E \begin{pmatrix} \Psi_1 \\ \Psi_2 \\ \Psi_3 \\ \Psi_4 \\ \vdots \\ \Psi_{N-1} \end{pmatrix} \quad (2.14)$$

Таким чином скінченно-різницева схема (2.14). для математичної моделі рівняння Шредінгера є еквівалентною проблемі відшукування власних значень матриці

$$A_{nm} = \begin{pmatrix} -2a+V_1 & a & 0 & 0 & 0 & \dots & 0 \\ a & -2a+V_2 & a & 0 & 0 & \dots & 0 \\ 0 & a & -2a+V_3 & a & 0 & \dots & 0 \\ 0 & 0 & a & -2a+V_4 & a & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & -2a+V_{N-1} \end{pmatrix}, \text{ що однозначно}$$

визначає спектр E_n і хвильову функцію кожного стану Ψ_n .

Розглянемо тепер більш реалістичну і складнішу математичну модель, що відповідає локальній області на рисунку 2.1. Для такої моделі врахування граничних умов (2.10) є обов'язковим. Використовуючи скінченно-різницеве наближення для першої похідної у вигляді:

$$\frac{d}{dx} f(x_i) = f'(x) \approx \frac{f(x_i) - f(x_{i-1}))}{\Delta} \quad (2.15)$$

будемо мати:

$$\begin{cases} \Psi_0 = \Psi_1; \\ \frac{\Psi_1 - \Psi_0}{\Delta} = \frac{\Psi_2 - \Psi_1}{\Delta}; \end{cases} \begin{cases} \Psi_{N-1} = \Psi_N; \\ \frac{\Psi_{N-1} - \Psi_{N-2}}{\Delta} = \frac{\Psi_N - \Psi_{N-1}}{\Delta}, \end{cases} \quad (2.16)$$

або те саме, що:

$$\begin{cases} \Psi_0 = \Psi_1; \\ \Psi_0 - 2\Psi_1 + \Psi_2 = 0; \end{cases} \begin{cases} \Psi_{N-1} = \Psi_N; \\ \Psi_{N-2} - 2\Psi_{N-1} + \Psi_N = 0. \end{cases} \quad (2.17)$$

В результаті різницева схема (2.14) модифікується та набуває такого вигляду:

$$\begin{pmatrix} V_1 & a & 0 & 0 & 0 & \dots & 0 \\ a & -2a+V_2 & a & 0 & 0 & \dots & 0 \\ 0 & a & -2a+V_3 & a & 0 & \dots & 0 \\ 0 & 0 & a & -2a+V_4 & a & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & V_{N-1} \end{pmatrix} \times \begin{pmatrix} \Psi_1 \\ \Psi_2 \\ \Psi_3 \\ \Psi_4 \\ \vdots \\ \Psi_{N-1} \end{pmatrix} = E \begin{pmatrix} \Psi_1 \\ \Psi_2 \\ \Psi_3 \\ \Psi_4 \\ \vdots \\ \Psi_{N-1} \end{pmatrix}. \quad (2.18)$$

Алгоритмічний підхід до розв'язання різницевої схеми (2.18) аналогічний як і до різницевої схеми (2.14), він полягає у відшукуванні власних чисел та власних

функцій матриці $A_{nm} = \begin{pmatrix} V_1 & a & 0 & 0 & 0 & \dots & 0 \\ a & -2a+V_2 & a & 0 & 0 & \dots & 0 \\ 0 & a & -2a+V_3 & a & 0 & \dots & 0 \\ 0 & 0 & a & -2a+V_4 & a & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & V_{N-1} \end{pmatrix}.$

2.3 Типові локальні потенціали, що використовуються у якості математичних моделей локалізованих квантових ям, їх візуалізація.

2.3.1 Гармонічний осцилятор

Гармонічним осцилятором називається механічна або квантово-механічна система, яка характеризується потенціальною енергією, що може бути подана у вигляді:

$$U(x) = \frac{1}{2} kx^2. \quad (2.19)$$

У залежності від розміру області локалізації частинки у системі з гармонічним осцилятором будуть проявлятися різного типу ефекти, пов'язані з випромінюванням та генерацією електромагнітного поля. У дипломній роботі розглядаються лише об'єкти, що відносяться до мікро- та наносвіту, тому визначальною є координатна залежність потенціалу в розглядуваних мезоскопічних системах.

Для гармонічного осцилятора в нашому випадку, тобто для одиначної квантової ями, модифікований конфайнмент потенціальної схеми отримується в результаті виконання програмної складової, що входить до розробленої інформаційної системи. Приклад реалізації програмного коду цієї складової подано на рисунку 2.3.

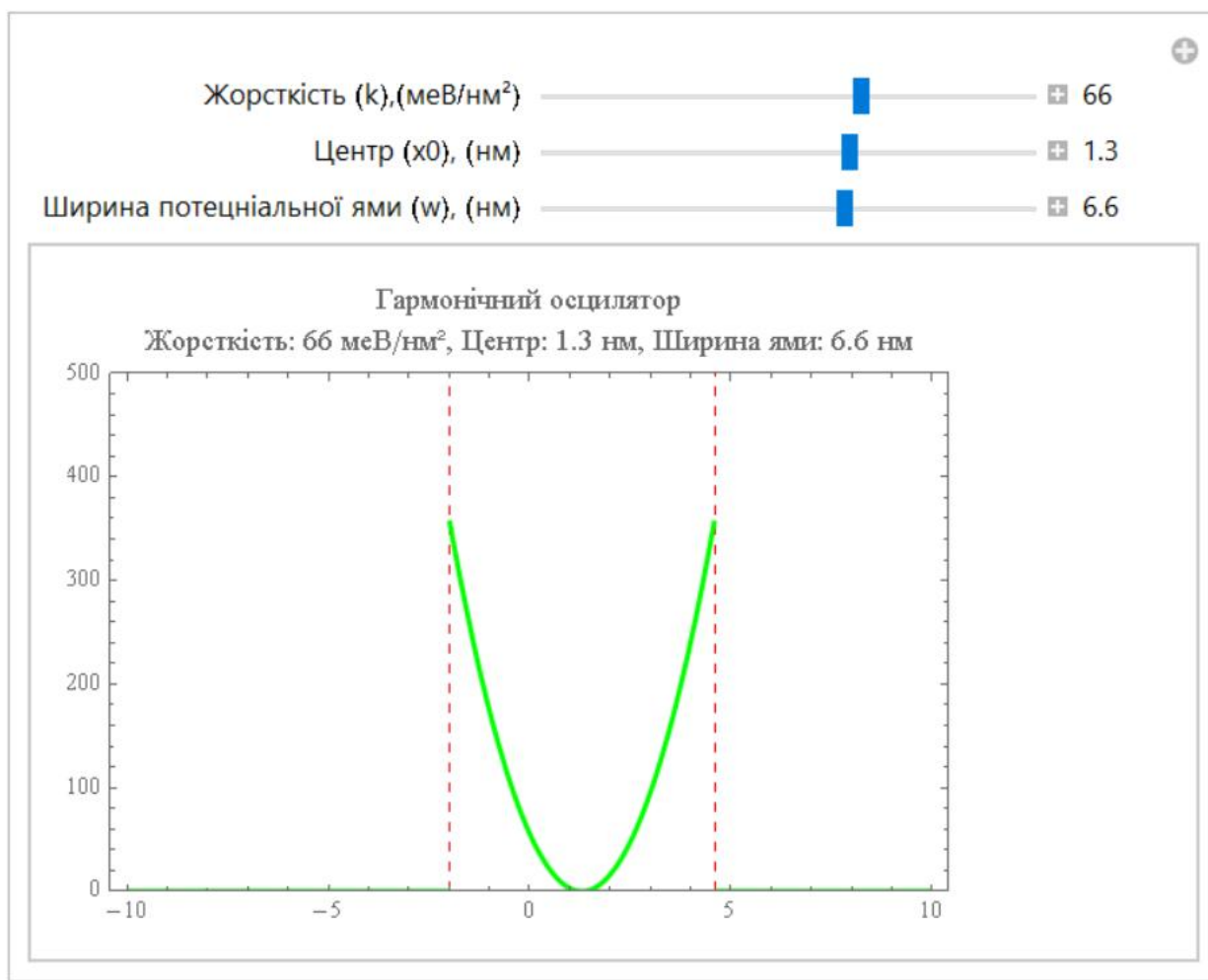


Рисунок 2.3 – Графічна реалізація гармонічного осцилятора у квантовій ямі з довільними розмірами, які можна змінювати та координатами центру x_0

Таким чином складова розробленої нами програмної системи дозволяє ефективно здійснювати алгоритмізацію математичної моделі гармонічного осцилятора шляхом зміни його вхідних параметрів за допомогою слайдерів (див. рис. 2.3). У результаті користувач програмної системи може змінювати ефективну жорсткість гармонічного осцилятора, а також змінювати геометрію квантової ями (її фактичну ширину), в межах якої осцилятор чинить вплив на локалізовані в ній частинки, а також координати центра потенціальної ями x_0 .

2.3.2 Ангармонічний осцилятор

Аналогічним чином виконувалась програмна реалізація математичних моделей для усіх вищезгаданих ефективних потенціалів. Ключовим пріоритетом ставилась необхідність забезпечення реалізації максимально можливої кількості вхідних параметрів. Зокрема, для ангармонічного осцилятора, потенціальна енергія якого задається виразом:

$$U(x) = \frac{1}{2}kx^2 + \alpha x^4, \quad (2.20)$$

вхідними параметрами є ефективна жорсткість k та параметр, що характеризує ангармонічність α . Реалізація програмної компоненти для ангармонічного осцилятора є такою, як показано на рисунку 2.4.

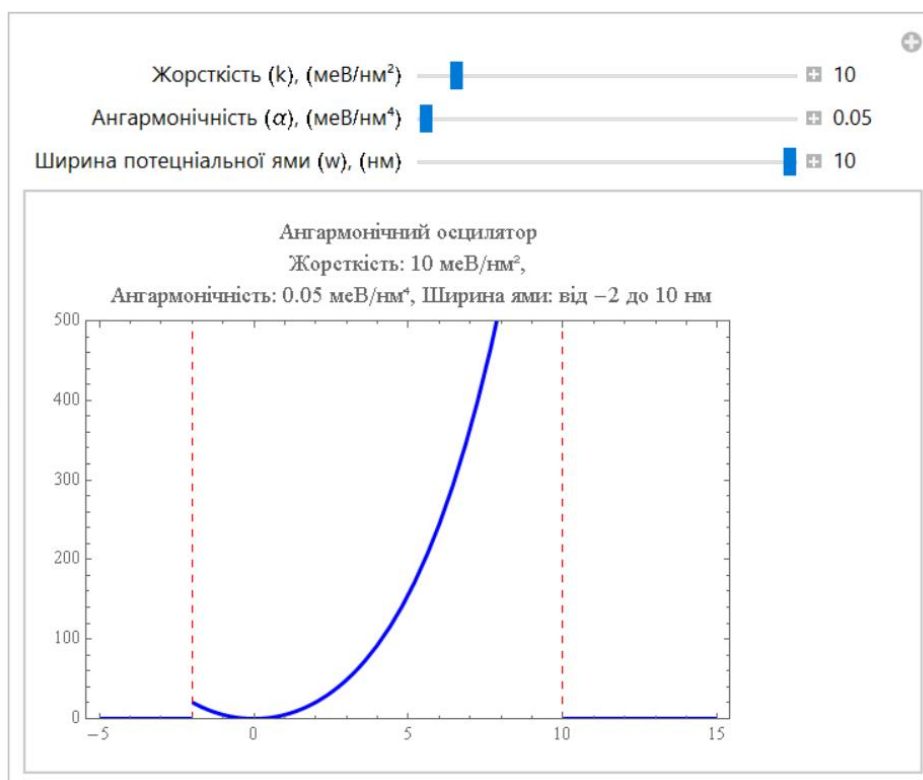


Рисунок 2.4 – Графічна реалізація ангармонічного осцилятора з довільними параметрами ефективної жорсткості та ангармонічності

Аналогічним чином як у попередньому випадку, складова розробленої нами програмної системи дозволяє ефективно здійснювати алгоритмізацію математичної моделі ангармонічного осцилятора шляхом зміни його вхідних параметрів за допомогою слайдерів (див. рис. 2.4). При цьому користувач програмної системи може змінювати ефективну жорсткість гармонічного осцилятора, розміри квантової ями, та параметр α , що характеризує ангармонічність осцилятора.

2.3.3 Потенціал Пешля-Теллера

Потенціал Пешля-Теллера може бути описаний формулою, поданою нижче:

$$U(x) = -U_0 \operatorname{sech}^2\left(\frac{x}{a}\right). \quad (2.21)$$

Вхідними параметрами у такій математичній моделі є ширина потенціальної ями a та її глибина U_0 . Реалізацію програмної компоненти для візуалізації зображення поведінки математичної моделі на основі потенціалу Пешля-Теллера продемонстровано на рисунку 2.5.

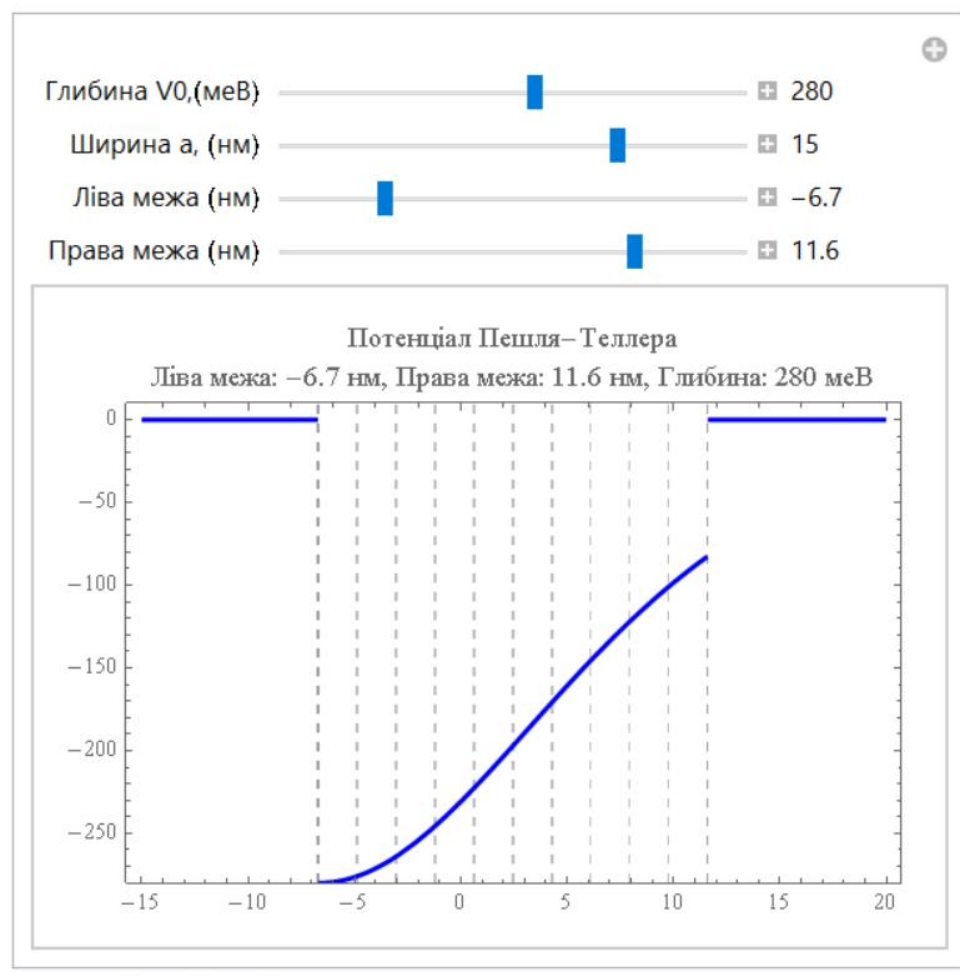


Рисунок 2.5 – Графічна реалізація потенціалу Пешля-Теллера у квантовій ямі з довільними розмірами

У реалізованій компоненті програмної системи, математична модель потенціалу Пешля-Теллера дозволяє в загальному випадку за допомогою слайдерів міняти глибину потенціальної ями, ширину цієї потенціальної ями, а також положення її лівої та правої меж.

2.3.4 Модифікований потенціал Пешля-Теллера

Аналогічним чином виконалась програмна реалізація математичної моделі модифікованого потенціалу Пешля-Теллера, який задається наступною формулою:

$$U(x) = \frac{-U_0}{\cosh^2\left(\frac{x}{a}\right)}. \quad (2.22)$$

Як і у випадку потенціалу Пешля-Теллера, модифікований потенціал має вхідними параметрами ширину потенціальної ями a та її глибину U_0 . Реалізацію програмної компоненти для математичної моделі модифікованого потенціалу Пешля-Теллера продемонстровано на рисунку 2.6.

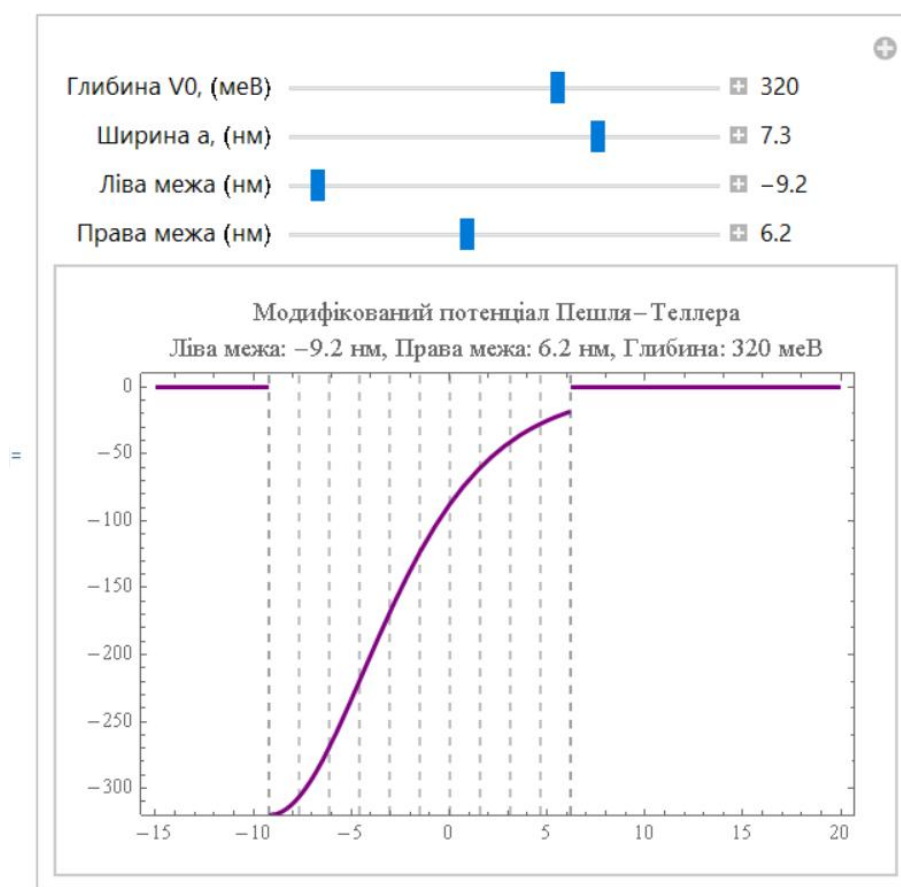


Рисунок 2.6 – Графічна реалізація модифікованого потенціалу Пешля-Теллера у квантовій ямі з довільними розмірами

Аналогічно вище розглянутим випадкам, окрім можливості зміни глибини та ширини потенціальної ями за допомогою слайдерів, в загальному випадку для математичної моделі передбачено можливість міняти положення її лівої та правої меж, що дуже важливо для моделювання прикладних застосувань.

