УДК 519.6

О. М. Хіміч, д-р фіз.-мат. наук,
В. А. Сидорук, аспірант
Інститут кібернетики ім. В. М. Глушкова НАН України, м. Київ

ГІБРИДНИЙ АЛГОРИТМ РОЗВ'ЯЗУВАННЯ СИСТЕМ ЛІНІЙНИХ РІВНЯНЬ З РОЗРІДЖЕНИМИ МАТРИЦЯМИ МЕТОДОМ ВЕРХНЬОЇ РЕЛАКСАЦІЇ

Розроблено і досліджено гібридні алгоритми неявного ітераційного методу розв'язування систем лінійних алгебраїчних рівнянь (СЛАР) з розрідженими симетричними додатно визначеними матрицями на основі трикутних методів: Зейделя, верхньої релаксації. Запропоновано підхід з попереднім перевпорядкуванням елементів вихідної матриці до блочно-діагональної матриці з обрамленням. Розглянуто питання програмної реалізації алгоритму на комп'ютерах з графічними процесорами.

Ключові слова: *паралельні обчислення, CUDA, гібридний алгоритм, СЛАР, розріджені матриці, метод верхньої релаксації.*

Вступ. Інтерес до проблеми побудови ефективних методів розв'язування СЛАР з розрідженими матрицями переважно обумовлений їх численними застосуваннями. Зокрема, системи рівнянь з розрідженими матрицями виникають у задачах аналізу міцності конструкцій в цивільному та промисловому будівництві, авіабудуванні, суднобудуванні тощо. Область застосування методів розв'язування СЛАР з розрідженими матрицями постійно розширюється.

Разом з тим вимоги до високопродуктивних обчислень набагато випереджають можливості традиційних паралельних комп'ютерів, навіть не зважаючи на багатоядерність процесорів. Розв'язання проблеми прискорення обчислень на комп'ютерах з багатоядерними процесорами розглядається в площині використання для прискорення обчислень гібридних систем на основі багатоядерних процесорів (CPU) і графічних прискорювачів (GPU).

Одним із напрямків розв'язування лінійних систем з розрідженими матрицями є ітераційні процеси на основі трикутних методів [1]: метод Зейделя, метод верхньої релаксації, метод симетричної верхньої релаксації, поперемінно-трикутний метод. В роботі розглядається гібридний алгоритм методу верхньої релаксації, у якому перевпорядкування ненульових елементів і побудова регуляризатора, виконується на CPU, а знаходження розв'язку ітераційним методом відбувається на GPU.

Постановка задачі. Розглядається СЛАР з симетричною додатно визначеною розрідженою матрицею нерегулярної структури

$$Ax = b \tag{1}$$

Для знаходження розв'язку розглянемо неявний двокроковий метод [1]

$$B\frac{x_{k+1}-x_k}{\tau_k}+Ax_k=b\ ,\ k=\overline{1,2,...} \qquad (2)$$

Оператор B вибирається із умови мінімізації числа ітерацій та його економічності. До числа економічних методів належить, зокрема, метод верхньої релаксації [1] ($\tau_k=\tau,\ \tau\in[1,2)$).

Розглянемо застосування запропонованого підходу до систем рівнянь з блочно-діагональними матрицями з обрамленням, до яких можна звести системи рівнянь з розрідженими матрицями, зокрема, методом паралельних перерізів [2].

Внаслідок використання методу паралельних перерізів, матриця матиме вигляд:

$$\tilde{A}=P^T AP=\begin{pmatrix} A_{11} & 0 & 0 & \cdots & 0 & A_{1p} \\ 0 & A_{22} & 0 & \cdots & 0 & A_{2p} \\ 0 & 0 & A_{33} & & 0 & A_{3p} \\ \vdots & \vdots & & \ddots & & \vdots \\ 0 & 0 & 0 & & A_{p-1p-1} & A_{p-1p} \\ A_{p1} & A_{p2} & Ap_3 & \cdots & A_{pp-1} & A_{pp} \end{pmatrix}, \qquad (3)$$

де P — матриця перестановок, а блоки A_{ii}, A_{ip} та A_{pi} зберігають розріджену структуру. Таким чином, задача розв'язування вихідної системи (1) зведеться до розв'язування системи

$$\tilde{A}\tilde{x}=\tilde{b} \quad \tilde{x}=P^T x, \quad \tilde{b}=P^T b\,. \qquad (4)$$

Далі розглянемо блочну реалізацію алгоритму (2).

Блочний алгоритм методу верхньої релаксації. Розглянемо метод верхньої релаксації для блочно-діагональної матриці з обрамленням вигляду (3). Представимо матрицю A у вигляді

$$A=D+L+U\ ,$$

а матрицю B з (2) у трикутному вигляді

$$B=D+\tau L\ ,$$

де

$$L=\begin{pmatrix} L_{11} & & & 0 \\ 0 & \ddots & & \\ 0 & 0 & L_{ii} & \\ A_{p1} & \cdots & A_{pi} & L_{pp} \end{pmatrix} \qquad U=\begin{pmatrix} U_{11} & 0 & 0 & A_{1p} \\ & \ddots & 0 & \vdots \\ & & U_{ii} & A_{ip} \\ 0 & & & U_{pp} \end{pmatrix}$$

$$D = \begin{pmatrix} D_{11} & & & 0 \\ & \ddots & & \\ & & D_{ii} & \\ 0 & & & D_{pp} \end{pmatrix}$$

Тоді блочний алгоритм (2) можна представити наступним чином.

Для діагональних блоків крім останнього маємо

$$x_i^{(k+1)} = \left(D_{i,i} + \tau L_{i,i}\right)^{-1}\left[\tau b_i - \tau U_{i,i} x_i^{(k)} - \tau A_{i,p} x_p^{(k)} + (1-\tau) D_{ii} x_i^k\right], \quad (5)$$

$$i = 1, \ldots, p-1.$$

Для останнього діагонального блоку має місце рівність

$$x_p^{(k+1)} = \left(D_{p,p} + \tau L_{p,p}\right)^{-1} \times$$
$$\times\left[\tau b_p - \sum_{i=1}^{p-1}\left(\tau A_{pi} x_i^{(k+1)}\right) - \tau U_{p,p} x_p^{(k)} + (1-\tau) D_{p,p} x_p^{(k)}\right]. \quad (6)$$

Далі розглянемо паралельний алгоритм методу блочної верхньої релаксації для блочно-діагональної матриці з обрамленням для комп'ютера гібридної архітектури.

Вважаємо, що в пам'яті кожного GPU знаходяться відповідні частини вектора правих частин та вектора x і відповідні блоки матриць D, L та U. Паралельний алгоритм на кожній ітерації реалізується наступним чином:

- одночасно обчислюються $x_i^{(k+1)}$ для всіх діагональних блоків крім останнього за формулою (5);
- паралельно виконуються матрично-векторні операції $\left(A_{pi} x_i^{(k+1)}\right)$;
- модифікується вектор правих частин, що відповідає останньому діагональному блоку. $\tilde{b}_p = \tau b_p - \sum_{i=1}^{p-1}\left(A_{pi} x_i^{(k+1)}\right)$;
- обчислюється вектор x для останнього діагонального блоку

$$x_p^{(k+1)} = \left(D_{p,p} + \tau L_{p,p}\right)^{-1}\left[\tilde{b}_p - U_{p,p} x_p^{(k)} + (1-\tau) D_{p,p} x_p^{(k)}\right].$$

Програмна реалізація та чисельний експеримент. Для реалізації алгоритму використовуються стандартні обчислювальні процедури (множення матриці, розв'язування трикутних систем тощо), що реалізовані у відомих бібліотеках програм, наприклад CUSPARSE [3], CUSP [4], Paralution [5]. Для зберігання матриць застосовано розріджений рядковий формат, вектори зберігаємо як щільні.

Розглянемо більш детально програмну реалізацію алгоритму, а саме роботу з GPU. Основними блоками операцій, що виконуються з

GPU є: виділення пам'яті для змінних; копіювання даних на GPU; запуск обчислень; копіювання результатів в оперативну пам'ять; звільнення пам’яті GPU.

Оскільки технічний конструктив сучасних материнських плат дозволяє встановлення в одній системі кількох процесорів і кількох графічних прискорювачів, при розробці алгоритмів і їх програмній реалізації необхідно враховувати прив’язку відповідних блоків даних до певних ядер на CPU і одного з наявних в системі GPU. Враховуючи архітектуру обчислювальної системи, яка приводиться нижче, можна зробити наступну прив’язка даних до GPU: дані, що відповідають діагональним блокам з парними номерами зберігаються на другому GPU, для блоків з непарними — на першому.

Слід зазначити, що для реалізації алгоритму на системах із спільною пам’яттю використовується технологія ***UVA*** (Unified Virtual Address) [6], що доступна в CUDA починаючи з версії 4.0. Дана технологія дозволяє працювати з адресним простором всіх GPU одночасно, що значно полегшує написання коду.

Результати експериментів наведені для гібридної архітектури 1 CPU, 1 GPU.

Розрахунки проводились на вузлі кластеру Інпарком-G [7] з наступними характеристиками.

- Процесори: 2 Xeon 5606 (4 ядра з частотою 2.13 ГГц).
- Графічні прискорювачі: 2 Tesla M2090 (6 Гб пам’яті).
- Об’єм оперативної пам’яті: 24 Гб.
- Комунікаційне середовище: InfiniBand 40 Гбіт/с (з підтримкою GPUDirect), Gigabit Ethernet.

Також на вузлах встановлена бібліотека MKL 10.2.6 та CUDA починаючи з версії 3.2.

Обчислення проводились на матрицях приведених в таблиці 1. У таблиці 2 показані часи виконання алгоритму на CPU (реалізовано з використання функцій з бібліотеки MKL) і GPU, а також наведено величину отриманого прискорення в часі виконання.

Таблиця 1

Набір тестових матриць з Флоридської колекції розріджених матриць

Назва	**Проблемна область**	**Порядок**	**Кількість ненульових елементів**
G3_circuit	circuit simulation problem	1 585 478	7 660 826
G2_circuit	circuit simulation problem	150 102	726 624
cvxbqp1	optimization problem	50 000	349 968
parabolic_fem	computational fluid dynamics problem	525 825	3 674 625
apache2	structural problem	715 176	4 817 870
minsurfo	optimization problem	40 806	203 622

Таблиця 2

Співвідношення часів виконання послідовної версії алгоритму (CPU) та гібридного алгоритму (GPU) при $\tau_k = 1.99$, $\varepsilon = 0.0001$ *і кількості діагональних блоків рівній трьом*

Назва	**CPU(сек.)**	**GPU(сек.)**	**Прискорення**
G3_circuit	213.45	22.0828	9.665893818
G2_circuit	15.7472	3.2393	4.861297194
cvxbqp1	0.812186	0.215096	3.775923
parabolic_fem	107.161	12.3211	8.697356567
apache2	343.013	46.1909	7.425986504
minsurfo	1.69252	3.77597	0.448234

На рис. 1. наведено профілі наповненості ненульовими елементами вихідних матриць. На рис. 2–4. наведено графіки залежності часів виконання на CPU і GPU, а також залежність прискорення від кількості діагональних блоків.

Найнижча ефективність алгоритму виявляється для матриць, у яких найменша заповненість блоків обрамлення A_{ip} та A_{pi} після перевпорядкування елементів розрідженої матриці. Графіки на рис. 3, 4. показують слабку залежність ефективності алгоритму від розмірності (кількості) блоків, при умові, що всі елементи матриці можуть розміститись у глобальній пам'яті GPU.

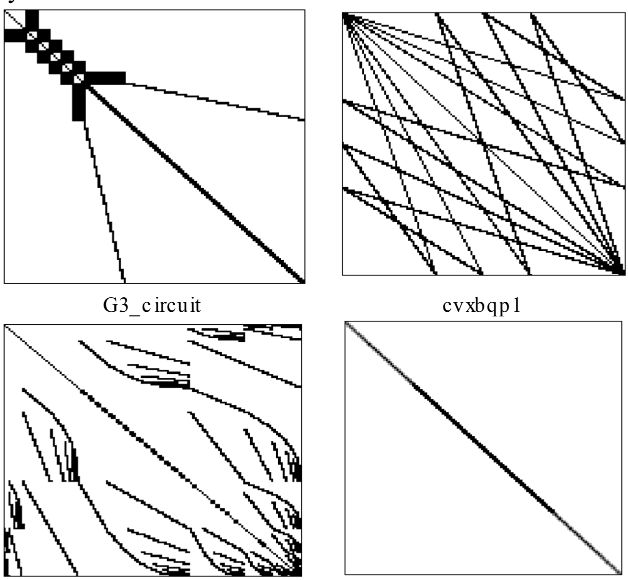

Рис. 1. *Профілі наповненості ненульовими елементами вихідних матриць*

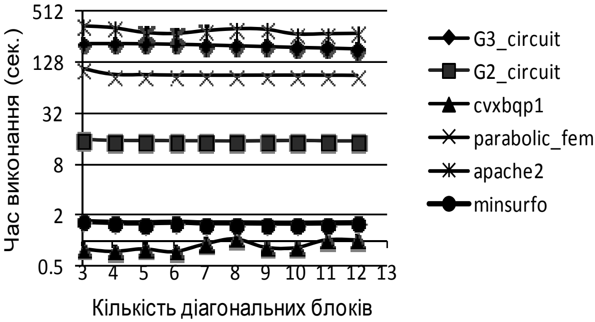

Рис. 2. *Залежність часу виконання програми на CPU від кількості діагональних блоків у матриці*

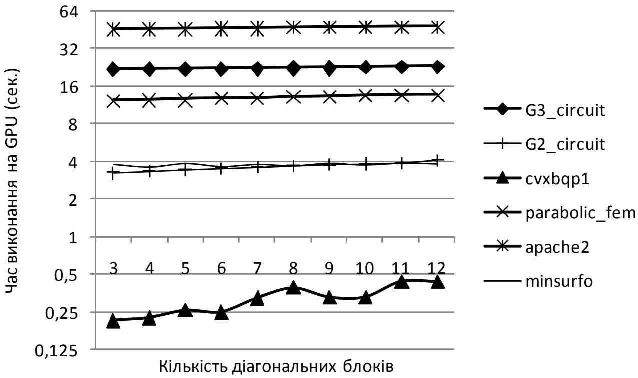

Рис. 3. *Залежність часу виконання програми на GPU від кількості діагональних блоків у матриці*

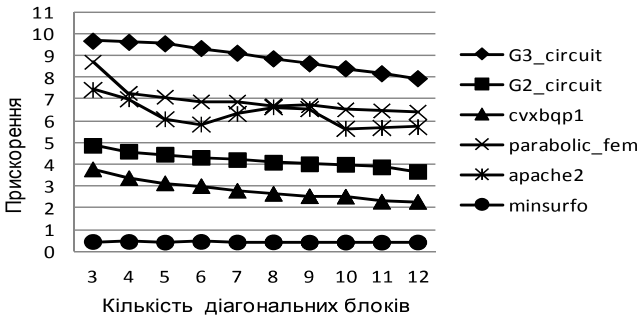

Рис. 4. *Залежність прискорення від кількості діагональних блоків у матриці*

Висновки. Розроблено та експериментально досліджено гібридний алгоритм методу верхньої релаксації для розв'язування систем лінійних алгебраїчних рівнянь з розрідженими матрицями нерегулярної структури на комп’ютерах з графічними прискорювачами. Основою пропонованого підходу є структурна регуляризація матриць (перевпорядкування елементів матриці до блочно-діагонального вигляду з обрамленням) та відомі ітераційні процедури на основі трикутних методів. Показано ефективність гібридного алгоритму для ітераційних процесів розв’язання СЛАР з розрідженими симетричними додатно визначеними матрицями на основі такого підходу. В подальшому будуть розглянуті реалізації паралельного варіанту алгоритму на архітектурах з *n* CPU, *m* GPU зі спільною або розподіленою пам’яттю.

Список використаних джерел:

1. Самарский А. А. Методы решения сеточных уравнений / А. А. Самарский, Е. С. Николаев. — М. : Наука, 1978. — 592 с.
2. Джордж А. Численное решение больших разреженных систем уравнений / А. Джордж, Дж. Лю. — М. : Мир, 1984. — 334 с.
3. CUDA CUSPARSE_Library — Santa Clara : Nvidia, 2012. — 92 p.
4. Режим доступу: http://code.google.com/p/cusp-library.
5. Режим доступу: http://www.paralution.com/
6. CUDA C Programming Guide Version 4.2. — Santa Clara : Nvidia, 2012. — 173 p.
7. Молчанов И. М. Интеллектуальные параллельные компьютеры на графических процессорах для решения научно-технических задач / И. М. Молчанов, В. И. Мова // Праці Міжнародної молодіжної математичної школи «Питання оптимізації обчислень (ПОО-XXXVII). — К. : Інститут кібернетики імені В. М. Глушкова НАН України, 2011. — С. 121-122.

A hybrid algorithm implicit iterative method for solving systems of linear algebraic equations (SLE) with sparse symmetric positive definite matrix based on triangular methods: Seidel, over relaxation is developed and investigated. The approach of the previous rearrange elements output matrix to block-diagonal matrix of the frame is proposed. The problems of software implementation of the algorithm on a computer with a graphics processors are considered.

Key words: *parallel computing, CUDA, a hybrid algorithm, SLE, sparse matrix method of over relaxation.*

Отримано: 28.10.2013