Журнал обчислювальної та прикладної математики | 2016, №1(121) | Journal of Computational & Applied Mathematics

УДК 519.6

MSC 2010.65M22

THE EFFICIENCY OF DS-ALGORITHM FOR THE SYSTEM OF NAVIER-STOKES EQUATIONS ON MULTIPROCESSOR COMPLEXES

Olekcander Gryshchenko, Anna Martsafei, Viacheslav Onotskyi[1], Olexander Popov[2]

[1]Faculty of Cybernetics, Taras Shevchenko National University of Kyiv, Kyiv, Ukraine, E-mail: Ogryshchenko@ukr.net,anna_shine@i.ua, vingar@ukr.net.

[2]Glushkov Institute of Cybernetics of NASU, Kyiv, Ukraine, E-mail: alex50popov@gmail.com.

ЕФЕКТИВНІСТЬ ЗАСТОСУВАННЯ ДС-АЛГОРИТМУ ДЛЯ СИСТЕМИ РІВНЯНЬ НАВ'Є-СТОКСА НА БАГАТОПРОЦЕСОРНИХ КОМПЛЕКСАХ

О. Ю. Грищенко[1], А. С. Марцафей[1], В. В. Оноцький[1], О. В. Попов[2]

[1]Факультет кібернетики, Київський національний університет імені Тараса Шевченка, Київ, Україна, E-mail: Ogryshchenko@ukr.net,anna_shine@i.ua, vingar@ukr.net.

[2]Інститут кібернетики імені Глушкова НАНУ, Київ, Україна, E-mail: alex50popov@gmail.com.

Abstract. This work is devoted to the analysis of results of realization of numerical DS-algorithm [1] on multiprocessor systems, and highlighting new opportunities that this algorithm opens, compared to other methods. The analysis conducted by the example of initial value problem for the system of Navier-Stokes. The scheme of the proposed approach, tested and the analysis results.

Keywords: system of Navier-Stokes equations, parallel computing.

Резюме. Дану роботу присвячено аналізу результатів реалізації чисельного ДС-алгоритму [1] на багатопроцесорних системах та висвітленню нових можливостей, які цей алгоритм відкриває, в порівнянні з іншими методами. Аналіз проведено на прикладі початково-крайової задачі для системи рівнянь Нав'є-Стокса. Розроблено схему реалізації підходу, проведено тестування і зроблено аналіз результатів.

Ключові слова: системи рівнянь Нав'є-Стокса, паралельні обчислення.

Вступ

Розробка чисельних методів для паралельних комп'ютерних комплексів інтенсивно розпочалася ще в 70-х роках минулого сторіччя. Достатньо повний огляд робіт того часу зроблено в [2]. У [3, 4] запропоновано нові методи розпаралелювання при побудові розв'язків сіткових рівнянь, в [5] розглянуто можливість використовувати метод скінченних елементів при розв'язуванні стаціонарної системи рівнянь Нав'є-Стокса на обчислювальному комплексі CRAY-90. Паралельну стратегію, запропоновану в [6], використовували спільно з декомпозицією області для побудови полів потоків.

Досвід використання обчислювальних алгоритмів показує, що досить часто навіть високоефективні на послідовних обчислювальних системах чисельні методи виявляються мало або зовсім не ефективними на багатопроцесорних [2]. У роботі [1] запропоновано ДС-алгоритм для розв'язування початково-крайових задач для систем Нав'є-Стокса, який добре зарекомендував себе при одно поточному обчисленні. У даній роботі розроблено методику реалізації цього алгоритму на багатопроцесорних системах. Основну концепцію і дослідження алгоритму висвітлено у роботах [1, 7]. Метою цієї роботи є метод реалізації і аналіз практичної ефективності використання даного ДС-алгоритму на багатопроцесорних комплексах.

Як відомо [2, 8], основною характеристикою ефективності паралельних процесів $E_p = \frac{S_p}{p}$ є його прискорення по відношенню до однопроцесорного режиму $S_p = \frac{T_{seq}}{T_{p\ par}}$, де T_{seq} — час виконання алгоритму на послідовному комп'ютері, $T_{p\ par}$ — час виконання на p паралельних. Це, зокрема, досягається рівномірністю і ступенем завантаженості процесорів та мінімізацією частоти комунікаційних обмінів. Ефективність алгоритму також визначають чотири внутрішні концепції:

1. можливість розпаралелення системи рівнянь на рівні рівнянь,
2. розпаралелння системи за просторовими змінніми,
3. розпаралелення системи за фізичними процесами,
4. розпаралелення системи за допомогою розщеплення даних.

