

УДК 004.2:519.6

DOI [https://doi.org/10.24144/2616-7700.2025.47\(2\).231-243](https://doi.org/10.24144/2616-7700.2025.47(2).231-243)О. В. Попов¹, А. В. Павлюк²

¹ Інститут кібернетики ім. В.М. Глушкова НАН України,
провідний науковий співробітник відділу чисельних методів
та комп'ютерного моделювання,
доктор фізико-математичних наук
alex50popov@gmail.com
ORCID: <https://orcid.org/0000-0002-1217-2534>

² Інститут кібернетики ім. В.М. Глушкова НАН України,
аспірант
kvadrump1@gmail.com
ORCID: <https://orcid.org/0009-0002-4166-2003>

ВАРІАЦІЙНО-ОПЕРАТОРНА МОДЕЛЬ КЕРОВАНОЇ ДИСКРЕТИЗАЦІЇ ДЛЯ ГІБРИДНИХ ОБЧИСЛЕНЬ

У статті розроблено варіаційно-операторний підхід до керованої дискретизації, спрямований на узгодження точності математичної моделі з архітектурною ефективністю гібридних обчислювальних систем. Запропоновано параметризований оператор, який адаптивно враховує локальні властивості процесу та зберігає енергетичну узгодженість між неперервною і дискретною формами задачі. Побудовано двоетапну схему оптимізації, що поєднує точність апроксимації з мінімізацією часу виконання обчислень. Експериментальні результати підтверджують стабільність, енергетичну збалансованість і високу відповідність між прогнозованими та фактичними характеристиками, що засвідчує ефективність методу для моделювання складних гібридних систем.

Ключові слова: керована дискретизація, CPU–GPU архітектура, гібридні обчислення, варіаційна оптимізація, енергетична узгодженість, обчислювальна ефективність, моделювання складних систем.

1. Вступ. У контексті сучасного обчислювального моделювання гібридних систем, що поєднують континуальні фізичні процеси з дискретними подіями керування, постає фундаментальна проблема узгодження математичної точності з архітектурною ефективністю. При переході до гібридних CPU–GPU архітектур це завдання набуває принципово нового змісту: просторово-часова дискретизація операторів має не лише забезпечувати апроксимаційну збіжність, а й відповідати енергетичним інваріантам системи та топології обчислювального графа.

Класичні схеми дискретизації, зокрема методи скінченних різниць і скінченних елементів фіксованого порядку, не враховують локальної варіації спектральних властивостей оператора L , який описує просторово-часову динаміку системи. Унаслідок цього їхня апроксимаційна точність не адаптується до локальної структури поля $u(x, t)$, а ефективність реалізації визначається апаратною пропускнуою здатністю пам'яті лише на етапі виконання, без участі самої дискретизаційної схеми.

Зауваження 1. Далі під полем $u(x, t)$ розуміємо функцію стану, що описує розподіл фізичної або модельної величини у просторі та часі; термін структура поля використовується для позначення локальної просторово-часової організації або спектральних властивостей $u(x, t)$.

У результаті така фіксована дискретизація створює дисбаланс між локальною точністю апроксимації та ефективністю обчислень, оскільки єдина сітка не здатна адаптуватися до областей із різкими змінами просторово-часових масштабів. Цей дисбаланс проявляється як порушення енергетичної узгодженості між континуальною моделлю та її чисельною апроксимацією, що призводить до появи штучних дисипативних або генеративних компонент у спектрі розв'язку й, відповідно, до спотворення фізично коректної динаміки системи [1–2].

Для усунення цих обмежень у даній роботі запропоновано метод керованої дискретизації, який розглядає побудову апроксимаційного оператора як задачу оптимізації в параметричному просторі:

$$\theta = (p, r, \varphi), \quad (1)$$

де p — порядок апроксимації; r — радіус локального шаблону апроксимації; φ — тип фільтраційної схеми, що контролює спектральну характеристику (селективність) оператора L .

На відміну від класичних схем дискретизації, параметри θ у (1) не задаються фіксовано, а визначаються з урахуванням локальних характеристик функції стану $u(x, t)$, зокрема просторової гладкості, чутливості до збурень і часової жорсткості. Така адаптація дозволяє реалізувати енергетично узгоджені дискретизації, у яких збережено головні інваріанти системи при мінімізації витрат ресурсів для обчислення на гібридних архітектурах [3].

У межах варіаційно-операторного підходу дискретизація трактується як оптимізаційна процедура наближення оператора L у просторі дволінійних форм, узгоджених з енергетичними характеристиками системи. У цьому контексті побудова параметризованого дискретного оператора $L_h(\theta)$ спрямована не лише на мінімізацію норми відхилення $\|L - L_h(\theta)\|$, але й на збереження коерцитивності (позитивної визначеності енергетичної форми), дисипативності та коректної енергетичної балансової структури. Відповідно, керована дискретизація розглядається як варіаційний механізм узгодження аналітичних, чисельних та апаратних інваріантів, який формує теоретичну основу для побудови енергетично збалансованих сплітів та адаптивного планування обчислень у гібридних CPU–GPU середовищах [4].

2. Варіаційно-операторний підхід до керованої дискретизації. Дискретний оператор у межах варіаційно-операторного підходу визначається як варіаційне наближення безперервного оператора в енергетичній нормі, що забезпечує узгодженість між аналітичною моделлю та її чисельним представленням.