2.3.5 Потенціал Морзе

Потенціал Морзе – це аналітична формула для виконання розрахунку потенціальної енергії міжатомної взаємодії. На відміну від квантового осцилятора, даний ефективний потенціал одразу враховує ефекти від розриву зв'язків між станами та ангармонічність. Для потенціалу Морзе потенціальна енергія розраховується наступним чином:

$$U(x) = D_0(1 - e^{-\frac{x}{a}})^2, \quad (2.23)$$

для такої математичної моделі вхідними параметрами є ширина потенціальної ями a та її глибина D_0 . Приклад програмної реалізації математичної моделі із візуалізації потенціалу Морзе, як складової розроблено програмної системи подано на рисунку 2.7.

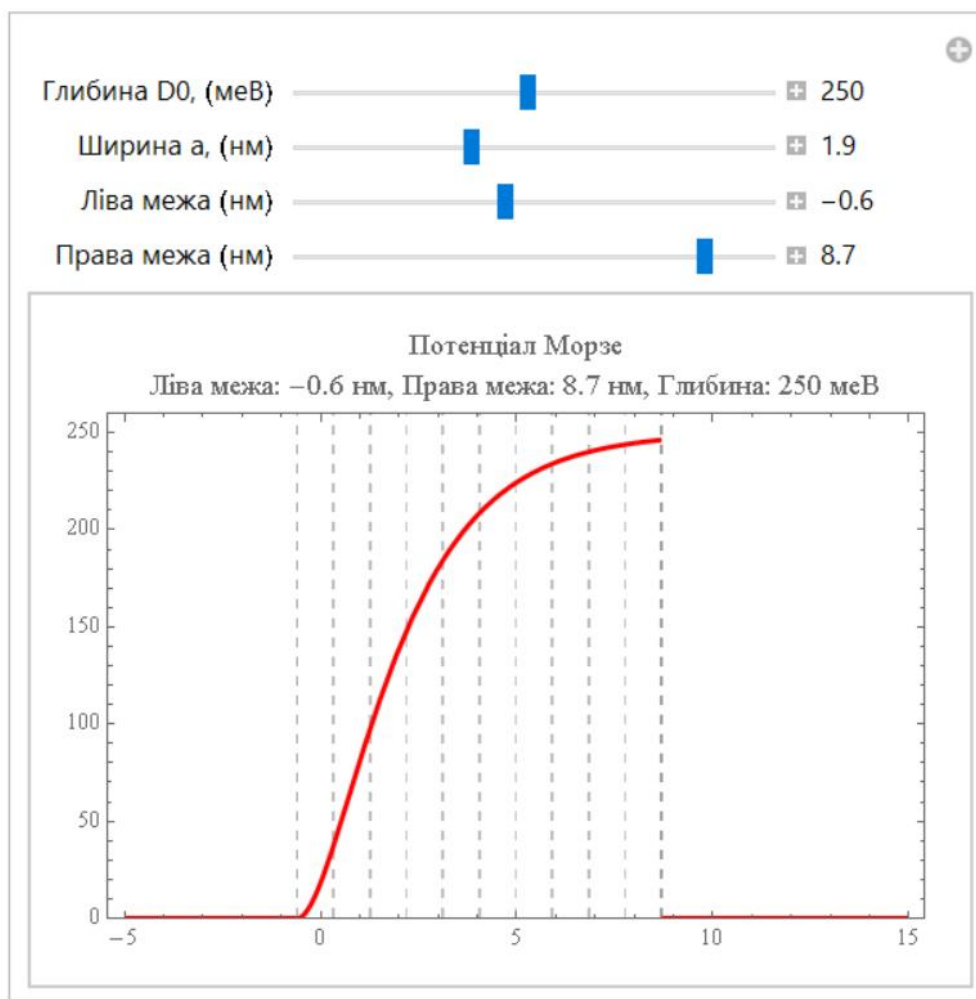


Рисунок 2.7 – Графічна реалізація потенціалу Морзе
у квантовій ямі з довільними розмірами

У даній складовій програмної систем ми також реалізували можливості зміни глибини та ширини потенціальної ями за допомогою слайдерів, а також забезпечили можливість міняти положення лівої та правої меж квантової ями.

2.3.6 Потенціал Леннарда-Джонса

На завершення роботи над цією частиною програмної системи, ми розробили її компоненту, що забезпечує візуалізацію математичної моделі потенціалу Леннарда-Джонса. Потенціал Леннарда-Джонса – це ефективний

потенціал, який описує залежність енергії взаємодії двох нейтральних атомів від відстані між ними. Даний потенціал задається наступною формулою:

$$U(x) = 4\varepsilon\left(\left(\frac{\sigma}{x}\right)^{12} - \left(\frac{\sigma}{x}\right)^6\right), \quad (2.24)$$

вхідними параметрами є енергія взаємодії ε та характерна відстань σ . Реалізація програмної компоненти для потенціалу Леннарда-Джонса є такою, як показано на рисунку 2.8.

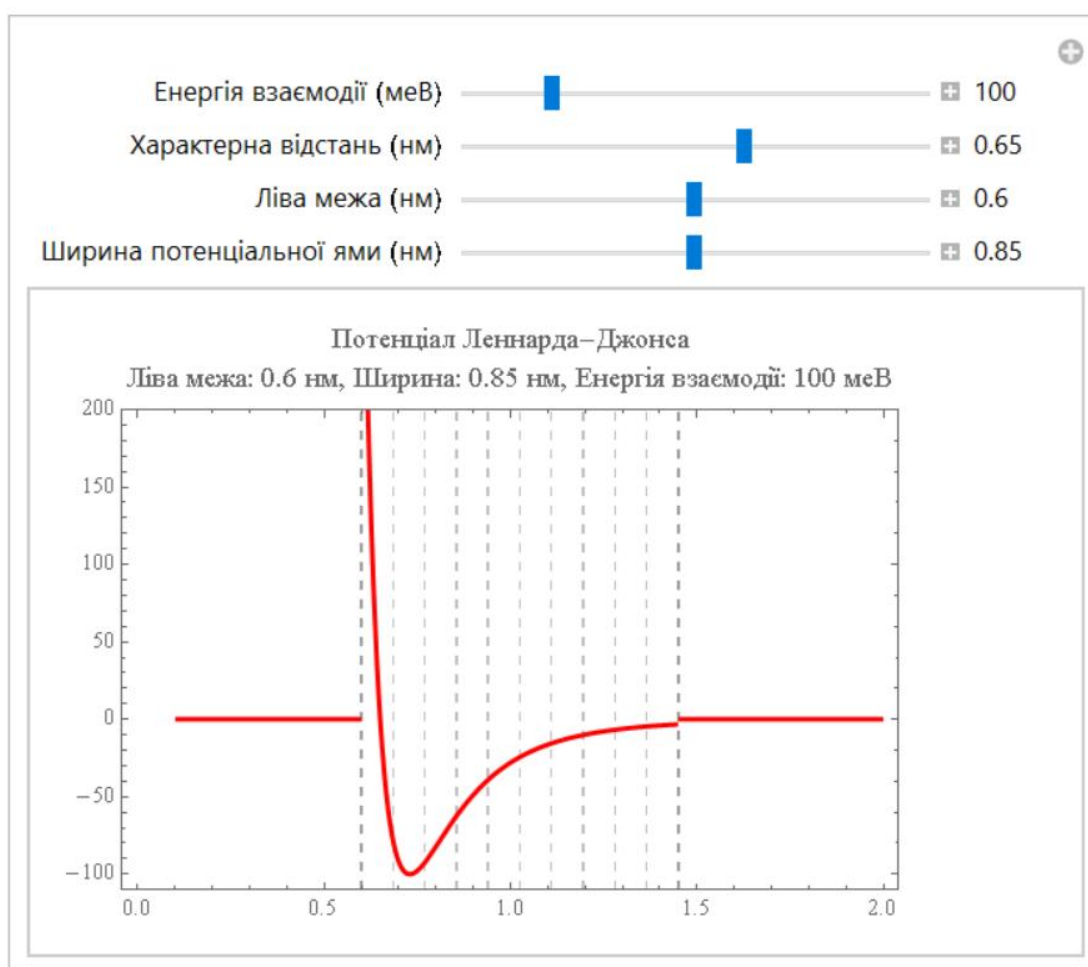


Рисунок 2.8 – Графічна реалізація потенціалу Леннарда-Джонса у квантовій ямі з довільними розмірами

Реалізована програмна компонента математичної моделі потенціалу Леннарда-Джонса дозволяє не тільки змінювати розміри потенціальної ями та її

розміщення, але й вже наочно встановлювати при яких параметрах енергії взаємодії та характеристичного параметра ця взаємодія має характер притягання чи відштовхування.

РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ДЛЯ МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ЕНЕРГІЙ АКТИВАЦІЇ З РІЗНИМИ ПОТЕНЦІАЛАМИ.

3.1 Методологія розробки

Для розробки спеціалізованого програмного забезпечення було обрано інкрементивну методологію. Інкрементивна методологія розробки програмного забезпечення є одним із найбільш ефективних підходів до створення складних систем, що базується на поступовому нарощуванні функціональності через послідовні етапи розробки. На відміну від монолітних підходів, де система створюється одразу в повному обсязі, інкрементивна методологія передбачає розбиття проєкту на менші, керовані частини, кожна з яких додає нову цінність до кінцевого продукту.

Основна ідея інкрементивної розробки полягає в тому, що складні системи краще створювати поетапно, де кожен етап представляє собою функціонально завершений інкремент. Цей підхід ґрунтується на принципі "divide and conquer" – розділай і володарюй, що дозволяє розробникам зосередитися на вирішенні конкретних завдань, не втрачаючи загального бачення проєкту.

Конкретний приклад інкрементивної розробки програмного комплексу, який створювався для роботи з потенціалами то даний проєкт, що було розбито на чотири основні інкременти, кожен з яких мав свою специфіку та цілі.

Перший інкремент: візуалізація потенціалів. Початковий етап було присвячено створенню системи візуалізації потенціалів. Цей вибір був стратегічно виправданим, оскільки візуалізація є тією частиною системи, яка найбільш наочно демонструє можливості програмного комплексу і дозволяє зацікавленим сторонам зрозуміти концепцію проєкту.

На цьому етапі основна увага приділялася розробці алгоритмів рендерингу, вибору оптимальних способів представлення даних і створенню базової архітектури для роботи з графічними об'єктами. Результатом цього інкременту стала система, здатна відображати потенціали у зрозумілому для користувача вигляді, що створило міцний фундамент для подальшої розробки.

Другий інкремент: логіка обчислень. Другий етап був присвячений створенню обчислювального ядра системи. Тут розроблялися алгоритми розрахунку потенціалів, оптимізувалися математичні операції і створювалася архітектура для ефективного управління обчислювальними процесами.

Третій інкремент: графічний інтерфейс користувача. Третій етап був зосереджений на створенні зручного і інтуїтивно зрозумілого графічного інтерфейсу. На цьому етапі система набула вигляду повноцінного програмного продукту, придатного для використання кінцевими користувачами.

Четвертий інкремент: інтерактивне керування параметрами. Завершальний етап був присвячений створенню системи інтерактивного керування параметрами, що перетворило статичну систему на динамічний інструмент для дослідження і аналізу потенціалів. Цей інкремент додав системі гнучкості і розширив можливості її практичного застосування.

Досвід розробки описаного програмного комплексу яскраво демонструє основні переваги інкрементивної методології. По-перше, кожен етап давав робочий результат, який можна було демонструвати і тестувати незалежно від інших частин системи. Це забезпечувало постійний зворотний зв'язок і можливість раннього виявлення проблем.

По-друге, інкрементивний підхід значно знизив ризики проєкту. Кожен етап був відносно невеликим і керованим, що дозволяло швидко реагувати на проблеми і вносити необхідні корективи без кардинальної перебудови всієї системи.

Попри очевидні переваги, інкрементивна розробка також створює певні виклики. Одним з найбільших є необхідність створення гнучкої архітектури, яка дозволяє легко додавати нову функціональність без порушення існуючого коду. Це вимагає від розробників високого рівня планування і передбачення майбутніх потреб системи.

Ще одним викликом є управління залежностями між інкрементами. Важливо забезпечити, щоб зміни в одному інкременті не призводили до проблем

в інших частинах системи. Це потребує ретельного проєктування інтерфейсів і використання принципів слабого зв'язування.

3.2 Аналіз вимог для спеціалізованого програмного забезпечення з моделювання енергій активації.

Для створення ефективного програмного комплексу моделювання енергій активації в цеолітах необхідно чітко визначити вимоги, що гарантують його функціональність, зручність використання, надійність і відповідність очікуванням науковців та інженерів. Вимоги до такого програмного забезпечення формуються на основі глибокого аналізу потреб користувачів та особливостей предметної області – моделювання процесів на молекулярному рівні. Вони охоплюють як функціональні характеристики, що описують основні можливості системи (розрахунок енергій, візуалізація, обробка даних), так і нефункціональні аспекти, пов'язані з продуктивністю, безпекою, масштабованістю та іншими технічними параметрами, які забезпечують високу якість роботи і наукову достовірність результатів.

Функціональні вимоги:

- Обчислення та візуалізація потенціалів: користувач має змогу обирати тип потенціалу: нескінченна яма, скінченна яма, гармонічний осцилятор, Пешля-Теллера та ін. Система включає модуль візуалізації форми вибраного потенціалу.
- Розрахунок спектру енергій та енергії активації: програма обчислює власні значення рівнів енергії для заданого потенціалу. Результати подаються як графіки та надаються числові значення обчислених енергій активації.
- Візуалізація розрахунку: вивід графіків енергетичних рівнів, переходів між ними. Побудова графіків залежностей енергії активації від температури, параметрів ями тощо.

- Розрахунок та візуалізація хвильових функцій: обчислення хвильових функцій (ψ) для кожного енергетичного рівня. Можливість виведення як аналітичного, так і чисельного розв'язку. Побудова графіків хвильових функцій у координатному просторі.

Нефункціональні вимоги:

- Продуктивність: розрахунок спектру та хвильових функцій виконується < 5 секунд при типовому розмірі сітки.
- Надійність: перевірка граничних умов, стабільності чисельного методу. Автоматичне збереження сесій моделювання.
- Розширюваність: модульна архітектура для додавання нових потенціалів, типів задач або методів обчислення.

Визначення функціональних і нефункціональних вимог є основою для успішної реалізації системи, оскільки вони встановлюють чіткі цілі та стандарти для її розробки. Функціональні вимоги забезпечують користувачів необхідним інструментарієм для математичного моделювання енергій активації, а нефункціональні гарантують зручність, масштабованість і надійність системи.

3.3 Проєктування архітектури програмної системи. Компоненти та функціонал програмної системи.

При проєктуванні системи було визначено парадигму розроблюваного комплексу. Парадигма розробленої програмної системи визначена наступним чином:

1. Обчислювально-орієнтована Система розроблена як науково-інженерне середовище, основа якої — чисельне розв'язання диференціальних та інтегральних рівнянь, що описують фізичні явища на нанорівні. Вона орієнтована на точне та стабільне моделювання квантовомеханічних характеристик, таких як:

- спектр енергій електронів у потенційних полях;
- хвильові функції у неоднорідних середовищах;
- розрахунок енергії активації залежно від структурних параметрів.

Перевага: гнучкість математичного апарату та можливість застосування різних чисельних методів (скінченні різниці, власні значення, варіаційні підходи).

2. Параметрично-кероване моделювання. Користувач безпосередньо визначає параметри середовища моделювання:

- геометрію та розмір потенційної ями/моделі цеоліту;
- фізико-хімічні властивості (глибина потенціалу, маса частинки, тощо);
- граничні умови та точність обчислень.

Система надає гнучкий інтерфейс для задання цих параметрів і дозволяє проводити серійні обчислення з автоматичною зміною вхідних умов (сценарії або пакетна обробка).

3. Багатошарова архітектура. Архітектура програмного комплексу побудована за принципом розділення обов'язків між основними логічними рівнями:

- ядро обчислень – модулі, що реалізують чисельні методи та математичну логіку;
- інтерфейс користувача – GUI для введення параметрів, запуску розрахунків, перегляду результатів.
- модулі візуалізації – компоненти побудови графіків, відображення хвильових функцій, потенціалів, енергетичних рівнів тощо.

Перевага: забезпечення розширюваності, повторного використання компонентів і незалежності інтерфейсу від реалізації ядра.

4. Модульність і конфігурованість. Система побудована за принципом модульності, що дозволяє:

- легко підключати нові типи потенціалів, розв'язувачів або алгоритмів;
- реалізовувати окремі функції (наприклад, постпроцесинг або вивід) у вигляді незалежних плагінів;

- налаштовувати сценарії обчислень без перекомпіляції — через конфігураційні файли або графічний редактор параметрів.

Це відкриває можливість гнучкої адаптації системи до різних задач — від освітніх до дослідницьких.

Враховуючи особливості розробленої математичної моделі, її числової реалізації, а також виходячи із необхідності постійного візуального контролю за вхідними та вихідними даними, необхідним є опис варіантів використання, що значно спростить проєктування та розробку системи (таблиця 3.1).

