УДК 681.513.675

ІЛЬЇН М.І.

МОДИФІКАЦІЯ МЕТОДУ ОБЧИСЛЮВАЛЬНОЇ РЕАЛІЗАЦІЇ КРАЙОВИХ ЗАДАЧ НА ОСНОВІ Д4 ДЕКОМПОЗИЦІЇ ДЛЯ ГІБРИДНИХ СИСТЕМ НА БАЗІ ГРАФІЧНИХ ПРОЦЕСОРІВ

В роботі запропоновано паралельну модифікацію методу обчислювальної реалізації крайових задач на основі червоно-чорного впорядкування (Д4 декомпозиції) для гібридних високопродуктивних обчислювальних систем на базі графічних процесорів архітектури Nvidia CUDA. Метод відрізняється зменшеним використанням оперативної пам'яті (на 30%) та вищою продуктивністю в порівнянні з реалізацією для CPU.

Modified parallel method of boundary value problem solving is proposed. Method is based on modified red-black ordering (D4 decomposition) and uses hybrid high performance computing systems with Nvidia CUDA GPUs. Proposed modification uses 30% less memory and provides higher computational performance compared to existing CPU implementations.

Ключові слова: метод Д4 декомпозиції, обчислювальна реалізація крайових задач, GPGPU

1. Вступ

При застосуванні кінцево-різницевої апроксимації та неявних схем до рівнянь в часткових похідних моделей багатьох фізичних процесів отримувані системи лінійних алгебраїчних рівнянь (СЛАР) мають високу розмірність та як наслідок високу обчислювальну складність. Так, у випадку процесу забруднення атмосфери при дослідженні явищ дифузії, переносу, джерел викидів СЛАР 5 або 7 діагональні (у двовимірному та трьохвимірному представленні області), кількість невідомих відповідає кількості вузлів сітки – та швидко зростає при збільшенні роздільної здатності моделі. Задача моделювання має трансобчислювальну складність, є актуальним дослідження, створення та вдосконалення обчислювальних методів, що використовуються для її розв'язання.

Одним з напрямків вдосконалення є розробка обчислювальних методів на базі гібридних високопродуктивних систем, що окрім традиційних центральних процесорів (CPU) застосовують прискорювачі. Такі системи мають вищу енергоефективність (згідно рейтингу Green500 за червень 2017 р. ТОП-10 гібридні, 9 використовують прискорювачі Nvidia Tesla P100 [1]), та набувають все більшого поширення при побудові високопродуктивних систем (91 з Top500 станом на червень 2017 р., 74 використовують прискорювачі Nvidia, 17 Xeon Phi [2]). Найбільш поширеними є графічні прискорювачі архітектури Nvidia CUDA. Їх використання вимагає нетривіальної модифікації обчислювальних методів – висока продуктивність досягається завдяки великій кількості паралельних обчислювальних ядер меншої продуктивності ніж у центральному процесорі (обчислювальні алгоритми мають бути суттєво паралельні), обсяг оперативної пам'яті прискорювача суттєво менший ніж доступний для CPU.

Відомі дослідження паралельних методів для систем архітектури CUDA, в яких продемонстровано ефективність реалізації явних кінцево-різницевих схем, обчислень над щільними матрицями, ітераційних методів розв'язання СЛАР [3-5]. Разом з тим, безумовно стійкі прямі методи (що можуть бути більш ефективними для жорстких СЛАР) досліджені недостатньо, розробка методів, що враховують обмеженість пам'яті графічних процесорів також далека від завершення.

В роботі запропоновано паралельну модифікацію методу обчислювальної реалізації крайових задач на основі червоно-чорного впорядкування (Д4 декомпозиції) для гібридних високопродуктивних обчислювальних систем на базі графічних процесорів архітектури Nvidia CUDA.