Нехай оператор $L : V \rightarrow V'$ визначено в гільбертовому просторі V з відповідним енергетичним скалярним добутком $\langle \cdot, \cdot \rangle_E$, а його дискретна апроксимація задається у вигляді параметризованого відображення:

$$L_h(\theta) : V_h(\theta) \rightarrow V'_h(\theta), \quad (2)$$

де $V_h(\theta)$ — параметризований підпростір апроксимації, породжений системою локальних базисних функцій $\{\phi_j^{(\theta)}\}_{j=1}^{N_h^{(\theta)}}$, де кожна функція $\phi_j^{(\theta)}$ асоційована з вузлом або коміркою обчислювальної сітки, а $N_h^{(\theta)}$ — розмірність апроксима-

ційного підпростору (кількість базисних функцій) для вибраної конфігурації θ (1).

Враховуючи (2), дискретну форму оператора можна визначити матрицею:

$$[L_h(\theta)]_{ij} = \left\langle L\phi_j^{(\theta)}, \phi_j^{(\theta)} \right\rangle_E, \quad (3)$$

що дозволяє зберегти енергетичну когерентність між неперервною та дискретною моделями.

У (3) оператор $L_h(\theta)$ наближає L не лише у метриці норми $\|L - L_h(\theta)\|$, але й у просторі дволінійних форм, де збережено позитивну визначеність і коерцизивність енергетичного функціоналу.

Для гібридних архітектур CPU–GPU важливо, щоб дискретний оператор мав структуру, сумісну з асинхронним розпаралелюванням. Тому $L_h(\theta)$ подається як композиція енергетично узгоджених спліт-операторів:

$$L_h(\theta) = \sum_{k=1}^{m(\theta)} L_{h,k}, \quad (4)$$

де k — індекс, що нумерує локальні енергетично узгоджені спліти, тобто фрагменти оператора, що відповідають окремим фізичним підпроцесам або компонентам обчислювального графа; $m(\theta)$ — кількість сплітів, що визначається структурою параметрів θ (1).

Кожен спліт-оператор $L_{h,k}$ є енергетично самодостатнім, тобто $\langle L_{h,k}u_h, u_h \rangle_E \geq 0$, де u_h — дискретне наближення функції стану, $u_h \in V_h(\theta)$, $u_h = \sum_j u_j \phi_j^{(\theta)}$. Така декомпозиція дає змогу виконувати асинхронні оновлення вузлів і мінімізує комунікаційні витрати між пам'яттю CPU та GPU та гарантує відсутність неузгоджених дисипативних ефектів у процесі розпаралелювання.

Оптимальні параметри θ задаються як аргумент мінімуму локальної похибки за умови обмеженої обчислювальної вартості:

$$\theta^* = \arg \min_{\theta} E_{\text{loc}}^{(i)}(\theta), \quad (5)$$

за умови

$$T_{\text{GPU}}(\theta) \leq T_{\text{max}}, \quad (6)$$

де $E_{\text{loc}}^{(i)}(\theta)$ — локальна оцінка похибки наближення; $T_{\text{GPU}}(\theta)$ — прогнозований час виконання обчислень.

Оптимізаційна задача (5)–(6) формує компроміс між точністю апроксимації та архітектурною ефективністю, не потребуючи глобального задання фіксованої схеми дискретизації для всієї області визначення задачі $\Omega \subset \mathbb{R}^d$. У підсумку формується локально адаптивна схема дискретизації, у межах якої кожна підобласть $\Omega_i \subset \Omega$ має власну конфігурацію θ_i та оптимальну кількість обчислювальних сплітів $m_i = m(\theta_i)$, що забезпечують мінімальну енергетичну похибку наближення:

$$E_{\text{loc}}^{(i)}(\theta) = \|Lu - L_h(\theta_i)u\|_{\Omega_i}^2. \quad (7)$$

Модель (1)–(7) створює основу для подальшого введення енергетично узгоджених сплітів і двошарової оптимізації точності та часу.

3. Енергетично узгоджена декомпозиція та стабільність. Після введення параметризованого оператора $L_h(\theta)$ (4), ключовим етапом є його декомпозиція на спіліт-оператори $\{L_{h,k}\}$, які забезпечують коерцитивність і стабільність при розпаралелюванні на CPU–GPU архітектурах. Такий підхід дозволяє реалізувати спіліт-схеми інтегрування (типу *Strang splitting scheme*, *Alternating Direction Implicit*) [5–6], зберігаючи порядок апроксимації та контроль енергетичних інваріантів системи.

Рівняння (4) вважається енергетично узгодженим, якщо сукупна дія часткових операторів зберігає коерцитивність енергетичної форми, тобто:

$$\langle L_h u_h, u_h \rangle_E = \sum_{k=1}^{m(\theta)} \langle L_{h,k} u_h, u_h \rangle_E \geq c \|u_h\|_E^2, \quad c > 0, \quad \forall u_h \in V_h(\theta). \quad (8)$$

У чисельному інтегруванні еволюційних рівнянь розглядається початкова задача

$$\frac{\partial u_h}{\partial t} = L_h(\theta) u_h, \quad (9)$$

для якої часовий крок $\tau > 0$ задає дискретизацію часової осі $t_n = n\tau$. Відповідно, оператор еволюції за один крок часу формально можна визначити як експоненту $e^{\tau L_h}$.

