

МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ТА ОБЧИСЛОВАЛЬНІ МЕТОДИ

УДК 519.6:544.431.11:[502.521]

А. П. Власюк¹, В. В. Жуковський², Н. А. Жуковська²

КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ ПРОЦЕСУ ВЕРТИКАЛЬНОЇ МІГРАЦІЇ РАДІОНУКЛІДІВ У КАТАЛІТИЧНОМУ ПОРИСТОМУ СЕРЕДОВИЩІ З ВИКОРИСТАННЯМ ПАРАЛЕЛЬНИХ ОБЧИСЛЕНЬ

¹ Національний університет «Острозька академія», Острог

² Національний університет водного господарства та природокористування, Рівне

Анотація. У статті проведено комп'ютерне моделювання процесу вертикальної міграції забруднюючих речовин у каталітичному пористому середовищі. Математична модель враховує мікро-, мезо- та макромасштаби процесу масоперенесення. Отримано чисельний розв'язок відповідної одновимірної крайової задачі методом скінченних різниць. Програмна реалізація здійснена з використанням паралельних обчислень для систем із спільною пам'яттю. Чисельні експерименти комп'ютерного моделювання виконані за допомогою власноствореного програмного комплексу NanoSurface. Встановлена алгоритмічна складність методів прогонки та час виконання паралельних обчислень. Взято до уваги специфікації компілятора C99.

Ключові слова: комп'ютерне моделювання, паралельні обчислення, OpenMP, крайова задача, масоперенос, наночастинки, очищення.

Аннотация. В статье произведено компьютерное моделирование процесса вертикальной миграции загрязняющих веществ в каталитической пористой среде. Математическая модель учитывает микро-, мезо- и макромасштабах процесса массопереноса. Получено численное решение соответствующей одномерной краевой задачи методом конечных разностей. Программная реализация осуществлена с использованием параллельных вычислений для систем с общей памятью. Численные эксперименты компьютерного моделирования выполнены с помощью программного комплекса NanoSurface. Выведена алгоритмическая сложность методов прогонки и время выполнения параллельных вычислений. Принято во внимание спецификации компилятора C99.

Ключевые слова: компьютерное моделирование, параллельные вычисления, OpenMP, краевая задача, массоперенос, наночастицы, очистка.

Abstract. In the paper the results of an application of computer modeling of vertical migration process of the pollutants in the catalytic porous media with filter traps are presented. The mathematical model takes into account micro and meso/macro scales factor of mass transfer process. It was obtained the numerical solution of respective one dimensional boundary value problem by the method of finite differences. Software implementation was done using multi-platform shared-memory parallel programming. All computer modeling experiments have been made using NanoSurface tool. Introduced algorithmic complexity of the sweep methods and calculated execution time for parallel calculation. C99 compiler specification took into account.

Key words: computer modeling, parallel computing, OpenMP, boundary-value problem, mass transfer, nanoparticle, refinement.

Вступ

Сучасна технологічна діяльність людини призводить до того, що ґрунти стають забрудненими великою кількістю розчинних речовин, в тому числі і радіонуклідами. Сільське господарство в свою чергу використовує широкий асортимент пестицидів, добрив та фумігантів, що є додатковими джерелами забруднень. Враховуючи це, постає питання створення ефективних механізмів управління та прогнозування поширення забруднень в поверхневих шарах ґрунту (кореневій зоні) та запобігання їх міграції в докволишні поверхневі та підземні води. Зокрема, на сьогодні широкий спектр фізико-хімічних і біологічних методів досліджено для видалення органічних і/або неорганічних забруднень (коагуляція-флокуляція, мембранні процеси, адсорбція, тощо). Найбільш ефективним і недорогим процесом для видалення колоїдів та органічних речовин є використання адсорбційних мікро- та наночастинок, що представляє собою каталітичне пористе середовище [1]. Разом з тим, складність натурних експериментів змушує використовувати апарат математичного і комп'ютерного моделювання. Математичні моделі, що враховують низку факторів (наприклад, гетеродифузний перенос радіонуклідів у поверхневих шарах ґрунту [2], в середовищі з пастками [3], процеси десорбції в нанопористому середовищі [4]), дозволяють спрогнозувати концентрацію забруднених речовин протягом часу. А необхідність швидкого обрахунку і можливості оперативного втручання в ситуацію з метою її корекції призводить до використання паралельних обчислень.

Актуальність

Тривалий час спостерігається стрімкий розвиток комп'ютерної та обчислювальної техніки, що завжди підтверджується законом Мура [5]. У зв'язку з цим зростає необхідність оптимального використання всіх обчислювальних потужностей сучасної техніки для проведення чисельних експериментів. Протягом останніх років виробники процесорів віддають перевагу багатоядерній архітектурі замість збільшення тактової частоти та забезпечення паралелізму на рівні інструкцій. Тому для отримання макси-

А. П. Власюк, В. В. Жуковський, Н. А. Жуковська, 2017

мального ефекту від збільшеної продуктивності програмний код повинен бути адаптований під багатоядерне використання. Це досягається завдяки паралельним обчисленням.

На відміну від розподілених обчислень, паралельні обчислення для систем із спільною пам'яттю не набули широкого розповсюдження. Однак даний підхід до обчислень вже знаходить своє місце в роботах вчених, що займаються математичним моделюванням. Зокрема, в роботах [6–8] розглядаються задачі оптимізації обчислювальних процесів за допомогою кластерів при розв'язуванні складних задач математичного моделювання (процес електрокінетичної очистки ґрунту, динаміки ґрунтового масиву, тощо). Програмний комплекс Nadra-3D, що призначений для комп'ютерного моделювання процесів фільтрації, дифузії і зміни напружено-деформованого стану в багатокомпонентних тривимірних об'єктах, які можуть містити тонкі включення, так само успішно використовує паралельні обчислення при розв'язуванні системи алгебраїчних рівнянь методом скінченних елементів [9]. Однак кожний випадок застосування паралельних обчислень потребує індивідуального підходу та аналізу при розв'язуванні задач математичної фізики.

