

# Паралельний поперемінно трикутний метод для розв'язування СЛАР з розрідженими матрицями

Хіміч О.М., Полянко В.В.

Інститут Кібернетики ім. В.М. Глушкова НАН України, просп. Глушкова 40, Київ, Україна

khimich\_ic@mail.ru, polyanko\_victor@ukr.net

**Анотація.** У роботі досліджується паралельна реалізація неявних ітераційних методів розв'язування систем лінійних алгебраїчних рівнянь (СЛАР) з розрідженими симетричними додатно визначеними матрицями з використанням поперемінно-трикутного методу (ПТМ) для побудови передобумовлювача. Запропоновано ефективний спосіб розподілу даних, який базується на попередньому структуруванні розрідженої матриці методом паралельних перерізів (МПП). Побудовано паралельний алгоритм ітераційного методу знаходження розв'язку СЛАР, обчислено величини прискорення та ефективності алгоритму. Запропонований алгоритм може бути використаний у прикладних задачах, де розв'язування СЛАР з розрідженими матрицями є основним або проміжним етапом розрахунків.

## Ключові слова

паралельні обчислення, системи лінійних алгебраїчних рівнянь, розріджені матриці, поперемінно-трикутний метод, метод паралельних перерізів, метод спряжених градієнтів.

## 1 Вступ

При розв'язуванні СЛАР з матрицями великих порядків не рідко реалізація прямих методів, навіть тих, що враховують розрідженість матриці, на комп'ютері може виявитися досить складною і неефективною. Проблеми машинної реалізації прямих методів пояснюються обмеженням об'єму оперативної пам'яті та зниженням швидкодії при використанні дискової. Запис систем з матрицями, які мають певну специфіку, у вигляді процедури обчислення вектора  $Ax$  потребує у ряді випадків менше комп'ютерної пам'яті, ніж при реалізації прямих методів. У такому випадку для розв'язування СЛАР доцільно використовувати швидкозбіжні ітераційні методи.

Значного ефекту від використання ітераційних методів можна отримати у задачах з розрідженими матрицями, оскільки при використанні прямих методів відбувається збільшення кількості ненульових елементів, а відтак зростають вимоги до пам'яті та об'єму обчислень. Особливо відчутною може бути різниця у випадку паралельних обчислень, оскільки тоді час розв'язування задачі можна скоротити ще більше.

## 2 Теоретична частина

Для знаходження наближеного розв'язку системи лінійних алгебраїчних рівнянь

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

з додатно визначеною симетричною матрицею  $A$  ітераційні наближення обчислюватимемо за допомогою методу спряжених градієнтів [1]

$$Bx_{k+1} = \alpha_{k+1}(B - \tau_{k+1}A)x_k + (1 - \alpha_{k+1})Bx_k + \alpha_{k+1}\tau_{k+1}b, \tag{2}$$
$$k = 1, 2, \dots,$$

$$Bx_1 = (B - \tau_1A)x_0 + \tau_1b, x_0 \in H,$$

де ітераційні параметри  $\alpha_{k+1}$  і  $\tau_{k+1}$  знаходяться за формулами

$$\tau_{k+1} = \frac{(w_k, r_k)}{(Aw_k, w_k)}, \alpha_{k+1} = \left( 1 - \frac{\tau_{k+1}}{\tau_k} \frac{(w_k, r_k)}{(w_{k-1}, r_{k-1})} \frac{1}{\alpha_k} \right)^{-1}, k = 0, 1, \dots, \alpha_1 = 1. \quad (3)$$

За виконання умов:

$$\gamma_1 B \leq A \leq \gamma_2 B, \gamma_1 > 0, \quad B = B^* > 0,$$

для визначення кількості ітерацій для досягнення точності  $\varepsilon$  справедлива оцінка [1]

$$n \geq n_0(\varepsilon) = \ln(0.5 \varepsilon) / \ln \rho, \quad (4)$$

де  $\rho = \frac{(1 - \sqrt{\xi})}{(1 + \sqrt{\xi})}$ ,  $\xi = \gamma_1 / \gamma_2$ .