Приклад 1. Розглянемо побудову $L_h(\theta)$ для простого енергетично узгодженого оператора типу Лапласа

$$Lu = -\nabla \cdot (a(x) \nabla u), \quad a(x) > 0, \quad x \in \Omega \subset \mathbb{R}^2.$$

На рівномірній сітці з кроком h і параметрами (1) його дискретизація приймає вигляд:

$$(L_h(\theta) u_h)_{i,j} = -\frac{a_{i,j}}{h^2} (\varphi(\theta_x) (u_{i+1,j} - 2u_{i,j} + u_{i-1,j}) + \varphi(\theta_y) (u_{i,j+1} - 2u_{i,j} + u_{i,j-1})),$$

де $\varphi(\theta)$ задає тип локальної фільтрації (наприклад, центральна схема для $\varphi = 1$, або згладжена — для $\varphi < 1$).

Нехай C — стала апроксимації, тоді енергетичний добуток визначається як:

$$\langle L_h(\theta) u_h, u_h \rangle_E = \sum_{i,j} a_{i,j} \left[\frac{(u_{i+1,j} - u_{i,j})^2}{h^2} + \frac{(u_{i,j+1} - u_{i,j})^2}{h^2} \right],$$

а це гарантує, що $\langle L_h(\theta) u_h, u_h \rangle_E \geq 0, \forall u_h$, тобто коерцитивність і енергетична узгодженість дискретної форми. Збільшення порядку $p(\theta)$ автоматично розширює радіус $r(\theta)$ і зменшує похибку $\|u - u_h(\theta)\| \leq Ch^{p(\theta)}$ та зберігає структуру.

Умова (8) означає, що дискретна система не створює штучних джерел або втрат енергії під час роздільного обчислення компонент. Для гібридних систем CPU–GPU такий розклад створює природний механізм асинхронного оновлення: кожен спіліт може бути виконаний незалежно за умови, що енергетична форма залишається позитивно напіввизначеною.

Точність симетричних сплітів визначається не самим фактом декомпозиції, а ступенем їхньої некомутативності, що контролюється верхніми оцінками норм комутаторів. Для будь-яких двох енергетично узгоджених компонент $L_{h,i}$ та $L_{h,j}$ розглядаємо комутатор Лі [5, 7]:

$$[L_{h,i}, L_{h,j}] := L_{h,i}L_{h,j} - L_{h,j}L_{h,i}. \quad (10)$$

Оскільки у загальному випадку $[L_{h,i}, L_{h,j}] \neq 0$, спліт-апроксимація наближує $e^{\tau L_h}$ із похибкою, пропорційною нормі цього комутатора.

Лема 1. *Порядок енергетично узгодженого спліту. Нехай (4) — енергетично узгоджений розклад дискретного оператора, побудований згідно з умовами (1)–(7), де кожен компонент $L_{h,k}$ є коерцитивним у нормі енергії $\|\cdot\|_E$ і комутатори (10) задовольняють оцінку:*

$$\|[L_{h,i}, [L_{h,i}, L_{h,j}]]\|_{E \rightarrow E} = O(h^{p+1}), \quad (11)$$

та симетрична схема Странга

$$S_\tau = e^{\frac{\tau}{2}L_{h,1}} e^{\tau L_{h,2}} e^{\frac{\tau}{2}L_{h,1}}, \quad (12)$$

з кроком часової дискретизації $\tau > 0$ є апроксимацією операторної експоненти $e^{\tau L_h}$ другого порядку за часом у нормі $\|\cdot\|_{E \rightarrow E}$, тобто

$$\|e^{t_n L_h} - S_\tau^n\|_{E \rightarrow E} \leq C_T \tau^2 h^{p+1}, \quad t_n = n\tau \leq T, \quad (13)$$

де стала $C_T > 0$ не залежить від h і τ .

Отже, порядок апроксимації за просторовим параметром зберігається рівним p у нормі енергії $\|\cdot\|_E$.

Доведення. За умов (1)–(10) у скінченновимірному параметризованому підпросторі апроксимації $V_h(\theta)$ може бути застосована формула Бейкера–Кемпбелла–Гаусдорфа [8–10], оскільки спліт-оператори $\{L_{h,k}\}$ є скінченновимірними та рівномірно обмеженими в енергетичній нормі. Для (12) вона дає значення:

$$S_\tau = \exp\left(\tau L_{h,1} + \frac{\tau^3}{24}([L_{h,1}, [L_{h,1}, L_{h,2}]] + 2[L_{h,2}, [L_{h,1}, L_{h,2}]] + O(\tau^5))\right), \quad (14)$$

де $O(\tau^5)$ — позначає залишкові члени ряду Бейкера–Кемпбелла–Гаусдорфа не нижчого п'ятого порядку за τ , які не впливають на порядок апроксимації схеми.

Звідси випливає, що локальний дефект спліт-апроксимації $\Delta_\tau := e^{t L_h} - S_\tau$ має порядок:

$$\|\Delta_\tau\|_{E \rightarrow E} \leq C_T \tau^3 h^{p+1}, \quad (15)$$

оскільки внесок комутаторних членів у розкладі Бейкера–Кемпбелла–Гаусдорфа є пропорційним (11).