У запропонованому алгоритмі реалізовано першу та останню концепції.

1. Об'єкт дослідження

Дослідження проводиться на прикладі математичної моделі, яка в області $G = \{\Omega \times (0 < t < T)\}$, $\Omega = \{(x, y) : 0 < x < a,\ 0 < y < b\}$ відображає динаміку руху в'язкої нестислої рідини і при заданих початкових та граничних умовах описується системою рівнянь Нав'є-Стокса

$$\frac{\partial u}{\partial t} = -\frac{1}{\rho}\frac{\partial P}{\partial x} + v\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) - \frac{\partial (u)^2}{\partial x} - \frac{\partial\,(uv)}{\partial y} + \psi_1\,(x, y, t)\,; \qquad (1)$$

$$\frac{\partial v}{\partial t} = -\frac{1}{\rho}\frac{\partial P}{\partial y} + v\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) - \frac{\partial\,(uv)}{\partial x} - \frac{\partial (v)^2}{\partial y} + \psi_2\,(x, y, t)\,; \qquad (2)$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \qquad (3)$$

де $u,\ v$ — проекції вектора швидкості, P — тиск, ρ — щільність, v — в'язкість. Для знаходження розв'язку цієї задачі, тобто, значень u, v, P, у роботах [1, 7] запропоновано адаптацію ДС-алгоритму до цієї системи, записаної як у дивергентній (рівняння(1)–(3)), так і недивергентній формі подання [7]. Крім того, у роботі [7] обґрунтовано консервативність алгоритму, дисипативність та сумарну апроксимацію $O\left(\tau^2 + h^2\right)$. Далі для аналізу можливостей розпаралелювання даного алгоритму використано різницеву схему для недивергентного вигляду у відповідності до роботи [1].

За загальною методикою ДС-алгоритму введену на Ω сіткову область $\omega_{\tau h} = \{x_i, y_j, t_n : \ x_i = ih_x,\ y_j = jh_y,\ t_n = n\tau,\ \tau > 0\}$ розділено на дві допоміжні $\Omega_{\tau h}^{(1,n)}$ й $\Omega_{\tau h}^{(2,n)}$. До першої віднесемо точки, для яких сума індексів $s = i + j + n$ — парна, а до другої — решту. Через $\partial\omega_{\tau h}$ позначимо сіткову границю $\omega_{\tau h}$. Значення, обчислені на непарних кроках $(2n + 1)$, сприймаються як допоміжні (предиктор). Розв'язком задачі вважаємо значення, обчислені на парних кроках $(2n + 2)$ (коректор). Це реалізує перехід з часового шару $2n\tau$ на шар $(2n + 2)\,\tau$. Для побудови чисельного розв'язку задачі, як і в [1,7], від системи, записаної у вигляді системи (1)–(3), переходимо до системи, в якій рівняння (3) замінено на рівняння Пуассона для тиску [9]

$$\frac{\partial^2 P}{\partial x^2} + \frac{\partial^2 P}{\partial y^2} = f\left(u, v, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial v}{\partial x}, \frac{\partial v}{\partial y}\right). \tag{4}$$

Параболічність рівнянь (1), (2) та еліптичність (4) дозволяє знаходити розв'язки P та u, v на різних часових кроках: P — на непарних $(2n + 1)$, а u та v — на парних $(2n + 2)$.

2. Алгоритм розв'язування

Алгоритм побудови розв'язку складається з семи блоків.

Блок 1. Введення даних.
Блок 2. Введення початкових та граничних даних.
Блок 3. Ітераційне обчислення значень тиску P на кроці $(2n + 1)$, $n = 0,\ 1\ldots$.
Блок 4. Обчислення значень u та v на $(2n + 1)$ кроці для сіток $\Omega_{\tau h}^{(1,2n+1)}$ та $\Omega_{\tau h}^{(2,2n+1)}$ (предиктор).
Блок 5. Обчислення значень u та v на $(2n + 2)$ кроці для $\Omega_{\tau h}^{(1,2n+2)}$ та $\Omega_{\tau h}^{(2,2n+2)}$ (розв'язок).
Блок 6. Перевірка умов закінчення розрахунку.
Блок 7. Виведення отриманих даних.

