УДК 519.6

В.А. СИДОРУК

ОДИН ІТЕРАЦІЙНИЙ МЕТОД РОЗВ'ЯЗУВАННЯ СЛАР З РОЗРІДЖЕНИМИ СИМЕТРИЧНИМИ МАТРИЦЯМИ

Запропоновано алгоритм розв'язування систем лінійних алгебраїчних рівнянь з симетричними розрідженими матрицями методом симетричної верхньої релаксації на комп'ютерах з процесором Intel Xeon Phi. *Приведено результати апробації алгоритму на персональному суперкомп'ютері Інпарком_ph.*

© В.А. Сидорук, 2018

Вступ. Велика кількість наукових та практичних задач, зокрема при дослідженні стійкості конструкцій, розрахунку динаміки напружено-деформованого стану об'єктів різної природи та інші, зводяться до розв'язання часткової узагальненої алгебраїчної проблеми власних значень з симетричними стрічковими додатно-означеними матрицями великих розмірів.

Важливою особливістю таких задач лінійної алгебри, які виникають при дискретизації, є те, що кількість ненульових елементів матриць складає kn, де $k << n$, а n – порядок матриці, тобто матриці є розрідженими [1, 2]. Структура розрідженої матриці визначається нумерацією невідомих задачі і часто є стрічковою, блочно-діагональною з обрамленням, профільною і тому подібне. В даній статті розглядаються симетричні додатно-означені матриці блочно-діагонального виду з обрамленням.

Для розв'язування систем лінійних алгебраїчних рівнянь з розрідженою структурою матриць великих розмірів виникає необхідність створення нових алгоритмів, які забезпечують підвищення ефективності обчислень на комп'ютерах з багатоядерними процесорами Intel Xeon Phi.

Постановка задачі. Розглянемо задачу розв'язання системи лінійних алгебраїчних рівнянь (СЛАР)

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

з розрідженою симетричною додатно визначеною матрицею нерегулярної структури.

Використаємо неявний однокроковий метод [3]:

$$B\frac{x_{k+1}-x_k}{\omega_{k+1}}+Ax_k=b\,,\ k=1,2,\ldots,B\in H, \tag{2}$$

де ω_{k+1} – ітераційний параметр.

Як відомо від оператора (регуляризатора) *B* залежать як число ітерацій, які потрібно виконати для отримання заданої точності, так і число арифметичних операцій, необхідних для виконання однієї ітерації. Нині розвиток ітераційних методів йде шляхом конструювання економічних операторів. Зокрема, до таких належать оператори, яким відповідають діагональна, тридіагональна, трикутна матриці, а також їх комбінації.

Розглянемо випадок коли матриця системи блочно-діагонального вигляду з обрамленням. Матрицю такої структури можна отримати методами перевпорядкування ненульових елементів розрідженої матриці, зокрема, методом паралельних перерізів, який описано вище. Після використання методу паралельних перерізів матриця набуває вигляду:

$$A=P^TAP=\begin{pmatrix} D_1 & 0 & 0 & \cdots & 0 & C_1\\ 0 & D_2 & 0 & \cdots & 0 & C_2\\ 0 & 0 & D_3 & & 0 & C_3\\ \vdots & \vdots & & \ddots & & \vdots\\ 0 & 0 & 0 & & D_{p-1} & C_{p-1}\\ C_1 & C_2 & C_3 & \cdots & C_{p-1} & D_p \end{pmatrix},$$

де *P* – матриця перестановок, а блоки D_i та C_i зберігають розріджену структуру. Таким чином, задача розв'язування вихідної системи (1) зводиться до розв'язування системи $\hat{A}\hat{x}=\hat{b},\quad \hat{x}=P^{\mathrm{T}}x,\quad \hat{b}=P^{\mathrm{T}}b$.

Далі розглянемо паралельний алгоритм на основі трикутних ітераційних методів для комп'ютерів на базі процесорів Intel Xeon Phi.