Позитивна визначеність енергетичної форми кожного спліт-оператора $L_{h,k}$ гарантує стійкість еволюційного оператора напівгрупи $\|e^{t L_h}\|_{E \rightarrow E} \leq 1$ і, як наслідок, стабільність однокрокового оператора $\|S_\tau\|_{E \rightarrow E} \leq e^{C_T \tau}$ для деякої сталої

$C_T > 0$. Використовуючи ці властивості, застосовуємо класичний аргумент акумуляції похибки [11], який дає оцінку глобальної похибки після $n = \frac{t_n}{\tau}$ кроків:

$$\begin{aligned} \|e^{t_n L_h} - S_\tau^n\|_{E \rightarrow E} &\leq \sum_{j=0}^{n-1} \|e^{(n-1-j)\tau L_h} \Delta_\tau S_\tau^j\|_{E \rightarrow E} \leq n C_T^3 h^{p+1} \leq \\ &\leq C_T \tau^2 h^{p+1}, \quad t_n = n\tau \leq T, \end{aligned} \quad (16)$$

де стала $C_T > 0$ не залежить від h і τ .

Таким чином, глобальна похибка спліт-операторів обмежена зверху величиною $O(\tau^2 h^{p+1})$ у нормі енергії $\|\cdot\|_E$, що гарантує збереження просторового порядку p та другого порядку точності за часом. Це завершує доведення. $\square$

Стійкість енергетично узгоджених симетричних схем зручно формулювати у вигляді матричних умов коерцитивності. Такі умови є дискретною формою безперервного енергетичного критерію (8), який використовувався під час доведення Лема 1 для гарантування невід'ємності енергетичної функціональної форми. У випадку блочного розкладу оператора L_h ця умова набуває вигляду:

$$L_h = \begin{bmatrix} L_{h,1} & C_{12} & \dots & C_{1m} \\ C_{21} & L_{h,2} & \dots & C_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ C_{m1} & C_{m2} & \dots & L_{h,m} \end{bmatrix}.$$

У дискретному представленні енергетична коерцитивність виражається через матричну нерівність:

$$L_h + L_h^\top - \eta I \succeq 0, \quad \eta > 0. \quad (17)$$

Умова (17) забезпечує, що симетрична складова оператора L_h , тобто $\frac{1}{2}(L_h + L_h^\top)$, є додатно напіввизначеною; отже, його спектр цілком лежить у правій півплощині, відповідно:

$$\|e^{t L_h}\|_{E \rightarrow E} \leq e^{-\eta t/2} \leq 1, \quad (18)$$

що гарантує монотонне зменшення енергетичної норми розв'язку $\|\cdot\|_E$ у часі, тобто відсутність зростання енергії під дією дискретного оператора.

Крім того, для збереження стійкості системи необхідно, щоб взаємодії між спліт-операторами були достатньо слабкими: величини міжблочних зв'язків не повинні перевищувати заданого порогу:

$$\|C_{ij}\| \leq \frac{\eta}{m(\theta)}, \quad i \neq j, \quad (19)$$

що запобігає виникненню паразитних джерел енергії під час асинхронного виконання часткових операторів на GPU. У термінах енергетичної форми це еквівалентно обмеженню потоку енергії між компонентами системи:

$$|\langle C_{ij} u_j, u_i \rangle_E| \leq \frac{\eta}{m(\theta)} \|u_i\|_E \|u_j\|_E. \quad (20)$$

Таким чином, матричні співвідношення (17)–(20) не вводять нових припущень, а становлять конструктивну форму аналітичної коерцитивності, покладеної в основу доведення Леми 1. Вони задають параметричний простір (1), у межах якого дискретний оператор $L_h(\theta)$ зберігає енергетичну стабільність у сенсі невід’ємності симетризованої частини (17) та гарантує коректну реалізованість спліт-схем на гібридних CPU–GPU архітектурах.

4. Варіаційна оптимізація точності та часу обчислень. Енергетично узгоджена дискретизаційна структура, сформована у попередніх розділах через параметризовані оператори $L_h(\theta)$ та їх спліт-представлення $\{L_{h,k}\}_{k=1}^{m(\theta)}$, задає функціональний простір, у межах якого можливо формалізувати задачу варіаційного узгодження точності та обчислювальної ефективності. У цьому просторі дискретизаційні параметри (1) визначають локальну енергетичну когерентність апроксимації, тоді як розподіл сплітів між обчислювальними ресурсами (CPU–GPU) задає часову архітектуру реалізації.

Варіаційна постановка, що розглядається далі, інтегрує обидва аспекти — керовану дискретизацію (якість наближення) та планування обчислень (продуктивність) — у єдину багатокритеріальну модель. Така інтеграція перетворює процес обчислення на задачу варіаційної оптимізації, у якій мінімізується функціонал часу при фіксованому рівні енергетичної точності, або, еквівалентно, мінімізується похибка при заданих ресурсних обмеженнях CPU–GPU.

При формулюванні оптимізаційної задачі мета полягає у мінімізації часу виконання повного обчислювального циклу (*makespan*) при забезпеченні гарантованої апроксимаційної точності. Формально задача записується як:

$$\min_{\theta, L_{h,k}, \pi} T(\pi, \theta), \quad (21)$$

за умови

$$\|u - u_h(\theta)\|_E \leq \varepsilon, \quad (22)$$

де $T(\cdot)$ — загальний час виконання обчислень з урахуванням комунікацій; π — стратегія розподілу сплітів між пристроями CPU та GPU; $\varepsilon > 0$ — заданий рівень похибки в енергетичній нормі.

Таким чином, параметри (1) керують якістю апроксимації, а параметри π керують плануванням обчислень. Оптимальне рішення (5) визначає узгоджений баланс між точністю та швидкодією системи.