Постановка та мета задачі

Розглянемо задачу вертикальної міграції забруднених речовин (наприклад, радіонуклідів) у шарі ґрунту (рис. 1).

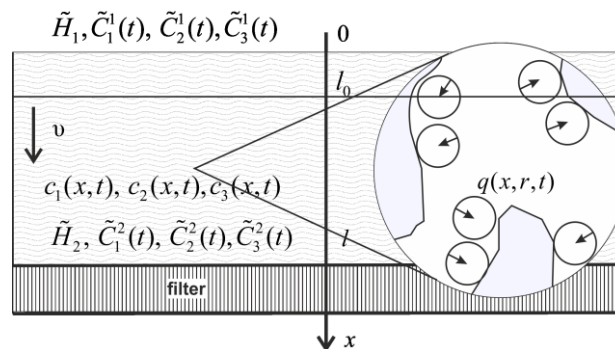


Рисунок 1 – Процес міграції радіонуклідів в одновимірному випадку в каталітичному пористому середовищі

На глибині l в ґрунті розміщений фільтр-вловлювач, наповнений сорбуючим елементом (наприклад, базальтовим туфом). Відомі п'єзометричні напори на верхній та нижній (підстилаючій) поверхнях ґрунту \tilde{H}_1 та \tilde{H}_2 ($\tilde{H}_1 > \tilde{H}_2$) відповідно. Також відомий розподіл концентрацій радіонуклідів (мігранту) в початковий момент часу $t = 0$: $\tilde{C}_1^0(x)$ (для частинок, що знаходяться в конвективно рухомому поровому розчині), $\tilde{C}_2^0(x)$ (для частинок, що знаходяться у зв'язаній зі скелетом ґрунту воді), $\tilde{C}_3^0(x)$ (для частинок в скелеті ґрунту) та $\tilde{Q}^0(x, r)$. На верхній поверхні ґрунту відомі концентрації $\tilde{C}_1^1(t)$, $\tilde{C}_2^1(t)$ та $\tilde{C}_3^1(t)$, а на фільтрі-вловлювачі задаються концентрації $\tilde{C}_1^2(t)$, $\tilde{C}_2^2(t)$, $\tilde{C}_3^2(t)$ або концентрації потоків відповідно $\left. \frac{\partial c_i}{\partial x} \right|_{x=l} = 0, i = 1, 3$.

Потрібно розрахувати розподіли концентрацій $c_1(x, t)$, $c_2(x, t)$, $c_3(x, t)$ та $q(x, r, t)$ по області фільтрації з часом, що дасть можливість зробити прогноз щодо ступеня очищення ґрунтів та їх подальшого використання. Провести чисельні експерименти та здійснити їхній аналіз.

Також метою є підвищення ефективності обчислень шляхом функціональної декомпозиції алгоритму розв'язування задачі математичного моделювання. Оцінку ефективності застосування паралельних обчислень доцільно провести з використанням критерію швидкодії програми або її продуктивності. Ряд факторів формують критерій продуктивності програми: архітектура обчислювальної системи, апаратна платформа (швидкість процесора, оперативної пам'яті тощо), операційна система (методи управління ресурсами, час створення потоків, загальна завантаженість), ефективність паралельного алгоритму та вибраної технології його реалізації, мова програмування та ефективність компілятора.

Практичним результатом роботи, згідно поставленого завдання, буде проектування та розробка нової версії кросплатформного програмного комплексу NanoSurface.

Математична модель задачі

Математичну модель сформульованої вище задачі у лінійній постановці при наявності так званих пасток [3] в одновимірному випадку можна описати наступною крайовою задачею [10–12]:

$$\sigma_1 \frac{\partial c_1}{\partial t} = D_1 \frac{\partial^2 c_1}{\partial x^2} - v \frac{\partial c_1}{\partial x} - \gamma_1 c_1 + \gamma_2 c_2, \quad 0 < x < l, \quad t > 0, \quad (1)$$

$$\frac{\partial c_2}{\partial t} = D_2 \frac{\partial^2 c_2}{\partial x^2} + \gamma_1 c_1 - \gamma_2 c_2 + \gamma_3 c_3 - \chi(l_0) \theta \frac{\partial q}{\partial r} \Big|_{r=R}, \quad 0 < x < l, \quad t > 0, \quad (2)$$

$$\frac{\partial c_3}{\partial t} = D_3 \frac{\partial^2 c_3}{\partial x^2} + \gamma_2 c_2 - \gamma_3 c_3, \quad 0 < x < l, \quad t > 0, \quad (3)$$

$$\frac{\partial q}{\partial t} = D_0 \left(\frac{\partial^2 q}{\partial r^2} + \frac{2}{r} \frac{\partial q}{\partial r} \right), \quad 0 < r < R, \quad t > 0, \quad (4)$$

$$v = -k(c_1) \frac{dh}{dx} + v_c \frac{\partial c_1}{\partial x}, \quad \text{div} \vec{v} + \frac{\partial n}{\partial t} = 0, \quad 0 < x < l, \quad t > 0, \quad (5)$$

$$\frac{\partial q(x, r, t)}{\partial r} \Big|_{r=0} = 0, \quad 0 < x < l, \quad t > 0, \quad (6)$$

$$q(x, r, t) \Big|_{r=R} = k_e \cdot c_2(x, t), \quad 0 < x < l, \quad t > 0, \quad (7)$$

$$l_1 c_1(0, t) = \tilde{C}_1^1(t), \quad l_2 c_1(l, t) = \tilde{C}_1^2(t), \quad (8)$$

$$l_3 c_2(0, t) = \tilde{C}_2^1(t), \quad l_4 c_2(l, t) = \tilde{C}_2^2(t), \quad (9)$$

$$l_5 c_3(0, t) = \tilde{C}_3^1(t), \quad l_6 c_3(l, t) = \tilde{C}_3^2(t), \quad (10)$$

$$h(0) = \tilde{H}_1, \quad h(l) = \tilde{H}_2, \quad (11)$$

$$c_1(x, 0) = \tilde{C}_1^0(x), \quad c_2(x, 0) = \tilde{C}_2^0(x), \quad (12)$$

$$c_3(x, 0) = \tilde{C}_3^0(x), \quad q(x, r, 0) = \tilde{Q}^0(x, r). \quad (13)$$

