

УДК 378.147:004:51

DOI https://doi.org/10.32782/cusu-pmtp-2026-1-9

ІНТЕГРАЦІЯ ФУНДАМЕНТАЛЬНИХ МАТЕМАТИЧНИХ ДИСЦИПЛІН У ПРАКТИЧНУ ПІДГОТОВКУ ФАХІВЦІВ КОМП'ЮТЕРНИХ НАУК: МЕТОДОЛОГІЧНИЙ АСПЕКТ

Мироненко Оксана Василівна,

кандидат фізико-математичних наук,
доцент кафедри інформаційних технологій
Економіко-технологічного інституту імені Роберта Ельворті
ORCID ID: 0000-0001-8967-0171

Паращук Степан Дмитрович,

кандидат фізико-математичних наук,
доцент кафедри інформаційних технологій
Економіко-технологічного інституту імені Роберта Ельворті
ORCID ID: 0000-0002-8609-3206

Ізвалов Олексій Володимирович,

кандидат технічних наук,
доцент кафедри інформаційних технологій
Економіко-технологічного інституту імені Роберта Ельворті
ORCID ID: 0000-0002-4935-7153

В умовах розвитку AI, GameDev та кібербезпеки виникає потреба у розумінні математичного апарату, прихованого за програмним кодом. Дослідження спрямоване на подолання розриву між академічним викладанням математики та прикладними вимогами IT-індустрії. Робота присвячена аналізу міждисциплінарних зв'язків між розділами та темами фундаментальних математичних дисциплін та структурними елементами фахових дисциплін у підготовці бакалаврів комп'ютерних наук. Розглядається практична імплементація математичного апарату в таких сферах, як Data Science, комп'ютерна графіка, кібербезпека та розробка ігрового штучного інтелекту. Особливу увагу приділено впровадженню розв'язання реальних інженерних кейсів в навчальний процес.

Метою роботи є демонстрація та аналіз проєкції на актуальні IT-технології та у конкретні інженерні навички класичних математичних теорій і ключових розділів математичних дисциплін: елементарна математика та тригонометрія, вища математика (лінійна алгебра та математичний аналіз), теорія ймовірностей, ймовірнісні процеси та математична статистика (окрему увагу приділено використанню законів розподілу: нормальний розподіл, розподіл Пуассона), теорія інформації, дискретна математика (зокрема, теорія графів), теорія чисел і криптографія та чисельні методи.

Результатом дослідження стала розробка матриці практичних інженерних кейсів, яка встановлює пряму відповідність між математичним концептом та IT-технологією. Запропонований підхід дозволяє сформувати у студентів цілісний інженерний світогляд, підвищити мотивацію до вивчення фундаментальних дисциплін та забезпечити підготовку конкурентоспроможних фахівців, здатних створювати та усвідомлено використовувати оптимізовані та надійні програмні продукти.

Ключові слова: IT-освіта, підготовка фахівців комп'ютерних наук, теорія ймовірностей, дискретна математика, теорія графів, лінійне програмування, Big Data, процедурна генерація, теорія ігор, криптографія.

Myronenko Oksana, Parashchuk Stepan, Izvalov Oleksii. Integration of fundamental mathematical disciplines into the practical training of computer science specialists: methodological aspect

In the context of the development of AI, GameDev and cybersecurity, there is a need to understand the mathematical apparatus hidden behind the program code. The research is aimed at bridging the gap between academic teaching of mathematics and the applied requirements of the IT industry. The work is devoted to the analysis of interdisciplinary connections between sections and topics of fundamental mathematical disciplines and structural elements of professional disciplines in the training of bachelors of computer science. The practical implementation of mathematical tools in such areas as Data Science, computer graphics, cybersecurity, and the development of gaming artificial intelligence is considered. Special attention is paid to the implementation of solving real engineering cases in the educational process.

The aim of the work is to demonstrate and analyze the projection onto current IT technologies and into specific engineering skills of classical mathematical theories and key sections of mathematical disciplines: elementary mathematics and trigonometry, higher mathematics (linear algebra and mathematical analysis), probability theory, probabilistic processes and mathematical statistics (special attention is paid to the use of distribution laws: normal distribution, Poisson distribution), information theory, discrete mathematics (in particular, graph theory), number theory and cryptography and numerical methods.

The result of the research was the development of a matrix of practical engineering keys, which establishes a direct correspondence between a mathematical concept and IT technology. The proposed approach allows students to form a holistic engineering worldview, increase motivation to study fundamental disciplines and ensure the training of competitive specialists who are able to create and consciously use optimized and reliable software products.

