УДК 004.4

Синявський О. Л., Хоменко О. І.

ПАРАЛЕЛЬНИЙ АЛГОРИТМ РОЗВ'ЯЗАННЯ КРАЙОВИХ ЗАДАЧ ДЛЯ СИСТЕМ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ ВИЩИХ ПОРЯДКІВ ЗА МЕТОДОМ МОНТЕ-КАРЛО

Запропоновано паралельний алгоритм розв'язання першої крайової задачі теорії пружності. Ця задача має місце у випадку пружного тіла, затисненого на частині поверхні або замкненого в жорстку обойму. Для побудови алгоритму методу Монте-Карло було узагальнено теорему про середнє значення, відому з теорії гармонічних функцій. Так, у нашому випадку рівнянням Ляме відповідає інтегральне співвідношення між вектором переміщень в центрі кулі та переміщеннями на її поверхні [1]. Воно отримане на основі розв'язку задачі про деформацію пружної кулі з затисненою границею при дії на неї зосередженої одиничної сили в центрі. На цьому співвідношенні побудовано розв'язок задачі у вигляді континуального інтеграла, який оцінюється за допомогою процесу блукання сферами.

1. Аналітичне формулювання задачі

Нехай $G \in N^3$ - деяка обмежена скінченно зв'язана 3-вимірна область евклідового простору. Позначимо символом Г границю даної області. Точка у просторі позначатиметься як *x,* де *x* має координати (x_1, x_2, x_3).

Стан у деякій точці *x* області *G* за відсутності масових сил описується класичним статичним рівнянням Ляме [2]:

$$\Delta u(x) + \frac{1}{1-2\nu}\, grad\, div\, u(x) = 0, \quad x \in G, \qquad (1)$$

де $u(x) = (u_1(x_1, x_2, x_3), u_2(x_1, x_2, x_3), u_3(x_1, x_2, x_3))$ – вектор переміщень, чиїми компонентами є регулярні функції дійсних значень;

ν - коефіцієнт Пуассона, яким характеризується деформований матеріал.

Перша гранична задача теорії пружності для рівняння Ляме полягає у знаходженні вектор-функції $u \in C^2(G) \cap C(\bar{G})$, що задовольняє граничну умову

$$u(x) = g(x), x \in \Gamma, \qquad (2)$$

де $g \in C(G)$ — задана вектор-функція, значеннями якої є переміщення на границях області.

2. Інтегральне подання розв'язку задачі

Саме інтегральне подання вектора переміщень точки пружного простору через переміщення точок сфери, описаної з даної точки як з центру, лежить в основі ймовірнісної схеми розв'язання просторової задачі теорії пружності методом статистичних випробувань.

Нехай, як і раніше,

$u = (u_1, u_2, u_3)$ – вектор переміщень;

$\sigma_n = (\sigma_{n1}, \sigma_{n2}, \sigma_{n3})$ – напруження на майданчику, дотичному до поверхні сфери S радіуса R;

V – область, обмежена сферою S;

$X = (X_1, X_2, X_3)$ – вектор об'ємних сил.

Досліджуваний напружено-деформований стан за відсутності масових сил визначається граничними умовами задачі (2) та рівнянням Ляме (1). Назвемо цей стан основним. Для отримання інтегрального подання вектора переміщень уведемо до розгляду допоміжний напружено-деформований стан. У ролі останнього візьмемо напружено-деформований стан кулі з закріпленою поверхнею, який створюється зосередженою одиничною силою, прикладеною в центрі кулі. Позначимо зірочкою величини, що стосуються допоміжного стану.

