

1. Lipowski A., Lipowska D. Bio-linguistic transition and Baldwin effect in an evolutionary naming-game model. *International Journal of Modern Physics C*. 2008. Т. 19, вип. 3. С. 399–407. URL: <https://doi.org/10.1142/S0129183108012248>.
2. Lipowska D. Naming Game and Computational Modelling of Language Evolution. *Computational Methods in Science and Technology*. 2011. Т. 17, вип. 1-2. С. 41–51. URL: <https://doi.org/10.12921/CMST.2011.17.01.41-51>.
3. Moloney P. Evolutionary Naming Game: an Agent-based Model for the Emergence of Language : M.Sc. Computer Science - Augmented and Virtual Reality / Trinity College Dublin. Dublin, 2021. URL: <https://publications.scss.tcd.ie/theses/diss/2021/TCD-SCSS-DISSERTATION-2021-042.pdf> (дата звернення: 31.10.2025).

GPU-ОРІЄНОВАНА БІДІАГОНАЛІЗАЦІЯ В АЛГОРИТМІ СИНГУЛЯРНОГО РОЗКЛАДУ З ШВИДКОЮ ДЕФЛЯЦІЄЮ ТА ПАРАЛЕЛЬНИМ РОЗБИТТЯМ / GPU-NATIVE BIDIAGONAL SVD WITH OPPORTUNISTIC DEFLATION AND PARALLEL SPLITTING

Сухарський С. С. / Sukharskyi S. S.

Інститут програмних систем НАН України

Київ, 03187, Україна, Проспект Академіка Глушкова, 40, корп. 5, +38 (044) 526-33-19

E-mail: serhii.sukharskyi@gmail.com

This research focuses on a GPU-native bidiagonal SVD that performs scale-aware, opportunistic deflation with immediate splitting into independent subproblems. We replace full upper/lower sweeps with an implicit-shift (Wilkinson) paired left/right Givens bulge-chasing, sharply reducing iterations. Householder reduction uses operations optimization techniques, and rotations on U and V are applied in tiled, GEMM-like passes. Avoiding per-rotation kernel launches and improving inter-block coordination yield 2–3× speedups over the initial version [5]. Early splitting exposes parallelism across subblocks, improving SM occupancy and time-to-solution for large dense matrices. The method preserves numerical stability and near-Jacobi orthogonality via scale-aware tests. We also propose a “step-0” heuristic targeting the largest-magnitude bidiagonal entries to hasten early deflation. Ongoing evaluation benchmarks against Jacobi, QR/DC, and polar-decomposition SVDs across different matrix sizes and conditions.

У роботі розглянуто GPU-орієнтований алгоритм SVD, що намагається усунути проблеми традиційних підходів завдяки масштабованій, опортуністичній дефляції на бідіагональній формі з негайним розбиттям на незалежні підзадачі. Таке раннє розбиття перетворює один великий, послідовний конвеєр складності $O(n^3)$ на кілька менших, паралельних ядер, що покращує використання локального кешу, підвищує завантаження поточкових мультипроцесорів (SM) та забезпечує ефективні за пропускну здатністю, GEMM-подібні оновлення накопичених ортогональних матриць U та V . У підсумку істотно зменшується час обчислень для великих щільних матриць, що є критично важливим для задач на кшталт аналізу основних компонент (PCA), спектральної фільтрації та аналітики даних.

Наявні бібліотеки зазвичай надають перевагу надійним методам: однобічному Якобі (відмінна ортогональність; сильний у пакетній обробці малих/середніх задач) [4, 5], класичним QR/Розділяй-і-Пануй (DC) (висока надійність, але часто повільніші на GPU без ретельного блокування) [1], а також новішим SVD на основі полярного розкладу (висока швидкість на дуже великих щільних задачах) [2]. Існують багатоетапні GPU-розклади — перетворення щільних матриць у бідіагональну форму (інколи через проміжну стрічкову), — однак подальша бідіагоналізація часто лишається «вузьким місцем» [3] через притаманну послідовність переслідування «горба» (bulge chasing) та відсутність якнайранішої дефляції з динамічним плануванням підзадач. Це окреслює потребу для розробки алгоритму бідіагоналізації як частини алгоритму сингулярного розкладу з використанням усіх можливостей графічних процесорів, що поєднує неявні зсуви, жадібну дефляцію й розбиття, а також ефективне паралельне застосування обертань до накопичених множників U і V .