Key words: *IT education, training of computer science specialists, probability theory, discrete mathematics, graph theory, linear programming, Big Data, procedural generation, game theory, cryptography.*

Вступ. Сучасна парадигма ІТ-освіти вимагає відходу від абстрактного викладання математики. Студенти часто не бачать прямого зв'язку між сухими формулами та написанням програмного коду. Актуальність інтеграції математичних дисциплін у підготовці фахівців комп'ютерних наук зумовлена стрімким розвитком інформаційних технологій, зростанням вимог до аналітичних навичок та здатності працювати з великими обсягами даних.

Сучасний ринок праці потребує спеціалістів, які володіють не лише практичними навичками програмування, а й здатні застосовувати математичні методи для моделювання, оптимізації та прогнозування. Тому питання інтеграції математики в ІТ-освіті є надзвичайно актуальним та важливим для забезпечення конкурентоспроможності випускників та їхньої готовності до викликів цифрової економіки.

Аналіз досліджень і публікацій. Проблему підготовки фахівців ІТ-сфери, що потребує вдосконалення математичної складової, рівня сформованості інформаційно-цифрової компетентності [5; 6], розширення і поглиблення вивчення математичних дисциплін розглядали автори робіт [2, 7–9]. Методику дослідження міждисциплінарних зв'язків при підготовці фахівців інженерних спеціальностей наведено в роботах [1; 3; 4]. Зокрема, вдосконалення навчальних програм математичних дисциплін для розвитку креативного мислення фахівців комп'ютерних наук розглядає група авторів Титова О., Лузан П., Сосницька Н., Кулешов С., Супрун О. [10]. Але не прослідковуються лінійні зв'язки безпосередньо з розділами базових математичних дисциплін, що входять до освітньо-професійної програми бакалавра спеціальностей 122–Комп'ютерні науки та 123–Комп'ютерна інженерія.

Проведений аналіз досліджень і публікацій показав, що цей напрямок дослідження взаємозв'язків між фундаментальними та фаховими дисциплінами ставить нові вимоги до ІТ-освіти.

Матеріали та методи. Методи аналізу і синтезу та порівняння застосовані до розгляду трансформації класичних розділів математики у конкретні навички програміста та результати його праці.

1. Елементарна математика та тригонометрія: кінематика віртуальних світів

Часто студенти молодших курсів ігнорують шкільну програму тригонометрії, вважаючи її "пройденим етапом". Проте у розробці ігор (GameDev) тригонометричні функції є основою

руху та взаємодії об'єктів. На відміну від абстрактних задач на доведення тотожностей, у програмуванні тригонометрія вирішує задачу перетворення кутів у вектори руху і навпаки. Рух об'єкта, тобто реалізація переміщення персонажа під певним кутом неможлива без розуміння проєкцій вектора швидкості на осі координат: $v_x = v \cdot \cos(\alpha)$ та $v_y = v \cdot \sin(\alpha)$.

Функція `atan2` (арктангенс двох аргументів (`Math.atan2(y, x)`)) є однією з найбільш використовуваних функцій у стандартних бібліотеках мов (C++, C#, Python, JS). Вона дозволяє визначити кут повороту об'єкта (наприклад, гармати танка або погляду персонажа) до цілі за координатами миші, автоматично враховуючи знаки чвертей координатної площини, що рятує код від громіздких умов `if-else`.

У процедурній анімації для створення ефекту "дихання" інтерфейсу, плавання об'єктів у воді, розрахунку віддачі зброї (`recoil patterns`) або генерації хвиль використовуються синусоїдальні функції (`sin(time)`). Розуміння періоду, амплітуди та частоти коливань дозволяє програмісту створювати "живі" анімації без використання важких графічних асетів.

2. Вища математика

Для повноцінної підготовки Computer Science фахівця (особливо в контексті тем 3D-моделювання, штучний інтелект (ШІ) та Data Science) критично важливим є обговорення таких фундаментальних розділів (стовпів) Вищої математики, як лінійна алгебра та математичний аналіз (Calculus).

Лінійна алгебра є найголовнішим елементом для 3D-моделювання та комп'ютерної графіки. Комп'ютерна графіка неможлива без матриць, векторів, кватерніонів та лінійних перетворень (обертання, масштабування, переміщення). В ML/AI: Робота нейронних мереж – це по суті множення величезних матриць. Без розуміння власних векторів та власних чисел (`eigenvectors/eigenvalues`) неможливо зрозуміти, як працює, наприклад, метод головних компонент (PCA) для зменшення розмірності даних.

В сучасному Machine Learning оптимізація базується на таких напрямках математичного аналізу (Calculus), як знаходження градієнта. Поняття похідної, градієнтного спуску (Gradient Descent), часткових похідних – це двигун, який навчає нейромережі. В свою чергу у GameDev фізичні рушії ігор базуються на диференціальних рівняннях та інтегруванні (наприклад, рух тіл та балістика).

3. Теорія ймовірностей та математична статистика

Одним із найбільш затребуваних напрямків сьогодні є робота з базами даних. У ході дискусії було визначено, що базові поняття теорії ймовірностей є критичними для розуміння Big Data та систем масового обслуговування.