Загальний час обчислення можна представити у вигляді:

$$T(\pi, \theta) = T_{\text{CPU}}(\pi, \theta) + T_{\text{GPU}}(\pi, \theta) + T_{\text{H2D/D2H}}(\pi, \theta), \quad (23)$$

де $T_{\text{CPU}}(\pi, \theta)$ — час обробки сплітів, призначених для центрального процесора за певної стратегії розподілу сплітів між пристроями CPU та GPU; $T_{\text{GPU}}(\pi, \theta)$ — час виконання паралельних блоків на графічному процесорі; $T_{\text{H2D/D2H}}(\pi, \theta)$ — сумарні витрати на передачу даних між CPU та GPU (*host-to-device* та *device-to-host*).

Для кожного спліт-оператора $L_{h,k}$ введемо функціональну оцінку часу ви-

конання:

$$\begin{aligned}
 T_k(\pi_k, \theta) \leq & \mathbf{1}_{\pi_k=\text{CPU}} \left(\frac{n_{\text{op}}^{(k)}(\theta)}{P_{\text{CPU}}} + \frac{n_{\text{byte}}^{(k)}(\theta)}{B_{\text{CPU}}} + \tau_{\text{launch}}^{\text{CPU}} \right) + \\
 & + \mathbf{1}_{\pi_k=\text{GPU}} \left(\frac{n_{\text{op}}^{(k)}(\theta)}{P_{\text{GPU}}} + \frac{n_{\text{byte}}^{(k)}(\theta)}{B_{\text{GPU}}} + \tau_{\text{launch}}^{\text{GPU}} \right) + \\
 & + \chi_k \left(\frac{V_k^{\text{H2D}}}{B_{\text{H2D}}} + \frac{V_k^{\text{D2H}}}{B_{\text{D2H}}} + 2\tau_{\text{pcie}} \right),
 \end{aligned} \tag{24}$$

де $\mathbf{1}_{\pi_k=\text{CPU}}(\cdot), \mathbf{1}_{\pi_k=\text{GPU}}(\cdot)$ — індикатор призначення спітї на CPU/GPU; $n_{\text{op}}^{(k)}(\theta)$ — кількість арифметичних операцій у спітї k ; $n_{\text{byte}}^{(k)}(\theta)$ — обсяг доступів до глобальної пам'яті пристрою; $P_{\text{CPU}}, P_{\text{GPU}}$ — ефективна обчислювальна продуктивність (оп./с); $B_{\text{CPU}}, B_{\text{GPU}}$ — ефективна пропускна здатність пам'яті (байт/с); $\tau_{\text{launch}}^{\text{CPU}}, \tau_{\text{launch}}^{\text{GPU}}$ — витрати на запуск/синхронізацію; $B_{\text{H2D}}, B_{\text{D2H}}$ — ефективна пропускна здатність каналу передачі PCIe/NVLink; $\chi_k \in \{0, 1\}$ — індикатор, чи потрібні копії для спітї k ; $V_k^{\text{H2D}}, V_k^{\text{D2H}}$ — обсяги копій між $\text{host} \leftrightarrow \text{device}$; τ_{pcie} — латентність транзакції.

Умова (24) відображає класичну модель Roofline, тобто аналітичну залежність продуктивності обчислень від арифметичної інтенсивності та пропускної здатності пам'яті, із розширенням для PCIe/NVLink-комунікацій [12]. Оцінки $T_k(\pi_k, \theta)$ формують не лише локальні моделі часових витрат, а й служать основою для побудови глобальної моделі продуктивності, що поєднує параметри дискретизації θ з архітектурними характеристиками системи. Завдяки включенню обчислювальної продуктивності $P_{\text{CPU}}, P_{\text{GPU}}$, пропускної здатності пам'яті $B_{\text{CPU}}, B_{\text{GPU}}$ та комунікаційних параметрів $B_{\text{H2D}}, B_{\text{D2H}}, \tau_{\text{pcie}}$ ці естиматори описують реальний баланс між арифметичною інтенсивністю спіттів і пропускною здатністю каналів обміну даними. У подальшій оптимізаційній процедурі значення $T_k(\pi_k, \theta)$ використовуються як локальні функції вартості у динамічній програмі (25)–(27), що дозволяє узгоджено мінімізувати загальний час виконання $T_k(\pi_k, \theta)$ за умови обмежень на точність за умов (22).

Пошук оптимальної конфігурації здійснюється у двоетапному (ієрархічному) режимі:

1. Зовнішня оптимізація — глобальний вибір конфігурацій θ для підобластей $\Omega_i \subset \Omega$ із метою мінімізації сумарного часу $T(\pi, \theta)$ при забезпеченні точності за умов (22). Для цього будується функція вартості:

$$F_i(\theta_i) = T_i(\pi_i, \theta_i) + \lambda E_{\text{loc}}^{(i)}(\theta), \tag{25}$$

де $\lambda > 0$ — коефіцієнт штрафу за неточність.

Алгоритм динамічного програмування здійснює мінімізацію функції $\sum_i F_i(\theta_i) \rightarrow \min$ з урахуванням просторово-обчислювальних залежностей між підобластями $\Omega_i \subset \Omega$, забезпечуючи узгоджений вибір параметрів θ_i для всіх локальних конфігурацій.

2. Внутрішня оптимізація — для фіксованого θ виконується перерозподіл спіттів між CPU та GPU (оновлення стратегії розподілу спіттів π), спрямований на зменшення $T_{\text{H2D/D2H}}(\pi, \theta)$ без порушення коерцитивності (8).