Тут: $c_1(x, t)$, D_1 – концентрація та коефіцієнт конвективної дифузії радіонуклідів, що знаходяться у фільтраційному потоці, $кг/м^3$ та $м^2/доба$ відповідно; $c_2(x, t)$, D_2 – концентрація та коефіцієнт молекулярної дифузії радіонуклідів, що знаходяться у воді, яка зв'язана зі скелетом ґрунту, $кг/м^3$ та $м^2/доба$ відповідно; $c_3(x, t)$, D_3 – концентрація і коефіцієнт дифузії радіонуклідів в скелеті ґрунту, ця дифузія є такою малою, що скелет ґрунту можна вважати так званою "пасткою" для цих частинок [3], $кг/м^3$ та $м^2/доба$ відповідно; $q(x, r, t)$, D_0 – концентрація та коефіцієнт дифузії радіонуклідів у частинках радіусу R , що знаходяться у скелеті ґрунту $кг/м^3$ та $м^2/доба$ відповідно; k_e – коефіцієнт впливу внутрішньочастинкового масопереносу на масоперенос у скелеті ґрунту; v – швидкість фільтрації, $м/доба$; k – коефіцієнт фільтрації води з мігрантами, $м/доба$; γ_1 , γ_2 , γ_3 – коефіцієнти масообміну, $доба^{-1}$; σ_1 – пористість ґрунту; x , $x \in [0; l]$ – вертикальна координата, напрямлена вертикально вниз, $м$; l_i , $i = \overline{1, 4}$ – диференціальні оператори, що задають граничні умови відповідно при $x = 0$ та $x = l$; t , $0 < t < t_1$ – час, $доба$; r , $0 < r < R$ – радіус (радіальна, горизонтальна змінна), $м$.

Рівняння (1) описує процес міграції радіонуклідів з концентрацією $c_1(x, t)$ шляхом конвективної дифузії у фільтраційному потоці (конвективно рухомому поровому розчині); (2) описує процес міграції радіонуклідів, що знаходяться у зв'язаній зі скелетом ґрунту воді; (3) описує міграцію радіонуклідів в скелеті ґрунту (з пастками); (4) описує внутрішньочастинковий масоперенос радіонуклідів з поточною концентрацією $q(x, r, t)$, що зв'язана із концентрацією $c_2(x, t)$ умовою (7); (5) описує фільтрацію в каталітичному пористому середовищі. Крайові умови для концентрацій $c_1(x, t)$, $c_2(x, t)$, $c_3(x, t)$, $q(x, r, t)$ та напору $h(x)$ мають вигляд (8)-(13).

В результаті різниці напорів відбувається міграція розчинених у воді речовин (радіонуклідів) у скелеті ґрунту (з пастками) фільтраційним потоком. При цьому процеси фільтрації та масоперенесення розчинених у воді речовин підлягають законам Дарсі та Фіка.

Чисельний розв'язок крайової задачі

Введемо різницеву сітку $\omega_{h_1 h_2 \tau}$ з кроками h_1, h_2 і τ по осях Ox, Or та Ot для змінних x, r, t відповідно

$$\omega_{h_1 h_2 \tau} = \left\{ (x_i, r_j, t_k) \left| \begin{array}{l} x_i = ih_1, r_j = jh_2, t_k = k\tau, \\ i = \overline{0, n_1}, j = \overline{0, n_2}, k = \overline{0, n_3}, \\ h_1 n_1 = l, r n_2 = R, \tau n_3 = T, \end{array} \right. \right\} \quad (14)$$

де n_1, n_2, n_3 – кількість кроків по просторових змінних та часу відповідно.

Для знаходження чисельного розв'язку крайової задачі (1)-(13) використано метод скінченних різниць. Зокрема, для рівнянь (1) та (4) використано монотонну різницеву схему, а для (2) та (3) – неявну різницеву схему.

Скінченнорізницеві аналоги (1)-(5), а також їх розв'язок та знаходження невідомих значень методом звичайної прогонки детально описано в [12; 13]. В даній статті, для прикладу, наведено алгоритм знаходження концентрації $c_1(x, t)$ згідно рівняння (1) та показано використання методів правої, лівої та блочної прогонки. Для апроксимації рівняння (1) використано монотонну різницеву схему О. А. Самарського [14] або уточнену монотонну різницеву схему, що побудована в роботі [15]. Тоді скінченнорізницевий аналог диференціального рівняння (1) та відповідних крайових умов матиме наступний вигляд:

$$\begin{aligned} \sigma'_1 \frac{c_{1,i}^{(k+1)} - c_{1,i}^{(k)}}{\tau} &= \chi_i^{(k)} \frac{c_{1,i+1}^{(k+1)} - 2c_{1,i}^{(k+1)} + c_{1,i-1}^{(k+1)}}{h_1^2} + \\ &+ \frac{(r^+)_i^{(k)} c_{1,i+1}^{(k+1)} - c_{1,i}^{(k+1)}}{D_1 h_1} + \frac{(r^-)_i^{(k)} c_{1,i}^{(k+1)} - c_{1,i-1}^{(k+1)}}{D_1 h_1} - \\ &- \gamma'_1 c_{1,i}^{(k+1)} + \gamma'_2 c_{2,i}^{(k+1)}, \end{aligned} \quad (15)$$

$$c_{1,i}^{(0)} = \tilde{C}_1^0(x_i), c_{1,0}^{(k)} = \tilde{C}_1^1(t_k), c_{1,n_1}^{(k)} = \tilde{C}_1^2(t_k), \quad (16)$$

де $i = \overline{1, n_1 - 1}, k = \overline{0, n_3}$.

