

РОЗВ'ЯЗАННЯ ГІДРОДИНАМІЧНОЇ ЗАДАЧІ ЗА МЕТОДОМ БАГАТОТОЧКОВОГО ПОШУКУ У РОЗПОДІЛЕНОМУ ОБЧИСЛЮВАЛЬНОМУ СЕРЕДОВИЩІ

Національний авіаційний університет

Запропоновано багатоточковий метод чисельного розв'язання гідродинамічної задачі. Розглянуто задачу оптимізації розподілу обчислювального навантаження між вузлами розподіленого обчислювального середовища при реалізації цього методу. Використання запропонованого методу дозволяє обійти локальні мінімуми цільової функції та оптимізувати використання обчислювальних ресурсів.

Ключові слова: гідродинамічна задача, система різницевих рівнянь, розподілене обчислювальне середовище.

Вступ. Постановка проблеми

При чисельному розв'язанні задач механіки суцільних середовищ та електродинаміки, моделювання розподілених процесів сітковими методами задача зводиться до систем однотипних, порівняно простих алгебраїчних або диференціальних рівнянь великої розмірності. Для пошуку розв'язків таких систем звичайно використовують ітераційні методи, що включає в себе виконання великої кількості однотипних математичних операцій. При цьому необхідно забезпечити збіжність ітераційного обчислювального процесу, і розв'язати задачу оптимального розподілу (або «балансування») обчислювальних завдань між вузлами багатопроцесорної обчислювальної системи.

Аналіз досліджень і публікацій

Відомі методи балансування навантажень для багатопроцесорних обчислювальних систем поділяють на статичні та динамічні [1-4].

Статичне балансування виконується до запуску розподіленої програми. При цьому рівномірний розподіл обчислювального навантаження між процесорами намагаються забезпечити за рахунок вибору підобластей, які мають однакову кількість вузлів, мінімальну кількість спільних границь (граней), мінімальну розмірність матриці різницевих рівнянь, мінімальну кількість зв'язків. Інколи, додатково, висувається вимога забезпечити відповідність топології і розміру підобластей можливостям сіткових генераторів [1].

До алгоритмів динамічного балансування відносять такі, які допускають перерозподіл областей в ході обчислень. Програмне забезпечення, що реалізує динамічне балансування, визначає навантаження обчислювальних вузлів, пропускну здатність ліній зв'язку, а також частоту обміну повідомленнями між логічними процесорами системи. На основі цих даних визначається, де саме буде виконуватися нове завдання, також тут визначається, які дії необхідно виконати, щоб перенести частину області чи областей з більш завантажених обчислювальних вузлів на менш завантажені. Для динамічного балансування можуть бути використані методи статичного балансування, застосовані з певним кроком по часу [5], а також і спеціальні методи балансування, що не мають статичних аналогів а для підвищення ефективності – паралельні версії послідовних алгоритмів [6, 7]. Огляд і порівняння динамічних методів балансування наведено в роботі [8].

До статичних методів балансування відносять: метод половинного ділення; комбінаторні методи; комбіновані підходи. До динамічних методів балансування – дифузний алгоритм; метод потенціалу; багаторівневі методи [9].

Найбільш поширені методи балансування ґрунтуються на застосуванні геометричних алгоритмів розбиття. Для розрахункових схем, що використовують регулярні сітки, задача декомпозиції сітки по даним вирішується, зазвичай, за допомогою розбиття області на блоки стовпців чи рядків [10, 11]; в

трьохвимірному просторі – розбиття на шари. Вертикальне та горизонтальне розбиття називають одновимірною декомпозицією, оскільки поділ відбувається в одному сітковому напрямку. Також застосовуються так звані «матричні» методи розбиття, або двовимірна декомпозиція, коли поділ відбувається одночасно за двома сітковими напрямками. Ці методи себе добре зарекомендували, вони досить прості в реалізації і дозволяють отримати добре збалансовану систему. Недоліком таких методів є те, що вони не є інваріантними до викривлення сітки. При зміні координат вузлів сітки доводиться проводити декомпозиції обчислювальної області.

Ціль статті

Ціль статті полягає в тому, щоб запропонувати багатоточковий метод пошуку розв'язку гідродинамічної задачі і дослідити задачу призначення обчислювальних робіт на вузли мережі при реалізації цього методу у розподіленому обчислювальному середовищі.