В кожному з цих блоків програмно реалізуємо вказані функції.

Блок 1. Вводимо розміри області a, b; кількість вузлів сітки по $x - m_x$ та по $y - m_y$; кількість процесорів p та задаємо крок сітки $h_x = \frac{a}{m_x}$, $h_y = \frac{b}{m_y}$, τ. Обраховуємо розміри масивів $M = m_x \times m_y$ для запису початкових значень P, u, v.

Блок 2. Вводимо початкові u^0, v^0,P^0 та граничні дані функцій P, u, v.

Блок 3. Ітераційно визначаємо тиск P на кроці $(2n + 1)$. Різницеве рівняння (4) розв'язуємо, використовуючи ітераційний процес, запропонований в [10]. Початкове наближення функції P на першому кроці $(n = 0)$

задаємо довільно, але узгоджено із заданими граничними умовами. Покладаючи s номером ітерації, а τ_1 — ітераційним параметром, обчислення тиску для кожного часового кроку $(2n+1)\,\tau$ проводимо за формулами:

на границі $P^{2s+1}\big|_{\partial\omega_{\tau h}} = P\,(x, y, (2n+1)\,\tau)|_{\partial\omega_{\tau h}}$,

на множині $\Omega_{\tau h}^{(1,2s+1)}$

$$\begin{aligned} P_{ij}^{2s+1} = P_{ij}^{2s}\left(1 - 2\tau_1\left(\tfrac{1}{h_x^2} + \tfrac{1}{h_y^2}\right)\right) + \tfrac{\tau}{h_x^2}\left(P_{i+1j}^{2s} + P_{i-1j}^{2s}\right) + \\ + \tfrac{\tau}{h_y^2}\left(P_{ij+1}^{2s} + P_{ij-1}^{2s}\right) - \tau_1\varphi_{ij}, \end{aligned} \tag{5}$$

для $\Omega_{\tau h}^{(2,2s+1)}$

$$\begin{aligned} P_{ij}^{2s+1} = \left[1\Big/\!\left(1 + 2\tau_1\left(\tfrac{1}{h_x^2} + \tfrac{1}{h_y^2}\right)\right)\right] \times \\ \times\left[P_{ij}^{2s} + \tfrac{\tau}{h_x^2}\left(P_{i+1j}^{2s+1} + P_{i-1j}^{2s+1}\right) + \tfrac{\tau}{h_y^2}\left(P_{ij+1}^{2s+1} + P_{ij-1}^{2s+1}\right) - \tau_1\varphi_{ij}\right]. \end{aligned} \tag{6}$$

Далі обчислюємо граничні умови $P^{2s+2}\big|_{\partial\omega_{\tau h}} = P\,(x, y, (2n+2)\,\tau)|_{\partial\omega_{\tau h}}$ і для $\Omega_{\tau h}^{(1,2s+2)}$

$$\begin{aligned} P_{ij}^{2s+2} = P_{ij}^{2s+1}\left(1 - 2\tau_1\left(\tfrac{1}{h_x^2} + \tfrac{1}{h_y^2}\right)\right) + \tfrac{\tau}{h_x^2}\left(P_{i+1j}^{2s+1} + P_{i-1j}^{2s+1}\right) + \\ + \tfrac{\tau}{h_y^2}\left(P_{ij+1}^{2s+1} + P_{ij-1}^{2s+1}\right) - \tau_1\varphi_{ij}, \end{aligned} \tag{7}$$

а для $\Omega_{\tau h}^{(2,2s+2)}$

$$\begin{aligned} P_{ij}^{2s+2} = \left[1\Big/\!\left(1 + 2\tau_1\left(\tfrac{1}{h_x^2} + \tfrac{1}{h_y^2}\right)\right)\right] \times \\ \times\left[P_{ij}^{2s+1} + \tfrac{\tau}{h_x^2}\left(P_{i+1j}^{2s+2} + P_{i-1j}^{2s+2}\right) + + \tfrac{\tau}{h_y^2}\left(P_{ij+1}^{2s+2} + P_{ij-1}^{2s+2}\right) - \tau_1\varphi_{ij}\right]. \end{aligned} \tag{8}$$