Завдання цього дослідження спроектувати та оцінити повний алгоритм SVD, оптимізований під сучасні графічні процесори, який охоплює: початкову редукцію матриці до бідіагональної форми з використанням ефективних перетворень Хауссхолдера; механізм бідіагоналізації з неявним зсувом і негайною дефляцією та розбиттям; застосування накопичених обертань до матриць U і V у паралельних проходженнях; ефективне планування підзадач за допомогою окремих ядер і динамічних черг на сучасних архітектурах GPU. Кінцева мета —

досягти суттєвого виграшу в продуктивності (ціль $\geq 1.5\times$ порівняно з сильними базовими QR/DC), зберігаючи рівень ортогональності Якобі та чисельну стабільність.

Постановка задачі. Для матриці

$$A \in R^{m \times n},$$

необхідно ефективно обчислити її сингулярний розклад (SVD)

$$A = U \Sigma V^T,$$

на графічному процесорі мінімізувавши час виконання за умови суворих чисельних допусків на залишок

$$\|A - U \Sigma V^T\| / \|A\|$$

та ортогональність векторних матриць

$$\|U^T U - I\|, \|V^T V - I\|.$$

Рішення має чітко визначити й реалізувати: перевірку дефляції та відповідну політику розбиття для бідіагоналі; схему оновлень з неявними зсувами під час діагоналізації; плиткову стратегію застосування обертань для ефективного оновлення накопичених U та V . Дослідження завершується еталонною реалізацією та суворою експериментальною оцінкою відносно сучасних базових реалізацій SVD Якобі, QR/DC і полярних методів на широкому діапазоні розмірів матриць і забезпечення виконання додаткових умов.

Після реалізації двоетапного SVD [5] ми перейшли до оптимізації його виконання на GPU. Перший підхід — запуск окремого ядра для кожного обертання Гівенса, що застосовується до U та V , — виявився повільнішим за монолітне ядро через значні накладні витрати від тисяч ітераційних запусків. Далі ми зосередилися на взаємодії між блоками CUDA та загальному завантаженні SM, що дало прискорення у 2–3 рази відносно початкової версії й дозволило зрівнятися з наявними базовими підходами. Наступним кроком ми досліджували розбиття великої задачі на менші: початкова стратегія ітеративного «пропускання» рядків/стовпців під час повного «обнулення» спочатку верхньої, а потім нижньої діагоналі, занадто повільно створювала нулі між потенційними підблоками. Це мотивувало перехід до неявно-зміщеного режиму: на кожному кроці застосовувати 2×2 обертання Гівенса спершу до верхньої, а тоді відразу й до нижньої діагоналі, використовуючи зсув Вілкінсона для пришвидшення збіжності. За попередніми тестами такий підхід різко зменшує кількість ітерацій і створює підґрунтя для реалізації розширеної логіки ранньої дефляції з метою подальшого розбиття матриці.

Висновки. Ми показали, що вибір гранулярності ядер критично впливає на продуктивність бідіагоналізації SVD на GPU: злиття багатьох обертань у більші ядра перевершує запуск «по одному обертанню» завдяки мінімізації накладних витрат. Поліпшення координації між блоками й підвищення завантаження SM забезпечують відчутне прискорення (у 2–3 рази) ще до зміни власне алгоритму. Заміна повних проходів «угору-вниз» на парне застосування обертання зліва/справа зі зсувом Вілкінсона істотно скорочує число ітерацій і раніше відкриває можливості для дефляції. Такий перехід не лише узгоджується з усталеними реалізаціями, а й дає практичний шлях розкладати великі задачі на незалежні підзадачі з подальшою паралельною обробкою. Запропонований дизайн відповідає сильним сторонам GPU (небагато великих ядер, плиткові GEMM-подібні оновлення U, V) при збереженні чисельної надійності. У подальшій роботі ми завершимо логіку якнайскорішої дефляції та оцінимо прискорення на великих матрицях відносно реалізацій Якобі, QR/DC і полярних методів SVD. Ми також плануємо запровадити новий «крок 0», який виконуватиметься перед основною логікою діагоналізації та застосовуватиме обертання Гівенса до найбільших за модулем елементів бідіагональної матриці, щоб отримати якнаймасивнішу діагональ.