У таблиці 3.1 нижче можна побачити детальний опис варіантів використання системи.

Таблиця 3.1 – варіанти використання системи

<i>Код</i>	<i>Основний актор</i>	<i>Найменування</i>	<i>Формулювання</i>
1	2	3	4
K1	Користувач	Вибір потенціалу	Для отримання результатів моделювання та візуалізації користувач має обрати необхідний ефективний потенціал
K2	Користувач	Введення параметрів задач	Користувач має змогу вводити фізичні та геометричні параметри задач, а також змінювати їх
K3	Користувач	Візуалізація потенціалу	Користувач має змогу візуалізувати набір ефективних потенціалів
K4	Користувач	Розрахунок та візуалізація хвильових функцій	Користувач має змогу розрахувати хвильові функції для заданої квантової ями, а також отримати їх візуалізацію у масштабі, приведеному до енергетичних рівнів

Продовження таблиці 3.1

K5	Користувач	Розрахунок та візуалізацію спектру і енергій активації	Користувач має змогу розраховувати спектр та енергії активації, а також візуалізовувати енергетичні рівні
----	------------	--	---

На основі складеної таблиці було намальовано діаграму варіантів використання. На рисунку 3.1 подано діаграму варіантів використання програмної системи, що дозволяє реалізовувати математичну модель із довільним локальним потенціалом, а також виконувати безпосередні розрахунки та здійснювати візуалізацію параметрів моделі та її характеристик на широкому діапазоні неперервної зміни вхідних даних.

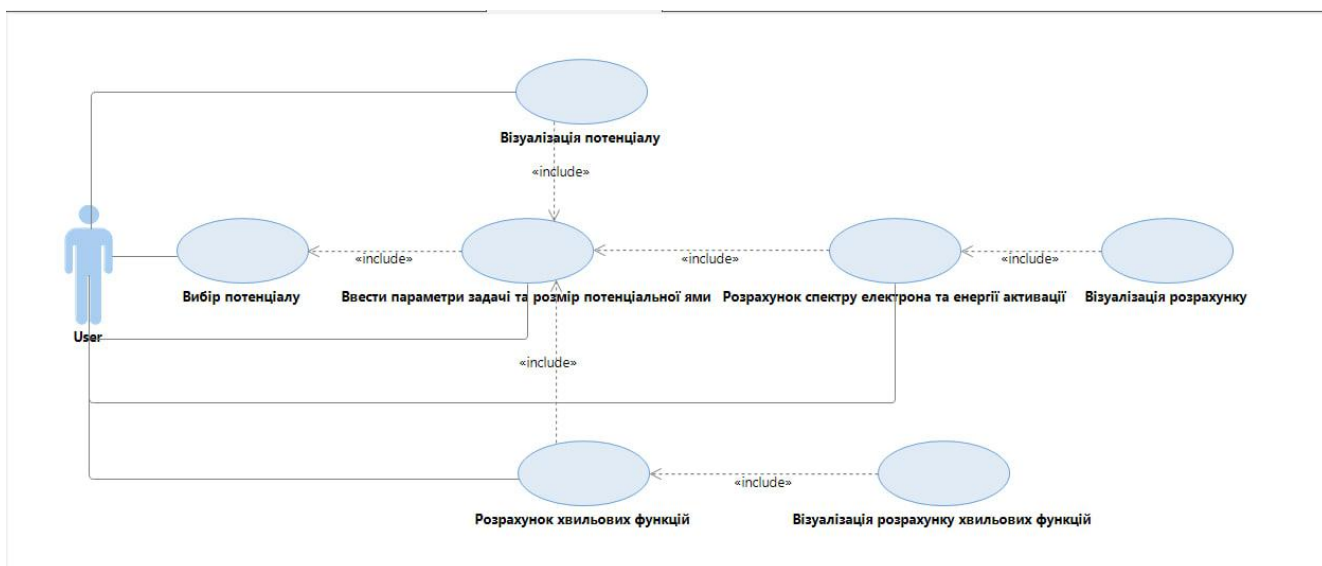


Рисунок 3.1 – Діаграма варіантів використання програмної системи, реалізуючої розроблені математичні моделі

Згідно із даною діаграмою, користувач може виконувати такі дії із програмною системою:

- вибір локального потенціалу;
- введення фізичних параметрів задачі та розміру потенціальної ями;

- візуалізація потенціалу (візуалізація становить окремий програмний блок і може використовуватись користувачами незалежно від необхідності використання безпосередніх розрахунків на основі розвиненої математичної моделі);
- розрахунок хвильових функцій електронних станів у системі (у даному підході передбачена можливість візуалізації квадратів хвильових функцій, що дозволяє користувачу слідкувати за локалізацією електрона у досліджуваній області системи);
- візуалізація хвильових функцій електронних станів у системі;
- розрахунок спектру електрона та енергій активації;
- візуалізація спектру електрона та енергій активації.

Для реалізації визначеного функціоналу необхідно розробити гнучку модульну архітектуру, яка буде забезпечувати всі описані функціональні та нефункціональні вимоги, а також реалізовувати визначені варіанти використання. На рисунку 3.2 можна побачити діаграму пакетів, що показує взаємозв'язки між модулями системи.

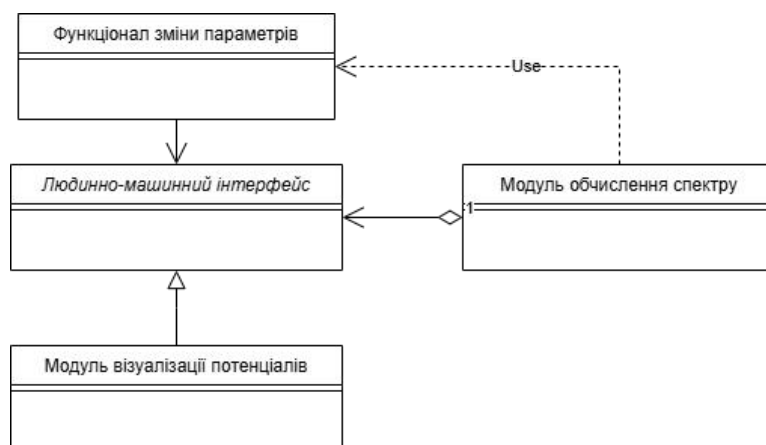


Рисунок 3.2 – Діаграма пакетів розроблюваної системи

Детальніше опишемо кожен з модулів:

1. Функціонал зміни параметрів

Цей пакет відповідає за гнучке конфігурування задач моделювання. Реалізує можливість змінювати:

- параметри потенціалів (глибина, ширина, форма);
- фізичні константи (маса частинки, $\hbar$ тощо);
- умови розрахунку (точність, сітка, граничні умови).

Взаємодіє з іншими модулями як джерело вхідних даних.

2. Людинно-машинний інтерфейс

Призначений для взаємодії користувача з системою. Дозволяє:

- обирати тип потенціалу;
- вводити параметри;
- запускати розрахунки;
- переглядати результати.

Отримує дані з пакету «Функціонал зміни параметрів» та «Модуль обчислення спектру» і візуалізує їх.

3. Модуль обчислення спектру

Основне ядро системи, що виконує чисельні розрахунки:

- знаходження власних значень (спектру) Гамільтоніана;
- обчислення енергій активації;
- побудова хвильових функцій.

Отримує параметри з функціоналу зміни параметрів.

4. Модуль візуалізації потенціалів

Відповідає за побудову графічних зображень ефективних потенціалів, є окремою складовою графічного інтерфейсу. Візуально підтверджує коректність введених параметрів і результатів розрахунку.

3.4 Розробка програмного комплексу для математичного моделювання енергій активації.

Програмна реалізація математичних моделей ефективних потенціалів як спектральних проблем для стаціонарного рівняння Шредінгера виконувалась як побудова (2.14)-(2.18) сіткових різницевих схем [21]. Враховувались усі розглянуті

та визначені вимоги, система конструювалась на основі раніше визначеної архітектури.

Ми починали з реалізації найпростіших складових цього блоку програмної системи, а саме для моделі потенціалу із безкінечно високими стінками, в межах потенціальної ями для якого $U(z)=0$. Основними параметрами, які можна міняти в такій моделі є ефективна маса електрона m та ширина потенціальної ями L . Також ми брали кількість розбиттів одномірної сітки рівні $N=500$, що відповідає матриці (2.14) і (2.15) розміру 500×500 , при цьому забезпечується точність 10^{-6} . Реалізація цієї складової програмної системи подана на рисунку 3.3.

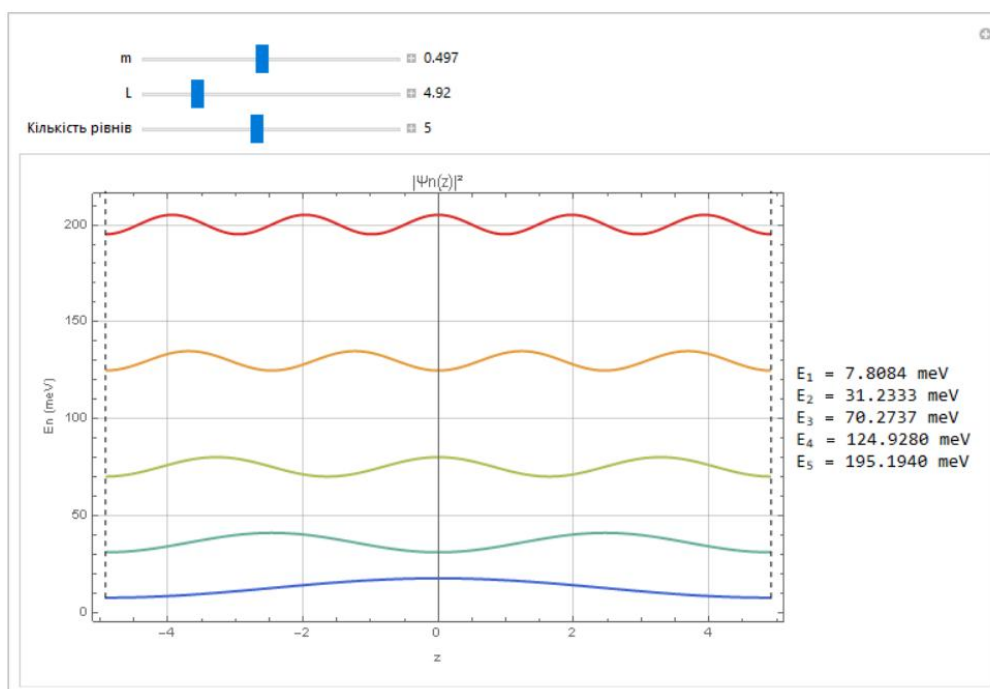


Рисунок 3.3 – Моделювання спектру та квадратів модулів хвильових функцій у потенціальній ямі з безмежно високими стінками

Безпосередня програмна реалізація сіткової схеми для такої математичної моделі подана на рисунку 3.4.

```

Manipulate[Module[{ab = 0.0529, Ry = 13605, U0 = 0, hbar2Div2m, dx, A, n = 500, Evals, wfn, sortedIndices, x},
|маніпулювати| |програмний модуль|
  hbar2Div2m = ((Sqrt[2] * ab * Sqrt[Ry])^2) / (2 * m);
  |квдратний ко...| |квдратний корінь|
  (*Крок*) dx = 2 * L / (n + 1);
  (*Матриця*) A = Table[0, {n}, {n}];
  |таблиця значень|
  Do[If[i > 1, A[[i, i - 1]] = -hbar2Div2m / dx^2];
  |...| |умовний оператор|
  If[i < n, A[[i, i + 1]] = -hbar2Div2m / dx^2];
  |умовний оператор|
  A[[i, i]] = 2 * hbar2Div2m / dx^2; {i, 1, n}];
  (*Власні значення та вектори*) {Evals, wfn} = Eigensystem[A];
  |власні числа і вектори|
  sortedIndices = Ordering[Evals];
  |список рангових індексів|
  Evals = Evals[[sortedIndices]];
  wfn = wfn[[All, sortedIndices]];
  |всє|
  (*Координата*) x = Table[-L + i * dx, {i, 1, n}];
  |таблиця значень|
  (*Побудова графіків*) Grid[{(Show[Table[Module[{PsiSquared},
  |таблиця| |пос...| |табл...| |програмний модуль|
    PsiSquared = 10 * wfn[[All, i]]^2 / Max[wfn[[All, i]]^2] + Evals[[i]];
    |всє| |максимум| |всє|
    ListLinePlot[Transpose[{x, PsiSquared}],
    |лінійний графік...| |транспозиція|
    PlotStyle -> {Thick, ColorData["Rainbow"][[i / 5]], PlotRange -> All}], {i, 1, levels}],
    |стиль графіка| |жирни...| |дані про кольори| |діапазон зна...| |всє|
    PlotRange -> All, Frame -> True, FrameLabel -> {"z", "En (meV)"}, PlotLabel -> "|Ψn(z)|^2", GridLines -> Automatic, ImageSize -> Large,
    |діапазон зна...| |всє| |рамка| |істина| |позначка для рамки| |позначка графіка| |ліній координ...| |автоматич...| |розмір зобр...| |крупний|
    Epilog -> {Dashed, Line[{{-L, Min[Evals]}, {-L, Max[Evals]}}, Dashed, Line[{{L, Min[Evals]}, {L, Max[Evals]}}, {(*Список енергій*)
    |епілог| |штрих...| |ламана) лінія| |мінімум| |максимум| |штрих...| |ламана) лі...| |мінімум| |максимум|
    Column[Table[Style[Row["E", Subscript["", i], " = ", NumberForm[Evals[[i]], {6, 4}]], FontSize -> 14], {i, 1, levels}]]}],
    |колонка| |табл...| |стиль| |ряд| |з нижнім індексом| |числова форма| |розмір шрифту|
    {{m, 0.186}, 0.05, 1, Appearance -> "Labeled"}, {{L, 10.0}, 1, 20, Appearance -> "Labeled"},
    |зовнішній вигляд| |зовнішній вигляд|
    {{levels, 5, "Кількість рівнів"}, 1, 10, 1, Appearance -> "Labeled"}, ControlPlacement -> Top}
    |зовнішній вигляд| |розташування елемен...| |вгору|

```

Рисунок 3.4 – Загальна структура алгоритму, реалізуючого математичну модель спектру та хвильових функції у вигляді програмного модуля

Можливість неперервної зміни ефективної маси електрона m та ширини потенціальної ями L виконано за допомогою слайдерів, номер електронного рівня також змінювався за допомогою спайдера із дискретним кроком. Для зручності квадрати модулів хвильових функцій $|\Psi_n(E_n, z)|^2$ приведено у шкалі енергій E_n , а самі енергії виводяться в бічному меню.

Далі виконувалась реалізації математичних моделей для різних ефективних потенціалів у потенціальній ямі із потенціальними стінками скінченної висоти U_0 . Складова програмної системи, що створена на основі моделі квантового осцилятора подана на рисунку 3.5. Окрім можливості змінювати ефективну масу електрона m та ширину потенціальної ями L виконано за допомогою слайдерів, кількість електронних рівнів із дискретним кроком, також реалізовано можливість змінювати висоту потенціального бар'єра U_0 та ефективну жорсткість a , тобто в

даній реалізованій математичній моделі ми можемо інтерактивно змінювати всі її параметри.

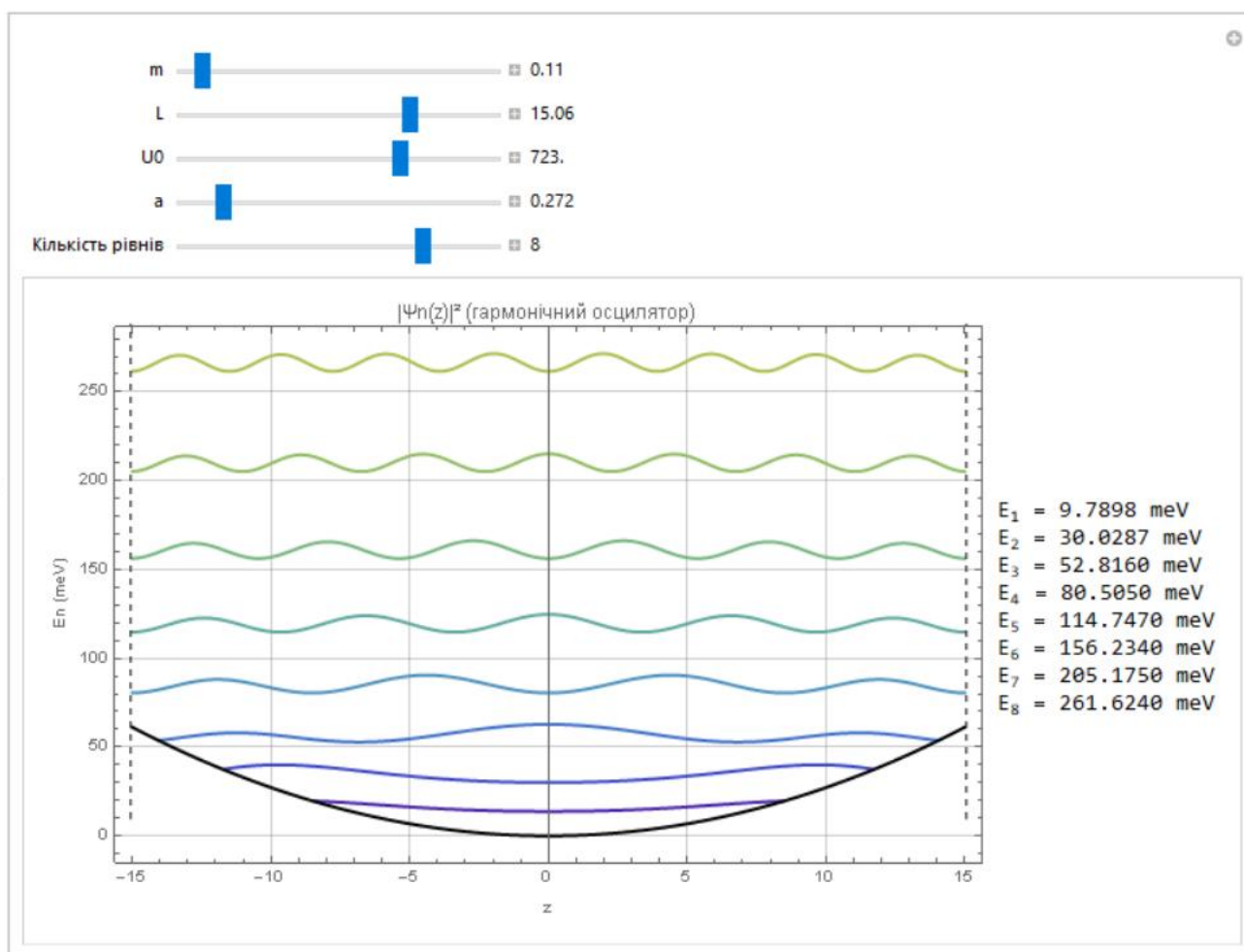


Рисунок 3.5 – Моделювання спектру та квадратів модулів хвильових функцій у потенціальній ямі з скінченними стінками для квантового гармонічного осцилятора

Схожа ситуація має місце для ангармонічного осцилятора, програмна реалізація математичної моделі якого подана на рисунку 3.6, тут на відміну від розглянутої вище моделі гармонічного осцилятора також реалізовано можливість змінювати параметр ангармонічності b .