Якщо виконується умова $\left\|P^{2s+2} - P^{2s+1}\right\| \geq \varepsilon$, збільшуємо значення s і повторюємо обчислення за формулами (5)–(8). У формулах (7), (8) φ_{ij} обчислюється за формулою

$$\begin{gathered} \varphi_{ij} = 2\frac{u_{i+1j}^{2n} - u_{i-1j}^{2n}}{2h_x}\frac{v_{ij+1}^{2n} - v_{ij-1}^{2n}}{2h_y} - 2\frac{u_{ij+1}^{2n} - u_{ij-1}^{2n}}{2h_y}\frac{v_{i+1j}^{2n} - v_{i-1j}^{2n}}{2h_x} + \\ + {\psi'}_{1x}^{2n+1} + {\psi'}_{2y}^{2n+1} \;\; \forall\, s \in N. \end{gathered}$$

Блок 4. Значення u та v на $(2n+1)$ кроці у відповідності до [1] обчислюємо в точках для $\Omega_{\tau h}^{(1,2n+1)}$

$$u_{ij}^{2n+1} = \frac{a_{12}^{2n}\,(ij)\,F_2^{2n}\,(ij) - a_{22}^{2n}\,(ij)\,F_1^{2n}\,(ij)}{a_{21}^{2n}\,(ij)\,a_{12}^{2n}\,(ij) - a_{11}^{2n}\,(ij)\,a_{22}^{2n}\,(ij)}, \tag{9}$$

$$v_{ij}^{2n+1} = \frac{a_{21}^{2n}\,(ij)\,F_1^{2n}\,(ij) - a_{11}^{2n}\,(ij)\,F_2^{2n}\,(ij)}{a_{21}^{2n}\,(ij)\,a_{12}^{2n}\,(ij) - a_{11}^{2n}\,(ij)\,a_{22}^{2n}\,(ij)}, \tag{10}$$

де

$$a_{11}^{2n}\,(ij) = \left(1 + \frac{\tau}{2h_x}\left(u_{i+1j}^{2n} - u_{i-1j}^{2n}\right)\right),\; a_{12}^{2n}\,(ij) = \frac{\tau}{2h_y}\left(u_{ij+1}^{2n} - u_{ij-1}^{2n}\right),$$

$$a_{21}^{2n}\,(ij) = \frac{\tau}{2h_x}\left(v_{i+1j}^{2n} - v_{i-1j}^{2n}\right),\; a_{22}^{2n}\,(ij) = 1 + \frac{\tau}{2h_y}\left(v_{ij+1}^{2n} - v_{ij-1}^{2n}\right),$$

$$F_1^{2n}(ij) = u_{ij}^{2n} + \tau\left(Lu_{ij}^{2n} - \frac{1}{\rho}\frac{P_{i+1j}^{2n+1} - P_{i-1j}^{2n+1}}{2h_x}\right),$$

$$F_2^{2n}(ij) = v_{ij}^{2n} + \tau\left(Lv_{ij}^{2n} - \frac{1}{\rho}\frac{P_{ij+1}^{2n+1} - P_{ij-1}^{2n+1}}{2h_y}\right)$$

при $Lu_{ij}^{2n} = \nu_{ij}^{2n+1}\left(\frac{u_{i-1j}^{2n} - u_{ij}^{2n} + u_{i+1j}^{2n}}{h_x^2} + \frac{u_{ij-1}^{2n} - u_{ij}^{2n} + u_{ij+1}^{2n}}{h_y^2}\right)$.

А в точках $\Omega_{\tau h}^{(2,2n+1)}$

$$u_{ij}^{2n+1} = \frac{b_{12}^{2n+1}(ij)\,\Phi_2^{2n+1}(ij) - b_{22}^{2n+1}(ij)\,\Phi_1^{2n+1}(ij)}{b_{21}^{2n+1}(ij)\,b_{12}^{2n+1}(ij) - b_{11}^{2n+1}(ij)\,b_{22}^{2n+1}(ij)}, \tag{11}$$

$$v_{ij}^{2n+1} = \frac{b_{21}^{2n+1}(ij)\,\Phi_1^{2n+1}(ij) - b_{11}^{2n+1}(ij)\,\Phi_2^{2n+1}(ij)}{b_{21}^{2n+1}(ij)\,b_{12}^{2n+1}(ij) - b_{11}^{2n+1}(ij)\,b_{22}^{2n+1}(ij)}, \tag{12}$$

