ТЕОРІЯ ОПТИМАЛЬНИХ РІШЕНЬ

УДК 519.6

О.М. ХІМІЧ, В.А. СИДОРУК

ГІБРИДНИЙ АЛГОРИТМ ДЛЯ ЛІНІЙНОЇ ЗАДАЧІ НАЙМЕНШИХ КВАДРАТІВ З НАПІВВИЗНАЧЕНОЮ РОЗРІДЖЕНОЮ МАТРИЦЕЮ

Запропоновано алгоритм отримання нормального псевдороз-в'язку систем лінійних рівнянь з розрідженими симетричними до-датно-напіввизначеними матрицями на комп'ютерах гібридної архітектури – комп'ютерах з багатоядерними процесорами і графічними прискорювачами. Алгоритм апробовано на низці тестових задач. Показана його ефективність.

Вступ. Значна кількість прикладних задач включає у себе як складову частину знаходження розв'язку системи лінійних алгебраїчних рівнянь (СЛАР) з розрідженими матрицями. Характерною особливістю СЛАР, що виникають у тих чи інших задачах є їх великий порядок. У деяких моделях порядок матриці СЛАР може досягати кількох мільйонів.

Разом з тим вимоги до високопродуктивних обчислень набагато випереджають можливості традиційних паралельних комп'юте-рів, 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 CPU і GPU.

У даній роботі запропоновано алгоритм знаходження розв'язку задачі найменших квадратів з мінімальною нормою для симетричних додатно-напіввизначених розріджених матриць. Алгоритм є альтернативою як методу регуляризуючого функціоналу [1], так і методу дискретної регуляризації [2], в основі якого лежить, наприклад, SVD-алгоритм, оскільки в обох випадках втрачається початкова структура матриці.

Постановка задачі. Розглянемо задачу найменших квадратів

$$\min_{x\in E^n\cap\Omega}\|x\|_E, \quad \Omega = Arg\min_{x\in E^n}\|Ax-b\|_E \qquad (1)$$

з симетричною додатно-напівизначеною розрідженою матрицею ($A = A^{\mathrm{T}}$, $A \geq 0$) порядку n і рангу k, розв'язок якої існує і єдиний [2].

В основі гібридного алгоритму знаходження розв'язку задачі найменших квадратів з мінімальною нормою (нормального псевдорозв'язку відповідної несумісної системи $Ax = b$) з симетричною додатно-напіввизначеною розрідженою матрицею лежить метод паралельних перерізів приведення симетричної розрідженої матриці до блочно-діагонального вигляду з обрамленням [3, 4] та триетапного алгоритму знаходження нормального псевдорозв'язку для останньої [5].

Теоретичною передумовою методу розв'язання задачі (1) для розріджених матриць на комп'ютерах гібридної архітектури є попереднє застосування до вихідної матриці методу паралельних перерізів, який приводить вихідну матрицю до блочно-діагонального вигляду з обрамленням

$$\tilde{A} = P^T A P = \begin{pmatrix} A_{11} & 0 & 0 & \cdots & 0 & A_{1p} \\ 0 & A_{22} & 0 & \cdots & 0 & A_{2p} \\ 0 & 0 & A_{33} & & 0 & A_{3p} \\ \vdots & \vdots & & \ddots & & \vdots \\ 0 & 0 & 0 & & A_{p-1\,p-1} & A_{p-1\,p} \\ A_{p1} & A_{p2} & A_{p3} & \cdots & A_{pp-1} & A_{pp} \end{pmatrix},$$

де P – матриця перестановок, а блоки A_{ii}, A_{pi} A_{ip} зберігають розріджену структуру.

Таким чином, задача розв'язування вихідної задачі (1) зводиться до розв'язування задачі

$$\min_{x \in E^n \cap \Omega} \|y\|_E, \quad \Omega = Arg \min_{x \in E^n} \left\| \tilde{A} y - \tilde{b} \right\|_E, \tag{2}$$

$$y = P^{\mathrm{T}} x, \quad \tilde{b} = P^{\mathrm{T}} b. \tag{3}$$

Оскільки в методі паралельних перерізів виконуються ортогональні перетворення, евклідова норма відносно яких інваріантна, задачі (1), (2) і (3) – еквівалентні. При цьому очевидно зберігається і додатна-напіввизначеність матриці.

