

А. Я. БОМБА, І. П. МОРОЗ, З. С. ЛІСТЬЄВ

ВИКОРИСТАННЯ СУЧАСНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ В СИСТЕМАХ АВТОМАТИЗОВАНОГО РОЗВ'ЯЗАННЯ СИНГУЛЯРНО ЗБУРЕНИХ ЗАДАЧ

Представлено архітектуру програмної системи для автоматизованого розв'язання сингулярно збурених задач у напівпровідниковій електроніці, розроблену на основі принципів предметно-орієнтованого проєктування (DDD – Domain-Driven Design). Технологія DDD була застосована до математичної моделі технічної системи «плазмовий діод» з метою демонстрації її потенційних можливостей у контексті складних інженерних задач. Розширено коло задач програмного проєктування за допомогою методів DDD, що дозволило більш точно моделювати складні системи та процеси. Особливістю представленої архітектури програмної системи є інтеграція штучної нейронної мережі, яка призначена для розпізнавання типів рекурентних послідовностей підзадач. Ці підзадачі утворюються в результаті декомпозиції базової задачі за методом прилежових поправок теорії збурень. Нейронна мережа аналізує отримані підзадачі та класифікує їх за відомими типами, що дозволяє автоматизувати вибір найбільш ефективних методів їх розв'язання та оптимізувати обчислювальні ресурси. Застосування DDD у контексті розробки програмної системи для розв'язання сингулярно збурених задач дозволило досягти глибокого розуміння предметної області та відобразити її концепції у програмному коді. Це сприяє створенню гнучких та масштабованих систем, які легко адаптуються до змін вимог та умов експлуатації. Математична модель «плазмового діода» слугувала тематичним дослідженням, яке підтвердило ефективність застосованого підходу. Розроблена система відкриває нові перспективи в автоматизації та оптимізації процесів моделювання та аналізу в галузі напівпровідникової електроніки. Застосування DDD та інтеграція інтелектуальних компонентів, таких як штучні нейронні мережі, сприяють розширенню можливостей програмного забезпечення та підвищенню його ефективності. Подальший розвиток системи передбачає розширення бази даних підзадач та методів їх розв'язання, вдосконалення алгоритмів навчання нейронної мережі, а також адаптацію архітектури для інших типів технічних систем та задач. Це сприятиме більш широкому застосуванню розробленого підходу в різних галузях науки та техніки.

Ключові слова: сингулярно збурені задачі, предметно-орієнтоване проєктування, штучна нейронна мережа, автоматизоване моделювання, метод прилежових поправок, символічні обчислення.

A. YA. BOMBA, I. P. MOROZ, Z. S. LISTIEV

USE OF MODERN INFORMATION TECHNOLOGIES IN SYSTEMS FOR AUTOMATED SOLUTION OF SINGULARLY PERTURBED PROBLEMS

The paper presents the architecture of a software system for automated solution of singularly perturbed problems of semiconductor electronics, developed based on the principles of domain-driven design (DDD). The DDD technology was applied to the mathematical model of the technical system "plasma diode" to demonstrate its potential in the context of complex engineering problems. The range of software design problems was expanded using DDD methods, which made it possible to model complex systems and processes more accurately. A special feature of the presented architecture of the software system is the integration of an artificial neural network designed to recognize the types of recurrent sequences of subproblems. These subproblems are formed as a result of decomposition of the basic problem using the method of boundary corrections of perturbation theory. The neural network analyzes the obtained subproblems and classifies them by known types, which allows automating the selection of the most effective methods for their solution and optimizing computing resources. The use of DDD in developing a software system for solving singularly perturbed problems allowed us to deeply understand the subject area and reflect its concepts in the program code. This contributes to creating flexible and scalable systems that can easily adapt to changing requirements and operating conditions. The "plasma diode" mathematical model served as a case study confirming the approach's effectiveness. The developed system opens up new prospects for automating and optimizing modeling and analysis processes in semiconductor electronics. The use of DDD and the integration of intelligent components, such as artificial neural networks, contribute to the expansion of software capabilities and increased efficiency. Further system development involves expanding the database of subtasks and methods for solving them, improving neural network training algorithms, and adapting the architecture to other technical systems and problems. This will contribute to a wider application of the developed approach in various fields of science and technology.

Key words: singularly perturbed problems, Domain-Driven Design, artificial neural network, automated modeling, method of boundary layer corrections, symbolic computations.

Вступ. Сингулярно збурені задачі теорії диференціальних рівнянь – це клас задач, які, зокрема, містять малі параметри при старших похідних чи частині старших похідних (для рівнянь у частинних похідних). Сингулярні збурення (доданки, що містять малий параметр) суттєво впливають на поведінку розв'язків цих задач, зокрема, на їх структуру та асимптотичні властивості. Відповідні постановки задач, які зазвичай є нелінійними, виникають в різноманітних прикладних задачах фізики (зокрема електроніки), хімії, біології і використовуються для опису математичних моделей процесів дифузії, фільтрації рідин в пористих середовищах, тепло- і масо-переносу та багатьох інших областях [1 – 3].

Проблема розв'язання сингулярно збурених задач обумовлена тим, що, зокрема, стандартні чисельні методи часто виявляються неефективними при спробі їх вирішення, оскільки не можуть адекватно відтворити різкі перепади розв'язку без значного збільшення обчислювальних витрат. Для вивчення сингулярно збурених нелінійних задач для диференціальних рівнянь в частинних похідних розроблені спеціалізовані асимптотичні методи: метод ВКБ [5], метод Люстернака – Вішика [6] та, зокрема, метод прилежових поправок [7 – 10], які дозволяють отримати наближені розв'язки задач, виражені степеневими рядами за малими параметрами. Відповідно, процедура розв'язання сингулярно збурених задач характеризується високим рівнем складності.

Слід відзначити, що сучасні прикладні задачі (у тому числі і сингулярно збурені), як правило, мають комплексну постановку, тобто в них поєднуються математичні моделі низки фізичних процесів, що протікають у технічній системі. Наприклад, у технічній системі «плазмовий діод» є необхідність контролю процесу формування електронно-діркової плазми та процесу нагріву плазми [9] тощо. Ця обставина додатково підвищує рівень склад-

ності проблем, що містять особливості (сингулярності).

Логічним рішенням, що знижує рівень складності початкової задачі, є проведення її *декомпозиції*. Цей підхід природно реалізується в низці асимптотичних методів, зокрема в методі примезових поправок [7].

Поряд із цим поява перспективних сучасних інформаційних технологій, таких, як *розпаралелювання*, використання віддалених серверів для виконання обчислень, використання символьних перетворень, штучного інтелекту під час обчислювального процесу, разом із збільшеними ресурсами комп'ютерних систем надають можливість підійти до розробки алгоритмів вирішення сингулярно збурених задач на новому якісному рівні. Такого типу обчислювальні алгоритми традиційно складають основу ядра *систем автоматизованого проектування (САПР)* [11 – 13]. Зауважимо і те, що математичні моделі різноманітних технічних систем, зокрема електронних напівпровідникових пристроїв, постійно удосконалюються, і, відповідно, виникає необхідність у поновленні ядра алгоритмів САПР.

З огляду на специфіку задач, що вирішуються на сучасному етапі розвитку математичного моделювання, та складність процесів моделювання, виникає необхідність залучення певних підходів до організації обчислювальної програмної системи і відповідних обчислювальних процесів. На даному етапі розвитку інструментарію розробки інформаційних систем популярною є *технологія предметно-орієнтованого проектування (DDD – Domain-Driven Design)* [14, 15]. DDD є підходом до створення та удосконалення складних програмних систем, який ґрунтується на модельній інформації про певну предметну область.

Метою роботи є розробка концепції архітектури програмного забезпечення для автоматизованого розв'язання сингулярно збурених задач з використанням методу примезових поправок, а також її реалізація для базової задачі напівпровідникової електроніки із залученням сучасних інформаційних технологій.

Постановка модельної задачі і проблематика розв'язання нелінійних сингулярно збурених задач. В *надвисокочастотній електроніці* широкого застосування набув напівпровідниковий $p-i-n$ діод (*плазмовий діод*) [16]. Пристрій утворюють формуванням контактів відповідних матеріалів з різними типами провідності (між напівпровідниками p – діркового) та n – (електронного) типів розміщують шар власного напівпровідника – матеріал i – типу). Робота напівпровідникового $p-i-n$ діода ґрунтується на можливості керування провідністю електронно-діркової плазми, що утворюється в активній області (i – області) такого пристрою.