де $b_{11}^{2n+1} = 1 + \frac{\tau}{2h_x}\left(u_{i+1j}^{2n+1} - u_{i-1j}^{2n+1}\right) + 2\tau v_{ij}^{2n+1}\left(\frac{1}{h_x^2} + \frac{1}{h_y^2}\right)$,
$b_{12}^{2n+1} = \frac{\tau}{2h_y}\left(u_{ij+1}^{2n+1} - u_{ij-1}^{2n+1}\right)$, $b_{21}^{2n+1} = \frac{\tau}{2h_x}\left(v_{i+1j}^{2n+1} - v_{i-1j}^{2n+1}\right)$,
$b_{22}^{2n+1} = 1 + \frac{\tau}{2h_y}\left(v_{ij+1}^{2n+1} - v_{ij-1}^{2n+1}\right) + 2\tau v_{ij}^{2n+1}\left(\frac{1}{h_x^2} + \frac{1}{h_y^2}\right)$,
$\Phi_1^{2n+1} = u_{ij}^{2n} + \tau v_{ij}^{2n+1}\left(\frac{u_{i-1j}^{2n+1} + u_{i+1j}^{2n+1}}{h_x^2} + \frac{u_{ij-1}^{2n+1} + u_{ij+1}^{2n+1}}{h_y^2}\right) - \frac{\tau}{\rho}\frac{P_{i+1j}^{2n+1} - P_{i-1j}^{2n+1}}{2h_x}$,
$\Phi_2^{2n+1} = v_{ij}^{2n} + \tau v_{ij}^{2n+1}\left(\frac{v_{i-1j}^{2n+1} + v_{i+1j}^{2n+1}}{h_x^2} + \frac{v_{ij-1}^{2n+1} + v_{ij+1}^{2n+1}}{h_y^2}\right) - \frac{\tau}{\rho}\frac{P_{ij+1}^{2n+1} - P_{ij-1}^{2n+1}}{2h_y}$.

Блок 5. Обчислюємо значення u та v на $(2n+2)$ кроці для $\Omega_{\tau h}^{(1,2n+2)}$

$$\begin{aligned} u_{ij}^{2n+2} = {} & u_{ij}^{2n+1}\left(1 - \frac{\tau}{2h_x}\left(u_{i+1j}^{2n+1} - u_{i-1j}^{2n+1}\right) - 2\tau\nu_{ij}^{2n+1}\left(\frac{1}{h_x^2} + \frac{1}{h_y^2}\right)\right) + \\ & + u_{ij-1}^{2n+1}\left(\frac{\tau v_{ij}^{2n+1}}{2h_y} + \frac{\tau\nu_{ij}^{2n+1}}{h_y^2}\right) + u_{ij+1}^{2n+1}\left(-\frac{\tau v_{ij}^{2n+1}}{2h_y} + \frac{\tau\nu_{ij}^{2n+1}}{h_y^2}\right) + \\ & + \frac{\tau\nu_{ij}^{2n+1}}{h_x^2}\left(u_{i-1j}^{2n+1} + u_{i+1j}^{2n+1}\right) - \frac{\tau}{\rho}\frac{P_{i+1j}^{2n+1} - P_{i-1j}^{2n+1}}{2h_x}, \end{aligned} \tag{13}$$

$$\begin{aligned} v_{ij}^{2n+2} = {} & v_{ij}^{2n+1}\left(1 - \frac{\tau}{2h_y}\left(v_{ij+1}^{2n+1} - v_{ij-1}^{2n+1}\right) - 2\tau\nu_{ij}^{2n+1}\left(\frac{1}{h_x^2} + \frac{1}{h_y^2}\right)\right) + \\ & + v_{i-1j}^{2n+1}\left(\frac{\tau u_{ij}^{2n+1}}{2h_x} + \frac{\tau\nu_{ij}^{2n+1}}{h_x^2}\right) + v_{i+1j}^{2n+1}\left(-\frac{\tau u_{ij}^{2n+1}}{2h_x} + \frac{\tau\nu_{ij}^{2n+1}}{h_x^2}\right) + \\ & + \frac{\tau\nu_{ij}^{2n+1}}{h_y^2}\left(v_{ij-1}^{2n+1} + v_{ij+1}^{2n+1}\right) - \frac{\tau}{\rho}\frac{P_{ij+1}^{2n+1} - P_{ij-1}^{2n+1}}{2h_y}, \end{aligned} \tag{14}$$