2. Модель задачі

Після застосування скінченно-вимірної апроксимації вихідних рівнянь математичної фізики з подальшим представленням у формі

розріджених СЛАР високої розмірності задача зводиться до розв'язання

$$AU = B$$

спеціального вигляду.

У випадку двовимірної області, матрична модель задачі (рис. 1) може бути зведена до блочного вигляду з можливістю незалежного розв'язання для окремих блоків (рис. 2):

$$\begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix} \cdot \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = \begin{bmatrix} B^* \\ F \end{bmatrix}$$

зводиться до (рис. 3)

$$\begin{bmatrix} A_1 & A_2 \\ 0 & A_4' \end{bmatrix} \cdot \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = \begin{bmatrix} B^* \\ F' \end{bmatrix}$$

що дозволяє отримати розв'язок вихідної системи вдвічі меншого розміру:

$$A_4' U_2 = F'$$

$$U_1 = A_1^{-1} B^* - A_1^{-1} A_2 U_2$$

$$U = [U_1, U_2]$$

На рис. 1-4 проілюстровано кроки алгоритму (відомого як модифікований Д4 [6]) для випадку двовимірної області $N \times P$, $N = 4$, $P = 5$.

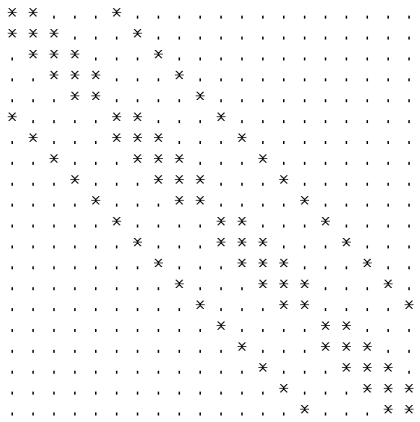

Рис. 1. Матрична модель для двовимірної області

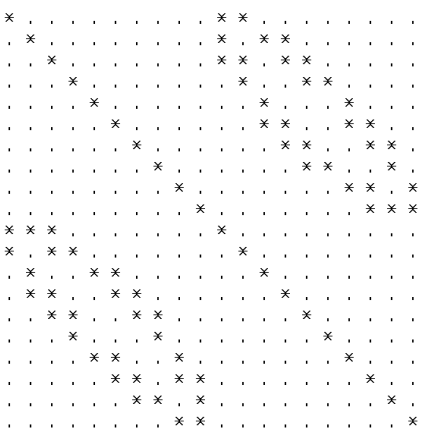

Рис. 2. Модель після застосування Д4 впорядкування

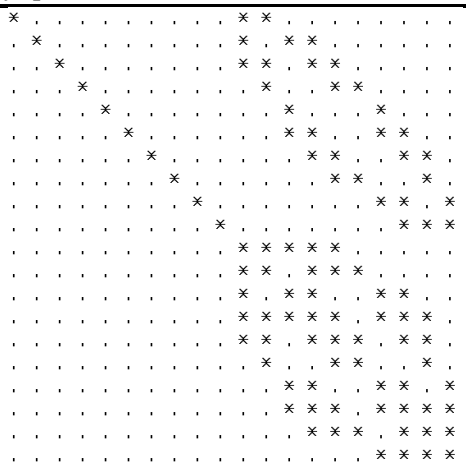

Рис.3. Перетворення А3

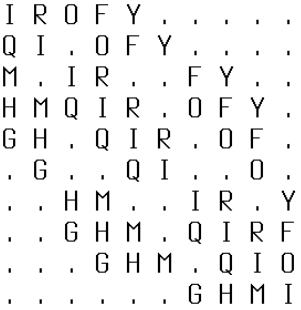

Рис.4. Аналітичний вигляд А4'

Перевагою застосування методу є можливість заміни розв'язання вихідної СЛАР розв'язком двох систем вдвічі меншого розміру. Особливістю є скорочення використання пам'яті на етапі розв'язку $A_4' U_2 = F'$, оскільки позиції ненульових елементів матриці обчислюються аналітично – без накладних витрат на збереження координат, але з більшими обчислювальними витратами на етапі звернення до елементів A_4'.