Для дослідження електропровідних характеристик i – області $p-i-n$ –діодів широко використовується *дифузійно-дрейфова модель*, яка базується на рівняннях неперервності електронного і діркового струмів і *рівнянні Пуассона* з відповідними *граничними умовами*. Для одновимірного стаціонарного випадку, доцільність розгляду якого зумовлена особливостями геометричної структури пристрою та типовими режимами його роботи, постановка *модельної задачі* має такий вигляд [17 – 19]:

$$\begin{cases} -\mu^2 \frac{d^2 \varphi}{dx^2} = (p - n + N_d), \\ \frac{d^2 n}{dx^2} - \frac{dn}{dx} \frac{d\varphi}{dx} - n \frac{d^2 \varphi}{dx^2} - A_n n = 0, \\ \frac{d^2 p}{dx^2} + \frac{dp}{dx} \frac{d\varphi}{dx} + p \frac{d^2 \varphi}{dx^2} - A_p p = 0; \end{cases} \quad (1)$$

$$\left(\frac{dn}{dx} - n \frac{d\varphi}{dx} - \gamma_n n \right) \Big|_{x=0} = \frac{J}{eD_n} \frac{w}{N_i}, \quad \left(-\frac{dp}{dx} - p \frac{d\varphi}{dx} - \gamma_p p \right) \Big|_{x=0} = 0; \quad (2)$$

$$\left(\frac{dn}{dx} - n \frac{d\varphi}{dx} - \gamma_n n \right) \Big|_{x=1} = 0, \quad \left(-\frac{dp}{dx} - p \frac{d\varphi}{dx} - \gamma_p p \right) \Big|_{x=1} = \frac{J}{eD_p} \frac{w}{N_i}, \quad \varphi|_{x=0} = 0, \quad \varphi|_{x=1} = U, \quad (3)$$

де $\varphi(x)$, $n(x)$, $p(x)$ – шукані функції розподілів потенціалу, концентрацій електронів і дірок в активній області $p-i-n$ –діода відповідно (використовується нормування $\tilde{x} = \frac{x}{w}$, $\tilde{\varphi} = \frac{e\varphi}{kT}$, $\tilde{n} = \frac{n}{N_i}$, $\tilde{p} = \frac{p}{N_i}$; у подальших викладах символ « \sim » опускається); $\mu^2 = \frac{\varepsilon\varepsilon_0 kT}{e^2 w^2 N_i}$ (малий параметр $\mu \sim 10^{-6} \div 10^{-8}$); ε – відносна діелектрична стала;

ε_0 – діелектрична стала; w – характерний розмір активної області діода; e – заряд електрона; N_i – концентрація носіїв заряду у власному напівпровіднику; k – стала Больцмана; T – температура ($^{\circ}K$);

$$A_n = \frac{w^2}{D_n \tau_n^*}, \quad A_p = \frac{w^2}{D_p \tau_p^*} - \text{сталі};$$

D_n, D_p – коефіцієнти дифузії дірок та електронів відповідно; τ_n^*, τ_p^* – характерні часи рекомбінації носіїв заряду в об'ємі активної області (у даній математичній моделі є сталими, які залежать, як і коефіцієнти дифузії, від обраного матеріалу напівпровідника); $N_d(x)$ – профіль легування активної області домішками (задана функція); J – густина струму інжекції; U – різниця потенціалів на активній області; γ_n, γ_p – коефіцієнти рекомбінації носіїв заряду на контактах.

Задача (1) – (3) є сингулярно збуреною. Розв'язки таких задач, згідно з [3, 7], шукаємо у вигляді асимптотичних рядів за малим параметром задачі наступного виду:

$$\begin{pmatrix} n(x) \\ p(x) \\ \varphi(x) \end{pmatrix} = \begin{pmatrix} N(x) + \underline{N}(\underline{\xi}) + \overline{N}(\overline{\xi}) \\ P(x) + \underline{P}(\underline{\xi}) + \overline{P}(\overline{\xi}) \\ \Phi(x) + \underline{\Phi}(\underline{\xi}) + \overline{\Phi}(\overline{\xi}) \end{pmatrix} = \begin{pmatrix} \sum_{i=0}^s \mu^i n_i(x) \\ \sum_{i=0}^s \mu^i p_i(x) \\ \sum_{i=0}^s \mu^i \varphi_i(x) \end{pmatrix} + \begin{pmatrix} \sum_{i=0}^s \mu^i \underline{N}_i(\underline{\xi}) \\ \sum_{i=0}^s \mu^i \underline{P}_i(\underline{\xi}) \\ \sum_{i=0}^s \mu^i \underline{\Phi}_i(\underline{\xi}) \end{pmatrix} + \begin{pmatrix} \sum_{i=0}^s \mu^i \overline{N}_i(\overline{\xi}) \\ \sum_{i=0}^s \mu^i \overline{P}_i(\overline{\xi}) \\ \sum_{i=0}^s \mu^i \overline{\Phi}_i(\overline{\xi}) \end{pmatrix} + \begin{pmatrix} R_{n(s)}(x, \mu) \\ R_{p(s)}(x, \mu) \\ R_{\Phi(s)}(x, \mu) \end{pmatrix}, \quad (4)$$

де $N(x), P(x), \Phi(x)$ – регулярні складові розв'язку, які подаємо у вигляді асимптотичних рядів за степенями малого параметра; $\underline{N}(\underline{\xi}), \underline{P}(\underline{\xi}), \underline{\Phi}(\underline{\xi}), \overline{N}(\overline{\xi}), \overline{P}(\overline{\xi}), \overline{\Phi}(\overline{\xi})$ – примежові поправки відповідно в околах точок $x=0$ та $x=1$, які також подаємо у вигляді відповідних асимптотичних рядів ($\underline{\xi} = \frac{x}{\mu}, \overline{\xi} = \frac{1-x}{\mu}$ – відповідні регуляризуючі розтяги); $R_{\Phi(s)}(x, \mu), R_{n(s)}(x, \mu), R_{p(s)}(x, \mu)$ – залишкові члени.

Шляхом виконання стандартної процедури декомпозиції вихідної задачі (1) – (3) (на основі «прирівнювання») приходимо до постановок рекурентної послідовності задач для систем звичайних диференціальних рівнянь з відповідними граничними умовами (за аналогією до [3, 4], де можна ознайомитись із детальним описом відповідного алгоритму).

Наведемо приклади елементів такої послідовності задач (для випадку, коли домішки у досліджуваній області відсутні: $N_d = 0$):

$$\begin{cases} p_0(x) - n_0(x) = 0, \\ \frac{d^2 n_0(x)}{dx^2} - \frac{d}{dx} \left(n_0(x) \frac{d\varphi_0(x)}{dx} \right) - A_n n_0(x) = 0, \\ \frac{d^2 p_0(x)}{dx^2} + \frac{d}{dx} \left(p_0(x) \frac{d\varphi_0(x)}{dx} \right) - A_p p_0(x) = 0, \end{cases}$$

$$\begin{cases} \left(\frac{dn_0(x)}{dx} - n_0 \frac{d\varphi_0(x)}{dx} - \gamma_n w n_0(x) \right) \Big|_{x=0} = \frac{J}{eD_n} \frac{w}{N_i}, \quad \left(-\frac{dp_0}{dx} - p_0 \frac{d\varphi_0(x)}{dx} - \gamma_p w p_0 \right) \Big|_{x=0} = 0, \\ \left(-\frac{dp_0(x)}{dx} - p_0 \frac{d\varphi_0(x)}{dx} - \gamma_p w p_0(x) \right) \Big|_{x=1} = \frac{J}{eD_p} \frac{w}{N_i}, \quad \left(\frac{dn_0}{dx} - n_0 \frac{d\varphi_0(x)}{dx} - \gamma_n w n_0 \right) \Big|_{x=1} = 0; \end{cases} \quad (5.1)$$

$$\begin{cases} \frac{d^2 \underline{\Phi}_0(\underline{\xi})}{d\underline{\xi}^2} = -(\underline{P}_0(\underline{\xi}) - \underline{N}_0(\underline{\xi})), \\ \frac{d^2 \underline{N}_0(\underline{\xi})}{d\underline{\xi}^2} = \frac{d}{d\underline{\xi}} \left(\left(n_0(0) + \underline{N}_0(\underline{\xi}) \right) \frac{d\underline{\Phi}_0(\underline{\xi})}{d\underline{\xi}} \right), \\ \frac{d^2 \underline{P}_0(\underline{\xi})}{d\underline{\xi}^2} = -\frac{d}{d\underline{\xi}} \left(\left(p_0(0) + \underline{P}_0(\underline{\xi}) \right) \frac{d\underline{\Phi}_0(\underline{\xi})}{d\underline{\xi}} \right), \end{cases}$$

$$\frac{\partial N_0}{\partial \xi} - N_0 \frac{\partial \Phi_0}{\partial \xi} - n_0(0) \frac{\partial \Phi_0}{\partial \xi} \Big|_{\xi=0} = 0, \quad -\frac{\partial P_0}{\partial \xi} - P_0 \frac{\partial \Phi_0}{\partial \xi} - p_0(0) \frac{\partial \Phi_0}{\partial \xi} \Big|_{\xi=0} = 0,$$

$$\lim_{\xi \rightarrow \infty} N_0(\xi) = 0, \quad \lim_{\xi \rightarrow \infty} P_0(\xi) = 0, \quad \varphi_0 + \Phi_0 \Big|_{\xi=0} = 0; \tag{5.2}$$

$$\begin{cases} \frac{d^2 \bar{\Phi}_0(\bar{\xi})}{d\bar{\xi}^2} = -(\bar{P}_0(\bar{\xi}) - \bar{N}_0(\bar{\xi})), \\ \frac{d^2 \bar{N}_0(\bar{\xi})}{d\bar{\xi}^2} = \frac{d}{d\bar{\xi}} \left((n_0(1) + \bar{N}_0(\bar{\xi})) \frac{d\bar{\Phi}_0(\bar{\xi})}{d\bar{\xi}} \right), \\ \frac{d^2 \bar{P}_0(\bar{\xi})}{d\bar{\xi}^2} = -\frac{d}{d\bar{\xi}} \left((p_0(1) + \bar{P}_0(\bar{\xi})) \frac{d\bar{\Phi}_0(\bar{\xi})}{d\bar{\xi}} \right), \end{cases}$$

$$\frac{\partial \bar{N}_0}{\partial \bar{\xi}} - \bar{N}_0 \frac{\partial \bar{\Phi}_0}{\partial \bar{\xi}} - n_0(1) \frac{\partial \bar{\Phi}_0}{\partial \bar{\xi}} \Big|_{\bar{\xi}=0} = 0, \quad -\frac{\partial \bar{P}_0}{\partial \bar{\xi}} - \bar{P}_0 \frac{\partial \bar{\Phi}_0}{\partial \bar{\xi}} - p_0(1) \frac{\partial \bar{\Phi}_0}{\partial \bar{\xi}} \Big|_{\bar{\xi}=0} = 0,$$

$$\lim_{\bar{\xi} \rightarrow \infty} \bar{N}_0(\bar{\xi}) = 0, \quad \lim_{\bar{\xi} \rightarrow \infty} \bar{P}_0(\bar{\xi}) = 0, \quad \varphi_0 + \bar{\Phi}_0 \Big|_{x=1} = U; \tag{5.3}$$

$$\lim_{\xi \rightarrow x^*} \Phi_0(\xi) = \lim_{\bar{\xi} \rightarrow \frac{1-x^*}{\mu}} \bar{\Phi}_0(\bar{\xi}), \quad \lim_{\xi \rightarrow x^*} \frac{\partial \Phi_0(\xi)}{\partial \xi} = \lim_{\bar{\xi} \rightarrow \frac{1-x^*}{\mu}} \frac{\partial \bar{\Phi}_0(\bar{\xi})}{\partial \bar{\xi}}. \tag{5.4}$$

При цьому зауважимо, що «надлишок» умов (а саме (5.4)) забезпечує можливість знайти ще й характерну точку $x = x^*$, в якій узгоджуються потенціальне та силове поля, породжені примежовими поправками до розподілів концентрацій електронів $n(x)$ та дірок $p(x)$ (відповідно в околах $x = 0$ та $x = 1$).