Триетапний алгоритм для знаходження нормального псевдорозв'язку (2), (3) реалізується наступним чином. Для довільного вибору параметра α (наприклад, $\alpha = 0{,}01$) реалізуємо наступні кроки алгоритму:

$$\left(\hat{A}+\alpha E\right)z=b,\left(\hat{A}+\alpha E\right)u=\hat{A}z,$$

$$u_H=\frac{u}{\max\limits_i|u_i|},$$

$$\left(\hat{A}+\alpha E\right)w=u_H,$$

$$\mu=\max_i|w_i|.$$

Якщо $\alpha\le\dfrac{\varepsilon}{2\mu\sqrt{1-\varepsilon}}$, то отримаємо розв'язок з гарантованою похибкою

$$\frac{\|x-u\|}{\|x\|}\le\varepsilon,$$

де u – наближений розв'язок задачі найменших квадратів з мінімальною нормою.

Очевидно, з точки зору комп'ютерної реалізації, алгоритм зводиться до $\hat{L}\hat{L}^T$ факторизації симетричної блочно-діагональної з обрамленням додатно-визначеної матриці $\hat{A}+\alpha E$ і багаторазового розв'язання системи лінійних рівнянь з трикутними блочними розрідженими матрицями $\hat{L}$ і $\hat{L}^T$.

Триетапна схема регуляризації є альтернативою як методу регуляризуючого функціоналу $\left(A^2+\alpha E\right)u_\alpha=Ab$ [1], так і методу дискретної регуляризації [2], в основі якого лежить, наприклад SVD-алгоритм.

Переваги алгоритму триетапної регуляризації підсилюються для матриць блочно-діагональних з обрамленням. Очевидно, що при першому підході, втрачається блочна структура матриці, а при другому – не вдається її ефективно використати.

Блочний алгоритм на основі LL^T факторизації. Враховуючи те, що матриця є блочно-діагональною з обрамленням, розглянемо блочний варіант алгоритму LL^T факторизації. Його можна представити у такій формі.

Факторизація:

для i, $i=\overline{0,p-1}$ послідовно виконуємо:

- факторизацію блоку A_{ii} $A_{ii}=L_{ii}L_{ii}^T$;
- модифікацію блоку обрамлення $L_{ii}L_{ip}=A_{ip}$;

- знаходження добутку $L_{ip}L_{ip}^T$ та модифікацію A_{pp} $\tilde{A}_{pp} = A_{pp} - \sum_{i=0}^{p-1} L_{ip}L_{ip}^T$.

На останньому кроці виконується факторизація $\tilde{A}_{pp}$ $\tilde{A}_{pp} = L_{pp}L_{pp}^T$.

Внаслідок виконання етапу факторизації матриця $\hat{L}$ матиме вигляд:

$$\hat{L} = \begin{pmatrix} L_{11} & 0 & 0 & \cdots & 0 & \\ 0 & L_{22} & 0 & \cdots & 0 & \\ 0 & 0 & L_{33} & & 0 & \\ \vdots & \vdots & & \ddots & & \vdots \\ 0 & 0 & 0 & & L_{p-1} & \\ L_{p1} & L_{p2} & L_{p3} & \cdots & L_{pp-1} & L_{pp} \end{pmatrix},$$

де $L_{pi} = L_{ip}^T$, $i = \overline{0, p-1}$.

Прямий хід:

послідовно для *i*, $i = \overline{0, p-1}$ виконуємо:

- розв'язання системи $L_{ii}y_i = b_i$;
- знаходження добутків $L_{pi}y_i$ і модифікацію *p*-ї частини вектора *b* в нульовому процесорі $\tilde{b}_p = b_p - L_{pi}y_i$.

На останньому кроці розв'язуємо систему $L_{pp}y_p = \tilde{b}_p$.