Тут введені наступні позначення:

$$\chi_i^{(k)} = \frac{1}{1 + \frac{h_1 |r_i^{(k)}|}{2D_1}} = 1 - \frac{h_1 |r_i^{(k)}|}{2D_1} + O(h_1^2), \quad \sigma'_1 = \frac{\sigma_1}{D_1}, \quad \gamma'_1 = \frac{\gamma_1}{D_1}, \quad \gamma'_2 = \frac{\gamma_2}{D_1}, \quad r_i^{(k)} = (r^+)_i^{(k)} + (r^-)_i^{(k)},$$

$$(r^+)_i^{(k)} = \frac{-v_i^{(k)} + |v_i^{(k)}|}{2} \geq 0, \quad (r^-)_i^{(k)} = \frac{-v_i^{(k)} - |v_i^{(k)}|}{2} \leq 0.$$

Для знаходження концентрації солей $c_1(x, t)$ з різницевого рівняння (15) використано метод прогонки. Для цього представимо (15) у вигляді

$$\begin{cases} a_i^1 c_{1,i-1}^{(k+1)} - \bar{c}_i^1 c_{1,i}^{(k+1)} + b_i^1 c_{1,i+1}^{(k+1)} = -c_{1,i}^{(k)} - f_i^{1,(k+1)}, \\ c_{1,0}^{(k+1)} = \mu_1^1 c_{1,1}^{(k+1)} + \mu_2^1, \\ c_{1,n_1}^{(k+1)} = \mu_3^1 c_{1,n_1-1}^{(k+1)} + \mu_4^1, \end{cases} \quad (17)$$

де

$$a_i^1 = \frac{\tau}{\sigma_1'} \left(\frac{\chi_i^{(k)}}{h_1^2} - \frac{(r^-)^{(k)}}{h_1 D_1} \right), \quad b_i^1 = \frac{\tau}{\sigma_1'} \left(\frac{\chi_i^{(k)}}{h_1^2} + \frac{(r^+)^{(k)}}{h_1 D_1} \right), \quad \bar{c}_i^1 = 1 + \frac{\tau}{\sigma_1'} \left(\frac{2\chi_i^{(k)}}{h_1^2} + \frac{1}{h_1 D_1} \left((r^+)^{(k)} - (r^-)^{(k)} \right) + \gamma_1' \right),$$

$$f_i^{1,(k+1)} = \frac{\tau}{\sigma_1'} \gamma_2' c_{2,i}^{(k+1)}, \quad \mu_1^1 \equiv 0, \quad \mu_3^1 \equiv 0, \quad \mu_2^1 = \tilde{C}_1^1, \quad \mu_4^1 \equiv \tilde{C}_1^2.$$

Значення концентрації $c_1(x, t)$ на часовому шарі $(k+1)$ знайдено методом правої прогонки, використовуючи співвідношення

$$c_{1,i}^{(k+1)} = \alpha_{i+1}^1 c_{1,i+1}^{(k+1)} + \beta_{i+1}^1, \quad (18)$$

$$\alpha_{i+1}^1 = \frac{b_i^1}{\bar{c}_i^1 - \alpha_i^1 a_i^1}, \quad \beta_{i+1}^1 = \frac{a_i^1 \beta_i^1 + c_{1,i}^{(k)} + f_i^{1,(k+1)}}{\bar{c}_i^1 - \alpha_i^1 a_i^1}, \quad (19)$$

$$i = \overline{1, n_1 - 1}, \quad k = \overline{1, n_3}, \quad \alpha_1^1 = \mu_1^1 \equiv 0, \quad \beta_1^1 = \mu_2^1 \equiv \tilde{C}_1^1,$$

або методом лівої прогонки, використовуючи наступну залежність

$$c_{1,i+1}^{(k+1)} = \xi_{i+1}^1 c_{1,i}^{(k+1)} + \eta_{i+1}^1, \quad (20)$$

$$\xi_i^1 = \frac{a_i^1}{\bar{c}_i^1 - \xi_{i+1}^1 b_i^1}, \quad \eta_i^1 = \frac{\eta_{i+1}^1 \beta_i^1 + c_{1,i}^{(k)} + f_i^{1,(k+1)}}{\bar{c}_i^1 - \xi_{i+1}^1 b_i^1}, \quad (21)$$

$$i = \overline{n_1 - 1, 1}, \quad k = \overline{1, n_3}, \quad \eta_n^1 = \mu_3^1 \equiv 0, \quad \eta_1^1 = \mu_4^1 \equiv \tilde{C}_1^2.$$

Можна показати, що умови стійкості прогонки $|\bar{c}_i^1| > |a_i^1| + |b_i^1|$ виконуються.