3. Паралельна модифікація Д4

Найбільш обчислювально трудомістким етапом методу є розв'язання системи $A_4' U_2 = F'$, її обчислення доцільно виконувати на графічних прискорювачах. В роботі пропонується два методи – паралельний прямий метод МД4, що використовує векторизований блочний LU розклад з вибором провідного елемента [7], та гібридний метод МД4, що використовує ітераційний стабілізований метод біспряжених градієнтів.

У прямому методі МД4 матриця A_4' розділюється на перетинаючися діагональні блоки розмірності $(k_u' + n_b) \times (k_l + n_b)$,

$k'_u = k_u + k_l$, де k_l – ширина нижньої частини стрічки, k_u – верхньої:

$$A_b^i = \begin{bmatrix} A_{11}^i & A_{12}^i & A_{13}^i \\ A_{21}^i & A_{22}^i & A_{23}^i \\ A_{31}^i & A_{32}^i & A_{33}^i \end{bmatrix}$$

де $A_{11}^i, A_{13}^i, A_{31}^i, A_{33}^i \in R^{n_b \times n_b}$, A_{12}^i, $A_{32}^i \in R^{n_b \times (k'_u - n_b)}$, A_{21}^i, $A_{23}^i \in R^{(k_l - n_b) \times n_b}$, A_{13}^i – нижня трикутна, A_{31}^i – верхня трикутна матриці, n_b – параметр алгоритму. Розрахунок LU декомпозиції здійснюється із застосуванням реалізації BLAS [7], що забезпечує можливість переносу програмної реалізації без змін вихідних кодів на нові моделі графічних процесорів. В термінах BLAS елементи розкладу $A_{ij} = L_{ij}U_{ij}$ обчислюються наступним чином:

1. Розраховується LU розклад з перестановками (алгоритм Гауса з вибором провідного елементу) матриці: $(A_{11}, A_{12}, A_{13})^T$
2. Обчислення DTRSM: $U_{12} \leftarrow L_{11}^{-1} A_{12}$
3. Обчислення DGEMM: $A'_{22} \leftarrow A_{22} - L_{21}U_{12}$
4. Обчислення DGEMM: $A'_{32} \leftarrow A_{32} - L_{31}U_{12}$
5. Обчислення DTRSM: $U_{13} \leftarrow L_{11}^{-1} A_{13}$
6. Обчислення DGEMM: $A'_{23} \leftarrow A_{23} - L_{21}U_{13}$
7. Обчислення DGEMM: $A'_{33} \leftarrow A_{33} - L_{31}U_{13}$
8. Обернена до кроку 1 перестановка системи: $(L_{11}, L_{12}, L_{13})^T$.

У гібридному методі на базі МД4 обчислення $A'_4 U_2 = F'$ виконуються ітераційно, паралельна реалізація методу біспряжених градієнтів (BiCGSTAB [8]) має вигляд:

1. $x_0 = r_0^* = U_2^*$, де U_2^* – розв'язок на попередньому кроці, 0 для першого;
   $r_0 = p_0 = F' - A'_4 x_0$
2. Цикл $j = 0,1,...$ доки $|r_j| > \varepsilon$:

$$\alpha_j = \frac{(r_j, r_0^*)}{(A'_4 p_j, r_0^*)}$$

$$s_j = r_j - \alpha_j A'_4 p_j$$

$$\omega_j = \frac{(A'_4 s_j, s_j)}{(A'_4 s_j, A'_4 s_j)}$$

$$x_{j+1} = x_j + \alpha_j p_j + \omega_j s_j$$

$$r_{j+1} = s_j - \omega_j A'_4 s_j$$

$$\beta_j = \frac{\alpha_j (r_{j+1}, r_0^*)}{\omega_j (r_j, r_0^*)}$$

$$p_{j+1} = r_{j+1} + \beta_j (p_j - \omega_j A'_4 p_j)$$

3. $U_2 = x_{j+1}$

де матричні та векторні операції обчислюються паралельно.