Аналогічно для поправок вищих порядків матимемо:

$$\begin{cases} p_1(x) - n_1(x) = 0, \\ \frac{d^2 n_1(x)}{dx^2} - \frac{d}{dx} \left(n_1(x) \frac{d\varphi_0(x)}{dx} + n_0(x) \frac{d\varphi_1(x)}{dx} \right) - A_n n_1(x) = 0, \\ \frac{d^2 p_1(x)}{dx^2} + \frac{d}{dx} \left(p_1(x) \frac{d\varphi_0(x)}{dx} + p_0(x) \frac{d\varphi_1(x)}{dx} \right) - A_p p_1(x) = 0, \end{cases}$$

$$\left(\frac{dn_1(x)}{dx} - n_0 \frac{d\varphi_1(x)}{dx} - n_1 \frac{d\varphi_0(x)}{dx} - \gamma_n w n_1(x) \right) \Big|_{x=0} = 0, \quad \left(-\frac{dp_1}{dx} - p_1 \frac{d\varphi_0(x)}{dx} - p_0 \frac{d\varphi_1(x)}{dx} - \gamma_p w p_1 \right) \Big|_{x=0} = 0,$$

$$\left(-\frac{dp_1(x)}{dx} - p_1 \frac{d\varphi_0(x)}{dx} - p_0 \frac{d\varphi_1(x)}{dx} - \gamma_p w p_1(x) \right) \Big|_{x=1} = 0, \quad \left(\frac{dn_1}{dx} - n_1 \frac{d\varphi_0(x)}{dx} - n_0 \frac{d\varphi_1(x)}{dx} - \gamma_n w n_1 \right) \Big|_{x=1} = 0; \tag{6.1}$$

$$\begin{cases} \frac{d^2 \Phi_1(\xi)}{d\xi^2} = -(P_1(\xi) - N_1(\xi)), \\ \frac{d^2 N_1(\xi)}{d\xi^2} = \frac{d}{d\xi} \left(\varphi'_{0x}(0) N_0(\xi) + \frac{d\Phi_1(\xi)}{d\xi} (n_0(0) + N_0(\xi)) + \frac{d\Phi_0(\xi)}{d\xi} (n_1(0) + \xi n'_\varphi(0)) + N_1(\xi) \frac{d\Phi_0(\xi)}{d\xi} \right), \\ \frac{d^2 P_1(\xi)}{d\xi^2} = -\frac{d}{d\xi} \left(\varphi'_{0x}(0) P_0(\xi) + \frac{d\Phi_1(\xi)}{d\xi} (p_0(0) + P_0(\xi)) + \frac{d\Phi_0(\xi)}{d\xi} (p_1(0) + \xi p'_\varphi(0)) + P_1(\xi) \frac{d\Phi_0(\xi)}{d\xi} \right), \end{cases}$$

$$\left. \begin{aligned} \frac{\partial N_1}{\partial \xi} - N_0 \frac{\partial \Phi_1}{\partial \xi} - n_0(0) \frac{\partial \Phi_1}{\partial \xi} - N_1 \frac{\partial \Phi_0}{\partial \xi} - n_1(0) \frac{\partial \Phi_0}{\partial \xi} - \varphi'_{0x}(0) N_0(\xi) - \gamma_n w N_0(\xi) \right|_{\xi=0} &= 0, \\ -\frac{\partial P_1}{\partial \xi} - P_0 \frac{\partial \Phi_1}{\partial \xi} - p_0(0) \frac{\partial \Phi_1}{\partial \xi} - P_1 \frac{\partial \Phi_0}{\partial \xi} - p_1(0) \frac{\partial \Phi_0}{\partial \xi} - \varphi'_{0x}(0) P_0(\xi) - \gamma_p w P_0(\xi) \right|_{\xi=0} &= 0, \\ \lim_{\xi \rightarrow \infty} N_1(\xi) = 0, \quad \lim_{\xi \rightarrow \infty} P_1(\xi) = 0, \quad \varphi_1 + \Phi_1|_{x=0} = 0; \end{aligned} \quad (6.2)$$

$$\left\{ \begin{aligned} \frac{d^2 \bar{\Phi}_1(\bar{\xi})}{d\bar{\xi}^2} &= -(\bar{P}_1(\bar{\xi}) - \bar{N}_1(\bar{\xi})), \\ \frac{d^2 \bar{N}_1(\bar{\xi})}{d\bar{\xi}^2} &= \frac{d}{d\bar{\xi}} \left(-\varphi'_{0x}(1) \bar{N}_0(\bar{\xi}) + \frac{d\bar{\Phi}_1(\bar{\xi})}{d\bar{\xi}} (n_0(1) + \bar{N}_0(\bar{\xi})) + \frac{d\bar{\Phi}_0(\bar{\xi})}{d\bar{\xi}} (n_1(1) - \bar{\xi} n'_{0\mu}(1)) + \bar{N}_1(\bar{\xi}) \frac{d\bar{\Phi}_0(\bar{\xi})}{d\bar{\xi}} \right), \\ \frac{d^2 \bar{P}_1(\bar{\xi})}{d\bar{\xi}^2} &= -\frac{d}{d\bar{\xi}} \left(-\varphi'_{0x}(1) \bar{P}_0(\bar{\xi}) + \frac{d\bar{\Phi}_1(\bar{\xi})}{d\bar{\xi}} (p_0(1) + \bar{P}_0(\bar{\xi})) + \frac{d\bar{\Phi}_0(\bar{\xi})}{d\bar{\xi}} (p_1(1) - \bar{\xi} p'_{0\mu}(1)) + \bar{P}_1(\bar{\xi}) \frac{d\bar{\Phi}_0(\bar{\xi})}{d\bar{\xi}} \right), \end{aligned} \right.$$

$$\left. \begin{aligned} \frac{\partial \bar{N}_1}{\partial \bar{\xi}} - \bar{N}_0 \frac{\partial \bar{\Phi}_1}{\partial \bar{\xi}} - n_0(0) \frac{\partial \bar{\Phi}_1}{\partial \bar{\xi}} - \bar{N}_1 \frac{\partial \bar{\Phi}_0}{\partial \bar{\xi}} - n_1(0) \frac{\partial \bar{\Phi}_0}{\partial \bar{\xi}} + \varphi'_{0x}(1) \bar{N}_0(\bar{\xi}) + \gamma_n w \bar{N}_0(\bar{\xi}) \right|_{\bar{\xi}=0} &= 0, \\ -\frac{\partial \bar{P}_1}{\partial \bar{\xi}} - \bar{P}_0 \frac{\partial \bar{\Phi}_1}{\partial \bar{\xi}} - p_0(0) \frac{\partial \bar{\Phi}_1}{\partial \bar{\xi}} - \bar{P}_1 \frac{\partial \bar{\Phi}_0}{\partial \bar{\xi}} - p_1(0) \frac{\partial \bar{\Phi}_0}{\partial \bar{\xi}} + \varphi'_{0x}(1) \bar{P}_0(\bar{\xi}) + \gamma_p w \bar{P}_0(\bar{\xi}) \right|_{\bar{\xi}=0} &= 0, \\ \lim_{\bar{\xi} \rightarrow \infty} \bar{N}_1(\bar{\xi}) = 0, \quad \lim_{\bar{\xi} \rightarrow \infty} \bar{P}_1(\bar{\xi}) = 0, \quad \varphi_1 + \bar{\Phi}_1|_{x=1} = 0; \end{aligned} \quad (6.3)$$

$$\lim_{\xi \rightarrow \frac{x}{\mu}}^* \Phi_1(\xi) = \lim_{\bar{\xi} \rightarrow \frac{1-x}{\mu}}^* \bar{\Phi}_1(\bar{\xi}), \quad \lim_{\xi \rightarrow \frac{x}{\mu}}^* \frac{\partial \Phi_1(\xi)}{\partial \xi} = \lim_{\bar{\xi} \rightarrow \frac{1-x}{\mu}}^* \frac{\partial \bar{\Phi}_1(\bar{\xi})}{\partial \bar{\xi}}. \quad (6.4)$$

У постановках задач (5) – (6): $\varphi_0(x)$, $n_0(x)$, $p_0(x)$ – головні члени регулярної частини асимптотики; $N_0(\xi)$, $P_0(\xi)$, $\Phi_0(\xi)$, $\bar{N}_0(\bar{\xi})$, $\bar{P}_0(\bar{\xi})$, $\bar{\Phi}_0(\bar{\xi})$ – головні члени прилежових поправок відповідно в околах точок $x=0$ та $x=1$; $\varphi_1(x)$, $n_1(x)$, $p_1(x)$, $N_1(\xi)$, $P_1(\xi)$, $\Phi_1(\xi)$, $\bar{N}_1(\bar{\xi})$, $\bar{P}_1(\bar{\xi})$, $\bar{\Phi}_1(\bar{\xi})$ – наступні члени асимптотики. Відмітимо, що для пошуку членів асимптотики з індексом $i \geq 2$ ($i \geq 1$ для прилежових складових електростатичного поля) будуться задачі, які аналогічні (5) – (6) [3, 4].