Чисельні експерименти

Проведено чисельні експерименти з використанням розробленого програмного забезпечення [12] з наступними вхідними даними:

$$l = 10 \text{ м}, \quad \dot{H}_1 = 1 \text{ м}, \quad \dot{H}_2 = 0 \text{ м}, \quad \sigma_1 = 0,25, \quad \tau = 30 \text{ діб}, \quad \gamma_1 = \gamma_2 = \gamma_3 = 0,0065, \quad R = 5 \cdot 10^{-8} \text{ м},$$

$$D_1 = 2 \cdot 10^{-3} \frac{\text{м}^2}{\text{добу}}, \quad D_2 = 1 \cdot 10^{-5} \frac{\text{м}^2}{\text{доба}}, \quad D_3 = 1 \cdot 10^{-10} \frac{\text{м}^2}{\text{доба}}, \quad D_0 = 1 \cdot 10^{-18} \frac{\text{м}^2}{\text{доба}}, \quad \tilde{C}_1^1(t) = 5 \frac{\text{кЗ}}{\text{м}^3},$$

$$\tilde{C}_1^2(t) = 0 \frac{\text{кЗ}}{\text{м}^3}, \quad \tilde{C}_1^3(t) = 0 \frac{\text{кЗ}}{\text{м}^3}, \quad \tilde{C}_2^1(t) = 5 \frac{\text{кЗ}}{\text{м}^3}, \quad \tilde{C}_2^2(t) = 0 \frac{\text{кЗ}}{\text{м}^3}, \quad \tilde{C}_2^3(t) = 0 \frac{\text{кЗ}}{\text{м}^3}, \quad \tilde{C}_1^0(x) = 5 \frac{\text{кЗ}}{\text{м}^3},$$

$$\tilde{C}_2^0(x) = 5 \frac{\text{кЗ}}{\text{м}^3}, \quad \tilde{C}_3^0(x) = 0 \frac{\text{кЗ}}{\text{м}^3}, \quad \tilde{Q}^0(t) = 0 \frac{\text{кЗ}}{\text{м}^3}, \quad l_0 = 0 \text{ м}.$$

Після проведення серії чисельних експериментів отримано графіки та таблиці розподілів концентрацій та швидкості фільтрації. На рис. 2 наведено 3D графіки розподілів концентрацій $c_1(x, t)$, $c_2(x, t)$ та $c_3(x, t)$ на глибині x шару ґрунту в часі.

Провівши чисельні експерименти та проаналізувавши отримані графіки, стає очевидним вплив дифузії в скелеті ґрунту. Як видно з рис. 2, концентрація $c_3(x, t)$ розподіляється повільніше за концентрацію $c_2(x, t)$ на часовому кроці №5 (30 місяців). В той час, коли концентрації $c_1(x, t)$ та $c_2(x, t)$ спадають, концентрація радіонуклідів $c_3(x, t)$ залишається присутньою в кристалічній решітці скелету ґрунту (потрапляє в так звані «пастки»). Дану особливість потрібно враховувати при прогнозуванні очищення ґрунту за допомогою спецзасобів.

Ефективність паралельних обчислень

Програмна реалізація чисельного розв'язку математичної моделі (1)-(13) є нетривіальною задачею. Це впливає із взаємозв'язків між шуканими концентраціями $c_1(x, t)$, $c_2(x, t)$, $c_3(x, t)$, $q(x, r, t)$ та швидкістю фільтрації U . Тому кожна математична модель повинна розглядатися комплексно і універсального рішення для використання паралельних обчислень не існує. Враховуючи це, обчислювальний алгоритм у нашому випадку можна розпаралелити тільки для задання початкових та граничних умов першого роду (8)-(13), а далі послідовно знаходити швидкість фільтрації U , концентрації $q(x, r, t)$,

$c_3(x, t)$, $c_2(x, t)$ та $c_1(x, t)$. Однак, незважаючи на послідовність деяких обчислень їх можна розпаралелити на нижчому рівні – зокрема на етапі роботи методу прогонки.

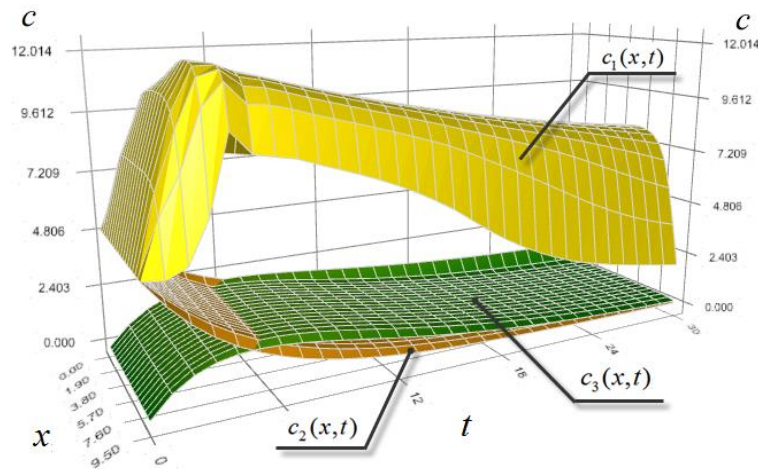


Рисунок 2 – Час роботи обчислювального алгоритму при розпаралеленні різних частин коду для 8 потоків

В даній роботі виведені прогоночні коефіцієнти для методу прогонки, що зазвичай виконується в однопоточному режимі. Для роботи в двох потоках на системах із спільною пам'яттю використовують комбінацію правої та лівої прогонки, так звану зустрічну прогонку.

Оцінимо складність алгоритму методу правої прогонки. При виконанні прямого ходу за формулами (19) потрібно $8(N-2)+2$ операцій. Для виконання зворотнього ходу за формулою (18) потрібно $2(N-1)+5$ операцій. Таким чином, загальне число операцій можна оцінити величиною

$$T_1 = 10n + O(1), \quad (22)$$

а час розв'язку системи методом прогонки при великих N буде визначатися як $T = 10n\tau$, де τ – час виконання однієї операції. Неважко помітити, що алгоритмічна складність лівої прогонки (20), (21) становить також $10n + O(1)$.

Комбінація методів лівої і правої прогонки дає метод зустрічної прогонки, який допускає розпаралелення процесу обчислень на два потоки. Розділимо систему між двома потоками - перший буде оперувати рівняннями з номерами $1 \leq i \leq p$, другий – рівняннями $p \leq i \leq n$, де $p = \lfloor n/2 \rfloor$.