Основна частина

В якості модельної використано двовимірну гідродинамічну задачу, розглянуту у статті [12]. Система рівнянь Нав'є-Стокса зводиться на прямокутній розрахунковій сітці: до наступної різницевої схеми:

$$2f_{i,j} \frac{f_{i+1,j} - f_{i,j}}{\Delta_x} + g_{i,j} \frac{f_{i,j+1} - f_{i,j}}{\Delta_y} + f_{i,j} \frac{g_{i,j+1} - g_{i,j}}{\Delta_y} + \frac{1}{\rho} \frac{P_{i+1,j} - P_{i,j}}{\Delta_x} - v \frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{\Delta_x^2} - v \frac{f_{i,j+1} - 2f_{i,j} + f_{i,j-1}}{\Delta_y^2} = 0 ; \quad (1)$$

$$g_{i,j} \frac{f_{i+1,j} - f_{i,j}}{\Delta_x} + f_{i,j} \frac{g_{i+1,j} - g_{i,j}}{\Delta_x} + 2g_{i,j} \frac{g_{i,j+1} - g_{i,j}}{\Delta_y} + \frac{1}{\rho} \frac{P_{i,j+1} - P_{i,j}}{\Delta_y} - v \frac{g_{i+1,j} - 2g_{i,j} + g_{i-1,j}}{\Delta_x^2} - v \frac{g_{i,j+1} - 2g_{i,j} + g_{i,j-1}}{\Delta_y^2} = 0 ; \quad (2)$$

$$\frac{f_{i+1,j} - f_{i,j}}{\Delta_x} + \frac{g_{i,j+1} - g_{i,j}}{\Delta_y} = 0 , \quad (3)$$

де $f_{i,j}, g_{i,j}$ – шукані значення функцій $U_X(x,y) = f(x,y); U_Y(x,y) = g(x,y)$ – компонентів вектора швидкості середовища за координатами x та y відповідно – в точках розрахункової сітки з номерами (i,j) ;

$P_{i,j}$ – шукані дискретні значення функції тиску $P = P(x,y)$ в точках розрахункової сітки з номерами (i,j) ;

v та ρ – в'язкість та щільність рідини (постійні величини);

Δ_x та Δ_y – кроки розрахункової сітки за координатами x та y відповідно.

У рівняннях системи (1) – (3) використано порядок індексів, що відповідає алфавітному розташуванню змінних (перший індекс відповідає координаті x , другий – координаті y).

Спроби розв'язувати систему (1) – (3) показали, що метод градієнтного спуску не дозволяє знайти глобальний мінімум цільової функції, оскільки вона має велику кількість локальних мінімумів. Для прикладу розглянемо розв'язання задачі про встановлене ламінарне протікання рідини через трубу. Опишемо цю задачу системою (1) – (3) на обчислювальній області 5×5 точок в просторі двох параметрів:

$$\xi_1 = U_x [i,2] = U_x [i,4], \\ \xi_2 = U_x [i,3], i = 1..5. \quad (4)$$

Ця задача має тривіальний розв'язок $[\xi_1 = 1.5; \xi_2 = 2]$. Однак процес градієнтного спуску зупиняється в одному з численних локальних мінімумів цільової функції, що не дозволяє отримати бажаний результат.

З метою подолання цієї проблеми ми пропонуємо модифікувати цільову функцію. Нехай пошук розв'язку відбувається у просторі узагальнених параметрів $\{\xi_p\}$, $p = 1..P$. До числа цих параметрів можуть входити фізичні величини або узагальнені, пов'язані з фізичними певними співвідношеннями. Розмірність простору пошуку P може дорівнювати кількості шуканих фізичних змінних або відрізнятися від неї. Так, в багатьох задачах можна зменшити розмір-

ність простору пошуку на основі певних припущень про вигляд розв'язків системи, як у прикладі (4) Нехай

$$H(\xi_1, \xi_2, \dots, \xi_p) - \quad (5)$$

нев'язка системи (1) – (3) при значеннях параметрів $\{\xi_1, \xi_2, \dots, \xi_p\}$. З метою обходу локальних мінімумів, введемо множину векторів зсуву у просторі узагальнених параметрів

$$\{\Delta^{(s)}\}, s=1..S, \quad (6)$$

і замість невід'язки (5) будемо оптимізувати суму невід'язок

$$\sum_{s=1}^S H(\xi_1 + \Delta_1^{(s)}, \xi_2 + \Delta_2^{(s)}, \dots, \xi_p + \Delta_p^{(s)}),$$