Застосування методу прилежових поправок для розв'язання нелінійної задачі (1) – (3) (нагадаємо, що задача (1) – (3) є типовою задачею напівпровідникової електроніки) має свої переваги. Застосування відповідних обчислювальних схем до розв'язання такої задачі супроводжується витратами значних обчислювальних ресурсів, виникають проблеми із стійкістю побудованих обчислювальних схем [20]. Задачі (5) – (6) на відміну від (1) – (3) є значно простішими, у даному випадку – лінійними, основна частина з яких є неоднорідними, із змінними коефіцієнтами. Такого типу задачі є добре вивченими. Все ж процедура розчеплення сингулярно збурених задач з використанням, наприклад, методу прилежових поправок є громіздкою, як показує частина схеми розпаралелювання (5) – (6). Вона пов'язана із розчепленням (зокрема, введенням регуляризуючого розтягу) та розв'язанням відповідних задач (зокрема, для пошуку прилежових поправок).

Зауважимо, що постановка задачі (1) – (3) є базовою. Як було зазначено вище, додатково ускладнює модель врахування, наприклад, ефектів розігріву електронно-діркової плазми; ефектів, що пов'язані із впливом зовнішніх фізичних полів, неоднорідностями досліджуваної структури; необхідністю врахування складної геометрії області розв'язання задачі, поверхневих ефектів тощо. Окремої уваги заслуговують відповідні нестационарні задачі. Зокрема, у напівпровідникових матеріалах на динаміку носіїв заряду впливає розігрів кристалу та електронно-діркової плазми внаслідок виділення в об'ємі $p-i-n$ -діода *Джосулевого тепла* та вивільнення енергії внаслідок рекомбінації носіїв заряду [9, 21]. Ці процеси приводять до зміни значення малого параметра задачі μ ($\mu(T)$) та значення коефіцієнтів дифузії носіїв заряду D_n ($D_n(T)$), D_p ($D_p(T)$). Для опису температурного поля використовується рівняння теплопровідності (7) з відповідними граничними умовами виду (8) (у одновимірному випадку):

$$c_p \rho \frac{\partial T}{\partial t} - \frac{\partial}{\partial x} \left(\lambda_t \frac{\partial T}{\partial x} \right) = (j_n + j_p) \cdot E + (R - G) \left(E_g - \frac{\alpha T^2}{T + \beta} \right), \quad (7)$$

$$\left. \frac{\partial T}{\partial x} \right|_{x=0} = k(T - T_0), \quad \left. \frac{\partial T}{\partial x} \right|_{x=1} = -k(T - T_0), \quad T(t, x)|_{t=0} = T^*(x), \quad (8)$$

де c_p – питома теплоємність напівпровідникового матеріалу; ρ – густина речовини; λ_t – коефіцієнт теплопровідності; E_g – ширина забороненої зони напівпровідника; $R = R_n + R_p$ – швидкість рекомбінації носіїв заряду; $G = G_n + G_p$ – швидкість генерації електронів та дірок; α, β – феноменологічні сталі (наприклад, для кремнію (Si) – $E_g = 1.169 \text{ eV}$, $\alpha = 4.9 \cdot 10^{-4} \text{ eV/K}$, $\beta = 655 \text{ K}$ [18]); $k = \sigma / \lambda_t$, σ – коефіцієнти теплопередачі, $T^*(x)$ – задане температурне поле у початковий момент часу.

Доцільним вважаємо розробку програмного забезпечення для проведення автоматизованої декомпозиції модельної задачі та синтезу її розв’язків. Цей алгоритм був би корисною компонентою ядра не тільки спеціалізованих САПР, але й програм, які забезпечують виконання символічних перетворень для розв’язання сингулярно збурених задач математичної фізики у цілому. Відзначимо, що кількість задач відповідного типу та сфери застосування методів теорії збурень зростають. Розробці таких алгоритмів сприяє поява низки сучасних інформаційних технологій таких, як *штучний інтелект (AI)*, *машинне навчання (ML)*, хмарні обчислення, технології розпаралелювання, символічні обчислення тощо [22 – 25]. Зокрема, хмарні обчислення дозволяють використовувати обчислювальні ресурси у мережі інтернет, не вдаючись до власних фізичних серверів або обчислювальних машин, що забезпечує гнучкість, масштабованість та ефективність використання ресурсів. Технології розпаралелювання забезпечують можливість одночасного виконання кількох обчислювальних процесів, що значно збільшує швидкість обробки даних та ефективність вирішення складних науково-дослідницьких обчислювальних задач за рахунок розподілу виконання програм між декількома процесорами або комп’ютерами, що об’єднані в мережі, та оптимізації доступних ресурсів. Символічні обчислення на відміну від числових обчислень, що використовують апроксимацію для представлення чисел та виконання математичних операцій, дозволяють точно маніпулювати виразами, виконуючи алгебраїчні перетворення, інтегрування, диференціювання та інші математичні операції.

Розробка концепції архітектури програмного забезпечення для вирішення сингулярно збурених задач (ПЗ ВСЗЗ). Архітектура ПЗ ВСЗЗ, що проєктується, очевидно, визначається набором задач, на вирішення яких націлено ПЗ. Цілі ПЗ зрозумілі із контексту постановки та застосування математичних методів розв’язання відповідних задач моделювання математичних (наприклад, (1) – (3)) та інформаційних (наприклад, задач декомпозиції, агрегування). Спираючись на постановку нелінійної задачі моделювання (1) – (3) та керуючись загальною методикою розв’язання сингулярно збурених задач [7, 8], приходимо до розуміння того, що результат моделювання (набір функціональних залежностей між характеристиками досліджуваної технічної системи) досягається шляхом, по-перше, виконання символічних перетворень базових рівнянь та умов в постановці задачі (у базові рівняння здійснюється підстановка відповідних степеневих рядів, що побудовані за малим параметром задачі; проводиться групування членів рівнянь (умов) при однакових степенях малого параметра). Отримуємо рекурент-декомпозицію базової задачі (1) – (3) (рис. 1, **етап I**). По-друге, визначається тип кожної задачі із отриманої послідовності і приймається рішення про спосіб її розв’язання (рис. 1, **етап II**). Далі, у залежності від визначеного типу підзадачі, задіюється певний метод аналітичного чи чисельного розв’язання підзадачі (рис. 1, **етап III**). На останньому етапі проводиться агрегування отриманих даних у потрібному для користувача програмного продукту вигляді.

Згідно методології предметно-орієнтованого проєктування (DDD) на початковому етапі розробки програмної системи виділяємо компоненти предметної області, їх основні властивості [14, 26] та формуємо систему відповідних ідентифікаторів. В нашому випадку, оскільки в основу інформаційної системи закладено постановку певної математичної проблеми, доцільним (і очевидним) є використання математичних виразів різного виду (рівнянь, умов, постановок задач) у якості структурних одиниць та словника з переважанням математичних (математичної теорії збурень) і технічних (теорії пристроїв напівпровідникової електроніки) термінів.

Компоненти предметної області доцільно розділити на дві групи. Перша включає у себе елементи постановки базової математичної проблеми (постановку модельної задачі (1) – (3)), які є вхідними даними інформаційної системи. Друга містить об’єкти, що формуються у результаті перетворення первинної вхідної інформації за системою відповідних правил (обґрунтованих, наприклад, методом примежових поправок), і є фактично постановками конкретних математичних задач визначеного типу (наприклад, постановками задач Коші, крайових задач, наприклад (5) – (6)). Якщо елементами першої групи є текстова інформація, яка визначає опис математичного рівняння чи умови, то елементами другої групи є інформаційні об’єкти з наборами даних для пошуку аналітичних (якщо це можливо) чи чисельних розв’язків відповідними методами. Між елементами відповідних груп предметної області існує відповідність. Фактично елементи другої групи є результатами проведення декомпозиції початкової задачі асимптотичними методами (рис. 1).

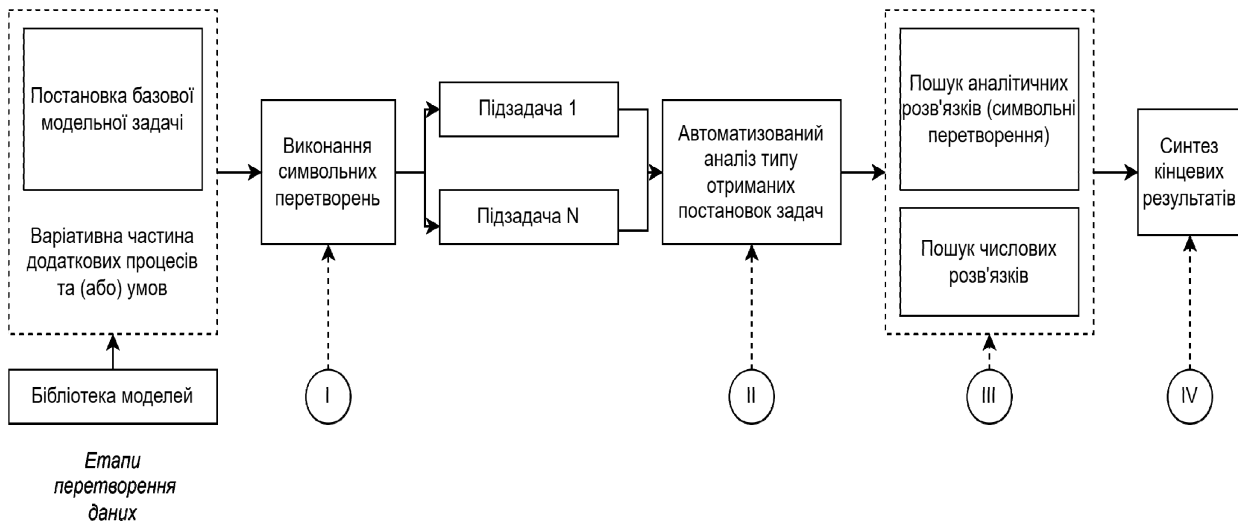


Рис. 1 – Структурна схема вирішення сингулярно збурених задач.

