

УДК 519.6

В. О. Богаєнко, канд. техн. наук

Інститут кібернетики імені В. М. Глушкова НАН України, м. Київ

ПАРАЛЕЛЬНІ АЛГОРИТМИ МОДЕЛЮВАННЯ ПРОЦЕСУ ФІЛЬТРАЦІЙНОЇ КОНСОЛІДАЦІЇ ПІД ДІЄЮ ДВОКОМПОНЕНТНОГО РОЗЧИНУ

У статті розглядається задача моделювання процесу фільтраційної консолідації ґрунтів під дією двокомпонентного розчину. Математична модель таких процесів в одновимірному випадку поширюється на тривимірний та пропонується ряд паралельних алгоритмів розв'язання задач щодо неї, зокрема багатопоточний алгоритм, алгоритми для кластерних систем та графічних процесорів (GPU).

Ключеві слова: *фільтраційна консолідація, двокомпонентні розчини, математичне моделювання, паралельні алгоритми.*

Вступ. Комплексне математичне моделювання консолідації ґрунтів є актуальною проблемою при прогнозуванні гідродинамічних процесів. При цьому важливим є врахування таких факторів, як солоність порового розчину, неізонермічність та релаксаційність процесів. На цей час розроблена низка математичних моделей [1—6], що описують такі процеси, деякі з яких досліджені лише для одновимірного випадку [4—6]. Процедури розв'язання задач щодо таких моделей мають високу обчислювальну складність, що зумовлює необхідність їх адаптації для високопродуктивних обчислювальних систем, зокрема, кластерів.

У цій статті розглядаються паралельні алгоритми для розв'язання задач моделювання процесів фільтраційної консолідації ґрунтів під дією двокомпонентного розчину в неізотермічних умовах у тривимірному випадку. На основі одновимірної, будується тривимірна модель, скінченно-елементний алгоритм розв'язання і схеми його розпаралелення для систем з розподіленою та спільною пам'яттю.

Математична модель. Розглядається математична модель фільтраційної консолідації ґрунтів у припущенні їх ізотропності під дією двокомпонентного розчину з урахування впливу температурного режиму у векторній формі, побудована на основі одновимірної моделі, описаної у [5; 6]:

$$\frac{\partial H}{\partial t} = \operatorname{div}(C_v \operatorname{grad} H) - \operatorname{div}(\operatorname{grad} G) + \alpha_T \frac{\partial T}{\partial t}, \quad \alpha_T = \frac{\sigma \beta_T C_v}{k},$$

$$\sigma \frac{\partial C_i}{\partial t} = D_i \operatorname{div}(\operatorname{grad} C_i) + \vec{u} \cdot \operatorname{grad} C_i + D_T \operatorname{div}(\operatorname{grad} T), \quad i = 1, 2,$$

$$C_T \frac{\partial T}{\partial t} = \lambda \operatorname{div}(\operatorname{grad} T) + \rho C_P \bar{u} \cdot \operatorname{grad} T, \quad (1)$$

$$G = \kappa_1 C_1 + \kappa_2 C_2 + \theta T, \quad \kappa_i = \frac{v_i C_v}{k}, \quad \theta = \frac{\mu C_v}{k},$$

$$\bar{u} = \operatorname{grad}(kH - v_1 C_1 - v_2 C_2 - \mu T),$$

де H — надлишковий напір, C_i — концентрації солей у поровому розчині, T — температура, k — коефіцієнт фільтрації, C_v — коефіцієнт консолідації, β_T — коефіцієнт термічного розширення, σ — середнє значення пористості ґрунту, D_i — коефіцієнти молекулярної дифузії компонентів розчину, v_i , μ — коефіцієнти хімічного та термоосмосу, D_T — коефіцієнт термодифузії, C_T , C_P — об'ємна та питома теплоємності порового розчину, λ — коефіцієнт теплопровідності.

Систему рівнянь (1) доповнимо наступними початковими й крайовими умовами:

$$\begin{aligned} H(\bar{x}, 0) = 0, \quad H(\bar{x}, t)|_{\Gamma_1} = H_1, \quad \frac{\partial}{\partial \bar{n}} H(\bar{x}, t)|_{\Gamma_2} = 0, \quad H(\bar{x}, t)|_{\Gamma_3} = 0, \\ C_i(\bar{x}, 0) = 0, \quad C_i(\bar{x}, t)|_{\Gamma_1} = C_{i0}, \quad \frac{\partial}{\partial \bar{n}} C_i(\bar{x}, t)|_{\Gamma_2} = 0, \quad C_i(\bar{x}, t)|_{\Gamma_3} = 0, \quad (2) \\ T(\bar{x}, 0) = T_0, \quad T(\bar{x}, t)|_{\Gamma_1} = T_1, \quad \frac{\partial}{\partial \bar{n}} T(\bar{x}, t)|_{\Gamma_2} = 0, \quad T(\bar{x}, t)|_{\Gamma_3} = 0, \end{aligned}$$

де Γ_1 — частина границі області моделювання, де відбувається контакт з рідиною, Γ_2 — непроникна частина границі, Γ_3 — границя проникної основи.

У безрозмірних змінних система (1)-(2) прийме наступний вигляд:

$$\begin{aligned} \frac{\partial H'}{\partial t'} &= \operatorname{div}(\operatorname{grad} H') - \operatorname{div}(\operatorname{grad} G') + \alpha'_T \frac{\partial T'}{\partial t'}, \\ \sigma \frac{\partial C'_i}{\partial t'} &= D'_i \operatorname{div}(\operatorname{grad} C'_i) + \bar{u}' \cdot \operatorname{grad} C'_i + r_i \operatorname{div}(\operatorname{grad} T'), \quad i = 1, 2, \\ \frac{\partial T'}{\partial t'} &= \lambda' \operatorname{div}(\operatorname{grad} T') + \bar{v}' \cdot \operatorname{grad} T', \\ G' &= \kappa'_1 C'_1 + \kappa'_2 C'_2 + \theta' T', \quad (3) \\ \bar{u}' &= \operatorname{grad}(u' H' - v'_1 C'_1 - v'_2 C'_2 - \mu' T'), \\ \bar{v}' &= \operatorname{grad}(u'_1 H' - \delta'_1 C'_1 - \delta'_2 C'_2 - \gamma'_1 T'), \end{aligned}$$