а для $\Omega_{\tau h}^{(2,2n+2)}$

$$\begin{aligned} u_{ij}^{2n+2} = {} & \left[1\Big/\left(1 + \frac{2\tau\nu_{ij}^{2n+1}}{h_x^2} + \frac{2\tau\nu_{ij}^{2n+1}}{h_y^2}\right)\right]\left[u_{ij}^{2n+1}\left(1 - \frac{\tau}{2h_x}\left(u_{i+1j}^{2n+2} - u_{i-1j}^{2n+2}\right)\right) - \right. \\ & - \frac{\tau\nu_{ij}^{2n+1}}{2h_y}\left(u_{ij+1}^{2n+2} - u_{ij-1}^{2n+2}\right) + \frac{\tau\nu_{ij}^{2n+1}}{h_x^2}\left(u_{i-1j}^{2n+2} + u_{i+1j}^{2n+2}\right) + \\ & \left. + \frac{\tau\nu_{ij}^{2n+1}}{2h_y^2}\left(u_{ij-1}^{2n+2} + u_{ij+1}^{2n+2}\right) - \frac{\tau}{\rho}\frac{P_{i+1j}^{2n+1} - P_{i-1j}^{2n+1}}{2h_x}\right], \end{aligned} \tag{15}$$

$$v_{ij}^{2n+2}=\left[1/\left(1+\frac{2\tau\nu_{ij}^{2n+1}}{h_x^2}+\frac{2\tau\nu_{ij}^{2n+1}}{h_y^2}\right)\right]\left[v_{ij}^{2n+1}\left(1-\frac{\tau}{2h_y}\left(v_{ij+1}^{2n+2}-v_{ij-1}^{2n+2}\right)\right)-\right.$$
$$-\frac{\tau u_{ij}^{2n+1}}{2h_x}\left(v_{i+1j}^{2n+2}-v_{i-1j}^{2n+2}\right)+\frac{\tau\nu_{ij}^{2n+1}}{h_x^2}\left(v_{i-1j}^{2n+2}+v_{i+1j}^{2n+2}\right)+$$
$$\left.+\frac{\tau\nu_{ij}^{2n+1}}{2h_y^2}\left(v_{ij-1}^{2n+2}+v_{ij+1}^{2n+2}\right)-\frac{\tau}{\rho}\frac{P_{ij+1}^{2n+1}-P_{ij-1}^{2n+1}}{2h_y}\right]. \tag{16}$$

Блок 6 перевіряє умови припинення процесу.
Блок 7 у розрахунках не приймає участі.

3. Реалізація алгоритму на багатопроцесорних комплексах

В апаратній реалізації використовуємо MIMD-архітектуру системи. Така структура забезпечує передачу багатьох потоків команд та багатьох потоків даних так, що кожен процесор працює під керівництвом своїх власних програм, обробляючи свої масиви даних. У нашому випадку блок 1 та блок 2 алгоритму є підготовчими для блоків 3–5. Потоки початкових або вже знайдених на попередньому кроці значень функцій тиску P та проекцій вектора швидкості u та v обробляються командами за формулами (5)–(8) та (9)–(16) відповідно до блоків 3–5, причому кожна з функцій обчислюється незалежно одна від одної. Це дає можливість використовувати першу з концепцій розпаралелювання системи на рівні рівнянь. Поглибити процес розпаралелення вдається, використовуючи при обчисленні кожного з блоків окремі групи з p процесорів. При обчисленні функції тиску P увесь масив даних, який складається з m_x стовпчиків, кожен з яких має m_y елементів, розбиваємо на окремі потоки так, що перший містить стовпчики від 1 до (m_1+1), другий — від m_1 до (m_2+1), а останній — від m_{p-1} до m_x. Вказана в блоці 3 послідовність обчислень дозволяє кожному з p процесорів виконувати свою роботу незалежно від інших. Аналогічно реалізовано процеси розпаралелювання блоків 4 та 5. Відзначимо, що при даному підході запропонований алгоритм не потребує зшивання розв'язків на границях потоків даних. Потрібно лише додатково обмінюватись між сусідніми процесорами приграничними стовпчиками. Це мінімізує кількість комунікаційних зв'язків. Оскільки затрати на розрахунок одного елемента стовпчика для кожного процесора рівні, то бажано, щоб кількість елементів у в кожному з масивів, що припадає на один процесор, була однакова. Тобто, щоб $m_k-m_{k-1}=k=\text{const}\ \forall\ k=\overline{1,p}$ та $kp=m_x$.