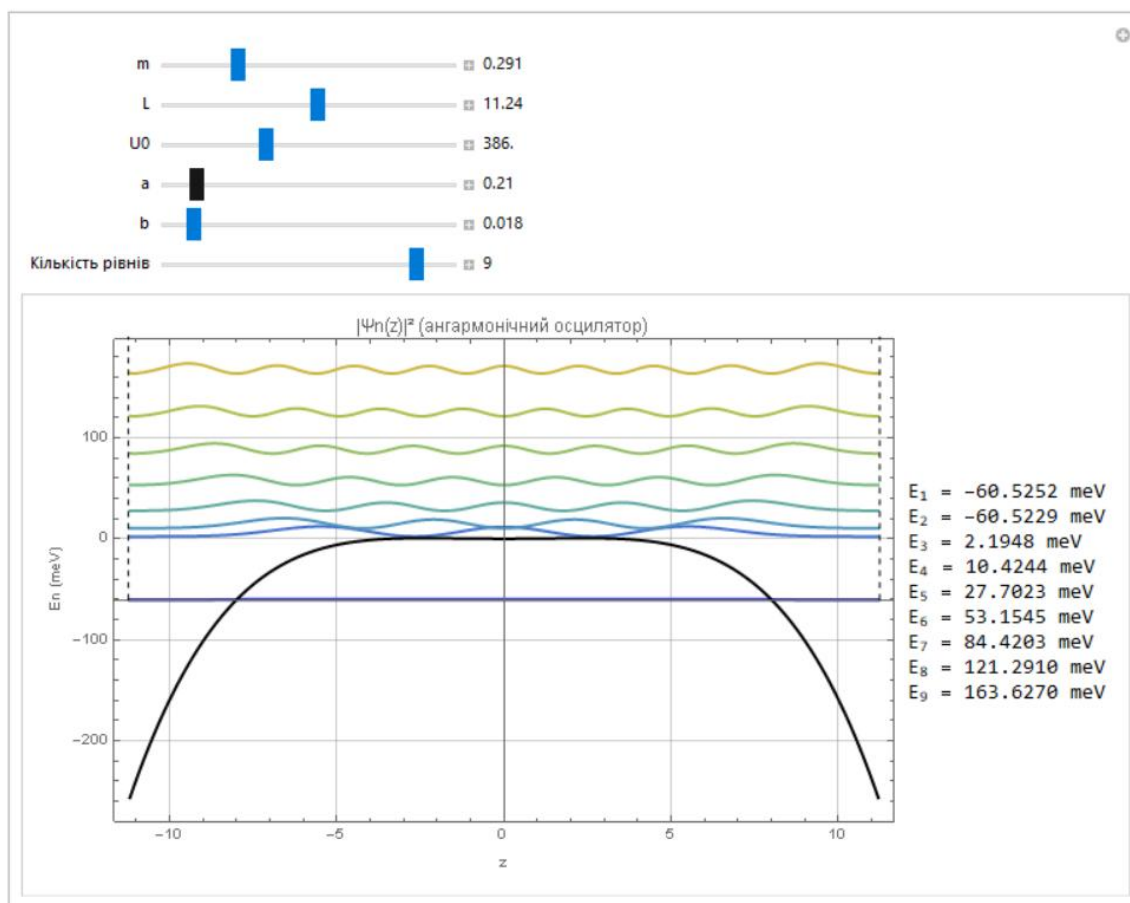


Рисунок 3.6 – Моделювання спектру та квадратів модулів хвильових функцій у потенціальній ямі з скінченними стінками для квантового ангармонічного осцилятора

Далі виконувалась програмна реалізація потенціалу Пешля-Теллера:

$$V(x) = -\frac{V_0}{\cosh^2(ax)} .$$

Така математична модель додатково характеризується

величиною глибини V_0 та узагальненою величиною ширини квантової ями a .

Результати такого математичного моделювання у вигляді програмного блоку подано на рисунку 3.7. На відміну від математичних моделей гармонічного та ангармонічного осцилятора, які мають місце в моделях молекулярних спектрів, потенціал Пешля-Теллера гарно підходить для апроксимації потенціальних пасток для летких вуглеводнів.

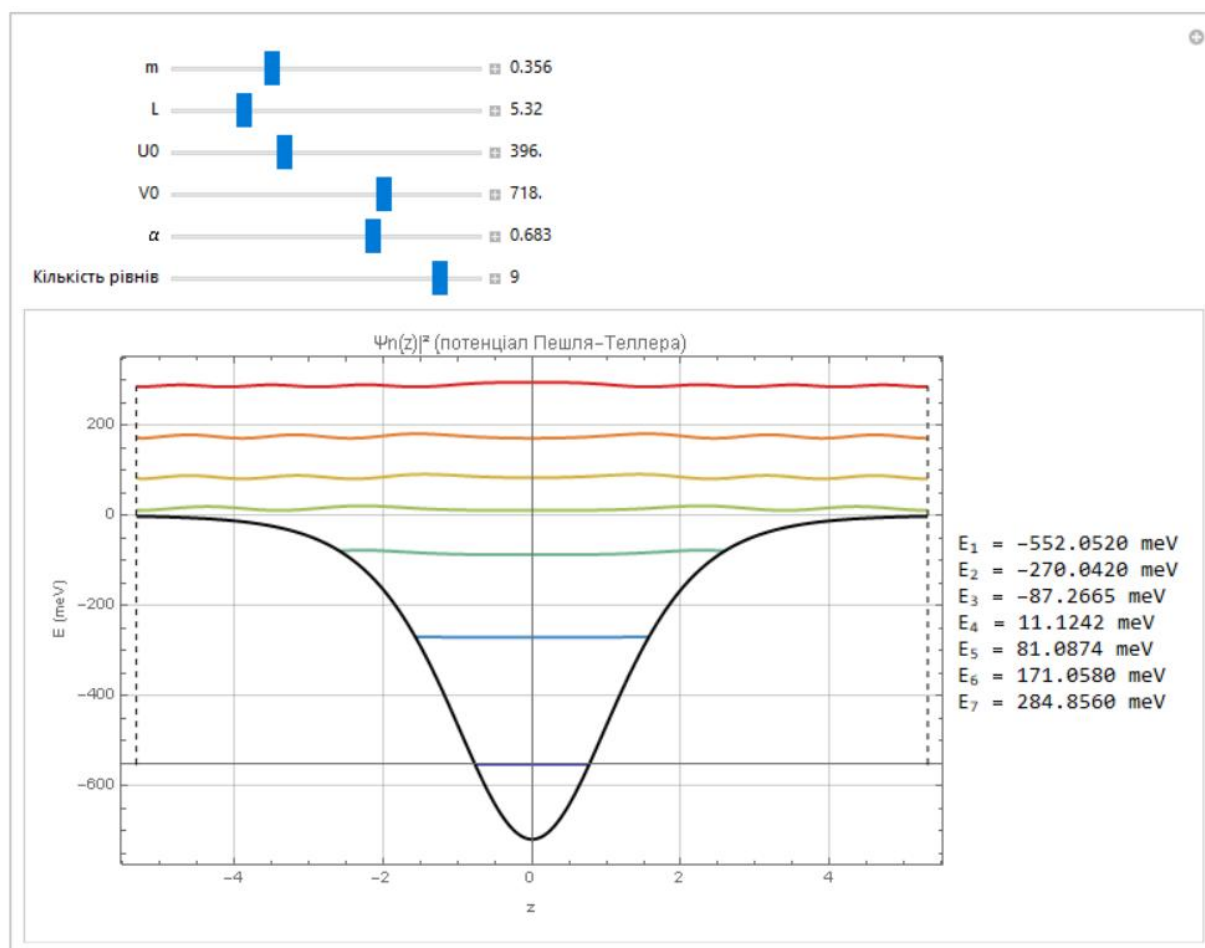


Рисунок 3.7 – Моделювання спектру та квадратів модулів хвильових функцій у потенціальній ямі з скінченними стінками для потенціалу Пешля-Теллера

Одним з альтернативних варіантів для математичних моделей потенціальних пасток є модифікований потенціал Пешля-Теллера:

$$V(x) = -\frac{V_0}{\cosh^2(ax)} + V_1 \tanh^2(ax) .$$

Він має більшу гнучкість та надійність для

моделювання, оскільки в такій моделі міститься додатковий параметр “підгонки”. Результати розробленого програмного блоку такої математичної моделі при наявності слайдерів зміни для всіх параметрів подано на рисунку 3.8.

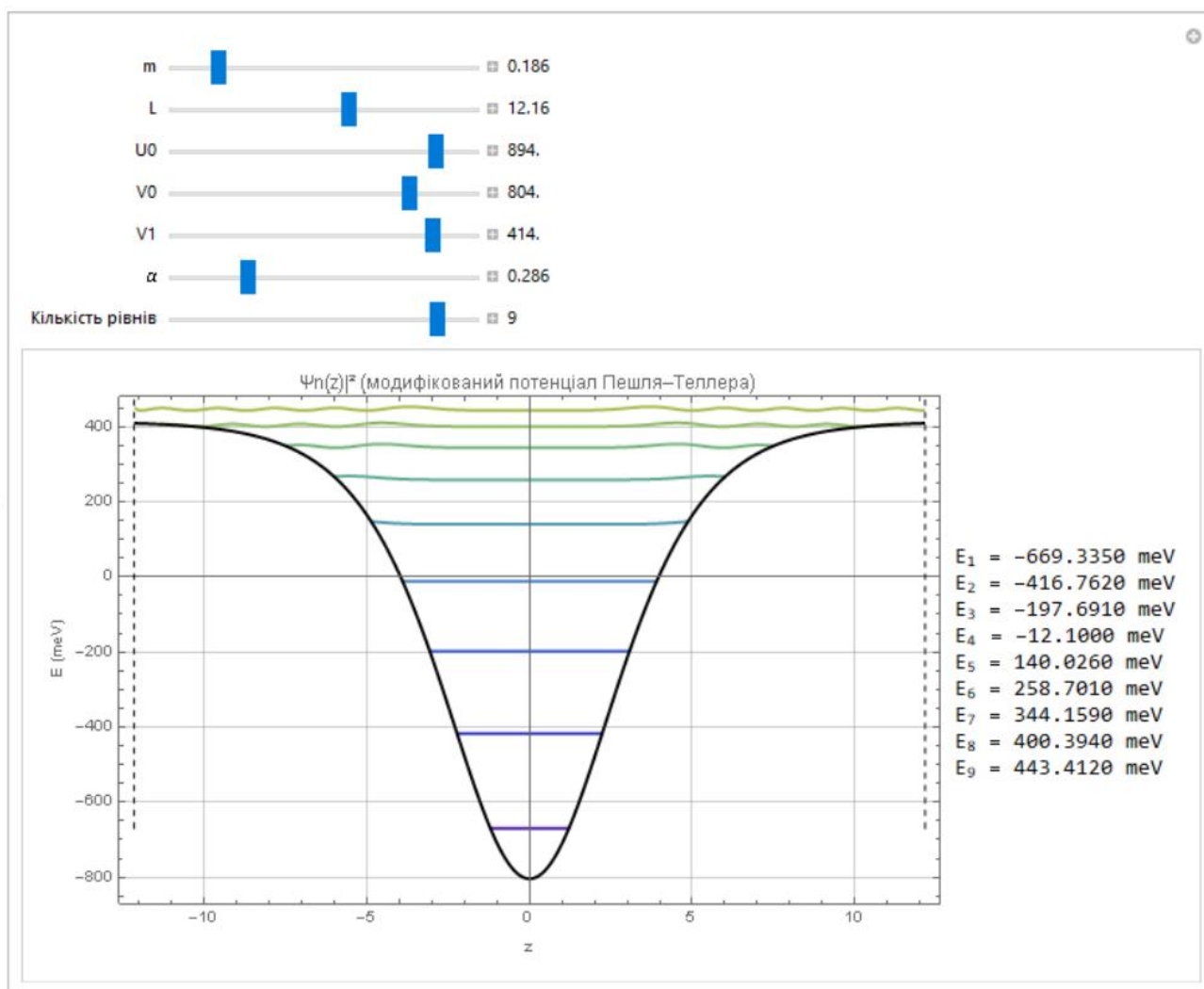


Рисунок 3.8 – Моделювання спектру та квадратів модулів хвильових функцій у потенціальній ямі з скінченними стінками для модифікованого потенціалу Пешля-Теллера

Для математичного моделювання коливальних процесів у двохатомних молекулах і в якості потенціальних пасток для таких молекул часто застосовується потенціал Морзе: $V(x) = D_e \left(1 - e^{-a(x-x_0)}\right)^2$. Цей потенціал у нашій узагальненій моделі характеризується таким додатковими параметрами: глибиною потенціальної ями D_e , ефективною шириною потенціалу a , та положенням мінімуму ефективного потенціалу x_0 . Складова програмної системи, що забезпечує ефективне математичне моделювання спектру та квадратів модулів хвильових функцій працює так, як це подано на рисунку 3.9. Важливою

особливістю розробленого нами програмного блоку є можливість прямого обчислення спектру і енергій квантових переходів, що дає змогу встановити енергії іонізації та енергії активації, можливі як у електроніці так і молекулярній динаміці.

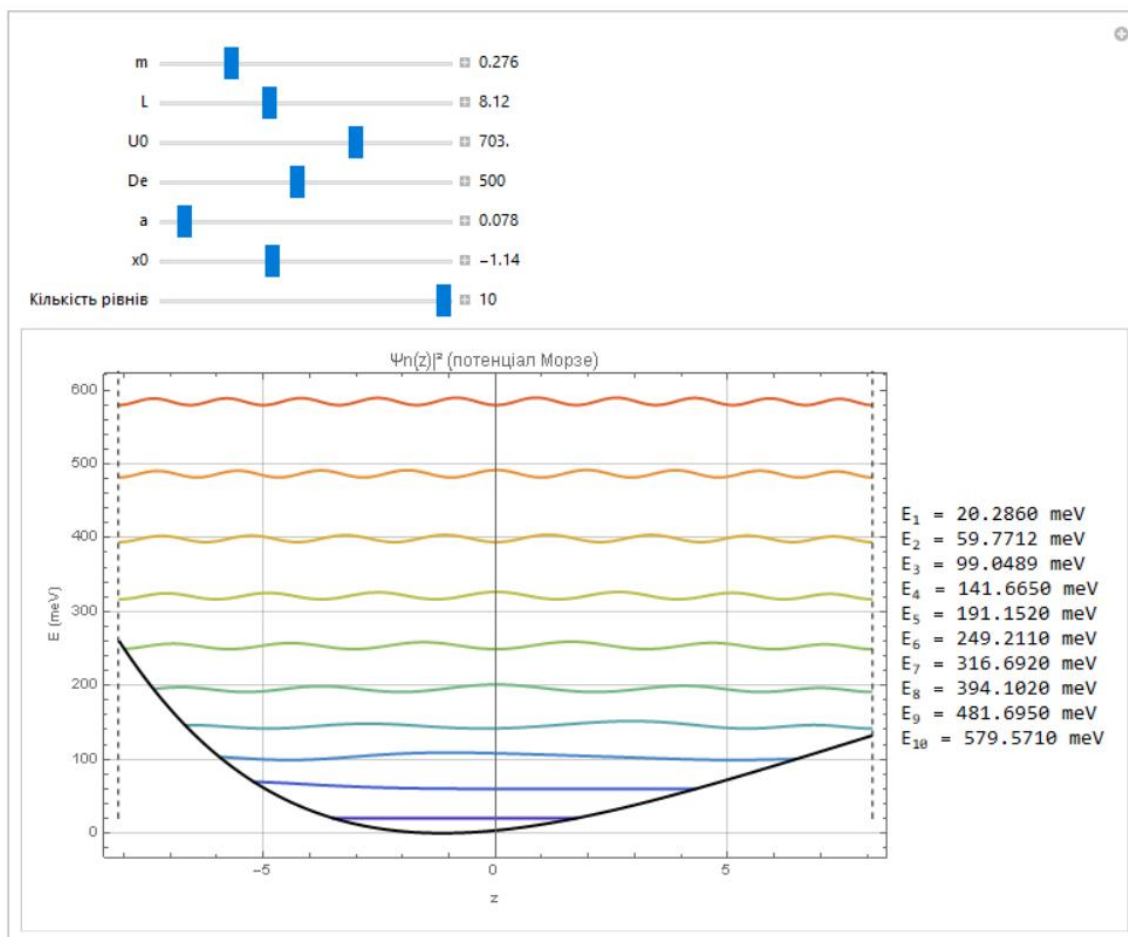


Рисунок 3.9 – Моделювання спектру та квадратів модулів хвильових функцій у потенціальній ямі з скінченними стінками для потенціалу Морзе

Останньою складовою блоку розробленої нами програмної системи є складова, що забезпечує математичне моделювання потенціалу Леннарда-Джонса:

$$V(x) = 4\varepsilon \left[\left(\frac{\sigma}{x} \right)^{12} - \left(\frac{\sigma}{x} \right)^6 \right], \text{ де } \varepsilon - \text{це глибина потенціальної ями, } \sigma - \text{відстань,}$$

при якій потенціал рівний нулю. Особливість даного потенціалу – часткові стани, що зв'язані потенціальною ямою при $V(x) < 0$, переходи між цим станами визначають енергію активації під час процесів адсорбції молекул в

функціональних матеріалах із мікропорами. Для повного вирішення цієї проблеми необхідно алгоритмічно реалізувати модель, що дозволяє виконувати математичне моделювання згаданих енергій при можливій зміні всіх параметрів як потенціальної ями, так і самого потенціалу Леннарда-Джонса. Результат роботи такої складової програмної системи подано на рисунку 3.10.