Такі числові характеристики випадкових величин, як поняття моди, медіани, дисперсії та середнього квадратичного відхилення не є просто абстрактними величинами. Для розробників сервісів на кшталт YouTube або Netflix це інструменти аналізу «утримання аудиторії» та аналізу поведінки користувачів. Наприклад, аналіз квантилів дозволяє відсіяти аномально короткі або довгі сесії перегляду, допомагає точніше налаштувати рекомендаційні алгоритми (рис. 1).

У комп'ютерній графіці та 3D-моделюванні широко застосовуються закони розподілу випадкових величин. Окремий інтерес становить застосування законів розподілу випадкових величин у процедурній генерації контенту (PCG).

Рівномірний розподіл - найпростіший тип випадковості (`rand()`), який, однак, виглядає неприродно при створенні органічних об'єктів.

Нормальний розподіл (Гауса) використовується для генерації ландшафтів (висоти гір) та розкиду характеристик ігрових персонажів.

Розподіл Пуассона є основою для вирішення проблеми з наведеного прикладу задачі моделювання «посипки на пончику» (Blender Guru tutorial context). Рівномірний розподіл призводить до злипання об'єктів та утворення порожнеч. Натомість використання алгоритмів,

заснованих на розподілі Пуассона (Poisson Disk Sampling), дозволяє досягти рівномірного, але випадкового покриття поверхні, що є стандартом у сучасній комп'ютерній графіці (рис. 2).

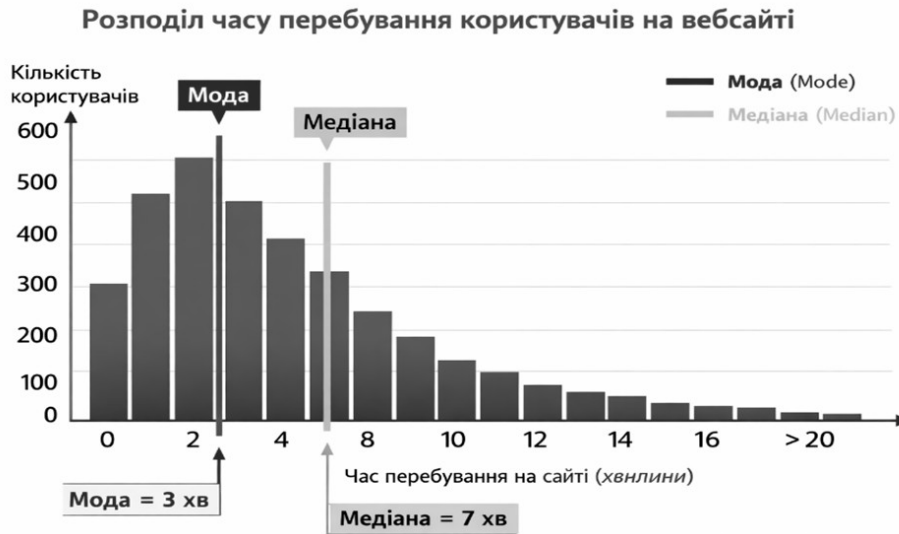


Рис. 1. Гістограма розподілу часу перебування користувачів на вебсайті із зазначенням моди та медіани

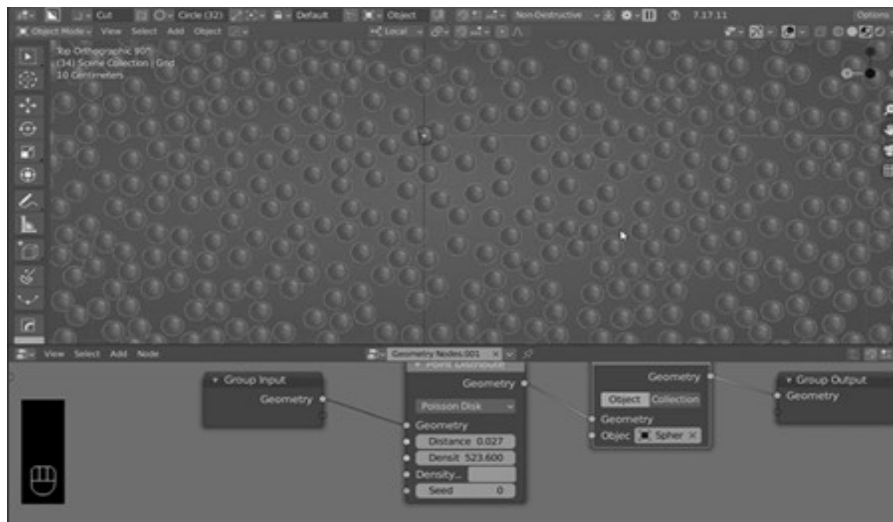


Рис. 2. Розподіл об'єктів у програмі 3D-моделювання Blender за диском Пуассона

Випадкові процеси та теорія масового обслуговування знайшли своє застосування у роботі комп'ютерних мереж. Так, стаціонарні випадкові процеси є математичною моделлю для проектування архітектури серверів. Задача оцінки необхідних потужностей для вебсайту, щоб він витримав пікові навантаження (DDoS-атаки або «ефект Чорної п'ятниці»), вирішується методами теорії масового обслуговування. Потік заявок на сервер моделюється як найпростіший потік подій (потік Пуассона).