де $\Delta_j^{(s)}$ – j -й компонент s -го вектора зсуву у просторі параметрів. У випадку двовимірного простору параметрів, це означатиме, що ми оптимізуємо не невід'язку системи в одній пошуковій точці, яка зрештою в результаті пошуку співпадає з точкою мінімуму цільової функції, а суму невід'язок у групі точок, розташованих навколо цільової за певним законом. Вибором кількості цих точок та закону їх розташування ми зможемо позбутися впливу локальних мінімумів на результати пошуку. Для обчислення такої цільової функції, всі обчислення у кожній точці сітки доведеться повторити S разів, щоразу вводячи відповідний зсув на один з вектор $\Delta^{(s)}$ з множини (6). З метою оптимізації кількості обчислень пропонується динамічна реконфігурація множини (6), що може включати в себе використання змінної кількості пошукових векторів в ході пошуку та варіацію їхніх компонентів.

Для створення початкової моделі розподілу обчислень розглянемо вертикальний та горизонтальний поділи обчислювальної області на домени (підобласті), що є зонами відповідальності окремих обчислювальних вузлів (рис. 1, *a*, *б*). Кількість таких доменів дорівнює кількості обчислювальних вузлів системи. Вертикальний та горизонтальний поділи розрізняються принципово; анізотропія області обумовлена тим, що розглядається задача про протікання рідини у «горизонтальному» напрямку, тому «ліва» та «права» (за рис.1) границі області є перерізами вхідного

та вихідного потоків, а «верхня» та «нижня» відповідають твердим стінкам.

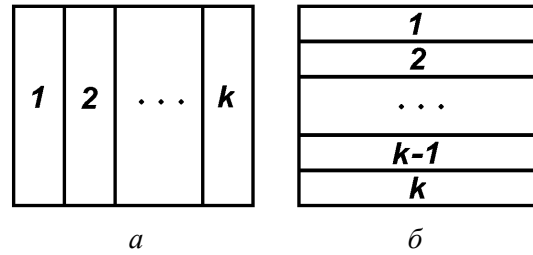


Рис. 1. «Вертикальне» (*a*) та «горизонтальне» (*б*) розбиття обчислювальної області на домени

Нехай використовується розрахункова сітка $N \times M$, де N – кількість точок по вертикалі (кількість «рядків матриці»); M – кількість точок по горизонталі (кількість «стовпців матриці»). Нехай кількість обчислювальних вузлів дорівнює k . Такою ж має бути кількість доменів, на які розбиваємо обчислювальну область.

Позначимо O_a кількість числових значень, що передаються між точками, сусідніми у вертикальному напрямку, але належними до різних доменів (рис. 1, *a*). Позначимо O_b кількість числових значень, що передаються між точками, сусідніми у горизонтальному напрямку і належними до різних доменів (рис. 1, *б*).

З системи рівнянь (1) – (3) видно, що обчислення невід'язок у кожній з точок області вимагає врахування значень функцій у сусідніх точках. Для розрахунку невід'язки системи рівнянь у даній точці з індексами (i, j) необхідно отримати 5 значень величин, що відносяться до точок, сусідніх з даною у вертикальному напрямку $(f_{i+1,j}, g_{i+1,j}, P_{i+1,j}, f_{i-1,j}, g_{i-1,j})$, і 5 значень величин, що відносяться до точок, сусідніх з даною у горизонтальному напрямку $(f_{i,j+1}, g_{i,j+1}, P_{i,j+1}, f_{i,j-1}, g_{i,j-1})$. В обох випадках дані про три величини передаються у «від'ємному» (відносно індексації точок) напрямку, а про дві – у «додатному». Отже, якщо дві сусідні по вертикалі чи по горизонталі точки віднесені до різних обчислювальних вузлів, то для обчислення невід'язок на одному кроці ітераційного процесу між ними необхідно передати $O_v = O_h = 5$ значень, 3 з яких буде передано в один бік і 2 – в другий.