Список джерел

1. Dongarra, J., Gates, M., Haidar, A., Kurzak, J., Łuszczek, P., Tomov, S., & Yamazaki, I. (2018). The singular value decomposition: Anatomy of optimizing an algorithm for extreme scale. *SIAM Review*, 60(4), 808–865. <https://doi.org/10.1137/17M1117732>
2. Nakatsukasa, Y., & Higham, N. J. (2013). Stable and efficient spectral divide and conquer algorithms for the symmetric eigenvalue decomposition and the SVD. *SIAM Journal on Scientific Computing*, 35(3), A1325–A1349. <https://doi.org/10.1137/120876605>
3. Tomov, S., Dongarra, J., & Baboulin, M. (2010). Towards dense linear algebra for hybrid GPU accelerated manycore systems. *Parallel Computing*, 36(5–6), 232–240.
4. Lahabar, S., & Narayanan, P. J. (2009). Singular value decomposition on GPU using CUDA. In *Proceedings of the 2009 IEEE International Symposium on Parallel & Distributed Processing* (pp. 1–10). IEEE. <https://doi.org/10.1109/IPDPS.2009.5161058>
5. Sukharskyi, S. (2026). Two-Step Singular Value Decomposition on a Single GPU with CUDA. In: Potapov, I., et al. *Digitalisation and Digital Transformation. RTC-Digital 2023*.

КАУЗАЛЬНЕ МОДЕЛЮВАННЯ ПОВЕДІНКИ КОРИСТУВАЧА У ЦИФРОВИХ СЕРВІСАХ

Білінський П.О. / Bilinskyi P.

Національний університет “Києво-Могилянська Академія” / National University of Kyiv-Mohyla Academy

04655, Київ, вул. Григорія Сковороди, 2, факультет інформатики, кафедра математики
E-mail: p.bilinskyi@ukma.edu.ua

This research focuses on developing approaches to model the behavior of a customer in digital services. A conceptual framework is proposed based on Structural Causal Models along with a basic example of causal model for customer retention prediction. The problem of causal identification is outlined, and existing approaches are briefly discussed. This work defines the direction of further research and presents basic methodology for the next steps. A representation learning approach is proposed as a potential tool to learn causal relations from the temporal customer behavior data.

Каузальний аналіз є потужним інструментом для моделювання залежностей між випадковими величинами. На відміну від класичних методів статистики і машинного навчання, каузальний аналіз моделює причинно-наслідкові зв'язки між спостережуваними величинами, що дозволяє більш точно оцінювати ефект від втручань [1], а також - робить моделі більш стійкими до змін у середовищі [3]. Одним із важливих та недосліджених можливих застосувань є пояснення та передбачення поведінки користувача у цифрових сервісах. Актуальність проблеми зумовлена, з одного боку, поширенням таких сервісів (мобільні застосунки, державні портали та ін), з іншого боку - наявністю детальних даних про активність користувача та відсутності методів, що б перетворювали складні та неоднорідні дані послідовностей подій у моделі, що пояснюють поведінку користувача та допомагають оцінювати ефект від втручань та змін у сервісі.

В цій роботі окреслюється напрямок дослідження - розробка методів побудови каузальних представлень часових даних із застосуванням до аналізу поведінки користувача в цифровому продукті.

У структурній каузальній моделі [2], спостережувані випадкові величини $X_1, X_2, \dots, X_N$ асоціюються із вершинами орієнтованого ациклічного графу (каузального графу), що описує залежності між ними у функціональній формі:

$$X_i := f_i(P A_i, U_i)$$

де $P A_i$ - вершини-батьки X_i , а U_i - випадкові шуми, незалежні між собою.

Якщо граф коректно визначений - тоді відомі відношення незалежності, що дозволяє отримати розклад сумісного розподілу $P(X_1, X_2, \dots, X_N)$ на незалежні компоненти. Маючи сумісний розподіл, можна оцінювати значення параметрів на основі спостережуваних даних.

Ми пропонуємо базову модель поведінки користувача як процесу із дискретним часом: