КОМПЬЮТЕРНАЯ МАТЕМАТИКА

УДК 519.6

В.А. СИДОРУК, І.А. ОЛЕНЧЕНКО

ГІБРИДНИЙ АЛГОРИТМ РОЗВ'ЯЗУВАННЯ ЛІНІЙНИХ СИСТЕМ З РОЗРІДЖЕНИМИ МАТРИЦЯМИ НА ОСНОВІ МЕТОДУ ВЕРХНЬОЇ РЕЛАКСАЦІЇ

Розглядається новий гібридний алгоритм розв'язування систем лінійних алгебраїчних рівнянь з розрідженими симетричними додатно визначеними матрицями на комп'ютерах з графічними прискорювачами. Подано результати апробації алгоритму на багатоядерному комп'ютері з графічними прискорювачами Інпарком.

Вступ. При чисельному моделюванні процесів та явищ різної природи у багатьох випадках виникає необхідність розв'язувати задачу (або декілька задач) лінійної алгебри. Наприклад, задачі лінійної алгебри виникають при дискретизації крайових задач або задач на власні значення проекційно-різницевим методом (скінченних різниць, скінченних елементів). Причому, як правило, розв'язування лінійних систем займає значну частину (50 % і більше) часу розв'язування всієї задачі загалом.

Важливою особливістю задач лінійної алгебри, що виникають при дискретизації, є високий порядок їх матриць – до десятків мільйонів. Це спричинено бажанням оперувати більш точними дискретними моделями, що дозволяють отримувати наближені розв'язки більш близькі до розв'язків вихідних задач, краще враховувати локальні особливості даного процесу або явища. Також характерною особливістю задач лінійної алгебри, які виникають при дискретизації, являється те, що кількість ненульових елементів матриць таких задач складає kn, де $k << n$, а n – порядок матриці, тобто матриці є розрідженими [1].

Структура розрідженої матриці визначається нумерацією невідомих задачі та часто є стрічковою, блочно-діагональною з обрамленням, профільною тощо.

В роботі розвивається актуальний на даний час напрямок, пов'язаний з прискоренням обчислень за рахунок використання графічних прискорювачів.

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

$$Ax = b. \qquad (1)$$

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

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

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

$$\mathrm{A} = D + L + U,$$

а матрицю B з (2) у трикутному вигляді

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

де

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

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

Тоді блочний алгоритм (2) можна представити так.

Для діагональних блоків, крім останнього, маємо

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

Для останнього діагонального блоку має місце рівність

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

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 таким чином:

- одночасно обчислюються $x_i^{(k+1)}$ для всіх діагональних блоків крім останнього за формулою (5) та 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

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

Введемо наступні позначення: $nz()$ – кількість ненульових елементів у матриці, n_i – довжина i частини вектора, що відповідає діагональному блоку з номером i, $i = \overline{1, p}$.

Знайдемо загальну кількість операцій виконуваних при реалізації гібридного алгоритму методу верхньої релаксації. При підрахунках будемо використовувати наступні факти:

- кількість операцій, виконуваних при розв’язанні трикутної системи з розрідженою матрицею становить $nz(A) - n$;
- кількість операцій, необхідних для знаходження добутку розрідженої матриці на вектор становить $2nz(A)$.

Для реалізації однієї ітерації гібридного алгоритму методу верхньої релаксації необхідно виконати

$$\alpha_i = 2nz\left(L_{ii}\right) - n_i + 2nz\left(U_{ii}\right) + 2nz\left(C_{ip}\right) + 2nz\left(C_{ip}\right) + 2nz\left(D_{ii}\right)$$

або

$$\alpha_i = 4nz\left(L_{ii}\right) + 4nz\left(C_{ip}\right) - n_i$$

арифметичних операцій з плаваючою комою на кожному GPU крім останнього.

На останньому GPU виконується

$$\alpha_p = 2nz\left(L_{pp}\right) - n_p + 2nz\left(U_{pp}\right) + 2nz\left(D_{pp}\right) = 4nz\left(L_{pp}\right) - n_p$$

операцій з плаваючою комою.

Оскільки, при приведені матриці до блочно-діагонального вигляду з обрамлення ми можемо керувати порядками блоків у розбитті, а не кількістю ненульових елементів, які в них потрапляють – загальну кількість операцій, виконуваних на всіх GPU, можна оцінити величиною

$$N_1 \approx \sum_{i=1}^{p-1} \alpha_i + \alpha_p .$$

Враховуючи той факт, що порядок останнього діагонального блоку невеликий і кількість ненульових елементів, що в ньому розміщуються є досить малою, доданком α_p можна нехтувати

$$N_1 \approx \sum_{i=1}^{p-1} \alpha_i .$$

Тоді максимальну кількість операцій, виконуваних одним GPU, можна оцінити величиною

$$N_p \approx \max_{1<i\le p-1} \alpha_i .$$

Прискорення гібридного алгоритму методу верхньої релаксації для розрідженої матриці блочно-діагональної структури з обрамленням оцінюється формулою

$$S_p \approx (p-1)\left(\left(\frac{1}{\beta}\right)\left(\max_{1\le i<p-1} \alpha_i + n_p\left(\left(p-1\right)\tau_{opp} + 2p\tau_{opg}\right)\right)\right)^{-1},$$

де $\alpha_i = 4nz\left(C_{ip}\right) + 4nz\left(L_{ii}\right) - n_i$, $\beta = \dfrac{\sum_{i=1}^{p-1} \alpha_i}{\left(p-1\right)}$.

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. Основні блоки операцій, що виконуються з GPU – це 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 та одного з наявних у системі GPU. Враховуючи архітектуру обчислювальної системи, яка приводиться далі, можна зробити наступну прив’язку 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.

Обчислення здійснювались на матрицях приведених у табл. 1. Часи виконання алгоритму на CPU (реалізовано з використання функцій з бібліотеки MKL) і GPU, а також наведено величину отриманого прискорення за часом виконання (табл. 2).

На рис. 1 – 2 показано графіки залежності часів виконання на CPU і GPU послідовної версії програми. На рис. 3. показано часи виконання програми на різних архітектурах для однакової конфігурації матриці.

Найнижча ефективність алгоритму виявляється для матриць, у яких найменша заповненість блоків обрамлення A_{ip} та A_{pi} після перевпорядкування елементів розрідженої матриці.

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

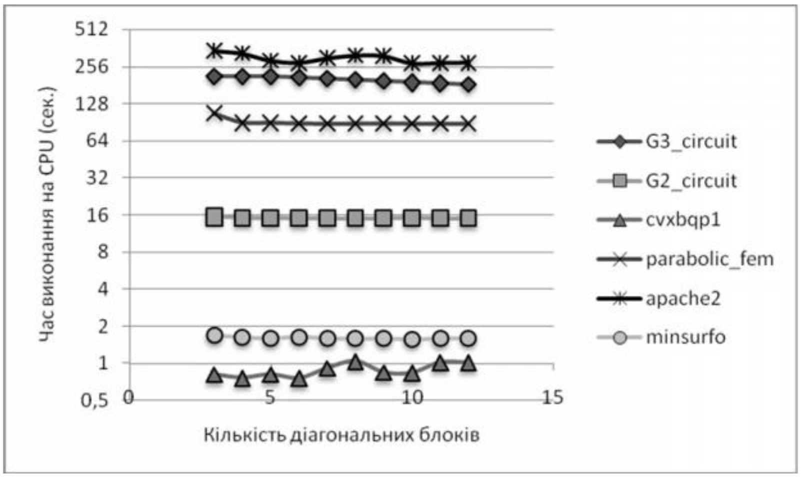

РИС. 1. Залежність часу виконання послідовної програми на CPU від кількості діагональних блоків у матриці

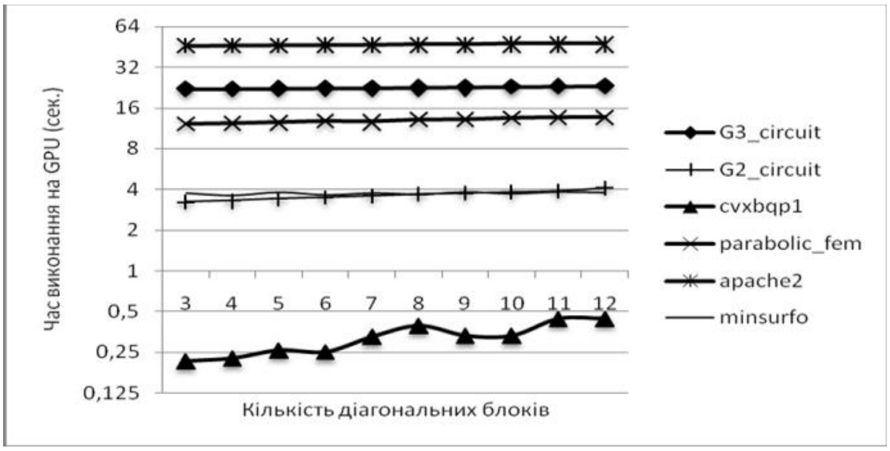

РИС. 2. Залежність часу виконання послідовної програми на GPU від кількості діагональних блоків у матриці