Тоді можна записати інтегральну рівність, що виражає принцип взаємності робіт [3]:

$$\sum_{i=1}^{3} \left| \int_S (\delta^*_{ni} u_i - \delta_{ni} u^*_i) ds + \int_V (X^*_i u_i - X_i u^*_i) dv \right| = 0. \qquad (3)$$

Вирази для компонент переміщень та напружень усередині кулі відрізняються від відомого фундаментального розв'язку рівнянь просторової задачі теорії пружності певними доданками, що забезпечують відсутність переміщень на поверхні кулі. При дії в центрі кулі одиничної зосередженої сили вздовж координатної осі x_j *(j = 1, 2, 3)* на затисненій поверхні кулі виникають напруження:

$$\sigma^{(j)}_{nj} = \frac{3}{8-12\nu}\left(5\frac{x_j^2}{R^2} + 1 - 4\nu\right). \qquad (4)$$

З рівності (3), $\sigma^{(j)}_{ni} = \frac{3}{8-12\nu}\left(5\frac{x_i x_j}{R^2}\right)$, $(i \neq j)$; прийнявши $X^{(j)}_i = \sigma_{nj}\sigma(x)$, отримаємо:

$$u(x) = \frac{1}{4\pi R^2} \int_{S_x} A(x, y) u(y) dS_y, \qquad (5)$$

де S_x – сфера з центром у точці x (x_1, x_2, x_3);

$y = y(y_1, y_2, y_3)$ – точка на сфері S_x;

елементи матриці $A = a_{ij}$ $(i = 1, 2, 3)$ мають вигляд:

$$a_{ii} = \frac{3}{8-12\nu}\left[5\frac{(x_i - y_i)^2}{R^2} + 1 - 4\nu\right]; \qquad (6)$$

$$a_{ij} = \frac{3}{8-12\nu}\, 5\frac{(x_i - y_i)(x_j - y_j)}{R^2}, (i \neq j).$$

Таким чином, маємо інтегральне подання, що пов'язує переміщення в центрі кулі з переміщеннями на її поверхні. Воно виражає теорему про середнє для просторової задачі теорії пружності. Матриця A є функцією Гріна першої крайової задачі для кулі.

3. Алгоритм знаходження розв'язку задачі

3.1. Визначення сферичного процесу

Перед побудовою алгоритму знаходження розв'язку поставленої задачі доцільно окреслити поняття сферичного процесу в тривимірному просторі [4].

Означення 1.

Маючи певну область $G \in N^3$ з границею Γ і деяку точку $x \in G \cup \Gamma$, називатимемо S_x максимальною тривимірною сферою з центром x та радіусом R, якщо $R = inf_{y \in \Gamma} \| y - x \|$, де S_x порожня, якщо $x \in \Gamma$.

Означення 2.

Маючи певну область $G \in N^3$ з границею Γ і деяку точку $x \in G \cup \Gamma$, називатимемо $\Phi(x)$ сферичним процесом, що починається з точки x, якщо

– $\Phi(x) = \{K(x, \varphi), 0 \le x \le 1\}$; тобто $\Phi(x)$ є множиною всіх послідовностей $K(x, \varphi)$ точок тривимірного простору, де

– кожне значення φ задає послідовність точок $K(x, \varphi) = \{P_{i+1}(x, \varphi), i = 0, 1, ...\}$, згенерованих таким чином:

– навколо точки $P_0(x, \varphi) = x$, як із центру, описати максимальну сферу S_{P_0};

– обрати точку $P_1(x, \varphi)$ з рівномірного розподілу на сфері S_{P_0};

– точку $P_{i+1}(x, \varphi)$ визначити рекурсивно за точкою $P_i(x, \varphi)$ та сферою S_{P_i} таким же чином, як точка $P_1(x, \varphi)$ була обрана за $P_0(x, \varphi)$.

Таке визначення сферичного процесу фактично задає ймовірнісний простір, чиїми елементарними подіями є послідовності напрямів, що обираються випадково з рівномірним розподілом, причому так, що вони є незалежними між собою. Ймовірності задаються таким чином. Точка Q_j на одиничній сфері F ставиться у відповідність напрямку j-того переходу, тобто переходу від точки $P_{j-1}(x, \varphi)$ до точки $P_j(x, \varphi)$. Якщо F_j є деякою вимірною за Лебегом підмножиною F міри $m(F_j)$, то обирання напрямку j-того переходу рівномірним випадковим чином еквівалентне заданню $\Pr\{Q_j \in F_j\} = m(F_j) / m(F)$. Обрання взаємонезалежних напрямків еквівалентне заданню $\Pr\{Q_1 \in F_1, Q_2 \in F_2, ..., Q_s \in F_s\} = \prod_{j=1}^{s} \Pr\{Q_j \in F_j\}$ для кожного s.