Регресійний аналіз та кореляційні зв'язки необхідні для моделювання, зокрема, у бізнес-інформатиці. Вивчаючи дисципліну «Бізнес-інформатика», студенти використовують кореляційно-регресійний аналіз для прогнозування реальних процесів. В якості прикладу наведено практичну задачу з аналізу даних сервісів велопрокату: залежність попиту від локації станції, погоди та часу доби дозволяє оптимізувати логістику переміщення велосипедів (електричних самокатів) між станціями.

4. Теорія інформації

Теорія інформації – місток між ймовірністю та Data Science. Сучасна підготовка ІТ-фахівця неможлива без основ теорії інформації Клода Шеннона. Ця дисципліна демонструє, як логарифми та ймовірності трансформуються у ефективність зберігання та передачі даних.

Вивчення теорії інформації надає студентам розуміння поняття ентропії, що дозволяє їм зрозуміти межі стиснення інформації. Алгоритми архівації (ZIP) або кодування медіа (JPEG, MP3) базуються на видаленні надлишковості даних. Розуміння алгоритму Хаффмана (Huffman coding) є чудовим прикладом застосування бінарних дерев та ймовірнісного підходу для мінімізації обсягу пам'яті.

У контексті штучного інтелекту теорія інформації пояснює, як мережа «вчиться». Функція втрат перехресної ентропії (Cross-Entropy Loss), яка є стандартом для задач класифікації в Deep Learning, фактично вимірює відстань між передбаченим розподілом ймовірностей та реальним. Без розуміння логарифмічної природи цієї метрики студенту важко зрозуміти поведінку градієнтного спуску при навчанні моделі.

5. Дискретна математика: логіка, множини та криптографія

Комп'ютерна дискретна математика формує «алгоритмічний скелет» мислення будь-якого розробника, зокрема розробника програмного забезпечення. Розглянемо деякі її розділи в світлі комп'ютерних технологій.

Теорія множин є основою у базах даних та 3D-геометрії. Операції над множинами мають пряму проєкцію в інженерні задачі. Зокрема, SQL-запити, такі як операції JOIN, UNION, INTERSECT є прямою реалізацією теоретико-множинних операцій та їх законів. У 3D-моделюванні булеві операції над множинами (об'єднання, різниця, перетин) використовуються для створення складних форм з примітивів CSG (Constructive Solid Geometry). Наприклад, щоб зробити отвір у стіні, необхідно від об'єкта «стіна» відняти об'ємний об'єкт «паралелепіпед».

Невід'ємною частиною кодування є булева алгебра та алгоритми оптимізації коду. Розуміння законів де Моргана, диз'юнктивних та кон'юнктивних нормальних форм дозволяє спрощувати складні логічні умови в коді (if, while). Це не лише покращує читабельність коду (Clean Code), але й, у випадку низькорівневого програмування або розробки під мікроконтролери, суттєво економить такти процесора.

Комбінаторика та основи кібербезпеки. Комбінаторний аналіз є базою для розуміння стійкості криптографічних систем. На практичних заняттях студенти розраховують ентропію паролів: перехід від алфавіту з 26 символів до алфавіту з цифрами та спеціальними символами (наприклад, 72 символи) при довжині пароля L збільшує складність перебору від 26^L до 72^L . Це наочно демонструє студентам складність коду та важливість політики безпеки.

Теорія графів є одним з найбільш застосовних розділів комп'ютерної дискретної математики у різноманітних напрямках Computer Science.

Розділи теорії графів, зокрема маршрути та ваги на графах, є фундаментом роботи нейронних мереж у штучному інтелекті. Так, алгоритми пошуку шляху є первинною основою від класичної задачі комівояжера до алгоритмів A^* (A-star), адже все це базується на зважених графах. Студенти реалізують ці алгоритми як для логістичних задач (навігатори, доставка), так і для переміщення юнітів у стратегічних іграх.

Деревні конфігурації, T-факторизації, проходження гілок, рівнів та розгалужень на деревах дає можливість розуміти і знаходити всі можливі варіанти рішень практичних задач та ігрових ситуацій. Участь студентів у різноманітних конкурсах, таких як IBM "Ponder This", часто вимагає нестандартного використання графів. Зокрема, деревні конфігурації та генерація лабіринтів були згадані в задачі, де генерація ідеального лабіринту зводилася до побудови кістякового дерева (spanning tree) на графі клітин. Також дерева рішень (Game Trees) є основою для ство-

рення ботів у іграх з повною інформацією таких, як «Хрестики-нулики», візуалізація дерева рішень якої представлена на рис. 3, шахи та інші.