У результаті аналізу предметної області приходимо до висновку про те, що програмна реалізація моделі повинна включати блоки (домени) символічних перетворень; інтелектуального аналізу вхідної інформації, що включає етапи проведення декомпозиції вихідної задачі та визначення типу математичної підзадачі, яка є результатом декомпозиції; чисельного (або символічного) розв'язання математичних задач згідно отриманих постановок підзадач, блок синтезу кінцевих результатів, а також інтерфейсний блок. Зазначені блоки повинні включати в себе низку взаємопов'язаних алгоритмів. Наприклад, алгоритми роботи з текстом: пошук, об'єднання, вилучення (текстовий процесор); алгоритми розпізнавання типу математичної задачі (класифікації) і прийняття рішень про спосіб її розв'язання (аналіз даних відповідними нейронними мережами); алгоритми символічно-числового розв'язання математичних задач (числовий процесор); алгоритми синтезу кінцевого результату (агрегатор) тощо. Відзначимо, що на етапах пошуку розв'язків математичних задач (етап III, рис. 1) та агрегування кінцевих результатів (етап IV, рис. 1) з'являється можливість розпаралелювання обчислювальних процесів. Перелік основних структурних елементів програмної системи утворює її змістовне ядро.

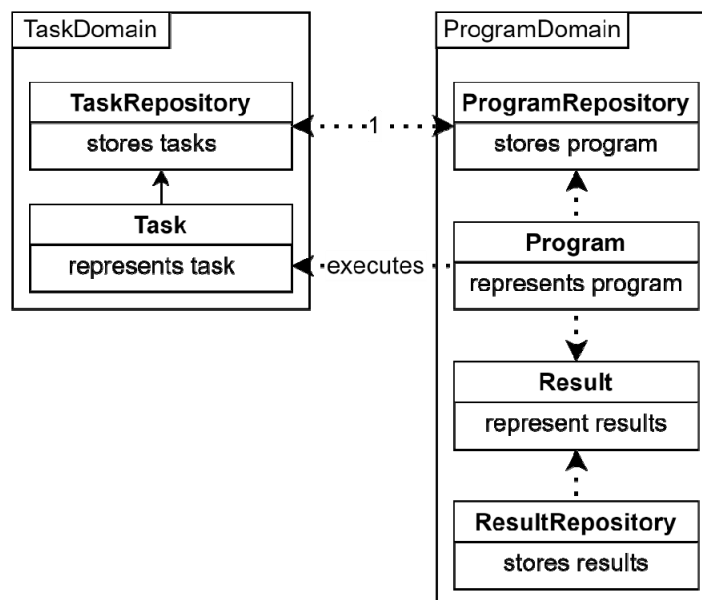


Рис. 2 – Діаграма контекстів (Context Map).

Діаграма контекстів (рис. 2) ілюструє взаємозв'язок між двома основними доменами (блоками): *TaskDomain* і *ProgramDomain*, що є ключовими елементами в розробці програмної системи згідно методології DDD [27]. У *TaskDomain* представлені *TaskRepository*, інтерфейсний блок якого відповідає за зберігання вхідних даних послідовності задач (наприклад, задач (4) – (5)), та *Task*, який містить набір методів реалізації алгоритмів розв'язання конкретних математичних задач і задач, що пов'язані з обробкою даних. *TaskRepository* взаємодіє з *Task*, забез-

печуючи управління їхнім життєвим циклом, що є критично важливим для символічних перетворень, інтелектуального аналізу вхідної інформації, чисельних перетворень та синтезу результатів. У ProgramDomain відображені ProgramRepository, що зберігає програми (Program), які визначають загальну логіку розв'язання задач, що структуруються на основі методів TaskRepository. Компонента ProgramDomain Result відображає результати виконання структурованого алгоритму, які передаються ResultRepository для збереження. ProgramRepository взаємодіє з Program, яка виконує задачі, отримані з TaskDomain, забезпечуючи ефективне чисельне або символічне розв'язання математичних задач, задач інтелектуального аналізу та синтезу результатів виконання структурованих алгоритмів. Взаємозв'язок між Task та Program демонструє, що Task виконується за допомогою Program, а результати виконання зберігаються в ResultRepository для подальшого аналізу та використання. Така структура підкреслює модульність і чітке розділення обов'язків між доменами, що полегшує управління задачами та програмами в системі, а також сприяє побудові ефективної системи символічних та числових перетворень, задач аналізу та синтезу.

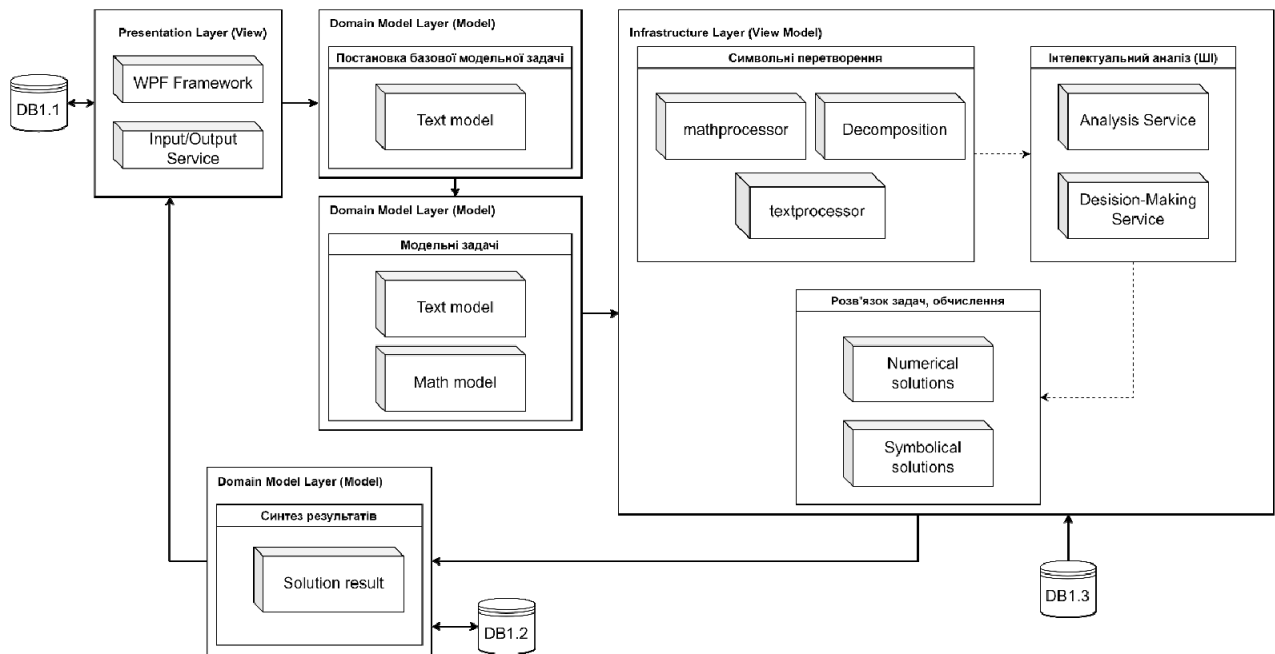


Рис. 3 – Архітектура програмного забезпечення моделювання та аналізу з використанням WPF, моделей домену та інфраструктурних сервісів.

Діаграма архітектури ПЗ (рис. 3) [28] демонструє багаторівневу архітектуру системи, що складається з чотирьох основних шарів: *Presentation Layer (View)*, *I/O Layer*, *Domain Model Layer (Model)* та *Infrastructure Layer (View Model)*. *Presentation Layer* відповідає за відображення даних користувачеві та взаємодію з ним за допомогою *WPF Framework*, що забезпечує зручний інтерфейс користувача. *I/O Layer* обробляє вхідні та вихідні дані через *Input/Output Service*, який є посередником між *Presentation Layer* та *Domain Model Layer*. Дані тут подаються у вигляді рядків, що відображають рівняння моделі та додаткові умови. При цьому існує можливість використовувати стандартні структури даних із бази даних *DB1.1* запропонованої програмної системи. Також тут потрібно передбачити зручний (дружній) інтерфейс користувача для формування остаточного вигляду модельної задачі. *Domain Model Layer* реалізує логіку домену та моделювання предметної області, включаючи *ER model*, *Text model* та *Math model*, що дозволяє проводити декомпозицію задачі (при цьому використовуються інструменти для символічних перетворень) та визначати типи математичних підзадач (методами інтелектуального аналізу). Нарешті, *Infrastructure Layer* забезпечує інфраструктурні сервіси для моделі представлення, включаючи *Analysis Service*, *Decision-Making Service*, *Decomposition*, *textprocessor*, *mathprocessor* та бази даних *DB1.3*. Особливістю компонентів аналізу в системі є процес прийняття рішень, для забезпечення якого пропонується використання *нейронних мереж*. Нейронні мережі здатні ефективно моделювати складні взаємозв'язки між вхідними даними та результатами [29 – 31], забезпечуючи належну точність і адаптивність при розв'язанні завдань. Вони дозволяють системі здійснювати інтелектуальний аналіз і автоматично приймати рішення на основі навчання на великих масивах даних. Завдяки їх здатності до самонавчання, нейронні мережі можуть покращувати результативність з часом, що робить їх ідеальними для задач, пов'язаних з аналізом, класифікацією, прогнозуванням і оптимізацією в рамках програмної системи. Бібліотеки (*textprocessor*, *mathprocessor*) зберігають алгоритми символічних перетворень (*textprocessor*) та чисельних перетворень (*mathprocessor*) і при цьому використовують дані про фізичні характеристики матеріалів з *DB1.3*. Взаємодія між цими шарами здійснюється через чітко визначені інтерфейси, що дозволяє підтримувати модульність та розширюваність системи, забезпечуючи ефективну реалізацію методів символічного та числового розв'язання задач та синтезу кінцевих результатів. Результати проміжних

перетворень агрегуються у кінцевий результат – розв’язок поставленої задачі, який подається у табличному (або графічному) вигляді і зберігається у базі даних *DBI.2*.