3.2. Побудова алгоритму

Користуючись означенням сферичного процесу та рівністю (5), яка виражає теорему про середнє у просторовій теорії пружності, побудуємо розв'язок крайової задачі для внутрішньої точки Q області G, коли на її границі Г задано переміщення.

Позначимо через Γ_ε підмножину точок області G, що віддалені від Г не більше ніж на ε. Переміщення точок ε-околу границі приблизно дорівнюють переміщенням найближчих точок границі. Оскільки максимальна сфера дотикається до границі тіла, як правило, лише в кількох точках, то ймовірність переходу з центра сфери в ці точки близька до нуля. Саме тому для побудови алгоритму блукання сферами і вводиться ϵ-окіл границі.

Обрання ε-околу залежить від градієнта функції, заданої на Г, і бажаної точності розв'язку. Отримати аналітичні залежності між цими величинами для довільної просторової задачі дуже складно. Тому на практиці значення ε обирають у частинах основного геометричного розміру досліджуваної області.

Оточимо точку Q максимальною сферою S_Q. Коли б значення $u(Q_i)$ були відомі для всіх $Q_i \in S_Q$, то обчислення $u(Q)$ звелося би до інтегрування відомої функції по сфері S_Q. Метод обчислення визначених інтегралів за допомогою статистичних випробувань реалізується в даному випадку у вигляді наступної процедури. Генеруються N випадкових точок Q_i^0, рівномірно розподілених по сфері S_Q. Вектор $u(Q)$, відповідно до (5), обчислюється як середнє від добутків $A(Q, Q_i^0) u(Q_i^0)$

$$u(Q) \approx \frac{1}{N} \sum_{i=1}^{N} \xi_i, \text{ де } \xi_i = A(Q, Q_i^0)\, u(Q_i^0) \qquad (7)$$

Насправді, в сумі (7) визначеними будуть лише доданки для $Q_i^0 \in S_Q^\varepsilon = S_Q \cap \Gamma_\varepsilon$. Тому якщо $Q_i^0 \in \Gamma_\varepsilon$, то значення випадкового вектора $\xi_i = A(Q, Q_i^0)\, u(Q_i^0)$ і буде наближеною оцінкою i-того доданка ряду (7). Якщо ж випадкова точка $Q_i^0 \notin \Gamma_\varepsilon$, то з центром у цій точці проводиться нова максимальна сфера $S_{Q_i^0}$ і генерується випад-

кова точка Q_i^1 з рівномірного розподілу по сфері $S_{Q_i^0}$. Якщо $Q_i^1 \in \Gamma_\varepsilon$, то відповідним i-тим доданком суми (7) буде добуток $\xi_i = A(Q, Q_i^0) A(Q_i^0, Q_i^1) u(Q_i^1)$, а для точки $Q_i^1 \notin \Gamma_\varepsilon$ процес продовжиться.

Траєкторіям, які на k-тому кроці потраплять на ε-окіл границі, відповідає функціонал $\xi_i = A(Q, Q_i^0) A(Q_i^0, Q_i^1) \ldots A(Q_i^{k-2}, Q_i^{k-1}) u(Q_i^{k-1})$.

У загальному випадку, траєкторія блукання може мати довільну довжину, але зі ймовірністю, що дорівнює одиниці, вона закінчується в Γ_ε *[4].* При побудові алгоритмів типу блукання сферами довжину траєкторії обмежують деяким числом k. Це число варто вибрати достатньо великим, щоб із імовірністю, близькою до одиниці, врахувати внески всіх траєкторій, які є можливими для даної області.