При паралельному розв'язку в першому потоці за формулами (19) обчислюються прогоночні коефіцієнти α_i and β_i при $1 \leq i \leq p$, а в другому потоці за формулами (21) знаходяться ξ_i і η_i при $p \leq i \leq n$. При $i = p$ проводиться спряження рішень у формі (18) і (20), тобто знаходимо значення x_p з системи рівнянь:

$$\begin{cases} x_p = \alpha_{p+1}x_{p+1} + \beta_{p+1} \\ x_{p+1} = \xi_{p+1}x_p + \eta_{p+1} \end{cases} \quad (23)$$

Знайшовши вказане значення, в першому потоці за формулою (18) можна знайти всі x_i при $1 \leq i \leq p$, а в другому, за формулою (20) – всі x_i при $p \leq i \leq n$.

Час виконання методу зустрічної прогонки можна оцінити як $T_2 = 5n\tau + \delta$, де δ – час, необхідний на організацію та закриття паралельної секції. Варто зазначити, що розрахунки і при прямому, і при зворотньому ходах здійснюються незалежно, тому теоретичне прискорення дорівнює двом.

У системах, де можна задіяти більше двох ядер процесорів для обчислень, використовують блочну прогонку [16].

Оцінимо складність алгоритму паралельного варіанту методу прогонки [16]. Відповідно до введених раніше позначень n є порядок системи лінійних рівнянь (кількість кроків по змінній x), а p ,

$p < n$, визначає число потоків. Тим чином, матриця коефіцієнтів $a_i^1, b_i^1, \bar{c}_i^1, f_i^{1,(k)}$ має розмір $n \times n$ і, відповідно, $m = n/p$ є розмір смуги матриці на кожному процесорі.

При виконанні прямого ходу алгоритму на кожній ітерації кожен процесор повинен здійснити виключення у межах своєї смуги піддіагональних елементів (що вимагає $8(m-1)$ операцій) і наддіагональних елементів (що вимагає $7m$ операцій).

Потім варто провести збірку допоміжної тридіагональної системи рівнянь в одному потоці і здійснити її розв'язок методом прогонки. Відповідно до оцінки (22) витрати на виконання цього послідовного етапу обчислень складуть близько $10p$ операцій.

На наступному етапі алгоритму кожен процесор виконує зворотний хід алгоритму, який вимагатиме $5(m-1)$ операцій. Таким чином, загальну складність паралельного методу прогонки можна оцінити як

$$T_p = 20m + 10p, \quad (24)$$

За результатами виконаного аналізу ми можемо знайти показники прискорення та ефективності паралельного варіанту методу прогонки [17]. Прискорення (speedup) являє собою відношення послідовного часу виконання задачі до паралельного алгоритму:

$$S_p = \frac{T_1}{T_p} = \frac{10n}{20\frac{n}{p} + 10p} = p \frac{10n}{20n + p^2}. \quad (25)$$

Ефективність (efficiency) використання паралельним алгоритмом процесорів (потоків) при розв'язку блочної прогонки визначається наступним співвідношенням:

$$E_p = \frac{S_p}{p} = \frac{10n}{20n + p^2}. \quad (26)$$

Ідеальний випадок досягається при $S_p = p$ та $E_p = 1$, тобто кожний новий процесор збільшує швидкість виконання задачі при 100% їх використанні. Однак спроби підвищення одного показника (прискорення чи ефективності) можуть привести до погіршення ситуації по іншому показнику. З наведених вище співвідношень видно, що у нашому випадку компроміс між даними показниками досягається при розв'язанні системи рівнянь з великою кількістю невідомих, при якому $p \ll n$ і показники прискорення та ефективності визначаються як

$$S_p \approx \frac{p}{2}, \quad E_p \approx \frac{1}{2}. \quad (27)$$

Особливості реалізації паралельних обчислень та їх аналіз

Реалізація паралельних обчислень здійснена за допомогою технології OpenMP [18]. Стандарт, що описує дану технологію, надає опис директив компілятора, бібліотечних процедур і змінних оточення, які призначені для програмування багатопотокових додатків на багато процесорних системах із спільною пам'яттю.

Було реалізовано методи зустрічної та блочної прогонки із різними ступенями розпаралелення. В основному використовувались синтаксичні директиви *for* для циклів і початкової ініціалізації; *sections* в зустрічній прогонці; *if (умова)* як певна умова використання розпаралелення.

Дослідження ефективності паралельних обчислень проводились на комп'ютері з наступними характеристиками:

- Процесор: Intel(R) Core(TM) i7-3630QM CPU @ 2.40GHz (8 CPUs)
- Оперативна пам'ять: 8192MB RAM
- Відеокарта: NVIDIA GeForce GT 650M x2
- Операційна система: Windows 10 Pro 64-bit
- Компілятор: MinGW 32/64bits (gcc 5.3.0)

На рис. 3 зображено порівняння графіків часу обчислення для алгоритму без використання паралельних обчислень, з додаванням розпаралелення початкової ініціалізації всіх змінних, з додаванням розпаралелення ініціалізації прогоночних коефіцієнтів a_i, b_i, \bar{c}_i, f_i , з додаванням розпаралелення швидкості фільтрації і з додаванням розпаралелення концентрації $q(x, r, t)$. З даних графіків видно, що найбільший ефект розпаралелення дає механізм ініціалізації прогоночних коефіцієнтів і обчислення концентрації $q(x, r, t)$.

Також з рис. 3 можна побачити, що розпаралелення програми дає хороші результати. Наприклад, для 120000 кроків розбиття по змінній x , час обчислення звичайного коду від паралельного відрізняється в 3.5 разів (при 8 потоках). Це свідчить про достатню оптимізацію коду, а зі збільшенням кількості потоків, швидкість обчислення буде тільки зростати.