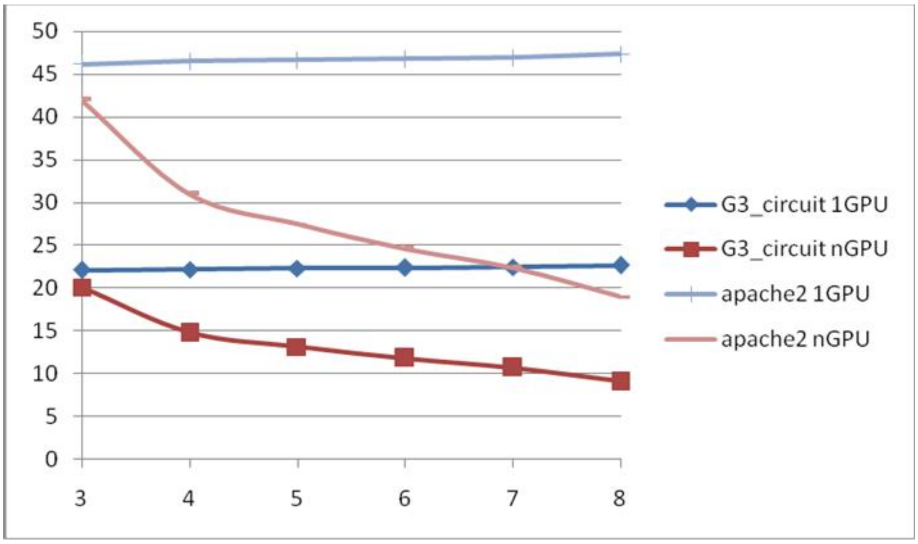

РИС. 3. Часи виконання залежно від кількості GPU

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.А. Сидорук, И.А. Оленченко

ГИБРИДНЫЙ АЛГОРИТМ РЕШЕНИЯ ЛИНЕЙНЫХ СИСТЕМ С РАЗРЕЖЕНЫМИ МАТРИЦАМИ НА ОСНОВАНИИ МЕТОДА ВЕРХНЕЙ РЕЛАКСАЦИИ

Рассматривается новый гибридный алгоритм решения систем линейных алгебраических уравнений с разрежеными симметричными положительно определенными матрицами на компьютерах с графическими ускорителями. Представлены результаты апробации алгоритма на многоядерном компьютере с графическими ускорителями Инпарком.

V.A.Sydoruk, Olenchenko I.A.

HYBRID ALGORITHM FOR SOLVING LINEAR SYSTEMS WITH SPARSE MATRICES BASED ON OVER-RELAXATION METHOD

A new hybrid algorithm for solving systems of linear algebraic equations with a sparse symmetric positive definite matrices on computers with GPU is considered. The results of testing of the algorithm on Inparcom multicore computer with GPU are presented.

1. *Самарский А.А., Николаев Е.С.* Методы решения сеточных уравнений. М.: Наука, 1978. – 592 с.
2. *Джордж А., Дж. Лю*. Численное решение больших разреженных систем уравнений. М.: Мир, 1984. 334 с.
3. CUDA CUSPARSE_Library – Santa Clara: Nvidia, 2012. 92 p.
4. http://code.google.com/p/cusp-library/
5. http://www.paralution.com/
6. CUDA C Programming Guide Version 4.2. Santa Clara: Nvidia, 2012. 173 p.

Одержано 08.06.2017

Про авторів:

Сидорук Володимир Антонович,
кандидат фізико-математичних наук, науковий співробітник
Інституту кібернетики імені В.М. Глушкова НАН України,

Оленченко Ілля Андрійович,
магістрант факультету комп'ютерних наук та кібернетики
Київського національного університету імені Тараса Шевченка.