3.3. Схема послідовного алгоритму

Побудований вище алгоритм можна схематично подати у вигляді такої послідовності кроків:

1. ОТРИМАТИ на вхід цільову точку Q.
2. Поки не досягнуто ДОСТАТНЬОЇ кількості N траєкторій, не довших ніж k,
   - *a*. Викликати СФЕРИЧНИЙ ПРОЦЕС із початком у точці Q й отримати результат R;
   - *b*. Якщо R ПОРОЖНІЙ, перейти на КРОК *2;*
   - *c*. ДОДАТИ R до U;
   - *d*. ЗБІЛЬШИТИ лічильник вдалих траєкторій.
3. ПОВЕРНУТИ $\frac{U}{N}$.

СФЕРИЧНИЙ ПРОЦЕС складається з такої послідовності кроків:

1. ОТРИМАТИ на вхід точку Q початку блукань.
2. Якщо ПЕРЕВИЩЕНО максимальну довжину траєкторії, ПОВЕРНУТИ порожній результат.
3. ЗБІЛЬШИТИ на *1* довжину траєкторії.
4. Якщо дана точка БЛИЗЬКА до границі $(Q \in \Gamma_\varepsilon)$, то
   - *a*. ЗНАЙТИ точку P на границі, найближчу до даної;
   - *b*. ПОВЕРНУТИ як результат значення заданої на границі функції $g(P)$.
5. Якщо дана точка ДАЛЕКА від границі $(Q \notin \Gamma_\varepsilon)$, то
   - *a*. Провести навколо Q максимальну сферу S_Q;
   - *b*. ЗГЕНЕРУВАТИ випадкову точку T на S_Q;
   - *c*. РЕКУРСИВНО викликати КРОК *1* відносно точки T і отримати значення R;
   - *d*. ПОВЕРНУТИ як результат $A(Q, T) R$.

4. Паралельна версія алгоритму

Точність роботи побудованого вище алгоритму напряму залежить від числа N. Воно задає кількість випадкових блукань, тобто кількість незалежних експериментів, які необхідно провести. Результат кожного з таких експериментів дає свій внесок в оцінку розв'язку задачі. Досягнення необхідної точності розв'язків вимагає надання числу N достатньо великого значення. З іншого боку, кожне з N блукань передбачає виконання сотень арифметичних операцій, оскільки на всіх кроках випадкової траєкторії потрібно знаходити відстань від точки до границі, генерувати випадкову точку на сфері, множити матрицю на вектор. Поданий вище алгоритм виконує усі N блукань послідовно і витрачає на це значну кількість часу.

Застосування технологій паралельного програмування та сучасних комп'ютерних потужностей дасть змогу істотно підвищити швидкість і ефективність роботи запропонованого алгоритму. Алгоритм блукання сферами методу Монте-Карло має максимальні можливості для розпаралелювання. Усі N блукань не залежать одне від одного, а тому існує можливість виконувати їх одночасно на кількох процесорах.

У випадку поданого вище алгоритму блукання сферами найбільш доцільно застосувати функціональну стратегію декомпозиції задачі. На перший погляд, кожне з N блукань, що реалізує сферичний процес, не залежить від іншого блукання, й усю кількість експериментів можна рівномірно розподілити між наявними процесорами (позначимо їх кількість як P), застосувавши тим самим декомпозицію за даними. Але, з іншого боку, кінцевий результат роботи алгоритму залежить від успішного завершення усіх N експериментів і не буде готовим доти, поки кожен з процесорів не виконає свою частину завдання. В умовах, коли процесори, задіяні в даних обчисленнях, не є однаковими за потужністю або мають різний рівень завантаженості, декомпозиція за даними буде спричиняти значні затримки та неефективність алгоритму. Саме тому для декомпозиції доцільно застосувати функціональний підхід.

Таким чином, можна розбити N експериментів на групи по n у кожній. Тоді завдання для кожного процесора будуть надходити порціями розміру n і замінятимуться новими мірою їх виконання доти, поки сумарно не буде проведено N експериментів. P процесорів можна розділити на дві частини: 1 керівний і P-1 виконавчий. Керівний процесор управлятиме роботою виконавчих, навантажуватиме їх завданнями і виконуватиме узагальнення отриманих від них результатів.

Така стратегія декомпозиції дозволить оптимально збалансувати навантаження на процесори та досягти оптимальної ефективності виконуваних ними обчислень.

4.1. Схема паралельного алгоритму

Пропонуємо виділити два окремі потоки управління в паралельному алгоритмі: потік керівного процесора та потік виконавчих процесорів. Їхні схеми подано нижче.

I. Керівний потік

1. ОТРИМАТИ на вхід цільову точку Q.
2. РОЗДАТИ виконавчим процесам точку Q.
3. РОЗДАТИ виконавцям перші порції завдань по n експериментів.
4. ПОКИ не виконано N експериментів
   a. ОТРИМАТИ від виконавця W проміжний результат T;
   b. ДАТИ W НАСТУПНУ порцію з n експериментів;
   c. ВРАХУВАТИ проміжний результат $(U = U + T)$;
   d. ЗБІЛЬШИТИ кількість ВИКОНАНИХ експериментів на n.
5. ЗУПИНИТИ виконавчі процеси, роздавши їм порожні завдання.
6. ПОВЕРНУТИ як результат $\frac{U}{N}$.

II. Виконавчий потік

1. ОТРИМАТИ від керівного процесу ЦІЛЬОВУ точку Q.
2. ОТРИМАТИ завдання t від керівного процесу.
3. Якщо t ПОРОЖНЄ, ЗАВЕРШИТИ роботу.
4. t разів виконати СФЕРИЧНИЙ ПРОЦЕС (2.3.3) з точки Q, додаючи його результат до R.
5. ПОВЕРНУТИ керівному процесу проміжний результат R.
6. ПЕРЕЙТИ на крок *2*.

5. Аналіз часової складності

Аналіз часової складності T послідовного алгоритму подано в термінах кількості кроків сферичного процесу, оскільки обсяг операцій t, які відбуваються на кожному кроці, є фіксованим і не впливає на асимптотичні оцінки.

5.1. Залежність від розміру ε-околу границі

Кількість T_f кроків траєкторії з даної точки до границі області є випадковою функцією відстані від даної точки до границі та розміру її ε-околу. У роботі [5] показано, що сферичний процес, застосований нами, має таку часову складність:

$$T_f(\varepsilon) = O(|\ln \varepsilon|). \qquad (8)$$

Звідси випливає, що

$$T(\varepsilon) = O(T_f t) = O(|\ln \varepsilon|\ t) = O(|\ln \varepsilon|). \qquad (9)$$

Це означає, що загальний час виконання алгоритму пропорційний величині $|\ln \varepsilon|$ і прямує до нескінченності при наближенні ε до нуля:

$$\lim_{\varepsilon \to 0} T(\varepsilon) = \lim_{\varepsilon \to 0} |\ln \varepsilon| = \infty. \qquad (10)$$

5.2. Залежність від кількості експериментів

Збільшення кількості експериментів N істотно поліпшує точність результатів. Але варто з'ясувати, яким чином впливає параметр N на складність алгоритму. Позначимо через t часову складність проведення одного експерименту, тобто розігрування одного випадкового блукання за правилами сферичного процесу. Як видно зі структури алгоритму, часова складність одного блукання не залежить від загальної кількості блукань, тобто

$$t\,(\mathrm{N}) = \mathrm{const} \qquad (11)$$

Тоді, подавши часову складність всього алгоритму у вигляді

$$T\,(N) = O\,(N\,t\,(N)) = O\,(N), \qquad (12)$$

приходимо до очевидного висновку, що його складність є лінійною функцією від кількості експериментів.

5.3. Ефективність розпаралелювання

Варто зауважити, що алгоритм блукання сферами методу Монте-Карло є алгоритмом мінімальної зв'язності (зв'язність алгоритму - об'єм даних, що передаються від n-го кроку алгоритму до $(n + 1)$-го кроку). Саме завдяки цьому ефективність від його розпаралелювання є дуже великою порівняно з іншими методами розв'язання подібних задач.