де

$$\begin{aligned} \alpha'_T &= \frac{\alpha_T T_0}{H_1}, \quad t' = C_v t, \quad H' = H / H_1, \quad D'_i = \frac{D_i}{C_v}, \quad r_i = \frac{D_T T_0}{C_v C_{i0}}, \\ \lambda' &= \frac{\lambda}{C_T C_v}, \quad C'_i = C_i / C_{i0}, \quad T' = T / T_0, \quad \kappa'_i = \frac{\kappa_i C_{i0}}{C_v H_1}, \quad \theta' = \frac{\theta T_0}{C_v H_1}, \\ u' &= \frac{k H_1}{C_v}, \quad v'_i = \frac{v_i C_{i0}}{C_v}, \quad \mu' = \frac{\mu T_0}{C_v}, \quad u'_1 = \frac{\rho C_p}{C_T} u', \quad \delta'_i = \frac{\rho C_p}{C_T} v'_i, \\ \gamma'_1 &= \frac{\rho C_p}{C_T} \mu'. \end{aligned}$$

$$\begin{aligned} H'(\bar{x}, 0) &= 0, \quad H'(\bar{x}, t') \Big|_{\Gamma_1} = 1, \quad \frac{\partial}{\partial \bar{n}} H'(\bar{x}, t') \Big|_{\Gamma_2} = 0, \quad H'(\bar{x}, t') \Big|_{\Gamma_3} = 0, \\ C'_i(\bar{x}, 0) &= 1, \quad C'_i(\bar{x}, t') \Big|_{\Gamma_1} = C'_{i1} = \frac{C_{i1}}{C_{i0}}, \quad \frac{\partial}{\partial \bar{n}} C'_i(\bar{x}, t') \Big|_{\Gamma_2} = 0, \quad C'_i(\bar{x}, t') \Big|_{\Gamma_3} = 0, \quad (4) \\ T'(\bar{x}, 0) &= 1, \quad T'(\bar{x}, t') \Big|_{\Gamma_1} = T'_1 = \frac{T_1}{T_0}, \quad \frac{\partial}{\partial \bar{n}} T'(\bar{x}, t') \Big|_{\Gamma_2} = 0, \quad T'(\bar{x}, t') \Big|_{\Gamma_3} = 0. \end{aligned}$$

У подальшому знак штрих над безрозмірними величинами опустимо.

Скінченно-елементна дискретизація. Систему (3)—(4) дискретизуємо за методом скінченних елементів. Будемо шукати її розв'язок у вигляді

$$\begin{aligned} \hat{H} &= \sum_{i=1}^{N_v} a_i(t) N_i^{(1)}(\bar{x}) + W_1(\bar{x}), \quad \hat{C}_1 = \sum_{i=1}^{N_v} b_i(t) N_i^{(2)}(\bar{x}) + W_2(\bar{x}), \\ \hat{C}_2 &= \sum_{i=1}^{N_v} q_i(t) N_i^{(3)}(\bar{x}) + W_3(\bar{x}), \quad \hat{T} = \sum_{i=1}^{N_v} p_i(t) N_i^{(4)}(\bar{x}) + W_4(\bar{x}), \end{aligned} \quad (5)$$

де $N_i^{(j)}(\bar{x})$ — базисні функції N_v -вимірного підпростору простору Соболева функцій, що задовольняють крайовим умовам для відповідних величин (H для $j = 1$ і т.д.), а $W_1(\bar{x})$ — відомі функції, які задовольняють цим умовам.

Перейшовши до варіаційної постановки задачі (3)—(4) та враховуючи (5) отримуємо систему звичайних диференціальних рівнянь щодо коефіцієнтів

$$\begin{aligned} \bar{A} &= \{a_i(t)\}_{i=1}^{N_v}, \quad \bar{B} = \{b_i(t)\}_{i=1}^{N_v}, \quad \bar{P} = \{p_i(t)\}_{i=1}^{N_v}, \quad \bar{Q} = \{q_i(t)\}_{i=1}^{N_v} : \\ M_1 \frac{d\bar{A}}{dt} + L\bar{A} - \alpha_T M_2 \frac{d\bar{P}}{dt} + \bar{F}_1 &= 0, \end{aligned}$$

$$\begin{aligned}\sigma M_3 \frac{d\bar{B}}{dt} + L_1 \bar{B} + \bar{F}_2 &= 0, \\ \sigma M_4 \frac{d\bar{Q}}{dt} + L_2 \bar{Q} + \bar{F}_3 &= 0, \\ M_5 \frac{d\bar{P}}{dt} + L_3 \bar{P} + \bar{F}_4 &= 0,\end{aligned}\tag{6}$$

де

$$\begin{aligned}M_{1ij} &= (N_j^{(1)}, N_i^{(1)}), M_{2ij} = (N_j^{(1)}, N_i^{(4)}), M_{3ij} = (N_j^{(2)}, N_i^{(2)}), \\ M_{4ij} &= (N_j^{(3)}, N_i^{(3)}), M_{5ij} = (N_j^{(4)}, N_i^{(4)}), \\ L_{ij} &= \sum_{k=1}^3 \left(\frac{dN_j^{(1)}}{dx_k}, \frac{dN_i^{(1)}}{dx_k} \right),\end{aligned}$$