При розбитті вертикальними границями (випадок рис. 1, а) кількість обмінів на одному кроці розрахунку дорівнює:

$$Q_a = (N \cdot O_h - c_1) \cdot (k - 1), \quad (7)$$

де множник $(k - 1)$ представляє кількість границь, що використовуються у розбитті рис. 1, а; $N \cdot O_h$ – кількість даних, які необхідно було б передавати через одну границю між доменами довжиною N за відсутності особливостей, пов'язаних з наявністю граничних умов; c_1 – поправка кількості обмінів, що відображає особливості формулювання граничних умов. Якщо за постановкою задачі граничні умови задані на краях обчислювальної області рівностями з постійними величинами, то для них не потрібно здійснювати на кожній ітерації інформаційні обміни між обчислювальними вузлами; достатньо передати і зафіксувати це значення однократно. Так, наприклад, у задачі, що розглядається, на верхній та нижній границях розрахункової області з умов нерозривності та прилипання витікає, що компоненти швидкості мають постійні значення:

$$\begin{aligned} \forall j f_{0,j} = 0; \forall j f_{N-1,j} = 0; \\ \forall j g_{0,j} = 0; \forall j g_{N-1,j} = 0. \end{aligned}$$

Тобто, між двома сусідніми по горизонталі точками верхньої або нижньої границь через границю домену, проведену вертикально, необхідно передавати тільки одне значення $P_{i,j+1}$. Тому

$$c_1 = 2 \cdot 4 = 8. \quad (8)$$

Підставивши (8) в вираз (7), отримаємо:

$$Q_a = (5N - 8) \cdot (k - 1). \quad (9)$$

З аналогічних міркувань знайдемо кількість обмінів даними між обчислювальними вузлами при розбитті розрахункової області горизонтальними границями, як на рис. 1, б:

$$Q_b = (M \cdot O_v - c_2) \cdot (k - 1), \quad (10)$$

де c_2 – поправка – кількість обмінів, які можна не виконувати при «горизонтальному»

розбитті внаслідок особливості формулювання граничних умов. Для задачі про «горизонтальне» протікання рідини через трубу будемо вважати, що величини тиску на вхідному і вихідному перерізах постійні по всьому перерізу (експериментальний об'єкт підключено до джерела тиску – аналог джерела напруги, що розглядається у теорії електричних ланцюгів), а вертикальні (поперечні) компоненти швидкості на вхідному і вихідному перерізах відсутні. Тоді

$$\begin{aligned} \forall i P_{i,0} = const; \forall i P_{i,M-1} = const; \\ \forall i g_{i,0} = 0; g_{i,M-1} = 0; \\ c_2 = 2 \cdot 3 = 6, \end{aligned} \quad (11)$$

Підставимо (11) в вираз (10), отримаємо

$$Q_b = (5M - 6) \cdot (k - 1). \quad (12)$$

Знайдемо кількість обмінів при змішаному розбитті (рис. 2). Нехай розрахункова сітка $N \times M$ точок розбита на k_1 вертикальних та k_2 горизонтальних смуг, $k = k_1 \cdot k_2$ – загальна кількість доменів. За постановкою задачі $k_1 \geq 2$, $k_2 \geq 2$ (якщо хоч одна з цих нерівностей не виконується, то змішане розбиття вироджується в один з випадків, розглянутих вище), тому $k \geq 4$.

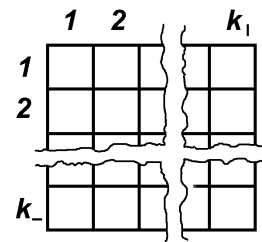


Рис. 2. «Змішане» розбиття обчислювальної області на домени

Загальна кількість обмінів при змішаному розбитті дорівнює:

$$\begin{aligned} Q_m &= (N \cdot 5 - 8) \cdot (k_1 - 1) + (M \cdot 5 - 6) \cdot (k_2 - 1) = \\ &= (5N - 8) \cdot (k_1 - 1) + (5M - 6) \cdot \left(\frac{k}{k_1} - 1 \right). \end{aligned} \quad (13)$$

З порівняння виразів (9) та (13) визначимо, при яких значеннях k_1 змішане розбиття

дасть меншу кількість обмінів, ніж вертикальне.