Паралельний алгоритм методу симетричної верхньої релаксації. Одним з методів розв'язання системи (1) є метод симетричної верхньої релаксації. Метод симетричної верхньої релаксації [3, 4] базується на застосуванні двох методів верхньої релаксації, що поєднуються таким чином, що матриця ітерацій, яка отримується в результаті була подібна до симетричної.

Метод симетричної верхньої релаксації у матричній формі описується наступними співвідношенням:

$$\left(D-\omega L\right)x_{k+\frac{1}{2}}=\left(\omega U+\left(1-\omega\right)D\right)x_k+\omega b, \tag{3}$$

$$\left(D-\omega U\right)x_{k+1}=\left(\omega L+\left(1-\omega\right)D\right)x_{k+\frac{1}{2}}+\omega b. \tag{4}$$

$$L = \begin{pmatrix} L_1 & & & 0 \\ 0 & \ddots & & \\ 0 & 0 & L_i & \\ C_1 & \cdots & C_i & L_p \end{pmatrix}, \quad U = \begin{pmatrix} U_1 & 0 & 0 & C_1 \\ & \ddots & 0 & \vdots \\ & & U_i & C_i \\ 0 & & & U_p \end{pmatrix},$$

$$D = \begin{pmatrix} \tilde{D}_1 & & & 0 \\ & \ddots & & \\ & & \tilde{D}_i & \\ 0 & & & \tilde{D}_p \end{pmatrix},$$

де L_i ($1 \leq i \leq p$) – нижні трикутні матриці діагональних блоків D_i без діагональних елементів, U_i – верхні трикутні матриці діагональних блоків D_i без діагональних елементів, блоки $\tilde{D}_i$ є нижні трикутні матриці блоків D_i.

Для реалізації алгоритму будемо використовувати блочний розподіл даних: у пам'яті CPU(i), $i = \overline{1, p-1}$ зберігаються відповідні блоки L_i, U_i, $\tilde{D}_i$, C_i; в пам'яті CPU(p) зберігаються блоки L_p, U_p, $\tilde{D}_p$.

Паралельний алгоритм реалізації (3) має наступний вигляд:

на всіх CPU(i) $i = \overline{1, p-1}$ обчислюємо $x_i^{\left(k+\frac{1}{2}\right)}$

$$x_i^{\left(k+\frac{1}{2}\right)} = \left(\tilde{D}_i - \omega L_i\right)^{-1}\left(\omega b_i + \omega U_i x_i^{(k)} + \omega C_i x_p^{(k)} + (1-\omega) D_i x_i^{(k)}\right)$$

і знаходимо добутки

$$G_i = C_i x_i^{(k+\frac{1}{2})};$$

за допомогою мультизбирання модифікуємо вектор b_p

$$b_p^* = \omega b_p - \sum_{i=1}^{p-1} G_i;$$

у CPU(p) обчислюється вектор $x_p^{\left(k+\frac{1}{2}\right)}$, за формулою:

$$x_p^{\left(k+\frac{1}{2}\right)} = \left(\tilde{D}_p + \omega L_p\right)^{-1}\left[b_p^* + U_p x_p^{(k)} + (1-\tau)\tilde{D}_p x_p^{(k)}\right].$$

Паралельний алгоритм реалізації (4) має наступний вигляд:

на всіх CPU(i) $i = \overline{1, p-1}$ обчислюємо

$$b_i^* = \left(\omega L_i + (1-\omega)\tilde{D}_i\right) x_i^{(k+\frac{1}{2})} + \omega b_i$$

і знаходимо добутки $G_i = C_i x_i^{(k+\frac{1}{2})}$;

за допомогою мультизбирання модифікуємо вектор b_p

$$b_p^* = \omega b_p + \sum_{i=1}^{p-1} G_i;$$

у CPU(p) обчислення проводяться за наступною формулою:

$$b_p^* = \left(\tilde{D}_p + \omega L_p\right) x_p^{(k+\frac{1}{2})} + b_p^*;$$

у CPU(p) виконуємо розв'язання системи з верхньою трикутною матрицею $x_p^{(k+1)} = \left(\tilde{D}_p - \omega U_p\right)^{-1} b_p^*$ і виконуємо мультирозсилку $x_p^{(k+1)}$ у CPU(i), $i = \overline{1, p-1}$;