## 2.1 Поперемінно-трикутний метод (ПТМ)

Основними вимогами при побудові ефективного ітераційного методу є економічність обчислень та швидка збіжність. З цієї точки зору варто зазначити, що оцінка (4) вказує на можливість прискорення збіжності ітераційного процесу у випадку вибору такого  $B$ , при якому відношення  $\gamma_1/\gamma_2$  буде максимальним. Крім того, природною є вимога економічності обернення  $B$ .

Вимоги економічності задовольняє поперемінно-трикутний метод (ПТМ), суть якого полягає у побудові оператора  $B$  у спеціальному факторизованому вигляді

$$B = (E + \omega R_1)(E + \omega R_2),$$

де  $\omega$  – ітераційний параметр, що буде визначений далі,  $R_1$  та  $R_2$  – оператори, яким відповідають трикутні матриці. Ці оператори визначаються з умов

$$\begin{aligned} \hat{A} &= R_1 + R_2, & \gamma_1(Bx, x) &\leq (\hat{A}x, x) \leq \gamma_2(Bx, x); \\ (R_1x, u) &= (x, R_2u), & u, x &\in H. \end{aligned}$$

ПТМ належить до класу методів, ітераційні параметри для яких вибираються з урахуванням апріорної інформації про оператори ітераційної схеми. Для ПТМ ця інформація полягає у заданні величин  $\delta$  та  $\Delta$  з нерівностей

$$\|x\|_A^2 \geq \delta \|x\|^2, \quad \|R_2x\| \leq \Delta \|x\|_R^2 / 4.$$

Тоді, згідно [1]  $\gamma_1 = \delta(1 + \omega\delta + \delta\Delta\omega^2/4)^{-1}$ ,  $\gamma_2 = 1/2\omega$ . З умови максимуму відношення  $\gamma_1/\gamma_2$  приходимо до виразу для оптимального параметру  $\omega$ :  $\omega = \frac{2}{\sqrt{\delta\Delta}}$ . При цьому  $\frac{\gamma_1}{\gamma_2} = 2\sqrt{\eta}(1 + \sqrt{\eta})^{-1}$ ,  $\eta = (\Delta\delta)^{-1}$  [3].

## 2.2 Паралельний алгоритм поперемінно-трикутного методу

Метод на основі схеми (2) можна ефективно розпаралелити у випадку симетричної матриці  $A$ . Попередньо застосовуючи метод паралельних перерізів [2] задачу розв'язування вихідної системи (1) зведемо до розв'язування еквівалентної системи

$$\hat{A}\hat{x} = \hat{b}, \hat{A} = P^T A P, \hat{x} = P^T x, \hat{b} = P^T b$$

де  $P$  – матриця перестановок:  $P = (e^{\sigma_{(i)}})^T$ ,  $\sigma_{(i)}$  – номер вершини у новій нумерації, а  $e_i$  – вектор довжини  $n$ , всі компоненти якого, окрім  $i$ -ї зі значенням 1, дорівнюють 0. Матриця  $P$ , отримана методом паралельних перерізів, зводить вихідну матрицю системи  $A$  до вигляду

$$\hat{A} = P^T A P = \begin{pmatrix} D_1 & 0 & 0 & \dots & 0 & B_1 \\ 0 & D_2 & 0 & \dots & 0 & B_2 \\ 0 & 0 & D_3 & & 0 & B_3 \\ \vdots & \vdots & & \ddots & & \vdots \\ 0 & 0 & 0 & & D_{p-1} & B_{p-1} \\ B_1 & B_2 & B_3 & \dots & B_{p-1} & D_p \end{pmatrix},$$

де блоки  $D_i$  відповідають розділеним групам вершин графа матриці, а  $B_i$  – множинам вершин розділювачів. Для ПТМ передобумовлювач має вигляд

$$B = (E + \omega \hat{R})(E + \omega \hat{R}^T), \quad \hat{A} = \hat{R} + \hat{R}^T$$