4. Тестування алгоритму

При тестуванні алгоритму покладаємо

$$\psi_1(x,y,t)=0,\quad \psi_2(x,y,t)=\alpha e^{2-2t}e^{2\alpha y},\quad \rho=1.$$

Тоді точний розв'язок задачі (1), (2), (4) має вигляд:

$$u^*(x,y,t)=-e^{1-t}e^{\alpha y}\sin(\alpha x),$$
$$v^*(x,y,t)=e^{1-t}e^{\alpha y}\cos(\alpha x),$$
$$P^*(x,y,t)=\frac{\rho}{\alpha}\left(e^{1-t}e^{\alpha y}\cos(\alpha x)+e^{1+\alpha}+1\right).$$

Приймемо $\alpha = \pi/8$, $a = b = 1/8$, $t > 0$, ε — точність, τ_1 — параметр ітераційного процесу.

Порівняння одержаних результатів з точними значеннями розв'язку показали, що при $\varepsilon = 10^{-3}$, $\tau_1 = 10^{-3}$ на 1600 часових кроках максимальна абсолютна похибка $(u, v, P) = (0.000405,\ 0.000437,\ 0.01)$, а максимальна відносна похибка (%) становить відповідно $(u, v, P) = (0.0069,\ 0.00496,\ 0.0254)$. Доцільно відмітити високу практичну швидкість збіжності ітераційного алгоритму [10]. У процесі обчислення встановлено, що на кожному кроці, починаючи з другого, алгоритм збігається з заданою точністю $\varepsilon = 10^{-3}$ переважно за одну ітерацію (2 кроки ДС-алгоритму). Проведено два варіанти тестування. У першому розрахунок проводимо для чотирьох областей, починаючи від квадратної зі стороною a до прямокутної зі збільшенням горизонтальної сторони від двох до восьми раз на одно-, дво-, чотири- та восьмипроцесорних системах. Областю обчислення кожного з процесорів є вертикальні сіткові смуги. При збільшенні області кроки сітки не змінюємо, а збільшуємо кількість точок розбиття по x. Затрачений час на розв'язування кожної з цих задач наведено в таблиці 1.

Таблиця 1. Результати обчислення часу розрахунків

Кількість процесорів\Сітка	40×40	80×40	160×40	320×40
1	1.18	1.68	3.5	7.38
2	0.907569	1.1665	2.00183	3.91243
4	0.698148	0.682298	1.05686	1.90533
8	0.667001	0.509744	0.740436	1.27268

У другому варіанті зміни області залишаються такими ж, але кожному процесору виділяється горизонтальна смуга постійної ширини, але зі змінною довжиною. Результати обчислення часу розрахунків наведено у таблиці 2.

Таблиця 2. Результати обчислення часу розрахунків

Кількість процесорів\Сітка	40×40	80×40	160×40	320×40
1	1.18	1.68	3.5	7.38
2	1.00045	1.07261	1.99539	3.74008
4	0.829162	0.739025	1.26495	2.3099
8	1.06393	1.70293	1.0464	1.51764

Порівняння результатів з таблиці 1 та таблиці 2 показує, що збільшення числа елементів обміну суттєво збільшує час розрахунку. Використовуючи дані таблиці, визначимо прискорення $S_p = \frac{T_{seq}}{T_{p\ par}}$ та ефективність паралельних алгоритмів $E_p = \frac{S_p}{p}$ у кожному з цих випадків.