$$\begin{aligned}L_{1ij} &= D_1 \sum_{k=1}^3 \left(\frac{dN_j^{(2)}}{dx_k}, \frac{dN_i^{(2)}}{dx_k} \right) - \sum_{k=1}^3 \left(\frac{\partial}{\partial x_k} (u\hat{H} - v_2 \hat{C}_2 - \mu\hat{T}) \frac{dN_i^{(2)}}{dx_k}, N_j^{(2)} \right), \\ \left(\frac{\partial}{\partial x_k} \hat{H} \frac{dN_i^{(2)}}{dx_k}, N_j^{(2)} \right) &= \sum_{l=1}^{N_r} a_l \left(\frac{dN_l^{(1)}}{dx_k} \frac{dN_i^{(2)}}{dx_k}, N_j^{(2)} \right) + \left(\frac{dW_1}{dx_k} \frac{dN_i^{(2)}}{dx_k}, N_j^{(2)} \right), \\ L_{2ij} &= D_2 \sum_{k=1}^3 \left(\frac{dN_j^{(3)}}{dx_k}, \frac{dN_i^{(3)}}{dx_k} \right) - \sum_{k=1}^3 \left(\frac{\partial}{\partial x_k} (u\hat{H} - v_1 \hat{C}_1 - \mu\hat{T}) \frac{dN_i^{(3)}}{dx_k}, N_j^{(3)} \right), \\ L_{3ij} &= \lambda \sum_{k=1}^3 \left(\frac{dN_j^{(4)}}{dx_k}, \frac{dN_i^{(4)}}{dx_k} \right) - \sum_{k=1}^3 \left(\frac{\partial}{\partial x_k} (u_1 \hat{H} - \delta_1 \hat{C}_1 - \delta_2 \hat{C}_2) \frac{dN_i^{(4)}}{dx_k}, N_j^{(4)} \right), \\ F_{1j} &= \sum_{k=1}^3 \left(\frac{\partial}{\partial x_k} (W_1 - \kappa_1 \hat{C}_1 - \kappa_2 \hat{C}_2 - \theta\hat{T}), \frac{dN_j^{(1)}}{dx_k} \right), \\ \left(\frac{\partial}{\partial x_k} \hat{T}, \frac{dN_j^{(1)}}{dx_k} \right) &= \sum_{l=1}^{N_r} p_l \left(\frac{dN_l^{(4)}}{dx_k}, \frac{dN_i^{(1)}}{dx_k} \right) + \left(\frac{dW_4}{dx_k}, \frac{dN_i^{(1)}}{dx_k} \right), \\ F_{2j} &= \sum_{k=1}^3 \left(\frac{\partial}{\partial x_k} (D_1 W_2 + r_1 \hat{T}), \frac{dN_j^{(2)}}{dx_k} \right) + \\ &+ \sum_{k=1}^3 \left(v_1 \left(\frac{\partial \hat{C}_1}{\partial x_k} \right)^2 - \frac{\partial}{\partial x_k} (u\hat{H} - v_2 \hat{C}_2 - \mu\hat{T}) \frac{dW_2}{dx_k}, N_j^{(2)} \right),\end{aligned}$$

$$\begin{aligned}
 \left(\frac{\partial}{\partial x_k} \widehat{T} \frac{dW_2}{dx_k}, N_j^{(2)} \right) &= \sum_{l=1}^{N_v} p_l \left(\frac{dN_l^{(4)}}{dx_k} \frac{dW_2}{dx_k}, N_j^{(2)} \right) + \left(\frac{dW_4}{dx_k} \frac{dW_2}{dx_k}, N_j^{(2)} \right), \\
 \left(\left(\frac{\partial \widehat{C}_1}{\partial x_k} \right)^2, N_j^{(2)} \right) &= \sum_{l_1=1}^{N_v} \sum_{l_2=1}^{N_v} b_{l_1} b_{l_2} \left(\frac{dN_{l_1}^{(2)}}{dx_k} \frac{dN_{l_2}^{(2)}}{dx_k}, N_j^{(2)} \right) + \\
 &+ 2 \left(\frac{\partial}{\partial x_k} \widehat{C}_1 \frac{dW_2}{dx_k}, N_j^{(2)} \right) + \left(\left(\frac{dW_2}{dx_k} \right)^2, N_j^{(2)} \right), \\
 F_{3j} &= \sum_{k=1}^3 \left(\frac{\partial}{\partial x_k} (D_2 W_3 + r_2 \widehat{T}), \frac{dN_j^{(3)}}{dx_k} \right) + \\
 &+ \sum_{k=1}^3 \left(v_2 \left(\frac{\partial \widehat{C}_2}{\partial x_k} \right)^2 - \frac{\partial}{\partial x_k} (u \widehat{H} - v_1 \widehat{C}_1 - \mu \widehat{T}) \frac{dW_3}{dx_k}, N_j^{(3)} \right), \\
 F_{4j} &= \sum_{k=1}^3 \left(\lambda \frac{\partial W_4}{\partial x_k}, \frac{dN_j^{(4)}}{dx_k} \right) + \sum_{k=1}^3 \left(\gamma_1 \left(\frac{\partial \widehat{T}}{\partial x_k} \right)^2 - \frac{\partial}{\partial x_k} (u \widehat{H} - \delta_1 \widehat{C}_1 - \delta_2 \widehat{C}_2) \frac{dW_4}{dx_k}, N_j^{(4)} \right), \\
 M_1 \bar{A}(0) &= \left\{ (1 - W_1), N_i^{(1)} \right\}_i, \quad M_3 \bar{B}(0) = - \left\{ W_2, N_i^{(2)} \right\}_i, \\
 M_4 \bar{Q}(0) &= - \left\{ W_3, N_i^{(3)} \right\}_i, \quad M_5 \bar{P}(0) = - \left\{ T_0 - W_4, N_i^{(4)} \right\}_i.
 \end{aligned} \tag{7}$$

Задачу (6), (7) будемо розв'язувати за допомогою наступної лінеаризованої неявної різницевої схеми [5—7]:

$$\begin{aligned}
 M_1 A_t + L \bar{A} - \alpha_T M_2 P_t + F_1 (\bar{B}, \bar{Q}, \bar{P}) &= 0, \\
 \sigma M_3 B_t + L_1 (A, \bar{Q}, \bar{P}) \bar{B} + F_2 (A, B, \bar{Q}, \bar{P}) &= 0, \\
 \sigma M_4 Q_t + L_2 (A, B, \bar{P}) \bar{Q} + F_3 (A, B, Q, \bar{P}) &= 0, \\
 M_5 P_t + L_3 (A, B, Q) \bar{P} + F_4 (A, B, Q, P) &= 0,
 \end{aligned} \tag{8}$$

де $\{A, B, Q, P\}$ — вектори коефіцієнтів на поточному кроці,
 $\{\bar{A}, \bar{B}, \bar{Q}, \bar{P}\}$ — вектори коефіцієнтів на попередньому кроці,
 $\{A, B, Q, P\}_t = \frac{1}{\tau} (\{A, B, Q, P\} - \{\bar{A}, \bar{B}, \bar{Q}, \bar{P}\})$ — різницеві похідні по

часу.

Введемо наступні позначення:

$$\begin{aligned}
 \mathbf{K} &= \{A, B, Q, P\}, M_{IJ} = \left\{ \left(N_j^{(I)}, N_i^{(J)} \right) \right\}_{i,j}, \bar{M}_{IJ} = \left\{ \sum_{k=1}^3 \left(\frac{dN_j^{(I)}}{dx_k}, \frac{dN_i^{(J)}}{dx_k} \right) \right\}_{i,j}, \\
 S_{LJK} (K_I) &= \left\{ \sum_{k=1}^3 \sum_{l=1}^{N_v} K_{Il} \left(\frac{dN_l^{(I)}}{dx_k}, \frac{dN_i^{(J)}}{dx_k}, N_j^{(K)} \right) \right\}_{i,j}, \\
 W_{LJK} &= \left\{ \sum_{k=1}^3 \left(\frac{dW_I}{dx_k}, \frac{dN_i^{(J)}}{dx_k}, N_j^{(K)} \right) \right\}_{i,j}, \quad (9) \\
 \bar{w}_{IJ} &= \left\{ \sum_{k=1}^3 \left(\frac{dW_I}{dx_k}, \frac{dN_i^{(J)}}{dx_k} \right) \right\}_i, \bar{w}_{LJK}^{(2)} = \left\{ \sum_{k=1}^3 \left(\frac{dW_I}{dx_k}, \frac{dW_J}{dx_k}, N_i^{(K)} \right) \right\}_i, \\
 \bar{a}^{(2)}(C_1) &= \left\{ \sum_{k=1}^3 \left(\frac{\partial \bar{C}_1}{\partial x_k} \right)^2, N_j^{(2)} \right\}_j.
 \end{aligned}$$

Слід зазначити, що $S_{LJK}(K_I)K_J = S_{JK}(K_J)K_I$.

Враховуючи (6) у позначеннях (9) система (8) прийме наступний вигляд:

$$\begin{aligned}
 &\frac{1}{\tau} M_1 A - \frac{1}{\tau} \alpha_T M_2 P = \Phi_0 = \\
 &= \frac{1}{\tau} M_1 \hat{A} - \frac{1}{\tau} \alpha_T M_2 \hat{P} - (\bar{M}_{11} \hat{A} + \bar{w}_{11}) + \kappa_1 (\bar{M}_{21} \hat{B} + \bar{w}_{21}) + \\
 &\quad + \kappa_2 (\bar{M}_{31} \hat{Q} + \bar{w}_{31}) + \theta (\bar{M}_{41} \hat{P} + \bar{w}_{41}), \\
 &\frac{\sigma}{\tau} M_3 B - u (S_{212} (\hat{B}) + W_{212}) A = \Phi_1 = \\
 &= \frac{\sigma}{\tau} M_3 \hat{B} - D_1 (\bar{M}_{22} \hat{B} + \bar{w}_{22}) + u (W_{122} \hat{B} + \bar{w}_{212}^{(2)}) - \nu_1 \bar{a}^{(2)} (\hat{C}_1) - \\
 &\quad - \nu_2 (S_{322} (\hat{Q}) \hat{B} + W_{322} \hat{B} + W_{232} \hat{Q} + \bar{w}_{232}^{(2)}) - \\
 &\quad - \mu (S_{422} (\hat{P}) \hat{B} + W_{422} \hat{B} + W_{242} \hat{P} + \bar{w}_{242}^{(2)}) - r_1 (\bar{M}_{42} \hat{P} + \bar{w}_{42}), \quad (10) \\
 &\frac{\sigma}{\tau} M_4 Q - u (S_{313} (\hat{Q}) + W_{313}) A + \nu_1 (S_{323} (\hat{Q}) + W_{323}) B = \Phi_2 = \\
 &\frac{\sigma}{\tau} M_4 \hat{Q} - D_2 (\bar{M}_{33} \hat{Q} + \bar{w}_{33}) + u (W_{133} \hat{Q} + \bar{w}_{313}^{(2)}) - \nu_1 (W_{233} \hat{Q} + \bar{w}_{323}^{(2)}) - \\
 &\quad - \mu (S_{433} (\hat{P}) \hat{Q} + W_{433} \hat{Q} + W_{343} \hat{P} + \bar{w}_{343}^{(2)}) - r_2 (\bar{M}_{43} \hat{P} + \bar{w}_{43}) - \nu_2 \bar{a}^{(2)} (\hat{C}_2),
 \end{aligned}$$

$$\begin{aligned} & \frac{1}{\tau} M_5 P - u_1 \left(S_{414}(\hat{P}) + W_{414} \right) A + \delta_1 \left(S_{424}(\hat{P}) + W_{424} \right) B + \\ & + \delta_2 \left(S_{434}(\hat{P}) + W_{434} \right) Q = \Phi_3 = \frac{1}{\tau} M_5 \hat{P} - \lambda \left(\bar{M}_{44} \hat{P} + \bar{w}_{44} \right) - \gamma \bar{a}^{(2)}(\hat{T}) + \\ & + u_1 \left(W_{144} \hat{P} + \bar{w}_{414}^{(2)} \right) - \delta_1 \left(W_{244} \hat{P} + \bar{w}_{424}^{(2)} \right) - \delta_2 \left(W_{344} \hat{P} + \bar{w}_{434}^{(2)} \right). \end{aligned}$$

У матричному вигляді систему лінійних алгебраїчних рівнянь (10) можна представити як

$$\begin{aligned} & \bar{A} \bar{K} = \Phi, \\ & \bar{A} = \begin{bmatrix} \bar{A}_{00} & 0 & 0 & \bar{A}_{03} \\ \bar{A}_{10} & \bar{A}_{11} & 0 & 0 \\ \bar{A}_{20} & \bar{A}_{21} & \bar{A}_{22} & 0 \\ \bar{A}_{30} & \bar{A}_{31} & \bar{A}_{32} & \bar{A}_{33} \end{bmatrix} = \\ & = \begin{bmatrix} \frac{1}{\tau} M_1 & 0 & 0 & -\frac{1}{\tau} \alpha_T M_2 \\ -u \left(S_{212}(\hat{B}) + W_{212} \right) & \frac{\sigma}{\tau} M_3 & 0 & 0 \\ -u \left(S_{313}(\hat{Q}) + W_{313} \right) & u_1 \left(S_{323}(\hat{Q}) + W_{323} \right) & \frac{\sigma}{\tau} M_4 & 0 \\ -u_1 \left(S_{414}(\hat{P}) + W_{414} \right) & \delta_1 \left(S_{424}(\hat{P}) + W_{424} \right) & \delta_2 \left(S_{434}(\hat{P}) + W_{434} \right) & \frac{1}{\tau} M_5 \end{bmatrix} \quad (11) \end{aligned}$$

де $\Phi = \{\Phi_0, \Phi_1, \Phi_2, \Phi_3\}$ — вектор правих частин рівнянь (10).