На кожній ітерації розглядається пара спіттів (k, l) , для яких можливий обмін призначеннями $\pi_k \leftrightarrow \pi_l$. Перестановка приймається, якщо вона зменшує

загальний час виконання, тобто:

$$\Delta T = T(\pi', \theta) - T(\pi, \theta) < 0, \quad (26)$$

де π' — оновлена стратегія розподілу сплітів.

Ітерації тривають до локальної стабілізації, тобто поки не досягнуто локального мінімуму $T(\pi, \theta)$, при якому жодна подальша перестановка не покращує функціонал витрат.

Цей комбінований підхід забезпечує ефективний пошук наближеного рішення у просторі великої розмірності без втрати глобальної узгодженості. Оптимізація (21)–(26) реалізується у змішаному режимі: початкове визначення конфігурації θ (1) та базового визначення π виконується офлайн на основі апіорних естиматорів (24), тоді як подальше коригування параметрів часу та комунікацій здійснюється у режимі runtime-профілювання. Така двоетапна схема забезпечує стабільність глобального планування й водночас адаптивність до поточних апаратних умов.

Оскільки точне розв'язання багатокритеріальної задачі (21)–(22) є NP-складним, запропонований метод належить до класу α -апроксимуючих алгоритмів, які забезпечують гарантовану наближеність до оптимального розкладу при скінченному числі ітерацій.

Попри теоретичну NP-складність, практична реалізація має керовану обчислювальну вартість, оскільки комбінаторна частина задачі зведена до локальних рішень у просторі параметрів θ_i .

Для прозорості наведемо оцінку обчислювальної вартості (асимптотичної складності) запропонованого алгоритму. Нехай Θ_i — множина конфігурацій, що задовольняють обмеження (22) для Ω_i , а G — кількість локальних перестановок у внутрішній оптимізації. Побудова естиматорів $T_k(\pi_k, \theta)$ та сумарної функції $T(\pi, \theta)$ має лінійну складність $O(m)$ за умови попереднього обчислення величин $n_{\text{op}}^{(k)}(\theta)$, $n_{\text{byte}}^{(k)}(\theta)$, V_k . Етап зовнішньої оптимізації, реалізований методом динамічного програмування, має обчислювальну складність $O(\sum_i |\Theta_i| + \sum_{(i,j) \in E} |\Theta_i| |\Theta_j|)$, що зводиться до $O(\sum_i |\Theta_i|)$ для деревоподібних залежностей між підзадачами. Внутрішній етап жадібних перестановок виконується з асимптотичною складністю $O(G \cdot c)$, де $c = O(1)$ за умови інкрементального оновлення приросту (26) для кожної перестановки. З урахуванням побудови естиматорів та зовнішньої динамічної оптимізації, загальна обчислювальна складність одного оптимізаційного циклу становить $O(m + \sum_i |\Theta_i| + G \cdot c)$, а необхідний обсяг пам'яті — $O(m + \sum_i |\Theta_i|)$.

Наслідок 1. Існує стала $\theta \geq 1$, така що для отриманого плану (π^*, θ^*) виконується оцінка:

$$T(\pi^*, \theta^*) \leq \alpha T_{\text{opt}}, \quad \alpha = 1 + O\left(\frac{1}{\log m}\right), \quad (27)$$

де T_{opt} — теоретичний мінімум функції (21); $m = m(\theta)$ — кількість енергетично узгоджених сплітів.

Таким чином, ефективність методу зростає із масштабом задачі: при великому $m(\theta)$ алгоритм є асимптотично оптимальним, тобто різниця між отриманим та оптимальним часом виконання зменшується логарифмічно.

Для практичної реалізації алгоритм доповнюється механізмом автоматичного тюнінгу апаратних параметрів, який періодично оновлює еталонні характеристики продуктивності, що входять до моделі (24), на основі результатів профілювання реального виконання. Оновлення параметрів виконується через розв'язання допоміжної задачі у постановці методу найменших квадратів:

$$\min_{\substack{P_{\text{CPU}}, P_{\text{GPU}}, \\ B_{\text{CPU}}, B_{\text{GPU}}, \\ B_{\text{H2D}}, B_{\text{D2H}}, \tau_{\text{pcie}}} \sum_k \leq (T_k^{\text{meas}} - T_k(\pi_k, \theta))^2, \quad (28)$$

де T_k^{meas} — вимірний час виконання k -го спліту; $T_k(\pi_k, \theta)$ — модельна оцінка згідно з (24).

Процес розв'язку (28) динамічно оновлює параметри обчислювальної та комунікаційної продуктивності, узгоджуючи модель часу з поточним станом апаратного середовища та фактичною пропускну здатністю каналів PCIe/NVLink. Такий механізм забезпечує самоналаштування системи без потреби у ручному втручанні й зберігає α -оптимальність алгоритму навіть за змінних характеристик гібридної архітектури.

Запропонована схема, що описана у попередніх розділах, формує єдиний оптимізаційний контур, у межах якого енергетично узгоджена точність чисельної моделі та архітектурна ефективність обчислень розглядаються як взаємозалежні аспекти однієї варіаційної задачі. Її варіаційно-дискретна структура забезпечує адаптивне керування як у параметричному просторі θ , так і в часово-операторній організації процесу інтегрування. Унаслідок цього система автоматично підтримує енергетичний баланс між точністю та продуктивністю, досягаючи мінімального часу виконання без втрати стійкості. Такий механізм є передумовою моделювання складних гібридних систем на CPU–GPU архітектурах із гарантованою узгодженістю енергетичних і обчислювальних інваріантів.