Таблиця 3. Прискорення для першого варіанту

$S_p \backslash$Сітка	40×40	80×40	160×40	320×40
p=2	1.30	1.44	1.75	1.89
p=4	1.69	2.46	3.31	3.87
p=8	1.77	3.29	4.73	5.80

Таблиця 4. Ефективність для першого варіанту

$E_p \backslash$Сітка	40×40	80×40	160×40	320×40
p=2	0.65	0.72	0.86	0.95
p=4	0.42	0.62	0.83	0.97
p=8	0.22	0.41	0.59	0.73

Таблиця 5. Прискорення для другого варіанту

$S_p \backslash$Сітка	40×40	80×40	160×40	320×40
p=2	1.18	1.57	1.75	1.97
p=4	1.42	2.27	2.77	3.19
p=8	1.11	0.99	3.34	4.86

Таблиця 6. Ефективність для другого варіанту

$E_p \backslash$Сітка	40×40	80×40	160×40	320×40
p=2	0.59	0.79	0.88	0.99
p=4	0.36	0.57	0.69	0.80
p=8	0.14	0.12	0.42	0.61

Як видно з таблиць, прискорення та ефективність алгоритму суттєво залежить від відношення часу чисто обчислювальних процесів системи до часу комунікаційних процесів.

Висновки

Для розв'язування початково-крайових задач динаміки нестислої в'язкої рідини у роботі запропоновано методику розповсюдження ДС-алгоритму на розв'язування систем рівнянь Нав'є-Стокса з використанням обчислювальних комплексів з паралельною архітектурою MIMD на платформі програмування MPI. Проведені обчислення показали високу ефективність запропонованого підходу, а саме, його швидкодію та точність, і підтвердили зроблені раніше теоретичні висновки. Так при розв'язуванні системи рівнянь Нав'є-Стокса на сітковій області порядку 12×10^3 вузлів час, витрачений на розв'язування задачі, при збільшенні числа процесорів від двох до восьми спадає від 3.91 до 1.27 секунди у першому варіанті, тобто, прискорення зростає, але ефективність (навантаження на один процесор) спадає. Середнє значення прискорення на восьмипроцесорній системі становить 5.35, а ефективність відповідно 0.67. При цьому максимальна абсолютна похибка на всьому часовому відрізку не перевищує відповідно $(u, v, P) = (0.000405,\ 0.000437,\ 0.01)$.

Література

1. Грищенко О. Ю., Довбня П. І. Модифікація ДС-алгоритму розв'язування початково-крайових задач для системи рівнянь Нав'є-Стокса // Журн. обчисл. та прикл. матем. — 2011. — № 3(106). — С. 19–26.
2. Воеводин В. В. Математические модели и методы в параллельных процессах. — М.: Наука, 1986. — 296 с.
3. Марчук Г. И., Ильин В. П. Параллельные вычисления в сеточных методах решения задач математической физики. — Новосибирск, 1979. — 23 с. (Препринт/ВЦ СО АН СССР: № 220.)
4. Коновалов А. Н., Бугров А. Н., Елинов В. Д. Алгоритмы распараллеливания сеточных задач. — В кн.: Актуальные проблемы вычислительной и прикладной математики. Новосибирск: Наука, 1983. — С. 95–99.
5. Tony W. H. Sheu, Morten M. T. Wang, and S. F. Tsai. Element-by-Element Parallel Computation of Incompressible Navier–Stokes Equations in Three Dimensions // SIAM Journal on Scientific Computing. — Vol. 21. — No. 4 (1999). — pp. 1387–1400. DOI: 10.1137/S1064827598335416
6. A. Lin. Solving numerically the Navier–Stokes equations on parallel systems // International Journal for Numerical Methods in Fluids. — Vol. 10. — No. 8 (1990). — pp. 907–923. DOI: 10.1002/fld.1650100805
7. Марцафей А. С. Обґрунтування ДС-алгоритмів при моделюванні ізотермічних потоків нестислої рідини // Журн. обчисл. та прикл. матем. — 2014. — № 3(117). — С. 66–75.
8. Химич А. Н. Молчанов И. Н., Попов А. В., Чистякова Т. В., Яковлев М. Ф. Параллельные алгоритмы решения задач вычислительной математики. — К.: Наук. думка, 2008. — 247 с.
9. Роуч П. Вычислительная гидродинамика. — Москва: Мир, 1980. — 618 с.
10. Марцафей А. С. Застосування чисельного ДС-алгоритму для ітераційного моделювання стаціонарних процесів переносу // Журн. обчисл. та прикл. матем. — Київ. — 2009. — № 3(99). — С. 63–69.

Надійшла 03.03.2016