Зазначимо, що компоненти моделі архітектури програмної системи для розв’язання задач напівпровідникової електроніки (симуляції) можуть бути постійні (складають ядро програмної системи), варіабельні (входять у структуру алгоритму, але додаються в модель у залежності від потреб моделювання (наприклад, врахування різних зовнішніх впливів, врахування різної природи процесів у системі (процеси рекомбінації можуть мати різну природу), геометрія просторової області тощо, рис. 1 – бібліотека моделей)) та нові (що невідомі на момент розробки моделі і відповідної програмної системи) – ті, що розширюють можливості стандартної програмної системи.

На діаграмі варіантів (способів) використання (*Use Case*, рис. 4) [32] зображено послідовність дій, необхідних для розв’язання задач. Вона охоплює різноманітні етапи процесу моделювання, від створення математичної моделі до агрегування результатів, і включає взаємодію між кількома користувачами та варіантами використання, що представляють ключові кроки вирішення задачі.

Діаграма відображає етапи використання ПЗ. Початковим є етап створення математичної моделі (*Create Mathematical Model*), на якому здійснюється опис фізичних процесів у технічній системі та їх перетворення у відповідні математичні вирази чи систему диференціальних рівнянь. Цей крок є фундаментальним, оскільки точність і коректність подальших розрахунків залежать від якості сформульованої математичної моделі. Після створення моделі здійснюється декомпозиція задачі (*Decompose Task*), яка розбиває складну задачу на підзадачі, що можуть бути розв’язані незалежно. Взаємодія між створенням моделі та декомпозицією підкреслює, що остання залежить від коректності початкової моделі.

Наступним етапом є налаштування обчислювального середовища (*Setup Computational Environment*), що виконується іншим користувачем. Цей етап включає підготовку необхідних апаратних і програмних ресурсів для проведення чисельних чи аналітичних розрахунків. Взаємодія на цьому етапі відбувається через обмін даними про характеристики задачі та необхідні обчислювальні ресурси, що мають бути доступні для виконання процедури розв’язання задачі. Після налаштування середовища здійснюється вибір методу розв’язання (*Select Solution Method*), що передбачає визначення найбільш оптимального підходу для кожної підзадачі. Цей вибір базується на аналізі типу задачі, що дозволяє забезпечити максимальну ефективність і точність обчислень.

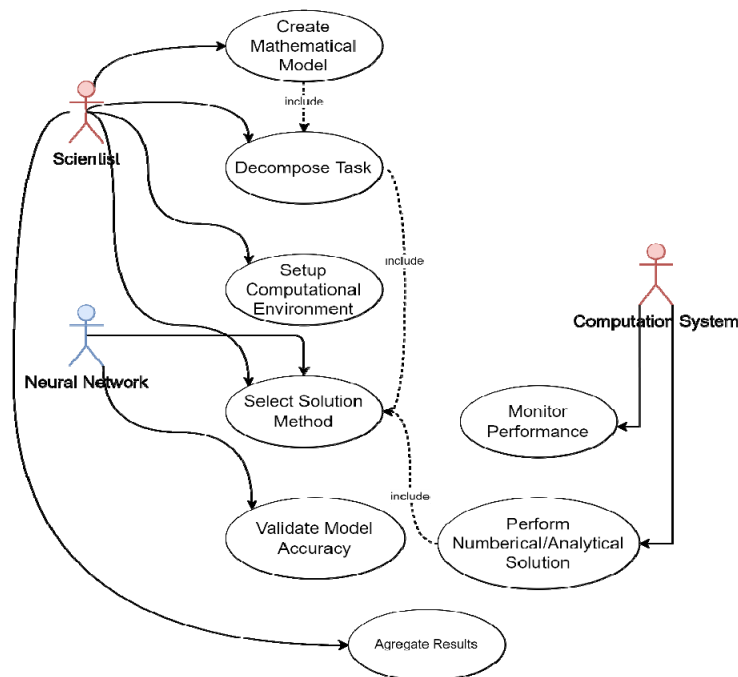


Рис. 4 – Діаграма варіантів (способів) використання (*Use Case*).

Виконання чисельного або аналітичного розв’язання (*Perform Numerical/Analytical Solution*) є наступним кроком після вибору методу. Тут реалізується застосування обраних методів до відповідних підзадач. Взаємодія з попереднім етапом (*Select Solution Method*) забезпечується використанням обраного методу розв’язання для кожної окремої підзадачі. Перевірка точності моделі (*Validate Model Accuracy*) проводиться на декількох етапах процесу, щоб упевнитися в коректності отриманих результатів і, за необхідності, здійснити коригування обчислювальних параметрів чи самого методу.

Останні етапи включають моніторинг продуктивності (*Monitor Performance*) і агрегування результатів (*Aggregate Results*), які забезпечують збір та аналіз та відображення результатів розрахунків для остаточної оцін-

ки моделі та її відповідності очікуваним фізичним і технічним характеристикам. Всі ці етапи тісно взаємопов'язані між собою, забезпечуючи інтегрований підхід до розв'язання складних задач моделювання в напівпровідниковій електроніці. Така структура підкреслює важливість комплексного підходу, що поєднує декомпозицію задачі, налаштування обчислювальних ресурсів, вибір методів розв'язання та їхню *валідацію*, що в сукупності забезпечує точність і ефективність математичного моделювання.

Основним інструментом розробки даного ПЗ є *система програмування C#*, яка забезпечує високу продуктивність та широкі можливості для реалізації складних обчислювальних алгоритмів [33]. Інтерфейс користувача створюється за допомогою *фреймворка WPF (Windows Presentation Foundation)*, що дозволяє розробляти сучасні графічні інтерфейси [34].

Реалізацію архітектури програмної системи, що розробляється, доцільно проводити на основі використання шаблонів проектування *Model-View-ViewModel (MVVM)* [35]. Даний архітектурний шаблон застосовується для відокремлення бізнес-логіки від візуальної частини програми. У цій архітектурі Model представляє бізнес-логіку та дані додатка, забезпечуючи *інкапсуляцію логіки* та доступ до даних. View відповідає за візуальне подання даних і взаємодію з користувачем, реалізуючи графічний інтерфейс. ViewModel слугує посередником між Model та View, забезпечуючи двосторонню прив'язку даних та обробку подій, що дозволяє автоматично оновлювати візуальне подання при зміні даних.

Для обробки математичних обчислень та символічних перетворень пропонується використання бібліотеки *Math.NET* [36], яка надає широкий спектр функцій для виконання числових та символічних операцій. Крім того, для розпаралелювання обчислювальних процесів застосовується технологія *Task Parallel Library (TPL)* [24], що входить до складу *.NET Framework*, дозволяючи ефективно розподіляти обчислення між декількома процесорами.

Один із варіантів реалізації штучної нейронної мережі полягає у використанні ресурсу *TensorFlow.NET* [23], який є інтерфейсом для популярної бібліотеки TensorFlow, адаптованої під *.NET середовище*. TensorFlow.NET надає широкі можливості для створення, тренування та застосування нейронних мереж, що забезпечує можливість виконання складних задач таких, як класифікація, розпізнавання та обробка великих масивів даних тощо. Крім того, його інтеграція з TPL надає можливість оптимізувати обчислювальні процеси та підвищити продуктивність системи, використовуючи багатоядерні ресурси.

Важливу роль у реалізації запропонованого рішення відіграють хмарні обчислення. Використання *Microsoft Azure* [22] надає можливість масштабування обчислювальних ресурсів та забезпечує високу доступність сервісів, що є критично важливим для виконання ресурсоемних обчислювальних задач.

Висновки. На основі детального вивчення предметної області «плазмовий діод» (відповідних засобів моделювання) запропоновано концепцію архітектури спеціалізованого програмного забезпечення для автоматизованого розв'язання відповідних сингулярно збурених задач напівпровідникової електроніки з використанням методу примежових поправок. Однією із особливостей запропонованої архітектури є наявність у ній такого компонента, як штучна нейронна мережа, яка моделює інтелектуальний процес аналізу та ідентифікації типу математичних задач, що отримані у результаті декомпозиції вихідної задачі і вибору найбільш ефективного методу розв'язання останніх. Також, як особливість запропонованої програмної системи, відзначимо можливість вивчення фізичних процесів у досліджуваній технічній системі на різних рівнях структурування (вклад у процес складових, які зв'язані із малими параметрами певного порядку і якими, у переважній більшості випадків, нехтують) з доступним функціоналом для спеціалістів різного рівня підготовки і кінцевих споживачів.