Алгоритм розв'язання системи лінійних алгебраїчних рівнянь. Спеціальний вид матриці \bar{A} системи лінійних рівнянь (11), яка виникає після дискретизації розглядуваної задачі, дозволяє розв'язувати її використовуючи декомпозиційний підхід за наступним ітераційним процесом:

$$\begin{aligned} A^{(k+1)} &= \bar{A}_{00}^{-1} \left(\Phi_0 - \bar{A}_{03} A^{(k)} \right), \\ B^{(k+1)} &= \bar{A}_{11}^{-1} \left(\Phi_1 - \bar{A}_{10} A^{(k+1)} \right), \\ Q^{(k+1)} &= \bar{A}_{22}^{-1} \left(\Phi_2 - \bar{A}_{20} A^{(k+1)} - \bar{A}_{21} B^{(k+1)} \right), \\ P^{(k+1)} &= \bar{A}_{33}^{-1} \left(\Phi_3 - \bar{A}_{30} A^{(k+1)} - \bar{A}_{31} B^{(k+1)} - \bar{A}_{32} Q^{(k+1)} \right). \end{aligned} \quad (12)$$

Оцінимо обчислювальну складність цього алгоритму.

Будемо розв'язувати розріджені СЛАР за допомогою ітераційних методів, зокрема BiCGStab [8], складність яких пропорційна кількості ненульових елементів $V(\bar{A})$ матриці \bar{A} : $\bar{T} = k_s V(\bar{A})$. Тоді складність обчислень по формулі (12), у припущенні, що коефіцієнти складності розв'язання СЛАР рівні, можна оцінити як

$$\begin{aligned} \bar{T}_1 = & k_s \left(V(\bar{A}_{00}) + V(\bar{A}_{11}) + V(\bar{A}_{22}) + V(\bar{A}_{33}) \right) + \\ & + V(\bar{A}_{03}) + V(\bar{A}_{10}) + V(\bar{A}_{20}) + V(\bar{A}_{21}) + V(\bar{A}_{30}) + V(\bar{A}_{31}) + V(\bar{A}_{32}). \end{aligned} \quad (13)$$

У припущенні, що кількість ненульових елементів у підматрицях матриці \bar{A} рівна, оцінка (13) прийме вигляд

$$\bar{T}_1 = \frac{1}{4} k_s V(\bar{A}) + \frac{7}{16} V(\bar{A}),$$

а складність розв'язання системи $\bar{A}\bar{K} = \Phi$ декомпозиційною процедурою (12)

$$\bar{T}_{1f} = \frac{1}{16} k_2 (4k_s + 7) V(\bar{A}), \quad (14)$$

де k_2 — кількість ітерацій процесу (12).

З іншої сторони, складність розв'язання безпосередньо системи (11) складає

$$\bar{T}_2 = k_s V(\bar{A}). \quad (15)$$

Порівнюючи оцінки (14) та (15) доходимо до висновку, що за умови, коли як система $\bar{A}\bar{K} = \Phi$, так і її підсистеми $\bar{A}_{ii}\bar{K}_i = \Phi_i$ розв'язуються за однакову кількість ітерацій, схема (12) матиме меншу складність при $k_2 \leq 3$.

Кількість ітерацій k_2 для розглядуваної задачі залежить від значення коефіцієнту термічного розширення β_T . Обчислювальні експерименти показали, що умова $k_2 \leq 3$ виконується при $\beta_T < 10^{-1}$, в той час, коли значення β_T речовин мають порядок $10^{-4} - 10^{-6}$.

Паралельні алгоритми для кластерних систем. При використанні для розрахунків розподіленої обчислювальної системи, будемо розподіляти матрицю \bar{A} по ній за рядками, що є оптимальною схемою для паралельного обчислення множення матриці на вектор, операції, на виконання якої витрачається більшість часу при розв'язанні СЛАР ітераційними методами.

Позначимо як $\bar{A}^{(i)}$ блок рядків матриці \bar{A} , який зберігається у пам'яті процесу i , а як $\bar{A}^{(i,j)}$ — підматрицю матриці \bar{A} , яка зберіга-

ється у пам'яті процесу i , а діапазон її стовпців відповідає діапазону рядків, що зберігаються у пам'яті процесу j .

Тоді при виконанні операції множення розподіленої розрідженої матриці \bar{A} на вектор y , обміни фрагментами вектора y відбуваються лише між процесами i та j для яких $\bar{A}^{(i,j)} \neq 0 \vee \bar{A}^{(j,i)} \neq 0$.

Обчислення значень елементів матриці \bar{A} та вектора правої частини Φ розпаралелюється за даними й не потребує обмінів.

Обсяг же обмінів між процесами при розв'язанні СЛАР залежить від структури матриці.

Подібна схема розпаралелення породжує задачу оптимального завантаження процесорів, яка виникає через різну кількість ненульових елементів у рядках розрідженої матриці \bar{A} .

Час роботи паралельного алгоритму можна оцінити як

$$T_P = \left(k_3^{(i)} + k_4 \right) \max_i V(\bar{A}^{(i)}) + \max_i N_{ex}^{(i)}, \quad (16)$$

де $k_3^{(i)}$ — коефіцієнт складності обчислення значень елементів СЛАР, k_4 — коефіцієнт складності розв'язання СЛАР, $N_{ex}^{(i)}$ — об'єм обмінів даними, що їх здійснює процес i , P — кількість процесів.

Виникає оптимізаційна задача знаходження такого розподілу рядків $R = (R_1, \dots, R_P)$, який мінімізував би оцінку (16).

У реалізованому паралельному алгоритмі ця задача наближено розв'язується наступним чином:

- задля мінімізації обмінів даними використовується перенумерування рядків й стовпців матриці за алгоритмом Катхіла-МакКі [9];
- окремо розв'язується задача рівномірного завантаження процесів обчислювальними задачами. Починаючи з однакових розмірів блоків $\bar{A}^{(i)}$, блоки послідовно збільшуються чи зменшуються таким чином, щоб $\left(k_3^{(i)} + k_4 \right) V(\bar{A}^{(i)}) = \frac{1}{P} \left(T_P - \max_i N_{ex}^{(i)} \right)$. При цьому значення коефіцієнтів $k_3^{(i)}$ та k_4 оцінюються, а $N_{ex}^{(i)}$ — вимірюється в процесі роботи алгоритму.