$$Q_a > Q_m ; \quad (14)$$

$$(5N - 8) \cdot (k - 1) > (5N - 8) \cdot (k_1 - 1) + (5M - 6) \cdot (k_1 - 1);$$

$$(5N - 8) \cdot (k - 1) - (5N - 8) \cdot (k_1 - 1) - (5M - 6) \cdot \left(\frac{k}{k_1} - 1 \right) > 0;$$

$$(5N - 8) \cdot (k - k_1) - (5M - 6) \cdot \left(\frac{k}{k_1} - 1 \right) > 0;$$

$$(5N - 8) \cdot (k_1 k - k_1^2) - (5M - 6) \cdot (k - k_1) > 0. \quad (15)$$

Позначимо

$$5N - 8 = C_1, \quad 5M - 6 = C_2. \quad (16)$$

Тоді з (15) отримаємо:

$$C_1 \cdot (k_1 k - k_1^2) - C_2 \cdot (k - k_1) > 0;$$

$$C_1 \cdot k_1 k - C_1 k_1^2 - C_2 k + C_2 k_1 > 0;$$

$$C_1 k_1^2 - (C_1 k + C_2) k_1 + C_2 k < 0. \quad (17)$$

Позначимо

$$y_1(k_1) = C_1 k_1^2 - (C_1 k + C_2) k_1 + C_2 k. \quad (18)$$

Коефіцієнт при доданку другого ступеня $C_1 = 5N - 8$ позитивний при будь-якому $N \geq 2$, тобто для будь-якої сітки; тому графік функції $y_1(k_1)$ – парабола, кінці якої спрямовані вгору. Ця функція набуває позитивних значень при великих позитивних і великих (за модулем) негативних значеннях k_1 , тобто нерівність (14) виконується тільки на відріжку $k_1 < k_1 < k_{1_2}$, де k_{1_1}, k_{1_2} – корені квадратного рівняння

$$C_1 k_1^2 - (C_1 k + C_2) k_1 + C_2 k = 0. \quad (19)$$

Дискримінант рівняння (19) дорівнює

$$\begin{aligned} D &= (C_1 k + C_2)^2 - 4 \cdot C_1 \cdot C_2 \cdot k = \\ &= C_1^2 \cdot k^2 + 2 \cdot C_1 \cdot C_2 \cdot k + C_2^2 - 4 \cdot C_1 \cdot C_2 \cdot k = \\ &= C_1^2 \cdot k^2 - 2 \cdot C_1 \cdot C_2 \cdot k + C_2^2 = \\ &= (C_2 - C_1 \cdot k)^2. \end{aligned} \quad (20)$$

Корені рівняння (19):

$$\begin{aligned} k_{1_1} &= \frac{(C_1 k + C_2) - \sqrt{(C_2 - C_1 \cdot k)^2}}{2 \cdot C_1} = \\ &= \frac{C_2 + C_1 \cdot k - |C_2 - C_1 \cdot k|}{2 \cdot C_1}; \end{aligned} \quad (21)$$

$$\begin{aligned} k_{1_2} &= \frac{(C_1 k + C_2) + \sqrt{(C_2 - C_1 \cdot k)^2}}{2 \cdot C_1} = \\ &= \frac{C_2 + C_1 \cdot k + |C_2 - C_1 \cdot k|}{2 \cdot C_1}. \end{aligned} \quad (22)$$

Перший випадок:

$$C_2 - C_1 \cdot k = (5M - 6) - (5N - 8) \cdot k > 0 ; \quad (23)$$

$$|C_2 - C_1 \cdot k| = C_2 - C_1 \cdot k.$$

Розділимо обидві частини нерівності (23) на $C_1 = 5N - 8$. Знак нерівності при цьому не змінюється, оскільки, як уже було вказано, $5N - 8 > 0$ при будь-якому $N \geq 2$, тобто для будь-якої сітки.

$$\frac{5M - 6}{5N - 8} - k > 0; \quad k < \frac{5M - 6}{5N - 8}; \quad k < \frac{C_2}{C_1}. \quad (24)$$

(21) та (22) дають:

$$\begin{aligned} k_{1_1} &= \frac{(C_1 k + C_2) - \sqrt{(C_2 - C_1 \cdot k)^2}}{2 \cdot C_1} = \\ &= \frac{C_2 + C_1 \cdot k - C_2 + C_1 \cdot k}{2 \cdot C_1} = \frac{2 \cdot C_1 \cdot k}{2 \cdot C_1} = k; \\ k_{1_2} &= \frac{(C_1 k + C_2) + \sqrt{(C_2 - C_1 \cdot k)^2}}{2 \cdot C_1} = \\ &= \frac{C_2 + C_1 \cdot k + C_2 - C_1 \cdot k}{2 \cdot C_1} = \frac{2 \cdot C_2}{2 \cdot C_1} = \frac{C_2}{C_1}. \end{aligned}$$

Отже, умова (17) могла б бути виконана при таких k_1 , що

$$k < k_1 < \frac{C_2}{C_1}. \quad (25)$$

Однак за постановкою задачі, що розглядається, $k > k_1$, тому перша нерівність з (25) не може бути виконана.