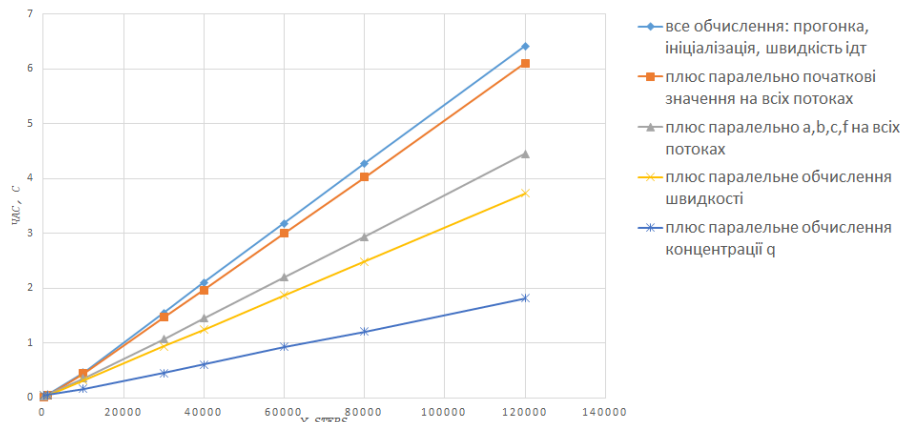


Рисунок 3 – Час роботи обчислювального алгоритму при розпаралеленні різних частин коду для 8 потоків

Під час написання програми була виявлена певна особливість компілятора: при заданні змінної типу `double` з використанням оператора `new` (наприклад, `double * value = new double[r_steps + 1]`) при паралельному обчисленні концентрації $q(x, r, t)$, різниці в часі виконання майже немає, а коли використовуємо форму задання змінної типу `double` без використання оператора `new` (наприклад, `double value[r_steps + 1]`), то час виконання при 80000 кроках стає в 6 разів менший (рис. 4). Тобто, оголошення масиву змінної довжини (Variable length arrays), що стало можливим згідно стандарту C99 [19], дає суттєвий приріст швидкодії.

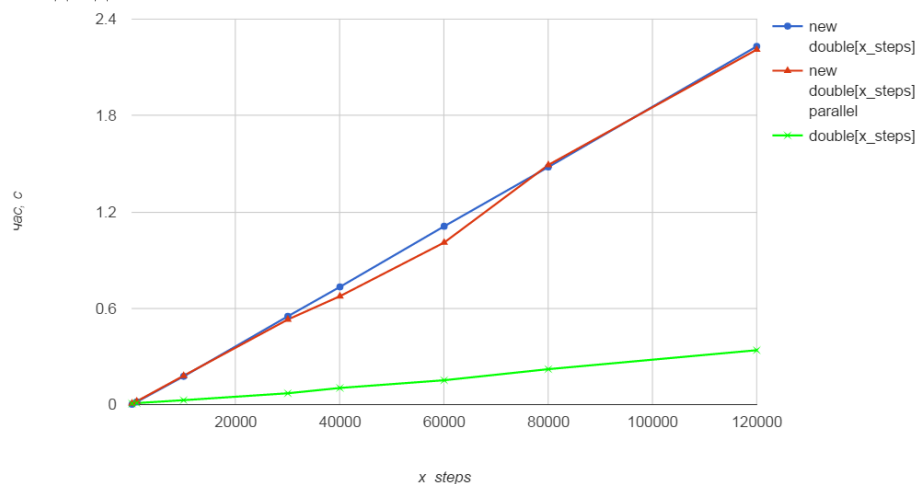


Рисунок 4 – Порівняння швидкості обчислення при різних способах оголошення масивів

Подібні ситуації можуть виникнути під час оголошення та роботи з динамічними двовимірними масивами (коли розмір масиву визначається в процесі виконання програми). Оголошення `double a[n][n]` та `std::array<std::array<double, n>, n>` не працюють, оскільки `new` може приймати тільки перше динамічне значення n під час виконання виразу. Для вирішення даної проблеми можна написати наступний код:

```
// Array of size n of pointers to double
double **a = new double *[n];
for (int i = 0; i != n; ++i)
{
    a[i] = new double[n];
}
```

У цьому випадку ми можемо просто отримати доступ до необхідного елемента масиву за допомогою `a[i][j]`. Але цей масив a є одновимірним масивом вказівників, а не одним безперервним блоком даних

із швидким доступом, що виділяються в пам'яті. Тому доступ до i -го j -го елемента буде повільнішим, ніж це було б, якщо дані були поміщені в один двовимірний масив.

Висновки

В роботі сформульовано постановку та побудовано математичну модель процесу вертикальної міграції радіонуклідів у горизонтальному шарі ґрунту великої протяжності. Шар ґрунту містить мікро- та наночастинки пористої структури і представляє собою каталітичне пористе середовище при наявності фільтрів-вловлювачів. Чисельний розв'язок поставленої крайової задачі знайдено методом скінченних різниць. Для проведення чисельних експериментів розроблено програмний комплекс NanoSurface [12]. В результаті проведення чисельних експериментів та аналізу отриманих графіків, стає очевидним ефективний вплив мікрочастинок на швидкість міграції радіонуклідів та необхідність врахування дифузії в скелеті ґрунту. Даний факт потрібно враховувати при подальших дослідженнях.