В архітектуру програмного забезпечення включено об'ємну варіативну складову, яка в перспективі забезпечить врахування алгоритмів розв'язання відповідних нестационарних задач, пошук розв'язків в областях складної геометрії, таких, що враховують низку фізичних процесів, які впливають на електропровідні характеристики досліджуваної технічної системи – явища розігріву, рекомбінаційні, інжекція тощо, вплив зовнішніх фізичних полів (електромагнітне (в різних діапазонах), теплове) тощо. Зауважимо, що програмна система є відкритою стосовно можливості її удосконалення для забезпечення автоматизованого розв'язання різного роду сингулярно збурених задач.

Список літератури

1. Koroliouk D., Lyashko S., Limnios N. Computational Methods and Mathematical Modeling in Cyberphysics and Engineering Applications. – London : Wiley, 2024. – 442 p. DOI:10.1002/9781394284344.
2. Бомба А. Я., Присяжнюк І. М., Присяжнюк О. В. Методи теорії збурень прогнозування процесів тепломасоперенесення в пористих та мікропористих середовищах. – Рівне : О. Зень, 2017. – 291 с.
3. Bomba A., Moroz I. Analysis of Nonlinear Processes in the P-I-N Diodes Plasma by the Perturbation Theory Methods // 2023 13th International Conference on Advanced Computer Information Technologies, ACIT-2023. – Wroclaw, Poland. – P. 117 – 120.
4. Бомба А. Я., Мороз І. П. Математичне моделювання динамічних процесів у $p-i-n$ діоді методами теорії збурень // Вісник Національного технічного університету «ХПІ» Серія : Математичне моделювання в техніці та технологіях. – 2023. – № 2(5). – С. 23 – 32. DOI: 10.20998/2222-0631.2023.02(5).03.
5. Weinberg S. Lectures on Quantum Mechanics. – London : Cambridge University Press, 2015. – 419 p.
6. Вишик М. И., Люстерник Л. А. Регулярное вырождение и пограничный слой для линейных дифференциальных уравнений с малым параметром // УМН. – 1957. – №. 12(5). – С. 3 – 122.
7. Vasil'eva A. B., Butusov V. F., Kalachev L. V. The Boundary Function Method for Singular Perturbation Problems. – SIAM, Philadelphia, 1995. – 212 p. DOI: 10.1137/1.9781611970784.

8. Smith D. R. *Singular-Perturbation Theory. An Introduction with Applications*. – Cambridge : Cambridge Univ. Press, 1985. – 520 p.
9. Bomba A., Moroz I., Boichura M. Development and analysis of a mathematical model of plasma characteristics in the active region of integrated P-I-N-structures by the methods of perturbation theory and conformal mappings // *Східно-Європейський журнал передових технологій*. – 2021. – №5 (113). – С. 51 – 61. DOI: 10.15587/1729-4061.2021.243097
10. Huifang Liu Perturbation method for differential control equations and its application in the analysis of dynamic characteristics of curved beam // *Alexandria Engineering Journal*. – 2020. – Vol. 59. – Iss. 4. – pp. 2583 –2590. DOI: 10.1016/j.aej.2020.04.022.
11. Dirk Jansen. *The Electronic Design Automation Handbook*. – NY : Springer, 2003. – 676 p.
12. Maxfield C., Goyal K. *EDA (Electronic Design Automation): Where Electronics Begins*. – TechBITES INTERactive, 2001. – 97 p.
13. Норенков И. П. *Основы автоматизированного проектирования : Учеб. для вузов. – М. : Изд-во МГТУ им. Н.Э. Баумана, 2006. – 448 с. (Norenkov I. P., No. 82. Fundamentals Of Computer-Aided Design: Textbook For Universities. 2nd Ed., Reprint. And Additional. – Moscow : Publishing House Of The Bauman Moscow State Technical University, 2002).*
14. Evans E. *Domain-Driven Design: Tackling Complexity in the Heart of Software*. – Boston:Addison-Wesley Professional, 2003. – 560 p.
15. Tkachuk M., Gamzaev R., Martinkus I. Towards Effectiveness Assessment of Domain Modeling Methods and Tools in Software Product Lines Development // *Enterprise Modelling and Information Systems Architecture – International Journal of Conceptual Modeling*. – 2018. – Vol. 13. – P. 190 – 206. DOI: 10.18471/emisa.hcm.14.
16. Sze S., Kwok K. *Physics of Semiconductor Devices*. – New York: Wiley-Interscience, 2006. – 815 p. DOI: 10.1002/ 0470068329.
17. Adirovich E. I., Karageorgii-Alkalaev P. M., Leiderman A. Iu. *Currents Double Injection in Semiconductors*. – Moscow : Sov. radio, 1978. – 320 p.
18. Bomba A. Ya., Moroz I. P., Boichura M. V. The optimization of the shape and size of the injection contacts of the integrated $p-i-n$ structures on the base of using the conformal mapping method // *Radio Electronics, Computer Science, Control*. – 2021. – № 1. – pp. 14 – 28. DOI: 10.15588/1607-3274-2021-1-2.
19. Baliga B. *Fundamentals of power semiconductor devices*. – NY : Springer, 2010. – 1086 p.
20. Polsky B. S., Rimshans J. S. Numerical simulation of transient processes in $2-D$ bipolar transistor // *Solid State Electron*. – 1981. – v. 24. – P. 1081 – 1085.
21. Бомба А. Я., Мороз І. П. Математичне моделювання дифузійно-дрейфового процесу в активній області $p-i-n$ діодів з врахуванням розігріву та рекомбінації методами теорії збурень // *Журнал обчислювальної та прикладної математики*. – 2021. – №1 (135). – С. 29 – 35. DOI: <https://doi.org/10.17721/2706-9699.2021.1.03>.
22. Microsoft Azure Documentation. – Режим доступу : <https://docs.microsoft.com/en-us/azure/>. – Дата звертання : 15 жовтня 2024 р.
23. Abadi M., Barham P., Chen J. TensorFlow: A System for Large-Scale Machine Learning // *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16)*. – 2016. – P. 265 – 283.
24. Campbell C., Johnson R., Miller A., Toub S. *Parallel Programming with Microsoft® .NET: Design Patterns for Decomposition and Coordination on Multicore Architectures*. – Cambridge : Cambridge University Press, 2010. – 224 p.
25. Wolfram S. *Mathematica: A System for Doing Mathematics by Computer*. – 2nd edition. – Redwood-City: Addison-Wesley, 1991. – 961 p.
26. Martinkus I. Evaluation of the Effectiveness of Domain Modeling Methods in Terms of Model Complexity // *International Conference on Information and Communication Technologies in Education, Research, and Industrial Applications*. – 2019. – P. 2 – 4.
27. Evans E. *Domain-Driven Design Reference : Definitions and Pattern Summaries*. – Indianapolis : Dog Ear Publishing, 2014. – P. 7 – 8.
28. Vernon V. *Implementing Domain-Driven Design*. – Boston : Addison-Wesley Professional, 2013. – 656 p.
29. Добровська Л. М., Добровська І. А. *Теорія та практика нейронних мереж: навчальний посібник*. – Київ : НТУУ «КПІ», Видавництво «Політехніка», 2015. – 396 с.
30. Hardesty L. Explained: Neural Networks // MIT News Office. – April 14, 2017. – Режим доступу : <http://news.mit.edu/2017/explained-neural-networks-deep-learning-0414>. – Дата звертання : 16 жовтня 2024 р.
31. Haykin S. *Neural Networks: A Comprehensive Foundation*. – 2nd ed. – Upper Saddle River, NJ: Prentice Hall, 1999. – 842 p.
32. Marinescu F., Avram A. *Domain-Driven Design Quickly*. – Lugano : InfoQ Enterprise Software Development Series, 2007. – 108 p.
33. Albahari J., Albahari B. *C# 7.0 in a Nutshell: The Definitive Reference*. – Sebastopol: O'Reilly Media, 2017. – 1088 p.
34. Nathan A. *WPF 4.5 Unleashed*. – Indianapolis : Sams Publishing, 2013. – 864 p.
35. Smith J. WPF Apps With The Model-View-ViewModel Design Pattern // *MSDN Magazine*. – 2009. – vol. 24. – № 2. – Режим доступу : https://jacobfilipp.com/MSDN/2018-2009/2009chm/msdnmagazine2009_02en-us/02/Smith.WPFPatterns.0209/chm.htm. – Дата звертання : 16 жовтня 2024 р.
36. Math.NET Numerics. – Режим доступу : <https://numerics.mathdotnet.com/>. – Дата звертання : 16 жовтня 2024 р.