Зворотній хід:

- виконуємо розв'язання системи $L_{pp}x_p = y_p$.

Послідовно виконуємо наступні операції:

- знаходимо добутки $L_{ip}x_p$ і модифікуємо *i*-ту частину вектора *y* $y_i = y_i - L_{ip}x_p$;
- розв'язуємо систему $x_i = L_{ii}^{-1}y_i$.

Гібридний алгоритм. Розглянемо декомпозицію даних для комп'ютерів гібридної архітектури з багатоядерними (CPU) процесорами і графічними (GPU) прискорювачами. Нехай для виконання задачі маємо $p-1$ процесорне ядро. Тоді отримаємо наступний розподіл даних:

- у всіх процесах розміщуються відповідні діагональні блоки A_{ii};
- на GPU, що відповідають процесам з номерами i, $i = \overline{0, p-1}$ зберігаються відповідні блоки A_{ip} і відповідні частини векторів *x*, *y*, *b*.

Враховуючи таку декомпозицію даних можна записати гібридний алгоритм прямого методу на основі LL^T-факторизації.

Факторизація:

у всіх процесах з номерами i, $i = \overline{0, p-1}$ паралельно виконуємо:

- факторизацію блоку A_{ii} : $A_{ii} = L_{ii}L_{ii}^T$;
- копіювання на відповідний GPU блоку L_{ii}.

На GPU виконуємо:

1) модифікацію блоку обрамлення $L_{ip} = L_{ii}^{-1}A_{ip}$;
2) знаходження добутку $L_{ip}L_{ip}^T$.

Копіюємо на відповідний CPU результат виконання $L_{ip}L_{ip}^T$ і виконуємо мультизбирання та модифікацію A_{pp} у процесі з номером 0. $\tilde{A}_{pp} = A_{pp} - \sum_{i=0}^{p-1} L_{ip}L_{ip}^T$.

В нульовому процесі виконується факторизація $\tilde{A}_{pp}$: $\tilde{A}_{pp} = L_{pp}L_{pp}^T$. Копіюємо результат у відповідний GPU.

Прямий хід:

на GPU, що відповідають процесам з номерами i, $i = \overline{0, p-1}$ виконуємо:

- розв'язання системи $L_{ii}y_i = b_i$;
- знаходження добутків $L_{pi}y_i$.

Копіюємо результати виконання $L_{pi}y_i$ на відповідний процесор і виконуємо мультизбирання та модифікацію p-ї частини вектора b в нульовому процесі $\tilde{b}_p = b_p - L_{pi}y_i$.

На GPU, що відповідає процесу з номером 0 копіюємо $\tilde{b}_p$ і розв'язуємо систему $L_{pp}y_p = \tilde{b}_p$.

Зворотній хід:

на GPU, що відповідає процесу з номером 0 розв'язуємо систему $L_{pp}x_p = y_p$.

Копіюємо результат виконання на процесор і виконуємо розсилку x_p у інші процесори.

На відповідних GPU виконуються наступні операції:

- знаходимо добутки $L_{ip}x_p$ і модифікуємо i-у частину вектора y $y_i = y_i - L_{ip}x_p$;
- розв'язуємо систему $L_{ii}x_i = y_i$.

Результати чисельних експериментів. Для реалізації стандартних обчислювальних процедур (множення матриць, розв'язування трикутних систем) можна використовувати функції відомих бібліотек, наприклад, CUSPARSE [6],

CUSP [7], Paralution [8]. Для зберігання матриць застосовано розріджений рядковий формат, вектори зберігаємо як щільні. Слід зазначити, що в програмній реалізації алгоритму для факторизації діагональних блоків A_{ii} використовувалась функція факторизації стрічкової матриці з бібліотеки LAPACK, що міститься в MKL, блоки A_{ip} і A_{pp} зберігаються як щільні.