Встановлено алгоритмічну складність алгоритмів правої, лівої, зустрічної та блочної прогонки. На основі цих даних проведено чисельні експерименти з метою виявлення часових затрат та аналізу ефективності паралельних обчислень. Реалізація обчислювальних методів за допомогою засобів паралельного програмування дає суттєвий приріст ефективності. Зокрема, як показано на рис. 3 для 120000 кроків розбиття по змінній x , час виконання звичайного коду від паралельного алгоритму обчислення відрізняється в 3,5 рази (при 8 потоках). Рис. 4 наочно демонструє зменшення швидкості обчислень у 6 разів для 80000 ітерацій при використанні Variable length arrays замість динамічного задання масиву за допомогою оператора new. Однак, дана можливість доступна в компіляторах із підтримкою стандарту C99 або вище. Тому необхідно звертати увагу на вибір правильного інструментарію розробника та форми оголошення масивів.

Стаття написана і опублікована за підтримки Державного фонду фундаментальних досліджень України, проект № Ф76/93-2017 від 08.09.17 р. «Інформаційне забезпечення розвитку конкурентоспроможного органічного сільського господарства України в умовах євроінтеграції».

Список літератури

1. *Auffan M. Nanomaterials as adsorbents / M. Auffan, H. J. Shipley, S. Yean, A.T. Kan, M. Tomson, J. Rose, J.-Y. Bottero // Environmental Nanotechnology: Applications and Impacts of Nanomaterials, McGraw-Hill, New York. – 2007. – С.371–392.*
2. *Бурак Я.Й. Вихідні положення математичної моделі гетеродифузного переносу радіонуклідів у приповерхневих шарах Землі / Я.Й. Бурак, Є.Я. Чапля // Доп. НАН України. – 1993. – Т.10. – С.59–63.*
3. *Чапля Є.Я. Моделювання процесів стаціонарної гетеро дифузії розпадної речовини у середовищі з пастками / Є.Я. Чапля, О.Ю. Чернуха, Гончарук, В. Є., Паби́рівський, В. В. // Моделювання та інформаційні технології. – 2013. – Т.70. – С.96–108.*
4. *Петрик М.Р. Математичне моделювання двокомпонентної десорбції в нанопористому середовищі / М.Р. Петрик, Д.М. Михалик, О.Ю. Петрик, М.І. Шинкарик // Вісник ХНТУ „Математичне моделювання фізичних і технологічних процесів і технічних систем“. – 2016. – №3 (58). – С.384–387.*
5. *No exponential is forever: but " Forever" can be delayed! [semiconductor industry] / ред. G. E. Moore: IEEE, 2003. – 20-23.*
6. *Богаєнко В.А. Об оптимизации вычислительного процесса математического моделирования сложных задач / В.А. Богаєнко // Управляющие системы и машины. – 2010. – №6. – С.9–24.*
7. *Богаєнко В.О. Паралельні алгоритми моделювання процесу фільтраційної консолідації під дією двокомпонентного розчину / В.О. Богаєнко // Математичне та комп'ютерне моделювання. Серія: Фізико-математичні науки. – 2011. – .*
8. *Богаєнко В.А. Методика численного моделирования процесса электрокинетической очистки грунтов в неизотермических условиях / В.А. Богаєнко // Управляющие системы и машины. – 2013. – №1. – С.3–10.*
9. *Белоус М.В. Использование программного комплекса НАДРА-3D для моделирования регионального режима фильтрации воды / М.В. Белоус, В.С. Дейнека // Компьютерная математика. – 2010. – №1. – С.35–42.*
10. *Власюк А.П. Математичне моделювання переносу сольових розчинів при фільтрації підземних вод у ґрунтових масивах / А.П. Власюк, О.П. Остапчук. – Рівне: Національний університет водного господарства та природокористування, 2015. – 214 с.*
11. *Diffusion Fundamentals / ed. J. Kärgler. – Leipzig: Leipziger Universitätsverlag, 2005. – 615 S.*
12. *Власюк А.П. Математичне моделювання вертикальної міграції радіонуклідів в каталітичному пористому середовищі / А.П. Власюк, В.В. Жуковський // Вісник Київського національного університету. Сер. фіз.-мат. наук. – 2015. – Т.1. – С.89–95.*

13. *Vlasyuk A.P. Mathematical Modelling of Vertical Migration of Radionuclides in Catalytic Porous Media with Traps in Linear Case / A.P. Vlasyuk, V.V. Zhukovskyy, M.M. Bondarchuk // Theoretical and Applied Aspects of Cybernetics. Proceedings of the 5th International Scientific Conference of Students and Young Scientists. – 2015. – P.208–219.*
14. *Самарский А.А. Теория разностных схем: Учеб. пособие для вузов по спец. "Прикл. математика" / А.А. Самарский. – Москва: Наука, 1989. – 616 с.*
15. *Власюк А.П. Монотонна різницева схема для нелінійного одновимірного рівняння параболічного типу, що містить першу похідну / А.П. Власюк, О.П. Остапчук, О.М. Степанченко // Вісник Київського університету. Сер. фіз.-мат. наук. – 2005. – Т.Вип. 2. – С.188.*
16. Баркалов К.А. Методы параллельных вычислений / Баркалов К.А., 2011. – 124 с.
17. *Гергель В. Высокопроизводительные вычисления для многопроцессорных многоядерных систем: [учебник для вузов] / В. Гергель: Физматлит, 2010.*
18. OpenMP.org. – <http://openmp.org/wp/>.
19. ISO/IEC 9899:1999 - Programming languages - C. – http://www.iso.org/iso/iso_catalogue/catalogue_ics/catalogue_detail_ics.htm?csnumber=29237.
Стаття надійшла: 14.05.2017.

Відомості про авторів

Власюк Анатолій Павлович – професор, академік Академії наук вищої школи України, доктор технічних наук, професор Національного університету «Острозька академія», Україна.

Жуковський Віктор Володимирович – старший викладач кафедри прикладної математики Національного університету водного господарства та природокористування, Україна.

Жуковська Наталія Анатоліївна – кандидат технічних наук, старший викладач кафедри прикладної математики Національного університету водного господарства та природокористування, Україна.