При експериментальному дослідженні запропоновані методи демонструють вищу продуктивність в порівнянні з Д4 та BiCGSTAB, залежність часу виконання обчислень $AU = B$ для двовимірного випадку проілюстровано на рис. 5.

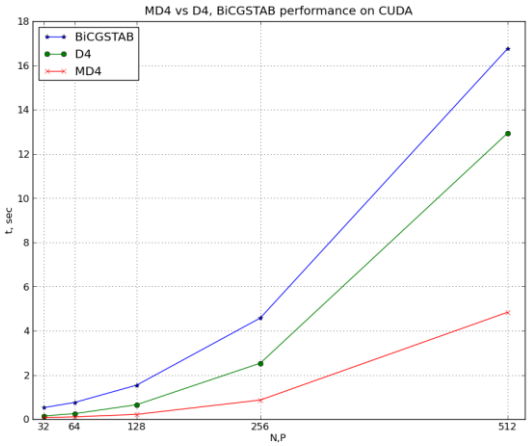

Рис.5. Експериментальна оцінка продуктивності для двовимірної області

Запропоновані методи можуть застосовуватися і для трьохвимірного випадку, за відповідної модифікації аналітичної процедури розрахунку A'_4. В рамках роботи було оцінено продуктивність тривимірної модифікації, рис. 6.

4. Оцінки скорочення вимог до оперативної пам'яті модифікованого методу

Аналітична процедура розрахунку позицій ненульових елементів A'_4 дозволяє скоротити накладні витрати на її зберігання в оперативній пам'яті графічного процесору.

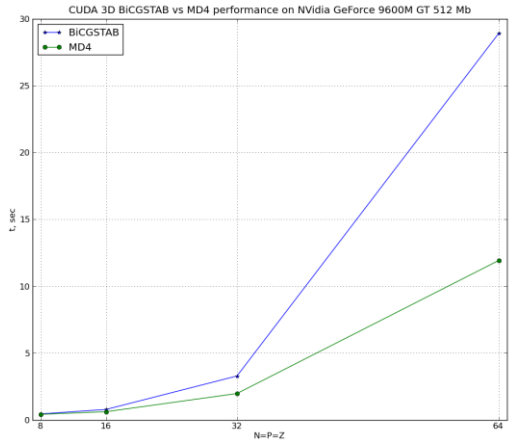

Рис.5. Експериментальна оцінка продуктивності для трьохвимірної області

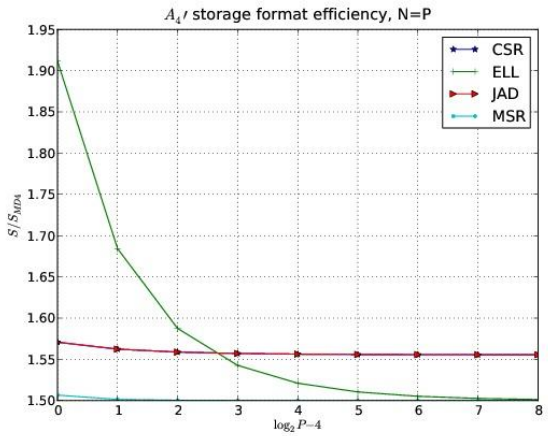

Рис.6. Експериментальна оцінка використання пам'яті для випадку N=P

Порівняємо вимоги до оперативної пам'яті модифікованого методу з відомими форматами збереження розріджених матриць. В табл. 1 наведено витрати на зберігання A_4' для двовимірних випадків $N = 2P, P = 128$.