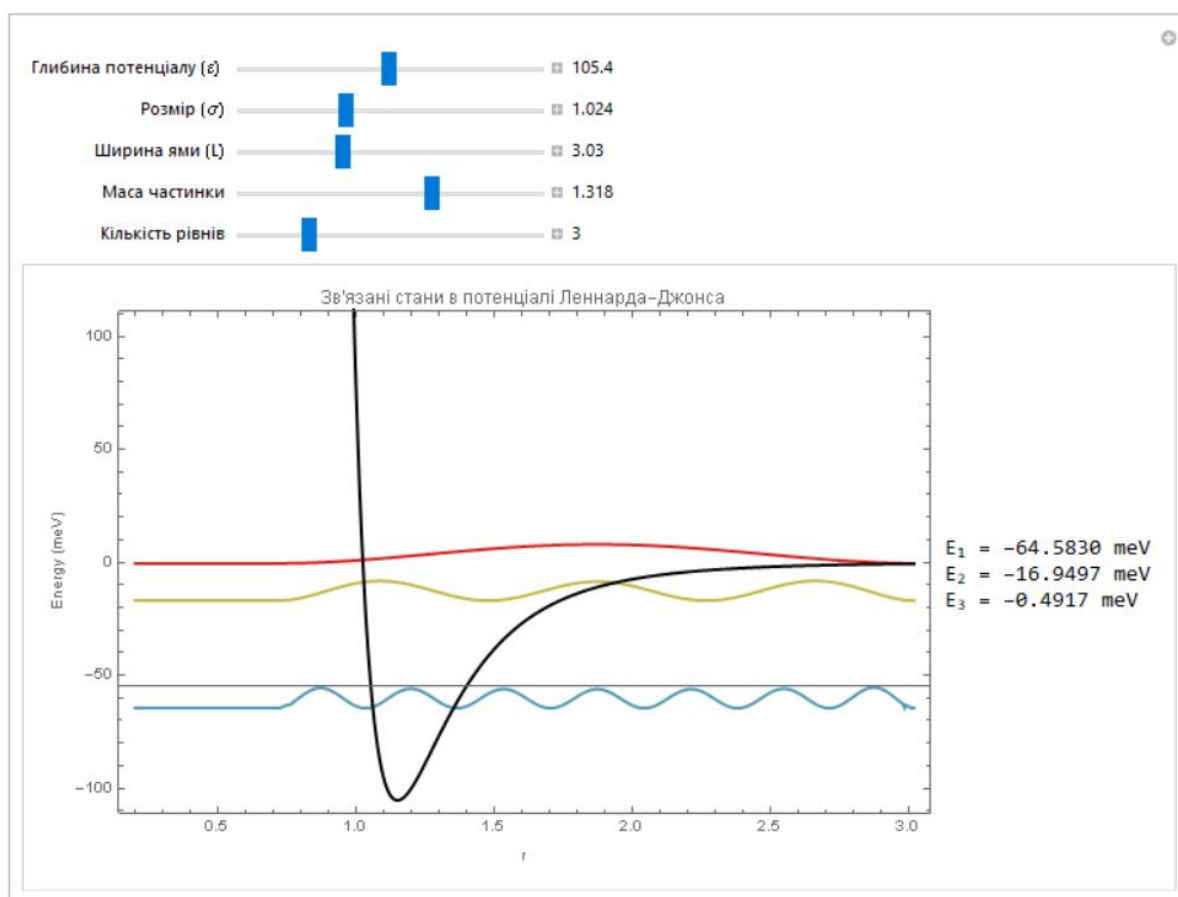


Рисунок 3.10 – Моделювання спектру та квадратів модулів хвильових функцій у потенціальній ямі з скінченними стінками для потенціалу Леннарда-Джонса

Розроблена система дає змогу розрахунку та візуалізації спектру електрона, квадратів модулів хвильових функцій та енергій активації.

3.5 Тестування та верифікація

Тестування та верифікацію програмного комплексу здійснено шляхом оцінки адекватності математичних моделей при використанні широкого спектра вхідних даних. Цей підхід забезпечує всебічну перевірку коректності роботи системи в різних умовах експлуатації та гарантує надійність отриманих результатів.

Методологія тестування базувалася на принципі поступового ускладнення тестових сценаріїв – від перевірки базових математичних операцій до валідації складних багатопараметричних моделей. Такий підхід дозволив виявити і усунути потенційні помилки на ранніх етапах тестування, що значно підвищило ефективність процесу верифікації.

Початковий етап тестування був зосереджений на валідації алгоритмів моделювання ефективних потенціалів. На цьому рівні перевірялася коректність базових математичних операцій, точність обчислення потенціальних функцій та адекватність їх представлення в дискретному просторі.

Тестування ефективних потенціалів включало:

- верифікацію аналітичних розв'язків для простих модельних систем;
- порівняння результатів з еталонними значеннями з наукової літератури;
- перевірку поведінки потенціалів на граничних умовах;
- аналіз стабільності обчислень при варіації параметрів моделі.

Комплексне параметричне тестування проводилось враховуючи якомога більшу кількість параметрів для забезпечення точності математичних моделей. Цей підхід передбачав систематичну варіацію всіх значущих параметрів системи в межах фізично обґрунтованих діапазонів.

Всі різницеві схеми досліджувались на збіжність за встановленим критерієм, а саме розраховуваною величиною відносного значення густини ймовірності. Аналіз збіжності є фундаментальним аспектом верифікації чисельних методів, оскільки гарантує, що при подібненні розрахункової сітки результати

наближаються до точного розв'язку рівняння. Для нашої моделі збіжність досяглась при 500 розбиттях, які забезпечують точність обчислень 10^{-7} (рис. 3.11). Процес збіжності автоматично запускається при компіляції програми, що забезпечує надійність функціонування системи.

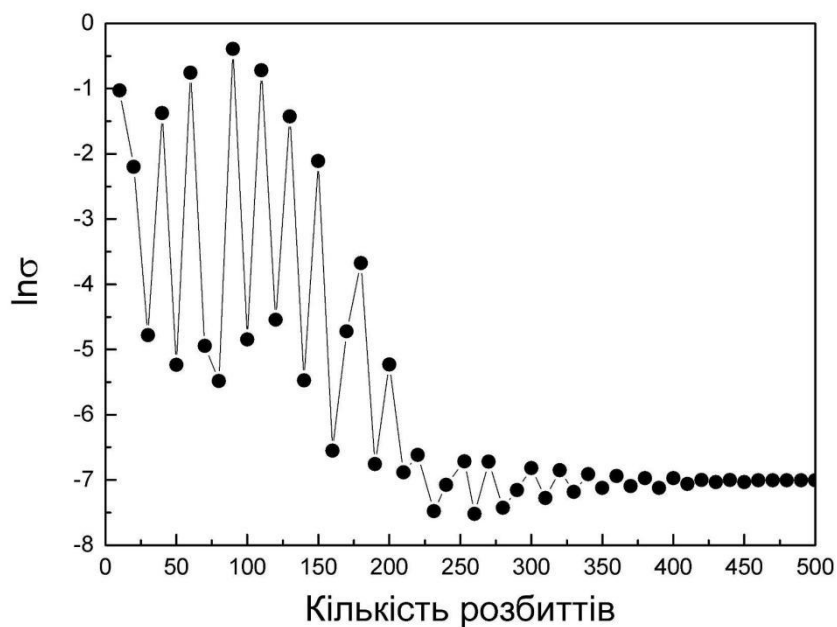


Рисунок 3.11 – Збіжність розрахунків у розробленій програмній системі

Така комплексна система тестування та верифікації забезпечила високу надійність програмного комплексу та довіру до отриманих з його допомогою результатів.

РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Долікарська допомога при ураженні електричним струмом.

Використання електричної енергії є невід'ємною складовою наукових досліджень, зокрема при роботі з комп'ютерною технікою. Незважаючи на відносно низькі рівні напруги в побутових ПК (220 В), неправильна експлуатація або технічні несправності можуть призвести до ураження електричним струмом. Дія електричного струму на організм людини залежить від типу струму, напруги, тривалості його проходження, шляху проходження, індивідуальних особливостей і оточуючого середовища [22].

Електричний струм, проходячи через тіло людини, викликає низку фізіологічних ефектів: судоми, зупинку серця, термічні опіки, електроліз тканин. У контексті експлуатації ПК виникають наступні ситуації:

- пошкодження ізоляції проводів живлення або блоку живлення;
- дотик до неізованих елементів при розібраному корпусі ПК;
- використання несправних подовжувачів або розеток;
- висока вологість у приміщенні або відсутність заземлення.

Особлива увага приділяється дотриманню наступних правил при користуванні ПК чи ПЕОМ, відповідно до яких необхідно:

- використовувати справні мережеві фільтри з функцією захисту від перенапруги;
- уникати відкритого контакту з внутрішніми компонентами ПК, особливо при ввімкненому живленні;
- регулярно перевіряти заземлення комп'ютерної техніки;
- дотримуватись інструкцій виробника при використанні лабораторного устаткування;
- не працювати на вологих поверхнях або з мокрими руками.

Важливо пам'ятати, що навіть при низьких напругах можливе ураження струмом – особливо при підвищеній вологості або пошкодженій ізоляції [23].

Згідно з Порядком надання домедичної допомоги постраждалим при ураженні електричним струмом або блискавкою Наказу Міністерства охорони здоров'я України від 09 березня 2022 року № 441, першочергові кроки до та при наданні домедичної допомоги включають [24]:

- 1) перед наданням допомоги переконатися у відсутності небезпеки для себе, оточуючих, постраждалого та тільки за її відсутності перейти до наступного кроку;
- 2) якщо постраждалий у свідомості, заспокоїти та пояснити свої наступні дії;
- 3) здійснити виклик екстреної медичної допомоги та дотримуватись вказівок диспетчера прийому виклику;
- 4) при ураженні постраждалого електричним струмом:
 - якщо постраждалий без свідомості, впевнитись, що дія електричного струму на постраждалого припинена;
 - всі дії по припиненню дії електричного струму слід здійснювати за умови проходження відповідного навчання або здійснити виклик за єдиним телефонним номером системи екстреної допомоги населенню 112;
 - якщо дія електричного струму на постраждалого припинена, слід надати йому домедичну допомогу, відповідно до наявних пошкоджень;
- 5) забезпечити постійний нагляд за постраждалим до приїзду бригади екстреної (швидкої) медичної допомоги;
- 6) при погіршенні стану постраждалого до приїзду бригади екстреної (швидкої) медичної допомоги повторно здійснити виклик екстреної медичної допомоги;
- 7) за можливості зібрати у постраждалого чи оточуючих максимально можливу інформацію стосовно обставин отримання травми. Всю отриману інформацію передати працівникам бригади екстреної (швидкої) медичної допомоги або диспетчеру служби екстреної медичної допомоги [24].

Знання та практичні навички надання долікарської допомоги при ураженні електричним струмом є необхідними для кожного, хто працює з електронним обладнанням, особливо в рамках розробки наукових програмних продуктів.

Врахування електробезпеки під час роботи з ПК не лише зберігає здоров'я, а й запобігає втратам обладнання та зупинці дослідницького процесу.

4.2 Інженерно-технічні рішення з охорони праці.

Розробка та експлуатація програмного комплексу для математичного моделювання енергій активації у цеолітах передбачає інтенсивне використання комп'ютерної техніки. Важливо забезпечити безпечні електротехнічні, ергономічні та інженерно-технічні умови не лише для збереження здоров'я персоналу, а й неперервності дослідницького процесу.

У межах охорони праці під час проєктування та експлуатації програмного комплексу впроваджуються такі інженерно-технічні рішення, відповідно до ДСТУ EN ISO 11064-5:2017 та ДСТУ EN ISO 9241-13:2017 [25, 26]:

1) оснащення робочого місця , а саме використання :

- використання ергономічного крісла з відповідною високою, шириною, глибиною та кутом сидіння, щоб забезпечувати зміни пози і достатній комфорт для ефективного виконання завдання;
- використання стола з регулюванням висоти, який має дозволяти зручне та ефективне положення верхньої частини рук, передпліч і кистей рук;
- регулювання монітора на рівні очей, з дотриманням кута зору(оптимальний 0°) не більше 40° будь-де на активній області дисплею:

2) організація проводки, а саме:

- з'єднання надійно закріплені так, щоб вони не спричиняли небезпеки, тягнучись упоперек робочої поверхні чи підлоги;
- довжина кабелів достатня, щоб задовольнити фактичні і прогнозовані потреби користувачів;
- проводка здатна забезпечити повний діапазон регулювання, якщо є регульовані поверхні;

- 3) утримання провітрюваного приміщення з рівномірним освітленням: ефективний рух повітря (щоденне провітрювання або припливно-витяжна вентиляція) та рівномірне, без тіней, освітлення — важлива частина забезпечення нормального мікроклімату та профілактики перевтоми очей;
- 4) регламентація режиму праці й відпочинку шляхом впровадження перерви для відпочинку тривалістю 15 хвилин через кожні дві години, впровадження вправ для очей та м'язів верхніх кінцівок [25, 26];
- 5) дотримання оптимально мікроклімату та чистоти: оптимально 18–24 °С, вологість – $\leq 60\%$; очистка підставок, корпусів та вентиляційних каналів для забезпечення електротехнічної безпеки і стабільності роботи електроніки
- 6) використання звукопоглинальних матеріалів для обробки стін, стель і підлоги в приміщеннях з декількома комп'ютерними станціями для зниження шуму [27].

Інженерно-технічні заходи передбачають не лише створення безпечного середовища, а й:

- проведення інструктажу з охорони праці;
- навчання основам електробезпеки, правил надання домедичної допомоги при ураженні струмом [28].

Варто зазначити, що ПК загального призначення не використовуються в умовах підвищеної небезпеки, тому експлуатація комп'ютерної техніки такого типу не вважається роботою з підвищеною небезпекою. Отже впроваджені інженерно-технічні рішення не враховують заміну небезпечних матеріалів на менш небезпечні в проєктуванні. Також розробка окремої інструкції з охорони праці при використанні комп'ютерної техніки є недоцільною, достатньо використати чи розробити інструкцію з електробезпеки, яка буде враховувати специфіку обладнання, що використовується.

Використання сучасних інженерно-технічних рішень, згідно з нормами інтер'єру, мікроклімату та акустики, має не лише утримувати наукове обладнання в працездатному стані, але й створювати безпечне середовище для персоналу.

ВИСНОВКИ

У дипломній роботі проведено аналіз предметної області галузей мікро- та наноелектроніки в розрізі застосування методології та підходів інженерії програмного забезпечення для розробки та верифікації математичних моделей ефективних потенціалів, виникаючих при моделюванні квазічастинкових станів у мезоскопічних структурах. Встановлено, що реалізації таких математичних моделей є необхідний цілісний підхід, який в кінцевому результаті має бути поданий у вигляді складної програмної системи, кожна з програмних компонент якої відповідає за візуалізацію потенціалу та реалізацію самої математичної моделі.

У результаті, розроблено математичні моделі різноманітних ефективних потенціалів й з використанням середовища Wolfram Mathematica здійснено їх візуалізацію за допомогою програмних компонент, забезпечуючих зміну їх основних параметрів. Ці програмні блоки складають першу частину розробленої програмної системи.

Для другого блоку програмної системи на основі математичних моделей крайових задач із різноманітними ефективними потенціалами для рівняння Шредінгера побудовано їх різницеві схеми, виконано їх програмну реалізацію. Кожна із складових цього програмного блоку забезпечує математичне моделювання спектру, локалізації та розподілу квазічастинкових станів шляхом зміни параметрів потенціальних ям, усіх параметрів ефективних потенціалів та фізичних параметрів використовуваних матеріалів.

Для даної програмної системи розроблено гнучку архітектуру, яка забезпечує чіткий та повністю візуалізований доступ до графічної та числової інформації, отримуваної від кожної із побудованих моделей. Програмна система за своїм призначенням знайде застосування як в галузі електроніки, так і суміжних і міждисциплінарних галузях, пов'язаних із застосуванням

мікропористих матеріалів у системах очистки повітря, та моніторингу викидів шкідливих речовин у атмосферу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дистанційний курс «Кваліфікаційні роботи бакалаврів» сайту дистанційного навчання ТНТУ. *Atutor*. URL: <https://dl.tntu.edu.ua/bounce.php?course=5329> (дата звернення: 15.02.2025)
2. Петрик М.Р. Моделі та методи моделювання складних процесів в наноструктурах і нанопористих середовищах (на основі високопродуктивних обчислень): монографія / Петрик М.Р., Бойко І.В. – Тернопіль : Вид-во ТНТУ імені Івана Пулюя, 2021. – 140 с.
3. S. Wang, X. Zhou, J. Zheng. Optimization of GeSn nanostructures via tuning of femtosecond laser parameters // *Applied Surface Science*. 2025. №679. P. 161153.
4. K. Li, C. Wang, L. Sun. Laser-assisted electrohydrodynamic jet printing of hierarchical nanostructure // *Applied Thermal Engineering*. 2024. №253. P. 123659.
5. J. P. Labis, H. A. Albrithen, M. A. Shar. Synthesis of Sn-ZnO nanostructures on MgO<0001> by hybrid pulsed laser ablation and RF magnetron sputtering tandem system for CO gas-sensing application // *Journal of Saudi Chemical Society*. 2024. №28. P. 101941.
6. J. Liu, Q. Jiang, J. Yan, J. Geng, L. Shi. Femtosecond pulse train-facilitated periodic nanostructuring on TiN films via laser-oxidation // *Optics & Laser Technology*. 2024. №177. P. 111189.
7. I. Solana, F. Chacon-Sanchez, M. Garcia-Lechuga, J. Siegel. Versatile femtosecond laser interference patterning applied to high-precision nanostructuring of silicon // *Optics & Laser Technology*. 2024. №179. P. 111360.
8. K. Cao, W. Ye, Y. Zhang. Fabrication of multifunctional Co,N co-doped UIO-rGO aerogel with properties of hydrophobic, anti-corrosion, heat insulation, infrared stealth and electromagnetic wave absorption // *Chemical Engineering Journal*. 2024. №492. P. 152275.

9. Z. Ling, J. Chen, S. Li. A multi-band stealth and anti-interference superspeed light-guided swimming robot based on multiscale bicontinuous three-dimensional network // Chemical Engineering Journal. 2024. №485. P. 150094.
10. Y. Zhang, G. Zhang, C. Wang. The construction of structural defects in MoAlB thin films by leveraging the enhanced Kirkendall effect for improved infrared stealth performance // Ceramics International. 2024. №50. P. 21175-21183.
11. S. M. Zanata, N. M. El-Shafai, A. M. Beltagi. Bio-study: Modeling of natural nanomolecules as a nanocarrier surface for antioxidant and glucose biosensor // International Journal of Biological Macromolecules. 2024. №264. P. 130634.
12. J. M. Feugang, G. M. Ishak, M. W. Eggert. Intrafollicular injection of nanomolecules for advancing knowledge on folliculogenesis in livestock // Theriogenology. 2022. №192. P. 132-140.
13. X. Li PhD, M. Wu MS, J. Wang MS. Ultrasmall bimodal nanomolecules enhanced tumor angiogenesis contrast with endothelial cell targeting and molecular pharmacokinetics // Nanomedicine: Nanotechnology, Biology and Medicine. 2019. №15. P. 252-263.
14. J. L. Santailler, T. Duffar, F. Théodore. Some features of two commercial softwares for the modeling of bulk crystal growth processes // Journal of Crystal Growth. 1997. №180. P. 698-710.
15. D. Leng, P. Li, F. Kong. Experimental and numerical study on single ice crystal growth of deionized water and 0.9 % NaCl solution under static magnetic field // International Journal of Refrigeration. 2024. №168. P. 297-306.
16. NextNano Software. URL: <https://www.nextnano.de/products/overview.php> (date of access: 20.02.2025)
17. NEMO – 3D. URL: <https://engineering.purdue.edu/gekcogrp/software-projects/nemo3D/> (date of access: 20.02.2025)
18. J. D. Cooper; A. Valavanis; Z. Ikonić; P. Harrison; J. E. Cunningham. Finite difference method for solving the Schrödinger equation with band nonparabolicity in mid-infrared quantum cascade lasers // J. Appl. Phys. 2010. №168. P. 297-306.

19. W. Wang, T.-M. Hwang, W.-W. Lin, J.-L. Liu. Numerical methods for semiconductor heterostructures with band nonparabolicity // Journal of Computational Physics. 2003. №108. P. 113109.
20. T. Xue, Y. Yang, D. Yu, Q. Wali,. 3D Printed Integrated Gradient-Conductive MXene/CNT/Polyimide Aerogel Frames for Electromagnetic Interference Shielding with Ultra-Low Reflection // Nano-Micro Lett. 2023. №15. P. 45.
21. Igor Boyko, Sophia Khemii. SOFTWARE SYSTEM FOR MATHEMATICAL MODELING THE INFLUENCE OF EFFECTIVE POTENTIALS ON ELECTRON STATES IN QUANTUM WELLS. // Scientific Journal of TNTU. — Tern.: TNTU, 2025. — Vol 118. — No 2. — P. 15–26.
22. Желібо Є. П. Безпека життєдіяльності. навч. посіб. / Є. П. Желібо, Н. М. Заверуха, В. В. Зацарний. 6-те вид. Київ : Каравела, 2023. 344 с.
23. ТЕМА 17. Електробезпека. URL: <https://dl.tntu.edu.ua/content.php?cid=289206> (дата звернення: 13.06.2025).
24. Про затвердження порядків надання домедичної допомоги особам при невідкладних станах: наказ МОЗ України від 09.03.2022 №441. URL: <https://zakon.rada.gov.ua/laws/show/z0356-22/conv#n800> (дата звернення: 13.06.2025).
25. ДСТУ EN ISO 11064-5:2017. Проектування центрів керування ергономічне. Частина 5. Засоби відображення інформації та органи керування (EN ISO 11064-5:2008, IDT; ISO 11064-5:2008, IDT). Чинний від 2019-1-1. Вид. офіц. Київ : УкрНДНЦ, 2017.
26. ДСТУ EN ISO 9241-13:2017 Ергономічні вимоги до роботи з відеотерміналами в офісі. Частина 13. Настанова щодо використання (EN ISO 9241-13:1998, IDT; ISO 9241-13:1998, IDT). Чинний від 2019-1-1. Вид. офіц. Київ : УкрНДНЦ, 2017.
27. Навчально-методичний посібник до практичних занять з дисципліни «Безпека життєдіяльності, основи охорони праці» для студентів освітнього ступеня „бакалавр" усіх спеціальностей та форм навчання / Укладачі : О. Я. Гурик, І. Б. Окіпний, В. С. Сенчишин, С. Ю. Мариненко, О. І. Король. Тернопіль : ТНТУ

імені Івана Пулюя, 2025. 123 с. URL: <http://elartu.tntu.edu.ua/handle/lib/48496> (дата звернення: 13.06.2025).

28. Грибан В. Г., Негодченко О. В. Г 82 Охорона праці. Навч. посіб. 2-ге вид. - К.: Центр учбової літератури, 2019. - 280 с.

ДОДАТКИ

ДОДАТОК А

Лістинг програмного коду для візуалізації ефективних потенціалів та математичного моделювання енергій активації з різними потенціалами

Гармонічний осцилятор

```
harmonicPotential[x_, k_, x0_, wellWidth_] := If[Abs[x - x0] <= wellWidth/2, (1/2) k
(x - x0)^2, 0]

Manipulate[Module[{left, right}, left = x0 - wellWidth/2; (*Ліва межа ями*)
right = x0 + wellWidth/2; (*Права межа ями*)
Plot[{harmonicPotential[x, k, x0, wellWidth]}, {x, -10, 10},
PlotRange -> {0, 500}, AxesLabel -> {"x (нм)", "V(x) (меВ)"},
Ticks -> Automatic, PlotStyle -> {Thick, Green}, GridLines -> {{left, right}, None},
GridLinesStyle -> {{Directive[Red, Dashed]}, None}, PlotTheme -> "Scientific",
PlotLabel -> "Гармонічний осцилятор\nЖорсткість: "<>ToString[k]<>" меВ/нмІ,
Центр: "<>ToString[x0]<>" нм, Ширина ями: "<>ToString[wellWidth]<>" нм"]],
{{k, 10, "Жорсткість (k), (меВ/нмІ)"}, 1, 100, 1, Appearance -> "Labeled"},
{{x0, 0, "Центр (x0), (нм)"}, -5, 5, 0.1, Appearance -> "Labeled"},
{{wellWidth, 5, "Ширина потенціальної ями (w), (нм)"}, 1, 10, 0.1, Appearance -> "Labeled"}]
```

Ангармонічний осцилятор

```
anharmonicPotential[x_, k_, α_, wellWidth_] := If[2 <= x <= wellWidth, (1/2) * k * x^2 + α *
x^4, 0]

Manipulate[Module[{left, right}, left = -2; right = wellWidth;
Plot[anharmonicPotential[x, k, α, wellWidth], {x, -5, 15},
PlotRange -> {0, 500}, AxesLabel -> {"x (нм)", "V(x) (меВ)"},
Ticks -> Automatic, PlotStyle -> {Thick, Blue}, GridLines -> {{left, right}, None},
GridLinesStyle -> {{Directive[Red, Dashed]}, None}, PlotTheme -> "Scientific",
PlotLabel -> StringJoin["Ангармонічний осцилятор\n Жорсткість:
", ToString[k], " меВ/нмІ, \n Ангармонічність: ", ToString[α], " меВ/нм^4,
Ширина ями: від -2 до ", ToString[wellWidth], " нм"]]],
{{k, 10, "Жорсткість (k), (меВ/нмІ)"}, 1, 100, 1, Appearance -> "Labeled"},
{{α, 0.05, "Ангармонічність (α), (меВ/нм^4)"}, 0, 5, 0.1, Appearance -> "Labeled"},
{{wellWidth, 10, "Ширина потенціальної ями (w), (нм)"}, -1, 10, 0.1, Appearance -> "Labeled"}]
```

Потенціал Пешль-Теллера

```
potential[x_, V0_, a_, L1_, L2_] := If[L1 <= x <= L2, -V0 Sech[(x - L1)/a]^2, 0]

Manipulate[Module[{left, right, gradLines}, left = L1; right = L2;
gradLines = Table[i, {i, left, right, (right - left)/10}];
Plot[potential[x, V0, a, L1, L2], {x, -15, 20},
PlotRange -> {Min[-V0, -10], 10}, AxesLabel -> {"x (нм)", "V(x) (меВ)"},
Ticks -> Automatic, PlotStyle -> {Thick, Blue},
GridLines -> {Join[{left, right}, gradLines], None},
```

```
GridLinesStyle->{Join[{Directive[Red,Dashed]}],
Table[Directive[Gray,Dashed,Opacity[0.5]],{10}]],None},
PlotTheme->"Scientific",PlotLabel->StringJoin[{"Потенціал Пешля-
Теллера\nЛіва межа: ",ToString[L1]," нм, ", "Права межа: ",ToString[L2]," нм,
","Глибина: ",ToString[V0]," меВ"}]],
{{V0,100,"Глибина V0, (меВ)"},10,500,10,Appearance->"Labeled"},
{{a,10,"Ширина a, (нм)"},1,20,1,Appearance->"Labeled"},
{{L1,-5,"Ліва межа (нм)"},-10,5,0.1,Appearance->"Labeled"},
{{L2,8,"Права межа (нм)"},0,15,0.1,Appearance->"Labeled"}]
```

Модифікований потенціал Пешль-Теллера

```
modifiedPTPotential[x_,V0_,a_,L1_,L2_] := If[L1<=x<=L2,-V0/Cosh[(x-L1)/a]^2,0]

Manipulate[Module[{left,right,gradLines},left=L1;right=L2;
gradLines=Table[i,{i,left,right,(right-left)/10}];
Plot[modifiedPTPotential[x,V0,a,L1,L2],{x,-15,20},
PlotRange->{Min[-V0,-10],10},AxesLabel->{"x (нм)","V(x) (меВ)"},
Ticks->Automatic,PlotStyle->{Thick,Purple},
GridLines->{Join[{left,right},gradLines],None},
GridLinesStyle->{Join[{Directive[Red,Dashed]}],
Table[Directive[Gray,Dashed,Opacity[0.5]],{10}]],None},
PlotTheme->"Scientific",PlotLabel->StringJoin[{"Модифікований потенціал
Пешля-Теллера\nЛіва межа: ",ToString[L1]," нм, Права межа: ",ToString[L2],"
нм, Глибина: ",ToString[V0]," меВ"}]],
{{V0,100,"Глибина V0, (меВ)"},10,500,10,Appearance->"Labeled"},
{{a,1,"Ширина a, (нм)"},0.1,10,0.1,Appearance->"Labeled"},
{{L1,-5,"Ліва межа (нм)"},-10,5,0.1,Appearance->"Labeled"},
{{L2,8,"Права межа (нм)"},0,15,0.1,Appearance->"Labeled"}]
```

Потенціал Морзе

```
morsePotential[x_,D0_,a_,L1_,L2_] := If[L1<=x<=L2,D0 (1-Exp[-(x-L1)/a])^2,0]

Manipulate[Module[{left,right,gradLines},left=L1;right=L2;
gradLines=Table[i,{i,left,right,(right-left)/10}];
Plot[morsePotential[x,D0,a,L1,L2],{x,-5,15},PlotRange->{0,D0+10},
AxesLabel->{"x (нм)","V(x) (меВ)"},Ticks->Automatic,
PlotStyle->{Thick,Red},GridLines->{Join[{left,right},gradLines],None},
GridLinesStyle->{Join[{Directive[Red,Dashed]}],
Table[Directive[Gray,Dashed,Opacity[0.5]],{10}]],None},
PlotTheme->"Scientific",
PlotLabel->StringJoin[{"Потенціал Морзе\nЛіва межа: ",ToString[L1]," нм,
Права межа: ",ToString[L2]," нм, Глибина: ",ToString[D0]," меВ"}]],
{{D0,100,"Глибина D0, (меВ)"},10,500,10,Appearance->"Labeled"},
{{a,1,"Ширина a, (нм)"},0.1,5,0.1,Appearance->"Labeled"},
{{L1,-2,"Ліва межа (нм)"},-5,5,0.1,Appearance->"Labeled"},
{{L2,8,"Права межа (нм)"},0,10,0.1,Appearance->"Labeled"}]
```

Потенціал Леннарда-Джонса

```
lennardJonesPotential[x_, ε_, σ_, L_, wellWidth_] := If[L <= x <= L + wellWidth, 4 ε
((σ/x)^12 - (σ/x)^6), 0]

Manipulate[Module[{left, right, gradLines}, left = L; right = L + wellWidth;
gradLines = Table[i, {i, left, right, (right - left) / 10}];
Plot[lennardJonesPotential[x, ε, σ, L, wellWidth], {x, 0.1, 2},
PlotRange -> {-(ε + 10), 2 ε}, AxesLabel -> {"x (нм)", "V(x) (меВ)"},
Ticks -> Automatic, PlotStyle -> {Thick, Red},
GridLines -> {Join[{left, right}, gradLines], None},
GridLinesStyle -> {Join[{Directive[Red, Dashed]}],
Table[Directive[Gray, Dashed, Opacity[0.5]], {10}]], None}, PlotTheme -
> "Scientific", PlotLabel -> StringJoin["Потенціал Леннарда-Джонса\nЛіва межа: ",
ToString[L], " нм, Ширина: ", ToString[wellWidth], " нм, Енергія взаємодії: ",
ToString[ε], " меВ"]]],
{{ε, 100, "Енергія взаємодії (меВ)"}, 10, 500, 10, Appearance -> "Labeled"},
{{σ, 0.3, "Характерна відстань (нм)"}, 0.1, 1, 0.05, Appearance -> "Labeled"},
{{L, 0.5, "Ліва межа (нм)"}, 0.2, 1.0, 0.05, Appearance -> "Labeled"},
{{wellWidth, 1.0, "Ширина потенціальної ями (нм)"}, 0.2, 1.5, 0.05, Appearance -
> "Labeled"}]
```

Система для математичного моделювання енергій активації з різними потенціалами

```
(*Визначаємо параметри задачі*) m = 0.186;
ab = 0.0529;
Ry = 13605;
U0 = 0;
L = 10.0; (*ширина ями*) n = 500; (*кількість вузлів дискретизації*) (*Визначаємо
масштабний коефіцієнт*) hbar2Div2m = ((Sqrt[2] * ab * Sqrt[Ry])^2) / (2 * m);

(*Крок дискретизації*)
dx = 2 * L / (n + 1);

(*Ініціалізація матриці*)
A = Table[0, {n}, {n}];

(*Заповнення матриці методом кінцевих різниць*)
Do[If[i > 1, A[[i, i - 1]] = -hbar2Div2m / dx^2]; (*Ліва сусідня
точка*) If[i < n, A[[i, i + 1]] = -hbar2Div2m / dx^2]; (*Права сусідня
точка*) A[[i, i]] = 2 * hbar2Div2m / dx^2; (*Головна діагональ*), {i, 1, n}];

(*Розв'язок на власні значення*)
{Evals, wfn} = Eigensystem[A];

(*Сортуємо власні значення та відповідні власні вектори*)
sortedIndices = Ordering[Evals];
Evals = Evals[[sortedIndices]];
wfn = wfn[[All, sortedIndices]];

(*Масштабування координати (зміщене на центр ями)*)
```

```

x=Table[-L+i*dx,{i,1,n}];

(*Побудова графіка енергетичних рівнів з хвильовими функціями*)
Show[Table[Module[{PsiSquared},PsiSquared=15*wfn[[All,i]]^2/Max[wfn[[All,i]]^2]+Evals[[i]];(*Масштабування*)ListLinePlot[Transpose[{x,PsiSquared}],PlotStyle->{Thick,ColorData["Rainbow"][i/5]},PlotRange->All]],{i,1,5}],PlotRange->All,Frame->True,FrameLabel->{"z","Energy and |Ψ(z)|^2"},PlotLabel->"Ψn(z)|^2",GridLines->Automatic,ImageSize->Large,Epilog->{Dashed,Line[{{-L,Min[Evals]}},{-L,Max[Evals]}],Dashed,Line[{{L,Min[Evals]}},{L,Max[Evals]}]}]]

(*Визначаємо параметри задачі*)m=0.186;
ab=0.0529;
Ry=13605;
U0=500;(*Висота стінок*)L=10.0;(*ширина ями*)n=500;(*кількість вузлів\
дискретизації*)(*Визначаємо масштабний
коефіцієнт*)hbar2Div2m=((Sqrt[2]*ab*Sqrt[Ry])^2)/(2*m);

(*Крок дискретизації*)
dx=2*L/(n+1);

(*Ініціалізація матриці*)
A=Table[0,{n},{n}];

(*Заповнення матриці методом кінцевих різниць*)
Do[If[i>1,A[[i,i-1]]=-hbar2Div2m/dx^2];(*Ліва сусідня
точка*)If[i<n,A[[i,i+1]]=-hbar2Div2m/dx^2];(*Права сусідня
точка*)A[[i,i]]=2*hbar2Div2m/dx^2+(-L+i*dx)^2;(*Головна діагональ
(потенціал x^2)*),{i,1,n}];

(*Додавання потенціалу стінок до матриці*)
A[[1,1]]+=U0;(*Потенціал на лівій стінці*)A[[n,n]]+=U0;(*Потенціал на
правій стінці*)(*Розв'язок на власні значення*){Evals,wfn}=Eigensystem[A];

(*Сортуємо власні значення та відповідні власні вектори*)
sortedIndices=Ordering[Evals];
Evals=Evals[[sortedIndices]];
wfn=wfn[[All,sortedIndices]];

(*Масштабування координати (зміщене на центр ями)*)
x=Table[-L+i*dx,{i,1,n}];

(*Вибираємо рівні нижче 500*)
levelsBelow500=Select[Transpose[{Evals,wfn}],First[#]<U0&];
EvalsBelow500=levelsBelow500[[All,1]];
wfnBelow500=levelsBelow500[[All,2]];

(*Побудова графіка енергетичних рівнів з хвильовими функціями та
потенціалом*)
Show[Table[Module[{PsiSquared},PsiSquared=15*wfnBelow500[[i]]^2/Max[wfnBelow500[[i]]^2]+EvalsBelow500[[i]];(*Масштабування*)ListLinePlot[Transpose[{x,PsiSquared}],PlotStyle->{Thick,ColorData["Rainbow"][i/Length[EvalsBelow500]]},PlotRange->All]],{i,1,Length[EvalsBelow500]}],ListLinePlot[Transpose[{x,x^2}],PlotStyle->{Thick,Black}],(*Додавання графіка потенціалу*)PlotRange->All,Frame->True,FrameLabel->{"z","Energy and |Ψ(z)|^2"},PlotLabel->"Ψn(z)|^2 (рівні нижче 500)",GridLines->Automatic,ImageSize->Large,Epilog->{Dashed,Line[{{-L,Min[EvalsBelow500]}},{-L,Max[EvalsBelow500]}],Dashed,Line[{{L,Min[EvalsBelow500]}},{L,Max[EvalsBelow500]}]}]]