у CPU(i), $i = \overline{1, p-1}$ модифікуємо b_i^* $b_i^* = b_i^* - C_i x_p^{(k+1)}$;

в CPU(i), $i = \overline{1, p-1}$ виконуємо розв'язання трикутних систем

$$x_i^{(k+1)} = \left(\tilde{D}_i - \omega U_i\right)^{-1} b_i^*.$$

Для оцінки якості паралельних алгоритмів будемо використовувати коефіцієнт прискорення S_p, що обчислюються за формулою

$$S_p = T_1 / T_p,$$

де p – кількість використовуваних для розрахунків CPU, T_1 – час розв'язування задачі на гібридному комп'ютері з одним CPU, T_p – час розв'язування тієї ж задачі на комп'ютері з використанням p CPU.

$$T_1 = Nt_c,\ T_p = Nt_c + M_1 t_{opp} + Q_1 t_{cpp},$$

де t_c – середній час виконання однієї арифметичної операції на GPU, t_{opp} – час, необхідний для обміну одним машинним словом між двома процесами, t_{cpp} – час, який потрібний для встановлення зв'язку між двома процесами, M_i, Q_i – кількість відповідних обмінів та синхронізацій.

Для паралельного алгоритму методу симетричної верхньої релаксації коефіцієнт прискорення оцінюється наступним співвідношенням:

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

де $\alpha_i = 8nz(C_i) + 8nz(L_i) - 2n_i$, $\beta = \dfrac{\sum_{i=1}^{p-1}\alpha_i}{(p-1)}$, $nz()$ – кількість ненульових елементів у матриці, n_i – довжина i частини вектора, що відповідає діагональному блоку з номером i, $i = \overline{1, p}$.

Результати чисельних експеритментів. Розрахунки проводились на персональному суперкомп'ютері Інпарком-xp [5, 6], який має такі характеристики:

- процесор: Intel Xeon Phi (64 ядра) з частотою 2.13 ГГц;
- обсяг оперативної пам'яті: 192 Гб;
- комунікаційне середовище: Gigabit Ethernet.

У програмній реалізації даного алгоритму, для виконання основних матрично-векторних операцій використовувались функції з бібліотеки Intel MKL [7].

Чисельні експерименти проводились на розріджених матрицях, що наведені в табл. 1. Також у таблиці наведені характеристики матриці такі як: порядок матриці, кількість ненульових елементів.

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
parabolic_fem	computational fluid dynamics problem	525 825	3 674 625
apache2	structural problem	715 176	4 817 870

Результати розрахунків тестових задач наведені в табл. 2.

ТАБЛИЦЯ 2. Часи виконання розрахунків при $\varepsilon = 0.0001$ і кількості діагональних блоків рівній кількості ниток

Назва	1 CPU (сек.)	4 CPU (сек.)	8 CPU (сек.)	16 CPU (сек.)	32 CPU (сек.)	64 CPU (сек.)
G3_circuit	512.36	426.9667	213.4833	53.37083	6.671354	2.084798
G2_circuit	170.7867	142.3222	71.16111	17.79028	2.223785	0.694933
parabolic_fem	448.735	373.9458	186.9729	46.74323	5.842904	1.825907
apache2	237.945	124.6486	62.32431	15.58108	1.947635	0.608636

На рисунку далі показано профілі наповненості ненульовими елементами матриць з Флоридської колекції розріджених матриць (https://sparse.tamu.edu/). Дана колекція містить еталонні матриці з різних предметних областей: оптимізаційні задачі, задачі стійкості, задачі газодинаміки і т. д.