Окрім цього, в межах одного вузла кластерної системи, який містить багатоядерні процесори, обчислення можна організувати використовуючи багатопоточне розпаралелення. При цьому один процес займає весь вузол системи й на кожному з ядер виконується один з його потоків. Різниця між такою схемою й схемою, коли кожен процес розподіленої програми виконується на одному ядрі, полягає в тому, що не знижуючи швидкодії розрахункової частини алгоритму, обміни даними виконують-

ся між меншим числом процесів, що є ефективним у випадку, коли обсяг обмінів збільшується при збільшенні числа процесів.

У випадку розглядуваної задачі, залежність обсягу обмінів від числа процесів визначається структурою матриці A , яка у свої чергу залежить від форми області моделювання та її дискретизації на скінчені елементи.

Паралельний алгоритм для графічних процесорів (GPU). Графічні процесори є високопродуктивними обчислювальними системами, які проте мають певні архітектурні особливості, що ускладнює розробку ефективних алгоритмів та програм для них.

Обчислення на GPU є масово-паралельними, при чому потоки обчислень об'єднуються у блоки й засоби синхронізації існують лише в межах блоків. С точки зору доступу до пам'яті, у сучасних GPU виділяють повільну глобальну пам'ять, до якої мають доступ всі потоки, й більш швидко локальну, яка розділяється між потоками в межах блоку. В додачу до цього, декілька операцій доступу до пам'яті при виконанні певних умов можуть бути об'єднані в одну, що суттєво пришвидшує виконання програми.

Враховуючи ці особливості, побудову алгоритмів для GPU можна розбити на такі етапи:

- виділення незалежних блоків обчислень;
- оптимізація доступу до пам'яті;
- оптимізація відношення кількості арифметичних операцій та операцій доступу до пам'яті (чим більше це відношення, тим ефективніше буде виконуватись програма);
- вибір оптимальної кількості потоків та групування потоків у блоки.

Виділяють [10] декілька варіантів алгоритмів формування СЛАУ у методі скінчених елементів відштовхуючись від того, обчислення стосовно яких елементів проводиться кожним потоком:

- формування по ненульовим елементам матриці;
- формування по рядках матриці;
- формування по окремим скінченим елементам.

У перших двох випадках обчислення є апіорі незалежними, у третьому — скінчені елементи мають бути розбиті на підмножини, в межах яких елементи не мають спільних вершин. Стандартним підходом до проведення такого розбиття є фарбування скінченоелементної сітки, як графу залежностей між елементами. В кожному з цих підходів існує можливість блокування потоків й використання локальної пам'яті для доступу до даних, які необхідні всім потокам з блоку. В усіх випадках, для доступу до елементів розрідженої матриці ефективно використовувати наперед прораховані індекси елементів

замість дуже неефективної при реалізації на GPU операції їх пошуку в процесі розрахунків.

Питання оптимізації доступу до пам'яті GPU є важливим при реалізації алгоритмів розв'язання СЛАР й, в першу чергу, операції множення розрідженої матриці на вектор. Оскільки алгоритм множення безпосередньо залежить від формату зберігання матриці, задача оптимізації доступу до пам'яті зводиться до задачі знаходження такого формату зберігання, якій дав би можливість якнаймога в більшому об'ємі використовувати спільну між потоками пам'ять й об'єднувати операції доступу до неї. З найчастіше використовуваних форматів можна виділити [11] покоординатний, стиснутий рядковий (compressed row storage, CSR), діагональний, блочний стиснутий рядковий (block compressed row storage, BCSR) та гібридні схеми. Слід зауважити, що наразі існують спеціальні бібліотеки розроблені, зокрема, такими компаніями як NVIDIA та IBM, у яких реалізовані алгоритми для GPU множення матриці на вектор.

У випадку розглядуваної задачі на вибір підходів при розробці GPU-алгоритмів суттєво впливають два фактори:

- нелінійність задачі, яка ускладнює обчислення та збільшує можливість повторного використання даних;
- великий коефіцієнт наповненості матриць СЛАР.

Перший фактор, за умови того, що всі скінчені елементи мають однакову кількість вершин, зумовив використання схеми формування СЛАР у якій кожен блок потоків оброблює один скінчений елемент, при чому кожен потік змінює елементи СЛАР, пов'язані лише з однією вершиною. При цьому виконується умова незалежності обчислень й з'являється можливість сумісної роботи потоків блоку через локальну пам'ять з даними, пов'язаними з конкретним скінченим елементом.

Другий фактор зумовив вибір BCSR формату збереження розрідженої матриці, ефективність використання якого зростає при збільшенні заповненості матриці. Оптимальний розмір блоку на основі даних обчислювальних експериментів був вибраний таким, що дорівнює 16.

У випадку розглядуваної задачі, використання GPU у якості сопроцесора без суттєвих модифікацій комбінується з алгоритмами для кластерних систем, що дозволяє використовувати для розв'язання цих задач GPU-кластери.

Алгоритми були реалізовані з використанням технології OpenCL й тестувались на кластері СКІТ-3GPU Інституту кібернетики ім. В.М. Глушкова на задачі з 29791-елементною сіткою, матрицею СЛАР розмірністю 119164x 119164 з 8016008 ненульових елементів. Отримані дані щодо швидкодії алгоритмів подані у табл. 1.

Час роботи алгоритмів, мс

	Загальний час	Час формування СЛАР	Час розв'язання СЛАР
1 CPU	4220	3560	660
3 CPU	2260	1480	780
1 GPU	1260	1020	240
3 GPU	1090	870	220

Як видно з отриманих даних, використання GPU дозволяє у 2—3 рази пришвидшити розрахунки в додачу до пришвидшення, що досягається використанням кластерних систем.

Чисельні експерименти. Стосовно алгоритмів для кластерних систем була проведена серія чисельних експериментів у яких вимірювався час роботи алгоритмів в цілому та їх частин в залежності від розмірності задачі, кількості та конфігурації задіяних ресурсів.