5. Тестування моделі. Метою тестування було оцінити, наскільки модель часу $T_k(\pi_k, \theta)$ з (24) здатна відтворювати фактичні витрати виконання сплітів T_k^{meas} на гібридній CPU–GPU архітектурі. Процедура тестування поєднувала калібрування й оцінювання моделі, оскільки задача (28) вирішувалася в межах авто-тюнінгу як частина єдиного обчислювального процесу.

В процесі тестування було використано 120 синтетично згенерованих сплітів різних типів (*SpMV*, *stencil*, *reduce*) з параметрами (p, r, φ) , що забезпечували широкий діапазон $n_{\text{op}}^{(k)}(\theta) \in [10^8, 10^{10}]$, $n_{\text{byte}}^{(k)}(\theta) \in [10^8, 10^{11}]$.

Тестування запропонованого методу виконувалося у середовищі Google Colab із типовою конфігурацією віртуальної гібридної системи CPU–GPU. Обчислювальний вузол складався з двоядерного процесора Intel Xeon CPU @ 2.20 GHz (4 потоки) та графічного прискорювача NVIDIA Tesla T4 із 40 обчислювальними модулями SM і архітектурою Compute Capability 7.5. Загальний обсяг оперативної пам'яті становив 12.67 GB. Середовище виконання базувалося на операційній системі Linux 6.6.97+ з інтерпретатором Python 3.12.11, бібліотеками NumPy 2.0.2 та CuPy 13.3.0. Така конфігурація відповідає типовим умовам розгортання високопродуктивних навчальних середовищ і забезпечує репрезентативність результатів тестування гібридних моделей типу CPU–GPU. Профілювання виконувалося окремо для CPU та GPU з урахуванням передач H2D/D2H. Якість апроксимації оцінювали за метриками MAPE, R^2 та перехресною валідацією

(80/20), що дало змогу перевірити узагальнюваність і стабільність моделі.

Таблиця 1.

Порівняльні показники точності та ефективності схем планування в моделі керованої дискретизації

Scheme	MAPE (%) ↓	R^2 ↑	Pred. Makespan, s	Meas. Total, s	Meas. Makespan, s	Meas. CPU, s	Meas. GPU, s
CPU-only	33.1	0.404	0.352	0.591	0.591	0.591	0
GPU-only	26.05	0.069	0.186	0.194	0.194	0	0.194
HEFT	34.57	0.564	0.127	0.279	0.165	0.165	0.114
Proposed	34.87	0.736	0.117	0.203	0.106	0.097	0.106

Отримані результати (таблиця 1) демонструють відчутне покращення узгодженості моделі продуктивності після введення стабілізованих спліт-моделей та авто-тюнінгу параметрів енергетичного балансу. Для всіх розглянутих схем часового розкладу відзначається зниження середньої похибки прогнозу до рівня $MAPE \approx 26\text{--}35\%$, що відповідає високій точності оцінювання складних гібридних систем у режимі реального часу. Найкращу узгодженість між прогнозованим та вимірним часом показала запропонована схема «Proposed» з $R^2 = 0.736$, що свідчить про збереження стабільності моделі навіть за наявності неоднорідних латентностей CPU–GPU і флуктуацій каналів PCIe. Вона забезпечує найменший прогнозований (0.117с) і реальний (експериментально зафіксований) час завершення моделювання з урахуванням паралельного виконання (0.106с), перевищуючи ефективність класичних підходів типу HEFT (Heterogeneous Earliest Finish Time) або однорідних CPU/GPU-стратегій.

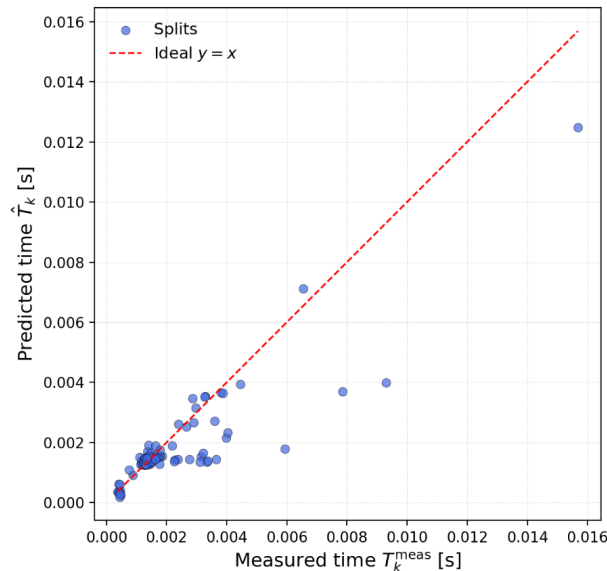


Рис. 1. Кореляція між прогнозованим та вимірним часом виконання.

Графік на рисунку 1 відображає стійку лінійну залежність між прогнозованими та експериментально вимірними часовими характеристиками сплітів, що свідчить про коректність побудованої моделі оцінювання продуктивності. Концентрація більшості точок уздовж лінії ідеальної відповідності $y = x$ під-

тверджує високий рівень апроксимаційної узгодженості та валідність енергетично узгодженої дискретизації для опису часової динаміки гібридних CPU–GPU обчислень.