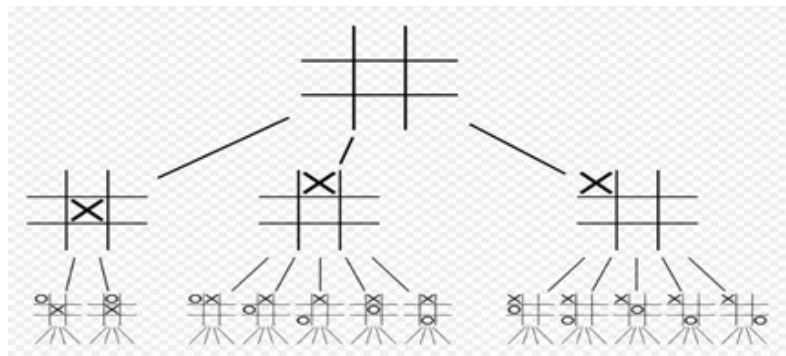


Рис. 3. Дерево рішень для гри «Хрестики-нулики»

5. Дослідження операцій та методи оптимізації

В комп'ютерному програмуванні широко використовується математичне моделювання ситуацій, де необхідно знайти оптимальне (найкраще) рішення при обмежених ресурсах (вихідних даних). Лінійне програмування дозволяє вирішувати безліч практичних задач, зокрема економічних та побутових.

Класична «задача про наплічник (рюкзак)» (Knapsack problem) або транспортна задача розглядаються не лише теоретично. Студенти вчаться розв'язувати їх декількома шляхами:

- 1) аналітично, використовуючи логічні міркування та методи лінійного програмування (симплекс-метод і метод потенціалів);
- 2) інструментально, використовуючи надбудову для пошуку розв'язку «Розв'язувач» (Solver) в MS Excel для швидкого прототипування;
- 3) програмно, реалізуючи симплекс-метод або жадібні алгоритми мовами Python та C++.

Пакет завдань для студентів також включає задачу про призначення (Assignment problem) – оптимальний розподіл співробітників по задачах на основі їх ефективності, що є актуальним для менеджменту ІТ-проектів.

Теорія ігор та мультиагентні системи. На старших курсах (зокрема, в рамках дисципліни «Штучний інтелект») теорія ігор переходить у практичну площину. Студенти розробляють інтелектуальних агентів для ігор. Навчальний процес будується поетапно:

- I. Створення бота, що робить випадкові ходи (Random Walker).
- II. Створення бота, що аналізує поточну ситуацію (Greedy strategy).
- III. Створення бота, що прораховує ходи наперед (Minimax algorithm).

Фіналом навчання є проведення турніру між програмами студентів, що додає елемент гейміфікації в освітній процес (рис. 4).

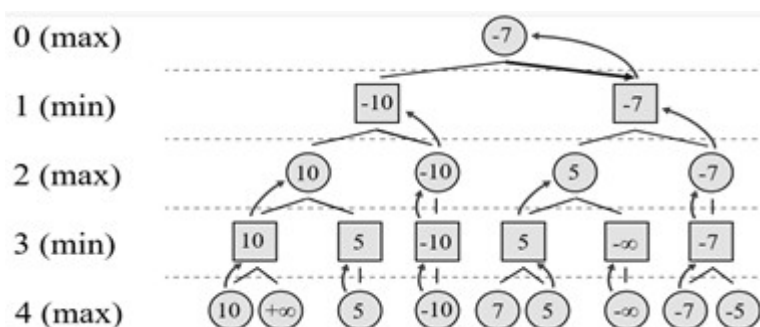


Рис. 4. Приклад схеми роботи алгоритму Мінімаксу

6. Чисельні методи

Комп'ютери не працюють з дійсними числами ідеально точно (проблема $0.1 + 0.2 \neq 0.3$). Ця особливість збереження дійсних чисел у бінарному форматі із обмеженням довжини мантиси можна проілюструвати навіть у Excel і для студентів нетехнічних спеціальностей (маркетинг, менеджмент). Розглядаємо уявну ситуацію: у банку акція – кожен день гроші на рахунку збільшуються в 11 разів, а банк за це бере одну гривню. Пропонуємо студентам розрахувати, як зміниться сума вкладу протягом місяця, якщо спочатку вона становила 10 копійок (0.1 гривні). Для цього у комірку A1 вводиться число 0.1, а у комірку A2 вводиться формула: $= A1 * 11 - 1$. З чисто математичної точки зору сума не повинна змінюватися, оскільки $0.1 * 11 - 1 = 0.1$. Зміни також і не помітні, якщо склонувати формулу з комірки A2 на кілька рядків нижче. Проте, при подальшому клонуванні похибка стане все більш явною (рис.5).