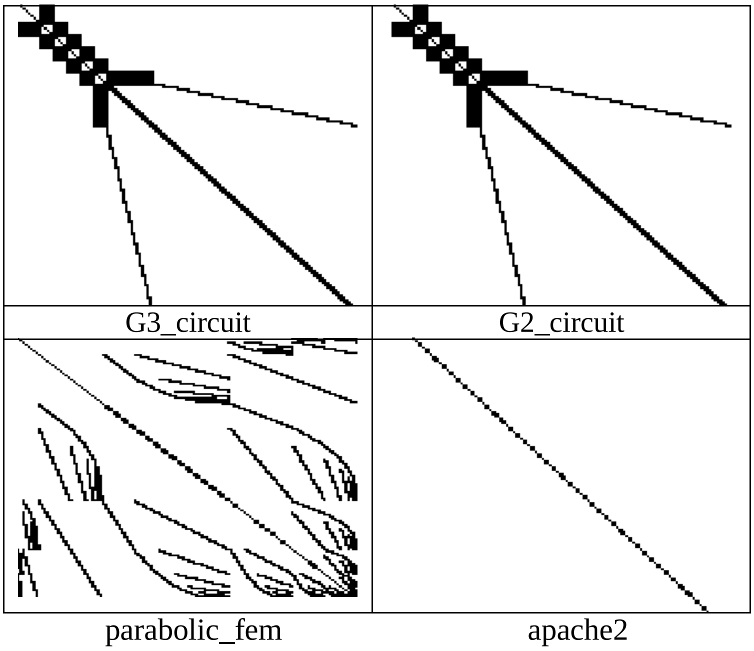

РИСУНОК. Портрети тестових матриць

Висновки. Розроблено та експериментально досліджено паралельний алгоритм методу симетричної верхньої релаксації для розв'язування систем лінійних алгебраїчних рівнянь з розрідженими матрицями нерегулярної структури на комп'ютерах з процесорами Intel Xeon Phi. Основою пропонованого підходу є структурна регуляризація матриць (перевпорядкування елементів матриці до блочно-діагонального вигляду з обрамленням) та застосування відомих ітераційних процедур на основі трикутних методів.

Показано, що виконання структурної регуляризації, використання на всіх етапах обчислень функцій з ефективних програмних бібліотек, а також урахування особливостей архітектури комп'ютера дають можливість отримати суттєве прискорення.

В.А. Сидорук

ОДИН ИТЕРАЦИОННЫЙ МЕТОД РЕШЕНИЯ СЛАУ
С РАЗРЕЖЕННЫМИ СИММЕТРИЧНЫМИ МАТРИЦАМИ

Предложен алгоритм решения систем линейных алгебраических уравнений с симметричными разреженными матрицами методом симметричной верхней релаксации на компьютерах с процессором Intel Xeon Phi. Приведены результаты апробации алгоритма на персональном суперкомпьютере Инпарком_ph.

V.A. Sydoruk

ONE ITERATIVE METHODS FOR SOLVING OF SLAE WITH SPARSE SYMMETRIC MATRIX

The algorithm for solving systems of linear algebraic equations with symmetric sparse matrices is proposed in the article by the method of symmetric upper relaxation on computers with the Intel Xeon Phi processor. The results of approbation of the algorithm on the personal supercomputer Inparcom_ph are presented.

Список літератури

1. Писанецки С. Технология разреженных матриц. М.: Мир, 1988.
2. Джордж А., Лю Дж. Численное решение больших разреженных систем уравнений. М.: Мир, 1984. 334 с.
3. Самарский А.А., Николаев Е.С. Методы решения сеточных уравнений. М.: Наука, 1978. 592 с.
4. Сидорук В.А., Оленченко І.А. Гібридний алгоритм розв'язування лінійних систем з розрідженими матрицями на основі методу верхньої релаксації. *Компьютерная математика*. 2017. Вип. 1. С. 150 – 157.
5. Химич А.Н., Молчанов И.Н., Мова В.И. и др. Численное программное обеспечение MIMD-компьютера Инпарком. Киев: Наукова думка, 2007. 222 с.
6. Молчанов И.Н., Перевозчикова И.В., Химич А.Н. Опыт разработки семейства кластерных комплексов Инпарком. *Кибернетика и системный анализ*. 2009. 45, № 6. С. 88 – 96.
7. Math Kernel Library (Intel (R) MKL) Documentation. https://software.intel.com/en-s/intel-mkl

Одержано 04.04.2018