У якості тестової розглядалась тривимірна задача моделювання процесу консолідації ґрунтового масиву, розміщеного на проникній основі. Значення параметрів задачі (1), (2) брались наступними:

- граничні та початкові умови:
 $C_{10} = 350 \text{ г/л}$, $C_{11} = 240 \text{ г/л}$, $T_0 = 10\text{C}$, $T_1 = 25\text{C}$, $H_1 = 5 \text{ м}$;
- характеристики ґрунтового масиву:
 $l = 1\text{м}$, $\sigma = 0.655$, $k = 0.001\text{м/добу}$, $c_v = 0.03232\text{м}^2/\text{добу}$;
- коефіцієнти дифузії та осмосу:
 $D_1 = 0.02\text{м}^2/\text{добу}$, $D_2 = 0.04\text{м}^2/\text{добу}$, $v_1 = 2 \cdot 10^{-5} \text{ м}^5/\text{кг} \cdot \text{добу}$,
 $v_2 = 3 \cdot 10^{-5} \text{ м}^5/\text{кг} \cdot \text{добу}$, $\mu = 10^{-5} \text{ м}^2/\text{град} \cdot \text{добу}$; (17)
- термічні коефіцієнти:
 $C_T = 2 \cdot 10^6 \text{ Дж/кг} \cdot \text{м}^3 \cdot \text{град}$, $C_p = 0.8 \cdot 10^3 \text{ Дж/кг} \cdot \text{град}$,
 $\rho = 1100 \text{ кг/м}^3$, $\lambda = 69120 \text{ Дж/м} \cdot \text{град} \cdot \text{добу}$,
 $\beta_T = 0.5 \cdot 10^{-5} \text{ K}^{-1}$, $D_T = \frac{\lambda}{\rho C_p} = 0.078 \text{ м}^2/\text{добу}$.

Дані щодо швидкодії для задачі з сіткою розмірністю 77310 вузлів (розмірність матриці — 309240x309240) наведені у табл.2, а для задачі з сіткою розмірністю 220210 вузлів (розмірність матриці — 880488x880488) — у табл. 3. Розв'язок задачі на прямій $x=z=0$ зображений на рис.1а, аналогічний розв'язок задачі, що не враховує осмотичні явища та терморозширення, зображений на рис. 1 б.

Таблиця 2

*Час обчислення одного кроку для задачі з сіткою
розмірністю 77310 вузлів, мс*

Кількість потоків	Кількість процесів	Кількість задіяних вузлів	Безпосереднє розв'язання СЛАР			Декомпозиційний алгоритм		
			Загальний час	Час обчислення значень елементів СЛАР	Час розв'язання СЛАР	Загальний час	Час обчислення значень елементів СЛАР	Час розв'язання СЛАР
1	1	1	30420	18020	12400	24610	18300	6310
8	2	2	8770	5110	3660	9040	5810	3230
28	7	7	4380	2220	2160	3400	1630	1770
40	10	10	3960	1640	2320	3460	1530	1930
8	8	2	13150	6070	7080	6510	3520	2990
32	32	8	4270	1980	2290	3280	1240	2040
40	40	10	3940	1780	2160	3370	1060	2310

Таблиця 3.

*Час обчислення одного кроку для задачі з сіткою
розмірністю 220210 вузлів, мс*

Кількість потоків	Кількість процесів	Кількість задіяних вузлів	Безпосереднє розв'язання СЛАР			Декомпозиційний алгоритм		
			Загальний час	Час обчислення значень елементів СЛАР	Час розв'язання СЛАР	Загальний час	Час обчислення значень елементів СЛАР	Час розв'язання СЛАР
1	1	1	74500	39050	35450	79480	54590	24890
8	2	2	24010	10980	13030	21380	11630	9750
28	7	7	13750	5880	7870	8470	3940	4530
40	10	10	12610	5070	7540	8370	3210	5160
8	8	2	27460	8440	19020	18250	9170	9080
12	12	3	19250	6890	12360	17140	7620	9520

Розв'язувалась також задача моделювання процесу фільтраційної консолідації ґрунтового масиву розміром 40х40 м, за наявності двох ставків, заповнених розчинами різних солей, й часткового непроникного бар'єру між ними (рис. 2). Моделювання процесу проводилось на період у 12 місяців зі значеннями параметрів (17). Ізоповітряні отриманої функції розподілу температури зображені на рис. 3.

Ізолінії функції надлишкових напорів на розрізі $z=12$ м зображені на рис. 4 а., а сумарної функції засоленості на рис. 4 б.

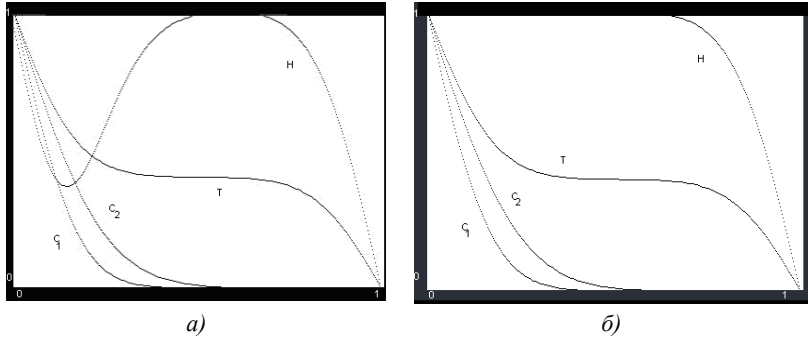


Рис. 1. Розв'язок задачі на прямій $x=z=0$

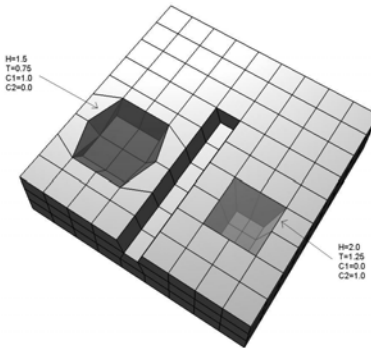


Рис. 2. Область моделювання

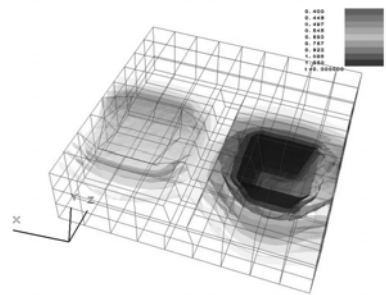
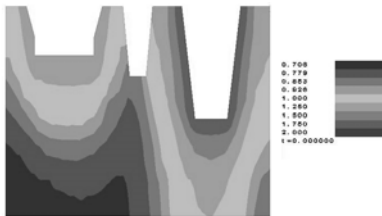
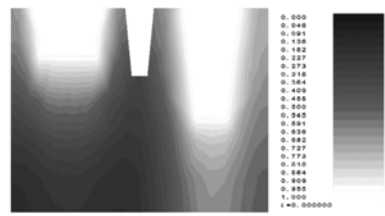


Рис. 3. Ізоповерхні функції розподілу температури



а) надлишкових напорів



б) сумарної засоленості

Рис. 4. Ізолінії функцій

В цілому, експерименти підтвердили ефективність застосування декомпозиційного алгоритму розв'язання СІАР. Так, для задачі меншої розмірності пришвидшення при цьому склало $\sim 16\%$, а для задачі більшої розмірності — $\sim 45\%$.