References (transliterated)

1. Koroliouk D., Lyashko S., Limnios N. *Computational Methods and Mathematical Modeling in Cyberphysics and Engineering Applications*. London, Wiley, 2024. – 442 p. DOI:10.1002/9781394284344.
2. Bomba A. Ya., Prysiazhniuk I. M., Prysiazhniuk O. V. Metody teoriiy zburen' prognozuvannya protsesiv teplomasoperenesennya v porystykh ta mikroporystykh seredovyshhakh [Perturbation theory methods for predicting heat and mass transfer processes in porous and microporous media]. Rivne, O. Zen' Publ., 2017. – 291 p.
3. Bomba A., Moroz I. Analysis of Nonlinear Processes in the P-I-N Diodes Plasma by the Perturbation Theory Methods // 2023 13th International Conference on Advanced Computer Information Technologies, ACIT-2023. Wroclaw, Poland. pp. 117–120.
4. Bomba A., Moroz I. Matematychnе modeluvannya dynamichnykh protsesiv v $p-i-n$ diodi metodamy teoriiy zburen' [Simulation of Dynamic Processes in a $p-i-n$ Diode by the Methods of Perturbation Theory]. *Visnyk Natsional'nogo Tekhnichnogo universytetu «KhPI». Seriya : Matematychnе modeluvannya v tekhnitsi ta tekhnologiyakh* [Bulletin of the National Technical University "KhPI". Series: Mathematical modeling in engineering and technologies]. Kharkiv, NTU "KhPI" Publ., 2023, no. 2(5), pp. 23–32. DOI:10.1002/9781394284344.
5. Weinberg S. *Lectures on Quantum Mechanics*. London, Cambridge University Press, 2015. 419 p.
6. Vishik M. I., Lusternik L. A. Reguljarnoe vyrogdienie i pogranichny sloi dlya linejnyh differentsyal'nyh uravnenii s malym parametrom [Regular degeneration and boundary layers for linear differential equations with small parameter]. *UMN [Successes in mathematical sciences]*. 1957, Vol. 12, issue 5 (77), pp. 3–122.
7. Vasil'eva A. B., Butusov V. F., Kalachev L. V. *The Boundary Function Method for Singular Perturbation Problems*. SIAM, Philadelphia, 1995. 212 p. DOI: 10.1137/1.9781611970784.
8. Smith D. R. *Singular-Perturbation Theory. An Introduction with Applications*. Cambridge, Cambridge Univ. Press, 1985. 520 p.
9. Bomba A., Moroz I., Boichura M. Development and analysis of a mathematical model of plasma characteristics in the active region of integrated $p-i-n$ structures by the methods of perturbation theory and conformal mappings. *Skhidno-evropeiskiy zhurnal peredovykh tekhnologiy* [Eastern-European Journal of Advanced Technologies]. 2021, no. 5 (113), pp. 51–61. DOI: 10.15587/1729-4061.2021.243097.
10. Huifang Liu. Perturbation method for differential control equations and its application in the analysis of dynamic characteristics of curved beam. *Alexandria Engineering Journal*. 2020, vol. 59, Iss. 4, pp. 2583–2590. DOI: 10.1016/j.aej.2020.04.022.

11. Dirk Jansen. *The Electronic Design Automation Handbook*. NY, Springer, 2003. 676 p.
12. Maxfield C., Goyal K. EDA (Electronic Design Automation): Where Electronics Begins. TechBITES INTERactive, 2001. 97 p.
13. Norenkov I. P. *Osnovy avtomatizirovannogo proektirovaniya : Ucheb. dlya vuzov* [Fundamentals of computer-aided design : Textbook for universities]. Moscow, Izd-vo MGTU im. N.E. Bauman Publ., 2006. 448 p. (Norenkov I. P. no. 82. *Fundamentals of Computer-Aided Design : Textbook for Universities. 2nd Ed., Reprint. And Additional*. Moscow, Publishing House of the Bauman Moscow State Technical University Publ., 2002.)
14. Evans E. *Domain-Driven Design : Tackling Complexity in the Heart of Software*. Boston, Addison-Wesley Professional, 2003. 560 p.
15. Tkachuk M., Gamzaev R., Martinkus I. Towards Effectiveness Assessment of Domain Modeling Methods and Tools in Software Product Lines Development. *Enterprise Modelling and Information Systems Architecture – International Journal of Conceptual Modeling*. Germany, 2018, Vol. 13 (2018), pp. 190–206. DOI: 10.18471/emisa.hcm.14.
16. Sze S., Kwok K. *Physics of Semiconductor Devices*. New York, Wiley-Interscience, 2006. 815 p. DOI: 10.1002/0470068329.
17. Adirovich E. I., Karageorgii-Alkalaev P. M., Leiderman A. Iu. *Currents Double Injection in Semiconductors*. Moscow, Sov. Radio Publ., 1978. 320 p.
18. Bomba A. Ya., Moroz I. P., Bojchura M. V. The optimization of the shape and size of the injection contacts of the integrated $p-i-n$ structures on the base of using the conformal mapping method. *Radio Electronics, Computer Science, Control*. 2021, no. 1, pp. 14–28. DOI: 10.15588/1607-3274-2021-1-2.
19. Baliga B. *Fundamentals of power semiconductor devices*. NY, Springer, 2010. 1086 p.
20. Polsky B. S., Rimshans J. S. Numerical simulation of transient processes in 2-D bipolar transistor. *Solid State Electron*. 1981, vol. 24, pp. 1081–1085.
21. Bomba A. Ya., Moroz I. P. Matematychno modeluvannya dyfuziyno-dreyfovogo protsesu v aktyvnyi oblasti $p-i-n$ diode z urakhuvanniam rozrivu ta rekombinatsiyi metodamy teoriiy zburen' [Mathematical modeling of the diffusion-drift process in the active region of p-i-n diodes taking into account heating and recombination using perturbation theory methods]. *Zhurnal obchyslival'noyi ta prykladnoyi matematyky* [Journal of Numerical and Applied Mathematics]. 2021, no. 1 (135), pp. 29–35. DOI: <https://doi.org/10.17721/2706-9699.2021.1.03>.
22. *Microsoft Azure Documentation*. Available at : <https://docs.microsoft.com/en-us/azure/> (accessed 15 October 2024).
23. Abadi M., Barham P., Chen J. TensorFlow: A System for Large-Scale Machine Learning. *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16)*. 2016. pp. 265–283.
24. Campbell C., Johnson R., Miller A., Toub S. *Parallel Programming with Microsoft® .NET: Design Patterns for Decomposition and Coordination on Multicore Architectures*. Cambridge, Cambridge University Press, 2010. 224 p.
25. Wolfram S. *Mathematica: A System for Doing Mathematics by Computer. 2nd edition*. Redwood-City, Addison-Wesley, 1991. 961 p.
26. Martinkus I. Evaluation of the Effectiveness of Domain Modeling Methods in Terms of Model Complexity. *International Conference on Information and Communication Technologies in Education, Research, and Industrial Applications*. 2019. pp. 2–4.
27. Evans E. *Domain-Driven Design Reference: Definitions and Pattern Summaries*. Indianapolis: Dog Ear Publishing, 2014. pp. 7–8.
28. Vernon V. *Implementing Domain-Driven Design*. Boston, Addison-Wesley Professional, 2013. 656 p.
29. Dobrovska L. M., Dobrovska I. A. *Teoriya ta praktyka neironnykh merezh: Navchal'nyy posibnyk* [Theory and Practice of Neural Networks: A Tutorial]. Kyiv, NTUU "KPI", Vydavnytstvo "Politekhnika" Publ., 2015. 396 p.
30. Hardesty L. Explained : Neural Networks. *MIT News Office*. April, 2017, no. 14. Available at : <http://news.mit.edu/2017/explained-neural-networks-deep-learning-0414>. (accessed 16 October 2024).
31. Haykin S. *Neural Networks: A Comprehensive Foundation. 2nd ed*. Upper Saddle River, NJ, Prentice Hall, 1999. 842 p.
32. Marinescu F., Avram A. *Domain-Driven Design Quickly*. Lugano, InfoQ Enterprise Software Development Series, 2007. 108 p.
33. Albahari J., Albahari B. *C# 7.0 in a Nutshell: The Definitive Reference*. Sebastopol, O'Reilly Media, 2017. 1088 p.
34. Nathan A. *WPF 4.5 Unleashed*. Indianapolis, Sams Publishing, 2013. 864 p.
35. Smith J. WPF Apps With The Model-View-ViewModel Design Pattern. *MSDN Magazine*. 2009, vol. 24, no. 2. Available at : https://jacobfilipp.com/MSDN/2018-2009/2009chm/msdnmagazine2009_02en-us/02/Smith.WPFPatterns.0209/chm.htm. (accessed 16 October 2024).
36. *Math.NET Numerics*. Available at : <https://numerics.mathdotnet.com/>. (accessed 16 October 2024).

Надійшла (received) 25.10.2024

Відомості про авторів / Information about authors

Бомба Андрій Ярославович – доктор технічних наук, професор, професор кафедри комп'ютерних наук та прикладної математики, Національний університет водного господарства та природокористування, м. Рівне; тел.: (097) 346-18-90; ORCID: <https://orcid.org/0000-0001-5528-4192>; e-mail: abomba@ukr.net.

Bomba Andrii Yaroslavovych – Doctor of Technical Sciences, Professor, Professor at the Department of Computer Science and Applied Mathematics, National University of Water and Environmental Engineering, Rivne; tel.: (097) 346-18-90; ORCID: <https://orcid.org/0000-0001-5528-4192>; e-mail: abomba@ukr.net.

Мороз Ігор Петрович – кандидат фізико-математичних наук, доцент кафедри інформаційних технологій та моделювання, завідувач кафедри інформаційних технологій та моделювання, Рівненський державний гуманітарний університет, м. Рівне; тел.: (096) 721-91-24; ORCID: <https://orcid.org/0000-0001-6381-2266>; e-mail: igor_moroz@yahoo.com.

Moroz Igor Petrovych – PhD, Associate Professor of the Department of Information Technologies and Modeling, Head of the Department of Information Technologies and Modeling, Rivne State Humanities University, Rivne; tel.: (096) 721-91-24; ORCID: <https://orcid.org/0000-0001-6381-2266>; e-mail: igor_moroz@yahoo.com.

Лістєв Захар Сергійович – аспірант кафедри комп'ютерних наук та прикладної математики, магістр, Національний університет водного господарства та природокористування, м. Рівне; тел.: (050) 220-51-60; ORCID: <https://orcid.org/0009-0001-1846-4191>; e-mail: z.c.listiev@nuwm.edu.ua.

Listiev Zakhar Serhiyovych – Graduate student of the Department of Computer Science and Applied Mathematics, Master of Science. National University of Water Management and Nature Management, Rivne; tel.: (050) 220-51-60; ORCID: <https://orcid.org/0009-0001-1846-4191>; e-mail: z.c.listiev@nuwm.edu.ua.