Другий випадок:

$$C_2 - C_1 \cdot k = 0 .$$

Дискримінант (20) дорівнює нулю, рівняння (19) має кратний розв'язок:

$$k_{1_1} = \frac{(C_1 k + C_2) - 0}{2 \cdot C_1} ; k_{1_2} = \frac{(C_1 k + C_2) + 0}{2 \cdot C_1} ;$$

$$k_{1_1} = k_{1_2} = \frac{k}{2} + \frac{C_2}{2C_1} . \quad (26)$$

За постановкою задачі $k_1 < k$, тому величини k_{1_1} і k_{1_2} , як такі, що задають діапазон значень величини k_1 , повинні бути менші, ніж k . Застосовуючи цю умову до (26), отримаємо нерівність $\frac{k}{2} + \frac{C_2}{2C_1} < k$, звідки отримуємо, що такий випадок може мати місце при

$$\frac{C_2}{2C_1} < \frac{k}{2} ; \frac{C_2}{C_1} < k .$$

При цьому нерівність (19) не має розв'язків, оскільки графік функції (18) має лише точку дотику до осі абсцис $k_1 = 0$, але не пересікає цю вісь.

Третій випадок. (Знак нерівності не змінюється, аналогічно до випадку переходу (23) – (24)):

$$C_2 - C_1 \cdot k = (5M - 6) - (5N - 8)k < 0 ;$$

$$k > \frac{C_2}{C_1} ; \quad (27)$$

$$|C_2 - C_1 \cdot k| = C_1 \cdot k - C_2 ;$$

$$k_{1_1} = \frac{(C_1 k + C_2) - \sqrt{(C_2 - C_1 \cdot k)^2}}{2 \cdot C_1} =$$

$$= \frac{C_2 + C_1 \cdot k - C_1 \cdot k + C_2}{2 \cdot C_1} = \frac{C_2}{C_1} ;$$

$$k_{1_2} = \frac{(C_1 k + C_2) + \sqrt{(C_2 - C_1 \cdot k)^2}}{2 \cdot C_1} =$$

$$= \frac{C_2 + C_1 \cdot k + C_1 \cdot k - C_2}{2 \cdot C_1} = k .$$

Отримуємо, що умову (14) може бути виконано при

$$\frac{C_2}{C_1} < k_1 < k . \quad (28)$$

Таке k_1 може існувати, якщо $\frac{C_2}{C_1} < k$, тобто за умови (27). Ця нерівність є умовою того, що змішане розбиття дасть кращі результати

Тепер порівняємо (12) та (13) і з'ясуємо умови, за яких розбиття горизонтальними границями виявиться гіршим, ніж змішане:

$$Q_b > Q_m ;$$

$$(5M - 6) \cdot (k - 1) >$$

$$> (5N - 8) \cdot (k_1 - 1) + (5M - 6) \cdot (k_- - 1) ;$$

$$(5M - 6) \cdot (k - 1) -$$

$$- (5N - 8) \cdot (k_1 - 1) - (5M - 6) \cdot (k_- - 1) > 0 ;$$

$$(5M - 6) \cdot (k - k_-) - (5N - 8) \cdot (k_1 - 1) > 0 ;$$

$$(5M - 6) \cdot (k - k_-) - (5N - 8) \cdot \left(\frac{k}{k_-} - 1\right) > 0 ;$$

Враховуючи, що $k_- > 0$, і застосовуючи (16), отримуємо:

$$(5M - 6) \cdot (kk_- - (k_-)^2) -$$

$$(5N - 8) \cdot (k - k_-) > 0 ;$$

$$C_2 k k_- - C_2 (k_-)^2 - C_1 k + C_1 k_- > 0 ;$$

$$C_2 (k_-)^2 - (C_2 k + C_1) k_- + C_1 k < 0 . \quad (29)$$

Нерівність (29) може бути отримана з нерівності (17) шляхом заміни коефіцієнтів: C_1 на C_2 , а C_2 на C_1 . Застосовуючи цю ж заміну до співвідношення (28), що задає умови існування розв'язку нерівності (17), отримуємо умову існування розв'язку нерівності (29):