Розрахунки проводились на вузлах кластеру Інпарком-G [9], які мають наступні характеристики:

- процесори: 2 Xeon 5606 (8 ядер) з частотою 2.13 ГГц;
- графічні прискорювачі: 2 Tesla M2090;
- обсяг оперативної пам'яті: 24 Гб;
- комунікаційне середовище: InfiniBand 40 Гбіт/с (з підтримкою GPUDirect), Gigabit Ethernet.

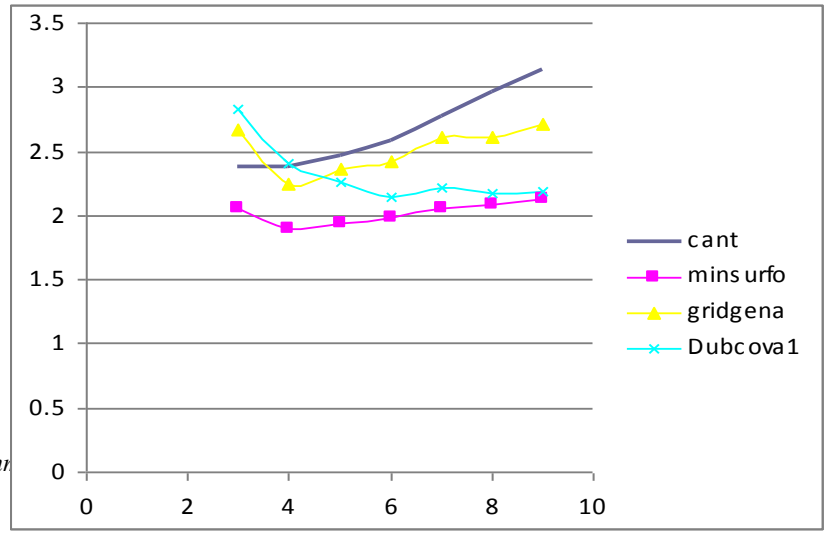
Також на вузлах встановлена бібліотека MKL 10.2.6 та CUDA починаючи з версії 3.2.

Чисельні експерименти здійснювались на наступних розріджених матрицях, характеристики яких наведені в таблиці. На рис. 1, 2 показані графіки, що показують часи виконання гібридного алгоритму і величину отриманих прискорень.

Отримані результати відповідають архітектурі 1 CPU та 1 GPU.

ТАБЛИЦЯ. Характеристики тестових матриць

Назва	Проблемна область	Порядок матриці	Кількість ненульових елементів
Dubcova1	2D/3D problem	16130	253009
minsurfo	optimization problem	40806	203622
gridgena	optimization problem	48962	512084
cant	2D/3D problem	62452	4007383