Позначимо через P кількість процесорів, задіяних у розрахунках за алгоритмом, через N - розмір задачі. Ідеальна (а отже, й недосяжна) ефективність від розпаралелювання певного алгоритму зі складністю послідовної версії $T(N)$ на P процесорах дає складність $T(N,P) = \frac{T(N)}{P}$. Такої ефективності досягти неможливо, оскільки в паралельній реалізації додається час $S\,(N, P)$, затрачений на обмін даними між процесорами.

У разі запропонованого нами паралельного алгоритму час $S\,(N, P)$ є лінійною функцією від розміру задачі та кількості процесорів.

Позначивши через n розмір порції для одного завдання, можемо дати таку часову оцінку $S\,(N, P)$:

$$S(N,P) = \frac{N}{(P-1)\,n}\,c, \qquad (13)$$

де c - час, затрачений на одну комунікацію між процесами.

Справді, якщо враховувати, що в даному алгоритмі завдання роздаються і збираються порціями по n блукань, то для проведення всіх N випробувань потрібно $\frac{N}{(P-1)\,n}$ обмінів повідом-

леннями між керівним та виконавчими процесами. Час c, затрачуваний на один такий обмін, є постійним і йде на пересилку одного цілого числа та трьох чисел з плаваючою комою.

Оскільки робота над порціями завдань не вимагає жодної синхронізації між виконавчими процесами, то час, затрачений усією групою процесорів на паралельне проведення розрахунків, в P - 1 разів менший, ніж у випадку послідовної реалізації. Таким чином, оцінку часової складності паралельного алгоритму можна подати у вигляді:

$$T(N,P)=\frac{T(N)}{(P-1)}+\frac{N}{(P-1)n}c. \qquad (14)$$

Така часова складність є доволі хорошою і свідчить про високу ефективність від розпаралелювання, що цілком виправдовує додаткові ресурси, затрачені на розробку, реалізацію та запуск відповідного програмного продукту.

1. *Дехтярюк Е. С., Сипявський О. Л.* Розв'язання задач теорії пружності методом Монте-Карло // VIII Міжнародний конгрес з застосування математики у технічних науках. - Веймар, НДР, 1978.
2. *Sabelfeld K. K., Shalimova I. A.* Spherical means for PDEs. - VSP, The Netherlands, Utrecht, 1997.
3. *Варвак Л. П., Дехтярюк Е. С., Ройтфарб И. З., Синявский А. Л., Хименко В. В.* Решение пространственной задачи теории упругости методом статистических испытаний // Сб. «Сопротивление материалов и теория сооружений». - Вып. XX. - К.: Будівельник, 1973.
4. *Muller M. E.* (1956). Some continuous Monte Carlo method for the Dirichlet problem. Ann. Math. Stat., 27, (3), 569-589.
5. *Sabelfeld K. K.* Monte Carlo Methods in Boundary Value Problems. - Berlin: Springer-Verlag, 1991.

O. Syniavsky, O. Khomenko

A PARALLEL ALGORITHM OF SOLVING BORDER VALUE PROBLEMS FOR SYSTEMS OF DIFFERENTIAL EQUATIONS OF THE HIGHER ORDERS BY THE MONTE CARLO METHOD

A parallel algorithm for solving the first border value problem of elasticity theory is suggested. The problem exists in the case of an elastic body restrained over a part of its surface or closed within a rigid shell. In order to construct the Monte-Carlo algorithm, the mean value theorem, known in harmonic functions theory, was generalized. Hence, in our case the Lame equations have a correspondent integral relation between the displacement vector in the center of a sphere and the displacements on its surface [1]. The relation was obtained from the solution of the elastic sphere deformation problem, in which a unit force is focused in the center of the sphere with restrained border. Based on the relation, the solution of the problem was obtained in the form of a continual integral, which is estimated by means of the walk-on-spheres process.