```

```

L, Max[EvalsBelow500]]}], Dashed, Line[{{L, Min[EvalsBelow500]}, {L, Max[EvalsBelow500]]}]]]

(*Визначаємо параметри задачі*)m=0.186;
ab=0.0529;
Ry=13605;
U0=500; (*Висота стінок*)L=10.0; (*ширина ями*)n=500; (*кількість вузлів\
дискретизації*) (*Визначаємо масштабний
коефіцієнт*)hbar2Div2m=((Sqrt[2]*ab*Sqrt[Ry])^2)/(2*m);

(*Крок дискретизації*)
dx=2*L/(n+1);

(*Ініціалізація матриці*)
A=Table[0,{n},{n}];

(*Заповнення матриці методом кінцевих різниць*)
Do[If[i>1,A[[i,i-1]]=-hbar2Div2m/dx^2];(*Ліва сусідня
точка*)If[i<n,A[[i,i+1]]=-hbar2Div2m/dx^2];(*Права сусідня
точка*)A[[i,i]]=2*hbar2Div2m/dx^2+(-L+i*dx)^2;(*Головна діагональ
(потенціал x^2)*),{i,1,n}];

(*Додавання потенціалу стінок до матриці*)
A[[1,1]]+=U0;(*Потенціал на лівій стінці*)A[[n,n]]+=U0;(*Потенціал на
правій стінці*)(*Розв'язок на власні значення*){Evals,wfn}=Eigensystem[A];

(*Сортуємо власні значення та відповідні власні вектори*)
sortedIndices=Ordering[Evals];
Evals=Evals[[sortedIndices]];
wfn=wfn[[All,sortedIndices]];

(*Масштабування координати (зміщене на центр ями)*)
x=Table[-L+i*dx,{i,1,n}];

(*Вибираємо рівні нижче 500*)
levelsBelow500=Select[Transpose[{Evals,wfn}],First[#]<U0&];
EvalsBelow500=levelsBelow500[[All,1]];
wfnBelow500=levelsBelow500[[All,2]];

(*Побудова графіка енергетичних рівнів з хвильовими функціями та
потенціалом*)
Show[Table[Module[{PsiSquared,clippedData},PsiSquared=15*wfnBelow500[[i]]^2
/Max[wfnBelow500[[i]]^2]+EvalsBelow500[[i]];(*Масштабування*)clippedData=Tr
anspose[{x,PsiSquared}];
    clippedData=Select[clippedData,#[[2]]>=#[[1]]^2&];(*Видаляємо точки
нижче потенціалу*)ListLinePlot[clippedData,PlotStyle-
>{Thick,ColorData["Rainbow"][i/Length[EvalsBelow500]]},PlotRange-
>All],{i,1,Length[EvalsBelow500]}],ListLinePlot[Transpose[{x,x^2}],PlotSty
le->{Thick,Black}],(*Графік потенціалу*)PlotRange->All,Frame-
>True,FrameLabel->{"z","Energy and |Ψ(z)|^2"},PlotLabel->"Ψn(z)|b2 (рівні
нижче 500)",GridLines->Automatic,ImageSize->Large,Epilog->{Dashed,Line[{{-
L,Min[EvalsBelow500]},{-
L,Max[EvalsBelow500]}]},{Dashed,Line[{{L,Min[EvalsBelow500]},{L,Max[EvalsBel
ow500]}]}}]}]]]

(*Визначаємо параметри задачі*)m=0.186;
ab=0.0529;
Ry=13605;

```

```

U0=500; (*Висота стінок*) L=10.0; (*ширина ями*) n=500; (*кількість вузлів\
дискретизації*)
(*Визначаємо функцію потенціалу*) potential[x_]:=0.5x^2+0.05x^4;
(*Тут можна змінити потенціал*) (*Визначаємо масштабний
коефіцієнт*) hbar2Div2m=((Sqrt[2]*ab*Sqrt[Ry])^2)/(2*m);

(*Крок дискретизації*)
dx=2*L/(n+1);

(*Ініціалізація матриці*)
A=Table[0,{n},{n}];

(*Заповнення матриці методом кінцевих різниць*)
Do[If[i>1,A[[i,i-1]]=-hbar2Div2m/dx^2]; (*Ліва сусідня
точка*) If[i<n,A[[i,i+1]]=-hbar2Div2m/dx^2]; (*Права сусідня
точка*) A[[i,i]]=2*hbar2Div2m/dx^2+potential[-L+i*dx]; (*Головна діагональ
(потенціал x^2)*),{i,1,n}];

(*Додавання потенціалу стінок до матриці*)
A[[1,1]]+=U0; (*Потенціал на лівій стінці*) A[[n,n]]+=U0; (*Потенціал на
правій стінці*) (*Розв'язок на власні значення*) {Evals,wfn}=Eigensystem[A];

(*Сортуємо власні значення та відповідні власні вектори*)
sortedIndices=Ordering[Evals];
Evals=Evals[[sortedIndices]];
wfn=wfn[[All,sortedIndices]];

(*Масштабування координати (зміщене на центр ями*)
x=Table[-L+i*dx,{i,1,n}];

(*Вибираємо рівні нижче 500*)
levelsBelow500=Select[Transpose[{Evals,wfn}],First[#]<U0&];
EvalsBelow500=levelsBelow500[[All,1]];
wfnBelow500=levelsBelow500[[All,2]];

(*Побудова графіка енергетичних рівнів з хвильовими функціями та
потенціалом*)
Show[Table[Module[{PsiSquared,clippedData},PsiSquared=10*wfnBelow500[[i]]^2
/Max[wfnBelow500[[i]]^2]+EvalsBelow500[[i]]; (*Масштабування*) clippedData=Tr
anspose[{x,PsiSquared}];
    clippedData=Select[clippedData,#[[2]]>=potential[#[[1]]]&];
    (*Видаляємо точки нижче потенціалу*) ListLinePlot[clippedData,PlotStyle-
>{Thick,ColorData["Rainbow"][i/Length[EvalsBelow500]]},PlotRange-
>All],{i,1,Length[EvalsBelow500]}],ListLinePlot[Transpose[{x,potential[x]}]
,PlotStyle->{Thick,Black}], (*Графік потенціалу*) PlotRange->All,Frame-
>True,FrameLabel->{"z","Energy and |Ψ(z)|^2"},PlotLabel->"Ψn(z)|b2 (рівні
нижче 500)",GridLines->Automatic,ImageSize->Large,Epilog->{Dashed,Line[{-
L,Min[EvalsBelow500]},{-
L,Max[EvalsBelow500]}],Dashed,Line[{L,Min[EvalsBelow500]},{L,Max[EvalsBel
ow500]}]}]}];

Manipulate[Module[{ab=0.0529,Ry=13605,U0=0,hbar2Div2m,dx,A,n=500,Evals,wfn,
sortedIndices,x},
    hbar2Div2m=((Sqrt[2]*ab*Sqrt[Ry])^2)/(2*m);
    (*Крок*) dx=2*L/(n+1);
    (*Матриця*) A=Table[0,{n},{n}];
    Do[If[i>1,A[[i,i-1]]=-hbar2Div2m/dx^2];

```

```

If[i<n,A[[i,i+1]]=-hbar2Div2m/dx^2];
A[[i,i]]=2*hbar2Div2m/dx^2;,{i,1,n}];
(*Власні значення та вектори*){Evals,wfn}=Eigensystem[A];
sortedIndices=Ordering[Evals];
Evals=Evals[[sortedIndices]];
wfn=wfn[[All,sortedIndices]];
(*Координата*)x=Table[-L+i*dx,{i,1,n}];
(*Побудова графіків*)Grid[{{Show[Table[Module[{PsiSquared},
PsiSquared=10*wfn[[All,i]]^2/Max[wfn[[All,i]]^2]+Evals[[i]];
ListLinePlot[Transpose[{x,PsiSquared}],
PlotStyle->{Thick,ColorData["Rainbow"][i/5]},PlotRange-
>All]],{i,1,levels}]],
PlotRange->All,Frame->True,FrameLabel->{"z","En (meV)"},PlotLabel-
>"|Ψn(z)|",GridLines->Automatic,ImageSize->Large,Epilog->{Dashed,Line[{{-
L,Min[Evals]},{L,Max[Evals]}]},Dashed,Line[{{L,Min[Evals]},{L,Max[Evals]}]}]},(*Список
енергій*)Column[Table[Style[Row[{"E",Subscript["",i],"="
,NumberForm[Evals[[i]],{6,4}], " meV"}],FontSize->14,{i,1,levels}]]}],
{{m,0.186},0.05,1,Appearance->"Labeled"},{{L,10.0},1,20,Appearance-
>"Labeled"},
{{levels,5,"Кількість рівнів"},1,10,1,Appearance-
>"Labeled"},ControlPlacement->Top]

```

```

Manipulate[Module[{ab=0.0529,Ry=13605,n=500,hbar2Div2m,dx,A,Evals,wfn,sorte
dIndices,x,levelsBelowU0,EvalsBelowU0,wfnBelowU0,potential},(*Потенціал*)po
tential[x_]:=a x^2-b x^4;

```

```

(*Масштабний коефіцієнт*)hbar2Div2m=((Sqrt[2]*ab*Sqrt[Ry])^2)/(2*m);
(*Крок дискретизації*)dx=2*L/(n+1);
(*Матриця*)A=Table[0,{n},{n}];
Do[If[i>1,A[[i,i-1]]=-hbar2Div2m/dx^2];
If[i<n,A[[i,i+1]]=-hbar2Div2m/dx^2];
A[[i,i]]=2*hbar2Div2m/dx^2+potential[-L+i*dx];,{i,1,n}];
(*Додаємо потенціал стінок*)A[[1,1]]+=U0;
A[[n,n]]+=U0;
(*Розв'язання*){Evals,wfn}=Eigensystem[A];
sortedIndices=Ordering[Evals];
Evals=Evals[[sortedIndices]];
wfn=wfn[[All,sortedIndices]];
x=Table[-L+i*dx,{i,1,n}];
(*Вибираємо рівні нижче
U0*)levelsBelowU0=Select[Transpose[{Evals,wfn}],First[#]<U0&];
EvalsBelowU0=EvalsBelowU0[[All,1]];
wfnBelowU0=wfnBelowU0[[All,2]];

(*Побудова*)Grid[{{Show[Table[Module[{PsiSquared,clippedData},PsiSquared=10
*wfnBelowU0[[i]]^2/Max[wfnBelowU0[[i]]^2]+EvalsBelowU0[[i]];
clippedData=Transpose[{x,PsiSquared}];
clippedData=Select[clippedData,#[[2]]>=potential[#[[1]]]&];
ListLinePlot[clippedData,PlotStyle-
>{Thick,ColorData["Rainbow"][i/Length[EvalsBelowU0]]},PlotRange-
>All]],{i,1,Min[Length[EvalsBelowU0],maxLevels]}],ListLinePlot[Transpose[{x,
potential/@x}],PlotStyle->{Thick,Black}],PlotRange->All,Frame-
>True,FrameLabel->{"z","En (meV)"},PlotLabel->"|Ψn(z)| (ангармонічний
осцилятор)",GridLines->Automatic,ImageSize->Large,Epilog->{Dashed,Line[{{-
L,Min[EvalsBelowU0]},{L,Max[EvalsBelowU0]}]},Dashed,Line[{{L,Min[EvalsBelowU0]},{L,Max[EvalsBelow
U0]}]}]}]},(*Вивід числових значень

```

```
енергій*)Column[Table[Style[Row[{"E",Subscript["",i]," =
",NumberForm[EvalsBelowU0[[i]],{6,4}]," meV"}],FontSize-
>14],{i,1,Min[Length[EvalsBelowU0],maxLevels]}]]]],(*Слайдери*){{m,0.186},
0.05,1,Appearance->"Labeled"},{{L,10.0},1,20,Appearance-
>"Labeled"},{{U0,500},50,1000,Appearance-
>"Labeled"},{{a,0.5},0,2,Appearance->"Labeled"},{{b,0.05},0,0.2,Appearance-
>"Labeled"},{{maxLevels,5,"Кількість рівнів"},1,10,1,Appearance-
>"Labeled"},ControlPlacement->Top]
```

```
Manipulate[Module[{ab=0.0529,Ry=13605,n=500,hbar2Div2m,dx,A,Evals,wfn,sortedIndices,x,levelsBelowU0,EvalsBelowU0,wfnBelowU0,potential},(*Потенціал гармонічного типу*)potential[x_]:=a x^2;
(*Масштабний коефіцієнт*)hbar2Div2m=((Sqrt[2]*ab*Sqrt[Ry])^2)/(2*m);
(*Крок дискретизації*)dx=2*L/(n+1);
(*Матриця*)A=Table[0,{n},{n}];
Do[If[i>1,A[[i,i-1]]=-hbar2Div2m/dx^2];
If[i<n,A[[i,i+1]]=-hbar2Div2m/dx^2];
A[[i,i]]=2*hbar2Div2m/dx^2+potential[-L+i*dx];,{i,1,n}];
(*Додаємо потенціал стінок*)A[[1,1]]+=U0;
A[[n,n]]+=U0;
(*Розв'язок*){Evals,wfn}=Eigensystem[A];
sortedIndices=Ordering[Evals];
Evals=Evals[[sortedIndices]];
wfn=wfn[[All,sortedIndices]];
x=Table[-L+i*dx,{i,1,n}];
(*Вибір рівнів нижче
U0*)levelsBelowU0=Select[Transpose[{Evals,wfn}],First[#]<U0&];
EvalsBelowU0=levelsBelowU0[[All,1]];
wfnBelowU0=levelsBelowU0[[All,2]];

(*Вивід*)Grid[{{Show[Table[Module[{PsiSquared,clippedData},PsiSquared=10*wfnBelowU0[[i]]^2/Max[wfnBelowU0[[i]]^2]+EvalsBelowU0[[i]];
clippedData=Transpose[{x,PsiSquared}];
clippedData=Select[clippedData,#[[2]]>=potential[#[[1]]]&];
ListLinePlot[clippedData,PlotStyle-
>{Thick,ColorData["Rainbow"][i/Length[EvalsBelowU0]]},PlotRange-
>All],{i,1,Min[Length[EvalsBelowU0],maxLevels]}],ListLinePlot[Transpose[{x,
potential/@x}],PlotStyle->{Thick,Black}],PlotRange->All,Frame-
>True,FrameLabel->{"z","En (meV)"},PlotLabel->"|Ψn(z)|I (гармонічний
осцилятор)",GridLines->Automatic,ImageSize->Large,Epilog->{Dashed,Line[{-L,Min[EvalsBelowU0]},{-L,Max[EvalsBelowU0]}],Dashed,Line[{L,Min[EvalsBelowU0]},{L,Max[EvalsBelowU0]}]}]],(*Вивід чисел*)Column[Table[Style[Row[{"E",Subscript["",i]," =
",NumberForm[EvalsBelowU0[[i]],{6,4}]," meV"}],FontSize-
>14],{i,1,Min[Length[EvalsBelowU0],maxLevels]}]]]],(*Слайдери*){{m,0.186},
0.05,1,Appearance->"Labeled"},{{L,10.0},1,20,Appearance-
>"Labeled"},{{U0,500},50,1000,Appearance-
>"Labeled"},{{a,0.5},0.01,2,Appearance->"Labeled"},{{maxLevels,5,"Кількість
рівнів"},1,10,1,Appearance->"Labeled"},ControlPlacement->Top]
```

```
Manipulate[Module[{ab=0.0529,Ry=13605,n=500,hbar2Div2m,dx,A,Evals,wfn,sortedIndices,x,levelsBelowU0,EvalsBelowU0,wfnBelowU0,potential},(*Пешля-Теллера потенціал*)potential[x_]:=-V0/Cosh[α x]^2;
(*Масштабний коефіцієнт*)hbar2Div2m=((Sqrt[2]*ab*Sqrt[Ry])^2)/(2*m);
(*Крок дискретизації*)dx=2*L/(n+1);
(*Матриця*)A=Table[0,{n},{n}];
```

```

Do[If[i>1,A[[i,i-1]]=-hbar2Div2m/dx^2];
  If[i<n,A[[i,i+1]]=-hbar2Div2m/dx^2];
  A[[i,i]]=2*hbar2Div2m/dx^2+potential[-L+i*dx];,{i,1,n}];
(*Додаємо потенціал стінок*)A[[1,1]]+=U0;
A[[n,n]]+=U0;
(*Розв'язання*){Evals,wfn}=Eigensystem[A];
sortedIndices=Ordering[Evals];
Evals=Evals[[sortedIndices]];
wfn=wfn[[All,sortedIndices]];
x=Table[-L+i*dx,{i,1,n}];
(*Вибір рівнів нижче
U0*)levelsBelowU0=Select[Transpose[{Evals,wfn}],First[#]<U0&];
EvalsBelowU0=levelsBelowU0[[All,1]];
wfnBelowU0=levelsBelowU0[[All,2]];

(*Побудова*)Grid[{{Show[Table[Module[{PsiSquared,clippedData},PsiSquared=10
*wfnBelowU0[[i]]^2/Max[wfnBelowU0[[i]]^2]+EvalsBelowU0[[i]];
  clippedData=Transpose[{x,PsiSquared}]];
  clippedData=Select[clippedData,#[[2]]>potential[#[[1]]]&];
  ListLinePlot[clippedData,PlotStyle-
>{Thick,ColorData["Rainbow"] [i/Length[EvalsBelowU0]]},PlotRange-
>All]],{i,1,Min[Length[EvalsBelowU0],maxLevels]}],ListLinePlot[Transpose[{x,
potential/@x}],PlotStyle->{Thick,Black}],PlotRange->All,Frame-
>True,FrameLabel->{"z","E (meV)"},PlotLabel->"Ψn(z)|I (потенціал Пешля-
Теллера)",GridLines->Automatic,ImageSize->Large,Epilog->{Dashed,Line[{{-
L,Min[EvalsBelowU0]},{-
L,Max[EvalsBelowU0]}],Dashed,Line[{{L,Min[EvalsBelowU0]},{L,Max[EvalsBelow
U0]}]}]}],(*Вивід чисел*)Column[Table[Style[Row[{"E",Subscript["",i]," =
",NumberForm[EvalsBelowU0[[i]],{6,4]]," meV"}],FontSize-
>14]],{i,1,Min[Length[EvalsBelowU0],maxLevels]}]}],(*Слайдери*){{m,0.186},
0.05,1,Appearance->"Labeled"},{{L,10.0},1,20,Appearance-
>"Labeled"},{{U0,500},50,1000,Appearance-
>"Labeled"},{{V0,500},50,1000,Appearance-
>"Labeled"},{{α,0.2},0.05,1,Appearance->"Labeled"},{{maxLevels,5,"Кількість
рівнів"},1,10,1,Appearance->"Labeled"},ControlPlacement->Top]

Manipulate[Module[{ab=0.0529,Ry=13605,n=500,hbar2Div2m,dx,A,Evals,wfn,sorte
dIndices,x,levelsBelowU0,EvalsBelowU0,wfnBelowU0,potential},(*Модифікований
Пешля-Теллера потенціал*)potential[x_]:=-V0/Cosh[α x]^2+V1*Tanh[α x]^2;
(*Масштабний коефіцієнт*)hbar2Div2m=((Sqrt[2]*ab*Sqrt[Ry])^2)/(2*m);
(*Крок дискретизації*)dx=2*L/(n+1);
(*Матриця*)A=Table[0,{n},{n}];
Do[If[i>1,A[[i,i-1]]=-hbar2Div2m/dx^2];
  If[i<n,A[[i,i+1]]=-hbar2Div2m/dx^2];
  A[[i,i]]=2*hbar2Div2m/dx^2+potential[-L+i*dx];,{i,1,n}];
(*Додаємо потенціал стінок*)A[[1,1]]+=U0;
A[[n,n]]+=U0;
(*Розв'язання*){Evals,wfn}=Eigensystem[A];
sortedIndices=Ordering[Evals];
Evals=Evals[[sortedIndices]];
wfn=wfn[[All,sortedIndices]];
x=Table[-L+i*dx,{i,1,n}];
(*Вибір рівнів нижче
U0*)levelsBelowU0=Select[Transpose[{Evals,wfn}],First[#]<U0&];
EvalsBelowU0=levelsBelowU0[[All,1]];
wfnBelowU0=levelsBelowU0[[All,2]];
(*Побудова