Кількість діагональних блоків

РИС. 1. Час виконання гібридного алгоритму

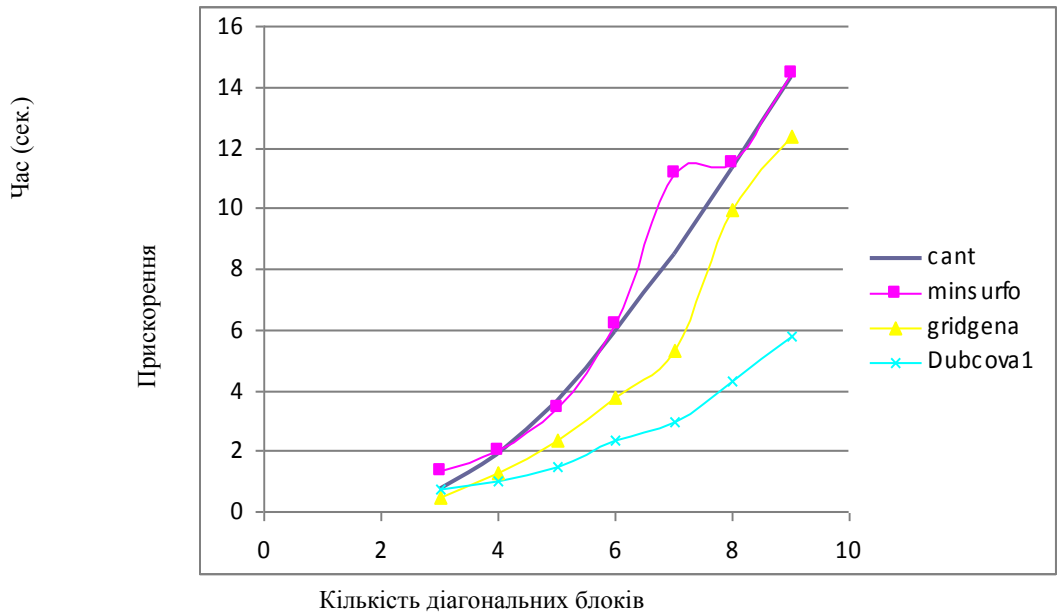


РИС. 2. Графік отриманих прискорень в залежності від кількості діагональних блоків гібридного алгоритму в порівнянні з блочним алгоритмом реалізованим на CPU

Висновки. Розроблено і експериментально досліджено гібридний алгоритм прямого методу для лінійної задачі найменших квадратів з розрідженою симетричною додатно-напіввизначеною матрицею. Отримані результати для гібридної архітектури з одним CPU і одним GPU показують ефективність алгоритму. Більший ефект можна очікувати від реалізації паралельного варіанта даного алгоритму на архітектурах типу n CPU + m GPU. Прискорення обчислень можна чекати також, використовуючи, наприклад, для факторизації діагонального блоку алгоритми, які значну кількість обчислень реалізують на GPU.

Зауваження. Слід зазначити, що згідно роботи [3], вищенаведений алгоритм має місце і для задачі з наближеними даними.

А.Н. Химич, В.А. Сидорук

ГИБРИДНЫЙ АЛГОРИТМ ДЛЯ ЛИНЕЙНОЙ ЗАДАЧИ НАИМЕНЬШИХ КВАДРАТОВ С РАЗРЕЖЕННОЙ ПОЛУОПРЕДЕЛЕННОЙ МАТРИЦЕЙ

Предложен алгоритм получения нормального псевдорешения системы линейных уравнений с разреженными симметричными положительно-полуопределенными матрицами на компьютерах гибридной архитектуры – компьютерах с многоядерными процессорами и графическими ускорителями. Алгоритм апробирован на ряде тестовых задач. Показана его эффективность.

A.N. Khimich, V.A. Sydoruk

HYBRID ALGORITHM FOR LINEAR LEAST SQUARES PROBLEM WITH SPARSE SEMIDEFINITE MATRIX

An algorithm of obtaining the normal pseudosolution of systems of linear equations with sparse symmetric positive-semidefinite matrices on hybrid architecture computers – computers with multicore processors and graphics accelerators is proposed. The algorithm is tested on a set of test problems. Shown its effectiveness.

1. *Тихонов А.Н., Арсенин В.Я.* Методы решения некорректных задач. – М.: Наука, 1986. – 287 с.
2. *Воеводин В.В., Кузнецов Ю.А.* Матрицы и вычисления. – М.: Наука, 1984. – 318 с.
3. *Джордж А., Лю Дж.* Численное решение больших разреженных систем уравнений. – М.: Мир, 1984. – 333 с.
4. *Хіміч О.М., Полянко В.В.* Оптимізація паралельного ітераційного процесу для лінійних систем з розрідженими матрицями // Теорія оптимальних рішень. – 2011. – № 10 – С. 47 – 53.
5. *Химич А.Н., Яковлев М.Ф.* О решении систем с матрицами неполного ранга // Компьютерная математика. – Киев: Ин-т кибернетики имени В.М. Глушкова НАН Украины, 2003. – № 1. – С. 119 – 125.
6. *CUDA* CUSPARSE_Library – Santa Clara: Nvidia, 2012. – 92 p.
7. http://code.google.com/p/cusp-library/
8. http://www.paralution.com/
9. *Молчанов И.Н., Химич А.Н., Мова В.И., Николайчук А.А.* Интеллектуальный персональный компьютер гибридной архитектуры // Искусственный интеллект. – 2012. – № 3. – С. 73–78.

Одержано 11.04.2014