	A	10	0.100000019	20	493.9198573
1	0.1	11	0.100000209	21	5432.11843
2	0.1	12	0.100002304	22	59752.30274
3	0.1	13	0.100025341	23	657274.3301
4	0.1	14	0.100278748	24	7230016.631
5	0.1	15	0.103066233	25	79530181.94
6	0.1	16	0.133728561	26	874832000.3
7	0.1	17	0.471014168	27	9623152003
8	0.1	18	4.181155846	28	1.05855E+11
9	0.100000002	19	44.9927143	29	1.1644E+12
10	0.100000019	20	493.9198573	30	1.28084E+13

Рис. 5. Накопичення похибки протягом 30 кроків

Тому важливим є розуміння похибок округлення, апроксимації функцій, інтерполяції – це те, що відрізняє інженера від кодера-початківця, особливо в наукомісткому софті.

7. Теорія чисел.

Якщо раніше теорія чисел вважалася "найчистішою" та найбільш абстрактною галуззю математики, то в епоху блокчейну та HTTPS вона стала основою цифрової безпеки.

Модульна арифметика: Операція знаходження залишку від ділення (modulo operation, %) є ключовою для побудови циклічних груп. Обмін ключами Діффі-Гелмана, що захищає з'єднання у месенджерах, базується на складності задачі дискретного логарифмування у скінченних полях. Студенти мають усвідомити, що безпека їхніх банківських транзакцій тримається на складності обернених математичних операцій.

Еліптичні криві (ECC): Сучасні стандарти шифрування відходять від класичного RSA у бік еліптичних кривих. Тут алгебраїчна геометрія зустрічається з теорією полів. Розуміння групового закону додавання точок на кривій дозволяє створювати ключі, які при меншій довжині (і менших витратах процесорного часу смартфонів) забезпечують вищу стійкість, ніж факторизація великих чисел.

Геш-функції та колізії: Принцип Діріхле ("задача про кроликів та клітки") з комбінаторики знаходить пряме відображення у розумінні колізій геш-функцій. Це критично важливо як для структур даних (Hash Map), так і для цілісності даних у блокчейн-системах (Proof-of-Work).

Результати. За результатами дослідження побудована матриця практичних інженерних кейсів за розділами математики (табл. 1), які впроваджуються у навчальний процес Економіко-технологічного інституту імені Роберта Ельворті та пропонуються для впровадження у інших вищих навчальних закладах ІТ-спрямування.

Таблиця 1

Математичний концепт	Технології IT/сфера застосування	Практичний інженерний кейс
Елементарна математика		
Тригонометричні функції (sin, cos, arctg)	GameDev, UI/UX	Розрахунок польоту снаряда (балістика), поворот гармати танка за курсором миші, створення плавної анімації інтерфейсу ("дихання" кнопки).
Інтерполяція (Лінійна, Безьє)	Комп'ютерна графіка	Плавний рух камери, побудова кривих у векторних редакторах (Figma, Adobe Illustrator).
Лінійна алгебра		
Матриці та операції над ними	3D-графіка (OpenGL/DirectX), Нейромережі	Трансформація об'єктів у просторі (поворот, масштаб). У ML – це шар нейромережі (множення ваг на входи).
Вектори (нормалізація, скалярний добуток)	GameDev, Пошукові системи	Визначення, чи бачить ворог гравця (кут між векторами). Розрахунок освітлення (Lambert shading). Пошук схожості текстів (Cosine similarity).
Власні числа та вектори (Eigenvalues)	Data Science, Google PageRank	Алгоритм PCA (стиснення даних зі збереженням суті), ранжування веб-сторінок у пошуку.
Математичний аналіз		
Похідна та градієнт	Machine Learning (Deep Learning)	Алгоритм зворотного поширення помилки (Backpropagation) – навчання нейромережі через мінімізацію функції втрат.
Чисельне інтегрування	Фізичні рушії (PhysX, Havok)	Розрахунок позиції тіла на основі швидкості та прискорення у кожному кадрі гри.
Теорія ймовірностей та статистика		
Нормальний розподіл (Гауса)	Game Design, Процедурна генерація	Генерація реалістичного ландшафту (висоти гір), розкид характеристик персонажів (щоб не всі були однакові).
Розподіл Пуассона	DevOps, Архітектура серверів	Моделювання черги запитів до веб-сервера, захист від DDoS (виявлення аномалій у трафіку).
Байсова ймовірність	Кібербезпека, ШІ	Робота спам-фільтрів (ймовірність, що лист є спамом за наявності певних слів).
Дискретна математика		
Теорія графів (пошук шляху, A* алгоритм Дейкстри)	Логістика, GameDev (AI)	Побудова маршруту в Google Maps або навігаторі. Пошук шляху ботом у лабіринті гри.
Дерева (Бінарні, B-дерева, Quadtrees)	Бази даних, 3D-оптимізація	Індексація в SQL (швидкий пошук). Відсікання невидимих об'єктів у іграх (Frustum Culling) для підняття FPS.
Булева алгебра та бітові операції	Системне програмування, Embedded	Робота з правами доступу (Unix chmod), маски мереж (Subnetting), оптимізація пам'яті на мікроконтролерах.
Теорія чисел та криптографія		
Модульна арифметика (залишки від ділення)	Криптографія, Блокчейн	Обмін ключами Діффі-Гелмана, генерація адрес криптогаманців, робота хеш-таблиць (Hash Map).
Еліптичні криві	Сучасний захист (HTTPS, TLS)	Шифрування повідомлень у месенджерах (Signal, WhatsApp), цифрові підписи.
Теорія інформації		
Ентропія Шеннона	Data Science, Стиснення даних	Алгоритми архівації (ZIP, RAR). Побудова дерев рішень у ML (вибір найкращого запитання для поділу даних).
Відстань Левенштейна	NLP (Обробка тексту)	Перевірка орфографії (автокорекція), пошук нечітких дублікатів у базах даних.
Чисельні методи		
Плаваюча точка (IEEE 754) та похибки	FinTech (Банківський софт)	Розуміння, чому $0.1 + 0.2 \neq 0.3$. Запобігання втраті грошей при транзакціях через помилки округлення.