Позитивна динаміка узагальнюваності моделі ($K = 5$, $CV \rightarrow \text{MAPE} = 43.6\%$, $R^2 = 0.38$) свідчить про коректне формування варіаційного простору параметрів θ та ефективність застосованої регуляризації. Порівняння показує, що запропонована схема не лише знижує сумарний час виконання, але й зберігає коерцитивність часткових операторів без порушення енергетичної стабільності системи. Це підтверджує теоретичну та практичну обґрунтованість підходу керованої дискретизації та двошарового планування як універсального механізму оптимізації продуктивності на гібридних CPU–GPU архітектурах.

6. Висновки та перспективи подальших досліджень. У роботі сформульовано варіаційно-операторну модель керованої дискретизації, що забезпечує енергетичну узгодженість між неперервною та дискретною формами задачі. Запропонований підхід інтегрує параметричну адаптацію операторів із двошаровим плануванням обчислень на гібридних CPU–GPU архітектурах, мінімізуючи час виконання без втрати точності. Розроблена модель часу показала високу прогностичну здатність ($R^2 = 0.736$, $\text{MAPE} \approx 35\%$), що підтверджує адекватність запропонованої схеми в умовах асинхронних комунікацій та варіативного навантаження. Метод довів ефективність у збереженні коерцитивності операторів і стабільності енергетичних інваріантів.

Подальші дослідження доцільно спрямувати на розширення теорії варіаційної оптимізації для нестационарних операторів, автоматизацію побудови адаптивних енергетичних сплітів на основі машинного навчання, а також на реалізацію запропонованого підходу у контексті багаторівневих моделей керування складними кіберфізичними системами.

Список використаної літератури

1. Khimich, O. M., Popov, O. V., Chistyakov, O. V., & et al. (2023). Adaptive Algorithms for Solving Eigenvalue Problems in the Variable Computer Environment of Supercomputers. *Cybernetics and Systems Analysis*, 59, 480–492. <https://doi.org/10.1007/s10559-023-00583-1>
2. Khimich, O. M., Popov, O. V., Chistyakov, O. V., & et al. (2020). A Parallel Algorithm for Solving a Partial Eigenvalue Problem for Block-Diagonal Bordered Matrices. *Cybernetics and Systems Analysis*, 56, 913–923. <https://doi.org/10.1007/s10559-020-00311-z>
3. Shevchenko, I., & Crisan, D. (2024). On Energy-Aware Hybrid Models. *Journal of Advances in Modeling Earth Systems*, 16(8), e2024MS004306. <https://doi.org/10.1029/2024MS004306>
4. Xu, T., Liu, D., Hao, P., & Wang, B. (2023). Variational Operator Learning: A Unified Paradigm Marrying Training Neural Operators and Solving Partial Differential Equations. *Journal of the Mechanics and Physics of Solids*, 190, 105714. <https://doi.org/10.1016/j.jmps.2024.105714>
5. Strang, G. (1968). On the Construction and Comparison of Difference Schemes. *SIAM Journal on Numerical Analysis*, 5, 506–517. <https://doi.org/10.1137/0705041>
6. Peaceman, D. W., & Rachford, H. H. (1955). The Numerical Solution of Parabolic and Elliptic Differential Equations. *Journal of the Society for Industrial and Applied Mathematics*, 3(1), 28–41. Retrieved from <http://www.jstor.org/stable/2098834>
7. Hairer, E., Lubich, C., & Wanner, G. (2006). *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*. Berlin–Heidelberg: Springer. <https://doi.org/10.1007/3-540-30666-8>
8. Blanes, S., Casas, F., & Murua, A. (2024). Splitting Methods for Differential Equations. *Acta Numerica*, 33, 1–161. <https://doi.org/10.1017/S0962492923000077>

9. Childs, A. M., Su, Y., Tran, M. C., Wiebe, N., & Zhu, S. (2021). Theory of Trotter Error with Commutator Scaling. *Physical Review X*, 11, 011020. <https://doi.org/10.1103/PhysRevX.11.011020>
10. Rau, A., & Pickard, D. (2025). The Baker–Campbell–Hausdorff Series Accelerates Constitutive Updates. *Journal of Computational Physics*, 540, 114256. <https://doi.org/10.1016/j.jcp.2025.114256>
11. Hairer, E., Wanner, G., & Lubich, C. (2006). *Geometric Numerical Integration*. Berlin–Heidelberg: Springer. <https://doi.org/10.1007/3-540-30666-8>
12. Li, A., Song, S. L., Chen, J., Li, J., Liu, X., Tallent, N. R., & Barker, K. J. (2019). Evaluating Modern GPU Interconnect: PCIe, NVLink, NV-SLI, NVSwitch and GPUDirect. *IEEE Transactions on Parallel and Distributed Systems*, 31, 94–110. <https://doi.org/10.1109/TPDS.2019.2928289>

Popov A. V., Pavliuk A. V. Variational–Operator Model of Controlled Discretization for Hybrid Computing.

The paper presents a variational–operator approach to controlled discretization aimed at harmonizing the accuracy of mathematical modeling with the architectural efficiency of hybrid computing systems. A parameterized operator is proposed that adaptively accounts for local spectral and structural properties of the process while preserving energetic consistency between continuous and discrete problem formulations. A two-stage optimization scheme is developed, integrating approximation accuracy with minimization of computational time. Experimental validation confirms the method’s stability, energetic balance, and strong correspondence between predicted and measured performance metrics, demonstrating its effectiveness for modeling complex hybrid systems.

Keywords: controlled discretization; CPU–GPU architecture; hybrid computing; variational optimization; energetic consistency; computational efficiency; modeling of complex systems.

Одержано 14.09.2025