змішаний метод розбиття дасть менше обмінів, ніж «горизонтальний», при умові

$$\frac{C_1}{C_2} < k_- < k_+$$

Висновки

В статті запропоновано багатоточковий метод чисельного розв'язання гідродинамічної задачі та розглянуто задачу оптимізації розподілу обчислювального навантаження між вузлами розподіленого обчислювального середовища при реалізації цього методу. Використання запропонованого методу дозволяє обійти локальні мінімуми цільової функції та оптимізувати використання обчислювальних ресурсів.

Можливим напрямком подальших досліджень може бути: деталізація запропонованої математичної моделі з урахуванням можливих варіантів організації обміну даними між вузлами, неоднорідності обчислювальних ресурсів вузлів та каналів зв'язку, а також властивостей каналів обміну даними; розробка алгоритмів реконфігурації множини векторів зсуву (6) з адаптацією до особливостей цільової функції.

Список літератури

1. Головченко Е.Н. Вычислительный эксперимент с алгоритмами параллельной декомпозиции больших сеток /Е.Н. Головченко, Е.Ю. Дорофеева, В.А. Гасилов, М.В. Якововский //Препринты ИПМ им. Келдыша. – М.: ИПМ РАН им. Келдыша. – 2013. – №7. – 32 с.

2. Волков К.Н. Балансировка нагрузки процессоров при решении краевых задач механики жидкости и газа сеточными методами /К.Н.Волков. //Вычислительные методы и программирование 2012. – Т.13. – С. 108-109.

3. Бутюгин Д.С. Методы параллельного решение СЛАУ на системах с распределенной памятью в библиотеке Krylov /Д.С. Бутюгин, В.П. Ильин, Д.В. Перевозкин. //Вестник ЮурГУ. Серия «Вычислительная математика и информатика». – 2012. – №47 (306). – С. 22-36.

4. Бутюгин Д.С. Алгоритмы решения СЛАУ на системах с распределенной памятью в применении к задачам электромагнетизма /Д.С. Бутюгин. //Вестник ЮУрГУ. Серия «Вычислительная математика и информатика». – 2012. – №46 (305). – С. 5-18.

5. Walshaw C. Parallel dynamic load balancing for adaptive unstructured meshes /C.Walshaw, M.Cross, M.Everett. //J. of Parallel and Distributed Computing. – 1997. – Vol. 47, No 2. – Pp. 102–108.

6. Долгой В.Е. Параллельная реализация итерационных методов решения систем линейных алгебраических уравнений /В.Е. Долгой. //Известия ЮФУ. Технич. науки. – 2011. – Вып. № 1, том 114. – С. 136-139.

7. Simon H.D. Partitioning of unstructured problems for parallel processing /H.D.Simon. //Computing Systems in Engineering. – 1991. – Vol. 2, No. 2, 3. – Pp. 135–148.

8. Hendrickson В. Dynamic load balancing in computational mechanics /B.Hendrickson, K.Devine. //Computer Methods in Applied Mechanics and Engineering. – 2000. – Vol. 184, No. 2–4. – Pp. 485–500.

9. Якововский М.В. Введение в параллельные методы решения задач /М.В. Якововский. – М.: МГУ, 2013. – 328 с.

10. Воропинов А.А. Декомпозиция данных для распараллеливания методики ТИМ-2D и критерии оценки ее качества /А.А. Воропинов. //Вестник ЮурГУ. Серия «Математическое моделирование и программирование». – 2009. – №37 (170). – С. 40-50.

11. Беликов Д.А. Распараллеливание пространственной модели переноса примеси на системах с распределенной памятью с помощью одно- и двумерной декомпозиции области /Д.А. Беликов, А.В. Старченко. //Вестник ЮурГУ. Серия «Математическое моделирование и программирование». – 2009. – Вып. 3, № 17 (150). – С. 12-23.

12. Глазок О.М. Математичний метод розв'язання задач обчислювальної гідродинаміки /О.М. Глазок. // Наукоємні технології. – 2014. – №2. – С. 168-171.