Висновки. Інтеграція математичної підготовки з курсами програмування (C++, Python, JS) забезпечує формування цілісного інженерного світогляду. Студенти, які розуміють математичну природу алгоритмів (від теорії ймовірностей у генерації зображень до теорії графів у побудові маршрутів), здатні створювати більш ефективний, оптимізований та надійний програмний продукт. Математика перестає бути абстракцією і стає потужним інструментом у руках IT-фахівця.

Література:

1. Наконечна, Т.В., Нікулін, О.В. Використання семантичних мереж при підготовці фахівців. *Питання прикладної математики і математичного моделювання*. 2022. Вип. 22, с. 113–125. <https://doi.org/10.15421/322212>
2. Паращук С.Д., Неділько В.М., Мироненко О.В., Цаплін О. О. Основи дослідження операцій: принципи та математичні методи. *Журнал «Наука і техніка сьогодні» (Серія «Техніка»)*. 2025. № 4(45), с. 1426–1434. [https://doi.org/10.52058/2786-6025-2025-4\(45\)-1426-1434](https://doi.org/10.52058/2786-6025-2025-4(45)-1426-1434)
3. Петрук, В., Семеніхіна, О., Сабадощ, Ю. Нові підходи до статистичного аналізу результатів педагогічного експерименту. *Фізико-математична освіта*. 2022. т.33(1), с. 36–42. <https://doi.org/10.31110/2413-1571-2022-033-1-006>
4. Семеріков С. О. Теоретико-методичні основи фундаменталізації навчання інформатичних дисциплін у вищих навчальних закладах : дис. ... д-ра пед. наук : 13.00.02 : Київ, 2009. 536 с. <https://doi.org/10.31812/0564/593>
5. Трифонова О. М. Визначення рівня сформованості інформаційно-цифрової компетентності у майбутніх фахівців комп'ютерних технологій. *Наукові записки. Серія : Педагогічні науки*. Кропивницький, 2019, Вип. 177(2). с. 128–135. [http://nbuv.gov.ua/UJRN/Nz_p_2019_177\(2\)_30](http://nbuv.gov.ua/UJRN/Nz_p_2019_177(2)_30)
6. Трифонова О. М. Методична система розвитку інформаційно-цифрової компетентності магістрів комп'ютерних технологій. *Наукові записки. Серія: Педагогічні науки*. 2019. Вип. 185, с. 174–179. <https://doi.org/10.36550/2415-7988-2019-1-185-174-179>
7. Astafieva M., Lytvynova S., Proshkin V. Mathematical Modeling as a Tool for Interdisciplinary Training of Future Computer Science and Cybersecurity Specialists. CEUR-WS.org, 3187, 2021. P. 103–116. <https://ceur-ws.org/Vol-3187/>
8. Myronenko O.V., Izvalov O.V. Fundamental mathematical disciplines as the basis of training computer science specialists. Scientific-Practical Conference: “Promising areas of theoretical and applied research ‘2025””, USA, Seattle, November, 2025, p. 41–45. <https://www.proconference.org/index.php/usc/issue/view/usc34-00/usc34-00>
9. Roger G. Hadgraft, Anette Kolmos. Emerging learning environments in engineering education. *Australasian Journal of Engineering Education*. 2020. Vol. 25. Is. 1. p. 3–16. <https://doi.org/10.1080/22054952.2020.1713522>
10. Titova O., Luzan P., Sosnytska N., Kulieshov S., Suprun O. Information and Communication Technology Tools for Enhancing Engineering Students' Creativity. In: Ivanov V., Trojanowska J., Pavlenko I., Zajac J., Peraković D. (eds) *Advances in Design, Simulation and Manufacturing IV*. DSMIE 2021. Lecture Notes in Mechanical Engineering. Springer, Cham. https://doi.org/10.1007/978-3-030-77719-7_33