При цьому  $\hat{R}$  зберігає блочно-трикутну структуру успадковану від матриці  $\hat{A}$ :

$$\hat{R} = \begin{pmatrix} R_1 \\ R_2 \\ R_3 \\ \vdots \\ R_{p-1} \\ R_p \end{pmatrix}^T = \begin{pmatrix} \tilde{D}_1 & 0 & 0 & \dots & 0 & 0 \\ 0 & \tilde{D}_2 & 0 & \dots & 0 & 0 \\ 0 & 0 & \tilde{D}_3 & & 0 & 0 \\ \vdots & \vdots & & \ddots & & \vdots \\ 0 & 0 & 0 & & \tilde{D}_{p-1} & 0 \\ C_1 & C_2 & C_3 & \dots & C_{p-1} & \tilde{D}_p \end{pmatrix}$$

де блоки  $\tilde{D}_i$  ( $1 \leq i \leq p$ ) є нижніми трикутниками блоків  $D_i$ .

Розглянемо комп'ютер MIMD-архітектури з розподіленою пам'яттю. Нехай  $\hat{A}$  – квадратна матриця порядку  $n$ . Представимо матрицю у блочному вигляді наступним чином

$$\hat{A} = \begin{pmatrix} A_1 & A_2 & \dots & A_p \end{pmatrix}^T,$$

де  $A_i = (0, \dots, 0, D_i, 0, \dots, 0, B_i)$ ,  $i < p$ ,  $A_p = (0, 0, \dots, 0, D_p)$ ,  $p$  – кількість процесів. Розташуємо матрицю по процесам таким чином, щоб  $A_i \in P_i$ ,  $i = \overline{1, p}$ . Зауважимо, що при такому способі розподілу кожен процес зберігає лише ненульові блоки матриці  $\hat{R}$ . З огляду на структуру  $\hat{R}$  це означає, що процеси з номерами  $1 \leq i < p$  зберігають блоки  $\tilde{D}_i$  та  $C_i$ , а процес з номером  $p$  зберігає блок  $\tilde{D}_p$ , де  $p$  – загальна кількість процесів. З урахуванням величин блоків матриці розподіляється по процесам вектор  $x$ . Позначимо  $x_i$  – частина вектора  $x$ , що зберігається у  $i$ -му процесі.

Паралельна форма алгоритму визначається блочно-трикутною структурою матриць  $R_i$  при розв'язанні системи  $Bw = r$ . Тобто паралельний алгоритм розв'язування нижньої трикутної системи

$$(E + \omega \hat{R})y = r$$

зводиться до одночасного та незалежного розв'язування систем

$$(E + \omega \tilde{D}_q)y_q = r_q, \quad 0 \leq q \leq p$$

та наступного обчислення

$$\tilde{y}_q = C_q y_q, \quad 0 \leq q < p$$

де  $q$  – номер процесу, після чого останній процес сумує значення  $\tilde{y}_q$ , надіслані від інших процесів:  $y_p = \sum_{q=1}^{p-1} \tilde{y}_q$ .

Аналогічно, для знаходження розв'язку системи

$$(E + \omega \hat{R}^T)w = y$$

одночасно з тим, як останній процес знаходить компоненти розв'язку системи

$$(E + \omega \tilde{D}_p^T)w_p = y_p$$

і розсилає їх іншим процесам, усі процеси незалежно розв'язують системи

$$(E + \omega \tilde{D}_q^T)w_q = y_q - C_q^T w_p$$

## 2.3 Прискорення та ефективність паралельного алгоритму

Нехай  $n$  – порядок матриці,  $p$  – кількість процесів,  $z$  – число ненульових елементів матриці, що зберігаються в одному процесі. Не втрачаючи загальності, вважаємо, що всі процеси мають однакову кількість рядків:  $s=n/p$ .