```

```

графіка*)Grid[{{Show[Table[Module[{PsiSquared,clippedData},PsiSquared=10*wfnBelowU0[[i]]^2/Max[wfnBelowU0[[i]]^2]+EvalsBelowU0[[i]];
    clippedData=Transpose[{x,PsiSquared}];
    clippedData=Select[clippedData,#[[2]]>=potential[#[[1]]]&];
    ListLinePlot[clippedData,PlotStyle-
>{Thick,ColorData["Rainbow"][i/Length[EvalsBelowU0]]},PlotRange-
>All]],{i,1,Min[Length[EvalsBelowU0],maxLevels]}],ListLinePlot[Transpose[{x,
potential/@x}],PlotStyle->{Thick,Black}],PlotRange->All,Frame-
>True,FrameLabel->{"z","E (meV)"},PlotLabel->"Ψn(z)|I (модифікований
потенціал Пешля\ [Dash]Теллера)",GridLines->Automatic,ImageSize-
>Large,Epilog->{Dashed,Line[{{-L,Min[EvalsBelowU0]},{-
L,Max[EvalsBelowU0]}],Dashed,Line[{{L,Min[EvalsBelowU0]},{L,Max[EvalsBelow
U0]}]}],(*Вивід числових
значень*)Column[Table[Style[Row[{"E",Subscript["",i]," =
",NumberForm[EvalsBelowU0[[i]],{6,4]]," meV"}],FontSize-
>14],{i,1,Min[Length[EvalsBelowU0],maxLevels]}]}],(*Слайдери*){{m,0.186},
0.05,1,Appearance->"Labeled"},{{L,10.0},1,20,Appearance-
>"Labeled"},{{U0,500},50,1000,Appearance-
>"Labeled"},{{V0,500},50,1000,Appearance->"Labeled"},{{V1,0},-
200,500,Appearance->"Labeled"},{{α,0.2},0.05,1,Appearance-
>"Labeled"},{{maxLevels,5,"Кількість рівнів"},1,10,1,Appearance-
>"Labeled"},ControlPlacement->Top]

Manipulate[Module[{ab=0.0529,Ry=13605,n=500,hbar2Div2m,dx,A,Evals,wfn,sortedIndices,x,levelsBelowU0,EvalsBelowU0,wfnBelowU0,potential},(*Потенціал
Морзе*)potential[x_]:=De*(1-Exp[-a*(x-x0)])^2;
(*Масштабний коефіцієнт*)hbar2Div2m=((Sqrt[2]*ab*Sqrt[Ry])^2)/(2*m);
(*Крок дискретизації*)dx=2*L/(n+1);
(*Матриця*)A=Table[0,{n},{n}];
Do[If[i>1,A[[i,i-1]]=-hbar2Div2m/dx^2];
If[i<n,A[[i,i+1]]=-hbar2Div2m/dx^2];
A[[i,i]]=2*hbar2Div2m/dx^2+potential[-L+i*dx];,{i,1,n}];
(*Додаємо потенціал стінок*)A[[1,1]]+=U0;
A[[n,n]]+=U0;
(*Розв'язання*){Evals,wfn}=Eigensystem[A];
sortedIndices=Ordering[Evals];
Evals=Evals[[sortedIndices]];
wfn=wfn[[All,sortedIndices]];
x=Table[-L+i*dx,{i,1,n}];
(*Вибір рівнів нижче
U0*)levelsBelowU0=Select[Transpose[{Evals,wfn}],First[#]<U0&];
EvalsBelowU0=levelsBelowU0[[All,1]];
wfnBelowU0=levelsBelowU0[[All,2]];
(*Побудова
графіка*)Grid[{{Show[Table[Module[{PsiSquared,clippedData},PsiSquared=10*wfnBelowU0[[i]]^2/Max[wfnBelowU0[[i]]^2]+EvalsBelowU0[[i]];
    clippedData=Transpose[{x,PsiSquared}];
    clippedData=Select[clippedData,#[[2]]>=potential[#[[1]]]&];
    ListLinePlot[clippedData,PlotStyle-
>{Thick,ColorData["Rainbow"][i/Length[EvalsBelowU0]]},PlotRange-
>All]],{i,1,Min[Length[EvalsBelowU0],maxLevels]}],ListLinePlot[Transpose[{x,
potential/@x}],PlotStyle->{Thick,Black}],PlotRange->All,Frame-
>True,FrameLabel->{"z","E (meV)"},PlotLabel->"Ψn(z)|I (потенціал
Морзе)",GridLines->Automatic,ImageSize->Large,Epilog->{Dashed,Line[{{-
L,Min[EvalsBelowU0]},{-
L,Max[EvalsBelowU0]}],Dashed,Line[{{L,Min[EvalsBelowU0]},{L,Max[EvalsBelow
U0]}]}],(*Вивід числових

```

```

значень*)Column[Table[Style[Row[{"E",Subscript["",i]," =
",NumberForm[EvalsBelowU0[[i]],{6,4}]," meV"}],FontSize-
>14],{i,1,Min[Length[EvalsBelowU0],maxLevels]}]]]],(*Слайдери*){{m,0.186},
0.05,1,Appearance->"Labeled"},{{L,10.0},1,20,Appearance-
>"Labeled"},{{U0,500},50,1000,Appearance-
>"Labeled"},{{De,500},50,1000,Appearance-
>"Labeled"},{{a,0.3},0.01,1,Appearance->"Labeled"},{{x0,0.0},-
5,5,Appearance->"Labeled"},{{maxLevels,5,"Кількість
рівнів"},1,10,1,Appearance->"Labeled"},ControlPlacement->Top]

Manipulate[Module[{U,ab=1,Ry=1,hbar2Div2m,n=500,dx,xgrid,A,Evals,wfn,sorted
Indices,EvalsNeg,wfnNeg,Uvals},(*Потенціал Леннарда-Джонса*)U[r_]:=4 ε
((σ/r)^12-(σ/r)^6);
(*Унікаємо r=0, щоб уникнути ∞*)dx=(L-0.2)/(n+1);(*Від 0.2 до
L*)xgrid=Table[0.2+i*dx,{i,0,n-1}];
(*Кінетичний множник*)hbar2Div2m=((Sqrt[2]*ab*Sqrt[Ry])^2)/(2*m);
(*Створення матриці*)A=Table[0,{n},{n}];
Do[If[i>1,A[[i,i-1]]=-hbar2Div2m/dx^2];
If[i<n,A[[i,i+1]]=-hbar2Div2m/dx^2];
A[[i,i]]=2 hbar2Div2m/dx^2+U[xgrid[[i]]];,{i,1,n}];
(*Власні значення та функції*){Evals,wfn}=Eigensystem[A];
sortedIndices=Ordering[Evals];
Evals=Evals[[sortedIndices]];
wfn=wfn[[All,sortedIndices]];
(*Відбір зв'язаних станів*)EvalsNeg=Select[Evals,#<0&];

wfnNeg=wfn[[All,Take[Position[Evals,_?(<0&)]][[All,1]],Length[EvalsNeg]]];
(*Потенціал для графіка*)Uvals=U/@xgrid;
Grid[{{Show[Table[Module[{normWfn,Psi2},normWfn=wfnNeg[[All,i]]/Sqrt[dx
Total[wfnNeg[[All,i]]^2]];
Psi2=10 normWfn^2+EvalsNeg[[i]];
ListLinePlot[Transpose[{xgrid,Psi2}],PlotStyle-
>{Thick,ColorData["Rainbow"][i/Length[EvalsNeg]]}],{i,1,Min[levels,Length[
EvalsNeg]]}],ListLinePlot[Transpose[{xgrid,Uvals}],PlotStyle-
>{Thick,Black}],Frame->True,PlotRange->{{Min[xgrid],Max[xgrid]},{-
100,100}},FrameLabel->{"r","Energy (meV)"},PlotLabel->"Зв'язані стани в
потенціалі Леннарда-Джонса",ImageSize->Large],(*Список
енергій*)Column[Table[Style[Row[{"E",Subscript["",i]," =
",NumberForm[EvalsNeg[[i]],{6,4}]," meV"}],FontSize-
>14],{i,1,Min[levels,Length[EvalsNeg]]}]]]],(*Керування
параметрами*){{ε,100,"Глибина потенціалу (ε) "},10,200,Appearance-
>"Labeled"},{{σ,1,"Розмір (σ) "},0.5,2,Appearance-
>"Labeled"},{{L,3.0,"Ширина ями (L) "},2.0,5.0,Appearance-
>"Labeled"},{{m,1,"Маса частинки"},0.05,2.0,Appearance-
>"Labeled"},{{levels,5,"Кількість рівнів"},1,10,1,Appearance-
>"Labeled"},ControlPlacement->Top]

```

ДОДАТОК Б

Тези доповіді на конференції

Міністерство освіти і науки України
 Тернопільський національний технічний університет
 імені Івана Пулюя
 Маріборський університет (Словенія)
 Технічний університет в Кошице (Словаччина)
 Каунаський технологічний університет (Литва)
 Львівський національний університет
 імені Івана Франка,
 Гірничо-металургійна академія ім. Станіслава Сташиця (Польща)
 Луцький національний технічний університет,
 Чернівецький національний університет
 імені Юрія Федьковича,
 Вроцлавський економічний університет (Польща)
 Університет технологій та економіки
 імені Хелени Ходковської (Польща)
 Донбаська державна машинобудівна академія



Студентське наукове
 товариство



VIII МІЖНАРОДНА

студентська науково - технічна конференція

**"ПРИРОДНИЧІ ТА ГУМАНІТАРНІ
 НАУКИ.**

АКТУАЛЬНІ ПИТАННЯ"

24-25 квітня 2025 р.

(збірник тез конференції)

Тернопіль 2025

VIII Міжнародна студентська науково - технічна конференція
"ПРИРОДНИЧІ ТА ГУМАНІТАРНІ НАУКИ. АКТУАЛЬНІ ПИТАННЯ"

Хемій С. ПРОГРАМНА СИСТЕМА ДЛЯ МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ СПЕКТРАЛЬНИХ ХАРАКТЕРИСТИК ЕЛЕКТРОННИХ СТАНІВ З РІЗНИМИ ЕФЕКТИВНИМИ ПОТЕНЦІАЛАМИ	219
Холод Ю. ПРОГРАМНИЙ МОДУЛЬ ДЛЯ ПЕРЕГЛЯДУ ЗОБРАЖЕНЬ	220
Хромов В. РОЗРОБКА RESTFUL API ДЛЯ ВЕБ-ДОДАТКІВ З ВИКОРИСТАННЯМ NODE.JS ТА EXPRESS	222
Чендей Б. РОЗРОБКА ВЕБ-ПЛАТФОРМИ ДЛЯ НАДАННЯ ПОСЛУГ РЕПЕТИТОРСТВА З ПРОГРАМУВАННЯ З ВИКОРИСТАННЯМ REACT.JS	223
Shtokalo A. VOICE-DRIVEN CODING ENVIRONMENTS: ENHANCING ACCESSIBILITY AND EFFICIENCY	224
Юрчак В. ПРОЦЕС РОЗРОБКИ МЕТАМОДЕЛІ ОБ'ЄКТІВ VI-СИСТЕМИ	226
Недошитко Л., Яворський Б ШТУЧНИЙ ІНТЕЛЕКТ У МЕДИЦИНІ: РЕВОЛЮЦІЯ В ОХОРОНІ ЗДОРОВ'Я	228
Якуб'як Ю. РЕАЛІЗАЦІЯ АЛГОРИТМУ ОПТИМІЗАЦІЇ ПРОЦЕСУ ЗБОРУ ДАНИХ ПОТЕНЦІЙНИХ КЛІЄНТІВ	229
Горват М. ЧИСЕЛЬНЕ РОЗВ'ЯЗАННЯ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ ПРИ ПРОВЕДЕННІ РОЗРАХУНКІВ НА ЖОРСТКІСТЬ	232
Драбик І. ПРО ЧИСЛЕНЕ РОЗВ'ЯЗАННЯ ЗВИЧАЙНИХ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ	234
Мартинюк С. ЛОГІЧНИЙ ПРИЙОМ ПОРІВНЯННЯ В МАТЕМАТИЧНОМУ АНАЛІЗІ	236
Островський О. РОЗВ'ЯЗОК ДИФЕРЕНЦІАЛЬНОГО РІВНЯННЯ ПРОХОДЖЕННЯ СТРУМУ В КАБЕЛІ	238
Дубиняк Т., Цюпа С., Микулик П. ЗАЛЕЖНІСТЬ ІНДУКТИВНОСТІ КОТУШКИ ВІД ВЕЛИЧИНИ ПОВІТРЯНОГО ЗАЗОРУ ПРИ КОНТРОЛІ ШОРСТКОСТІ	240

VIII Міжнародна студентська науково - технічна конференція
"ПРИРОДНИЧІ ТА ГУМАНІТАРНІ НАУКИ. АКТУАЛЬНІ ПИТАННЯ"

УДК 004.942

Хемій С. – ст. гр. СП-43

Тернопільський національний технічний університет імені Івана Пулюя

ПРОГРАМНА СИСТЕМА ДЛЯ МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ СПЕКТРАЛЬНИХ ХАРАКТЕРИСТИК ЕЛЕКТРОННИХ СТАНІВ З РІЗНИМИ ЕФЕКТИВНИМИ ПОТЕНЦІАЛАМИ

Науковий керівник: к.ф.-м.н., доцент Бойко І. В.

Khemii S.

Ternopil Ivan Puluj National Technical University

SOFTWARE SYSTEM FOR MATHEMATICAL MODELLING OF SPECTRAL CHARACTERISTICS OF ELECTRONIC STATES WITH DIFFERENT EFFECTIVE POTENTIALS

Supervisor: Ph.D. in Physics and Mathematics, associate professor Boyko I.V.

Ключові слова: спектральні характеристики, математична модель, ефективний потенціал
Keywords: spectral characteristics, mathematical model, effective potential

Однією з основних задач сучасної електроніки низьковимірних структур, що широко застосовуються у низці інженерних та природничих дисциплін, є створення точного математичного опису спектральних характеристик для кожного окремого випадку конкретно взятих параметрів. Підхід до розв'язання даних проблем виходить за межі класичних підходів та усе частіше вимагає застосування підходів, що належать до сфери інформаційних технологій, а саме інженерії програмного забезпечення. У розробці відповідного програмного забезпечення також помітно відсутність універсальних підходів до чисельного опису електронних станів та енергій активації у структурах зі складною геометрією потенціального поля та просторового конфайнменту. Крім того, бракує узагальнених алгоритмів, здатних працювати з такими математичними моделями в загальному випадку.

До розробки такого програмного забезпечення необхідно підійти комплексно, тобто розбивши проектувану систему на компоненти, а також виконуючи процес моделювання у кілька етапів. На першому етапі слід виконати побудову математичних моделей ефективних потенціалів. На другому етапі варто побудувати різницеві схеми математичних моделей, програмна реалізація яких забезпечує математичне моделювання спектру, локалізації та розподілу квазічастинкових станів шляхом зміни вхідних параметрів. Останнім етапом є розробка гнучкої архітектури програмної системи, що дозволяє реалізовувати математичну модель із довільним локальним потенціалом, а також виконувати розрахунки та здійснювати візуалізацію параметрів моделі та її характеристик на діапазоні неперервної зміни вхідних даних.

У даній роботі згадані проблеми вирішено шляхом розробки математичних моделей для різних ефективних потенціалів, що характеризують локальні квантові ями. Розроблених математичні моделі ефективних потенціалів реалізовано як спектральні проблеми для стаціонарного рівняння Шредінгера на основі побудованих сіткових різницевоїх схем. Розроблено архітектуру та складові компоненти програмної системи, призначеної для математичного моделювання спектральних характеристик електронних станів з різними ефективними потенціалами.

ДОДАТОК В

Диск з роботою