References:

1. Nakonechna, T.V., & Nikulin, O.V. (2022). Vykorystannya semantychnykh merezh pry pidhotovtsi fakhivtsiv. Pytannya prykladnoyi matematyky i matematychnoho modelyuvannya. [The use of semantic networks in the training of specialists]. *Issues of applied mathematics and mathematical modeling*, 22, 113–125. <https://doi.org/10.15421/322212> [in Ukrainian]
2. Parashchuk, S.D., Nedilko, V.M., Myronenko, O.V., & Tsaplin, O. O. (2025). Osnovy doslidzhennya operatsiy: pryntsyipy ta matematychni metody. [Fundamentals of operations research: principles and mathematical methods]. *Journal "Science and Technology Today" (Technology Series)*, 4(45), 1426–1434. [https://doi.org/10.52058/2786-6025-2025-4\(45\)-1426-1434](https://doi.org/10.52058/2786-6025-2025-4(45)-1426-1434) [in Ukrainian]
3. Petruk, V., Semenikhina, O., & Sabadosh, Yu. (2022). Novi pidkhody do statystychnoho analizu rezul'tativ pedahohichnoho eksperymentu. [New approaches to statistical analysis of pedagogical experiment results]. *Physical and mathematical education*, 33(1), 36–42. <https://doi.org/10.31110/2413-1571-2022-033-1-006> [in Ukrainian]
4. Semerikov, S. O. (2009). Teoretyko-metodychni osnovy fundamentalizatsii navchannia informatychnykh dystsyplin u vyshchyykh navchalnykh zakladakh [Theoretical and methodological foundations of fundamentalization of teaching of computer science disciplines in higher educational institutions]. *Doctor's thesis*. 536 p. Kyiv, 2009. 536 s. <https://doi.org/10.31812/0564/593> [in Ukrainian]
5. Tryfonova, O. M. (2019). Vyznachennia rivnia sformovanosti informatsiino-tsyfrovoi kompetentnosti u maibutnykh fakhivtsiv kompiuternykh tekhnolohii [Determining the level of information and digital competence in future computer technology specialists]. *Naukovi zapysky. Serii: Pedahohichni nauky*. Vyp. 177(2). p. 128–135. [http://nbuv.gov.ua/UJRN/Nz_p_2019_177\(2\)_30](http://nbuv.gov.ua/UJRN/Nz_p_2019_177(2)_30) [in Ukrainian]
6. Tryfonova, O. M. (2019). Metodychna systema rozvytku informatsiino-tsyfrovoi kompetentnosti mahistriv kompiuternykh tekhnolohii [Methodological system for the development of information and digital

competence of computer technology masters]. *Naukovi zapysky. Serii: Pedagogichni nauky*. Vyp. 185, p. 174–179. <https://doi.org/10.36550/2415-7988-2019-1-185-174-179.2> [in Ukrainian]

7. Astafieva, M., Lytvynova, S., & Proshkin, V. (2021). Mathematical Modeling as a Tool for Interdisciplinary Training of Future Computer Science and Cybersecurity Specialists. *CEUR-WS.org*, 3187, 118–132. <https://ceur-ws.org/Vol-3187/>

8. Myronenko, O.V., & Izvalov, O.V. (2025). Fundamental mathematical disciplines as the basis of training computer science specialists. *Scientific-Practical Conference: Promising areas of theoretical and applied research '2025, USA, Seattle, November*, p. 41–45. <https://www.proconference.org/index.php/usc/issue/view/usc34-00/usc34-00>

9. Roger, G. Hadgraft, Anette, Kolmos. (2020). Emerging learning environments in engineering education. *Australasian Journal of Engineering Education*, Vol. 25, Is.1, p. 3–16. <https://doi.org/10.1080/22054952.2020.1713522>

10. Titova, O., Luzan, P., Sosnytska, N., Kulieshov, S., & Suprun, O. (2021). Information and Communication Technology Tools for Enhancing Engineering Students' Creativity. In: Ivanov V., Trojanowska J., Pavlenko I., Zajac J., Peraković D. (eds) *Advances in Design, Simulation and Manufacturing IV*. DSMIE. *Lecture Notes in Mechanical Engineering*. Springer, Cham. https://doi.org/10.1007/978-3-030-77719-7_33

Дата першого надходження статті до видання: 28.01.2026

Дата прийняття статті до друку після рецензування: 23.03.2026

Дата публікації (оприлюднення) статті: 21.04.2026