Пришвидшення ж від використання багатопоточності спостерігалось лише на невеликій кількості задіяних вузлів, зокрема на двох чотирипроцесорних вузлах пришвидшення склало $\sim 12\%$ для задачі високої розмірності й $\sim 33\%$ для задачі низької розмірності. При задіяні трьох й більше вузлів пришвидшення від використання багатопоточності не спостерігалось.

Падіння прискорення при зростанні розмірності тут може бути пояснено зменшенням при цьому відношення часу, що витрачається на обміни, до часу, що витрачається на обчислення. Це призводить до ефекту, що спостерігався, так як використання багатопоточності разом з розпаралеленням для систем з розподіленою пам'яттю для розв'язання розглядуваної задачі потенційно зменшує саме обсяг обмінів при незмінному обсязі обчислень. По тій самій причині пришвидшення при використанні декомпозиційного алгоритму зростає при збільшенні розмірності, оскільки в цій ситуації зменшується як час, що витрачається на обміни, так і обсяг обчислень.

Висновки. Для задачі моделювання процесу фільтраційної консолідації ґрунту під дією бікомпонентного розчину були розглянуті обчислювальні аспекти розв'язання у випадку тривимірної постановки, зокрема пов'язані з розпаралеленням. На предмет досягнення найвищої швидкодії досліджувались чотири паралельні алгоритми для систем з розподіленою пам'яттю:

- з безпосереднім розв'язанням СЛАР;
- з використанням декомпозиційного алгоритму розв'язання СЛАР;
- з безпосереднім розв'язанням СЛАР та використанням багатопоточності;
- з використанням декомпозиційного алгоритму розв'язання СЛАР та багатопоточності.

При цьому в усіх варіантах для розв'язання розріджених СЛАР використовувався алгоритм BiCGStab, та застосовувалась схема оптимізації завантаження процесорів.

Як теоретичні оцінки, так і дані обчислювальних експериментів показали, що

- декомпозиційний алгоритм розв'язання СЛАР є швидшим за безпосереднє розв'язання СЛАР при невеликих значеннях коефіцієнту термічного розширення;
- використання багатопоточності дозволяє суттєво (на $\sim 33\%$) пришвидшити розрахунки лише для задач невеликої розмірності на невеликій (до двох чотирипроцесорних) кількості задіяних вузлів.

Окрім цього було досліджена можливість використання GPU у якості сопроцесора при розв'язанні розглядуваних задач. Були розроблені відповідні алгоритми та проведені обчислювальні експерименти, які показали 2—3-кратне прискорення в порівнянні з CPU-алгоритмами.

Список використаних джерел:

1. Бомба А. Я. Нелінійні математичні моделі процесів геогідродинаміки / А. Я. Бомба, В. М. Булавацький, В. В. Скопечкий. — К. : Наук. думка, 2007. — 292 с.
2. Власюк А. П. Математичне моделювання консолідації ґрунтів в процесі фільтрації сольових розчинів / А. П. Власюк, П. М. Мартинюк. — Рівне : Вид-во УДУВГП, 2004. — 211 с.
3. Власюк А. П. Математичне моделювання консолідації ґрунтів при фільтрації сольових розчинів в неізотермічних умовах / А. П. Власюк, П. М. Мартинюк. — Рівне : Вид-во НУВГП, 2008. — 416 с.
4. Булавацький В. М. Математическое моделирование динамики консолидационного процесса насыщенной бинарным солевым раствором пористой среды / В. М. Булавацький // Компьютерная математика. — 2008. — № 2. — С. 3—12.
5. Булавацький В. М. Математическое моделирование динамики одного неізотермічного консолидационного процесса / В. М. Булавацький, В. В. Скопечкий // Математичне та комп'ютерне моделювання. Серія: Фізико-математичні науки : зб. наук. пр. — Кам'янець-Подільський : Кам'янець-Поділ. нац. ун-т, 2010. — Вип. 3. — С. 28—36.
6. Булавацький В. М. Об одной неізотерміческой консолидационной математической модели геоинформатики / В. М. Булавацький, В. В. Скопечкий // Пробл. упр. и информатики. — 2010. — № 6. — С. 35—45.
7. Самарский А. А. Вычислительная теплопередача / А. А. Самарский, П. Н. Вабищевич. — М. : Едиториал УРСС, 2003. — 784 с.
8. Saad Y. Iterative methods for sparse linear systems, 2 edition / Y. Saad. — Society for Industrial and Applied Mathematics, 2003. — 528 p.
9. Cuthill E. Reducing the bandwidth of sparse symmetric matrices / E. Cuthill, J. McKee // In Proc. 24th Nat. Conf. ACM. — 1969. — P. 157—172.
10. Cecka C. Assembly of finite element methods on graphics processors / C. Cecka, A. J. Lew, E. Darve // International Journal for Numerical Methods in Engineering. — 2011. — P. 640—669.
11. Abhijeet Gaikwad, Ioane Muni Toke GPU based Sparse Grid Technique for Solving Multidimensional Options Pricing PDEs // Supercomputing'09, Proceedings of the 2nd Workshop on High-Performance Computational Finance — SC09-WHPCF09, ACM Press (2009)

Problem of modeling filtration consolidation processes in soils under the influence of bi-component solution has been considered in the paper. One-dimensional mathematical model of such processes has been extended to three-dimensional case and a set of parallel algorithms for solving problems about it have been proposed, in particular, multithreaded algorithms, algorithms for cluster systems and graphical processors (GPU).

Key words: *filtration consolidation, bi-component solutions, mathematical modeling, parallel algorithms.*

Отримано: 17.05.2011