Визначимо кількість арифметичних операцій (додавання та множення), які виконуються на одному кроці ітераційного методу в однопроцесорному випадку. Для знаходження нев'язки  $r_k$  потрібно виконати  $2z$  операцій, при розв'язуванні рівняння поправок –  $2z+n$  операцій, обчислення ітераційних параметрів  $\alpha_{k+1}$  і  $\tau_{k+1}$  вимагає  $2z+n$  операцій, а для знаходження нового наближення  $x_{k+1}$  –  $7n$  операцій. Отже, всього при виконанні одного кроку ітераційного методу кількість операцій дорівнює

$$N_1 = 6z + 9n.$$

Отриманий результат можна використати для знаходження кількості арифметичних операцій у випадку  $p$  процесів. Для цього слід врахувати, що операція скалярного добутку кожним процесором застосовується для  $s$  елементів векторів (а не для  $n$ ), а матричні операції проводяться лише для ненульових елементів, що зберігаються безпосередньо в процесі, кількість яких, як було зазначено вище, становить  $3s^2/2$  для частини матриці  $\hat{A}$  у кожному процесі. Таким чином, маємо

$$N_p = 9s(1 + s) \quad \text{або} \quad N_p = 9(n/p)(1 + n/p).$$

Зазначимо, що при заміні у виразі для  $N_1$  значення кількості ненульових елементів  $z$  на максимально величину, при якій можливе застосування алгоритму для  $p$  процесів (тобто  $3s^2p/2$ ), отримаємо

$$N_1 = 9(s^2p + n) \quad \text{або} \quad N_1 = 9n(1 + n/p).$$

Знайдемо кількість обмінів. При обчисленні нев'язки  $r_k$  пересилається  $s(p-1)$  елементів, при розв'язуванні рівняння поправок  $-2s(p-1)$  елементів. При знаходженні параметрів  $a_{k+1}$  і  $\tau_{k+1}$  двічі викликається функція зведення для обчислення скалярних добутків, що вимагає ще  $2(p-1)$  пересилань. Отже, маємо

$$M_p = (3s + 2)(p - 1) \quad \text{або} \quad M_p = (3(n/p) + 2)(p - 1).$$

При цьому, число синхронізацій, що відбуваються між процесами, складає

$$Q_p = 4(p - 1).$$

Обчислимо значення коефіцієнтів прискорення та ефективності алгоритму:  $S_p = T_1/T_p$ ,  $E_p = S_p/p$ , де  $T_1$  – час розв'язування задачі на одному процесі,  $T_p$  – час розв'язування на  $p$  процесорах. Часи розв'язування обчислюються як  $T_1 = N_1t$ ,  $T_p = N_p t + M_p t_o + Q_p t_c$ , де  $t$  – час виконання однієї арифметичної операції,  $t_o$  – час виконання одного обміну,  $t_c$  – час однієї синхронізації. Отже,

$$T_1 = 9n(1 + n/p) t, \quad T_p = 9(n/p)(1 + n/p) t + (3(n/p) + 2)(p - 1)t_o + 4(p - 1)t_c.$$

Отже,

$$S_p \approx p \left( 1 + \frac{p}{3n} \tau_o + \frac{4p^2}{9n^2} \tau_c \right)^{-1}, \quad E_p \approx \left( 1 + \frac{p}{3n} \tau_o + \frac{4p^2}{9n^2} \tau_c \right)^{-1}.$$

Звідси випливає, що запропонований алгоритм характеризується високою масштабованістю, а ефективність застосування багатопроцесорної системи тим доцільніша, чим більший порядок матриці.

### 3 Висновки

У роботі розроблено та досліджено паралельний поперемінно трикутний метод з попереднім структуруванням матриці методом паралельних перерізів. Факторизоване представлення передобумовлювача в поперемінно трикутному методі зводить процес передобумовлення до розв'язання систем з блочно-діагональними трикутними матрицями, що визначає високу ступінь паралелізму та збалансованості паралельного ПТМ. У рамках проведених досліджень отримано оцінки похибок розв'язків отриманих ітераційними методами, знайдено характеристики ефективності та прискорення алгоритмів.

### Посилання

- [1] Самарский А.А. Методы решения сеточных уравнений / Самарский А.А., Николаев Е.С. – М.: Наука, 1978. – 592 с.
- [2] Джордж А. Численное решение больших разреженных систем уравнений / А. Джордж, Дж. Лю. – М.: Мир, 1984. – 334 с.
- [3] Хімич О.М. Оптимізація паралельного ітераційного процесу для лінійних систем з розрідженими матрицями / О.М. Хімич, В.В. Полянко // Теорія оптимальних рішень – 2011. - № 10. – с. 136-142.