Табл. 1. Оцінки використання пам'яті

	N_float	N_int	Всього
DNS	268435456	0	2147483648
CSR	145922	162306	1816600
COO	145922	291844	2334752
MD4	145922	0	1167376

Для кожного випадку знайдено кількість N_{float} ненульових елементів матриці A_4' та накладні витрати на збереження позицій кожного ненульового елементу (N_{int}). Позначення форматів: DNS (dense) – зберігаються всі елементи матриці, включаючи нульові; CSR (compressed sparse row) – зберігаються координати рядків та ненульових елементів відносно початку рядка (без повторень рядків), COO (coordinate list) – зберігаються дві координати ненульового елементу, MD4 (modified D4) – координати ненульових елементів не зберігаються, обчислюються аналітично на етапі розв'язання СЛАР.

При збільшенні розмірності матриці A_4' відносна частка накладних витрат на збереження позицій елементів зменшується, але складає не менше 30% від використаної пам'яті (рис. 6, 7).

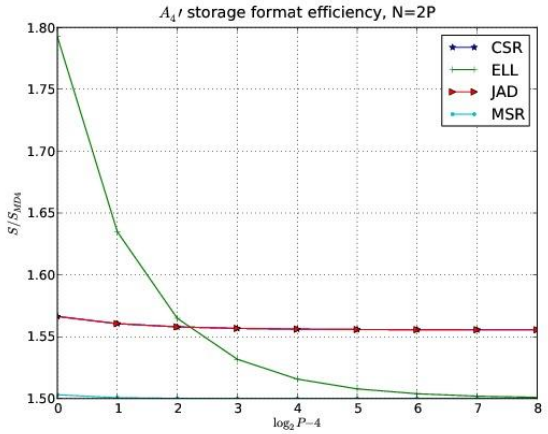

Рис 7. Експериментальна оцінка використання пам'яті для випадку N=2P

5.Висновки

В роботі запропоновано паралельну модифікацію методу обчислювальної реалізації крайових задач на основі червоно-чорного впорядкування (Д4 декомпозиції) для гібридних високопродуктивних обчислювальних систем на базі графічних процесорів архітектури Nvidia CUDA. Метод відрізняється зменшеним використанням оперативної пам'яті (на 30%) та вищою продуктивністю.

Подальші напрямки досліджень пов'язані з модифікацією запропонованих методів для платформи OpenCL, з метою застосування на прискорювачах відмінної від CUDA архітектури.

Список посилань

1. Green500 [Електронний ресурс] – Режим доступу: https://www.top500.org/green500/
2. Top500 June 2017 [Електронний ресурс] – Режим доступу: https://www.top500.org/lists/2017/06/

3. Cohen, J. Solving PDEs on Regular Grids with OpenCurrent / J. Cohen. – GTC 2010.
4. Anzt, H. Acceleration of GPU-based Krylov solvers via data transfer reduction / H. Anzt, S. Tomov, P. Luszczek, W. Sawyer, J. Dongarra // International Journal of High Performance Computing Applications.– 29. – 3. – 2015. – pp. 366-383.
5. Davis T. A Survey of Direct Methods for Sparse Linear Systems / T. Davis, S. Rajamanickam, W. Sid-Lakhdar // Acta Numerica. – 25. – 2016. – pp 383-566.
6. Згуровский М.З. Анализ и управление односторонними физическими процессами / М.З. Згуровский, А.Н. Новиков. – Киев: Наукова думка. – 1996. – С. 328.
7. Croz, J. D. Factorizations of band matrices using level 3 BLAS: Tech. Rep. 21 / J. D. Croz, P. Mayes, G. Radicati: LAPACK Working Note. – 1990. – http://www.netlib.org/lapack/lawnspdf/lawn21.pdf.
8. Saad, Y. Iterative Methods for Sparse Linear Systems, 2nd edition / Y. Saad. – PA: SIAM – 2003. – 520 p.